

My Whole Blog

Contents

Definiendo sintaxis y comportamiento	4
Creando el template	5
A los bifos	7
Los parámetros	7
Entendiendo los Nodos	9
Entendienfo figure	9
Explicando el método run	10
Explicando el método <code>_render_svg</code>	11
El método <code>_resolve_render_name</code>	12
El método <code>_separate_content</code>	13
Pensando en sphinx	13
Y como se usa todo esto	14
Y funciona?	14
Concept	82
The binding side of things	85
What should a shell scripting language be like?	145
What doesn't matter?	146
Now, the example	147
Why not use reportlab directly?	285
So, how do you do a report?	286
Trickier: tables	287
Frobniz time measurements	288
How would it work?	288
Missing pieces	289

Introduction	369
1. Grok the Disqus API	370
2. Grok the Haloscan comments file	371
3. Create the necessary threads and whatever in Disqus	371
4. Post the comments from Haloscan to Disqus	372
5. Hack the blog so the links to Haloscan now work for Disqus	374
What the Ipad means	384
The User, In Linux	389
The User, in Windows	389
The Developer, in Linux	389
Windows for the developer	391
Problem 1: You need an installer	391
Problem 2: system libraries don't exist	393
Creating a plugin	396
Making the Plugin Configurable	397
Making it Work	398
Titanic	417
I'm Mark Shuttleworth!	418
Python vs. Ruby	418
Butter	418
Visa discount!	418
A Stable Deployment Target	478
Deployment Services	478
Monetization Services	478
The Store Itself	478
Intro	491
Chucherías Chinas	491
Cómo Pagar	492

El amigo que viene de afuera	492
Accesorios	493
Conclusión	494
Cooking is easy	498
Cooking is fun	500
Cooking is good for you	500
Cooking is cheap	501
Cooking is Applied Nerdiness	501
He's Ill-Defined	502
The Excluded Middle and Popularity	502
It's Unethical to Believe in Heaven and Hell	503
Conclusion	503
Proxy Support	524
Persistent Cookies	524
Using Properties and Signals in Object Creation	525
Download Manager	526
Printing	528
Other Tricks	528
Features	634
Bugfixes	635
Filters	636
Bundles	637
Conclusion	637
Hosting	670
Distribution	670
Web Server	670
Mail Server	671

Misc	672
Status	673
Docutils System Messages	686

Sorry, spanish only post!

Estimado público lector, una cosa muy especial: un post que no escribí yo! Démosle una bienvenida al autor invitado, Juan BC!

Una de mis [promesas](#) en el año fue terminar uno de los proyectos mas ambiciosos que me había propuesto: terminar mi ambientación de rol utilizando el sistema matemático que había diseñado junto con un amigo tiempo atras llamado [Cros](#) aprovechando la barbaridad de imagenes que tenía guardada de mi tesis de grado.

Como no puede ser de otra manera como miembro de Python Argentina utilice el programa utilizado por [rst2pdf](#) creado por [Roberto Alsina](#) el cual es el dueño de este blog

En un punto necesitaba hacer una especie de plantillas de dibujos para repetir el estilo donde hay descripciones de un personaje pero el estilo se mantiene, como por ejemplo las tarjetas de este manual de [DC Universe RPG](#) :

y me dije a mi mismo

¿Por que demonios no puedo dibujar un svg_ dejar espacios en blanco y luego los lleno dentro del svg?

En definitiva, me imaginaba algo así:

```
.. template_svg: path/a/mi/archivo/svg/que/tiene/las/variables/definidas/ac
   :nombre_variable_1: valor_de_variable_1
   :nombre_variable_2: valor_de_variable_2
   ...
   :nombre_variable_N: valor_de_variable_N
```

Entonces me puse en campaña para poder programar el código python que realiza esa tarea como una extensión de [rst2pdf](#).

AVISO!!! este texto lo llevara a usted por cada una de las desiciones dediseño que involucran la creación de mi extensión para rst2pdf

Definiendo sintaxis y comportamiento

Por empezar aprendí de las limitaciones de la directiva, decidí un nombre para ella y definí cual era el comportamiento que hiba a tener.

- Dado que una directiva como `template_svg` me sonaba a larga decidí que el nombre sería `svgt` (SVG template).
- Las directivas soportan parámetros fijos, no puedo tener un número variable de argumentos para pasarle: `nombre_variable_1`, `nombre_variable_2`, `variable_N` a algo como `directiva(**kwargs)`.

Mi solución fue pasarle todo en un json.

- Por último definí que `svgt` generaría un nodo de tipo `figure` con lo cual mi directiva aceptaría, además de mis parámetros, los argumentos que ya tenía incorporados en dicha directiva.

Esta última decisión me llevó también a contemplar que el archivo `svg` debería convertirse en un `png`, y para esto utilizaría la herramienta [Inkscape](#) con una llamada al sistema para realizar dicha conversión (la gran ventaja de `inkscape` es que puede correr totalmente headless con el parámetro `-z`).

En definitiva mi directiva tomaría un nodo así:

```
.. svgt:: file.svg
   :vars:
     { "nombre_variable_1": "valor_de_variable_1",
       "nombre_variable_2": "valor_de_variable_2",
       ...,
       "nombre_variable_N": "valor_de_variable_N" }

   <figure parameters>
```

Y lo convertiría en un nodo así

```
.. figure:: file_con_parametros_resueltos.png
   <figure parameters>
```

Creando el template

Yo uso para crear `svg` a [Inkscape](#) y no pensaba en ningún momento dejar de hacerlo para crear este proyecto. Así que solo era cuestión de abrir el programa y poner en la parte de donde se referencia a una url de una imagen o en un texto algún formato que indique que eso es una variable (recordemos que por dentro el `svg` no deja de ser texto plano).

Para elegir el lenguaje de template decidí utilizar el formato que propone la clase [Template](#) que viene en la librería estandar de; la cual dispone que las declaraciones de hacen anteponiendo el símbolo `$` al nombre de la variable y pudiendo o no este nombre estar encerrado entre `{ y }`.

Por ejemplo en el siguiente imagen se ve como declaro dos variables con [Inkscape](#)

Por otra parte inkscape desde consola se ejecuta de la siguiente manera para convertir un svg a png

```
:: $ inkscape -z file.svg -e file.png -d 300
```

Donde:

- **inkscape** es el comando para correr inkscape.
- **-z** sirve para deshabilitar el entorno gráfico.
- **file.svg** es el archivo a convertir.
- **-e file.png** indica que se va a exportar el archivo `file.svg` al formato `png` y se guardara en el archivo `file.png`.
- **-d 300** dice que el archivo `file.png` se creara con 300 dpi de calidad.

Con esto me genero los siguientes problemas:

- *¿Qué sucede si otro me pasa un template y no se cuales son las variables que tiene adentro?*
- *¿Y si inkscape no está en el path de ejecución?*
- *¿Y si no me gustan los 300 dpi de calidad?*

A este punto usted lector ya se dara cuenta que todo se trata de agregarle mas variables a nuestra sintaxis ya definida: con lo cual todo este cachibache quedaría así:

```
.. svgt:: file.svg
  :vars:
    { "nombre_variable_1": "valor_de_variable_1",
      "nombre_variable_2": "valor_de_variable_2",
      ...,
      "nombre_variable_N": "valor_de_variable_N"}
  :dpi: 72
  :inkscape_dir: /usr/bin
  :list_vars_and_exit:

  <figure parameters>
```

Siendo:

- **:dpi: 72** dice que el archivo generado para la figura resultante tendra 72dpi.
- **:inkscape_dir: /usr/bin** indica que el comando inkscape vive dentro de la carpeta `/usr/bin`
- **:list_vars_and_exit:** establece que svgt solo listara por `stdout` la lista de variables existente dentro de `file.svg` y luego el programa terminara (notese que solo tiene motivos de debugging).

En código python sólo hay que crear un string con esos huecos y luego llenarlos como en el siguiente ejemplo:

```

\ :kn:'import' \ \ :nn:'os' \
\ :n:'cmd' \ \ :o:'=' \ \ :s:'"{isp} {svg} -z -e {png} -d {dpi}" \
\ :k:'print' \ \ :n:'cmd' \ \ :o:'. ' \ \ :n:'format' \ \ :p:'(' \ \ :n:'isp' \
\ \ :n:'svg' \ \ :o:'=' \ \ :s:'"file.svg" \ \ :p:', ' \
\ \ :n:'png' \ \ :o:'=' \ \ :s:'"file.png" \ \ :p:', ' \
\ \ :n:'dpi' \ \ :o:'=' \ \ :s:'"72" \ \ :p:')' \

\ :p:'[ ' \ \ :n:'out' \ \ :p:']' \ \ :o: '/' \ \ :n:'usr' \ \ :o: '/' \ \ :nb:'bin

```

A los bifés

Ahora si leyendo el manual de como crear [directivas para docutils](#) uno se encuentra que el esqueleto mínimo para crear estos bichos es el siguiente:

```

\ :c:'# imports necesarios' \
\ :kn:'from' \ \ :nn:'docutils' \ \ :kn:'import' \ \ :n:'nodes' \
\ :kn:'from' \ \ :nn:'docutils.parsers.rst' \ \ :kn:'import' \ \ :n:'direct'
\ :kn:'from' \ \ :nn:'docutils.parsers.rst' \ \ :kn:'import' \ \ :n:'Direct
\ :kn:'from' \ \ :nn:'docutils.statemachine' \ \ :kn:'import' \ \ :n:'Strin

\ :c:'# la directiva es una clase' \
\ :k:'class' \ \ :nc:'SVGTemplate' \ \ :p:'(' \ \ :n:'Directive' \ \ :p:')': \
\ \ :sd:'"" The svg directive""' \

\ :c:'# cuantos argumentos obligatorios tenemos' \
\ :n:'required_arguments' \ \ :o:'=' \ \ :mi:'0' \

\ :c:'# cuantos argumentos opcionales' \
\ :n:'optional_arguments' \ \ :o:'=' \ \ :mi:'0' \

\ :c:'# si la directiva termina con una linea en blanco' \
\ :n:'final_argument_whitespace' \ \ :o:'=' \ \ :bp:'False' \

\ :c:'# funciones de validación para los argumentos' \
\ :n:'option_spec' \ \ :o:'=' \ \ :p:'{'' \

\ :c:'# si puede tener mas rst adentro de la directiva' \
\ :n:'has_content' \ \ :o:'=' \ \ :bp:'True' \

\ :c:'# método que hace el procesamiento' \
\ :k:'def' \ \ :nf:'run' \ \ :p:'(' \ \ :bp:'self' \ \ :p:')': \
\ \ :k:'return' \ \ :p:'[]' \

\ :c:'# le pone nombre a la directiva y la registra' \
\ :n:'directives' \ \ :o:'. ' \ \ :n:'register_directive' \ \ :p:'(' \ \ :s:'"sv

```

Los parámetros

Primero empecemos definiendo un diccionario que relaciona cada nombre de parámetro (los cuales yo defino que son TODOS opcionales) con una función que los valida. Por

otra parte dado la estructura que queremos generar de [figure](#) posee contenido y la nuestra también debe poseerlo.

```

\ :kn:'import'\ \ :nn:'json'\

\ :c:'# docutils imports'\

\ :n:'SPECS'\ \ :o:'='\ \ :p:'{'\

    \ :c:'# variables más y funciones que la validan y convierten en datos
    \ :c:'# para nuestro procesamiento'\

    \ :c:'# convierte vars a un json'\
    \ :s:'"vars"\ \ :p:':'\ \ :n:'json'\ \ :o:'.'\ \ :n:'loads'\ \ :p:','\
    \ :c:'# convierte dpi a entero positivo'\
    \ :s:'"dpi"\ \ :p:':'\ \ :n:'directives'\ \ :o:'.'\ \ :n:'nonnegative'\
    \ :c:'# preprocesa el directorio quitandole espacios finales e iniciales
    \ :c:'# en blanco'\
    \ :s:'"inkscape_dir"\ \ :p:':'\ \ :n:'directives'\ \ :o:'.'\ \ :n:'pa'\
    \ :c:'# se fija que por ser una bandera no tenga ni un valor asignado'\
    \ :c:'# si eso sucede retorna None sino tira un ValueError'\
    \ :s:'"list_vars_and_exit"\ \ :p:':'\ \ :n:'directives'\ \ :o:'.'\ \

    \ :c:'# variables de figure'\
    \ :s:'"alt"\ \ :p:':'\ \ :n:'directives'\ \ :o:'.'\ \ :n:'unchanged'\
    \ :s:'"height"\ \ :p:':'\ \ :n:'directives'\ \ :o:'.'\ \ :n:'length_c'\
    \ :s:'"width"\ \ :p:':'\ \ :n:'directives'\ \ :o:'.'\ \ :n:'length_or'\
    \ :s:'"scale"\ \ :p:':'\ \ :n:'directives'\ \ :o:'.'\ \ :n:'percentag'\
    \ :s:'"align"\ \ :p:':'\ \ :k:'lambda'\ \ :n:'align'\ \ :p:':'\ \ :'\
    \ :s:'"target"\ \ :p:':'\ \ :n:'directives'\ \ :o:'.'\ \ :n:'uri'\ \
    \ :s:'"class"\ \ :p:':'\ \ :n:'directives'\ \ :o:'.'\ \ :n:'unchanged'\
    \ :s:'"name"\ \ :p:':'\ \ :n:'directives'\ \ :o:'.'\ \ :n:'unchanged'\
    \ :s:'"figwidth"\ \ :p:':'\ \ :n:'directives'\ \ :o:'.'\ \ :n:'length'\
    \ :s:'"figclass"\ \ :p:':'\ \ :n:'directives'\ \ :o:'.'\ \ :n:'unchar'\
\ :p:'}'\

\ :c:'# la directiva es una clase'\
\ :k:'class'\ \ :nc:'SVGTemplate'\ \ :p:'('\ \ :n:'Directive'\ \ :p:'):''\
\ :sd:'"" The svgt directive""'\

\ :n:'required_arguments'\ \ :o:'='\ \ :mi:'0'\

\ :c:'# cuantos argumentos opcionales'\
\ :n:'optional_arguments'\ \ :o:'='\ \ :nb:'len'\ \ :p:'('\ \ :n:'SPE'\

\ :n:'final_argument_whitespace'\ \ :o:'='\ \ :bp:'False'\

\ :c:'# funciones de validacion para los argumentos'\
\ :n:'option_spec'\ \ :o:'='\ \ :n:'SPECS'\

```

```

\ :n: 'has_content' \ \ :o: '=' \ \ :bp: 'True' \

\ :k: 'def' \ \ :nf: 'run' \ \ :p: '(' \ \ :bp: 'self' \ \ :p: ')': \
\ :k: 'return' \ \ :p: '[]' \

\ :n: 'directives' \ \ :o: '.' \ \ :n: 'register_directive' \ \ :p: '(' \ \ :s: '"sv

```

Todas las funciones de validación (menos `json.load`) están muy bien explicadas en la [documentación de docutils](#).

Entendiendo los *Nodos*

La creación de nodos se hace con funciones que habitan en el módulo `from docutils import nodes` y todos los nodos se crean así (tomando de ejemplo [figure](#)):

```

\ :n: 'my_new_node' \ \ :o: '=' \ \ :n: 'figure' \ \ :p: '(' \ \ :bp: 'self' \ \ :p: ')': \

```

donde:

- **rawsource:** es el código fuente del nodo que sirve para motivos de debugging, si salta un error te muestra donde está la falla basándose en este string. Así que mientras más representa al nodo este argumento mejor.
- ***children:** son los nodos hijos.
- ****attribute:** los atributos y opciones del nuevo nodo. si se pasa una opción inválida simplemente se ignora.

Entendiendo *figure*

Las figuras son un nodo que contiene 3 hijos:

- [image](#) que contiene la imagen propiamente dicha.
- [caption](#) que es un párrafo que le da una etiqueta a la imagen.
- [legend](#) que contiene todo los párrafos sobrantes que no son el [caption](#)

Osea algo así:

```

+-----+
|           figure           |
|                             |
|<----- figwidth ----->|
|                             |
| +-----+                 |
| |           image           | |
| |                             | |
| |<--- width ----->| |
| +-----+                 |
|                             |
|The figure's caption should|
|wrap at this width.       |

```

```

|                                     |
| Legend                             |
+-----+

```

Con esto en mente y con el concepto de que **TODO NODO DEBE SER PROCESADO ODESTRUIDO MANUALMENTE** vamos ya a encarar el algoritmo.

Explicando el método run

La función run sólo tiene una condición: **debe devolver un lista que contiene nodos de docutils a ser renderizados**

Así que a mano alzada run sería algo mas o menos así:

```

\ :k:'def'\ \ :nf:'run'\ \ :p:'('\ \ :bp:'self'\ \ :p:'):\
\ :c:'# sacamos la direccion donde se encuentra el svg'\
\ :n:'uri'\ \ :o:'='\ \ :bp:'self'\ \ :o:'.'\ \ :n:'arguments'\ \ :p:
\ :c:'# extraemos nuestras variables y les asignamos un valor por defec
\ :c:'# caso de no existir'\
\ :n:'options'\ \ :o:'='\ \ :nb:'dict'\ \ :p:'('\ \ :bp:'self'\ \ :o:
\ :n:'svgt_vars'\ \ :o:'='\ \ :n:'options'\ \ :o:'.'\ \ :n:'pop'\ \ :
\ :n:'svgt_dpi'\ \ :o:'='\ \ :n:'options'\ \ :o:'.'\ \ :n:'pop'\ \ :p
\ :n:'svgt_isd'\ \ :o:'='\ \ :n:'options'\ \ :o:'.'\ \ :n:'pop'\ \ :p
\ :n:'svgt_lvae'\ \ :o:'='\ \ :n:'options'\ \ :o:'.'\ \ :n:'pop'\ \ :
\ :k:'if'\ \ :s:'"list_vars_and_exit"'\ \ :ow:'in'\ \ :n:
\ :c:'# si tenemos seteado el flag list_vars_and_exit mostramos las var
\ :c:'# y salimos del programa'\
\ :k:'if'\ \ :n:'svgt_lvae'\ \ :p:':'\
\ :bp:'self'\ \ :o:'.'\ \ :n:'_show_vars'\ \ :p:'('\ \ :n:'uri'\ \
\ :n:'sys'\ \ :o:'.'\ \ :n:'exit'\ \ :p:'('\ \ :mi:'0'\ \ :p:')'\
\ :c:'# por como esta diseñado figure hay que evitar que la propiedad a
\ :c:'# le llegue a su imagen interior.'\
\ :n:'fig_align'\ \ :o:'='\ \ :n:'options'\ \ :o:'.'\ \ :n:'pop'\ \ :
\ :c:'# pasamos a crear el archivo png'\
\ :n:'png_path'\ \ :o:'='\ \ :bp:'self'\ \ :o:'.'\ \ :n:'_render_svg'
\ :c:'# agremamos la uri del png a las opciones'\
\ :n:'options'\ \ :p:'[\ \ :s:'"uri"'\ \ :p:]'\ \ :o:'='\ \ :n:'png'
\ :c:'# creamos el nodo imagen'\
\ :n:'image_node'\ \ :o:'='\ \ :n:'nodes'\ \ :o:'.'\ \ :n:'image'\ \
\ :c:'# el contenido de caption y legend viene todo mezclado'\
\ :c:'# como un iterable llamado docutils.statemachine.StringList'\
\ :c:'# hay que separarlo en dos partes para crear la figure.'\
\ :n:'caption_content'\ \ :p:','\ \ :n:'legend_content'\ \ :o:'='\ \

```

```

\ :c:\# creamos el nodo caption procesando su contenido con\
\ :c:\# nested_parse\
\ :n:\caption_node\ \ :o:='\ \ :n:\nodes\ \ :o:'.\ \ :n:\caption
\ :bp:\self\ \ :o:'.\ \ :n:\state\ \ :o:'.\ \ :n:\nested_parse\ \

\ :c:\# creamos el nodo legend y procesamos su contenido\
\ :n:\legend_node\ \ :o:='\ \ :n:\nodes\ \ :o:'.\ \ :n:\legend\
\ :bp:\self\ \ :o:'.\ \ :n:\state\ \ :o:'.\ \ :n:\nested_parse\ \

\ :c:\# restauramos la variable align para crear el nodo figure\
\ :k:\if\ \ :n:\fig_align\ \ :o:!='\ \ :bp:\None\ \ :p:':\
\ :n:\options\ \ :p:['\ \ :s:"align"\ \ :p:']\ \ :o:='\ \ :n:
\ :c:\# creamos el susodicho nodo figure pasandole sus hijos\
\ :n:\figure_node\ \ :o:='\ \ :n:\nodes\ \ :o:'.\ \ :n:\figure\
\ :n:\caption_node\ \ :p:',\ \ :n:\legend

\ :c:\# retornamos una lista con el nodo resultante\
\ :k:\return\ \ :p:['\ \ :n:\figure_node\ \ :p:']\

```

Explicando el método `_render_svg`

Este método utiliza otros dos, `_resolve_render_name` que sera explicado después y el método `_call` que debido a su extrema simplicidad se recomienda leer directamente la documentación del clase [subprocess.Popen](#)

```

\ :n:\CMD\ \ :o:='\ \ :s:"{isp} {svg} -z -e {png} -d {dpi}"\

\ :c:\# donde:\
\ :c:\# - uri es la direccion donde esta el svg a renderizar\
\ :c:\# - svgt_isd es el lugar donde esta inkscape\
\ :c:\# - svgt_dpi son los dpi del png a renderizar\
\ :c:\# - svgt_vars es un diccionario que contiene los valores de las variables
\ :k:\def\ \ :nf:\_render_svg\ \ :p:('\ \ :bp:\self\ \ :p:',\ \ :n:

\ :c:\# abrimos el svg y lo cargamos en un Template\
\ :k:\with\ \ :nb:\open\ \ :p:('\ \ :n:\uri\ \ :p:')\ \ :k:\as\
\ :n:\svg_tplt\ \ :o:='\ \ :n:\string\ \ :o:'.\ \ :n:\Templat

\ :c:\# reemplazamos las variables con sus valores\
\ :n:\svg_src\ \ :o:='\ \ :n:\svg_tplt\ \ :o:'.\ \ :n:\safe_subst

\ :c:\# obtenemos paths donde guardar:\
\ :c:\# - El svg con las variables reemplazadas (fname_svg)\
\ :c:\# - El png resultante (fname_png)\
\ :n:\fname_svg\ \ :p:',\ \ :n:\fname_png\ \ :o:='\ \ :bp:\self

\ :c:\# guardamos el svg reemplazado en su lugar\
\ :k:\with\ \ :nb:\open\ \ :p:('\ \ :n:\fname_svg\ \ :p:',\ \ :n:s

```

```

    \ :n:'fp'\ \ :o:'. '\ \ :n:'write'\ \ :p:'('\ \ :n:'svg_src'\ \ :p:

\ :c:'# Ponemos los parametros de ejecucion de inkscape'\
\ :n:'cmd'\ \ :o:'='\ \ :n:'CMD'\ \ :o:'. '\ \ :n:'format'\ \ :p:'('\
    \ :n:'svg'\ \ :o:'='\ \ :n:'fname_svg'\ \ :p:', '\
    \ :n:'png'\ \ :o:'='\ \ :n:'fname_png'\ \ :p:', '\
    \ :n:'dpi'\ \ :o:'='\ \ :n:'svgt_dpi'\ \ :p:') '\

\ :c:'# ejecutamos inkscape'\
\ :bp:'self'\ \ :o:'. '\ \ :n:'_call'\ \ :p:'('\ \ :n:'cmd'\ \ :p:') '\

\ :c:'# retornamos la direccion del nuevo png'\
\ :k:'return'\ \ :n:'fname_png'\

```

El método `_resolve_render_name`

Este método a primera vista es innecesario, ya que podríamos generar archivos temporales eficientemente con el modulo `_template`, pero dado como trabaja sphinx esto no es posible de primera mano.

```

\ :c:'# uri es la direccioón del svg original'\
\ :k:'def'\ \ :nf:'_resolve_render_name'\ \ :p:'('\ \ :bp:'self'\ \ :p:',

\ :c:'# leemos el directorio temporal de la variable global _tempdir'\
\ :c:'# que pudo haber sido modificada por sphinx o tomamos el dir temp
\ :c:'# del sistema operativo. Esto se debe a que sphinx corre en un sa
\ :c:'# y todos los archivos deben estar en la misma carpeta donde se'\
\ :c:'# encuentra el archivo de configuración conf.py'\
\ :n:'tempdir'\ \ :o:'='\ \ :n:'_tempdir'\ \ :k:'if'\ \ :n:'_tempdi

\ :c:'# extraemos el nombre del archivo sin su extensión'\
\ :n:'basename'\ \ :o:'='\ \ :n:'os'\ \ :o:'. '\ \ :n:'path'\ \ :o:'.

\ :c:'# agregamos al directorio temporal el nombre del archivo sin la'\
\ :c:'# extensión'\
\ :n:'render_name'\ \ :o:'='\ \ :n:'os'\ \ :o:'. '\ \ :n:'path'\ \ :o:

\ :c:'# generamos un nuevo nombre para el archivo svg agregando un ente
\ :c:'# al final para evitar coliciones'\
\ :n:'idx'\ \ :o:'='\ \ :s:''''\
\ :n:'unique_svg'\ \ :o:'='\ \ :n:'render_name'\ \ :o:'+'\ \ :s:''''\
\ :k:'while'\ \ :n:'os'\ \ :o:'. '\ \ :n:'path'\ \ :o:'. '\ \ :n:'exists
    \ :n:'idx'\ \ :o:'='\ \ :n:'idx'\ \ :o:'+'\ \ :mi:'1'\ \ :k:'if
\ :n:'unique_svg'\ \ :o:'='\ \ :n:'unique_svg'\ \ :o:'. '\ \ :n:'forma

\ :c:'# lo mismo para el png'\
\ :n:'idx'\ \ :o:'='\ \ :s:''''\
\ :n:'unique_png'\ \ :o:'='\ \ :n:'render_name'\ \ :o:'+'\ \ :s:''''\
\ :k:'while'\ \ :n:'os'\ \ :o:'. '\ \ :n:'path'\ \ :o:'. '\ \ :n:'exists
    \ :n:'idx'\ \ :o:'='\ \ :n:'idx'\ \ :o:'+'\ \ :mi:'1'\ \ :k:'if

```

```

\ :n:'unique_png'\ \ :o:'='\ \ :n:'unique_png'\ \ :o:'.'\ \ :n:'forma

\ :c:'# retornamos los dos valores'\
\ :k:'return'\ \ :n:'unique_svg'\ \ :p:','\ \ :n:'unique_png'\

```

El método `_separate_content`

Este es el último método que utiliza el método `run` y es lo último que necesitamos para hacer el trabajo de `svgt`

```

\ :c:'# recibe por parámetro un string list con todos los nodos de texto de
\ :c:'# contenido de svgt'\
\ :k:'def'\ \ :nf:'_separate_content'\ \ :p:'('\ \ :bp:'self'\ \ :p:','\

\ :c:'# primero creamos un nodo stringlist exclusivamente para el capti
\ :n:'caption_cnt'\ \ :o:'='\ \ :n:'StringList'\ \ :p:'('\ \ :n:'pare
\ :n:'parent_offset'\ \ :o:'='\ \ :n:'content'\

\ :c:'# ahora algo IMPORTANTE.. la leyenda va a ser el mismo nodo conte
\ :c:'# por que la otra alternativa es crear uno nuevo y borrar el vie
\ :c:'# por que TODO NODO TIENE QUE SER PROCESADO O ELIMINADO MANUALMEN
\ :n:'legend_cnt'\ \ :o:'='\ \ :n:'content'\ \ :c:'# all nodes need

\ :c:'# si tenemos contenido copiamos el primer elemento al caption y c
\ :c:'# ese elemento de la leyenda (que es lo mismo que el content)'\
\ :k:'if'\ \ :n:'content'\ \ :p:':'\
\ :n:'caption_cnt'\ \ :o:'.'\ \ :n:'append'\ \ :p:'('\ \ :n:'conter
\ :n:'content'\ \ :o:'.'\ \ :n:'pop'\ \ :p:'('\ \ :mi:'0'\ \ :p:')
\ :k:'return'\ \ :n:'caption_cnt'\ \ :p:','\ \ :n:'legend_cnt'\

```

Pensando en sphinx

Para que este bicho funcione en [sphinx](#) es necesario agregar a nivel de modulo una función `setup` que recibe un único parámetro que es la instancia de [sphinx](#) corriendo.

```

\ :k:'def'\ \ :nf:'setup'\ \ :p:'('\ \ :n:'app'\ \ :p:'):'\

\ :c:'# toma el valor la referencia a la variable de módulo _tempdir'\
\ :k:'global'\ \ :n:'_tempdir'\

\ :c:'# esta función reci be dos parámetros requeridos para usarse como
\ :c:'# slot de la señal build-finished de sphinx y sirve para limpiar
\ :c:'# el directorio temporal. '\
\ :c:'# - la aplicación que ejecuto la señal '\
\ :c:'# - la exception que genero el fin del procesamiento o None si
\ :c:'# finalizo normalmente. '\
\ :k:'def'\ \ :nf:'reset'\ \ :p:'('\ \ :n:'app'\ \ :p:','\ \ :n:'ex'\
\ :k:'if'\ \ :n:'_tempdir'\ \ :o:'!='\ \ :bp:'None'\ \ :ow:'and

```

```

        \ :n:'shutil'\ \ :o:'.'\ \ :n:'rmtree'\ \ :p:'('\ \ :n:'_tempdir'
    \ :k:'if'\ \ :n:'ex'\ \ :p:':'\
        \ :k:'raise'\ \ :n:'ex'\

\ :c:'# agregamos una configuracion mas a sphinx llamada tempdir y asiq
\ :c:'# como valor por defecto _temp'\
\ :n:'app'\ \ :o:'.'\ \ :n:'add_config_value'\ \ :p:'('\ \ :s:"tempdir"

\ :c:'# tomamos el valor de tempdir de la configuración'\
\ :n:'_tempdir'\ \ :o:'='\ \ :n:'app'\ \ :o:'.'\ \ :n:'config'\ \ :p:

\ :c:'# reiniciamos el directorio temporal'\
\ :n:'reset'\ \ :p:'('\ \ :n:'app'\ \ :p:','\ \ :bp:'None'\ \ :p:')'\

\ :c:'# lo volvemos a crear'\
\ :n:'os'\ \ :o:'.'\ \ :n:'mkdir'\ \ :p:'('\ \ :n:'_tempdir'\ \ :p:')'\

\ :c:'# conectamos la señal build-finished con la función reset'\
\ :n:'app'\ \ :o:'.'\ \ :n:'connect'\ \ :p:'('\ \ :s:"build-finished"

```

Si desean mas información lean la documentación del [api de extensiones de sphinx](#)

Y como se usa todo esto

Bueno con rst2pdf o bien lo tiran en la carpeta de extensiones, o en el path donde tienen su rst y luego ejecutan

```
\ :nv:'$ \ rst2pdf archivo.rst -e svgt
```

También pueden instalarlo desde pypi con easy_install o pip con los comandos:

```
\ :nv:'$ \ pip install docutils_ext
```

o

```
\ :nv:'$ \ easy_install docutils_ext
```

En el caso de sphinx, tienen que agregar el path donde se encuentre a sys.path y luego agregarla a la lista de extensiones.

Y funciona?

Pueden ver el código completo al momento de la publicación de este artículo [aca](#) además de poder descargar la última versión estable en la pestañita de downloads.

Lo de aca abajo dice

```
.. svgt:: img/temp.svg
   :vars: {"name": "that's all folks", "url": "img/troll.png"}
```

Yes, after many moons, it's out. Here is the (as usual) incomplete changelog:

- Added raw HTML support, by Dimitri Christodoulou

- Fixed Issue 422: Having no .afm files made font lookup slow.
- Fixed Issue 411: Sometimes the windows registry has the font's abspath.
- Fixed Issue 430: Using --config option caused other options to be ignored (by charles at cstanhope dot com)
- Fixed Issue 436: Add pdf_style_path to sphinx (by tyler@datastax.com)
- Fixed Issue 428: page numbers logged as errors
- Added support for many pygments options in code-block (by Joaquin Soriano)
- Implemented Issue 404: plantuml support
- Issue 399: support sphinx's template path option
- Fixed Issue 406: calls to the wrong logging function
- Implemented Issue 391: New --section-header-depth option.
- Fixed Issue 390: the --config option was ignored.
- Added support for many pygments options in code-block (by Joaquin Soriano)
- Fixed Issue 379: Wrong style applied to paragraphs in definitions.
- Fixed Issue 378: Multiline :address: were shown collapsed.
- Implemented Issue 11: FrameBreak (and conditional FrameBreak)
- The description of frames in page templates was just wrong.
- Fixed Issue 374: in some cases, literal blocks were split inside a page, or the pagebreak came too early.
- Fixed Issue 370: warning about sphinx.addnodes.highlightlang not being handled removed.
- Fixed Issue 369: crash in hyphenator when specifying "en" as a language.
- Compatibility fix to Sphinx 0.6.x (For python 2.7 docs)

This release did not focus on Sphinx bugs, so those are probably still there. Hopefully the next round is attacking those.

Turns out there was a major, show stopper bug in rst2pdf 0.90: sphinx support was absolutely broken. Not broken as in buggy, broken as in it had syntax errors.

system-message

WARNING/2 in `everything.txt`, line 3
 Duplicate explicit target name: "0.91 is now released".

So, [0.91 is now released](http://rst2pdf.googlecode.com).. 0.91 is now released: <http://rst2pdf.googlecode.com>, and only broken in the traditional sense. Enjoy!
In 1003, Erik the Red died after having ruled Greenland. Which of course, was a bad idea, color-combination-wise.

Figure 1: Article 1003 at piachepiu.com.ar

Also a bad idea was decree 1003/89 which pardoned every “subversive” or “terrorist”, including those dead or disappeared. This is after decree 1002, that pardons those who killed or disappeared them, of course.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “decree 1003/98”.
```

Much milder is [decree 1003/98](http://www.gob.gba.gov.ar/legislacion/legislacion/98-1003.html).. decree 1003/98: <http://www.gob.gba.gov.ar/legislacion/legislacion/98-1003.html>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “devicenzo”.
```

which gives an award to Roberto DeVicenzo. I wrote a program called [devicenzo](http://devicenzo.googlecode.com).. devicenzo: <http://devicenzo.googlecode.com>

, BTW.

Figure 2: Car 1003 of the 56 bus.

Bug #1003 in launchpad is about launchpad, the title is “projects +search page 404” and the description is “Page not found Page not found”. It’s fixed.

If you search for 1003 in Google Maps, it gives you *one result*, in Latvia.

But Bing gives you another place, in Lausanne.

1003 days ago, it was september 15th, 2002, the day of my 31st birthday.

Figure 3: USS Albatross, SP-1003

Page 1003 of Gray's anatomy is about the development of the eye and contains a drawing of "the eye of an eighteen days' embryo rabbit."

Figure 4: "1003 Rabbit Skins Snap Bib" (scary!)

Page 1003 of "Familiar Quotations, 10th ed of 1919" says "We are dancing on a volcano.", attributed to the Comte de Salvandy. Alto del Zapote, in Cuba is 1003 meters tall, but is not a volcano at all.

Figure 5: NGC 1003, a diffuse spiral galaxy in Perseus.

Oh, and this blog has 1003 posts.

'Thinner,' the old Gypsy man with the rotting nose whispers...

—Richard Bachman (Stephen King)

I was not always fat. I used to be really, really thin. So thin that when I was 16, my mother forced me to eat. I was 1.75 and weighted about 65 kilos when I finished highschool. So thin I used M sized shirts until I was 25.

That has, as those who know me can agree with, changed quite a bit. I broke the 100kg barrier 10 years ago or so, and reached 123.5Kg earlier this week. Also, I am not 3.5 meters tall, so I am close to having doubled my weight and body mass in these 25 years.

That is not a good thing. It's such a bad thing, that my doctor has explained to me that if I don't lose a lot of weight soonish, I am going to fucking go and die. Not next week, not next year, but not in 40 years either.

Since my weight started going up while I was in college and my dad was sick, I always blamed some anxiety problem, eating crap, a sedentary lifestyle, the usual suspects.

So, I got some tests done. Turns out I was more or less right. The “more” is because I do have to stop eating crap, and I need to exercise more. The “less” is because I have (among other things) hyperinsulinemia. Wanna guess what are the most visible symptom of that?

- High blood pressure (on medication since 5 years ago)
- Increased VLDL (diagnosed 2 years ago)
- Lethargy (fuck yes)
- Weight gain (double fuck yes)

And what does weight gain of this kind do to you? A lot of other bad things.

Since tuesday I am on a very specific diet, and taking a drug to reduce my insulin production (Metformin). I am feeling active, and have lost 3 kilos in 4 days (yes, I know that is not a sustainable rate and will plateau).

My feet stopped swelling.

I am not hungry all day.

I am walking 30 minutes, twice a day.

I want to code.

I feel *good*. I have felt *better*, when I started taking BP meds, I have felt *worse* when my liver function decreased, I have felt very *bad*, when my BP spiked, but *good*? I have not felt good in a very, very long time.

This may not be the drug or the diet. Maybe it’s placebo effect. Maybe it’s something else. On the other hand, I have decided that my life is too sweet to drop dead right now. So, let’s see how this goes.

Figure 6: This post is a joke.

Dr. Cooper, I hope this letter finds you in good health. I couldn’t avoid overhearing you talk with Dr. Hofstadter on our building’s staircase the other day.

Specifically when you mused about how life would be if people evolved from reptiles. I am *so* disappointed in you.

First, you mention that lizards are cold-blooded. Which is true. And that when it’s cold they get slower. Which is also true. But then you sayd something like “the weatherlizard would say ‘it’s slow outside’ instead of ‘it’s cold’”.

POPPYCOCK Dr. Cooper! If the lizard is slow because it’s cold, it would perceive everything out there as *fast*, not slow, just like slow cars see faster cars as, you know... *fast*?

Also, the mention about suggesting the lizard should wear a sweater is a slap on the face of physics. Sweaters are an insulator, not an energy source. What makes the

inside of the sweater warm is the human, Dr. Cold-blooded lizards would have no such effect beyond the tiny thermal inertia such an imperfect wool insulator would allow.

If you are interested on further argument in human-like reptile civilization and folklore I will be happy to indulge, but I must say I expected better of you.

Mrs. Vartabedian.

PS: things like this are the reason why I never ask *you* to come home for cereal.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "ubuntu one".

One of the nice things about working at Canonical is that we produce open source software. I, specifically, work in the team that does the desktop clients for [Ubuntu One](https://one.ubuntu.com).
ubuntu one: <https://one.ubuntu.com>

which is a really cool job, and a really cool piece

system-message

WARNING/2 in `everything.txt`, line 886
Block quote ends without a blank line; unexpected unindent. backrefs:

of software. However, one thing not enough people know, is that we offer damn nice APIs for developers. We have to, since all our client code is open source, so we need those APIs for ourselves.

So, here is a small tutorial about using some of those APIs. I did it using Python and PyQt for several reasons:

- Both are great tools for prototyping
- Both have good support for the required stuff (DBus, HTTP, OAuth)
- It's what I know and enjoy. Since I did this code on a sunday, I am not going to use other things.

Having said that, there is nothing python-specific or Qt-specific in the code. Where I do a HTTP request using QtNetwork, you are free to use libsoup, or whatever.

So, on to the nuts and bolts. The main pieces of Ubuntu One, from a infrastructure perspective, are Ubuntu SSO Client, that handles user registration and login, and SyncDaemon, which handles file synchronization.

To interact with them, on Linux, they offer DBus interfaces. So, for example, this is a fragment of code showing a way to get the Ubuntu One credentials (this would normally be part of an object's `__init__`):

```
\ :c:`# Get the session bus`\
\ :n:`bus`\ \ :o:`='\ \ :n:`dbus`\ \ :o:`.`\ \ :n:`SessionBus`\ \ :p:`()

\ :p:`:`\
\ :p:`:`\
\ :p:`:`\

\ :c:`# Get the credentials proxy and interface`\
```

```

\ :bp:'self' \ :o:'. ' \ :n:'creds_proxy' \ :o:'=' \ :n:'bus' \ :o:'. '
\ :s:'/credentials' \ :p:', '
\ :n:'follow_name_owner_changes' \ :o:'=' \ :bp:

\ :c:'# Connect to signals so you get a call when something'
\ :c:'# credential-related happens'
\ :bp:'self' \ :o:'. ' \ :n:'creds_iface' \ :o:'=' \ :n:'dbus' \ :o:'. '
\ :s:'com.ubuntuone.CredentialsManagement' \ :p:') '
\ :bp:'self' \ :o:'. ' \ :n:'creds_proxy' \ :o:'. ' \ :n:'connect_to_sigh
\ :bp:'self' \ :o:'. ' \ :n:'creds_found' \ :p:') '
\ :bp:'self' \ :o:'. ' \ :n:'creds_proxy' \ :o:'. ' \ :n:'connect_to_sigh
\ :bp:'self' \ :o:'. ' \ :n:'creds_not_found' \ :p:') '
\ :bp:'self' \ :o:'. ' \ :n:'creds_proxy' \ :o:'. ' \ :n:'connect_to_sigh
\ :bp:'self' \ :o:'. ' \ :n:'creds_error' \ :p:') '

\ :c:'# Call for credentials'
\ :bp:'self' \ :o:'. ' \ :n:'_credentials' \ :o:'=' \ :bp:'None'
\ :bp:'self' \ :o:'. ' \ :n:'get_credentials' \ :p:'()'

```

You may have noticed that `get_credentials` doesn't actually *return* the credentials. What it does is, it tells `SyncDaemon` to fetch the credentials, and then, when/if they are there, one of the signals will be emitted, and one of the connected methods will be called. This is nice, because it means you don't have to worry about your app blocking while `SyncDaemon` is doing all this.

But what's in those methods we used? Not much, really!

```

\ :k:'def' \ :nf:'get_credentials' \ :p:'(\ \ :bp:'self' \ :p:):'
\ :c:'# Do we have them already? If not, get'em'
\ :k:'if' \ :ow:'not' \ :bp:'self' \ :o:'. ' \ :n:'_credentials'
\ :bp:'self' \ :o:'. ' \ :n:'creds_proxy' \ :o:'. ' \ :n:'find_c
\ :c:'# Return what we've got, could be None'
\ :k:'return' \ :bp:'self' \ :o:'. ' \ :n:'_credentials'

\ :k:'def' \ :nf:'creds_found' \ :p:'(\ \ :bp:'self' \ :p:', ' \ :n:
\ :c:'# Received credentials, save them.'
\ :k:'print' \ :s:'creds_found' \ :p:', ' \ :n:'data'
\ :bp:'self' \ :o:'. ' \ :n:'_credentials' \ :o:'=' \ :n:'data'
\ :c:'# Don't worry about get_quota yet ;-)'
\ :k:'if' \ :ow:'not' \ :bp:'self' \ :o:'. ' \ :n:'_quota_info'
\ :bp:'self' \ :o:'. ' \ :n:'get_quota' \ :p:'()'

\ :k:'def' \ :nf:'creds_not_found' \ :p:'(\ \ :bp:'self' \ :p:', ' \
\ :c:'# No credentials, remove old ones.'
\ :k:'print' \ :s:'creds_not_found' \ :p:', ' \ :n:'data'
\ :bp:'self' \ :o:'. ' \ :n:'_credentials' \ :o:'=' \ :bp:'None'

\ :k:'def' \ :nf:'creds_error' \ :p:'(\ \ :bp:'self' \ :p:', ' \ :n:
\ :c:'# No credentials, remove old ones.'
\ :k:'print' \ :s:'creds_error' \ :p:', ' \ :n:'data'
\ :bp:'self' \ :o:'. ' \ :n:'_credentials' \ :o:'=' \ :bp:'None'

```

So, basically, `self._credentials` will hold a set of credentials, or `None`. Congratulations, we are now logged into Ubuntu One, so to speak.

So, let's do something useful! How about asking for how much free space there is in the account? For that, we can't use the local APIs, we have to connect to the servers, who are, after all, the ones who decide if you are over quota or not.

Access is controlled via OAuth. So, to access the API, we need to sign our requests. Here is how it's done. It's not particularly enlightening, and I did not write it, I just use it:

```

\ :k:'def' \ \ :nf:'sign_uri' \ \ :p:'(\ \ :bp:'self' \ \ :p:', \ \ \ :n:'uri'
  \ :c:'# Without credentials, return unsigned URL' \
  \ :k:'if' \ \ :ow:'not' \ \ :bp:'self' \ \ :o:'. \ \ \ :n:'_credentials' \
    \ :k:'return' \ \ \ :n:'uri' \
  \ :k:'if' \ \ :nb:'isinstance' \ \ :p:'(\ \ \ :n:'uri' \ \ :p:', \ \ \ :nb:
    \ :n:'uri' \ \ :o:'=' \ \ \ :nb:'bytes' \ \ :p:'(\ \ \ :n:'iri2uri' \ \
  \ :k:'print' \ \ :s:'"uri:"' \ \ :p:', \ \ \ :n:'uri' \
  \ :n:'method' \ \ :o:'=' \ \ \ :s:'"GET"' \
  \ :n:'credentials' \ \ :o:'=' \ \ \ :bp:'self' \ \ :o:'. \ \ \ :n:'_credenti
  \ :n:'consumer' \ \ :o:'=' \ \ \ :n:'oauth' \ \ :o:'. \ \ \ :n:'OAuthConsume
    \ :n:'credentials' \ \ :p:'[\ \ \ :s:'"con
  \ :n:'token' \ \ :o:'=' \ \ \ :n:'oauth' \ \ :o:'. \ \ \ :n:'OAuthToken' \ \
    \ :n:'credentials' \ \ :p:'[\ \ \ :s:'"token_seo
  \ :k:'if' \ \ :ow:'not' \ \ \ :n:'parameters' \ \ :p:': \
    \ :n:'_' \ \ :p:', \ \ \ :n:'_' \ \ :p:', \ \ \ :n:'_' \ \ :p:', \ \ \ :n:
    \ :n:'parameters' \ \ :o:'=' \ \ \ :nb:'dict' \ \ :p:'(\ \ \ :n:'cgi' \
  \ :n:'request' \ \ :o:'=' \ \ \ :n:'oauth' \ \ :o:'. \ \ \ :n:'OAuthRequest
    \ :n:'http_url' \ \ :o:'=' \ \ \ :n:'ur
    \ :n:'http_method' \ \ :o:'=' \ \ \ :n:
    \ :n:'parameters' \ \ :o:'=' \ \ \ :n:
    \ :n:'oauth_consumer' \ \ :o:'=' \ \ \
    \ :n:'token' \ \ :o:'=' \ \ \ :n:'token

  \ :n:'sig_method' \ \ :o:'=' \ \ \ :n:'oauth' \ \ :o:'. \ \ \ :n:'OAuthSigna
  \ :n:'request' \ \ :o:'. \ \ \ :n:'sign_request' \ \ :p:'(\ \ \ :n:'sig_meth
  \ :k:'print' \ \ :s:'"SIGNED:"' \ \ :p:', \ \ \ :nb:'repr' \ \ :p:'(\ \ \ :
  \ :k:'return' \ \ \ :n:'request' \ \ :o:'. \ \ \ :n:'to_url' \ \ :p:'()' \

```

And how do we ask for the quota usage? By accessing the <https://one.ubuntu.com/api/quota/> entry point with the proper authorization, we would get a JSON dictionary with `total` and `used` space. So, here's a simple way to do it:

```

\ :c:'# This is on __init__' \
  \ :bp:'self' \ \ :o:'. \ \ \ :n:'nam' \ \ :o:'=' \ \ \ :n:'QtNetwork' \ \ :o:
    \ :n:'finished' \ \ :o:'=' \ \ \ :bp:'self' \ \ :o:'. \ \ \ :n:'reply_fini

  \ :p:': \
  \ :p:': \
  \ :p:': \

  \ :k:'def' \ \ :nf:'get_quota' \ \ :p:'(\ \ \ :bp:'self' \ \ :p:'): \
    \ :sd:'""Launch quota info request.""' \

```

```

\ :n: 'uri' \ \ :o: '=' \ \ :bp: 'self' \ \ :o: '.' \ \ :n: 'sign_uri' \ \ :p:
\ :n: 'url' \ \ :o: '=' \ \ :n: 'QtCore' \ \ :o: '.' \ \ :n: 'QUrl' \ \ :p: '(' \
\ :n: 'url' \ \ :o: '.' \ \ :n: 'setEncodedUrl' \ \ :p: '(' \ \ :n: 'uri' \ \ :p:
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'nam' \ \ :o: '.' \ \ :n: 'get' \ \ :p: '(' \

```

Again, see how `get_quota` doesn't *return* the quota? What happens is that `get_quota` will launch a HTTP request to the Ubuntu One servers, which will, eventually, reply with the data. You don't want your app to block while you do that. So, `QNetworkAccessManager` will call `self.reply_finished` when it gets the response:

```

\ :k: 'def' \ \ :nf: 'reply_finished' \ \ :p: '(' \ \ :bp: 'self' \ \ :p: ', ' \ \ :
\ :k: 'if' \ \ :nb: 'unicode' \ \ :p: '(' \ \ :n: 'reply' \ \ :o: '.' \ \ :n: 'ur
\ :c: '# Handle quota responses' \
\ :bp: 'self' \ \ :o: '.' \ \ :n: '_quota_info' \ \ :o: '=' \ \ :n: 'json'
\ :k: 'print' \ \ :s: '"Got quota: "' \ \ :p: ', ' \ \ :bp: 'self' \ \ :o:
\ :c: '# Again, don't worry about update_menu yet ;-)' \
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'update_menu' \ \ :p: '()' \

```

What else would be nice to have? How about getting a call whenever the status of `syncdaemon` changes? For example, when `sync` is up to date, or when you get disconnected? Again, those are Dbus signals we are connecting in our `__init__`:

```

\ :bp: 'self' \ \ :o: '.' \ \ :n: 'status_proxy' \ \ :o: '=' \ \ :n: 'bus' \ \ :o:
\ :s: 'com.ubuntuone.SyncDaemon' \ \ :p: ', ' \ \ :s: '/status' \ \ :p: ')'
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'status_iface' \ \ :o: '=' \ \ :n: 'dbus' \ \ :o:
\ :n: 'dbus_interface' \ \ :o: '=' \ \ :s: 'com.ubuntuone.SyncDaemon.Status'
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'status_iface' \ \ :o: '.' \ \ :n: 'connect_to_sig
\ :s: 'StatusChanged' \ \ :p: ', ' \ \ :bp: 'self' \ \ :o: '.' \ \ :n: 'status

\ :c: '# Get the status as of right now' \
\ :bp: 'self' \ \ :o: '.' \ \ :n: '_last_status' \ \ :o: '=' \ \ :bp: 'self' \ \ :o:
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'status_proxy' \ \ :o: '.' \ \ :n: 'current_st

```

And what's `status_changed`?

```

\ :k: 'def' \ \ :nf: 'status_changed' \ \ :p: '(' \ \ :bp: 'self' \ \ :p: ', ' \ \ :
\ :k: 'print' \ \ :s: '"New status: "' \ \ :p: ', ' \ \ :n: 'status' \
\ :bp: 'self' \ \ :o: '.' \ \ :n: '_last_status' \ \ :o: '=' \ \ :bp: 'self' \
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'update_menu' \ \ :p: '()' \

```

The `process_status` function is boring code to convert the info from `syncdaemon`'s status into a human-readable thing like "Sync is up-to-date". So we store that in `self._last_status` and update the menu.

What menu? Well, a `QSystemTrayIcon`'s context menu! What you have read are the main pieces you need to create something useful: a Ubuntu One tray app you can use in KDE, XFCE or openbox. Or, if you are on unity and install `sni-qt`, a Ubuntu One app indicator!

Figure 7: My Ubuntu One indicator in action.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "u1-toys project in launchpad".
```

You can find the source code for the whole example app at my [u1-toys project in launchpad](https://launchpad.net/u1-toys/)..
u1-toys project in launchpad: <https://launchpad.net/u1-toys/>

and

```
system-message
WARNING/2 in everything.txt, line 1096
Block quote ends without a blank line; unexpected unindent. backrefs:
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "the full source code".
```

here is [the full source code](http://bazaar.launchpad.net/~ralsina/u1-toys/trunk/view/head:/index.html).. the full source code: <http://bazaar.launchpad.net/~ralsina/u1-toys/trunk/view/head:/index.html>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "icon resources".
```

(missing some [icon resources](http://bazaar.launchpad.net/~ralsina/u1-toys/trunk/view/head:/index.html).. icon resources: <http://bazaar.launchpad.net/~ralsina/u1-toys/trunk/view/head:/index.html>, just get the repo)

Coming soon(ish), more example apps, and cool things to do with our APIs!

Up to the previous post, without counting the long-format posts, and the posts translated to spanish, this blog contained 1463149 (prime!) characters. It took me 4448 days to do that, which means I wrote 328 characters a day.

This is what I talk about, which means I am boring:

Figure 8: BTW: this post is exactly 328 characters long now.

I want to do a few things with this blog(or rather, with my "web presence"), and the time to do them has to come from somewhere, so it's coming from the time used to write the blog.

Will return to a more frequent schedule in a week.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "mele a1000".
```

I have been using the [Mele A1000](#).. mele a1000: <http://liliputing.com/2012/03/mele-a1000-is-a-70-hackable-linux> for a few months now. I think I have

system-message

WARNING/2 in `everything.txt`, line 1150

Block quote ends without a blank line; unexpected unindent. backrefs:

settled into a usage pattern, so let's do a quick roundup of what it's good for and what it's not very good at.

First a quick reminder of what it *is*:

- Small computer running Android (4.0.4 right now)
- HDMI/VGA/Composite video output, ideal for using with a TV
- SATA connector, full size SD card, 3 USB ports: basically unlimited storage capacity.
- Remote control that works with most apps
- Wifi and Ethernet
- It will cost you ~ \$90 to get one.
- No noise, low power usage.

I have it in my bedroom, next to my old CRT TV. I could put it in the living room with the newer LED one, but I have a Mac Mini I got for work there anyway.

Having it plugged to an old TV means the video quality sucks. However, it works great for watching videos, and when connected to a modern display it works perfectly.

So, what do I *do* with it?

I watch movies and series on Netflix.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "this one."

Version 1.8.1 of the Netflix app works fine (later versions do not!) and lets you see movies and series legally for a flat low monthly cost. The remote provided doesn't really work all that well, because you can't do "drags" and therefore can't scroll the movie lists left and right. So, to take full advantage of this you need to get a keyboard/mouse set. I use [this one](#)... this one.: <http://dx.com/p/lenovo-2-4ghz-portable-handheld-wireless-keyboard-with>

However, if you know what you want to see, it's just a matter of searching for it.

I watch videos in general.

[MX Player](#) works very well, and gives smooth, well synced video. The Hardware video decoder works just fine, but **you have to switch to software audio decoding**.

Having a pluggable SATA drive is very useful here. Just get any 2.5" SATA drive, plug it in, and start using it.

File Server

I used [Samba Filesharing](#) and it "just works". You set it up, configure user and password, and your drives are now shared locally and accessible to all your other devices.

I am using it to store backups from my "real" computers, created using [rdiff-backup](#) but any backup tool that can store in a remote folder should work.

Radio & Music Player

There are plenty of android applications that handle music streaming and playing your local music collection. Just use the one you like.

Weather Gadget

I like my TV telling me the weather. Uses a random widget from the play store.

VNC Server

In order to use it more comfortably, or be able to do stuff in it when I am on the notebook, I am using [Droid VNC Server](#) and that even means I can use my own phone as a "remote" using the [Jump VNC client](#)

Yes, that means I am in my bed using my android phone as a remote to my android TV's screen. If the performance were better, it would be as cool as it sounds.

I am using Jump because it's the only Android VNC client I found that supports dragging things (and it's not perfect by a long shot).

Possible future uses

- Skype box (need to hook a webcam. Microphone?)
- Game console?

I have Need For Speed: Shift and I would love to play it on a large screen, but how would I control it? I need to investigate controllers, maybe joysticks or gamepads.

I bought this with the idea of using Ubuntu on it, but the lack of accelerated drivers and Netflix support has made that a non-starter. So, I am doing lots of things where Android really has lower quality software than Ubuntu, but those are forcing me to stay there.

I am quite happy with it, and as a "dumb-tv-smartener" it's quite awesome for its price.

A short tutorial explaining how to create reusable widgets using PyQt.

I decided to bring my office PC into the XXI century by adding a few USB ports to it.

I turned it off. I plugged in the board. I started screwing it... and dropped the screw inside it.

That has happened a billion times before, so I'm not too worried... until suddenly the computer turns itself on, and a awful smell comes from the ISA end of the mobo, along with acrid, black smoke.

I pull the plug, and it doesn't work. It seems to boot, but it makes a constant beeping sound.

Keyboard error.

I put another keyboard.

It all works, except the power LED is now always on as soon as I plug it. But the computer isn't. When I press the power switch, the power LED gets brighter, then it starts.

I wonder what exactly I burned :-)

The Leon County Sheriff's Office has arrested two men in connection with an August robbery of the Petro gas station, 3626 Mahan Drive. The gunmen stole cash out of the register and cigarettes before leaving the store, headed in an unknown direction, said Lt. Linda Butler, spokeswoman for the Sheriff's Office. The Robbery Task Force

Unit, following a lead, was able to connect two men to the robbery and arrested Marcus Kitchens, 21, and **Roberto Alsina**, 23. Both were charged with armed robbery.

The first time ever I found out about someone with my same name, and he*s in jail :-P

I am on a little vacation. I will be back next sunday.

Also, I*ve been quite busy, and the couple of hours a longish pst takes simply was not there.

And nothing interesting happened, so no short stories either ;-)

So, I'm back after a 9-day trip to the beach... relaxed, strong, happy... and TV-less since it seems to have burned something inside when I turned it on yesterday.

Oh, well, who needs it anyway.

And next week I'm seeing Kill Bill Vol. 2!

I am a Tarantino fan, and it has so far scored 100% at the tomatometer (www.rottentomatoes.com), so it seems to be good!

On a professional level, I just nabbed enough support contracts to do what I want to do this year.

A reasonable fixed income is something I haven't had in a couple of years, it has been up-and-down, great months and bad months, with a rather nice average, but...

I needed a fixed income this year because I intend to save a couple thousand dollars for september.

Why? Because I am going to meet a fellow KDE developer, the one that's closest going east from Buenos Aires...

A student asked me how to autologin on a linux terminal. I thought... that can't be hard... I just don't know how it's done.

So, I googled, and there's a patch for mingetty that allows autologin... but it's for an older version (0.9.4) and I can't even find it.

After a while, I check the SRPM from RH9... and it has the patch in the source tarball, but doesn't apply it. No wonder, since it doesn't work.

So, I patched the patch, and here it is: a autologin patch for mingetty 1.0.1... I have so clear in my mind why I don't program in C :-) (although it was rather easy).

The [patch](#)

I even have a patched SPEC file (no biggie there), and even src.rpm's that should work on any Red Hat like thing.

After you install the patched mingetty, you can put a line like this in your /etc/inittab

```
1:2345:respawn:/sbin/mingetty tty1 --autologin=ralsina
```

And a session for user ralsina will open in tty1. You can't even quite logout, since it logs you back in.

I have been using my Palm m100 for appointments and reading etexts for [two months](#) already and it has been really really nice.

However, after reading a slashdot [discussion](#) , I am upgrading, to a almost-new Sony Clie SJ-30

Reasons:

- Color vs Black&White (not grayscale)
- Higher resolution (320x320 vs 160x160)
- Smaller

- Rechargeable battery vs AAA batteries. This means that as long as I remember to plug it every once in a while, I won't lose data.
- I can sell the m100 and pay part of the Clie that way, so it's not all that expensive (about US\$100 in the end)
- More memory (8MB vs 2MB)
- Memory stick slot
- Almost everyone likes the Clies as far as I can see. Almost everyone hates the m100. Why? ;-)

The only possible con is that with a nice set of Duracells, the m100 could go on for two weeks of reading about 4 hours a day, and the better screen in the Clie will eat battery much faster (I've read about 2-day battery lifes from some users).

```
system-message
WARNING/2 in everything.txt, line 1350
Duplicate explicit target name: "discussion".
```

Well, I have been wanting to have a weblog for a while, specially since I got a permanent Internet connection at home.

After reviewing some of the freely available tools, PyCS/PyDS really look like the best combination. They are simple, yet powerful. I could even set a personal server to try stuff, and there's free space at pycs.net. What else could I ask for?

Even further, I think the structure and APIs are nice, so I may try hacking a quickie Qt app for it, since I also want to try the PyQt bindings and the eric3 IDE...

Not only can you now reach this humble weblog through ggl:lateral+opinion (if you are using KDE, of course), but a look at my referrers show that the following google searches have gotten the searcher here:

d-bus havoc (reasonable, I wrote about it) kde vs gnome (reasonable, the words are here)
kfte

Ok, I wrote kfte. But I never mentioned it here. I wonder how that worked?

I am trying to do something which is, I think, pretty cool, in python.

However, I like showing things working, and I am having troubles with the last final step on what I am trying to achieve.

Since I know a few better python programmers read this...

Suppose I have this:

What code should be in fun() so that it figures out if it has been called as C.a or as C.b?

I am thinking something like reading the higher step in a backtrace, but I don't know enough python to figure out how to do that.

Although it opened here (Buenos Aires, Argentina) at the exact same time as almost everywhere else, here's a late review of the movie.

I liked it. I have read the critic slam it, and you know what? I understand them. But I liked it anyway.

Now, why is that, and why don't I slam it as well? The problem critics had with Matrix: Revolutions is one of expectations.

They expected it to be a movie that lifted Matrix: Reloaded into a meaningful and worthy sequel of the original film. That is of course impossible for many reasons.

The main one is that it's 2003, not 1998. In 2003 lame movies like Bad Boys II have special effects that are about as good as The Matrix's were. We are jaded to them.

In fact, we are not only jaded to special effects that were special five years ago, we are jaded to surprise itself, like if showing us something new was an old thing itself.

So, even when there is a sequence that should awe us (like, say, the slow motion punch into Smith's face, look how you can see the fist pushing individual raindrops, the face deforming under the pressure, the detail of the impression of the knuckles... do you know how that was done? I don't!)... well, awe is old news.

So, I went to the theater expecting a movie that would be a fine action movie, with impressive sights, lame acting (not a decent actor in the bunch, except Lawrence Fishburne, and he was apparently directed not to act, but to look serious and important), some sophomoric babble, and some kung fu (I wanted more seraph, though).

In that level, it works. In making us all be five years younger, it didn't.

And to those who said the ending just drops the ball of the story threads, well, guys, this was a christian parable, Neo died for the sins of the programs, he was not the christ figure of the humans, but of the machines.

The robots in the city are the romans, the oracle and other "good bots" are israelites, the architect is Herod, Smith is the roman legions oppressing the israelites, and Neo's messianic "death" (he didn't die, but he is shown in a crucifixion-like pose, and doesn't move) is the second coming announcing the raising of the new temple, by turning the romans against their own legions and ushering peace on earth.

And the humans in Zion are vermin in Jerusalem's sewer system. Just ask the Merovingian (He, I have no idea who he is in the analogy, I just thought this as I was writing, be happy if it holds for another 30 seconds ;-) Ok, he is Mary Magdalene's pimp.

See, that's the kind of thing the Matrix trilogy does, it induces otherwise reasonable people into trying to shoehorn random data into patterns it doesn't fit (at least I tried to make a slightly original one).

That not specially cool quality is shared with other upper-mediocre-crust entertainment, such as "Catcher in the rye". While amusing, its produce is, of course mostly pedestrian.

The trick here, I think, is one used by people like Berlitz, writing about the magic proportions of the pyramids. If you have a large set of numbers, and allow yourself some freedom to tweak, they will match some of the hundred of "important" numbers in nature.

My height * 1000000 ~ the distance to the moon. I must be really important!

The Matrix movies have dozens or hundreds of nuggets of things we recognize from somewhere else, usually only slightly veiled, so we can feel smart about unveiling them, and smarter still about connecting them.

Well, like the size of the stones in the pyramids, you can plug those nuggets into almost any structure you want, as long as you are willing to stand a few holes and forced fitting... and man, did that get old quick after the first movie or not?

So, that's why it didn't work on the philosophical level: we are tired of it, we only had gas for one of those movies.

Add that to the special-effects-are-not-special-anymore syndrome, and you have a recipe for a movie that, while ok, can't work.

And that, friends, is why it didn't.

I looked at my old stories, and found some rather new comments :-)

In my [Squid authentication via POP or IMAP](#) article, Edwin Groothuis asked for it to be updated for Squid 2.5. Which is pretty reasonable, so I did it ;-)

And plovs, at 04:00:27 AM on April 05, 2004 posted the first comment on the very first article I ever posted here, [Simple KDE Trick #1](#) (first and last of the series ;-).

That first feedback only took... 14 months and 4 days, since the article was posted on March 1st 2003.

Isn't it amazing how nothing is ever lost?

As I mentioned, I saw Kill Bill Vol. 2.

While I enjoyed Volume 1's childish notion of grandeur by accumulation, I must confess I was waiting for that emotional moment, like S. Jackson's monologue at the Diner in Pulp Fiction.

Or the whattheheck moment like the death of Robert de Niro's character in Jackie Brown.

Or even the goofy quarter-pounder-with-cheese moment.

But all I got in Volume 1 was Uma Thurman in the slick Bruce Lee outfit, blood, Yakuzas, and the brass of RZA's version of "Battle without honour".

Which is not exactly a small amount of things to get, don't get me wrong, but not quite what I wished for.

And much later I started reading the reviews for Volume 2, and it seemed a more QT flick, and I like those, and I saw it.

And it's good. And it has the quater-pounder-with-cheese moment (Superman), and the whattheheck moment (Elle's), and it has the emotional moment (a lot of them), even if they are darker, and let's face it, basically evil.

So, there's a lot I liked. I was even happy about David Carradine's part (although his lisp drove me nuts (and I don't mean LISP (I mean he says yeth instead of yes))).

BTW: I read on a newspaper an excerpt from an interview he did for Uncut magazine... he's one crazy guy.

The Pai Mei stuff is hilarious. Michael Madsen's Budd is awesome. The very picture of moral decadence wrapped around the nastiest little-brother syndrome ever.

So, I actually quite loved this movie. It's not my favourite QT flick ever (that's Pulp Fiction, then Jackie Brown), but it's a hell of a lot better than almost everything else I saw in the last year or so.

All I want to know is what happened to Sofie Fatale (cheesiest name ever for a french character), and I would be happy as a clam.

A long while ago, I posted a silly article explaining what was first, the egg or the chicken. Noone seemed to notice until now.

Anyone reads estonian, cares to translate?

[Estonian stuff](#)

On a similar note: according to my [zeitgeist](#) this blog is mostly about nerdy stuff, and hinges.

So, here's another way to fool google. Pick a word noone cares about, and write an article about something popular while using a metaphor which has the useless word in it.

Suddenly, you are the number one [google link](#) for that word. This site has been, according to google, the place to go for information about hinges, for months already.

I saw Troy last night. I live close to the Atlas theater, which sports the largest screen in South America (although, sadly, no THX sound), which is well suited for this kind of thing.

And it is a traditional movie. It's like Cleopatra, only the girl can't act and the guy is not a middle-aged alcoholic.

I think Ebert got it right when he said that the actors are trying to make their characters human, and it's counterproductive because the characters in the Iliad are not human, they are archetypes.

It's a damn greek tragedy! The whole idea of greek tragedy is men without will being thrown into their fates by evil gods. I wonder how the greeks managed to stay religious considering all their deities seem to be bastards.

On the other hand, Bana as Hector was ok. He has that *what the hell am I doing here* look a character in tragedy would have if he at the same time was aware of what he is doing, thinks it's nuts but can't stop it, which I think is how anyone would look if the gods forced him to go fight hand to hand against a guy that's supposed to be invulnerable, while a thousand nice archers who are on your side just look.

What distracted me most during the movie is how modern military ideas made the logistics of the troy siege incomprehensible. For example, the landing is like a Delta-Day: D-Day with greeks. Saving Private Achilles. Any half-not-braindead guy would decide that maybe it was a better idea to land a few kilometers away in a non-defended coast, take over a few farms, whatever. But no, they land smack in front of Troy. The boats are rammed into the beach. You know what that does to a flat-bottom greek boat? (I am assuming they were flat-bottomed, or it's just too stupid).

Those things were pretty fragile. And if the tide was low, they would flood and sink as soon as it went up.

There is a reason why the amphibious troop transport ship was invented: you *can't* do that with regular seagoing ships.

Then they don't surround Troy. So, any Trojan that felt like fleeing could use a secondary gate and walk away from it all.

Oh, and just for kicks, the greek camp is at the bottom of a hill.

Not to mention massed armies **running** at each other over hundreds of meters. Marching was invented because if you do that, you are already tired when you get there. And fighting with bronze-age sword, spear and shield is a tiresome job!

Ok, I admit that 50000 greeks **walking** wouldn't be so cinematographically exciting, but hey, I am just whining here.

Oh, and it seems the siege lasted all of a month. I wonder why the trojans didn't just sleep through it. Ok, they are supposed to be morons, with the horse and all that.

But don't get me wrong, I have seen much worse movies, like *return of the lobster man* or *eyes wide shut*. It's just that Homer (or as he is known in Hades, "Spinning Guy") wrote a rather fun epic poem, and it really didn't need all that much tampering.

Ok, so, as I said, I bought myself a Sony Clie PEG-SJ30.

Just for good habit, I decided to go to Sony's website and check if there were any updates on the software for it, or whatever.

Mine is a PEG-SJ30/U model. Sony has never heard of it. If you check for it in Sony's product support DB, it says it's a bad model name.

Everywhere you gonna find pictures of the PEG-SJ30 and will see a silver-plastic case.

Well, mine has a teal-and-black plastic case.

The one you see everywhere has a flip-top nylon screen cover.

Mine has a (sony original) book-shaped leather cover (I don't like it, it's way too thick).

Oh, sure, it exists, you can find it in google, but it seems it's a very unusual model. It works great, though.

I am debugging a perl program written by someone else.

Here's all I am going to say about it:

- a) For someone with a hammer, everything looks like a nail.
- b) For a sufficiently advanced hammer, everything is indistinguishable from a nail.
- c) Perl **is** a sufficiently advanced hammer. It probably can do anything. It is not a pretty sight, though.

Found this [here](#).

Nice stuff about the Mork file format Mozilla is using for some stuff. Nice if you like pain, that is.

Here's [JWZ's original blog](#) specially read the comments.

Here's [the associated bugzilla entry](#).

Just because I'm whimsical, Mindi is a Linux distribution, used by [Mondo](#).

And just because I am a free software guy:

I have no idea what Konqueror does about its history file, but it seems Mozilla is now storing mail in Mork format? That's gonna hurt if you ever want to import it from some other app!

I will be speaking at the [Primer Congreso Nacional de Software Libre](#) on May 27th, at 15:00.

Now, I am not all that thrilled about it, since it's a commercial event and I am not getting any money for it, but it's a chance to show KDE, and I am a pretty good demo guy.

However, look at the sponsor list: IBM, HP, Sun, Novell, and Microsoft.

Yeah, Microsoft.

So, I did whine a little to the organizers, about how this was not exactly a free software event, and that I didn't just speak for free for anyone who asked me.

Well, they told me that the sponsors pay, but it's a free software event, and that I am important for the event, as a member of the free software movement, or whatever.

Ok, since it's like all other free software events, I am now considering wardrobe.

Jeans and T-Shirt is an obvious choice. I use that even on commercial events.

But... what t-shirt? Here's one [good candidate](#).

I specially like the "KILL BILL" model because it shows my combined admiration for Linux And Quentin Tarantino.

Too bad I can't order a t-shirt from North America. I will have to gimp it, hope they don't mind!

It has been **way** too long without posting a longer item, so... I recycled a script I wrote for a customer, and here it is:

A python script that lists (almost) all email addresses in a qmail system.

Also, a slightly tweaked CSS, thanks to Georg!

I have decided I am taking a month off (from work, not the blog, not hobby programming). And if I still feel tired, I am taking another week or two.

I have the money, business is always slow in January (summer vacations here), and not that good in february.

Add to it that maintenance can be done via ssh from any beach on which I happen to be drinking caipirinha, and that's it.

I intend to shut down teaching slowly, take no consulting gigs starting next week, and slowly reach a point, around december 17, where I have nothing to do.

My Sony Clie came with a game called Bejeweled.

I have played 450 games of bejeweled in the last 72 hours.

This game is:

- a) Simple
- b) Addictive
- c) Evil

I feel like a hamster on a wheel. Or like a rat in a maze. I can almost feel new connections in my brain cortex, as my subconscious brain is taught to react without rational thought to the sight of three almost-touching coloured thingies.

My current high score: **49980 points**

Update: Someone has **apparently** scored **10026350** . I feel so small.

Ok, so I touched it a little.

If anyone hates it too much, or it's broken somehow, let me know how to fix it, since I have no idea about CSS.

I think the code snippets look a lot better now, and specially, thanks to Georg's tip, they don't screw the layout anymore.

One weird thing, though (not really related).

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "here".
```

Apparently, long ago, I knew how to put the source of the item on a post (see [here](#), the *Source KDE Dot News* part).

```
system-message
WARNING/2 in everything.txt, line 1645
Duplicate explicit target name: "here".
```

Either the option is gone, or I forgot how to do it. Even if I edit that old post, I can't see where it is.

Sometimes, on saturday afternoons, I am a cheap bastard. When that happens, I go to the Electric.

The Electric is an old cinema, that shows two movies (usually 6-month old releases) for \$4.50.

That's 4.50 as in pesos. Roughly 1.5 Euros. As I said, on saturday afternoons, I am a cheap bastard.

The programmer there is probably a crotchety old geezer who has seen 89000 movies in fifty years, but the pairings he comes up with are a thing o beauty.

Right now, you can see *Secret Window* with *Taking Lifes*. Or *Hidalgo* with a Denzel Washington movie. Or *Startsky & Hutch* with *Master and Commander* (!?!)

I chose the first menu, armed myself with a radioactive-yellow drink called Pomelo Neuss, and prepared to see Mr. Depp get weird.

I had read rather bad reviews of both movies, and was surprised to like both of them.

Secret Window

I must confess I have read almost everything Stephen King published (noone has read **everything** he publishe, including him), and liked, when younger, a lot of it. I hadn't read this one, though.

It's a unusual movie. The opening through-the-mirror shot is damn good, and makes lots of sense in the end.

Depp is a disturbed individual. Here he plays a crazed person.

Taking lives

It features Angelina Jolie's naked breasts. That's an automatic two-star movie at least. It's also not a bad thriller, although it has enough plot holes to drive a truck into them.

The double feature (spoilers)

Amazingly, both movies are about the same subject. In one, a man contains two souls, in another, a soul so dislikes his human vessel, it reformats it into others through murder.

The clinical term for the first case is schizophrenia, for the second there isn't one because it only exists in fiction.

The idea of a person loathing himself so much he wishes to become someone else appears for example in *Les Miserables*. Here we have Jean Valjean as a psycho.

Jean Valjean is shown goodness by a priest, and decides to become good, and he becomes a respected man, then he is shown evil in law, and becomes a father, and a fugitive.

Here, the assassin is shown contempt by his mother, and becomes a hermit crab, changin human shells, becoming someone else for a few years at a time, over and over.

And Javert is played by Jolie, looking damn good in discrete white blouses and black suits (she should keep dressing that way).

In *Secret Window* Depp is harassed by himself, and ends the movie by embracing his other personality, and is, in the end, thoroughly happy with himself.

As you can see, the characters in both movies are victims of self-esteem issues, of different kinds.

I am writing this on a public computer, and the keyboard is sticky. In particular, the **d** often fails. So, some letters may be missing.

Someone with good bandwidth should do it.

Like Planetgnome, only about something interesting ;-)

I tried the [w3m](#) text-mode web browser. It suddenly started displaying images inside my konsole window!

How the heck did that happen??? Not even the [w3m](#) webpage suggests it can do that :-)

And yes, that's just a plain old konsole (it works on xterm, too), and I am running w3m-0.3.2.2-5

Ok, a rant.

A word you see a lot on the free software (whatever) circles, is flexibility.

In particular, people always like to say how a certain program, or operating system, is flexible.

For example, some people say gentoo (or GNOME, or KDE, or fvwm) is flexible.

Poor misguided souls. They should say they are configurable.

Flexible doesn't mean configurable!

A flexible program will handle many different conditions without user intervention (or with minimal intervention). The metaphor is that flexible objects are bent with minimal force.

Kudzu (or Knoppix's hwconfig) is flexible.

Recompiling your kernel to statically link a new driver is **inflexible**.

If you think that just because you can manually reconfigure your system into a bazillion different configurations, your system is flexible, then you surely agree that **marble** is flexible.

After all, you **can** turn marble into many different statues, all different!

No, **silly putty** is flexible, because you can shape it into different forms **easily** and **quickly** and with **little effort**.

If in order to change the ink settings on your printer you need to create a new printing queue, or print to a file and perform a magic incantation with a .ps file, your printing system is **not** flexible.

In fact, the only remotely flexible printing system on Linux is CUPS. The rest are rigid, inflexible and **bad** at their work (at least the ones I know).

If in order to access a device someone tries to use you have to reconfigure your system, or download more stuff, or hunt for a driver, or rebuild your kernel, or recompile applications, your system is **broken**.

If you think that your system is flexible just because after applying force to your system it performs, you are **wrong**. Your system is rigid, and you are the flexible buffer that is adapting, it's like a bakelite rod wrapped in rubberfoam. And you are the foam.

You know, when you take an inflexible object and you try to make it adapt to something, it **breaks**. That's why inflexible software is **broken** so often.

Since almost everyone has this silly idea about what flexible means, I will probably start using *malleable* or *plastic* instead.

Thank you very much, try the veal, I'll be here until thursday.

This site is no longer an important resource about hinges, says google. :-)

If that makes no sense for you, read [this](#) and maybe [that](#)

As mentioned [before](#) I am speaking about KDE today in a congress.

I always get very jittery when I have to speak to an audience. Which, to someone that teaches 3 or 6 3-hour classes a week, is pretty stressing.

On one hand, I am pretty sure I can speak about the life of crabs for two hours and have the people thank me on the way out. I **am** pretty good at this.

On the other hand, I have a tendency to underprepare when it's a once-off affair (like today).

I don't like using slideshows, so I just hook my comp. to the screen and start speaking.

I don't have notes.

I don't have a set of points planned.

So, every time I speak, it is a different thing. Which is good. But it makes me very nervous.

What will happen if today I fail to figure out what to say? What happens if I come 20 minutes short? Well, nothing happens, I just stretch the Q&A section a bit and let them go get free food earlier.

But it still makes me nervous.

Here is a new realtime PyQt tutorial. For those who have not seen the first one, here's the main idea:

- I decide I want to write something
- I write it (somewhat) quickly.
- I write a description of what I am doing, as I am doing it, and slap some timestamps.

In this particular instance, on a discussion at the [dot](#) I had to open my mouth about how writing a spatial file manager is easy.

Well, here's a piece of one. In particular, it's a sort of simple spatial file **browser** since it doesn't manage files at all, but it's a start ;-)

Last week, I gave a conference [\[1\]](#) about KDE in the first free software congress of Argentina.

I can't even remember how many "first(whatever)linux" of "first(whatever)free software(mumble)" events I have attended. one of these days, I expect to attend a second, and in a decade or so a **third**, but it seems organizing one of these things, even when they work nicely, is tiring work.

This one was organized by [Usuaría](#), a non-profit for computing diffusion [\[2\]](#), and they had some interesting sponsors, including Red Hat, Sun, and Microsoft.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "that".
```

Yes, [that](#) Microsoft.

```
system-message
WARNING/2 in everything.txt, line 1812
Duplicate explicit target name: "that".
```

Sadly, I couldn't assist the conference by the MS executive, because I missed about half of the congress for work.

My KDE stuff was shown at a smaller room, about 35/40 people. Since there was very little time (45 minutes) and I wanted to keep some for Q&A, I mostly showed simple stuff, like DCOP, some of the new apps, like Quanta.

I spoke a lot about rather the philosophical thrust of KDE development, how KDE tends to search for a technological solution to the UI problems, on the grounds that later, when everyone is using the API, if the UI changes and the API doesn't everyone wins.

Nothing special, really, and not one of my best ones, so my earlier nerviosism was warranted ;-)

I attended some other conferences, I remember one about comparing MTAs (he called Qmail difficult, so I didn't like it much ;-), one about Free Software economics by a guy from Maastricht [\[3\]](#) which was quite good.

Another one was by a Novell executive, who spoke about J2EE and .NET from a free software perspective.

Or rather, spoke about J2EE for a while, then mentioned Mono because he was running out of time ;-)

I met my third KDE developer! [\[4\]](#) Pupeno was there. Pupeno: you look like a younger, redheaded RMS. And your pants made me dizzy.

I couldn't tell you that personally. I like them :-)

I could tell this was a Linux even because hlf the people there had longer hair and/or longer beards than I do, when in regular events it's unlikely 10% do.

Met a few of the old fellows from my LUG in Santa Fe, one of them seems to enjoy suits now ;-)

But I bet since a few paragraphs above everyone is still having the word Microsoft bouncing in his head.

Yes, they were a sponsor. Further: they were, by far, the largest one.

I got a Microsoft pen, a copy of Unix Services for Unix, a brochure, and a canvas bag with Microsoft's logo embroidered.

Said bag is now the bed of my new kitten, Nini, which I adopted monday (but that's another story).

UPDATE: Someone who was there reminded me that I also got a box containing a fairly nice tukey sandwich, a brownie, and a small bottle of coca cola, so, **thank you, Microsoft!**

And no, I didn't have to sign anything to get the sandwich, not a NDA, not a license, and no, it wasn't wrapped in a bag saying "if you open this bag you agree..."

I have been thinking a little about being relatively old in IT/Internet/dog-year terms, and some things have changed. A lot.

When I was a kid (say, 8 years old?) there were only two TV channels where I lived. And I lived in a rather largish city (350K people). In fact, it was more like 1.5 channels, since one only worked from 6PM to midnight.

Now, I could have 300 channels, if I had any time free to watch TV.

Of course, it was black and white TV, color TV was still 2 years in the future. Hell, our TV still had vacuum tubes. Now I can watch DVDs with Dolby 5.1 sound. I only saw my first `_movie_` with Dolby sound when I was 14.

At that time (although I had not heard of those things yet), the [university](#) where I would later study and work was buying a computer. That modern marvel had 64KB of main memory, and supported up to three simultaneous users on teletypes (yes, teletypes), but at the time they only had punchcards. My phone has more computing power than that.

Eventually, I used that very same computer to learn Fortran III+ using a line editor and a line printer, although it only had one teletype, since it had been upgraded to glass ttys (wyse60s, IIRC). I still have some glass ttys at home (real wyse60s, nice toys).

Phone lines were so hard to get that a house with phone costed roughly 50% extra. To make a phone call to a town 60KM away, the procedure included calling an operator, and waiting two or three hours for the operator to call back.

Now, people get mad at me because I am not in MSN messenger all the time, and I have lost 5 phones in two years, but it's only a minor annoyance. In those days, the idea of losing a phone was about as bizarre as losing a doorstep.

Bills (power, phone, water, etc.) had to be paid on specific banks, on specific dates, and usually there were lines of a block or two on the last available date. Now I have them automatically paid from my bank account, and I get informed via email to authorize it.

Since I was 5 years old, I took a bus (not a school bus, a plain city bus) to school every morning and noone found that too weird. Now I am the only person I know who takes random cabs, because it's safer to call one by phone.

Every country in every direction, including my own, was under a repressive and criminal military dictatorship, and half the world was communist.

¹Is there a less pretentious word in english for this? In spanish I say I give a "charla" which means a conversation, a chat. Of course in english that's not right.

²No, I don't know what they do, although I gave classes in their classrooms for 6 months)

³Where D'Artagnan died (if he had existed, of course)

⁴Yes, I have only ever met three guys involved in KDE development. That makes one every 1.5 years.

But... the last time a man had gone to the moon was with Apollo, the coolest-looking and fastest plane in the world was the SR-71, Ema Peel looked great in The Avengers, summer was hot, girls on bathing suits made the heat bearable, girls make anything bearable, anyway, so some things never change.

I had just finished reading The Count of Montecristo, and I thought it was a great book, and it still is, although I appreciate all the sex and drug abuse of the book more now!

I was a chubby kid, I am a chubby adult. I was a skinny teenager, though, but that didn't last ;-)

So, just because I can now buy a real laser beam for the cost of a hamburger (which was a novelty food when I was a kid), while it was before the stuff Bond villains with lots of money played with, are things really different... well, yeah! ;-)

Good news: It seems I have been accepted to teach a PyQt tutorial at [akademy](#).

Bad news: a kitten I had adopted last monday died on saturday night. She was a sick kitten, and had a respiratory infection. Antibiotics helped for a few days, she looked very happy on friday, but on saturday she woke up weaker, and nothing I or the veterinarian did helped.

She died on my lap that night.

I am a grown up, and I had only had her for 5 days, so this shouldn't be important at all, but it is :-)

It works, the code is only 50% evil, and supports several blogging interfaces via XMLRPC.

I haven't done anything with it for a while, because I use PyDS's web interface (handier to use remotely), but with some loving, it could become a semi-real app pretty soon ;-)

Boudewijn Rempt's blog is at the link, it has a RSS feed, and he's hacking Krita.

Where can one mail suggestions for planetkde blogs?

Yup. A friend of mine just gave me a new one a couple of days ago.

His name is Gandalf (actually, I call it Gan).

What colour is he?

I am a fast reader. Not that I read specially fast, but I read hours and hours, and that way books pile around my bed, under the bathroom's sink, into closets, and on tables, and I have read them.

I hve read the Dune series in a week. The whole works of Heinlein in 20 days. The Lord Of the Rings (and The Hobbit) in a long weekend.

I have been reading Quicksilver for a week now. I am not even 30% done.

Quicksilver is almost 1000 pages long. And they are not the easiest pages, either. I feel like I'm drowning in it, I am being slapped by it, my mind is being raped by a book.

I like it.

My university experience was studying maths, so I am more likely to enjoy this book than most, though.

When you are surrounded by science, if you have even a little of historic sense, you have to wonder: how can it be that I, a reasonably intelligent 20th century guy, have been getting an education for 20 years, and still am quite ignorant, even though I am concentrating on a single field?

Well, after that, if you ever try to do any original scientific work, you have to imagine that those who created modern science were much better than you. Their lives were shorter, uglier, and they **didn't** have the books, yet they figured out a huge amount of stuff.

Usually, there's a mitigating factor in that the science they made is simpler than the science to be made now. That doesn't work for one guy. Maybe two.

That one guy is, of course, Newton. And the other is Leibnitz.

If someone invented Leibnitz for a novel, it would be silly. Inventing Newton would be preposterous. So, how come almost noone bothered making them into fictional characters before?

In taking that step, Stephenson shows genius.

I would write all this book brings to mind, but this blog is too small to contain it.

I saw these three movies in the last 48 hours or so.

Duplex: it's dumb, dark, and the end cheats.

Something's gotta give: If I were Diane Keaton, I would have taken Keanu. If I were Jack Nicholson, I wouldn't have dumped the daughter. However, the concept is cool, and Diane looks damn good.

Now, the sight of Jack Nicholson's butt was completely unnecessary. Now I am so self-conscious I am going to buy a robe. I never imagined anyone could look so bad, and I need to make a habit of hiding my body before it all falls down.

Lost in translation: I want to marry Scarlet Johansson's character. I doubt Mrs. Johansson herself would be half as much fun, although she is obviously just as pretty.

On the other hand, I want to say, make it Suntory time. Now, to the right, with intensity. More Roger Moore!

If you have not seen it, the above paragraph makes no sense. If you did: I want to be him, too. Only younger. And married to her.

I quite liked the movie, even if my date didn't (she says, and she is right, nothing happens in it). Had the movie had a plot, it would be amazing. What is there is good enough it even survives the plotlessness as good.

I suggest Jerry Bruckheimer hires Sofia Coppola. Although he has the worst imaginable taste in directors (the wrong Scott brother! Michael Bay!), he requires all his movies to have a beginning, a middle, and an ending, and that the ending be different from the beginning.

So, if Jerry wants to become a Weinstein-like fellow, and he keeps those requirements, eases up on the explosions, and forces Sofia to hire a writer (if at all possible, the one who wrote "Action;" with Jay Mohr?), man, that is going to be an actual movie, instead of just 2 hours of film.

Oh, you may say, Roberto is crazy! Sofia Coppola is artsy! Bruckheimer is crass!. There is such a thing as too artsy, and definitely, there is such a thing as too crass, but, surprise! There is such a thing as not artsy enough, and, worse of all, not crass enough.

Nowadays, movies fall either on one side or the other, except for *some* films, very few, vanishingly few. I propose exogamy, because the inbreeding of the artsy and the crass tribes is killing them.

Coming soon (not really): why Kiarostami should direct a Tarantino script.

For no particular purpose, except that it made me laugh loudly on a train:

'[he] tried to mask [the smell] with all the spices and aromatic gums of the East. The place was all a-dangle with small charms, cleverly made in the shape of trees, impregnated with rare perfumes.'

One of the themes of the book is that **everything** started in the 17th century, you know :-)

I have been installing qmail servers for a while.

I started by using Buruce Guenter's [untroubled](#) qmail+patches RPM set, which was good, but he seems to have abandoned them.

In the last year or so, I have kept working on them, and now they are somewhat more heavily patched.

I have added:

- quota (bounce over-quota immediately)
- realrcptto (bounce immediately unexistent addresses)
- tarpitting (add delays over n recipients)
- errno patch (to make it build on RH 9 or later)
- smtp auth+tls (well, smtp auth and TLS ;-)

And miscellaneous changes to the default config so that, for example smtpsuth simply works by default.

I think this is probably the easiest way to install a traditional multi-UID qmail on Red Hat-like systems.

It has been a little (not too much) work, because when you pile patch over patch, usually the third one doesn't apply, and this patchset has 22 patches in it.

But... the qmail source is pretty clean for C!

I think I am going to put it on sf.net or somesuch.

Clee: it's really quite simple, IMVHO.

If the goal of planetkde is to syndicate blogs about KDE, then you should only aggregate feeds that are about KDE. The goddess knows mine isn't, except occasionally (you should use this one [instead](#) (it is boring and empty, though)).

However, if the goal of planetkde is to syndicate the blogs of kde-related people, then you just aggregate them (with the agreement of the blogger) and that's that.

If someone posts something strange or sad or whatever, the others can post replies, encouragements, whatever, and life goes on.

Since it's your site, you are the one that chooses. I would prefer the second approach, though.

A Linux distro in my USB keychain. I will do a writeup on it in a day or two, probably, since the plan to work less is working (tuesdays and thursdays off already :-)

Imagine the most ridiculously over-the-top cliché movie about sports. Here are some of the ingredients, and how last night's match between Boca and River compares in a 1/10 scale.

- A bitter rivalry

¶ This is a century-old rivalry, between teams from the same city, who happen to be the most successful and popular teams in the country.

Trust me, this is serious stuff. So serious, that for security reasons, it was the first time ever in argentine football history that the visiting team's fans were forbidden from attending the match.

- It must be an important match

¶ It was a do-or-die affair, it being eliminatory home-and-away, but it was *only* the semifinal of the most important club tournament possible for them, the Libertadores Cup (S.Am. equivalent of the Champions League).

It would have been better for this to be the final, but there has never been a same-country final in this tournament (that I know), so this was as good as it gets.

- A clear favourite

5 The first match (at Boca's home) was won by Boca 1-0, but River could have won 2-1 or so, had they been even slightly lucky (Boca's coach is recognized as the luckiest man on earth by some).

River had kicked Boca's butt all over the field a few weeks ago, playing ten times better, but...

Boca has River's number, and has had it for ten years or so. Boca has won close, and has won clear, has won at home and away, has won in local, international and friendly matches, has won deservingly and has won by sheer luck. But they have won almost everything for 5 or 6 years, including three of these cups, and two Intercontinental cups against the Champions League winner, so... noone was going to say River was a clear favourite. At all.

- A hostile environment

9 The largest hostile crowd in Argentine football history. 66000 chanting, blood-thirsty River fans. And in the middle of it, about 20 Boca players, coaches and assistants. Scary stuff.

It wasn't Brazil-Uruguay in 1950, but then again, there has **never** been anything like Brazil-Uruguay in 1950 (and never could be again).

- Ridiculous suspense.

9 Boca came 1-0 ahead. River scored, evening the aggregate.

Boca player ejected. Things look grim for them.

Then River lost two players in the same play (one ejected, one injured, no replacements available).

In the 89th minute, Boca scores. They are now ahead.

In the 94th minute(!) River scores, all even. Penalties.

The first 8 penalties are scored. It comes down to the last penalty for each team. The River player misses! It all is now in the hands of a single player, and a single kick. Which brings us to...

- The unlikely hero

9 The guy is 21. He is not even a regular substitute, but all the regular penalty shooters are for some reason unable to take them, they have been ejected, or replaced, or injured.

So, he takes it, he scores, Boca wins, and 66000 looks as if someone has played a evil trick with their guts.

Man, I love this game.

I was listening to the radio today, and there were a group of "intellectuals" discussing why on a football match, some of the most common insults involve:

1. Doubting the other fans exclusive dedication to heterosexuality
2. Declaring that the other fans come from one of our fine neighbouring countries.

Well, they went around the subject, and brought forward stuff like ritual **ahem** of prisoners of war in ancient eras, and xenophobia, and homophobia, and racism, and insecurity about their own sexual inclinations, and how the anonymity of the mass changes behaviour, and many, many, many other things.

Well, of course, they are stupid morons who shouldn't be allowed to breath near a microphone, unless they are reading stuff written by someone who is actually thinking.

Those insults are used because **they work** in their intended context.

What you call someone else, assuming you are willing to say **anything** doesn't depend on the insulter. It depends on the insulted.

If I think someone will think offense of being called a wooden wind instrument, or a bunch of twigs, and I relly intend to hurt him, well, that insult will work (no, I don't usually do this).

If I think the other guy is a racist and will be deadly offended by suggesting his ancestors sailed fom Norway, well, calling him a Norseman would be effective.

Since the members of a holigan band are pretty much like the members of the other, this means they don't actually believe the other guy is gay, or Icelandic, or whathaveyou, but just that they expect the other guy to get angry, so a nice fight with cypole chains can take place (no, I don't do those things).

In fact, it doesn't even mean that they would get very offended, it's just a bunch of codewords: "they called us tree-fornicators (no, that's not a real insult), let's go bash their faces". It's a ritual.

How can five people babbling for two hours not see those obvious points, I have no idea.

If you want to see if society evolves about race or sex, it's not a matter of noone calling the others by racist or sexist words, it's about those words not hurting.

And that, dear reader, is why political correctness doesn't work worth a damn.

I have been in orkut for a while, since Taj invited me. I had pretty much forgotten about it, but got another mail from them because someone connected to me...

I have never been a forum guy. What's the use of it? (and yes, Uwe, I know that's what you asked ;-) Maybe I am missing something here.

Or is it that just because I actually know people in person I can't see the fun of it? ;-)

Should start posting stuff again now. Wait, this *is* a post! Talk about a self-fulfilling prophecy!

Lots of things happened in this very long break, so I won't bother even mentioning any of them, except one, which I promised to.

I am in love. And I will, as promised (to her), post a little love letter in my blog, and invite comments.

Since the long postless period probably means nobody reads my page anymore except when finding it via google, I invite anyone reading it in planetkde or wherever, to go to my page using the handy link that should be somewhere around this text, and post a comment if you want.

Be nice, please, because deleting comments from pycs is somewhat of a chore.

And here it goes:

Rosario, I love you. Sometimes you think I don't, but that's just because my skills of facial and verbal expression are awful. I love you when you act silly, or pout, or say wild stuff.

I don't love you **because** you pout or act silly, but pouting and silliness is simply the accent, loveliness is the language that you speak, and I can't do anything but listen to it.

These last two months have been mostly happy for me, and when they weren't, they were happier than they would have been without you, and I hope you felt the same thing.

Now you blush, girl.

Comming soon, the usual technical drivel.

Are you using a RPM or DEB based system? Are you using a program and can't find it in packaged form?

Are you tired of wondering what program you don't use anymore installed all that crap in /usr/local?

Then this article is for you. Learn how to create your own RPMs, that you can later uninstall, upgrade, and share with other computers and users, without much of a headache, thanks to checkinstall.

Ok, I want to retire [my notebook](#). It's about to turn 10 years old, which is about 270 in notebook years, so it's time.

I've been researching a bit, and here's my requirements:

- Cheap. About USD 950 would be fine
- Small.
- Light.
- Enough power to run KDE decently.
- Should work with linux. Nothing very fancy, but it being able to suspend would be nice ;-)
- DVD reader, CD writer. DVD writer would be nicer but consider cost.

I've been looking tenderly at a [Averatec 3220](#). It seems to fulfill all the requirements except it seems not to work so well [with Linux](#).

Before anyone say it's not powerful enough: My desktop has a slower CPU (A Duron) and I like it just fine.

The screen is rather smallish, but that's part of being small and light, I consider the 12" screen a plus.

If anyone knows of a better system in this price range (the Averatec can be found for about USD 890 or so), or of a cheaper way to get one, please comment here!

I have been a big fan of apt4rpm ever since I worked at Conectiva (the guys who created the thing).

Managing Red Hat and Fedora servers without it would be a PITA.

Sadly, I see lots of people who don't know about it, or know, but don't use it to its fullest. So, here's a few tips...

Just that.

For Rob: (No, I won't bother registering in a site to be able to post a comment, so this is easier).

Without even considering the political stuff, which I don't give a damn about, here's what I find nice about having Skype in Linux:

- You can use skypeout and call long distance or international very cheaply
- You can replace your company-wide phone system with it (or never bother to install phones), although I prefer to use {gnome,net}meeting or ophone for this sort of thing.
- It **does** work from behind a firewall, which is what hasn't let me use MSN or similar AV systems.
- Since it works on Linux, I can use it.

Of course all this assumes Skype works as advertised. But if you know of an alternative that can do all that, that would be even better!

I agree that for local calls, if you have little need for a phone, it may make no sense. Sometimes. In some places, local calls are free. Here, I pay by the minute.

Skype would save money in some cases, when I need to have a two-hour-long talk with a client.

In short: would I prefer to have a free, open source program that does all the above? Sure! (although I can't imagine how Skypeout could be free).

Would I prefer *not* to have Skype? Hell no!

So, while not ideal, it's good. And if it's good... well, that's good, isn't it? Then why gripe?

Ok, noone will bother, but you can listen to me speaking about KDE in a Linux event last year, in Ogg Format, in spanish.

I just ran into this by googling for my name :-)

It is the first time I like the way my voice sounds in a recording, too.

If you map a domain (x.com) to a single account (xuser) and then send a message to a@x.com, BCC to b@x.com, only one copy is delivered to xuser's mailbox.

Therefore, it's impossible to implement a correct multidrop account in sendmail that you can later pickup using fetchmail or similar software.

No, I haven't won the lottery or anything. And no, it's not because I'm in love (but I **am**). It's just that I look at the 33 years I have spent around and notice that:

I have never had anything really terrible happen to me. Sure, small problems, even large problems, but no stuff like loss of a limb.

I have never been really poor. Mind you, many reading this live in places where being poor means you have to ask for welfare (which I haven't), but here, being really poor means you die sleeping in the street. Or you go to prison and rot. Or you watch your own children starve.

Sure, I've had days when I didn't have enough to buy the newspaper. But I still had food in the fridge (and a fridge), and a roof to put the fridge under.

I have never been unemployed. Sure, I don't have a job that pays every month, but I earn my own money, and haven't had to ask anyone for anything in a very long while, except as a very short term loan.

I like what I do. It's easy (at least for me), leaves me enough days off to be with Rosario, is not physically exhausting. Usually it's done in air-conditioned offices. It lets me rationalize my craving for gadgets as job-related.

I managed to ride some of the worse times in this country's history and come out on my feet.

I am not dead inside.

I don't give a damn about stuff that's undeserving.

I still feel a need to help people when I can, but I can live with not helping those I can't. I have accepted I can't save everyone, so I try to save the ones I can, yet I don't forget the rest.

I haven't fallen into religion, nationalism, politics, greed, or any of the multiple traps that grab you and don't let you be a real human.

My family is well, with the usual allowances for human trouble, and I actually like each one of them.

All in all, I'm a fucking lucky son of a bitch.

I just remembered that in september 2000 I had created a KFTE project on sf.net.

So, I go check it and... yup, still open. Noone has even looked at the page since creation, no files, CVS, forum, mail or any other kind of activity since the creation of the project, but it is still open.

No wonder SourceForge has such an amazing number of projects registered! ;-)

But anyway, I uploaded release 0.7.0 here: <http://sourceforge.net/projects/kfte/>

Rob: you can remove it.

I removed it because I don't use it:

Configure Toolbars-> Location Toolbar (searchbar) and then move the only action to the left.

Apply (Or OK) and it's gone.

Now, for some reason, planet won't get my second post of the day until the next day. Maybe it's a bug on planet, maybe it's on PyCS, maybe it's on PyDS. Beats me.

A tutorial explaining how to setup your own APT-RPM repository, so you can install your custom (or custom-packaged) programs easily.

Now, this is too lame a piece to announce on the usual sites, so I will just post it here, and as a traceback for the other APT article.

It really feels like half-an-article without the pin and policy stuff, so it will surely get updated eventually.

Does such a thing exist? There are dozens, but none seems very good.

By good I mean:

- Has been maintained more recently than 4 years ago.
- Works via PAM (and just plain works)
- Is not awful to install
- Doesn't make you do weird stuff like running a SUID httpd (yes, I actually saw that once)
- secure (audited?)
- readable sources
- runs as a non-privileged user.

Usually this would be a SUID root cgi-bin, which is somewhat scary, and it would seem to me unnecessary.

Since the user will provide the current password, it should be possible for a non-privileged process to first switch to the desired user and then change the password, right?

Maybe someone can tell me. Or do I have to write it? I mean, it's going to be a python CGI if I do, and noone's gonna like it ;-)

You know, some things are usable. Some things aren't. Read on, to see one that isn't and one that (I think) is.

A semi-tongue-in-cheek usability article.

OK, so several have figured out that UseCalc is in large part a spoof on usability research.

Too bad it didn't cause the response I wished for, which was some reasoned usability explanation for why calculators are done that way.

On the other hand, it turns out several people like Usecalc and have proceeded to reimplement it in other languages, and with different interfaces.

Which is quite good, too, even if it surprised me a little. Now I have to think about it, too. I mean, maybe there is real potential for the thing.

From Speno's Pythonic Avocado.

So, I am thinking about calculators more. Whenever it isn't sunny and I don't have an urge to hang around with Rosario, I may put what I think into code, too ;-)

But anyway congrats to Ariya and mpyne since their calculators are already pretty good (haven't tried Zack's yet, I have no ruby).

But I still have some tricks up my sleeve!

I often see and read about cooperation in free software. I read about how multiple competing projects are counterproductive, about how cooperation would make things go faster...

You know, that's all wrong. I am not saying cooperation doesn't have its place, it does! It's just that not *everything* works better with cooperation.

I will just talk about what I know, and what I know is very little, so sorry if the examples are too tightly focused, but I am a small picture guy.

Example 1: KDE and GNOME

For the GNOME fans: If cooperation were more in vogue in 1996, your object of affection wouldn't exist, so that's it for you.

For the KDE fans, the case is a little harder to make, because we assume that we would be better along if GNOME hadn't existed (I had even said that a few times).

Although it is not possible to prove one thing or the other, since history is not an experimental science, allow me to point out that KDE's goals were much smaller, its scope immensely less ambitious. It was meant to be about 20 small programs, not a multimillion-LOC thing.

GNOME was part of the incentive to move into the larger stage. Think of it as an arms race.

Let's face it, some things were developed just because GNOME had them, or even because they didn't have them yet ;-)

I assume GNOME guys can say similar things, it should be even more obvious in their case, since the project was started because of KDE (Miguel once said something like KDE convinced him that it was possible to do[1])

So, cooperation would have been detrimental. It's not that cooperation hurts development, but that competition fuels ambition, and ambition fuels development.

Example 2: Mad scientist syndrome

Once upon a time, Rasterman had started developing what he called 'themes' for Gtk+.

You know, I had read some of his code when he started hacking fvwm, and while I respect his graphic skills (although his taste is horrid to me [2]), his coding skills... well, let's say that I don't hold mine in high esteem, but his code... well, I suppose it was good in some efficiency sense... I hope it was, else it was unforgivable.

Also, I was convinced that Qt was way better than Gtk+ in any conceivable sense[3], so I thought... how is he doing it?

It turns out that he was replacing the drawing code for the widgets with what would later be known as the “pixmap engine”. The drawing code read a config file, loaded pixmaps and pasted them over the place the widget would be.

Mind you: that scheme is hideous for many reasons, and simple badly designed (you can’t change your colours in any reasonable way, for example). But... I had troubles.

You see, we are talking Qt 1.x at the time. Under the non-free-software license. I couldn’t touch Qt, so I couldn’t modify the drawing code.

So, how could I show Qt was better? By doing an evil hack (TM)[4], of course. I intercepted the drawing events in the application class.

I held lists of widgets at creation, I diverted the expose events off the toolkit code, I subclassed the application class, I overrode the compiled code with LD_PRELOAD, I used almost every evil hackery I could find, and you know what? It worked. I had a screenshot to show, too.

In fact, in 36 hours I had a more advanced theming engine than Rasterman’s, by some measures (I had themed way more widgets).

I showed it to some KDE guys, Kalle showed it to some Trolls, they got so nauseated by it, I still think it partly convinced them to add styles in Qt 2.x, so I wouldn’t have them look at it again (It used, after all, only the public APIs).

So, where is the connection between that story and cooperation? Oh, grasshoper, the connection is that there wasn’t any connection.

Had I been a cooperative sort, I would have shown the ideas to someone and been laughed at. Or I would have contacted some of the Trolls (I knew a few of them back then) and would have been convinced that it was a bad idea (it was :-)

The only reason why I coded for 36 straight hours was because I really really really was pissed off that Rasterman could hack Gtk+ and I couldn’t do the same thing to Qt.

So, cooperation drives to the average. Cooperation begets bland. And while bland is good for you, if everything was bland, free software would suck immensely. Free software is not a tool for a goal, at least not for some of those writing it, it is a toy, in the good sense, something that is useful for itself, in itself, even if it does nothing. So are jewels, you know.

While a project can not be developed by a single person, a design almost certainly MUST, specially if it’s a new or strange one. And some stuff really has to be done in solitude, or it won’t be done.

[1] I really don’t like the guy (old story don’t ask), but hey, why would he lie about that?

[2] My taste is horrid to me, too.

[3] I never said my opinions were ver rational

[4] And I mean evil. Evil as in mordor-code, evil as in a design so hideous I still remember it :-)

Ariya: add a variable list display.

Also: my girlfriend’s mother is from Turkey ;-)

For completeness’ sake :-)

So, I was not too original ;-)

I’m writing a clone of Bejeweled.

If it’s half as much fun as the original, it’s gonna kill productivity for a while.

Here's the current version. You can actually play it already, it keeps score, and everything.

Sure, it has no sound, and you can't lose, but it's nice ;-)

The tiles are, of course, just a temp job.

Rob, I think the problem here is that you don't know how to use KSpread.

Here's the correct syntax for your examples (untested, but should work):

```
=F7 * ( 1 + MAX( 0.20 ; (F7-F2)/F2/5 ) )
```

```
=MAX(A1;100)
```

In fact, the one that you mention as working, MAX (A1:A2) works by coincidence (Unless you really want the max over a range, instead of the max over two cells).

Can we get that not-grumpy week anyway?

As I blogged [earlier](#) I am writing a game (and yes, it's pretty much playable already).

One thing I didn't mention is that I never wrote a game before. Yeah, I know everyone does it as one of his first projects, but I never did.

So, there are some things I really have no clue about [\[1\]](#), like sound and moving graphics around.

For the graphics stuff, QCanvas is just fine and dandy, but to make things bloop and warble and squeak when the time is right, I found Qt's sound support somewhat depressing.

Come on, NAS? Who uses that? And what about music? I had no idea.

So, I started trying to follow one of my leading principles of development: find a way to make it Someone Else's Problem (TM).

The usual way to do that is finding a library that handles the problem, write minimal glue, stick it to the side of the program, tell the program that's his new arm, and forget about it quickly.

Here's what I found.

Mi Dios!

I thought I should start by adding one of those annoying little tunes every game has. It's just a game tune, I don't want to have to include a 3MB OGG file for it, so I wanted an instrument-based format.

I remembered MIDI tunes. You may know them as ringtones nowadays, but they used to be just cheesy tunes generated by your SBPro's FM generator, not your phone.

In fact, I remember having a little proggy called playmidi, that would do that in Linux.

Well, it seems that in the past few years, either sound cards have forgotten how to play them, they fell out of fashion, or something, because the only things I found that could play MIDI are monstrosities that require a 9MB digital instrument set. And how was I to include that along with my 25KB game???

So, what's next? I had a C64, so...

MOD me up!

MOD files are like MIDI files, only the MOD includes it's own instrument set, called samples, and instructions on how to repeat and alter those samples to make a

tune.

Good news: there are nice-sounding, funny MOD files that are about 30KB in size.

Better news: There is a popular library to play them! It's called Mikmod, and your distro has it (and it's a dependency for KDE's multimedia packages too).

Even better news: It has support for playing simple sounds (samples in mod lingo) by calling a couple of functions.

Awesome news: It includes a software mixer so you can just tell it to play this, then play that, then that, and a tune in the background, and everything sounds at the same time.

So, we have a winner. This baby can handle everything I need for the game!

But... is that a snake in your pocket?

I can't find a Python binding for it. I am sure as soon as I post this article someone is going to come up and tell me, here they are, moron! But I just can't find any.

So, I decided to do something I wanted to do already and learn to use Pyrex. Pyrex is a tool to write python extensions, with almost-free access to C libraries, in an almost-python language (only minor syntax differences).

That way, I could write a Python module to use Mikmod.

You know what? It was almost scarily simple[2]_. I didn't wrap all of Mikmod[3]_ because I don't need it, but now I can do stuff for games and apps almost trivially.

Even more: Pyrex has awesome distutils support, so building the extensions, usually a pain in the rear, is trivial (mostly you just copy and delete stuff, with some search and replace).

One thing I found I did nicely is this: Mikmod requires you to call Mikmod_Update every once in a while so it fills the soundcard's buffer with stuff to play. If you don't, it skips.

So, I just started a thread that loops and takes care of it. You don't even have to know about it to use the extension. Oh, sure, if your Mikmod is not threadsafe, it breaks. Well, get a decent Mikmod package, then.

How does it look?

Here's a whole noisy proggie

1

```
system-message
WARNING/2 in everything.txt, line 2559
Duplicate explicit target name: "1".
```

As if that would surprise anyone!

2

```
system-message
WARNING/2 in everything.txt, line 2561
Duplicate explicit target name: "2".
```

On the other hand... waiting for stuff to **compile**... how quaint.

3

```
system-message
WARNING/2 in everything.txt, line 2563
Duplicate explicit target name: "3".
```

Actually, I am wrapping almost all of Mikmod, I am just not exposing it to Python because I don't need it.

Python code really doesn't look good on planetkde, so follow the link to my page to see it right if you are there ;-)

Danny Allen has been kind enough to send me some pretty pieces for my Ongoing Game Development Effort That Has No Name Yet (But Should Not Be Called Anything Jewelry-Related)[1]

So, thanks Danny!

I think I am going to switch to SVG-based image loading, too.

Tavia seems to be another KDE web browser.

I say nice.

Semi-related:

Because of a question on the PyKDE list, asking for a more complete HTML widget using Qt (not kdelibs) I am looking at kdenox, which is supposed to be a Qt-only (QT/E) konqy. Perhaps ripping off the khtml piece of it won't be too hard, and then it can be sip-wrapped.

Probably nothing will happen with it, but I *am* looking ;-)

ISC makes dhcpcd and BIND.

Dhcpcd can update a BIND zone using a variety of methods. The one I saw examples of is called the "interim" method. It requires that you create what's called a TISC key, used by BIND to authenticate the zone update requests.

All fine and dandy. BIND even has a tool to generate such keys.

Too bad about 1 in every 8 times, the key will include a / and will be a syntax error if you include it in a BIND named.conf file.

A minor bug? Sure. But come on, noone has noticed it????

Just installed it a couple of days ago, so I wrote a little about it.

Since porting HTML display engines from one place to another is all the rage these days, I did a little hacking on one of the oldest forms of the art:

KDE's kdenox module contains konq-embed, which is, among other things, a port of KHTML to almost-pure-Qt (I think).

So, I am hacking to make that really-pure-Qt.

So, it's basically porting KHTML from KDE to Qt.

Why? Because I want a decent HTML widget for PyQt that doesn't require kdelibs. So people using PyQt/Win32 can use it, for instance.

I don't know if it will work (hacking C++ code is almost painful to me nowadays).

Ok, I have done a little about that.

Turns out I was patching it for HEAD, not 3_3_1, so I fixed that, and now the available patches work for both versions.

Of course that doesn't mean either one actually builds, only that the stuff that needed to be patched for 3.2 is now patched for newer versions as well.

The biggest problem with this piece of code is that in order to do it nicely, I have to test every change I make against sources prepared from 3 different KDE releases (or rather, about 5 or 6, but I will not do that), so that when I touch one of the dropin/classes to provide a 3.3 interface, I don't break the 3.2 kdesrc/ build.

And I am oing it in a pretty slow box, so each build takes a couple of hours.

1

system-message

WARNING/2 in `everything.txt`, line 2577

Duplicate explicit target name: "1".

Or, as I call it, OGETHNNY(BSNBCAJ-R)

Ergo, I am doing it wrong. I am going to hack against 3.3.1 until it works, then 3.2 until it works, then 3.3.89.

Then all around again, until all three work without trouble. It's not going to be too easy.

To make matters worse, I realized that I don't **need** it to work with 3.3 for what I wanted, since a 3.2-based KHTML is already good enough.

Oh, well, at least I am hacking in public CVS again. Hopefully I won't break anything too badly.

I have roughly 95% of it building with kdelibs 3.3.1, with the patches committed up to now.

I have some weird problem with khtml_part where it doesn't see a function that's defined right there, but I am starting to get dizzy.

It's not very difficult coding (mostly cut&paste) but it's a *large* piece of it and this is a slooow computer :-)

This last week has been one of the most exhaustings I can remember. And that includes the one I didn't sleep, while drinking heavily with all my schoolmates in Bariloche (local tradition, never mind).

Let's see...

Last friday, after finishing work at 21:00, I boarded a bus at 23:20 with Rosario, headed for my birthplace, Santa Fe. That's 450km away, and it took until 5:30 am.

There I was greeted by my brother, who I had not seen in almost three years, and we went to his place to waste some time because I had to go buy a gift for my niece, who was having her 5th birthday party.

So, I slept about 2 hours, and then we went, bought some Barbie make up (my niece is going to be a model, trust me), and we sailed off, in his Peugeot Partner, towards Moises Ville (his town), about 200Km to the left.

Bad news: the Partner has only two seats. There's three of us. So, I went bouncing in the back, in a mattress, Rosario and Santiago (my brother) at the front.

We did that for about 80Km. Then we picked up Romina (ex-nanny of Muriel, my niece) and Martin (Romina's chatterbox, 1.5 year old).

So they went to the front, and Rosario, I, a few pounds of meat, a PC CRT monitor, a few suitcases, the gifts and some flowers went on the back.

Then we arrived at the next town, some 50Km further. I was about to get sick, so, when we picked up the Barney & friends foam suits, I switched places with Romina & Martin (hey, I'm about twice the size of both of them... together).

That's how we arrived at Moises, around 3PM.

Rosario and Santiago went for a nap, I helped Sandra (Santiago's wife) pick up some stuff, like, beverages or whatever, and at 16:30, the birthday party started.

Nice party. Muriel dressed like a princess (she's too cute, really. Princesses are usually uglier ;-)) 40 of her best friends (in a town pop. 1800), two inflatable castles to bounce in, cake, and all the sandwiches you can eat.

Good thing Rosario used to work teaching kindergarten, too, so she lent a hand...

Highlights of the day: Muriel waltzing with her uncle Iair dressed as the Beast (Disney version... ok, he looked more like the creatures in "The Village" to me, but to each his own).

Listened at random: "you know, he could have just used a zipper in his belly as a beast costume", which is an absolute defamation of the boy ;-)

Rosario painted every little girl as if they were about to be on TV...

Now, I have been to some 5-year-old gatherings, and this bunch has to be the sweetest-natured children I ever saw, except maybe for the kids at Urdinarrain, who even say hi to strangers on the street.

At 19:30 everyone went away, and we rested a little, then headed for one of Sandra & Santiago's business, a cybercafe/ping-pong den for a nice asado (that's why there was meat in the truck :-).

Awesome meal, laughed until Santiago seemed about to burst something, disgusted the female company with anecdotes, Muriel sleeping in a corner, while kids were playing counterstrike...

The following day more of the same, knishes, empanadas, long nap, thinking of how the hell we could get back to Buenos Aires before monday 18:00, when I had to meet a customer (yeah, that's right).

Sandra took us in a scary pickup ride in the darkness over a dirt road to Palacios, the town on the pavement road, where we waited in the darkness for a bus.

That bus was headed to Rosario (a city, not my girlfriend), and it was full, so Rosario got a seat, I got to stand up (and smooch) until Sunchales (some 40Km) where some people left, and we got seats, and then nothing much happened until we arrived, around 1am.

Then... no buses until the morning. We had almost no money now, but what the hell. We ate a couple of sandwiches, took a room in a cheap nearby hotel, and took a bus (finally a nice one) at 9:15 am.

Arrived at B.A. at 13:30, she went home, I went to work, went home at 21:30, collapsed at 21:32.

Discovered my house was infested with fleas at 21:33 (how? My cat never leaves! It's a 7th floor!)

lost consciousness at 21:38.

And it was still only monday.

Then lots of other stuff, like trying to buy a suit with Rosario and Maria Feray (her sister), and waling about 40 blocks looking for a piece of cloth... lots of work... even some hacking...

And you know... I have a ticket for San Juan (oh, roughly a 14hs bus ride ;-) for tomorrow. Once I get there at 10am on saturday, I will have to dress up for a party (a wedding, not mine), and then we will come back in a car. On sunday.

That means I am going to spend 12 hours in a car with Rosario, her sister and their father. Have mercy on my soul if noone sees me again!

I forgot to mention: I saw Nahuel, a black cocker spaniel that lived with me when he was a puppy... many years ago. He's blind, but he still recognized me. Emotional moment :-)

[Fwbuilder](#) was one of three GTK applications I used (inkscape and gimp are the other two).

And today I noticed that version 2 is Qt-based :-)

I like it a lot. In fact, as I say in my classes, it's the first firewall configuration application for Linux I've found that is actually *easier* than editing a script, since almost every other one I've found is a glorified rule editor.

Bonus cool application: [p3scan](#)

I found about this today, when a customer asked me for a way to viruscan his incoming mail. Which is stored in a bunch of accounts all over the internet :-)

For webmails, a squid redirector does the job, but for pop3? p3scan is completely transparent (only requires a tiny configuration and a NAT rule) and for what I tested it, works just fine.

I am using it along with clamav, but it also supports Kaspersky, F-Prot, and almost anything, really.

Cool gadget, and it's just a 72KB download.

... oh, screw it, why not say what it *doesn't* need? It's better for my blood pressure.

- It doesn't need Be refugees whining about how BeOS was so awesome and Linux desktops suck. You know what, guys? If BeOS was so cool, and Linux sucks so much, how come BeOS is dead? Now put a sock in it and go practice your OS-necrophilia.
- It doesn't need windows users saying "it's not like windows, it sucks". If Linux was just like windows, you would say that there is no reason to switch, so put a sock in it, because you can't have it both ways.
- It doesn't need Mac users saying how there's no point working on it because OSX is already as good a unix desktop as there's gonna be, so why bother?. Put a (designer) sock on it, because in order to use OSX you have to buy an ungodly expensive (except the 12" iBook, that's kinda cheap and cute) computer that provides only adequate performance, and I am a cheap bastard.
- It doesn't need old unix gurus explaining how there's no need to work on KDE or GNOME because Window Maker is lighter, and all that's needed is a way to keep their xterms sorted. Put a sock in it, because xterm sucks, Window Maker is featureless, and you can't do anything useful for a regular user with just xterms and Window Maker, so you are comparing apples to apple orchards.

Most of all, it doesn't need any "What linux needs on the desktop" editorials.

On other news, I will be speaking about Linux on the desktop at the UADE in Buenos Aires, Argentina, this saturday, and about PyQt/PyKDE programming this friday.

More information at the CafeLUG [site](#). If you read it here and mention my blog, I will pay you a beer (offers limited to a maximum stock of three beers).

As I mentioned before, I was at the 3rd CafeLUG event on friday and saturday.

It was a pretty large event, roughly 1300 people, 63 conferences.

I had two of my own, and here's how they went:

Friday: The PyQt/PyKDE one. Audience: about 50 sitting, 15 standing. I had no time whatsoever to prepare this. But I was lucky enough to have someone (thanks Cristian!) lend me a notebook, and so I spent the previous conference (pf/OpenBSD) working on mine :-)

I kept it on a low level, nothing too complicated, since I only had 60 minutes, but the audience seems to have liked it. Or at least they found my attempts at explanation funny.

Anyway, noone was disgusted enough to tell me to my face, and I had a general feeling of it having been good.

I was able to show a quick do-nothing app in 20 lines of code, and show signals and slots, and really **really** push the idea of interpreted languages as better for hobby programming (which I think is the right path). Not too many questions right then, but about a dozen kids attacked me with questions on the hall after leaving.

Saturday: My generic KDE-for-beginners speech. Audience: 100 sitting, 10 standing. I was a bit too heavy on the philosophical, but it went down nicely. Several of the guys from friday's PyQt session were there, so more evidence it didn't suck too badly.

It was fun, people were surprised by Ogg-ripping from Konqueror as usual, some fun, several questions.

All in all, a nice couple of sessions, and I think I did well, which encourages me to do it more often. Who knows :-)

I have been playing a bunch of their Java games that say they are incompatible with unix & macintosh computers.

They work great, sound and everything, except for the highscores (requires IE 4.0 or later). Since I am not even spoofing the user agent, I am willing to guess there is a chance of that working too.

But why is yahoo saying it doesn't work? I have not a guess of a hint of a shadow of a reason.

So far so good. Nice job, Georg :-)

Brand new cats.

Late, but linked: I was interviewed by one of the nice guys at KDE hispano, something like Tink's old interviews. In spanish.

I have been palm-less for a while, because my charger was broken.

So last week I bought a nicer charger, which works from the USB port, instead of Sony's powerbrick-from-hell (which you have to carry around if you ever want to sync).

And since all my data was gone, I looked around for etext reading software again.

Using a CLie means the latest Weasel Reader doesn't work right, and plucker is a pain in the ass. It's cool if you want it to self-update, but I don't. I read static books.

So, I was using TiBR, which is a basic, but nice palmDOC reader. Most of the free ones work like crap on a Clie (I mean you, weaselreader), but this one is nice.

But I got a bunch of zTXT docs (converted when I had a m100 and read using weasel), which TiBR can't read.

Enter [PalmFiction](#) a **nice** and **fancy** reader than handles palmDOC and zTXT. If it handled plucker I would be too happy :-)

But really, a very nice thing. Don't be scared by the page in russian, just get it and install it :-)

Only niggle so far: can't choose a smaller-than-normal font.

Funny article at [GlitchNYC](#) about looking at KDE 2.0 today.

But hey, I wrote a theming engine for Qt 1.x, so I've seen, remember, and even mucked up with worse stuff :-)

I have been playing with this [code](#) and it's been lots of fun.

I've hacked it into a functional spreadsheet in (according to eric3) 508 lines of non-doc code, of which 244 are generated by pyuic.

Here's my code so [far](#) (requires PyQt). Give it a look, I think it's kinda nice.

The only hard part I wrote (at least hard for me) was the cell dependency and recalculation support.

There's a [test file](#) you can use, too.

It is trivial to add functions you can use in the cells, just lookup python docs for eval() and check engine.py.

To use it, unpack it, and from the directory it creates run `python ssheet.py`

I don't plan to make it a real spreadsheet, but it should be fun to hack on :-)

...If you make ESR argue with a anarcho-libertarian-freemarketer-progressive (or whatever?)

A lot of hot gas!

It's a pity George Staikos' reasonable comments are in this piece, but hey, who could have guessed.

Anyway: the writer is the most annoying essayist I've ever read. And he goes and has an email exchange with ESR (the previous holder of the crown). And half-quotes it at length. Sigh. A trainwreck.

We are all now slightly dumber if we read this piece.

Why your programs are like frogs.

I wrote a little spreadsheet thingie a few days ago.[\[1\]](#)

Of course, it's a toy, not the real thing at all, but it was a nice hack, since it is a real, recalculating, extensible, dependency-checking, loop-avoiding spreadsheet engine in about 50 lines of code.

That's because I was using Python, which is a seriously cool language to write that kind of thing in, since all you have to do to evaluate an expression is call `eval()` on it.

Sure, that's nice, but the real core of the spreadsheet engine was that you could also create a dictionary-like object that recalculated on-demand its contents.

That way, when you ask for `sheet['a1']`, custom code goes to see what `a1` has in it (a formula), calculates it if needed, and maybe trigger a few extra recalculations if another cell depends on `a1`.[\[2\]](#)

But as anyone who uses spreadsheets can tell you, weird things exist in spreadsheet land.

For example, if you copy something, then you paste it, it gets modified in the process.

What other app does that???

Here's an example you can check in any spreadsheet:

- In A1, type "1".
- In B1, type "A1+1" (should display 2)
- In A2, type 2
- Copy B1 to B2, and it will display 3

Further, if you look at the formula in B2, it says `A2+1` now.

That's called relative cellnames (I think).

In order to do that trick, you have to parse the formula in B1, and then, when you paste it into B2, take into account the displacement and modify accordingly. Usually, if you want absolute names, you use `$A1` instead, and that would stay unmodified.

Now, that throws a nice monkeywrench into my neat little spreadsheet[\[3\]](#) because now it suddenly looks not like a spreadsheet at all!

So, I started thinking, how the hell could this be done? The whole advantage of a python sheet is using `eval()`, so switching to a parser (like if this were a C++ sheet) would be silly.

I delved into the python standard lib. As every python programmer knows, almost everything is there. If you write python, you read the library reference every day, and memorize chunks of it, because it's one of the things that make python cool. It's just chockfull of useful stuff!

And here I was reading about the compiler module, and the parser module, which can be used to do wondrous stuff with python code. But I couldn't understand jackshit about them. I'm a simple coder.

And just as I was going to say, let's write instead about the connection between free software and the sex life of frogs[\[4\]](#). I found `tokenize`.

`Tokenize` is a module that parses python and turns it into tokens. Here's how `a+2` looks after you tokenize it:

```

1, 0-1, 1:      NAME      ' a '
1, 1-1, 2:      OP        ' + '
1, 2-1, 3:      NUMBER    ' 2 '
2, 0-2, 0:      ENDMARKER  ' '

```

The numbers on the left side are positions in the text stream where the tokens were. It has just enough information that you can tokenize a piece of code, and then reassemble it. There's code to do just that, it's called [regurgitate](#) and it's written by Ka-Ping Yee.

So, the solution is obvious. When copying a formula:

- Tokenize the formula to be copied
- Look for tokens of type NAME
- See if it looks like a cellname, or `_cellname`
- If it's `_cellname`, leave as is. That will be our notation for absolute cells
- If it's `cellname`, displace it nicely
- Regurgitate it

Later, when evaluating a formula, if someone asks for cell `_a1` give him cell `a1`.

And voilà, relative cells.

This works, and it works well (ok, I had to introduce some ugly globals, I need to learn more stuff), and it is guaranteed to tokenize in the same way python does it. It's not even really slow[5]

I touched a bunch of other things, including support for all the functions in python's `math` module so you can use them in cells. Here's the code to do that:

```

for name in dir(math):
    if name[0]<>"_":
        self.tools[name]=eval('math.'+name)

```

Freaky stuff, isn't it?

What's the main issue? Performance. To put it simply, I seriously doubt a sheet written in python can be fast enough for general use. But hey, it's extensible, it's nice, and depending on what you are trying to do, it *may* be good enough.

And here's today's version of [StupidSheet](#) including relative cells. Don't worry, it's

a small download ;-)

Yesterday I was talking with Rosario, and we had a strange argument. She said I was not in touch with my feminine side, and that she didn't like that.

To which I replied I don't have a feminine side.

Of course that caused a reply of "you do, but you are not in touch with it" and whatever. I asked her what my feminine side is supposed to be, and she mentioned that it was the part of me where I could find my feelings and some other things.

To that, I say, nyah nyah!

The idea that just because I have a certain kind of chromosomes, in order to find my feelings I must have a side that's "feminine", I say that's just propaganda.

I am a guy. I am masculine. I cry whenever I watch the end of Forrest Gump, The part of me that's crying is not some feminine side. It's my masculine side. I cry masculine tears, because a movie triggers some response in my very masculine feelings.

If I were to tell a woman that she has to get in touch with her masculine side so she can, say, drive a car better, I would be called a chauvinist pig. And correctly.

The idea, I suppose, is that her feminine side can drive just fine. Well, my masculine side can cry on movies, and fall in love, and miss my mom, and pet my cat just fine, thank you very much.

So, I **don't** have a feminine side Rosario, it's just my other masculine side, and he loves you too.

Thanks to Ed Taekema, now I have a full browsable archive of the posts.

1

system-message

WARNING/2 in `everything.txt`, line 3002
Duplicate explicit target name: "1".

And almost noone noticed ;-)

2

system-message

WARNING/2 in `everything.txt`, line 3004
Duplicate explicit target name: "2".

That triggering is the only part I wrote myself, the rest is from ASPN's cookbook.

3

system-message

WARNING/2 in `everything.txt`, line 3006
Duplicate explicit target name: "3".

I call it StupidSheet.

4

system-message

WARNING/2 in `everything.txt`, line 3008
Duplicate explicit target name: "4".

I did write that [anyway](#)

⁵I don't advice you to copy a formula and paste it into a 10000x200 selection. It will never end. Optimization for this is unexistant. And unlikely.

Still need to make the CSS colors match mine, but it's close enough for now.

As some may remember, a long time ago I started writing a Bejeweled! lookalike using PyQt.

While that game is actually playable, it mostly showed me I had no idea how to write a decent game ;-)

So, I have been for a couple of weeks trying to create a high level package for creating games, specifically board games (yes, I classify Bejeweled as a board game).

Well, it's working. I am writing a few libraries, and I am writing three games using it.

- Scroogle: a dumb boggle/scrabble mixture
- Nameless1: The bejeweled clone
- Nameless2: A crafty-compatible chessboard

Since I have abstracted stuff like pieces, squares, and such, these games are quite small in size, which means easy to implement!

For example, scroogle can keep score, use a dictionary for word validation, replace used tiles with new ones, so it's basically complete.

It's under 200 lines of code.

The chess game is, again, almost complete, except for stuff like saving games, or board edition, but you can play (it can't tell if you win or lose and has no timer, though ;-)

It's 210 lines of code (the crafty interface is 70).

Why am I mentioning this? Well, I think that if I ever make this good enough to release it, developing simple games for KDE will become *much* easier.

And those games will be a 80KB download, including art.

That would be good, wouldn't it? But...

... if you are thinking of writing a game, I need your help. I want you to try using this, so I can figure out if it's good or how it should be changed!

Please contact me if you know a little PyQt and want to write games!

Be it evolution in general, micro-evolution or macro-evolution.

Let's start with Gervase's three points:

- It's a basic position that you hold about the nature of reality that's not provable - it's a faith position.
- It's something that you put your trust in, and live your life on the basis of.
- It's something you urge others to believe as the truth.

Perhaps the third applies to evolution, The other two are garbage.

Let's start with the first one. There's this nifty concept called falsifiability (sorry about the spelling). Something is falsifiable if it can be proven false.

Things that are unfalsifiable are not apt for rational discussion, since the truth value behind the proposition is unknowable.

Here's an example of something that's unfalsifiable: there is a supreme being that commands the whole universe, yet is not part of it, and can not be contacted.

Here's something that's not: random mutations get selected by the environment to prosper.

That is, because it can be proven false. If there were no mutations, it would be false. If exposing different breeds to environments causes no differential in survival rate, it is proven false.

Yes, this is the micro-evolution Gervase accepts. Macro-evolution simply extrapolates from this generally accepted theory into a larger one. Is it the right one? Not sure.

There is no imaginable experiment that can prove the inexistence of god. That's why god's existence is a religious matter, and not a scientific one. The existence of microevolution can be seen, and even experimented (ask any guy that happens to have a few hundred fruit flies in a bottle).

As for macroevolution, well, as mpyne says, there is a thinkable experiment. And even if it isn't technically feasible, there can be a rational discussion, regarding the recombination speeds, and radiation levels producing mutations, and volumes, and timescales.

So, discussing it is a rational process.

As for the second point: I doubt anyone lives his life based on evolution, although it can provide some reasonable tips about how to expect things to happen, but that's mostly reasoning by analogy.

Now religion.. one of its purposes is usually to describe how you should live.

So, what Gervase is describing is religion. And by his definition, evolution ain't.

Evolution theory may be wrong. That's ok. It happens to almost all theories in one way or another, usually they end replaced by a slightly evolved new version.

But religion? Well, dude, that's just another name for who the hell knows. If you accept religion as a premise, you are into voodoo land.

Maybe the universe only exists since 1987, and everything earlier is a thought in god's mind. Who knows? Presumably only him.

Maybe he's cranky and we all die and go to hell tomorrow. He's many things but not reasonable, as evidenced by his alleged handywork.

So, what's the point in arguing a subject where no rational discourse is appropriate, a subject which is the very basis of irrational discourse?

I have no idea. That's why I'm an agnostic.

The python shelve module is too limiting for the database needs of KRsN, so I'm moving it to SQLite and PySQLite.

While this adds another dependency, it's a small one, and the coding becomes way simpler. So, I'm doing it, because fighting against the tools is foolish, and shelve was not doing it without a fight :-)

Well, I got sick of the fedora upgrade treadmill.

Although I have a number of things helping me in the software management front, having to upgrade the OS in all my customers every 18 months or so is just boring.

Since they don't need cutting edge software at all, it just wasn't worth it. Being a cheap guy, switching to Novell or RHEL was not worth it, either.

So, I checked CentOS 4. A nice boring stodgy RHEL repackaging. I have been using it in selected environments (including my home box) for a couple of weeks.

It's **really** not terrible.

Ok, the default kernel it ships is terrible. Almost no peripherals work with it, and hotplug is broken like all hell, but I just got atrpm's FC3 kernel and all is fine.

The other main problems are:

- There are not enough package sources yet

- I can't find a decent APT for it. So I am using YUM.

Well, YUM **really** hurts it.

- For some reason, I can't remove software when I am offline. It hangs.
- It's sloooooooow. Damn slow. Incredibly slow. Makes APT look fast slow!
After I added a few extra repos, it takes about 3 minutes for "yum search pyxml"

But hey, it's not all that bad if you have the time. And for the real work, my boxes use custom-packaged stuff anyway. I just needed something that wasn't going to bite me later.

And since I am readjusting my life to spend more time at home, I do have the time.

On other news, I had to reinstall PyDS, of course. So, I decided to treat my own box the same way I treat my clients' and forbid myself from installing non-packaged software.

So, I **packaged** it. Mind you, that's on an argentinian cablemodem, so don't expect any kind of speed. And it won't be there for long.

Any required packages not in that place are available in the distro, or in **dag** or in **karan**

Rosario had her birthday party on saturday. She turned 0x20 (so only we geeks know the truth ;-)

Had some fun, although some of her friends are... well, let's just say they have a very limited array of conversation topics, and they are encyclopedic about perhaps the least interesting one.

But hey, the whole family was there, lots of food (My cuñaditas Maria Feray, Laura and Agustina made dips, cake, really nice stuff).

We even got an awesome picture of us that I intend to frame.

On other news, life's pretty good, business too, family is fine, I finally closed my old apartment's lease... if life were any better the government would change my tax bracket.

I am writing an article about a cool help system I'm thinking. Should be at least as much fun as the one about calculators :-)

The main bad thing about PyDS is that you need your own computer, and you need it to be connected to the nets.

I had my own box, but since I changed employment, it was a lonely disconnected box, until this week.

So, what happened in the last few months...

- I am coding some little stuff, as usual
- I am gonna get married
- I moved with Rosario (rather both of us moved into a new place)

Read The Confusion and The system of the world so I can print that "I survived the baroque cycle" T-shirt Lots and lots of other things. Mostly good, too :-)

First of all, a note: I intend all this post as encouragement to Emiliano, the author of Kalcoolus.

You see, he turned bc into a on-screen keyboard **thing** . I don't like that UI **much** .

I can't post comments on kde-look (forgot my id, or maybe never registered), and I am not going to register just for this, but it's interesting to see how the same idea is recycled over and over and over.

This app seems to be a conflation of the following:

- Writing frontends is good, because you don't need to write the hard part
- A GUI calculator has to look like the real thing, in the name of usability

Well, the bad news is: writing frontends is fragile. What you want are libraries. And if GUIs had to look like the real thing, the UI to Skype would be a numpad, the UI to Amarok would look like an Ipod, and the UI to KWord would look [different](#) .

So, please, Emiliano, put a CLI in it. Pretty please? :-)

[Trac](#) is cool. Easy to set up, easy to run, low maintenance, and you get:

- A ticketing system
- Milestones
- A webcvs-like thing for subversion
- A wiki (I mean,. what doesn't provide a wiki nowadays?)
- Bug reporting tool
- The bug reporting tool and the subversion changesets can be linked using Wiki markup (now that's cooler than it sounds ;-)
- You don't need to be root to set it up, and you don't need apache or anything else, really.

Really, really nice stuff.

On the other hand, [CherryPy](#) is a tool that lets you "publish your python objects on the web", which doesn't really mean much, but here's what I figured out:

```
CherryPy is the first way I have seen to write a useful web-based app in a
```

Example, I wrote a frontend to clamav (allowing me to remotely trigger scans of individual nodes on a network) using CherryPy and pyclamav in about 200 lines of code.

It works like a charm, it's robust, it even can be made to look nice using some sort of templating engine (haven't bothered yet).

And of course, I control that baby using a Trac project :-)

Here's the challenge: a nicer version of Qt's data-aware widgets, using PyQt.

Did I succeed? Probably not, but it was interesting to do. Maybe it will be interesting to read about ;-)

Ok, as everyone and its goat in the free software/open source/pet-keeping communities already has read, Eugenia [1] has been up to her old antics.

The latest shape of her discontent is, developers of GNOME (and lately of KDE, too [2]) are *developer-centric* instead of *user-centric*.

Basically, the gist is that developers have a responsibility to keep in mind their users desires, and follow their wishes when deciding what way their projects should go. That these "people" are developing said software in their free time and giving it away is unimportant, she says, the user's voice shall be heard. [3]

Well, I am not a developer nowadays, at least not of shared software [4] but I still recall when I was one. And I am [cranky](#). And I have little care about what people think about me. So...

Let me express my feelings when I read her editorial (and another one called the riddle of the sphinx, I think [5]).

Ptui!

I have read, in the editorials and (specially) the clueless talkbacks, the following:

- If you are going to do a half-assed job, don't put it on the internet

Well, dude, pretend it's not there. That works for me when I consider japanese tentacle porn, it can work for you and a mail client. If you pretend it's not there, it won't bother you. Then those who tolerate half-assed jobs can live a life of jolly fulfillment.

- The user has invested effort on learning and using the program, so the developer owes him something.

I suppose the solution for that is not allowing people to use the software unless they accept that it is provided as-is... oh, wait... that's already true. Never mind.

- That since GNOME (or KDE) claim to provide useful and user-friendly software, by failing to follow the user's lead we are engaging in some sort of false advertising.

Coca Cola promised me "Everything is better with Coca Cola". Then I tried pouring a can into my computer, hoping it would make it faster, or somehow better in some way. I have to tell you, those bastards won't know what hit'em!

- Posting bug descriptions in bugzilla is too much work.

Well, try fixing one someday, missy! That's WAY harder. You are saying that using a little energy is too much work, to people who spend hundred times more as a consequence.

It's like complaining that planting seeds is too much work, when you later get a harvest. Well, if you think, so, drop agriculture.

- That the development is somehow skewed towards developers, not users.

Well, to that I have many answers. Let me share a few.

1. So? You say that as if it's a bad thing.
2. Well, that's good for the developers.
3. What magic potion did I have to drink to become a developer? Developers are just users who are way too involved in the project.
4. As a lawyer once said, since phoenicians invented money, the question "what can I do for you," has a single answer. Well, think about it.
5. No, it isn't.

Finally, let me tell you the depressing part. In a comment, Eugenia said she dropped GNOME for WindowMaker. Let me show you what she said:

RE: So Eugenia refuses to use GNOME? By Eugenia (IP: ---.osnews.com)
- Posted on 2005-03-10 20:32:31

I moved back to WindowMaker, yes. At least I don't expect anything more from it, because it is not a prominent environment and I know it's a much smaller project with fewer responsibilities.

Talk about throwing the baby away with the bath water. She is simply saying that she choses to use worse software (according to her needs), because... well, I have no idea why.

Because she has low-enough expectations about WindowMaker that she won't be disappointed? That road leads to dating ex-convicts, you know.

In the end, well, I have no idea what people think. I simply don't understand the world. I mean, I intensely dislike GNOME, and even a few people involved in it, but it's just code. And code that is worth millions of dollars. And it's there, dudes! It's like finding a stack of bullion in your doorstep, and not only that, but on every doorstep [6].

Why don't people simply look up and thank the invisible dwarves who must have toiled in darkness for years in order to rip it off the earth? [7]

No, what we get instead is a bunch of jewelers asking for finer craftsmanship in the goods, or else they are just gonna buy some bullion from someone else.

You know what? Developer-centric is good. In an absolute, objective, measurable way. Because it's developer-centric you get developers. developers make code. Code brings users.

If all you attract are users because being a developer is a pain the butt, you have no developers, thus no code, thus no users.

What do you think is better for users in the long term?

So, say it loud, I'm developer-centric and I'm proud. At least this old hack is.

[1] of OSNEWS.COM fame. You decide if that's a good or bad thing.

[2] A new post in her site. I suppose she has to feed the hit-monster.

[3] That being published in a big site that lacks threaded replies in 2005, and a woman who, when she gets a complain about it, said she was doing it in her spare time, so stop bothering her. No, I don't get it either.

[4] Isn't that much nicer-sounding than open source and much less annoying and pompous than free software?

[5] Which is, I think, #4 in the list of "ten titles you should never use in an opinion piece". #1 is What I did last summer, but I digress...

[6] And by some economic miracle, it hasn't caused yet a global depression in the value of bullion.

[7] Well, at least we work in comfortable chairs.

I have my ticket, and I'm outta here!

Going to sunny (yet breezy and not hot) Mar del Plata, to my parent's home, where I will be fed and kept clean for weeks!

Man was I tired this year.

I'm playing with libapt, to try a couple of ideas about package management.

The conclusion so far:

- Yes, it can be used
- The author believes docs are for wimps
- I am lucky there is synaptic, which has a decent wrapper

I am no fan of source-based distributions. I think that for most practical purposes, a distribution where installing KDE takes over a day (I own a lowly Duron as my fast computer) is useless.

However, they are good for one particular niche.

Custom distributions. Weird distributions. Personal distributions.

I have several computers. Most of them too slow and [small](#).

As I was considering reinstalling Linux on my Libretto for fun, I looked at what was installed in it, and decided that really, 90% of it was of no use whatsoever.

The problem was, since it had Debian installed, it has a rather wide network of dependencies that simply could not be done without.

On a regular computer, that's not an issue, but in this tiny workhorse, with 16MB of RAM and a 800MB HD, it makes a lot of difference. The smallest I could get Debian to be, and still have network, PCMCIA and X11, was about 250MB.

And the performance was not awesome, either (but not terrible).

So, what would happen if instead of a regular distribution it had something like this:

- uClibc instead of glibc
- runit instead of SYSVinit
- dropbear instead of OpenSSH
- X built as kdrive (formerly tinyX)
- python/qt (or maybe python/ftk) for basic admin tools

And so on. Basically rejigger the whole software selection slanting it towards the small side, and dropping a bazillion non-fundamental dependencies along the way.

Well, it *can* be done. It's just that it's a heck of a lot of work. But here, a source-based distribution gives you a headstart.

For example, I decided to start from ucruX, a uClibc-based port of Crux. Since the native toolchain in ucruX is uClibc, I don't have to worry much about a whole class of mess that happens when you build uClibc-based binaries on a glibc-based system (it's practically cross-compiling).

Qemu lets me install ucruX and work on it somewhat faster than on the target P75 (if I could make KQEmu work I'd be happiest).

Since crux's software packaging mechanism is simplicity itself (a shell script that installs to a fake root), although it's severely underpowered (no dependencies), I can start from ucruX, hack the build of one package at a time, then put everything on a CD very quickly.

So, if you want to hack your own distribution, Crux (or some other similar kit) is quite a nice thing.

For general use... well, my requirements start at apt-get or equivalent ;-)

Now, if only TinyCC worked for more programs, that P75 would be a pocket development powerhouse!

Well, I am still experimenting with my concept-distro.

I am now up to a running PyQt using uClibc, which I thought unlikely ;-)

I completely removed all the sysv init stuff, and replaced it with runit, which has an interesting effect:

It boots to a graphical login in about 15 seconds. Inside qemu. In a 900Mhz duron. Including kernel loading.

Of course the trick is that you have the login while stuff is still loading, but I am working on that, too.

Since using runit it's pretty simple to get a overview of where the booting process is (services have dependencies, they simply get started in order, and in parallel), I will hack a system-wide ksplash-like thing on a side of the xdm (probably will end up writing my own whateverdm).

Think of it as Fedora's rhgb, only you can login instead of getting bored.

I also switched to a root-free system Ubuntu style. Not decided yet on it, but it's not hard to do (or use).

Next step: hack Knoppix HW-detection script (or rather rewrite them in a real language).

I understand why there are 743 Linux distros. It's quite a lot of fun to hack one together.

Oh, and it needs a name. It's not going to be useful for anyone, it's just a personal toy, but it needs one.

Come on, Planetkde guys, throw me names for a non-unix like linux, if you dare ;-)

The first tutorial coming from my custom-distro experiments. Since it's a good idea to start at the beginning, here is... Booting with runit.

Well, uCrux is not for me.

The main problem is what lead me to it in the first place, uClibc.

Sadly, I cannot recommend it except for embedded devices. It has some problems building or linking specific software, but that's not it. The problem is there is no upgrade route.

Since it promises to break binary compatibility every version, you have to rebuild the world to, for example, see if a bug is fixed in latest snapshot.

So, back to glibc, at least for a while.

And, since I decided to switch to glibc, why not look at Crux's child, Arch? It's downloading now.

First of all, I love Linux. I have used it exclusively since about 1994 (yeah, the last Windows I actually used for real was WfW 3.11).

Let's see how it makes no sense.

The Bin

Your system has /bin /sbin /usr/bin /usr/sbin /usr/local/bin /usr/local/sbin 6 different binary locations.

What sense does it make to split bin and sbin? It only makes it harder for regular users to use tools they can need, like netstat and ifconfig.

As for /bin and /usr/bin, it makes little more sense, if at all. Sure, put a basic, functional system outside /usr, because /usr can be a network FS. Well, who does that?

I mean, I have seen all-local and all-network systems, but I have never seen a /-local, /usr-remote system in ten years.

And I suppose someone does it, but that doesn't mean it makes sense. If you want a real, functional, unbreakable system you can use in case of network failure: use a separate partition. Or use a Live CD. Or use a floppy. All of those are more resilient than your /.

As for /usr and /usr/local... that's just a throwback to when people were scared of installing software. People should install packaged software anyway.

The Libs

/lib /usr/lib and /usr/local/lib. Just as much sense as the above.

The variable

/usr and /var. Here's what I think I heard: /usr is for unchanging application data (bins, libs, docs, etc.) /var is for mutable data (logs, spools, caches).

That way, you put /var in a separate partition and if apps run amok, your / doesn't fill.

Well... ok, I suppose. Except that the right way to handle that is to make sure your apps don't freaking run amok!

Say, logs? Rotate them by size, not by date!

Spools? Use disk quotas, and maximum sizes!

Caches? They should be space-limited.

And all services should be kind enough to figure out when your disk is about to burst and do something graceful with it.

Finally: if your /var fills, all your services will crash just as hard as if / filled. So what's the point? That you can log into the crashed box and fix it? You can do that with a full /, too.

The root of all evil

We live with the concept of a single almighty admin. Why?

If every service application had a single point of configuration and monitoring (ie: /etc/app and /var/service/app (in runit ;-)) and /var/log/app, it would be trivial, using ACLs, to allow partial management of the system.

Sure, there would be a real root for stuff like password management and such, but that's not so bad.

Why has no one bothered doing this?

Permission to barf

The Unix permission system is at the same time harder and less powerful than ACLs. That only on the last two years it has become practical to use ACLs on Linux, and that still you can't count on them in every distro is... ugly.

I could go on, but... I think you get the idea. Coming some day: a proposal to fix the mess.

I often check on the comments for the longer articles I write.

Since most of those are of a technical nature often I find someone posting a question, even years after the article was posted (thank google for that), and I like to answer them.

So, I check on my Linux Booting Process Unveiled article, and I find this:

This article is posted on this site as well, and in its entirety:

<http://www.section6.net/help/linuxboot.php> posted by Joshua Purcell at 2005-03-14 16:14:41.492867-06

And guess what? It is! Thanks Joshua!

Ok, it's not just a copy. For instance, he removed the joke at the beginning, and managed to misspell inittab in a title (as INNITAB).

I am not against copying content from my site. In fact, it says right here at the right side that you can. As long as you do the following:

- Keep my name on it
- If you change contents, clearly say so
- Link back at me.

These guys at Section 6, specifically TBonius decided to do the following:

- Replace my name with his alias. They even say all their content is (c) nobody. Well, not quite all, guys!.
- Edit and not mark it
- Not link back

I would contact them except that:

- They provide no easy way to do so
- This doesn't look like a honest mistake (like some vietnamese site did), because of the editing.

So... whatever dudes. I can write articles like this in a couple of hours, while hangover. Nice that you find it worth stealing (badly) down to the footnotes!. But hey, that's just lame. Or, as you may understand better:

Section6 1z 7h3 5ux0rz!

I am writing a sort of web-based admin tool for a client, and I had this problem: How do you validate a system user from a script?

Well, this is how:

Just get checkpassword-pam from somewhere.

Or, if you use some other sort of authentication scheme, some other checkpassword. They are meant for qmail, but they are very handy :-)

I have been writing web-based interfaces for applications for about 5 years. Nothing public, nothing very interesting, just tiny front ends for custom tools in clients' installations.

And I have hated every minute of it. PHP hurts, Twisted hurts, mod_python hurts...

For a couple of months I have been using CherryPy and I finally am having fun doing it.

And after I figured out how to do AJAX using it, it's even more fun (because the apps interaction model is not totally braindead :-)

I don't expect it to be as fun as PyQt/PyKDE, but it's totally not awful. I suppose the same epiphany comes to people when they use rails or some other decent, productive, fun framework.

All in all, I could get used to this.

I am, as always, playing with stuff. And I was faced with a problem I must have solved a dozen times before:

Given a list of items with obvious hierarchical names (say, a list of folders), turn it into a reasonable data structure.

Since there are not all that many names, there is no need to do it on-demand, or anything like that.

I must confess I suck at this kind of things because I hve to actually think them through. I don't **know** how to do this kind of things.

Mind you, I have done it before, but I said, hey, maybe google can help me...

And **yeah!** Python code from 1994 that does exactly what I needed. I had to touch a single line (`string.splitfields(a,'.') to a.split('/')`) and it worked.

The Internet never forgets!

I will be speaking at the Software Libre 2005 event, June 7 at 2PM, at the Sheraton Retiro, Buenos Aires.

I'd invite everyone, but it costs some money, so I will just say that I promise to buy a beer for everyone mentioning this blog. (Limit 10 beers, only national ones allowed)

For more info, click on "read more"

Voy a dar una charla en Software Libre 2005, el 7 de Junio a las 14 horas, en el Sheraton Retiro, Buenos Aires.

Invitaria a todo el mundo, pero cuesta dinero, asi que solamente los invito con una cerveza si mencionan esta pagina. (Limite 10 cervezas, solo nacionales)

Para mas datos, click en "Read more"

I think some months of formal math training would do wonders for the average guy. And I don't mean algebra, I mean logic, set theory, that kind of thing.

The obvious example is, of course, comparing two things. There is the concept of the well ordered set. In a well ordered set, for any subset, there is a larger and a smaller member.

In other words, if you compare any group of things, one is the largest, one is the smallest.

That may seem obvious, because people only think in terms of integer or real numbers when they see $>$, but real life is different. Real life objects are usually not well ordered, and that means you can't sort the whole set, nothing ever comes out on top.

Because the crappy math people learn makes them think in linear terms [1], reality is approached in linear, limited terms.

what I am thinking about:

People say Qt is better than GTK+ or viceversa.

Well, those are multidimensional objects you are comparing, and in order to make it meaningful, you need to establish a metric that will make the toolkit-space a well ordered set.

That metric, sadly, is subjective and personal. While Bruce Perens[1] may see the acquisition cost for proprietary developers as the most significant part of it[2], others may see technical features as more important.

But the telling point is this:

If you are a free software developer, why would you choose GTK+? What is your metric, where GTK+ is larger?

I really have no clue on that, since I haven't touched GTK+ since I recoiled in horror from it in early 1996, but I will try my best.

- You prefer coding in C
- You want your app to work well with GNOME
- You find the GTK+ API appealing
- You like some GTK+ development tool, like glade
- You see better performance in GTK+ than in other toolkits

I can't find any other reasons, and even one or two of those seem iffy to me ;-)

You prefer coding in C: Ok, that's reasonable. If you want to code in C, GTK+ is probably the way to go.

But why, oh, why do you want to do that? Really, come on, C is a poor language for almost every use. Your language metric is probably the guilty part of your choice of GTK+. But that's another set.

You want your app to work well with GNOME: Ok, that is reasonable. I could say Qt apps work well with GNOME, too, but that's my own subjective bias, and really not defensible, since GTK+ apps *do* work better with GNOME.

The GTK+ API is appealing to you: you are a sick person? ;-) Ok, that was harsh: you have bad taste in APIs? Ok, that was still harsh: I don't understand you. Yeah, that's better.

Luckily, you don't have to use it, you can use decent wrappers, nowadays.

You like a development tool, like glade: I don't see glade as a big thing compared to others, but hey, that's just my taste. Reasonable point.

Performance: You want performance? Use fltk. Even pure Qt seems to be about on par with GTK+. Of course if you pair this with the C argument, they synergize. After all, GTK+ does have better performance than Xaw3D or somesuch. Maybe XForms?

Please, if anyone reads this, I am not trying to flame GTK+ or its developers. I am sure GTK+ is the greatest C toolkit around, and developed by smart people.

I just also don't see any compelling reason to use the thing for free software development. For proprietary coding? Maybe, if you are low in cash. My own metric (for whatever little it's worth) doesn't make me see GTK+ as an interesting tool. If yours does, please help me improve it.

Yes, I know stuff like saying the API is ugly can't be improved, because you can't convince me that it isn't[4], but you can help provide new reasons, or strengthen others, or maybe tell me bad points about the toolkit I see as better.

[1] Ok, people think that way even if they don't learn math, the thing is, math *teaches* you to think non-linear, or multidimensional.

[2] An interesting fellow. Remember when he quit Debian in a huff?, and that he was going to develop a Red Hat-based distro that was more desktop-oriented? That was probably around 1998 or 1999, IIRC. BTW: you won't find all of it in the Debian archives, "someone" removed it.

[3] Which is really a bizarre argument, isn't it?

[4] I also think most of Dalí's and Picasso's paintings are ugly. How can anyone convince me they aren't? They can't.

Ok, not really, since SSH has made my life much simpler than it would be otherwise, but really, it has some usability issues.

And I mean **real** usability issues, not the usual crap.

- It can't be integrated into kdewallet

While there is a mechanism to have a GUI asking the password, this helper app (askpass) doesn't get any session info, so it's meaningless, unless you are trying to directly start a X app over ssh.

Which you probably aren't.

- Fingerprint management sucks.

Suppose you have a firewall. You keep port 22 as a way to log into it, and forward port 23 to a mail server in the DMZ. Well, it will complain and print huge, scary warnings each time you login into one or the other, depending on which one you used first.

Or, it can simply refuse to connect.

And that's just the easy two.

What can be done?

- Take the dropbear client (not openssh, dropbear code seems simpler), and put a putty-like UI into it. Use the konsole kpart for display.
- Take the GTK version of Putty and hack it into KDE shape, put kdewallet in it. I don't quite like the idea of having a separate, different terminal app for remote sessions.

I would probably go the dropbear route if:

1. I had a working PyKDE (maybe someday)
2. The idea of delving into someone else's C code didn't make me nauseous. (probably after I surgically remove my sense of taste).

Weird article in dot.kde.org.

I don't know what Navindra has been drinking, but it does wonders for his writing style!

Well, reading in planetkde about how [nice](#) data aware widgets would be, I have to say this:

- Data aware widgets **are** great
- Data aware widgets in C++ are not the best possible solution

Using a higher level language, and specifically a more dynamic language makes lots of things much simpler.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "here".
```

As a tiny, lame example, please check the little thing I wrote about data aware widgets in python [here](#).

```
system-message
WARNING/2 in everything.txt, line 3740
Duplicate explicit target name: "here".
```

I am sure that some of our better programmers (or more creative thinkers) can come up with awesome stuff if they divorce from C++ in this subject :-)

I will be speaking about this subject in 48 hours.

Here's the challenge: post as a comment (if you see this in planetkde, you will have to click somewhere to get to my page) describing your favourite innovative unusual and/or little known feature in KDE!

In a few days, I will post the whole thing as a .swf file somewhere so everyone can see it.

My goal is to make it dazzling, feature-packed, and *very* fast. If I can get to 100 features in 1 hour, that's about right.

So, what is it?

There are some problems with some of the KDE apps docs. The biggest are: Superficial:

They explain too broadly, not covering any specific details.

Uneven:

They explain some things in great detail and others not at all. For example, the kcalc docs explain what the lsh operator is. They don't explain what the M+/M-/C/CE buttons do.

Unfocused:

Being uneven is right, as long as what you don't explain is obvious (for example, not explaining the button 9 is good).

These problems are common to almost any technical document, and they are not easily fixed. In fact, the only known fix is having a very good technical writer, and a **very** good technical editor.

Sadly, those are very hard to find. And usually they won't do it for free.

So, here's a proposal for a solution that may make things easier for KDE4.

Open source the docs

Oh, sure, they are open source now. Anyone can hack them. As long as he knows how to get to their source code, which is in docbook.

I understand the docs team is willing to take plain text submissions and format them accordingly, but you can't quite figure that out from the docs themselves.

Even then, the barrier to entry is high, you have to write your explanation, send it through channels, and noone can see it until next release.

My proposal is to integrate in the KDE help viewer a limited wiki-like capability.

The user would have a "annotate this page" prominently displayed somewhere. And he could use that to attach annotations that should be clearly visible on the docs.

And, most importantly, he should be offered a chance to share those annotations, which would be uploaded to a wiki-like system. Said system would be moderated by the current docs team people, who could approve of them.

Then, the doc viewer should also have a feature to update the docs for a version, downloading new pages and annotations from that same system, completing the feedback loop.

Said annotations would then be shown along the docs, and could even be marked as "tips" which would be added to the tip-of-the-day system.

Should be a bit of work, but the KDE hackers are more than up to it, I'm sure :-)

Answer to comments on my previous docs item. First of all, chillout people. This is just the blog of a retired guy. It's not exactly Infoworld, ok?

Sorry if I ruined your day, that was not my goal. I had an idea. Nowadays, being a retired guy, when I have an idea, I either code it for myself or put it in my blog so others can read it. Sometimes it works, sometimes it doesn't.

It was not my goal to say that the writers or editor's job is bad. I meant to say that it can get better, and I had a small idea that may help there.

Rainer: hmmm... well, I translated all of KDE 1.0 (or maybe earlier?) to spanish once, before there was a translations team ;-) but I am sure that doesn't count.

I mentioned in the article that there is no need to write docbook, and that you can send plain text to the docs team. I really really did, so don't ride me about it.

BTW: the docs don't say *where* to send that plain text email. For example, the kcalc manual says the documentation is copyrighted by Bernd Wuebben and Pamela Roberts (in an obfuscated manner, but that's ok).

I am damn sure that if you send Bernd an email with corrections for kcalc's manual, he won't apply them, because he has been out of KDE development for 5 years. Pamela Roberts I don't know.

So, who exactly is one supposed to send a 20 line explanation of the M-buttons to?

Oh, I get it, you are supposed to figure out that docs.kde.org (which is not mentioned in the manual) has a "Contact us" link... which sends you to <http://www.kde.org/documentation/> which then suggests you learn docbook (which is not necessary) and go to <http://i18n.kde.org/doc/index-script.php> where you are told to contact the app author and kde-doc-english@kde.org.

For a 20-line writeup? Does that sound reasonable?

Maybe the template used to generate the docs could be modified to add something in the last page, like "For contributions or corrections to this manual, please send email to xxx@xxx.xxx".

I mean, if that doesn't ruin your day.

But hey, I can see the attitude of Lauri is more open, at least. And yeah, Lauri, thanks for your job, I'm sorry I dropped the app later :-)

Sorting through the really bad garbage is not all that difficult. Would I offer to triage? Sure, if it's made convenient. Make it come as clearly tagged mail with, say, two hyperlinks in it, one for sending to docs team for further review, one for declaring it crap, I can do a hundred of those a day.

Canllaith: I share the feeling about wiki markup. RestructuredText is nice, if you find a wiki that supports it.

I am not too active in the public speaking business anymore (not that I ever charged for speaking, either ;-).

A few years ago, I would do it about once a month, now it's about twice a year. But I have done it some 50 times, with crowds of up to 1000, and I think I am pretty good at it (but getting worse by the minute). And I have been to a million or two.

And for some reason, I have decided I want to talk about it. Many of the KDE guys & gals have to do it every once in a while, and a bunch will do it in aKademy so, maybe this can help someone :-)

Of course, this is surely the wrong way to do things for 90% of the people, so read without any care ;-)

Entertaining is better than boring

Obvious, right? Well, then why are 99% of these things so dull?

The main reason is, of course, that the one speaking is scared (not nervous. *Scared.*) , or that he really is a dull person on first contact.

Most of the free software people I know are not dull at all. But when you first talk to them, they are about as interesting as a brick.

That's probably shyness. Most extroverts don't make a living by staying immobile in small places and looking at the same three objects for hours at a time.

So, how can a shy person be interesting quickly? By not trying. For Christ's sake, don't make a joke (unless it's really unavoidable [1]). Just try to chill out, and speak about what you want to say, in a natural tone. Stammer if you have to. Think before answering questions.

Specially, allow the audience to ask questions *during* your exposition, if you can do it. Who knows, it may not let you say what you wanted to say, but it will let the audience learn about what they want to learn, which is just as good.

It helps a lot if you talk about something you **really** know. Don't try to overreach. I couldn't speak about C++ development because I am not good enough. Sure, I *could* give a crappy lecture. Just not a good one.

So, if you have to, stay on a simple topic. The audience will lead you from there, but if they lead you somewhere you don't know... well, say you don't, and then you both can try to figure it out or ask someone else.

Avoid slides unless absolutely necessary

One problem is that they will believe the slides are the lecture. They aren't.

If what you have to say can be said in 30 pages with 6 lines each, then why the hell are you taking 45 minutes to say it? If it can't, then what are they for?

They lock you in a path. Unless it's absolutely necessary that you get a specific message out, I say don't bother.

Use graphics? Sure. Use a live app showing how to do something cool? Sure. A blue background with a picture of app A saying "app A can do something cool"? Yeech.

To connect it with the other one, some people think slides will make things more interesting. Guys slides are incredibly boring. They are more boring than the speaker making funny shadows on the screen.

What the heck, put some nice screensaver hacks and leave it at that.

[1] I was once showing how to rip MP3s in konqi and asked anyone for a audio CD. I got one from a 16 year old. Written in huge letters on top of the CD was "crappy music". I won't say what I said, but you can say your own in the comments ;-)

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "here".

Just finished reading an article at Timothy Butler's site (see [here](#)) and it was a weird feeling.

On one hand, I find the idea of the immorality of avoiding ads strange. I mean, I am watching TV, ads come, I flip. Instantly. That's why I like that my TV has a way to program it to cycle through selected stations. That way I can keep coming back easily, until the ads end.

On magazines and newspapers... well, if someone came and told me that flipping through the ad pages is morally wrong, I would really not give a damn.

But the weird thing is not that (or rather, the *other* weird thing is something else). His argument goes like this:

- a) TT has to make money
- b) The way to make money is selling their intellectual property in some cases (in others, it's given for free)
- c) Timothy says that's not ok.

The other hand says:

- a) People who write on websites has to make money.
- b) The way to make money is to put ads on the site.
- c) People are avoiding the ads.
- d) That's not ok.

Those two ideas don't seem to have much in common, but lets rethink the issue, in terms of providing intellectual preperty, be it a website or software.

Suppose TT would make Qt adware. What do you think Timothy's position on Qt would be if he could use it for anything, but the app would always show an animated ad using 30% of the screen. With sound.

That is what websites are providing. Intellectual property as adware. TT is providing intellectual preperty in two ways:

- As free software

- As proprietary commercial software.

Of course we know that both properties are the same code, but it is provided in two ways.

I find both delivery mechanisms vastly superior to adware, or nagware. Timothy doesn't.

So, in the same way he suggested KDE and TT to part ways, I suggest how he can avoid all his trouble with adblockers.

Timothy Butler should turn ofb.biz into a paysite. He should charge you if you want to read it.

Or, he should be upfront about it like salon.com is: see this ad, then you can read. Of course now I check salon maybe once a week.

So, the main problem would be, for that solution, that probably noone would pay to read it. Well, welcome to capitalism, Timothy.

```
system-message
WARNING/2 in everything.txt, line 3921
Duplicate explicit target name: "here".
```

Have been reading the planets lately (I mean planetkde.org and planet.gnome.org, not astrology) and run into posts by [Aaron Seigo](#) and [Luis Villa](#) which are, let's say, interesting.

Luis' post took me to this [page](#) which is interesting too, and I would like to see something like it for KDE (and I am sure it is somewhere, but I can't seem to find it)

And I don't mean the page is interesting only for having Australia listed as an asian country ;-)

Some of the items talk about hundreds of thousands (or hundreds of millions) of desktops, and others talk about 11 seats.

Is there nothing in the middle, or is it just not reported?

I decided to put out another datapoint.

Here in Argentina, the best-selling OS has KDE as the default desktop. It's a Linux from [Pixart](#), and is more or less what on other countries is sold as Xandros.

It seems Pixart made some development work for Corel, and then for Xandros when they bought the linux distro business. Their boss is the former (?) boss of Corel argentina, too.

Almost every whitebox clone is sold with one of their distros installed and preconfigured.

Oh, sure, most of them get wiped out and replaced with a stolen windows xp in 24 hours, but it's quite a number. Think 100K or 200K sold each year, at least.

Of some concern is that some of the GNOME deployments used to be KDE deployments. For example, the Sao Paulo telecentros used to be Conectiva boxes with KDE (and windows, in dualboot).

The City of Largo used KDE for quite a while.

But what the heck, we are both desktops squeezed into a ketchup bottle, there's a whole world outside to spread into ;-)

To escape the heat of the afternoon, I dived into a movie theater downtown, and the movie I hadn't seen was this.

I am a fan of Samuel L. Jackson. I think his role in Pulp Fiction is incredible. I think his screen presence (and a little acting) makes his characters interesting and intense, and larger than life, which is a good thing in many movies.

But why, please, is he trapped inside of a fascist fantasy movie?

In this movie, 4 policemen and a prisoner (who is described as evil, sure) are in a car, and because of something he says, just a nod from one of the policemen prompts the other to hit the prisoner. Who is cuffed, BTW.

Ok, so that is a felony committed by one of the "good guys", with acquiescence from another. It is also, of course, a cowardly action, and think about it: just a nod, and whack. That shows habit, so the subtext is apparently that these nice guys, members of "the most respected police force in the world", routinely hit defenseless prisoners.

Yup, those are the *heros* in this movie.

But let's put aside the obviously rotten ideology this movie represents, where it's ok for a character, just because he is labeled as "good", to break the law without consequences, and without even suggesting that such a thing may be, you know, not the best thing to do.

Let's talk about cinema. This is the kind of movie where two characters of similar height and build fight, in a dark railroad set, backlit.

For minutes, they punch, and kick, and throw themselves at each other.

And it's physically impossible to see who the hell is beating the other, because you can't tell them apart.

That's lazy filmmaking, and it shows this morally bankrupt movie is not even well done garbage.

At least Leni Riefenstahl's movies are allegedly beautiful to look at. What's the excuse for this one?

For a bunch of apps I write, I often want to be able to add a systray icon.

But... I write them using PyQt, and the systray stuff is in PyKDE.

But... Torsten Marek did write a module to do that. The only problem for me is the python-ctypes requirement, but it's no big deal for my apps that are not massively deployed.

```
system-message
```

```
WARNING/2 in everything.txt, line 3  
Duplicate explicit target name: "here".
```

You can find his code, in a somewhat mangled form, [here](#)

```
system-message
```

```
WARNING/2 in everything.txt, line 3992  
Duplicate explicit target name: "here".
```

And since building extension modules is not **completely** trivial, here's a simple `setup.py` that will do it:

Put it in the same folder with his code, then you can do `python setup.py install` or `somesuch`, then you can use it in your app like in Torsten's `systray2.py` example, and it will work.

Just for the record, I have recently converted to Flying Spaghetti Monsterism.

```
system-message
```

```
WARNING/2 in everything.txt, line 3  
Duplicate explicit target name: "here".
```

More information [here](#) and [there](#).

system-message

WARNING/2 in `everything.txt`, line 4003
Duplicate explicit target name: "here".

A Little History

Many moons (almost two years!) ago, I wrote an article called [Small Linux](#) detailing what I had done to make a reasonable Linux fit in a Toshiba Libretto 50.

That's a very limited notebook, with under 800MB of disk, a 75Mhz Pentium CPU and 16MB of RAM, and it has served me well for a long time.

Now, however, I am using the almost exact opposite, a second-hand Toshiba Satellite 1955-S805.

Where Salma (the Libretto) had a 640x480 screen, Monty (Toshiba), has a 16" 1280x1024. The RAM has increased 32 times. But they have one thing in common...

The 800MB HD

You see, Monty is second hand. My future mother-in-law and brother-in-law brought it from New York when they visited, for Rosario (My future wife, until February 18 2006 ;-).

And it had a broken HD. And I wanted to use it while I got a nice new 60GB one.

So, overcoming my fear of destroying expensive equipment, I got the HD out of Salma and into Monty, and started thinking....

The Tallest Guy In The World

He had a problem: really bad feet. He died of it, too. In the same way, Monty now booted, but the app selection was outdated, and really, lots of things Salma couldn't do, Monty could.

What on earth can one install on that disk when you don't have any other real hardware limitations?

The choice of distribution was tricky.

I am a CentOS guy lately, but the package selection is entangled enough that you can hardly get X installed without crossing the 800MB. The **minimal** install is about 450MB.

Debian again? Well... no.

Knoppix?

Now, **that** has some serious potential, since I could run the OS from DVD/CD, and then use the whole 800MB for data. But I wanted something where I could choose what to install.

I could have gone the path of one of the modular Knoppix derivatives, but it was yet another task on the pile.

So, I went with.... ARCH.

ARCH Is Arch

Yes, ARCH is saucy. It installs in roughly 200MB, including kernel sources and GCC. That's quite small.

I managed to create a reasonable desktop in about 550MB, including:

Scite: A nice text editor

Python: Needed it to work in a few proggies.

Firefox: I have the RAM. The CPU is a 2.56 P4.

Xorg 8.2 with Nvidia drivers: And just for kicks, 3ddesktop to see if they work ;-)

fluxbox: I know it, and it's nice enough.

putty: Nicer than xterm, and has a handy SSH builtin.

ROX Filer: Not really all that useful, but what the hell, for 3MB you get sidebars, a file manager and a few extra tricks.

Slim: A very nice xdm replacement. Specially with the mindlock theme :-)

upx: Compress the exes. Save 30MB. Good.

CherryPy: It's what I am working with. And it's only 74KB.

habak: Simple, small tool to set the X root decoration.

rsync: Stay synced with my older desktop.

All included, I am at 120 packages, using 557MB of disk (with **extensive** trimming, see the original article for some examples).

So, what's the difference between this set of apps and my previous choice....

Well, look at the result of free:

	total	used	free	shared	buffers	cached	
Mem:		512544	408924	103620	0	33468	162916
-/+ buffers/cache:			212540	300004			
Swap:		0	0	0			

Just for laughs: here's the old one, when doing roughly the same things: editing an article, browsing the web, a few terminals:

	total	used	free	shared	buffers	cached	
Mem:			14708	14144	564	4644	816
-/+ buffers/cache:				8024	6684		5304
Swap:		47992	18880	29112			

Scary isn't it? I am using roughly 25 times the amount of memory I used on the libretto. It's easy to see why, tho.

Consider the desktop. It shows a pretty picture. It is 1280x1024. It is in millions of colors. That is in RAM. That is either 3932160 or 5242880 bytes. On the libretto, I was intentionally not using anything there :-)

So, it really is not comparable anyway, and Monty's life as a malformed box will be short. But it was quite a bit of fun :-)

[CherryPy](#) is a cool, pythonic, simple, quick, **fun** way to write web applications.

I often use [CherryPy](#) to write custom web admin tools for customers. Suppose you want to provide them with a simple way for password management. Usually I have the following requirements:

- It must be simple (not webmin)
- It must not be a terminal session (bye ssh :-)
- It must not be a graphical session (sadly, that leaves out PyQt :-)

- It needs to do custom stuff: set the samba password at the same time, send a mail warning about the next forced change, whatever.

Someday I may be able to use a single-app freeNX session, but right now that's a bit too much problem for different reasons.

So, I wrote a [CherryPy](#) page. Over time, I have become quite fond of it, and wrote a bunch of small tools around it. One of them was a way to login the user into the site using the system's users and passwords. Now I got to throw it away :-)

The new [CherryPy](#) 2.1 has a mechanism for implementing password-protected pages, called the [Session authenticate filter](#) which is sadly not documented yet anywhere I can find.

So, here is my attempt, so people googling it up can use it. Excuse me:

```
cherry.py sessionauthenticatefilter cherry.py sessionauthenticatefilter cherry.py ses-
sionauthenticatefilter cherry.py sessionauthenticatefilter cherry.py sessionauthenticate-
filter cherry.py sessionauthenticatefilter cherry.py sessionauthenticatefilter cherry.py ses-
sionauthenticatefilter cherry.py sessionauthenticatefilter cherry.py sessionauthenticate-
filter
```

That should do it :-)

What you need first is a function that takes a username and password, and returns **None** on success, or an error message for the failure.

For example, I can adapt something I wrote earlier using `checkpassword-pam`

Also, you may want a function that returns the login screen. If you do, remember the following:

1. It must set the form action to `doLogin`
2. The user field should be called `login`
3. The password field should be called `password`
4. You will take a `fromPage` argument that you should pass through, so the user will end on the page he wants.
5. You will take a `errorMsg` argument which is probably the result of a failed previous login. Display it red or something like it. Unless it's empty, in which case it should not be visible.

Here's mine.

Although I am using a template to display it nicely and with the right style, it should be pretty obvious how it works.

You **could** use the default login screen provided by the filter. While it works, it's just ugly.

Then you need to apply the filter to the set of your pass-protected pages. Suppose you want the whole site to be protected, except for your `/static` directory, which contains the stylesheet, images and such. Then you put this in your configuration file:

```
[/]
sessionAuthenticateFilter.on=True

[/static]
sessionAuthenticateFilter.on=False
```

Next thing is to hook the session authenticate filter to your custom auth code. In your app, do the following. It seems that you **can't** do this in the config file, though, so do it in code.

And that's it. Now your site is password protected. You can even have different authentication schemes for different pieces of the site, by resetting the hooks for the folder you prefer to a different function.

Also, I think this is a good example of why I like [CherryPy](#). This mechanism is both flexible, powerful **and** simple.

Nuevamente este año voy a estar en [CafeConf](#). Es una charla de 45 minutos sobre PyQt el 13 de octubre al mediodía.

La idea: La gente se sorprende cuando uno agarra KHTML, engancha un par de widgets y sale con un navegador web. Se deberían sorprender más de que uno puede partir de una receta de 20 líneas en una página web y terminar con una planilla que funciona ;-)

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "stupidsheet".
```

Por lo tanto, voy a mostrar [StupidSheet](#) como un ejemplo de que el desarrollo de software con interface gráfica es mas facil de lo que la gente cree.

Como siempre, si mencionás esta página, te ganás una cerveza. Máximo 2 cervezas, no muy buenas.

I will be at [CafeConf](#) again this year. It's a 45-minute thing about PyQt in October 13th, at noon.

My idea is: People are amazed when you hook Khtml to a couple of widgets and write a lame web browser. They should be more amazed that it is possible to start with nothing more than a 20-line recipe from a website and end with a functional spreadsheet ;-)

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "stupidsheet".
```

So, I will be showing [StupidSheet](#) as an example of how writing GUI software in Python is simpler than people think.

As usual, if you mention this page, you get a free beer, maximum 2 beers, and not very good beer.

```
system-message
WARNING/2 in everything.txt, line 4225
Duplicate explicit target name: "stupidsheet".
```

A long time ago, I wrote a piece about how I didn't like kcalc. It contained a very lame pyqt script showing a (IMHO) nicer calculator. Strangely, that lead to two very cool implementations of the concept!

One of them was written in Ruby, the other one in C++. I think that has some potential.

A few months later, I wrote a spreadsheet based on the same concept. Also based on PyQt.

This StupidSheet has some conceptual problems. Like, if you want to import Excel sheets, you would have to rewrite basic in python, so it's not a practical program, but it is a rather nice example showing programming using dynamic languages.

In fact, I used it as such last week at CafeConf.

Now, here's the challenge. If people that know how to write Ruby or Java apps using Qt (or KDE, why not) could write a similar application, we all could write a comparative guide to Qt/KDE programming on different languages.

Since we would all be starting with a not-too-complex, but really non-trivial example, and we would all do the same one, it should be pretty unbiased.

In fact, if you think this example is biased, please propose another one, and do this thing anyway.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

You can find StupidSheet [here](#)

system-message

WARNING/2 in `everything.txt`, line 4246
Duplicate explicit target name: "here".

It has some small bugs (try setting B1 to A1+1 with no value in A1 ;-) but they are easy to fix.

We could remove some features (like the weird pasting stuff) to make the example more didactic.

I hope this gets some answers :-)

I was trying to do something weird: take one of my PyQt proggies, and compile it into a standalone binary using Jython, koala/qtjava and gjc.

Sadly, it doesn't work :-)

Has anyone ever gotten a rather recent gjc and jython to work together?

The best I can get is this:

This is using gcc-java-3.4.3-22.1 and jython-2.2-0.a0.2jpp

I mean, this is completely unnecessary, but it would be a nice hack :-)

Well, I am in the middle of migrating a rather large mail server, using qmail+vpopmail+mysql+courier-imap.

And just when it's starting into production usage, we find the error in the title.

A lot.

Sadly, since that error indicates a failed MySQL query, it means you get stuff like random auth failures. Which sucks **a lot**.

Googling gave a lot of hints. But nothing would fix it.

Until.... lucky strike.

So, I will put my solution here for future googlers.

If you see this:

```
vmysql: sql error[3]: MySQL server has gone away
vmysql: sql error[3]: MySQL server has gone away
```

```
vmysql: sql error[3]: MySQL server has gone away
vmysql: sql error[3]: MySQL server has gone away
vmysql: sql error[3]: MySQL server has gone away
```

Do the following:

1. Check if you are exceeding mysql's max_connections parameter. The default is 100.
2. Check if your cpu can take the load. (No, don't look at the load average. Look at the CPU usage).
3. Make **really** sure that vpopmail and courier-authlib are built using the exact mysql you have installed. No, it doesn't matter if it's the same version. Go, and rebuild the things!

That fixed it for me.

On other bizarre news about this server, I have low CPU usage (95% idle), low IO (iostat reports under 1MB/sec, 40 tps), yet I have a load that can go up to 120.

Yet, interactive response is fine.

Weird.

Just in case anyone is wondering if you can install the Xen hypervisor on a CentOS box:

1. Sure. It works just fine.
2. Ok, there are a couple of glitches in the install.

Here's how I did it, on a CentOS 4.2 box:

- Get the install script for RHEL 4.1
- Fix the validate_os_version function:

```
validate_os_version() {
    #grep "Red Hat Enterprise Linux AS release 4 (Nahant Update 1)" /etc/re
    true
}
```

- Run it
- Fix your grub.conf adding this :

```
title Xen 3.0 / XenLinux 2.6
    kernel /boot/xen-3.0.gz dom0_mem=262144
    module /boot/vmlinuz-2.6-xen root=/dev/hda1 ro console=tty0
    module /boot/initrd-2.6-xen.img
```

- Fix your glibc by doing this:

```
mv /lib/tls/ /lib/tls.disabled
```

- Boot on the Xen option from Grub, and you already have a CentOS Dom0. Quite simple, wasn't it? :-)

Main problem: The Xen kernel you are booting sucks, because it supports no unusual hardware. So, it would be a good idea to hack a decent Xen-linux-kernel similar to CentOS unsupported one.

Now it's time to start toying with CentOS partitions, but they are supposed to be simple.

I have been trying to figure out how to use libsmoke from kdebindings for a few hours.

So far, I have been unable to find any kind of documentation whatsoever.

If anyone knows of something (short of reading the ruby bindings, for instance ;-) I would be thrilled.

Addendum: I just found on google that there is a whole language with Qt bindings I had never heard of!

<http://www.clifford.at/spl/>

I just realized I have not learned a whole new *real* language in almost 5 years.

I have learned a few dialects, some minor stuff to work around or through an obstacle in some project, but I have not written a whole program in anything but python since 2001.

So, since I **don't** want my brain to turn into muesli, I will now proceed to learn **Io**.

Nice OO language, prototype-based, which is something I haven't run into outside of JS, I think, a sort of LISPy flavor, with a dash of smalltalk for piquancy.

Maybe I will get really bored with it, maybe not :-)

And yes, the previous post was about having the idea of hacking a Qt binding for it, but it's probably not doable right now, since extending the language is pretty much undocumented, and extending it in C++ is unheard of.

To top that, it uses continuations, which probably cause all kinds of hell on Qt's memory management :-)

Just finished implementing a network backup solution for a customer using **Bacula** and reached these conclusions:

- Bacula is hard. But that is mostly because the problem is hard. You need to backup 30 different computers over a net, in a secure manner, with different data sets for each... well, it is going to involve defining 30 datasets and so on.
- Bacula is very good. It supports VSS, which alleviates the classic windows "that file is open you can not read it" problem.
- Backing up Windows sucks.
 1. It is pretty hard to know what to backup exactly to catch all configs and user data. Not so much on XP, but on older versions... yikes.
 2. You pretty much can't ever be sure you are allowed to back up everything. I have a fileserver with files the local administrator can not read. So, it seems a user can create unbackupable files!
- I still need a decent system backup for Linux. I have used Mondo for a long time, but it's a pain in the butt, and it's getting more painful as time passes. I want something that I call and I get a nice DVD with the whole system in it. If you have a suggestion, please drop a comment ;-)
- A system that automates the backup policy as much as Bacula does is great. Except that it makes it quite hard to guess the exact amount of storage (I am doing disk-based backups) you will need.

- For simpler stuff, you should use [flexbackup](#). Their slogan is *amanda is “too much”, and tarring things up by hand isn’t nearly enough* and it pretty much is what it does.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “[rdiff-backup](#)”.

For personal stuff, [rdiff-backup](#) is awesome. I really should write a graphical tool for it one of these years :-)

system-message

- **WARNING/2** in `everything.txt`, line 4421
Duplicate explicit target name: “[rdiff-backup](#)”.

This is what I understood of Smoke so far. I may be way off, since it is C++ sorcery of a higher level than I’m used to, but I really think I am getting the hang of it (and a bunch of thanks to Richard Dale and Ashley Winters who are the ones that made me understand so far. Any mistakes are my fault, any good thing is theirs ;-).

This piece is only half of the story, though. Maybe one third.

Concept

Since Smoke’s goal is to help you write bindings for languages other than C++, it provides a way to access Qt’s API. The original thing about Smoke is that it does so by providing you with a smaller, more *dynamic* API that maps onto Qt’s.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “[here](#)”.

You could write a Qt C++ program using Smoke as the API instead of using Qt’s. In fact, you can see it [here](#) written by Ashley Winters.

system-message

- **WARNING/2** in `everything.txt`, line 4443
Duplicate explicit target name: “[here](#)”.

I had to rename it `hello.cpp` to make it work because that looks like C but may not really **be** C ;-)

As you can see, the Smoke version is quite a bit more complex than the Qt one. But that’s ok, remember that the goal is a binding, which means that what you need to make your life simpler is less API... which is what makes the program more verbose.

Let’s examine the Smoke `hello.cpp` in detail.

One key point is `callMethod`:

```
// call obj->method(args)
void callMethod(Smoke *smoke, void *obj, Smoke::Index method, Smoke::Stack
    Smoke::Method *m = smoke->methods + method;
```

```

        Smoke::ClassFn fn = smoke->classes[m->classId].classFn;
        fn(m->method, obj, args);
    }

```

If you have programmed in Python or Ruby or a number of other dynamic languages you may guess what this does already.

It takes as arguments a number of things which still have to be explained but the main gist is there is an object, there is a method, there are args, and it ends calling `obj->method(args)`.

The first argument is a Smoke pointer, which is the big object in the Smoke library, created in our program by the `init_smoke` function.

A Smoke object is a strange beast. It contains a description (in this case, because we got `qt_smoke`) for the whole Qt API.

You can find classes in it indexed by their names, and methods for each class indexed by their names and types of arguments.

That is why you can have a generic `callMethod` that will work for any kind of object and for any method in the class, because **all** of them are somewhere in the Smoke object.

The second argument is void *obj* which is the object itself we are manipulating. So, if you are trying to call `QLabel::setText`, it will have to be a `QLabel` casted as `void*`.

In the `hello.cpp` example, we even create these objects using Smoke's API (see later).

The third argument is a `Smoke::Index` which is what Smoke uses to find the requested method in its method table. This Index we get using the `getMethod` function, which is the second key piece of code:

```

// given class-name and mangled function-name, return an unambiguous method
Smoke::Index getMethod(Smoke *smoke, const char* c, const char* m) {
    Smoke::Index method = smoke->findMethod(c, m);
    Smoke::Index i = smoke->methodMaps[method].method;
    if(i <= 0) {
        // ambiguous method have i < 0; it's possible to resolve the
        fprintf(stderr, "%s method %s::%s\n",
            i ? "Ambiguous" : "Unknown", c, m);
        exit(-1);
    }
    return i;
}

```

Here is an example of a `getMethod` call, where we are getting `QApplication::setMainWidget`

```

Smoke::Index method = getMethod(smoke, "QApplication", "setMainWidget#");

```

As you can see, we search for the method using strings of the class name and method name. Except for that pesky # at the end of `setMainWidget#`.

That is a basic argument-type-mangling scheme, since there can be more than one `QApplication::setMainWidget` on the Qt side of the fence, we are saying we want the one that has an **object** as the first and only argument. Here is the key to the mangling taken from `smoke.h`:

```
* The munging works this way:
$ is a plain scalar
# is an object
? is a non-scalar (reference to array or hash, undef)
e.g. QApplication(int &, char **) becomes QApplication$?
```

I am not yet completely clear on how this is enough to do all the work (for example, what happens if you have two methods that take **different objects** as only argument?) but it's what I saw :-)

The last argument, args is of Smoke::Stack type, and it's the tricky one, at least for me.

Here's how it's used in the previous example, QApplication::setMainWindow

```
// qapp->setMainWindow(1)
Smoke::Index method = getMethod(smoke, "QApplication", "setMainWindow#");
Smoke::StackItem args[2];
smokeCast(smoke, method, args, 1, 1, "QLabel");
smokeCastThis(smoke, method, args, qapp, "QApplication");
callMethod(smoke, args[0].s_class, method, args);
```

A Smoke::Stack is a way to pass the arguments to be used with the method getMethod gave us.

We first create an array of 2 StackItems:

```
Smoke::StackItem args[2];
```

Then we assign a value to the second of them:

```
smokeCast(smoke, method, args, 1, 1, "QLabel");
```

Here l is a pointer to a QLabel. (Ok, it is really declared as a void* because, remember, we are **not** using the Qt API, so we have no clue what a QLabel is ;-) and what we are doing is storing in args[1] a casted version of l.

The exact details of why you have to pass smoke and method are not that important, and they seem pretty involved, so I won't try to go there, at least not yet. This has to be done for each argument for the method.

Then we have a similar, yet different line:

```
smokeCastThis(smoke, method, args, qapp, "QApplication");
```

This puts the qapp void * in args[0], casted to QApplication. There are tricky C++ reasons why this is done slightly different here than on smokeCast, which I am not 100% sure I get right, so I will keep quiet ;-)

This special case is only for the object to which the method belongs (the *this* object).

Here is the code for smokeCast and smokeCastThis

```
// cast argument pointer to the correct type for the specified method argument
// args[i].s_class = (void*) (typeof(args[i])) (className*)obj
void smokeCast(Smoke *smoke, Smoke::Index method, Smoke::Stack args, Smoke::Index i) {
    // cast obj from className to the desired type of args[i]
    Smoke::Index arg = smoke->argumentList[
        smoke->methods[method].args + i - 1
    ];
};
```

```

        // cast(obj, from_type, to_type)
        args[i].s_class = smoke->cast(obj, smoke->idClass(className), smoke
    }

    // cast obj to the required type of this, which, dur to multiple-inheritance
    // from the one returned by new. Puts the pointer in args[0].s_class, even
    void smokeCastThis(Smoke *smoke, Smoke::Index method, Smoke::Stack args, vo
        args[0].s_class = smoke->cast(obj, smoke->idClass(className), smoke
    }

```

But where did we get `l` or `qapp`? You can use these same mechanisms to create an object:

```

void *qapp;
{
    // new QApplication(argc, argv)
    Smoke::Index method = getMethod(smoke, "QApplication", "QApplication$?")
    Smoke::StackItem args[3];
    args[1].s_voidp = (void*)&argc;
    args[2].s_voidp = (void*)&argv;
    callMethod(smoke, 0, method, args);

    qapp = args[0].s_class;
}

```

You get `QApplication::QApplication(scalar,undef)` which should hopefully map to `QApplication::QApplication(argc,argv)`. You create a `Smoke::Stack` of 3 items. The first is unset because this is a constructor, so it has no *this* yet, and the other two are `argc` and `argv`.

Then you call it through `callMethod`, and you get the resulting object via `args[0].s_class`.

Later you repeat this sort of operation for every method call you want, and you got yourself an application.

The binding side of things

So, how do you use this to bind your language to Qt?

Well, you will have to create an object in your language called, for example, `QApplication`, and map the “missing method” mechanism (your scripting language probably has one) which is used when a method is undefined so that somehow it finds the right `Smoke` method (mangling the name correctly should be the hard part), then create the `Smoke::Stack` with the passed arguments, call, get the result, wrap it in some language construct you can use of the side of your language, and pass it back.

It looks involved, and I am sure it’s tricky, but at least it only has to be done **once** unlike on traditional bindings, where you had to do it for every class and for every method.

The traditional solution was to automatically generate the code for such wrapping (like `SWIG` does). I think `Smoke` is less error prone.

If I keep on understanding things, there may be a second part of this article, explaining the `SmokeBinding` class, and perhaps a short example of how to bind Qt into a language (`Io` is a strong candidate).

The regularity of Io's syntax is probably going to make the binding simpler than most.

And no, I am not trying to take Aaron's spot as most frequent poster ;-)

I have known about Carlo Strozzi's [NoSQL](#) for about 6 years. I don't think many others do (it's google score is 39500, compared with 81200000 for MySQL) , and I think it's a shame, because it's a much more interesting idea.

What is it? It's a relational database.

Sure, I can almost hear you think [1] , "like Mysql!" or "like Postgres!" and you are **wrong** because it's very different.

NoSQL is a relational database implemented in [AWK](#) and operated via the unix shell. Yes, NoSQL is the most unixy RDBMS in the world. So all of you so-called unix lovers, shell freaks, you are gonna love this.

Installing it is perhaps a bit harder than it should (who knew there are no MAWK RPMs anymore!), but it seems to be packed for Debian at least ;-)

You can learn to use NoSQL with a nice [2] tutorial that was [published](#) on Linux Journal.

The tables are plain text files, and later you can do things like this (from the docs):

```
% column Item Cost Value < inventory |
row 'Cost > 50' | mean -l Value
Item    Cost    Value
-----
3       80     400
6       147    13083
7       175     875
-----
                        4786
```

Isn't that cool? What it does is take the inventory table, select over columns Item, Cost, Value the rows with Cost > 50, then calculate the mean of the Value column :-)

It even supports reports, joins, a bazillion things.

Honestly, I am not sure I can find a practical use for it [3], but it is a great piece of software, written by a dedicated guy, it is extremely original, and it does work. I'd even say it's pretty powerful, and you must accept, it's unixy as all hell.

So, Carlo Strozzi, here's a toast for you and NoSQL. You are cool.

[1] So good is my hearing

[2] Sadly it's quite old

[3] I did once, but that was when I was a real programmer ;-)

I always liked SuSE's Linux distros. They even used to mail me a box every now and then when there was code of mine in it. It always seemed nicely done, and well integrated. Of course I only used it as a workstation.

Recently, I had the displeasure of installing a SLES9-based server for a client.

I say displeasure, because it was by far the worse experience I had with any Linux distribution ever. It was worse than the time I had to install openldap on a P2 with gentoo that was connected to the internet over a 56k dialup and had only 100MB of free disk space.

And that one was a screamer!

What were the problems?

Well, for starters, it was a punishing throwback to the times of proprietary software installation.

I install it. Ok.

I try to update it. Not Ok.

The problem? You need to authenticate to the servers in order to get the updates. And there is no user/password anywhere in the box. You have a serial number (in a word file inside a CD, not on a sticker), and no explanation on how to go from one to the other.

It turns out you have to call Support to get the auth data, and that depending on how you do it you get data that lets you access to the novell servers or the suse servers (and not the other).

Then, after we got that (48 hours on tech support), I start installing the software I need.

The mission of this server is simple. It's a mail forwarding server. It handles outgoing mail, and it stores incoming mail for a few minutes until a CRM software (in windows) grabs it via POP3.

I can do such a server on CentOS, Debian, (hell, yes, even gentoo) in about 2 hours without counting install+updates, including migration of old data.

I install postfix and imapd (I think it's wu-imapd, which sucks, BTW, but the alternative was cyrus, which was gross overkill).

It seems to work. But the CRM can't fetch the mail. Outlook can, though. What the hell?

Well, SuSE decided to disable plain logins for POP and IMAP over non-SSL connections. And there is no way to enable it.

Since that's the only kind of connection the CRM will do, it will not work.

Mind you, in this case, it is absolutely no security risk whatsoever, since the mail server and the CRM are segregated from the user's network...

Ok, I will install courier-imap, which is better anyway. But it's not on SLES9 CDs, and the RPMs on the web are for every other SuSE and not SLES9. So I had to build it myself.

That is, of course, because there is no free repositories for SLES9. You have what comes with it, or what you can build yourself. Anything else, you are SOL.

The same thing happened for almost everything I wanted to install. Either it was not there, or it was for some other SuSE, so it was time to compile a RPM again.

It was like Gentoo, only without the automatical dependencies, and with no hope for future security updates unless I build them myself.

At that point I was already telling the customer that maybe I could just install OpenSUSE, which was free and would not have these problems (hell, I would even get apt4suse and avoid the damn novell servers).

Of course that means they would be a few hundred dollars poorer for no good reason.

But anyway, it took me roughly 3 extra days to set this up, which made me actually lose money on the gig. ANd I lost the time in the most pathetic way, sitting in a customer's office waiting for tech support, watching my money go away.

That had **never** happened to me before. I must say I am pretty disappointed.

But what was the root of the problems here?

- There is no free SLES clone like CentOS

If there were, then there would be 3rd party repos. The customer would still have bought SLES9 because they are support groupies, but my life would have been easier.

Of course, it would probably cut into SLES sales, but hey, that is not my problem, is it?

- Novell Argentina tech support sucks for Linux.

The guy on the phone literally had no idea what I talked about when I asked about how to get into YOU to update the box.

But don't worry Novell, I heard Red Hat Argentina's is quite bad too.

If you really want SuSE, buy a regular one, the ones with public FTP repositories, and avoid trouble. Or get OpenSUSE.

Me, I'm pretty bummed :-)

A long time ago, there was no Internet.

Ok, there was an internet, but I lived outside of it. It was 1992 or 1993, and I only saw my first webpage and send my first email in 1995. And I was perhaps the third person to have an account on a permanently internet-connected box in a 150 km radius.

But that didn't mean I had no access to internet stuff! What I did was buy CDs containing mirrors of repositories like Simtel.net (it was a single CD, too!) and in there you could find hundreds of programs.

Most of them shareware, most of them crap, but every once in a while, there was something cool, like DJGPP (a whole gcc suite for DOS! A real C compiler! For Free!)

At the time, I had a side job writing data entry software for statistics students. They were simple programs that showed a form, where data was loaded, then it did some simple manipulations of the data.

The natural language for that was something like Clipper or DBase, but I didn't have access to them (or a way to learn it. Remember, no Internet).

On one of those Simtel CDs I found Jorf. (Josephine's Recipe Filer). It was a OO language, with an interpreter for DOS or Windows, and it supported stuff that was **really** advanced for the time, and it made my coding a lot simpler.

Out of nostalgia, I downloaded a copy (yes, it is still there), and ran it in DosBOX (yes, it still works), to check if it was as good as I remembered.

You know what? It is.

In fact, if it had come out 2 or three years later, and as free software instead of shareware... I think it would have been big.

Here are some highlights of the language:

- OOP
- Has integrated windowing toolkit (for DOS and Windows)
- It had an interactive hypertext/windowing tutorial written in itself. In 1993.
- It looks like a cousin of Python. A freaky cousing, though.
 1. Comments start with |
 2. Strings limited with single or double quotes
 3. Automatic type conversions
 4. Intentionation controls flow :-)
 5. No declared data types
 6. Integrated editor and debugger

Sample Hello World:

```
Demo:Start
Msg:Add ("Quick Demonstration", "Ok")
    Sure you can say "Hello World" in one line of
    C code. But how many punctuation characters
    are required to display a dialog box like this?
Return (Ok)
```

That piece of code showed a window with the message in it, and a Ok button.

The funky thing is: in the tutorial, you saw the integrated editor open, and the text of the example start to appear, and then it ran.

That looked like magic at the time :-)

The toolkit supported radio buttons, checkboxes, text entries, all the basics, and it was a thousand times easier than what Turbo Pascal or Turbo C guys battled with at the time.

The author was Wayland Bruns. He lived, in 1993, in Colton, Oregon.

He later seems to have become CTO of a company that designs sync software for Goldmine, Lotus and other such things.

So, he became a suit ;-). However, he was once a guy that wrote, in his software's manual, things like:

```
JORF Company is just me, Wayland Bruns. I have been working on JORF
for six years, and ran out of money three years ago.
```

Or:

```
JORF(R) is a new computer language. JORF was created by a Grunt-
programmer frustrated by low level math based computer languages that
are inappropriate for business data processing.
```

And you know what? It was the right idea. If he started Jorf in 1987, that means he started it around the same time Perl 1.0, (and the syntax is much nicer ;-). He started it around the same time Guido started Python.

Here's a toast to JORF, which could have been Perl, or Python, or Ruby. But was not.

Last saturday I went to see King Kong with Rosario, and something happened I never saw before.

For whatever reason, we went at the 1:25AM session. I don't think I had ever been to one so late.

And then it lasted 3 hours. And it was almost the longest day of the year.

So when we left, at 4:30 AM, walking through a ghostly mall, it was dawn.

It's a small thing, but it was quite shocking :-)

The movie... she didn't like it and insists Naomi Watts is wearing, in one of the scenes, a Lurex dress, which couldn't possibly be the case in the 30s.

Me, I liked it quite a bit, perhaps my suspension of disbelief is not so easily taxed by textile issues, except for some serious moral trouble I got a day or so later.

You see, killing Kong on the Empire State building was **right**.

That damn beast had just stomped, thrown, smashed, chewn and swatted about 2500 people.

He was going all 9/11 on Manhattan, and just because he didn't feel like crushing one specific blonde (although he sure killed all her predecessors on the sacrificial girl job), we are supposed to feel sorry for him?

Cry me a river of giant alligator tears, I am not. I say we should bazooka the evil man-eating monkey, and put Denham in jail for reckless endangerment, along with all his accomplices.

AND he should lose his shirt (along with his theater bankrollers) in a civil suit to the family of the maori guy whose head got chewed by Kong, or Lumpy, the cook eaten by gigantic man-eating maggots.

At least in the original movie, the girl has the good sense to be scared senseless by the sight of a giant gorilla with romantic leanings.

Here? She laughs while they ice skate, I assume ignoring the blood stains all over the monkey's fur.

I have been teaching for almost my whole adult life.

My first real job was teaching assistant. I worked in a university for 10 years.

Then I started teaching Linux training courses for diverse companies (think LPI kinda stuff).

Then I stopped. Why? Because it makes no sense economically to teach in most cases.

The following is written in pesos, but the idea is probably about the same for other countries.

There are no money signs because cheetah templates hate them ;-)

A trainer is paid about 35 per hour. If he's pretty good, he gets 50.

A course is about 24 hours, so he gets about 1200 (I am going for the best case scenario here).

Usually he has between 6 and 12 students, which are charged about 900 + taxes, so the gross is average 5500.

Of that, half (or a little less) goes to the classroom rent, leaving about 3000 for the training company.

Pay the teacher, and you have a rather pathetic amount for the training company that pays overhead, salespeople, taxes and whatever.

So, who makes any money out of this? The classroom renters :-P

So, if you are going to work on training, please don't rent classrooms, it makes no sense.

Now suppose you have a consulting firm and you can do onsite training (at your clients), and just pocket the money.

If you have just three students, you need no facilities, you can charge slightly higher, because it's onsite.

It's the same effort for the trainer, because you just change where you commute to (training center vs client).

Since you can make more money with smaller classes, you can still charge a little higher (more personalized attention).

Since the students are all from one company they have more in common and you can structure the teaching better, which makes it way less boring.

So, if you want to make a living teaching, here's how.

- Market directly to companies, for onsite training.
- Have your own materials.
- Be flexible.
- Buy your own classroom if you have to.

And most of all, don't worry. It's pretty much impossible to **lose** money training, unless you open a exclusively training company.

Saw Resident Evil last night on HBO... it ends exactly the same way 28 days later begins :-)

Not to mention that both are cannibal zombie movies. Amazing coincidence.

Of course Milla Jovovich <> that english bloke.

[Robin](#) (only works in Firefox) via [reddit](#)

There is a python contest at <http://www.pycontest.net/>

The task is writing the shortest program to drive a seven-segment LCD thingy.

I have no hope of winning, but here's a helpful hint:

If your code is any longer than this (191 chars), it will not win ;-)

Note: I edited this item way too many times already ;-)

And yes, I can save two characters moving the return up.

A much uglier, yet much shorter (151) version:

And yes, that pretty much proves you can write ugly python. And I am giving up.

A shorter version probably involves a different algorithm, and I can't find any.

I am particularly proud of saving one character by writing 104004334054154302114514332064 as `int('a302ho6nqyp9vxxvpeow',36)`.

Also interesting is that the number I was using there originally started with 4 and was exactly the same length written both ways ;-)

Since that number is pretty arbitrary (it's an index table into the "graphics" array), I just shuffled the 1 and the 4. The 0 would have been better but then it didn't work, of course :-)

About <http://www.pycontest.net>... here's how it's done.

Sadly, it's pretty much impossible to put the code on the web because it has some **really** non-ascii stuff in it ;-)

Here's the key tricks (thanks to Remi and the others):

One of the key tricks is writing a short version of a very long octal number (or three).

I tried using base36 and `int('gobledygook',36)`.

You could use `0xhexacodehere`.

But here's the most space-efficient one: use base 256!

For each byte you want, create a char using `chr(byte)`. Then paste them all together. Then put it in your code as a string. If you are lucky you will have no quotes or newlines in it, and the python interpreter will eat it.

You can later get the byte number x using `ord('h!alsdf'[x])`, and each byte by some dividing and modulo operation.

Another important piece is encoding 7 3-character strings as compactly as possible, and splitting them by offsets (which are encoded in the previously mentioned big number in base 8 ;-)

One extra squeeze is finding the shortest way to concatenate strings.

It's `"".join()`, but if you are using it twice (or more), you save space by doing `j="".join` and using `j` later.

Last one: Instead of defining a function, do a lambda:

```
def seven_seg(x): return .....
```

```
seven_seg=lambda x:
```

```
6 chars less!
```

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

And [here](#) is the shortest code I got so far (121 characters, requires python 2.4. On 2.2, it has to be 124).

In the contest, some guys are at 120. I don't know how. I can't even guess how ;-)

Update: it turns out I *did* know how to do it check it [out](#) I think it's the first *public* 120 :-)

BTW: you can't write that using kwrite. Vim works, though.

system-message

WARNING/2 in `everything.txt`, line 5003
Duplicate explicit target name: "here".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

I just found [here](#) the announcement of the first free software I published (at least, that I recall), from may 13, 1996. So, It's going to be 10 years in 5 months!

Killer quote:

```
Requires:  
Python 1.3 (Maybe 1.2 would work too)  
XForms 0.80 (NOT 0.75)
```

system-message

WARNING/2 in `everything.txt`, line 5018
Duplicate explicit target name: "here".

Saw Narnia the other day.

Liked it.

Of course, the whole story makes no sense, but hey, that's how it's supposed to be.

If you have not seen it, and have not read the book, and intend to ignore the plot, please stop here. Ok?

There is this large lion, Aslan.

He gives his own life to save that of a sniveling treacherous, silly kid who sells others for turkish delight (a candy Rosario tried in Istanbul, and tells me is pretty good, so there is some sense in him).

Ok, so the kid is actually just scared, and petty, and a kid, and he's not that bad.

On the other hand, the lion...

For the whole movie, the White Witch is built up as evil, and monstrous and a killer. But it turns out that the worse she does is **freeze** her victims. Unharmd.

She eventually kills someone in a battle (but so does Peter, our 14-year old "hero").

The frozen guys can be revived at will by Aslan. Who for some reason had not done so.

On the other hand, Aslan's big sacrifice? He knew all along that he would be unharmd. Even his mane regrows after a few hours.

At least he could have told the two poor girls who thought had seen him die. And skipped the whole heavy-hearted “oh, I am **so** sacrificing myself” walk through the woods.

Oh, you may say, but the war is fought so that the true rulers of Narnia will ascend to their thrones!

Well, how in **hell** are those four english kids the true rulers? They had never been there, they have no connection to anyone there!

Hell they are the only four bloody humans in the world!

It looks amazingly racist to me. Species-ist?

So, hundreds of Narnia inhabitants die (and I mean **really** die, not fake-die like that Aslan kitten) so some carpetbaggers get to lord as kings over the plebeian masses, instead of **another** high-born chick.

My suggestion to the hordes of gryphons, sphinxes, polar bears, fauns, centaurs and dwarves:

Kill them all, and start living a decent life, without supporting useless parasites.

Free Phillip!

PS: Yeah, I did like it ;-)

I will make some of my work public.

The best candidate is one that will probably not appeal to anyone: my personal linux distro.

I have it, I use it all the time. It’s not published yet, though, since it consists of a base CentOS + a lot of work.

So, I will try to make it systematic, call it an installer, and drop it on the unsuspecting audience.

Don’t hold your breath, though.

Here are some of the features:

- Server oriented. I use it for my clients’ servers.
- Qmail+courier+vpopmail+roundcube+spamassassin+clamav mail system.
- Squid proxy+fwbuilder firewall
- Smart for package management
- runit-based boot/services.
- KDE as a GUI. Accessible via FreeNX/secure VNC.
- OpenVPN for simple VPN management.
- Custom cherry-py-based admin tools
- No SELinux (sorry, it’s a pain in the butt. Secure, but a pain).
- Bacula for backups
- Custom tool to back the whole disk to a bootable DVD (System Rescue DVD)
- /etc in SVN+Trac (yes, really, and the admin tools force you to use it)
- Probably djbdns for DNS.

All in all, it's a pretty ordinary CentOS-based thing, except that the switch to `runit` makes service management way simpler and regular (you can do things like having a non-root user that can manage *some* services, yay!)

On the other hand, the cherry-py-based tools would probably need a bit of a rewrite, since some are pretty cruddy.

Mostly, I have been gathering these pieces over the last 6 years as I really don't like the way any of the Linux server distros are built.

I based it on CentOS because doing the whole distro by hand is way too much work.

I can probably show it around June, if I don't get too sidetracked. Hopefully, someone will read this list and tell me "but graxst linux already does all that!" ;-)

- Date? Feb. 18th. 2006
- Where? San Isidro, Buenos Aires, Argentina.
- Who? Rosario Guerrero + Roberto Alsina
- Why? Because we love each other.

Look at this [picture](#).

Yes, it's what it looks like. It's part of our wedding invitations. How nerdy is that, exactly?

Here are [runit RPM](#) and [SRPM](#) for CentOS 4.2 (probably works on some other distros as well).

You install it, then clone your GRUB (or LILO) entry so you boot passing `init=/sbin/runit-init` to the kernel and when you boot it, it should start on a rather normal-looking runlevel 3.

To reboot or halt, use `reboot-runit` and `halt-runit`, please!

Keep in mind that this is just a beginning, since all your services (except your `ttys` ;-) are still running without supervision!

Should not destroy anything, but use at your own risk.

You **will** have to work further if you want to do more.

Included is `/etc/runit/services/test` a trivial template service.

Sorry about the spammy hosting, but I will get a decent server space for these files somewhere eventually. These rpms are just a teaser, anyway :-)

Rosario and I are pretty much non-believers, so we are getting married only with a civil ceremony (which is mandatory anyway).

So, this morning we went to the Registro Civil to arrange the appointment for the wedding.

Everything was done pretty quickly, no fuss, good vibes from the lady who's marrying us, we were pretty happy. But one thing called our attention.

We had to wait for about 30 minutes to set the appointment because we were the third couple.

There were we (dressed... perhaps too casually. This is just bureaucracy and it was early in the morning ;-)

Another couple of about 35/40 year-olds dressed very nicely, good clothes, nice shoes (we **do** live in a pretty fancy neighborhood).

And the last one were in their twenties, good looking people.

Now, the three couples were performing a necessary and pretty critical step in getting married, it's not romantic at all, but after you put your fingerprint in the request, you are saying that you are **really** going to get married.

And anyway, we were all asking for dates in the next month, so we were all pretty close to getting married.

And you know what, I was there, holding Rosario's hand, we were talking about how much we wanted to finally get all this done and be a "legal" couple, hugging and kissing (nothing scandalous ;-)... the other two couples hardly talked to each other.

The older ones, I think the guy only said "this is taking a while" and she answered with a nod.

What the hell? I was thinking about starting to scream at them **you are all going to be extremely unhappy!!! do not get married if you don't like each other!!!***

Then again, perhaps they had made the wrong line and were waiting for death certificates?

I mean, why the heck **were** they getting married?

Thoughts about platforms brought by a History Channel documentary on tools. Hope it's interesting :-)

I saw Penn & Teller's show about ESP last night. Besides being hilarious, a throw-away comment (by Penn, of course) made me think.

Here's what he said (give or take a word):

Telepathy is communicating your thoughts directly to someone else's brain.

Because it seems talking is too much work.

And it got me thinking... why would anyone want to be a telepath? What's the practical purpose of telepathy?

In this anecdote I see one of the two problems with telepathy. The fact that it's not necessary.

It's supposed to be a mechanism for communication. Well, as Penn said, you can talk, so there *are* alternatives. There must be a compelling reason why telepathy would be better. Let's consider some parameters of communication quality!

- Reliability of connection

I don't expect anyone will tell me telepathy is more reliable than a cell phone. At least not in the forms ESP research has any hope of finding.

If there were a really reliable telepath, he would have been found already. Unless he knows he's been sought, and uses his telepathy to hide himself ;-)

- Range

Well... let's look at the alternative! Satellite phones. They work (worked?) anywhere on earth. Surely telepathy's advantage over that would be marginal? And Iridium went bankrupt, so there was not much demand for that, either.

- Reliability of signal

By this I mean the odds of getting what the other guy is sending.

I suppose if telepaths transmit the thought in its raw form, interpreting it on the other side is going to be quite hard.

Take me, for example. As a side effect of how I learnt English, almost every conscious thought I have is in my head in Spanish and in English. At the same time.

I literally have English dubbing running fulltime in my head. That's gotta be annoying!

Even without that, it takes *me* some effort to understand what I'm thinking. And I really know me!

Please suggest other parameters in the comments if you want!

Now, I have a strong suspicion about why people want to believe in telepathy, and I think they fall in two camps.

- Those who want to read what you think without you knowing it (or intentionally sharing it).

This is some sort of puerile voyeuristic fantasy, like dreaming of being invisible (you know why they want that, right? ;-)

Probably the reason why the US government used to throw away money paying ESP researchers.

- The pathologically shy

They want you to really understand what they think, so you will understand and like and love them.

Well, that's all find and dandy, but it's perhaps easier to actually express yourself instead of dreaming about having a ESP capability you will never have?

I feel some tenderness for these maladjusted guys, mostly because I used to be one, and I still am in some ways.

But I have learned a few things since then. No, they will not like you if they understand you. Or maybe yes. But it's not guaranteed. So, be a nicer guy instead of dreaming of telepathy. Start understanding them, even!

Warning, silly pseudo-buddhist anecdote:

The Buddha saw one of his disciples cross a river walking over the water. He congratulated him, then gave the disciple a coin and told him: use this to pay the ferryman next time.

A miracle that only does the same you get for a coin is a miracle that's worth a coin.

Telepathy is a miracle worth 20 bucks a month for 500 minutes or whatever your cell plan is.

In the previous article, someone suggested cheating on exams as an interesting application of telepathy.

Which reminded me of my first interesting hack.

I was in college, back when computer time was allocated by the hour, and I had no computer, and I was about to take my final exam on linear programming.

For those who are not familiar with the [term](#), linear programming is not really about programming. It's about solving a specific kind of optimization problems.

And for those who don't know that, don't sweat it, it reduced, in real 1989 life, to applying a program called LINDO to find a local or global min or max for a function.

Now, we were a poor college, so there were like 10 computers. For 5000 students. And we, in that subject, were not allowed to use it.

And it didn't have linear programming software in it anyway. And it had no compilers or interpreters (except qbasic).

So, we did LINDO by hand. On paper (the Simplex method).

And it was **boring**. But we did it. It is pretty simple once you get the hang of it. But you have to make hundreds of calculations, so we were allowed calculators.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "this".

And I had [this](#) baby.

I had bought it with my second paycheck as a teacher assistant. It costed me about the same as a small bike.

Or a TV.

And it had 4KB of ram, an ascii keyboard, and 116 preloaded programs for common scientific calculation.

It was the best calculator I ever had :-)

And it was programmable in BASIC.

So, the night before the exam, as I did a sample drill, I decided to implement a solver.

But since we had to solve things on paper, I had to show intermediate steps.

So, instead of a program to solve the problem, I wrote a program that solved step-by-step as done by hand. Which was about 20x harder.

And it did fit and run in the 4KB of ram, and it displayed the intermediate results on the 2x32 char screen.

Sadly, there was no way to take a program out of it, so it was lost on the next battery change.

But hey, I think that was nice :-)

system-message

WARNING/2 in `everything.txt`, line 5301

Duplicate explicit target name: `"this"`.

Inspired by "From Dusk Till Dawn".

Story Nr. 1: Night of the predator

Genre: Supernatural Horror.

Notes: This story should be written in Lovecraftian prose, and try to provide a sense of foreboding, and imminent doom, while seeming completely obvious.

Francis was the seventh son of a seventh son of a seventh son. As such, his fate was preordained. On every full-mooned friday, he would experiment a horrid transformation.

He runs through the woods while remembering the strange admonitions from his father, who explained to him the bloodthirst of the werewolf, the horror of his actions, the curse upon his victims.

He feels a nagging sense of things being all wrong. A hunger for strange, unnamed things. An eagerness for forward motion. A predatorial wish.

Then, while in the woods, looking at the moon, he experiences a painful elongation of his body, a constriction of his limbs.

His skin changes quickly, his teeth grow too long and sharp for his mouth.

He flops around for a minute or two and dies.

The seventh son of a werewolf is a wereshark.

The rest of the story is a CSI-style police procedural about the origin of a shark corpse 500km away from the sea, the protagonists are park rangers.

Story Nr. 2: The man with the golden brain

Genre: bondian superspy.

Notes: This should include a lot of technojargon which makes no sense, and the prose should be quite bad. Think Ian Fleming.

Our superspyprotagonist is tasked with destroying the world-destroying weapon of a thirld world dictator with a surprisingly large moustache and a beret.

This dictator's headquarters are beneath a volcano, and his weapon would destroy the world by provoking catastrophes of a very complicated nature (to be determined. ideas: drive all farm animals into a killing frenzy, make all 7up bottles explode at random times, turn cocker spaniels into evil scientists).

Our hero enters the lair through some complicated path involving sewers and airducts.

He is captured by the evil tyrant, and tied to a table with a giant laser aimed at his groin.

While the tyrant prepares to kill him in this over-complicated manner, the hero says "before I die, can I ask you one thing?".

The tyrant replies by blowing hero's head with a colt 45 then says "Hell, no!" [1].

Unimpeded in his plans because all industrialized nations relied on a single guy working alone, he blackmails all the world into surrender.

The rest of the movie is a geopolitical thriller about:

- The difficulty of ruling the world from an underdeveloped country, with emphasis on telecommunication issues.
- The ecological situation stemming from the forced undevelopment of Western Europe and North America in order to reduce the stress on the environment.
- The psychological stresses on the world population when they realize their lives depend on the whim of the guy with the bigger gun.
- The compassionate rule of the aforementioned tyrant and benevolent dictator for life, who proceeds to disarm all armies (including his own), and never actually uses his weapon.

Epilogue:

200 years later, the impossibility of remotely controlled explosive 7up bottles is terminantly proven.

[1] This is taken from the "Guie for the Perfect Tyrant".

I have an idea that can kill the most frequent complain about python.

BPython.

BPython is a simple wrapper around pyhton which processes a .bpy file, produces a .py file, then compiles it into a .pyc.

What does it do? It uses braces to control flow.

Since braces are actual python syntax, you will have to use #{ and #}

As an added bonus, if you are careful, a bpython program will also be a valid python program.

Of course it has issues (like module loading) but those can be worked around.

The implementation should not be more than 30 lines of python. Or bpython ;-)

A new story in the "Fighting Spam with Qmail" series, after an over two year hiatus.

This one is about RBLs, qmail and a great tool called qmail-spp.

In January, I [suggested](#) it would be trivial to write a preprocessor that would accept a version of python which delimited blocks with braces instead of indentation.

Ok, almost, I suggested #{ and #} as delimiters.

Well, I had a few minutes free today, and here it is, a functional BPython->Python compiler, so to speak.

Here's the kind of imput it takes (notice how it's not indented at all:

```

def factorial(n):
#{
if n==1:
#{
return 1
#}
else:
#{
return n*factorial(n-1)
#}
#}

for x in range(1,10):
#{
print x,"!=",factorial(x)
#}

```

And it produces this (I am not happy with the indenting of the comments, but who cares):

As you can see, this is both a legal Python and a legal BPython program ;-)
It has some problems, like not working when you use a line like this:

```
#{ x=1
```

But hey, it *is* python with braces.

Here's the code. I predicted 30 lines. It's 34. And it's 99% a ripoff of Ka Ping Yee's regurgitate.py which you can find all around the web.

So, all of you who dislike python because of the lack of braces and the significant whitespace:

BPython has no significant whitespace, and braces are mandatory.
Enjoy coding!

1. PyDS is good enough to let you change how you host your site and keep it just working.

I switched from pycs.net (thanks for all the good work!) to a static site hosted at my ISP.

Since PyDS simply generates a whole lot of static HTML, this worked pretty good!

I generate the site, and then upload using lftp. No fuss, again, it just works.

1. Static hosting sucks.

On pycs.net I had referrer logs, stats, comments. The static template from PyDS doesn't have any of that, of course.

1. You can work around the suckage.

No comments? Haloscan can host them for you.

No stats? statcounter can do them for you.

No spam-free mail webform? I haven't looked yet, but I bet it exists someplace :-)

1. You learn stuff.

I now understand how PyDS's macros/templates/nuggets/uploadings work. I thought I did. Now I do.

For example, I like comments for longer articles. Before, I added the links by hand. Now I just hacked the templates and it's done automatically. I could have saved a few hours of my life if I had bothered to learn it.

1. Switching your URL sucks.

Since noone knows where the hell my page is. Luckily soon cleo will switch me at planetkde.org and things will start to flow again.

1. Depending on free hosting's tech support is not a good idea.

I love pycs.net. I loved how it worked. Until it stopped working.

I have no right to whine, though, so I offered to help (got no response yet). Then I moved out.

Since moving is annoying, I will probably not be going back, even if I start loving it again :-)

I am getting and reposting all the comments, but then gain haloscan has a 4-month limit on the comments, so I will have to implement some sort of comment-archiving and closing mechanism.

After a few agitated months... which I don't intend to explain here, I now am in position to post some new content in Lateral Opinion.

Hopefully, this time I will be more persistent. But I wouldn't hold my breath, if I were you :-)

On other news, I will be 2**5 years old tomorrow.

I saw an article on the newspaper about an event called Letras Latinas, a sort of typographic convention, and it said that there would be about 300 fonts displayed, and that the public would be able to download them.

Guess what, you already can! (you have to dig around a little, though) In fact, there's several hundred fonts, if you bother going to the author's websites.

In the past, I obtained rights to distribute some 400 fonts just by asking the authors. These lack license information, but I may do it again, if I have the time.

Some are pretty good. For example, Escritura Px is a nice bitmap font, and some of the "text fonts" are really not bad (brasilia, for example).

One of my greatest frustrations as an adequate programmer is that I think of things I believe should exist, yet I am not able to implement them myself.

Today I will mention one (or two) of them.

There should be a Xen-based distro.

[Xen](#) is a very good virtualization package, which gives you multiple simultaneous virtual linux installations with very little overhead.

Sadly, to make it work, right now, you have to get a linux working, then install Xen, then install one or more extra linuxes as xen machines.

Which is not terribly hard, but could be done much easier.

I am thinking of something like this:

Installing Xen-Linux installs a very simple basic setup as the supervisor partition, sets up Xen properly, and sets you with a single Xen virtual machine.

Also, in another part of the disk, it has a gzipped file with a simple, basic linux setup which you use as a template for further virtual machine initializations.

So, whenever you want a new machine, you run a script, it sets it up, starts it, you get something like firstboot asking you the usual network/language/whatever questions.

Then, you are dropped into a nice package selection tool, where you choose what you want installed in this virtual instance.

And that's it.

This would encourage the administrator to **always** set up his servers as virtual machines, which except in cases of real hardcore performance requirements is a good idea (you can implement HA as full-image fallback. You can migrate to a new box in minutes!)

I could do most of this. However, that leads me to another thing that should exist...

A simple installer

Wait, you may say "Linux installers are simple already!!!" and you are right.

I mean a simple installer for a guy trying to make his own distro!

I actually like anaconda a lot. But I would like it even better if I could just get some sort of tool that, given a list of packages, would create an anaconda-based CD with customizable install scripts.

And that is the piece I am not sure I can implement.

So, if anyone knows of any such thing, I would love to hear about it.

And as a preemptive message: No, I don't want to do it with Gentoo, or Debian, or Ubuntu, or Suse, or kickstart. I want to create a RPM-based, CentOS-based, anaconda-based installer that works just like a regular CentOS/RHEL installer CD.

What I know after one day of hacking around anaconda, the CentOS/Fedora/RHEL installer...

...Don't hack anaconda!

It's pretty evil stuff. It "builds" with dietlibc (!) in the CentOS 4 version.

The latest doesn't but it's unbuildable on anything other than FC5.

Oh, but that "builds" is really not quite true. It doesn't build at all. I don't know how they compiled it in the first place.

On the other hand, if you switch it to plain glibc, it crashes (no warning, no error) when running loader.

And how can you debug it? Well, you can't really, because there seems to be no way to run anaconda unless you are installing.

Why? Why not create *some* way to run it on a test setup?

There are no docs I could find on the internals, there is not even a damn explanation of what binary does what, so I have to track it down theoretically by mounting and examining the .img files!

Sure, I can probably hack it... but why?

Personally, I think it's better to wait until someone writes a sane installer. Then again, noone has in 12 years :-(

Dudes, here's what the installer should do:

- 1) Get you in a GUI, or text mode. It doesn't matter.
- 2) Partition disks
- 3) Install basic system (about 350MB, tops!)
- 4) Install bootloader
- 5) Boot into said system

6) Configure network

7) Start package management tool

Knoppix already had all the hw autodetect figured years ago. Why are we not using it everywhere?

I am just not the guy who can do it. I don't have the stamina :-P

After a **very** long time, I have decided to make one of my projects public.

It's a collection of qmail plugins, to be used with [qmail-spp](#) for diverse purposes.

It's C code, which I don't quite like, but I think it doesn't suck too much.

I had seen a few blogs that had "digg this!" or "submit to reddit" links, and I decided to add them.

Thanks to [explodingboy](#) that was rather simple, so now all items have a neat box that adds no clutter where you can do it.

Very simple thanks to [PyDS](#) and its template system.

So, RH pays 350 millions for JBoss.

And here are some quotes the guys at the register have [digg](#)ed up:

Our own talks with RH broke down, RH is NOT IN THE BUSINESS OF PAYING OPEN SOURCE DEVELOPERS, we are, that is why we created JBoss inc. RH wanted to keep the services revenues all to themselves. That is the dirty little secret, so for them to come out and claim they are the open source when we know the reality is distasteful.

Honestly? I have had my troubles with RH in the past about their positions on KDE, and everything, but hey... at least I didn't later sell my children to them ;-)

OTOH, Fleury is quite a character. Or a raving [paranoid](#) :

Daddy, why are you still working on a Sunday? Is IBM still after you?

My bet: you will not hear anything from Fleury for a while. Then, in some time (I put the over/ under at october) he will discreetly resign/be fired.

Rading an interview with Bruce Perens has been a real eye opener.

Consider this:

For example, when I considered being an employee one of the things standing in the way was the fact that Mark doesn't give his employees stock in his companies. If I'm going to work for someone I'm going to be a little entrepreneurial about it, so I felt that although Ubuntu and Canonical could do a great deal for Debian and be excellent community members, they were never going to be the core, and we could actually get closer to the core by following what I have set up for UserLinux.

Now I understand! In order to be "the core", a company should give Bruce Perens stock.

Yet another little program.

In this case, it's called rascan, and it's a rewrite of Inter7's simscan.

What does it do? It easily hooks clamav and spamassassin into your qmail-smtpd setup, so you will not accept viruses in your server, and tag spam.

It still lacks some features, like bouncing messages based on spamassassin hits, or blocking attachments, but it has at least one feature simscan lacks (adding footers to messages).

It's still alpha, so I wouldn't put it on a production server, but it may even work ;-)

In the last month or two I have been writing a fair bit of C code.

I really dislike C. But I have not been able to find a language to replace it with, except C++, and that's not much of an improvement for what I did.

Maybe someone can read this and help me. I don't mind learning a new language. Really, it's no problem.

The goal is writing small programs (in the hundreds of lines region), which have the following requirements:

1. The program **MUST** be correct. Can never segfault or crash in some other way.

Since the code is short, I think I have managed, using bstrlib, to get this, after quite a bit of pain, by auditing and unit-testing. But I am not sure, and I will never be.

It would be awesome if the language could be garbage-collected for this reason.

1. It **MUST** have a decent string handling.

The programs handle mail. So, it's strings galore.

1. It **MUST** have very low overhead. The programs should run and end quickly. Very quickly, because people may be waiting for them to end.
2. It **MUST** be a reasonably simple language. I am not a great programmer, and these tools will be hacked at by people with limited programming skills (sysadmins like me!)
3. Trivial access to C libraries.

So, let's see.

C: gives 3, 4 (in a fashion), and 5

C++: 2, 3, maybe 4, and 5

D: It seems to give me all of these, along with a C-like syntax... but I can't get it to work yet (correction, I got dmd to work, so it's looking good for D!).

So, any takers?

Yesterday (or rather, like 10 hours ago?) I posted about my quest for a specific language, and mentioned one of the candidates, called **D** which I had found appealing.

Well, because I am a quick coder (not good, but **quick**) I decided, after looking around for about 5 minutes, to take as my first project giving the D community a UI toolkit.

My first choice would have been Qt, of course, but D doesn't interface well (or at all) with C++, and I was not about to hack a QtC binding, because it would be unusable.

So, I looked around the web for a small multiplatform C toolkit, and I found there are very few of those left!

You have Gtk+ and ... well, almost nothing else! I found **IUP** though, and decided to bind it.

Since binding C libs to D is quite simple, I did it mostly by hand. And 76 files and 8872 lines of code later, here is **DIUP**, a **D** binding for **IUP**.

Now, how do you use it?

Get **IUP**, then get **DC**, then get **D**, then get **DIUP**, then open it, and check the examples.

The binding modules themselves are in the binding folder, so you can compile examples like this:

```
[ralsina@monty iupbinding]$ cd examples/elem/iupbutton/
[ralsina@monty iupbutton]$ dmd -g -I../../binding/ iupbutton.d -L-liup
gcc iupbutton.o -o iupbutton -g -lphobos -lpthread -lm -Xlinker -liup -Xlin
[ralsina@monty iupbutton]$ ls -lh iupbutton
-rwxrwxr-x 1 ralsina ralsina 217K Apr 18 02:30 iupbutton
```

You will probably be linking a few libs too many, but don't worry yet.
The good news:

- IUP works on windows and Unix.
- IUP is an order of magnitude smaller than WxWidgets.
- Most of it seems to be working.

The bad news:

- IUP looks ugly because it's motif on unix.
- IUP is pretty limited.

Someday, when I have a free afternoon, I could write a Qt backend for [IUP](#) and avoid the suckage **plus** porting it to OSX in the process, but...

- IUP is damn hard to build. Luckily they have binaries.

Mandatory [screenshot!](#)

So, enjoy and comment!

Just saw that the XFree86 core group disbanded.

Last I heard, the core group had only three active developers, so it's not such a huge thing, and the message says *this is an acknowledgement that the core team was no longer representative of the active, experienced and skilled XFree86 developers, or a place where technical discussion happens ...*

This probably means that development of XFree86 will go on pretty much as usual.

However, development of XFree86 has always been a strange beast. A open source project that was developed a lot like it wasn't (closed mailing lists for years, getting commit is harder than almost every other project).

Also, in the last year or so, it has been... centrifugal. It has spun out people and subprojects, usually not in the best of terms.

I mean, if you look at their mailing list archive, there seems to have been some serious personality clashes.

Those are often not a terrible problem: developers usually have huge egos, large enough that they can shrug off attacks would make lesser egos shrivel ;-)

But put all together, and it doesn't look like XFree86 is a very healthy developing project. Since it is also a very important one, I hope everyone will chill out, or else, that they will get overheated enough to really make it explode, and that one of the resulting shards will be strong enough.

This is a quick and dirty guide to setting up VPNs using CIPE. A very watered down version of a class I give on the subject.

Hopefully, it will explain how to do that kind of thing, since CIPE is cool :-)

After I hacked the D/IUP binding I thought... howcome I can not build IUP on my PC?

Well, their infrastructure is not auto*, but something they made which I really just don't understand.

So, I decided to hack the build system. if I intend to play with IUP, I should be able to package it!

So, after 30 minutes and some qmake doc reading, I have something I can build.

Oh, sure, I mercilessly hacked off the win32 sources, but that's not my interest ;-)

I will not upload it, but if anyone needs it, just email me.

Of course you know that means nothing, but yes, after a few hours of hacking, I got enough of a Qt backend for [IUP](#) so a window can be displayed.

Of course those are **all** the widgets that are implemented currently.

If by implemented you accept "can't connect callbacks and 90% of the attributes are inaccessible", that is!

But after I get the callbacks working (that's the hairy part), the rest is simple monotonous repetitive work.

It's interesting that the Qt backend is, of course, looking quite simpler than the Motif one, and in fact, a fair bit simpler than the IUP programs look.

But hey, you get a sort of LUA bindings for Qt at the same price...

OTOH, the third step is a nicer OOP, D, wrapper around IUP.

Or something like that ;-)

I have callbacks (some), geometry management, LED files loading (resource files) and it seems no hard parts are left.

That would mean the remaining work is routinary, even if it's a large amount of it.

It's nice that I can switch back and forth between the Motif and Qt implementations just by setting `LD_LIBRARY_PATH` to compare :-)

Ok, here is version 0.0.1 of my Qt4 backend for [IUP](#) . It's not very good, but a bunch of things work. A whole bunch more do not, of course.

I am showing this just because I want to rewrite it, because I noticed that I am writing it like it was Motif still, and I must understand this is actually C++ and Qt code ;-)

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "diup".

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "d".

Using it along with [DIUP](#) you can write silly Linux/Windows/Mac GUIs from [D](#), or you can use C. No LUA yet.

system-message

WARNING/2 in `everything.txt`, line 5846

Duplicate explicit target name: "d".

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "here".

You can download it [here](#) (click on the "Free" button and wait a while).

You will need Qt4 and qmake to make it work.

system-message

WARNING/2 in `everything.txt`, line 5852

Duplicate explicit target name: "diup".

system-message

WARNING/2 in `everything.txt`, line 5854

Duplicate explicit target name: "here".

Just a tiny update to the Makefile, so if 0.2.0 worked for you, ignore this.

I have refactored IUP/Qt into a lovely object hierarchy. And I now remember why I dislike C++.

Here's the basic problem (code may be wrong, I don't have the broken versions anymore):

I have a base class IObject, that has stub setters/getters for all the IUP attributes.

For each kind of widget, I create a class that inherits from IObject and the matching Qt widget (example: QDialog).

These have to be hooked into an Ihandle structure through the use of a void * (remember, that's C, I have that too ;-)

Now, this was my first naive approach:

That compiles ok. However, when you try to call a d member later (and I know I am using old C casts... they had worked for me until today!) ...

Blammo, segfault. Why? Because in one of the assignments to/from void *, the pointer for some reason starts to point elsewhere.

system-message

WARNING/2 in `everything.txt`, line 5876

Inline emphasis start-string without end-string.

Here's what Gliptic at #C++ suggested, and it does work:

Right, I have to cast it **twice**.

And here is how you get it back:

Isn't this just ridiculously weird for anyone coming from any other language?

Or am I missing something completely?

But the good news is, it works ok, and the Qt backend now has a decent structure to hack on.

It has a decent backend, even if it's limited.

Not all widgets are implemented, but they are easy to add. Not all attributes are implemented, but they are even easier.

And it can show rather complex windows, too:

That's an expression ("The world is a handkerchief") I have often heard, and often found true. It means the world is a very small thing and it's hard for something (or someone) to get lost in it.

Here are two examples:

I went to school with this guy. We were never friends or anything. I had completely forgotten about him. One day, I was walking to the bus stop after teaching, and someone calls me.

It's him. This was roughly 15 years after the last time I saw him.

So, we start talking: he worked 2 blocks away from where I did. And 550Km away from our school, and in a 12 million people city.

He had lived until three weeks before in an apartment around the block from my own.

He usually went to the same cafe I did, in the corner of our streets, but he did so 90 minutes earlier than I did.

And we had never seen each other.

Then we did. On a third place, which was not related to neither of our jobs or homes.

Second example:

I met Rosario on May 25th 2004. Except I **almost** met her in 1995 (or so).

I was telling her last night about an affair I once had with a girl that lived a ridiculous distance away from me (800km or so) and how I had last seen her in a congress of the FUA (Argentina's students organization) that took place in Santa Fe, where I was born.

That congress had one party (where I broke up with this girl) in my university's law school.

Well... Rosario was there. I don't remember her, but knowing the place where that party took place, I could not have been the whole night without passing 10 feet away from her.

In fact, I can't get out of my head a false memory of a girl looking remarkably like my wife sitting on a planter with a worried look in her face (she had had some issue that day), with a drink in one hand and a cigarette in the other.

I know I did **not** see her because I didn't remember this yesterday, but... I ended leaving the party with another woman (oh, being 24, rash and pretty stupid... that woman was trouble). What would have happened if I didn't, and started talking with the woman sitting in the planter?

I keep working on it, which is nice. I have finished the skeleton of the implementation of a Qt driver for the CD canvas.

I had it kinda working with the generic X driver, but it sucked a bit to have this largish chunk of Motif code I didn't understand in the middle of it all.

It has only the most basic stuff implemented, and the scrollbar stuff is not there, but it does draw and set colors ;-)

So, right now, the next step is getting the iupcontrols library up, but that shouldn't be too hard, I think.

But, we'll see.

PS: If anyone reads this and wants to check it out, just email me, I can ship it.

Ok, here's a little rant. I keep on reading the whining about IT outsourcing to India (note that I don't live in India and don't do outsourcing jobs) by US IT workers, and you know what? They are a bunch of brats who believe they are special.

They are not.

Hell IT workers in general are not. Those of you who have managed to make a living with it for the last 10 or 15 years, and want to keep your standard of living, learn a new skill, because it's all downhill from here.

Let's start with the outsourcing complaints:

- They only move the jobs to India because it's cheaper: Duh. Perhaps if you had bothered learning basic economics, you'd have seen it coming.
- The Indian companies are doing a worse job: The answer is: maybe somewhat, and it doesn't matter.

Work in a capitalist economy is a simple transaction. You get money, you give up a piece of your life and effort and productivity.

Now, if you charge less, you can often get away with producing less, within reason. Besides, the IT outsourcing is only starting, come back in 5 years and we can have a reasonable measure of how it went. The companies that are coming back after a year can just as easily go back to India in another year or two, and viceversa.

It's globalization of labour, just like making baseballs in Haiti. I bet the first Haitian baseballs sucked, too.

- It's not the same because it's skilled labour: it isn't. You live under the mistaken idea that IT work is somehow difficult. Let me tell you: it isn't.

Here is the real *problem*:

Programmers spend their sweat trying to make things easier: well, IT is way easier than it was 10 years ago, and it is getting easier all the time.

While there will still be a role for the superfreak who can hack the really hard stuff, 99.9% of IT workers are no such a thing. In fact, 90% of the really bad IT guys still manage to make their systems survive. And when they can't, they call a freelance that's smarter and charges more.

Most of you guys, had you been born in 1880, would be train engineers. That was a profession that required great skill and was respected.

However, it was never quite as respected as IT, because of the prejudice against manual labour, and it never went so low as IT will go because there is a limit on how simple an engine's "interface" can be.

The required skill set of a IT worker nowadays, while wide, consists of simple stuff. Practical networking is not really hard at all, system management is both getting simpler, more centralized (and thus easier to hire from a company, or to automate), and less frequent.

Hopefully, software reliability will improve, and thus disaster recovery requirements will become simpler to manage, as will contingency plans.

Look at it this way: the only reason why IT skills are needed is that development has been lacking. But development is monotonously growing, nothing is forgotten in that branch, so IT work is steadily reducing.

On the other hand, the Internet boom mistakenly lead a generation into the IT field, producing a huge glut on the market.

So, the size of the IT worker market is reducing, and the offer is growing... bad news for you.

But why don't IT workers see this?

- They overestimate themselves, and underestimate other professionals.

Mostly, they believe that because others don't understand their work, the others are dumber and they are smarter. Hello? You are probably dumber than 90% of the lawyers, and 95% of the doctors out there, and you don't understand their jobs, either.

The computer guy that says stuff like "how can he understand it, he's a lawyer?" is cliché... and a sure sign that the computer guy is a moron.

- They overestimate their work's importance

IT guys are about as important as the cooling and heating guy. Less if it's too hot or too cold.

Sure, computers are necessary for many jobs. So is power and a timely coffee cup. IT guys are labour. Skilled, yes, but just labour. And remember, there are 100K guys in India willing to do it cheaper.

I have seen people working with a DOS based system, without much trouble, without any IT assistance (except a timely computer vaccuming) for 15 years or so.

- Sense of self-entitlement

Who says you deserve a USD70K pay? The market. If you don't agree that you deserve it, you are SOL. When the market pushes the number down (and it will), you will still be SOL.

Oh, sure, you will have temporary aids by the government, who will probably put some trade barriers of some kind or another. Eventually, those always fail (look at the US steel industry).

The funny part here is, of course, that most IT guys claim to be libertarian, even to like Ayn Rand. Well, that's always easy when there's money, aint'it? Well, I bet in 10 years most of you will be eating crow for what you say about unions now.

A bit long, but here's the short version: You are going to become skilled factory workers in the next 10 years. Get used to it.

Well, timers are done, including the idle timer, and images are kinda there.

It is much simpler to add stuff now that the proper OO skeleton is in place.

However, I am not making progress in the iupcontrols library. Then again, I have the flu.

A customer asked me to implement for him a simple fax serving solution.

Here's what he wanted:

- A central received fax repository.
- A way to send, as simple as possible (the classic fax-printer scenario).

I vaguely remembered knowing that Linux could do that, so I said yes. Then I started trying to figure out how to do it.

The 500lb gorilla of linux fax software is of course Hylafax. And it's just about as pretty and cuddly as a 500lb gorilla, too!

Put it simply: waaaaaay too much software for the goal.

Hylafax is a very complex software package, and while it does have some tools to make management simple, it would probably have forced me to support this thing forever. And that's not mi idea of fun.

Not to mention that I couldn't find CentOS/RHEL4 packages (not too big a problem, but annoying).

Then I spent 10 hours trying to make it pick up the phone. And I started being annoyed.

So, I started looking for simpler stuff, and the second gorilla seems to be mgetty+sendfax.

The more I read about it, the more I liked it, and I finally implemented using it.

The good news:

- It's really simple. I was receiving faxes in 5 minutes.
- It was way less fidgety about modem setup than Hylafax.
- The fax quality was very good (probably not their merit, but it was nice).

Now, the bad news:

- It's **old** software. Check the [webpage](#) and look at the broken links and ancient releases.
- The windows client support was pathetic.

How pathetic: ok, here is how you make it work:

- Setup mgetty+sendfax enough that you can make it send from the CLI.
- Setup samba enough that you can do a shared printer from the windows side.
- The shared printer should print to [printfax.pl](#) (look at the germglisch page).
- On the windows side, start [respond](#) and then when you print to the fax, the fax server hooks to the client's port 5555, and respond show a **huge** popup asking for the number, recipient, and sender.
- Explain to your client that where it says "name" he should put his email address. There is no way to save that address so it's not asked again.
- After the fax is sent, the user gets an email (if he filled the form correctly) with the report.
- The alternative notification mechanism is winpopup. Which isn't there in XP anymore anyway.

Amazingly, it does work fairly well, and the client is not terribly annoyed. I would be, though.

So, what can be done:

- a) Now that there is a PyQt4/Win32 I may write a respond replacement.
- b) Maybe printfax.pl can be extended/replaced to give report information to that respond replacement.

Sounds like a nice fun project... NOT! But it is a fairly simple, necessary one.

So I will probably do it. Or else, I may have to learn how to properly use HylaFax.

One more thing to love about Open Source/Free Software:

Maybe each screw doesn't work exactly as you want. But you can hit them with a big enough hammer. And if you have a big enough hammer, every screw **is** a nail.

Mgetty+sendfax sucks a little in some areas. It supports no class1 faxes (ok, it may do if you rebuild with a switch it says doesn't really work).

Efax sucks a little in some areas. It has no spooling mechanism. It has no way at all to be used decently from windows clients.

So, remix them!

Take efax's efax command (the one that actually sends the faxes), and mgetty+sendfax's sendfax command.

They are quite similar. So, write a tiny shell wrapper that makes efax look like sendfax.

And Voilà. Mgetty+efax, which works on class1 fax/modems ;-)

The wrapper itself is left as an exercise for the reader.

FaxWeb, a web frontend for mgetty+sendfax is finished. It works. It's probably close to bugfree ;-)

The missing piece is a nicer reimplementation of respond (and this one will be cross-platform, too) using PyQt, which is 50% done.

I am only missing how to implement portable systray icons. On Mac they make no sense, on Linux I have it working, on Windows I have no idea.

Here's the simple interface for faxweb:

' <http://www.flickr.com/photos/96455042@N00/139230422/> ' _ ' <http://www.flickr.com/photos/96455042@N00/139230434/> ' _ It even has a little AJAXy "the page doesn't reload" niceties courtesy of MochiKit!

Also from MochiKit, the nicer, rounded look&feel. Compare to this older, uglier one:

' <http://www.flickr.com/photos/96455042@N00/137206420/> ' _ I know the new one is not good, either, but I have decided that since I can't aim for awesome, I should aim for adequate, and settle for boring and harmless.

Of course, if any CSS/XHTML guru volunteers for a makeover, I'd be very happy, since I use the same CSS everywhere (even on parts of this blog ;-).

All in all, a pleasure to write this thing, thanks to CherryPy!

I decided to redo most of it, because I think I figured out a better way.

a) Use efax instead of mgetty+sendfax

After all, efax works with my modem and sendfax doesn't ;-)

a) Write a generic spooling thingamajig for efax (this is easy)

b) Rewrite printfax.pl in a simpler way (done)

c) Simplify the windows side.

Forget about a Qt app. All that's needed is a proggy that listens on a port, accepts connections. The connection gives it a URL. It launches the default browser with that URL.

If anyone reading this can write that program, I would really like it. Use VB, use .NET, use Java, I don't care.

a) Use the web app as the only interface.

What's missing yet:

- Writing the queuing program (should use same format as mgetty+sendfax's)
- The windows "client"
- Minor tweaks to the web app (basically a form for the sending data)

And that's it.

Other than the windows thing, it's about a day's work.

- Wrote a whole queuing suite for efax. It kinda works, but not really, because it's not tested at all.
- Rewrote the web app to work with said suite instead of mgetty+sendfax
- Added status and sent pages
- Implemented I18N for the web app

Next step: spending some actual cents sending and receiving faxes from my home phone line to the guy across the street.

What will happen: I need to do lots of other things, so I will shelve it for a week or so.

I wrote Linux/Unix applications for ten years before I wrote my first man page. I used all the excuses:

- My app has interactive help
- My app is for private use
- Someone will write it eventually

But of course there are only two reasons not to write man pages.

- You think the man pages are obsolete.

This is what the FSF thinks. So, they give you info pages. And they put, at the bottom of each manpage some warning about how that may look like the docs, but the **real** docs are the info pages.

Then they don't provide a decent info reader outside of emacs. But hey, who cares. On KDE, just use a info: URL.

- You hate writing man pages.

Which is perfectly understandable, since this is how a man page looks like (and yes, I know it's specially ugly example):

```
.de }1
.ds ]X \&\\* (]B\\
.nr )E 0
.if !"\\$1" .nr )I \\$1n
}f
.ll \\n (L Lu
.in \\n () Ru+\\n (INu+\\n () Iu
```

```
.ti \\n (INu
.ie !\\n () Iu+\\n () Ru-\\w^G\\* (]X^Gu-3p \\* (]X
.br\\}
.el \\* (]X\\h^G|\\n () Iu+\\n () Ru^G\\c
.}f
```

And it is even possible that for some man pages you **need** to write such a thing.
 But for the average program? Just take something like [txt2man](#) and be happy.
 I wrote man pages for each of my [qmail-spp plugins](#) in a few minutes. Although
 they probably are lacking in their man-pageness in matters of style, they are ok.
 And they are not hard to write at all ([full](#)):

NAME

authchecks

DESCRIPTION

This plugin is meant to be used on the AUTH command and does two things:

1. Logs the user (see LOGGING)
2. If the user is authenticated, it sets QMAILQUEUE "/var/qmail/bin/simso" which is probably not what you want, but is what I use right now ;-)

And there is no reason why your program, be it KDE-based or not, should not have such a thing.

Bilingual entry here:

Yo nací en Santa Fe. Me gusta Python. Hay un evento de Python en Santa Fe.

Pienso asistir solamente como espectador, quedarme callado y escuchar cosas interesantes que digan los demas.

Sin embargo, si quieren experimentar un lugar donde se pueden comer 30 platos de pescado por 20 pesos (y te podes comer los 30, aunque yo conozco solamente una persona con la suficiente fortaleza estomacal), mas toda la cerveza que puedas tomar... bueno, nos vemos en Santa Fe el 3 de junio.

system-message

WARNING/2 in everything.txt, line 3
 Duplicate explicit target name: "aca".

Mas informacion [aca](#)

system-message

WARNING/2 in everything.txt, line 6269
 Duplicate explicit target name: "aca".

I come from Santa Fe. I like Python. There is a Python event on Santa Fe.
 I intend to attend in a strictly spectator fashion. I will listen to other guys telling interesting stuff, and keep quiet.

However, if you want to experience a place where you can eat 30 different fish dishes for about 7 dollars (and I mean you can eat all 30, although I know only one person of such superhuman fortitude), plus all the beer you can drink... well, I will be in Santa Fe on June 3rd.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "here".
```

More information [here](#)

```
system-message
WARNING/2 in everything.txt, line 6281
Duplicate explicit target name: "here".
```

Just because I am a shameless nerd, here is, in all it's geekitude, my 2002 essay about hacking and Lord Of The Rings, rescued from its deserved obscurity for a few minutes, with some extra footnotes...

A gtk engine that draws using the current Qt style.

Kudos to David Sansome, your code is an example of the kind of evil ideas the world needs. And I really mean that as a compliment :-)

Today, a new version of XView was released.

You know... openlook?

Yes, really.

And the FAQs include things like this:

Q: The second ttysw in my program doesn't work

A: XView only supports one ttysw per program!

I do some strange stuff. I read bout world cup matches on the newspapers of the other country. I read travelogues about the city I live in.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "this".
```

And that, gave me [this](#).

```
system-message
WARNING/2 in everything.txt, line 6311
Duplicate explicit target name: "this".
```

- Woke up at 7AM.
- Teaching until noon.
- Bus to a customer... 1.5hs
- Work with the customer... 10 minutes (I charge two hours minimum anyway)
- Bus back 2 hours (more traffic)

Man, this has to be the most tiresome birthday ever. And I am not even drunk. But hey, the day is still young! Oh, wait... it isn't. I am gonna get SO wasted tonight.

I have switched the RSS feeds to feedburner.

Why? Because I don't want people to ever lose my page again if I switch hostings :-)

My old page at pycs.net is still there, abandoned. I can't update it, I can't delete it... it's very sad, because they gave me great service for several years.

Of course it *may* cause the classic "planet thinks everything is new" but no other RSS reader did it, I swear I tested several!

As a bonus, I get traffic analysis of the RSS subscribers, which I never had.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "this".

In [this](#) russian site, you can see how they got my post about the Santa Fe python event. And apparently what impressed them is the thing about the fish and the beer.

system-message

WARNING/2 in `everything.txt`, line 6345
Duplicate explicit target name: "this".

I got this nice multifunction printer+scanner+copier, called HP PSC 1410.

Cheap, too!

Works nice, even!

Except... it can't be networked. You simply can't, using HP's software, print on that thing from any other computer on your network.

Except, that is, if they are both running windows XP. Not even other versions of windows will work. And HP says [so](#). (Correction, maybe not even with XP?)

On Linux? Well, it shoulda kinda work, says [linuxprinting.com](#).

And then I suppose I could export a postscript queue and let the linux driver do the job, and get **some** printing done from the **must-be-windows-98** notebook running the legal case management software.

But you know what? I am **so** tired of this crap. It doesn't work right on Linux. It's crippled on windows. What exactly are we supposed to run for this piece of engineering to work?

Why does this, a rather new model, lack a feature every godforsaken printer has had since the dawn on Windows For Workgroups 3.11?????

I know my page is read by no windows printing guru, but I really can use a hand here.

Fixed a silly bug in the ipthrottle plugin.

It had debug code enabled, and it was getting to the SMTP client :-)

If you tried to use it, and got "invalid response" errors (or something similar)... well, try it.

A simple (yeah, sure!) solution so you can watch TV on all the computers in your home. CherryTV!

It's small, it's cherryPowered, and it does work.

Having running statistics on your systems is always a good idea.

Since I am a qmail freak, I have been using the venerable [qmailmrtg7](#) for a long while.

In fact since I use my courier pop3/imap server with tcpserver, I can even use [qmailmrtg7](#) to give me stats about POP3 and IMAP, which it usually doesn't.

But... [qmailmrtg7](#) is kinda lame. In part it is lame because it uses mrtg. In part it is lame because it's one opaque thing.

[Munin](#), based on [rrdtool](#) is much nicer to look at than mrtg, and has a much more reasonable architecture.

Then, I found [qmrtd](#) which is **much** nicer than [qmailmrtg7](#), and also has a much nicer architecture.

So, why not make a qmunin based on [qmrtd](#) which will marry them?

Well, that's an idea.

It's not even very hard to do :-)) I should probably do it and publish it.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "runit".

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "rpm".

I have updated my [Booting with runit](#) story for the commands in [runit](#) 1.5.1 and included a mention for my easy runit [RPM](#) (which is now also up to 1.5.1).

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "runit".

If you are looking for an alternative way to boot your linux machines, or for a reliable way to run and control your services, please take a look. [Runit](#) is cool.

system-message

WARNING/2 in `everything.txt`, line 6414

Duplicate explicit target name: "runit".

system-message

WARNING/2 in `everything.txt`, line 6418

Duplicate explicit target name: "rpm".

But anyway, you haven't had a decent one until you had a "relleno con dulce de leche", and no, caramel is NOT the same thing.

I left it working and fetching overnight, and it hasn't crashed. It even seems to have got all the items from about 12 feeds without problems :-)

If anyone wants to give it a try, CVS from sourceforge has all the stuff. It can't be installed yet, though, just use it from its own folder.

As everyone knows, the Da Vinci Code is about to be premiered worldwide.

I will probably see it. By chance I ran into the plot [synopsis](#) in the site of Dan Brown (the author of the book).

Hmmmm... interesting.

For example, it describes Opus Dei as

Opus Dei [is a] clandestine, Vatican-sanctioned Catholic sect believed to have long plotted to seize the Priory's secret.

Whoa.... [Opus Dei](#) is a clandestine sect! I must say I dislike the guys (and I know at least 4 of them), but they are about as clandestine as the [Baptists](#). And much less so than the [Unitarians](#).

Not to mention that (I've heard) much of the plot revolves around clues left "hidden" in things like Da Vinci's paintings.

You know, if I were the Priory of Sion, and had a secret... sure, I would hide the keys to the secret in world-famous pieces everyone sees. Or maybe I would ... **hide** them? You know, as in not telling anyone about them. As if it was a secret?

What would you choose? Anyone? Anyone? Bueller?

On the other hand... the good guys are the ones (the Priory) that have the secret and the bad guys are another secretive organization (Opus Dei) that wants to get the secret?

Well... color me unimpressed, but what the hell does either one having the secret matter? If it is a secret, and noone should know about it, why not give it to the guys at Opus and let them keep it? They don't intend to announce it either.

Then the nice fellas at the Priory can go to the beach or something.

But who cares. They are all the same to [kdaptists](#).

A short tutorial explaining how to make Squid validate its users against a POP or IMAP server. It must be useful, because I use it ;-)

Prepare for a flamfest of moderate proportions, courtesy of LinuxWorld.

If Nick Petreley had written this article, say, two years ago, it would have generated enough heat to pop all the corn in Mexico, though.

Special ironic demerits for the use of a forced acronym as mechanism for criticism.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "runit".

I have posted in the past about [runit](#).

One of the problems people migrating to runit have is that all your services are SysV scripts.

The runit author has a collection of scripts you can use, but usually they require some adjustment to work on a specific version of Linux.

So, I wrote a lame python script that takes the SysV scripts you are currently using and turns them into runit services, including dependencies.

Suppose you usually start on runlevel 3. Then you save [this script](#) and run it like this:

```
mkdir services
python importinit.py 3
```

And you should end with a bunch of runit services inside services/

Those services will start in roughly the same order as if you were using SysV init. That's probably way too much dependencies.

The main difference is that kdm will start earlier and the ttys will start way earlier than you are used to.

I have found that my notebook boots faster using this, but I can't provide a [bootchart](#) because it simply doesn't seem to work in my computer.

If anyone is willing and able to run the tests and quantify the difference, I am all ears.

```
system-message
WARNING/2 in everything.txt, line 6515
Duplicate explicit target name: "runit".
```

For some reason kudzu, iptables and arptables_jf don't work with this approach, so just stick them at the bottom of /etc/runit/l

Also, please understand that these are not correct runit services. They are not managed, so if your service crashes it stays crashed.

So, you should still eventually migrate to correct scripts. This is just a way to make that simpler.

I have been contacted by a few people about Qt/IUP. Here's my current state of mind...

- It works mostly.
- It lacks a few widgets
- It needs someone to walk over the docs checking everything is implemented and semantically right.
- It has a (IMVHO) decent structure for the Qt backend. It should be possible to implement everything using it.
- The IUP people know about it.
- It's still just my toy, AFAIK

So, its future seems to rely on me having time and energy to finish it... and there's the catch.

I am Gimli. I am not a marathon runner. We dwarves are natural sprinters! So, I can usually cook up a decent project in a week or a month, and then it languishes.

Specially because I really don't **need** Qt/IUP.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "d".
```

I only got here because I was trying to learn [D](#) and I didn't like any of their portable toolkits, and I found no better small C toolkit I could wrap, and then I disliked it being Motif.

As you can see, an entirely too weak link from me to Qt/IUP to be sustainable.

I will try to push it today, implementing the missing widgets, and see what happens.

```
system-message
WARNING/2 in everything.txt, line 6550
Duplicate explicit target name: "d".
```

...to scratch an itch that isn't there!

For some reason I can't stop playing with my blog's site.

Latest ricer-like "improvements": a google search box and a google sitemap so the search box will work well.

At least the sitemap was something I needed to learn about for a client :-)

There is a command, head, to take the first piece of a file.

There is a command, tail, to take the last piece of a file.

Tail does a few extra things, because it's useful to watch the end of a file that is being updated and similar things, but they should be pretty similar pieces of code, supporting similar options, right?

Well, no.

- head supports a negative number of lines as argument, meaning "all but the last N lines", and tail doesn't.
- Same about bytes instead of lines

That means that you can't quite simply get "all but the first two lines of this file".

But don't worry, this is how you do it:

```
tac file |head --lines=-2 | tac
```

Correction: tail **has** what I wanted. I am just a silly guy that doesn't read the man pages **completely**, you can do tail -n +2 to do it.

On the other hand, it's not explained in the option, and the syntax is different from head's, so it's still slightly rant-worthy ;-)

If you are one of the three persons who actually see this blog on its site instead of reading it through some sort of aggregator, you may have noticed I have a banner.

This is what it looks like:

Nice, ah?

It's not very original (look for "spell with flickr" to see where I stole the idea), but I wrote a script to do it. So here it is, feel free to steal it too, it's your turn.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "this".

You will need [this](#), ImageMagick, and a Flickr API [key](#).

system-message

WARNING/2 in `everything.txt`, line 6605

Duplicate explicit target name: "this".

To use it simply call it like this:

```
python banner.py something
```

And you will have a `something.jpg` with `something` in it.

It really hurts my productivity, because I can't understand a radio show in english and code at the same time (although I manage to *pretend* to chew gum and walk), but hey... it's Penn Jillette, and I love the guy.

You can listen to his show for free here: <http://penn.freefm.com/>

Have fun!

Nice article by Robin Miller at Newsforge.

I agree with most of it, except for one thing.

The author says "I always considered the "scratch your own itch" hacker ethos an essentially masturbatory thing. It would be like me writing articles *I* want to read instead of articles I think *you* want to read."

Well, Robin, articles and code are extremely different things. For one thing, articles are 100% useless for the author, unless someone else reads them, while code can be 100% useful for the author even if noone else has tried it.

I would say it's like comparing apples and oranges, except... well, both are fruits and we compare them whenever we choose what fruit to buy. This is a less comparable pair.

For example, I am probably the only user of KRnS in the world. But it *is* useful for me. It saves me almost an hour a day, so I wrote it.

On the other hand, I will contradict myself: articles almost noone reads (like this one ;-)) are useful for shaping one's mind, although that's not a frequent goal.

On the gripping hand[1], masturbation is good, dude. Don't slam it unless you have never done it. And if you have never done it, don't slam what you don't know ;-)

In fact, programming because it itches is... well, it's like scratching. It may even be bad for you, but don't tell me it doesn't feel good.

[1] One geek credit to the one who catches the reference.

Thinking about [CherryTV](#) and ways to turn it into a real application, I thought the worse piece of it was the reliance on `v4lctl`, and how really you just don't know if it works or not, and how you can't finetune, and whatever, and run into [Python-v4l](#).

It has remained apparently untouched by two years, but I managed to build it with one edit and to make it work by switching a line to an alternate version (they are both there), and it's nice.

Here's the example TV viewing application using it:

If you have seen the equivalent C app... well... nice job here!

I saw *The Da Vinci Code* saturday night. This has spoilers. So don't read it if that bothers you, ok?

I can now post and noone can complain about how I had not read the book. Although of course, I *haven't*. But this is about the movie.

First, I want to say that I must correct my previous [comments](#), caused by reading a synopsis of the book in Dan Brown's site.

I must say (assuming the movie follows the book) that it doesn't paint Opus Dei as a clandestine sect at all (not even as specially interested in obtaining any secret). Only

one specific couple of Opus Dei guys are. Which is fine and dandy for me. Secret societies can chase each other around the globe. Their problem.

Of course it also means the following (again, if the book is like the movie):

- The comments defending the book's portrait of Opus Dei as a secret society as appropriate fiction are nonsense, because the book doesn't do that. You were defending the book of something that's not in the book.
- The guy that wrote the [synopsis](#) in Dan Brown's official site has not read the book.

Having said that, onto more serious matters...

Ron Howard is incapable of filming a comprehensible action scene. Don't trust me? Watch the Smartcar chase.

Tom Hank's hair is scary. It's a weird bilateral comb-over. I used to do that. I don't anymore. I am right about stopping. Besides, it's way too distracting.

Professor Langdon, I presume?

I enjoyed the movie as a popcorn flick with pretensions, but most of the plot follows no logic.

Saunière triggers an alarm, in the Louvre, and is then shot in the stomach. Then he traipses around the museum, finds a marker visible only on UV light, does things to three paintings, hides a key behind a large, heavy painting, takes off his clothes, creates an anagram, writes it on the floor along with some numbers, draws a pentacle on his chest, arranges himself in a position reminiscent of the Vitruvian man, and then dies.

He not only does all that instead of calling an ambulance on his cell, but he does all that before museum security gets there. In the gallery that has the Da Vincis.

Not only is it unlikely, but it also is stupid. Had he died earlier, he could, for instance, have been found with the key in his hand, and no clue left for the "good guys".

It's amazing there is still **any** paintings in that museum, with such security.

And don't get me started on the biiiiig secret. It turns out the Priory of Sion protects a secret about Jesus.

If said secret was revealed, it would damage the catholic church.

Of course... the catholic church also knows the nature (and details) of the secret, which means the catholic hierarchs dedicate their lives to a faith they know to be false. Which makes no sense, really.

And then it turns out that several historians **also** know the nature and details of the secret, and have published books about it (except, of course, they have no evidence).

LAST WARNING, HUGE SPOILER HERE

REALLY

There is, however, one detail only the Priory is supposed to know: the location of a corpse that could be used, via DNA analysis, to show that some person is a descendant of some specific other "historical figure".

Which is, of course, absolute nonsense.

Suppose I show you a corpse and tell you “this is the corpse of Joan of Arc”. You carbon-14 date it, and do the usual forensic analysis, and all agrees. It’s a woman, that died in a fire, at such age in so-and-so year.

Then I show you a DNA analysis that shows she is my great-great-granny.

Am I the scion of the Orleans Maiden?

Hell no! Because to accept that, you would have to accept that the corpse is hers!

You can only reasonably do that if there is a clear historical record of the whereabouts of the corpse until now, or else it’s a rather simple forgery.

For example, nowadays we used DNA of known descendents of Columbus to decide which of his two alleged bodies is the real one. About a known historical figure, world-famous in his life. We are just not sure of where his corpse is. We have two of those.

Since the “witnesses” of the authenticity of this corpse are the ones that are bound to gain from the claims, it’s suspect at best.

If you go back a certain number of generations, almost every corpse will be your granny.

I am pretty sure that a large percentage of modern europeans are related to almost any random 2000 year-old corpse.

And, in the specific case of the movie (or the book), even if you assume it **is** the corpse of who they say, so what? That shows she is the descendant of a certain woman, not of a specific man. Get it? You don’t prove the big premise. Only the little, meaningless premise, that M.M. (not Marilyn Monroe) had a child. Who gives a damn?

The secret the Church is trying to keep secret, the secret the Priory is not trying to make public anyway... doesn’t matter. It doesn’t cause what the church fears, it doesn’t cause what the Priory hopes, it does **nothing**.

So, really... much ado about very little. If I were the church, I would let them say whatever they want, and **nothing** would happen. Absolutely **nothing**.

Not to mention that the apocryphal Leonardian device, the *cryptex*... it’s ... I have no words. If you missed it, the idea is that there is a papyrus inside it, and a vial of vinegar. If you try to open it without the key, the vinegar “dissolves the papyrus”.

Do you know what papyrus is? It’s made of the stems of a plant, and it looks a lot like paper.

It’s cellulose. It doesn’t dissolve in vinegar. It’s like saying **lettuce** dissolves on vinegar.

You can make a paper that dissolves on vinegar, but papyrus is not paper.

It may make **some** sense if you said the ink used dissolves on vinegar, but it’s not what they said. What they said is stupid.

So, it defies reason how so many people can enjoy a book based on a premise innocent of logic, about a conspiracy to protect **nothing**.

Today’s entry has no pieces of neat python code, no questions, no critic of anything, no interesting link, no hint of anything I wrote/(would/will)write, no funny piece, no unfunny piece, no nothing.

I am **tired** today.

So, have a nice day :-)

Read to find out...

... if I want to store a password in kdewallet, from a non-kde app, how can I do it?

It would be better if there was a client API I could link to that didn’t involve kdelibs, or at least a DCOP interface.

This is nice technology, I want to use it.

I am now almost recovered from the trip to Santa Fe for the [Jornada Python](#) and I have reached a few conclusions.

- 1) I do kinda miss my old city.
- 2) It does take slightly longer than 15 minutes to show it ;-)
- 3) This kind of events is cool
- 4) I am **old**, and can no longer take a 6-hour bus ride and then go to work.

On more detail:

The city is much prettier than it was when I left 5 years ago. Lots of construction going on, everything is cleaner, nicer, and more organized, it seems.

Man, I had forgotten how great you can eat there. And how cheaply. Remember I mentioned that fish-eating place? Well, it's only \$22. That's pesos. So, it's about 7 dollars for all the delicious fish you can eat, and all the cold beer you can drink.

Which in my case is quite a lot. Although I am getting slow on the beer, too.

The only bad thing about it is that the guys that went on Sunday got Pacú and we didn't. Dammit!

Then there is the Sociedad Alemana. The ugliest place ever. The best *picada* [1] ever. It's a clubhouse founded by a former Graf Spee crewman. If they eat like that on the german navy, I would consider joining. And I get seasick.

Oh, and the lebanese restaurant. I was forgetting that.

Of course all this eating and imbibing also meant I missed half the event, but the morning conferences were introductory, so I got the good bits anyway.

Specially: I now understand **metaclasses!** and **decorators!** Yipee!

We had a good time with Rosario, she got to know better some of my friends, specially Javier and César [aka Ned Flanders] (and Cesar's family, Norma, Florencia and Julian).

We took a ride over the costanera (a riverwalk), peppered with comments like this (all dates are wrong):

Here was the Piedras Blancas beach, until the flood of 87.

That's where the old train bridge was until the flood of 69, then there was a liftchair, until the flood of 82.

That suspension bridge is not the original, which was washed off in 83 (and then stolen)

This piece of the costanera is new, the old one was gone in 78

And so on. You see, this place is... kinda floody. Architecture is not a permanent thing.

On yet another angle, I am going back to school next year. I don't intend to finish my Maths degree, but will go for System Engineering instead. We'll see what happens.

1

system-message

WARNING/2 in `everything.txt`, line 6859
Duplicate explicit target name: "1".

We had the german style picada. That means plates of potato salad, smoked ham, three different sausages,

Today is the first day of the world cup. While I fully expect the verbe and savoir faire of Paulo Wanchope to whip those german animatronic figures into submission[1]_ I started thinking...

The first cup I remember is Argentina 1978. Which, of course, we won. Thanks maybe to a shady grain shipment to Peru, maybe because Cruyff refused to come to the country, maybe because we had the world's strongest guy playing[2]_, maybe because the host nation had won like 60% of the time so far, but we did win.

I saw Argentina-Hungary in a movie theater with the screen decorated as a huge TV one day after it was played. Since we lived under a bloody dictatorship, everyone had to standup while we saw the taped anthem-singing, and there was a sort of desperate mandatory patriotism. But anyway, I was 7, it was fun.

Then came Spain 82. I was in school during most matches, so I didn't actually see almost any of them. We had a rather nice team, with Maradona and Ramon Diaz, but we met Brazil, which had a marvelous team. Then the final was the average versus the depressing, and the depressing won. Italy won with goals by an actual, proven game-fixer. And somehow noone remembers that.

Mexico 86. The first cup where I saw every damn match. And of course, one of the most memorable cups ever, since, again, we won. And we had the best player in the cup by some margin.

Of course, not everything could be flawless, and a deeply mediocre english team feels that they would have won except for the "Hand of God" goal. Well, here's the thing. If the english had won, it would have been injustice of such a mind-bending size, armageddon would have happened right there, 21 years too early.

Not to mention that the bitter dudes simply can't enjoy being beaten by the second goal[3]_.

Sure, we had to play only England, Belgium and Boxcar Willy[4]_ but then again, we have been on the "Group of Death" for 8 years in a row.

When you play in a park or schoolground, if there is a player that's incredibly good (or everyone else is incredibly bad) we say "with that gay, it's stealing". Well, in 86, with that guy, it was freaking armed robbery.

Italy 90. I have high blood pressure. I blame Italy 90. We started by being the first champion ever to lose the opening match, against Cameroon, because we didn't play Caniggia until minute 70. In the following ten minutes, two Cameroon[5]_ players got redcarded. If he had played from the start, we would have won by default because Cameroon would have ended with 5 players on the field.

Maradona had his left ankle the size of a grapefruit, and the color of Merlot. He could barely **walk**.

Our goalie was injured in the most gruesome accident I ever saw in a football field. On the slow motion reply you can see his leg turn into a Z in a very wrong way.

Then we played Russia where Maradona committed a clear penalty and wasn't called[6]_.

We seemed to advance in every round by penalties, and were pretty much unable to score at all.

In the match against Brazil, we were so overmatched it was kinda funny. And then we had **two** chances. One went in, the other hit the crossbar. Brazil had about 400

leverwurst, pickles, three cheeses, and a sausage cazuela. The classic Santa Fe style picada may include any of those plus meatballs, milanesa (schnitzel-like thing), chips, olives, lupines (a kind of pickled bean), small sandwiches, and a hundred other things. All that in small pieces, ready to eat with fingers and/or toothpicks. All salty. So you get extra beer.

shots on goal, and we had the luckiest guy on earth on goal, so...

We played Italy. We had one chance to score. It went in. They tied.

We won on penalties, thanks to the most incompetent penalty kicker I ever saw [7].
Serena, wherever you are, if you ever need a drink, I'll buy you one, round-footed-wonder!

The finals. We were without the only guy that could score in the team. Without our best defender. Our best player was so bandaged he couldn't use shinguards. Everyone was way too old. Batista was playing like Boris Karloff's Frankenstein. We had ran out of players, out of ideas, out of time.

And we lost. Because of a penalty. Which was **not** a penalty (see the replays, really).

There have only been two players ejected on a cup final ever. Guess who they were?

But you know what? The guys played like god damn lions. If you are great, then you are supposed to win. We were not great. We were not even average. That team was a lot of senior citizens, one fast guy, a crippled wonder and duct tape. And they got to the finals. It was as if a smart, poor fat guy with acne and bad breath picks up Nicole Kidman, but doesn't get to have sex with her. He is still to be admired.

Then came the real heartbreak. USA 94. That cup never happened. I refuse to think about it.

France 98... so so. We got to quarter finals, which was about right. We could have gone further, but Ortega is the dumbest player in the universe, and he decided to headbutt a dutch in front of the referee.

Japan-Korea 2002. We were favourites. We were the best thing since sliced bread. Our coach was a mad genius. A tortured soul. We had the last chance to play one of the most implacable scorers of all time. We had a dazzling midfield. A solid defense. We had beaten Brazil in Maracana a little earlier. We were kickass.

We had a hard group. It was tight. We lost a close match to England. We had to play Sweden and we had to win. But we didn't.

I was living in Buenos Aires and didn't have a TV, so I had to go to bars to see the matches. Argentina-Sweden was played at some ungodly hour, like 3AM, and I went downtown to see it.

Of course, we were one goal short. And we were out of the cup. I had never seen Argentina fail to exit the group stage. And I was at 5AM in the street, in mid-winter, feeling really, really, really bad. Which means that of the last three cups, two have been horrible, one half-decent.

Since Maradona retired, that's our harvest. This time... this time we had a decent team, coached by a nice guy. We have a probable superstar, but he's too young and a little injured. We have a terrible goalie, an aging defense, a lot of above average forwards... I say semis, or quarters. If we get any further, it will be in the Italy way, not the Mexico way.

But dammit, I will be watching.
I will start by assuming the following:

- 1) Everyone understands the offside rule.
- 2) Everyone saw the first Costa Rica goal in today's Costa Rica - Germany match

One basic rule of economics is that people, all things being equal, will rationally choose the course of action that gives the greatest reward.

Knowing that, why do teams, even teams in the highest levels of competition, insist on such plays as Germany attempted?

The idea, obviously, is that the defenders play in a line, and move forward a fraction of a second before the attacking forward starts the pass, thus leaving the receiver of the pass in offside. You do that, you get the ball. Simple.

1

system-message
WARNING/2 in `everything.txt`, line 6920
Duplicate explicit target name: "1".

Yeah right.

2

system-message
WARNING/2 in `everything.txt`, line 6922
Duplicate explicit target name: "2".

You can see Kempes dragging no less than three rather large dutch people hanging from his shirt. And he didn't even slow down.

3

system-message
WARNING/2 in `everything.txt`, line 6924
Duplicate explicit target name: "3".

I actually prefer Maradona's Belgium goal.

4

system-message
WARNING/2 in `everything.txt`, line 6926
Duplicate explicit target name: "4".

Our side of the draw were all bums.

5

system-message
WARNING/2 in `everything.txt`, line 6928
Duplicate explicit target name: "5".

Everyone remembers this Argentina team as thugs. Cameroon was about 10 times as rough, and noone remembers because they were the *feel good* story of the cup, with Milla being 56 years old or something. But if you attacked them, you ended barefooted, flipping in the air.

⁶God plays defense, too.

⁷Except Palermo, who missed three (3) penalties in the same match. Then again, he actually kicked two of them decently.

The bad news are, of course, that such plays work about 50% of the time. If that. It requires the defenders to move forward in the exact same fraction of a second. If one stays late: it doesn't work.

It requires them to synchronize that way by an audible call, because it's very hard for them to **see** their whole line: they are stading sideways almost every time.

If the passer is smart, he will simply avoid the pass and run a diagonal.

If the receiver is smart, he will fall back, avoiding offside, and rely on a teammate to sprint for the pass.

And all these problems can happen **every time** the play is attempted.

The risk is huge. In fact, nowadays, in order to lure the attacking team into this trap, you need to play it with all your 4 defenders! Each extra defender probably doubles the chance of error.

And what do you avoid? ...a low probability long pass. Remember, in football, every attack is low probability. A penalty is only about 80%!

So, the german coach is incompetent. He may do some things well, but Germany's defense is atrocious. They will not win the cup.

Poland is... enough about Poland.

Ecuador played ok, they tried to stay simple, do their thing, make the other team beat them. Which Poland had no interest in, so Ecuador scored a couple of goals just in case, and hey, they won! Poland is perhaps the worse team so far, and that includes Trinidad and Tobago. And Costa Rica.

England, I have heard, are slow starters. I hope for their sake that's the case. Ashley Cole has a serious case of believing he is way more skilled than he needs to be (I thought Rio Ferdinand was the one with that problem?) And anyway, Paraguay started weak, and had huge trouble attacking. So, good win for England, hopefully Paraguay was just nervous. Gamarra is a really unlucky guy.

Sweden... what a frustrating match. T&T was playing on guts alone, their best forward played as a defensive midfielder, and still they had the clearest shot at goal, out of a pass by the goalkeeper! Sweden was just incredibly unlucky, or their forwards were blindfolded.

Ivory Coast is a pretty good attacking team. They are a below average defense, though, which explains why Argentina scored twice on three chances, and the other one was saved 90% behind the goal.

Saviola played well, Riquelme didn't. The defense... they played waaaay back, and Abbandanzieri simply gave away every ball he touched with a long kick to an ivorian.

But hey, it's a win... no obvious trouble... need to play with some confidence... ok, we suck a little. But it can get better.

Palacio for some reason couldn't stand up in the field, was falling whenever he tried to run.

But I think everyone agrees: the Ivory Coast is a scary team. They are skilled, they are athletic, they are not too badly organized... they can be a chore for any team.

So far, of the "good" teams... none has looked very good. No candidates to win yet, IMVHO.

I have not posted in a few days, because I have been very busy.

I have not even been able to see all matches (I missed USA/Czech, Italy/Ghana).

But man, did watching Argentina/S&M pay off! :-)

A **very** good match by Argentina, which of course brings up the obvious questions...

- Is the S&M defense made of beheaded chickens?

Allegedly it's one of the best defenses in Europe, but their best player was out. So it's not conclusive. My bet is that they are not beheaded chickens, and that Argentina made them look bad. Or rather: 25% beheaded chickens, 75% Argentina's merit.

On the other hand, the player that said they should have attacked against the dutch was wrong. It's not called being a coward, it's called knowing you can't score.

- Was the good Dutch performance against S&M real?

Who knows. We will figure it out for sure after they play Ivory Coast. My bet? Tough match for the dutch, possibly a tie.

Now for the other teams:

England has **nothing**. They suck much more than you think now. Wait until they play a team that actually has a forward who can score (and is not playing 5 like Yorke). Set pieces for Beckham at midfield and Crouch trying to head it in is not something that's going to work a lot. For their sakes, Owen and Rooney better start showing something completely different than against T&T. Which still has a chance. I would prefer them and not Sweden to stay in the cup!

Germany has something. The defense was improved (although Poland has a lame attack), and their forwards are not bad.

Spain has a lot. Specially going forward, and they made Ukayne look like amateurs. We'll see if that was Ukayne's fault or Spain's merit later on.

Brazil has something big. His name is Ronaldo. Remove him and they are going to kick ass.

Last night I saw an episode of CSI Miami.

While I like a techno-geeky police procedural show as much as the next guy (and most guys do :-), it left me with a queasy feeling.

Have I been bamboozled?

In this episode, one of the corpses was killed in his office, where apparently he had been using his notebook to connect via Wi-fi to use another building's access point to reach an escort via email, bypassing the security of his office network.

Now, that's all cool and nice, but the problems started quickly.

First, they managed to read an email the deceased had sent, by inspecting some computer in the access point LAN.

Ok, maybe they had a SMTP proxy, or the guy had used the SMTP server and they had full message logging (but no sending address control?).

Unlikely, but possible.

The big problems came when they recovered the notebook and could see two response mails from the escort.

- They had IP addresses like 24.123.43.12.1109
- They saw that one had that address, and the other mail had .1108, so they decided that:
- One was a forgery
- Both came from computers in the same network
- They could find both computers in the cyberporn outfit where the escort worked

What's wrong?

Well, Those IP addresses are mighty long and have very large bytes. Also, why on earth did the computers that sent the messages have static IP addresses?

Shouldn't both have been NATed to the same IP?

Unless they used a private SMTP server, of course! In which case, it would have had email address checks on the sender. Remember, this was a cyberporn outfit, the porn people KNOW IT. Better than most companies!

Finally: they went after a suspect because of the IP address in an email???????

That is so easy to fake it's not funny. If the stuff I know about is so wrong, how right are the things I *don't* know about, like DNA testing and whatever.

BTW: one of the investigators had a blood sample "done", without telling the lab worker the reason, had a single page printout, and comparing two numbers to a similar page of a dead guy, he knew the blood was from the dead guy's daughter.

I am pretty sure that to test genetic relation, you need to do something a bit harder than that! ;-)

So, in short, sure, we are bamboozled every week. All the magic is smoke and mirrors. But hey, it's fun, and the CSI offices look like starship Enterprise circa 2300, except the desks are covered with bottles of pretty colours :-)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

I stumbled upon Thomas Thurman's post on his blog ([here](#)) where he comments about a discussion on The Guardian about how .99(recurring) is or is not the same as 1.

Of course to anyone who knows what a rational number is, .99(recurring) is simply a very long way to write 1. Hell, to anyone who bothered learning his fractions, that should be obvious!

But anyway, one of the comments mentions Hilbert's Hotel, which is a pet toy of mine.

If you are uncomfortable or annoyed by the concept of infinity, you may want to avoid the rest of this post.

Hilbert's Hotel is this paradox (Thanks [Wikipedia!](#)):

A hotel with an infinite number of rooms (1, 2, 3 and so on, so it's a *numerable* infinity) is full. A guest arrives. Yet he still gets a room. How?

The answer is that you ask the guest in room 1 to move to room 2, from 2 to 3 and so on. Then the new guest goes to room 1, which is now free.

Because of the nature of infinity, this works, while on a finite hotel it wouldn't.

- Unintuitive things about infinity: If you add any number to it, the result is the same infinity.

Now assume an infinite (numerable) number of guests arrives. You have to ask the guest in room 1 to go to room 2, the guest in room 2 to go to room 4, and guest n to go to $n*2$.

Now you have an infinite (numerable) number of free rooms: all the odd rooms.

- Unintuitive things about infinity: If you multiply it by any number, the result is the same infinity.
- Unintuitive things about infinity: If something is infinite, it contains a part the same size as the whole (follows from the previous two things, and is, in fact an "if and only if").

However, here's where it gets tricky. You could get a certain number of new guests and there could be no way to fit them in the rooms even if the hotel was empty.

Because there's infinity, and then there is **infinity**. You saw that whenever I mentioned the number of rooms I mentioned they were infinite (*numerable*)? That's because you can put an integer number to each, and number them all.

There are infinite sets of things that are **bigger**, they are literally *uncountable*. You can't put a number to each, even with an infinite amount of time (and yes, I know that *infinite amount of time* there is a **big** problem).

The simplest set imaginable that large is that of the real numbers. The real numbers are all the numbers you can imagine, allowing for infinite decimals, and allowing that those decimals may not ever be recurring (so you have things like 2, 1/3, and pi).

Showing there are more of those than there are integer numbers is not simple enough for this but go along with me for a while.

- Unintuitive things about infinity: There are different sizes of infinite. Go blame Georg Cantor.

Now it gets really weird. Suppose we call the size of the infinite in Hilbert's Hotel A0 (I have no idea how to do an Aleph, sorry), and the size of the real numbers C. Cantor showed how to build, once you have an infinite set, a **larger** infinite set called his *power set*.

That means we now have a whole infinite (numerable) "sizes" of infinite things. Those are called the transfinite numbers.

- Unintuitive things about infinity: There are **infinite** different sizes of infinite. Go blame Georg Cantor some more.

Which brings a lot of questions:

- Are there only those? Isn't there something between A0 and C which is some in-between size?
- Is there an infinite set that's smaller than the integers?
- Ok, infinite infinities... infinite (numerable) infinities, or infinite (something else) infinities?

Well... I have no idea. And last I checked, which was long ago, and my memory is no good, no one else knew.

This is the kind of things that will tell you whether you could be a mathematician. Do you find all this talk about transfinite numbers intriguing and mysterious, or just dull and boring and impractical?

If you find it dull and boring, it may be my writing, or you may be unsuited for maths.

If you find it intriguing and/or mysterious, it certainly is not my writing, and you would probably enjoy maths in your life. Where else are you going to run into "no one knows that" this quickly?

The problem is, of course, that most of the **fun** math has already been done at least a century ago, but there is always a chance of something fun and intriguing and new coming along.

The last I know of was Gödel's theorem, which is really simple enough for anyone with knowledge of arithmetic to follow, but weird enough for 99.99% of the people

to go crazy about (and for those who don't really understand it to write whole books about it applying it to totally improper subjects).

But you know, noone really had thought of such thing as "larger than infinity" quite as Cantor did, and noone thought about Gödel's subject quite as he did before him.

Maybe we **are** missing something absolutely simple, incredibly elegant, awesomely shocking somewhere in basic maths. Not likely. But possible. Wouldn't it be **fun** to find it?

BTW: Gödel starved himself to death and Cantor "suffered poverty, hunger and died in a sanatorium".

system-message

WARNING/2 in `everything.txt`, line 7137

Duplicate explicit target name: "here".

Everyone knows about the hidden satanic messages in songs.

You take a song, you play it backwards, and in certain places, you will have the singer saying something evil, like "I like eating puppies with cinnamon".

I have always assumed that this happens because our brains try to recognize patterns in the sounds they get, and they are a bit too good in that job, but now I have proof.

Here's a video Rosario (my wife, hi dear!) sent me:

In it you can hear pieces of pop songs, in english (and latin), and subtitles of what they **seem to say** in spanish.

Now, unless you believe Avril Lavigne actually says "Leiva quiso venderme el Ford" (Leiva tried to sell me a Ford), and Marley sings about "Where is Julia", the "picking too much signal" theory seems true.

Specially, if you are told what you **should** hear, it works much better!

I had heard these songs a million times, and I had never thought they said that, but with the subtitles... some of them are pretty close :-)

The issue of why subliminal messages encoded backwards in songs make no sense in the first place is another topic.

Read the comments. I am ashamed of mathematical education, right now. If these people has passed any mathematics tests (and some even claim to have gone to college), maths are hopelessly difficult.

Some choice quotes:

1/3 is a symbol for a set of 4 words, it is not a NUMBER.

Only a single number CAN POSSIBLY = 1. Other numbers may ADD UP to 1, but they don't EQUAL 1. Since 1 clearly = 1, .99999 repeating simply cannot equal 1.

.33(repeting) is irrational.

.99999 does not equal 1. It might in the CURRENT UNDERSTANDING of mathematics, but that don't make it true.

Mathematics cannot even prove that .99999 ... is not equal to 1.

Right now, math really can't deal with infinite numbers

.9 repeating, an irrational number, is ABSOLUTELY EQUAL to the rational number 1. Can this be used as proof to show there is no such thing as irrational numbers?

I'm half tempted to say there isn't really a right or wrong answer

I think I've come to the conclusion that $.999... = 1$ in the same sense that $.333... = 1/3$. Which is to say, it doesn't, quite, but we treat it like it does because our decimal system has problems.

0.9 recurring does not equal 1. Why? Because it's 0.9 recurring.

$1 = .9$ repeating IF WE WANT IT TO.

This is an exploitation of our numeric system, to arrive at an outcome that is indeed very close to being true, but the closer it gets to being true the further away it actually is.

2.9 repeating plus 2.9 repeating equals 5.9 repeating 8

And this last one is amazing. The poster proposes a number that is a 5.9 (an infinite number of 9s... and an 8). Right. An 8. After infinite 9s. At the end of them. Right there. Go to infinity position, then one more. There's the 8.

My mind boggles. And it's a mind that actually accepts $.99(\text{repeating}) = 1$.

If you have read the past 5 posts, you saw it was coming, right? ;-)

Via [Slate](#) here's a paper applying game theory to penalties:

<http://www.econ.brown.edu/fac/ipalacios/pdf/professionals.pdf>

The conclusion (and they made me retype this, because of stupid DRM):

The implications of the Minimax theorem are tested using natural data. The tests use a unique data set from penalty kicks in professional soccer games. In this natural setting experts play a one-shot two-person zero-sum game. The results of the tests are remarkably consistent with equilibrium play in every respect: (i) winning probabilities are statistically identical across strategies for players; (ii) players' choices are serially independent. The tests have substantial power to distinguish equilibrium play from disequilibrium alternatives. These results represent the first time that both implications of von Neumann's Minimax theorem are supported under natural conditions.

In human:

- Players are pretty good at making decisions according to game theory
- Game theorists call that good

However (this I am making up as I go)...

- A well kicked penalty is a goal, because there are places the keeper simply can't reach in time.
- Since there is a winning strategy, it makes no sense to apply minimax: there is a global maximum (yes, I know, it makes sense, you just need to consider missing the goal as the chance of failure, then it is not a winning strategy... and I am not going to read the 21-page paper to see if he thought about it).

Of course the winning strategy (strong kicks to the top angles of the goal) is impractical for mere humans. All the more reason to consider [RoboCup](#) the most important tournament for the future.

I wrote in my first post about the world cup this:

We have a probable superstar, but he's too young and a little injured. We have a terrible goalie, an aging defense, a lot of above average forwards... I say semis, or quarters. If we get any further, it will be in the Italy way, not the Mexico way.

When I said the Italy way, I meant as Argentina advanced in Italy 90: with lots of suffering.

Now, this was not even close to the equivalent match in Italy: relentlessly being dominated by Brazil, but boy was it painful.

Not a great match, but I have hope that's just because we match up badly with Mexico.

Germany beat the tar out of Sweden, and England has just finished beating Ecuador.

- Germany is good, but I still want to know what happens when they play a team that can actually score. The closest they had was Ecuador, but they played without their best forward.
- Sweden... what a depressing team.
- England... Like Sweden, but with a guy that can do free kicks.
- Ecuador... nervous. But not a bad game, they could have won if they had decided to bring it to England, which should have had some player thrown out for repeated fouling.
- Mexico... guts. Lots of them. People have said Lavolpe is crazy. His name means "the fox". So, yeah, he is crazy... like a fox! He made a perfect tactical setting, but was unlucky with injuries.
- Argentina... we can get better. I feel there is still another gear. But we need it on Friday.

And to all my German friends in KDE... good luck, and a painless 1-0 defeat to you! (just kidding ;-)

BTW: this is what happens when Argentina plays (game started at 16 hours):

Thanks to a post [Rich Burrige](#) I found out about [Talkr](#) which provides a cute RSS-to-voice service.

It works pretty well, and it was a piece of cake to add to the blog.

So, you can click on the "Listen to this post" links and ... listen to this post.

In any case, this confirms I am a blog-fidgeter. And if anyone knows of any other cute toys for it, please let me know :-)

I am not an artistic person. I am not able to appreciate whole arts (poetry doesn't move me, Lyric Singing annoys me). But I do have a taste, which is my own, although I understand it is not exactly *good* taste.

Now, what do I like?, or rather, why do I like it? Does it say something about me?

I find that I don't like any form of art without intrinsic difficulty. Or rather, that I enjoy more if it is somewhat difficult *technically*.

For example, I know all the theory behind why this is supposed to be a great painting:

Quoting:

Hermann Nitsch's work draws parallels between religion and the ritualistic spiritualism of creativity. Heavily entrenched in ancient philosophy and a dissident, questioning Christian theology, he actively seeks catharsis through pain and compassion, a rigorously disciplined quest for ethereal release and enlightenment through an embracing of primal instinct and ancient sacrament.

Erm... I see mostly a red blotch, which I suppose makes me a philistine. On the other hand, I see this, and I actually see a lot more that fits that description:

I like Ingres more than Rothko, I like Rubens more than Picasso... maybe I am just old fashioned?

I think not. I think I despise those who decide to master a game with no rules, where you can declare yourself winner without contrasting yourself to other players. That's why we watch the football world cup and not other games, because it's damn **hard** and you have to do it with your feet. I think modern painting is taking the ball in your hands and declaring yourself revolutionary.

This outlook, that having a good technique, a domain of a difficult craft **before** bothering with art has some strange effects in my life. I don't like the low hanging fruit. But then again, I am not really tall enough or strong enough for the one that's on the hard to reach branches.

That leads to a life of almost unending frustration and yearning, yet gives me lots of energy, and I think I have come to do some things I wouldn't have done had I settled for easier pickings.

I have been working for years on how to harness that thrust for my own benefit, and I am not too good at it yet. Maybe that's the toughest craft I need to master, and I am working on it.

So, we're out.

The match was pretty even, not badly played, but both teams played excellent defense, so ...

Penalty kicks can go either way, and we are out.

What I have learned:

- Enalapril does work.
- Tevez is a monster.
- Germany will not win the cup... unless they get some more luck. Like Ukraine in semis and France in the final. Anyone else and they are toast (ok, maybe Italy sucks enough).
- We have a **very** good shot in 2010.

I am obviously not feeling ok right now, but ... I said quarters or semis, and I said it would be with suffering. So there. I was right. Not that I actually **enjoy** being right. It's more like I **hate** being right.

... let's turn to a completely different subject.

In my professional life (that is, not counting odd jobs) I have had only three employers.

I worked for many years for the nice guys at [UNL](#) and for about a year for the not so nice guys at Conectiva (now Mandriva) Ok, I take that back, they were nice guys, they just were not a good place to work at. Or rather to **stop** working at.

And the third one has been me.

I must say I am pretty happy with me as a boss. I have managed to go through some very difficult moments in my country's economy, and never failed to cover payroll. Then again, I was very willing to take some paycuts, so I am a model employee, at least about that.

But... I have decided, a few months ago, that there are clear limits on how far being one's boss and sole employee can take you, so with a couple of friends we have decided to start a company, sortof.

It's pretty tentative right now, but we are all three quite knowledgeable in many things, and our areas of expertise overlap enough so that we can cover each other, and are different enough that we have a varied service offer.

Our main focus is system/network/DB management, and a little coding. So from now on, you will see every now and then, updates about this new project.

Where are we now:

- We have a few customers.

- Each of us is keeping most of their old ones, too.
- We are using Trac to manage issues like ticketing and documentation.

What we need to do:

- We need to start marketing.

Anyone can recommend a good **nerdy** hosting? Unmanaged, cheap, Linux? We are going right now with Tektonic, which has some very cheap unmanaged plans.

Anyone can recommend good literature on how to manage a service company?

Anyone thinks this is a **very** bad idea? Or a very **good** idea? Why?

I am pretty psyched about this, these are the two persons I would trust most in this subject, and we have been friends for a very long time. It's going to be fun :-)

Now, this is not a tutorial in the sense that you gonna learn how to **do** stuff.

However, it may show you what is *possible*!

So, hop in and enter the wonderful world of one evening of development in the company of PyQt, Eric3, and Python.

I tested the Skeletonz CMS. I liked it. Lots more details, of course, following the link.

A while ago, I wrote about my misery trying to use a HP PSC 1410 printer attached to a XP box as a network printer.

Basically:

- The Linux driver freezes the printer if you print remotely (but works locally)
- The **basic** windows driver will not even install on a WindowsMe notebook with 32MB of RAM (says it requires 128MB and closes)
- The full featured driver requires the printer to be locally connected to install successfully. And if you do that, you can't later tell it that the printer is remote (plus it installs about 600MB of garbage)

Since attaching the printer to the linux box and using PS drivers was not practical (because it has to be used as a scanner too...whatever), I thought... wait a second. Why not do that **on windows**?

And what the hell, it worked.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "this".
```

If you have a rebel can't be networked printer, do [this](#) and be sorta happy. It explains how to create a Ghostscript-based virtual printer that you can share.

And of course, the client PS drivers works just fine on any OS. Which means you can use printers that, for example, have no Linux drivers. Or no Windows9x/Me/NT drivers.

Only problem really is that it's a bit sluggish (it may wait a minute or two before it prints) because the XP box is slow... but it beats copying files.

```
system-message
WARNING/2 in everything.txt, line 7426
Duplicate explicit target name: "this".
```

So, I am trying to use this ancient card with Linux and all I get when I plug it is a message in the logs:

```
Aug  3 13:11:28 monty cardmgr[12752]: unsupported card in socket 0
Aug  3 13:11:28 monty cardmgr[12752]:  product info: "PCMCIA", "11M WLAN C
Aug  3 13:11:28 monty cardmgr[12752]:  manfid: 0x0274, 0x1601  function: 6
Aug  3 13:11:44 monty cardmgr[12752]: exiting
```

A little digging showed me the following:

- 1) This card is supposed to work with the hostap driver
- 2) It has never been a popular card.
- 3) Noone seems to have used the thing on Linux. Or if they did, they never documented it.

So... all you have to do is inform the PCMCIA thing in Linux that this card works with hostap.

After you installed hostap correctly for your kernel, edit `/etc/pcmcia/hostap_cs.conf` and add this:

```
card "TrendNet TEW PC16"
    manfid 0x0274, 0x1601
    bind "hostap_cs"
```

Reboot (or restart the pcmcia service, unplug, plug, etc), and that's it.

For some reason, there seems to be an image, somewhere in this site, that gets a bazillion hits from google's image search.

I have no idea what image it is, or why it is sought.

Since statcounter truncates the referrers, and it seems to be on many many pages... it's just weird.

For some reason, a domain in some language I don't speak (njoyschool.net) seems to be involved...

Here's some example referrers:

```
images.google.at/imgres?imgurl=http://www.njoyschool.net/data-new/
club/bbs033/sorry.gif&imgrefurl=http://lateral.netmanagers.com.ar/
lateral/sto
```

```
images.google.dk/imgres?imgurl=http://www.njoyschool.net/data-new/
club/bbs033/sorry.gif&imgrefurl=http://lateral.netmanagers.com.ar/
lateral/sto
```

```
images.google.co.ma/imgres?imgurl=http://www.njoyschool.net/data-new/
club/bbs033/sorry.gif&imgrefurl=http://lateral.netmanagers.com.ar/lateral/
```

Anyone has any idea what the heck those are? They are almost all I got in my referrers log!

If you are installing CIPE on Red Hat (and possibly other distributions), you need to remove the cipcb kernel module that comes with the distro's standard kernel, since it will mismatch your new CIPE's version.

I am playing with CMake. Specifically, I am trying to replace the simplistic hand-made Makefiles for my RA-Plugins project.

The parts about detecting libraries and conditionally compiling plugins based on what you have were surprisingly easy!

Until I ran into ... man pages.

Here is a Makefile that would build all the manpages for the plugins:

```
MANPAGES=plugins-ra.8 authchecks.8 rcptchecks.8
man: \$(MANPAGES)
%.8: %.man.txt
    txt2man -t \${basename $< .man.txt} < $< > $@
```

As you can see... trivial. All I need to do in order to add a man page is add whatever.8 to the list, and it will be created from whatever.man.txt.

But... how does one do that using CMake?

I started thinking of using a FILE (GLOB *.man.txt) and then a FOREACH over that, and then... then what? Really, I am stumped. I am a newbie, though, and getting the big, difficult stuff done is enough to switch. I should generate these before distribution anyway.

system-message

WARNING/2 in everything.txt, line 7509

[Inline emphasis start-string without end-string.](#)

So, I wrote a wee Makefile.manpages and added this for CMAKE:

```
ADD_CUSTOM_TARGET ( manpages make -f Makefile.manpages )
```

But I am curious about finding the *cmakeish* way.

Looking for a simple way to lookup the MXs for a given domain, I ran into libd-jbdns. Which has a delightfully simple API.

Of course it has its wrinkles.

Consider the MX lookup interface:

```
dns_mx (&out, &fqdn);
```

Where out and fqdn are strallocs (a sort of string that can contain 0).

But just try parsing out! Here's the explanation:

Each MX record is a two-byte MX distance followed by a 0-terminated dot-encoded domain name. If the domain does not exist in DNS, or has no MX records, out will be empty.

Because almost noone uses MX distances higher than 256, sadly, this usually looks like this (sorry for the notation ;-)

```
\0 10 f i r s t . m x \0 \0 20 s e c o n d . m x
```

So, you can not split on the NULLs because there are NULLs in the distances.

Which is not hard to parse, but is definitely more annoying than it should be. How hard would it be to return an array of structs?

The only current software I wrote that some people actually use is called RA-Plugins. It's a series of proggyes you plug in your SMTP server's conversation, and

do diverse things with it, like rejecting messages that fail certain criteria, checking the status of the recipient's account, whatever.

Injecting this stuff in the middle of SMTP is **good** because it means you will reject the messages before they get into your server. But... I am running out of ideas, so... have any you can spare? :-)

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "here".
```

You can see the current plugin list [here](#) and the only ideas I have left are:

- A plugin that calls back to the sender's email server and tries to email him, ala [milter-sender](#) to catch forged senders.
- A plugin to autowhitelist in spamassassin those addresses to which you send mail.
- A plugin to keep an account of how many connections you hold to each IP, and limit them. (Not currently possible)

If you understand what I wrote, and have any ideas... feel free to post them as comments and/or email me with them!

```
system-message
WARNING/2 in everything.txt, line 7567
Duplicate explicit target name: "here".
```

In the last week or so I have been reorganizing my workspace. Or rather, Rosario's workspace, and that meant moving some furniture my way.

One of the things I got is a long desktop, with room for two PCs (it was used in the classroom in Conectiva, and I got it as part of my severance payment).

So, I put my desktop on it, a keyboard and mouse of the tray below, and my home server on the other side of it.

That home server has been headless for a loooong time.

But since I have room in the table, I hooked a monitor, keyboard, mouse set to it, and thought "maybe it will work as a guest machine", for when someone comes to work or stay at my place.

Then I thought, there must be some way to use one keyboard and mouse to drive both machines, and started looking for KVM switches and stuff.

Then I thought... maybe there is a way to do it by software... and yeah! Enter [synergy](#) ...

You run a server on the box with the keyboard, and a client on the others. Explain the relationship between them in the simple config file... that's it.

Now I can slide the mouse off a screen to the other computer's, and you even share a clipboard!

Really, really, really nice stuff. Works flawlessly.

And since I am a bit weird in my ways, now... how can I make this work as a xinerama solution, where all the apps are in my notebook, and the other is just a display server?

Right now, I am starting a full screen FreeNX session pointed back at me, so everything is on the same box. But I can't move windows between both monitors.

I think with some creative thinking this **can** work. X app display migration...

I decided to try Arch Linux on my notebook. The reasons don't matter.

However, there was this problem about the CD drive being broken, and the lack of a floppy drive.

So, how did I do it?

I had a partition I could destroy (/dev/hda3)

I installed qemu.

Then, I started qemu using the whole HD and a CD ISO image (booting from the CD!):

```
qemu -hda /dev/hda -cdrom arch-0.7.1.iso -boot d
```

Then, **very carefully** I installed it on hda3 and did **not** install GRUB.

Copied the kernel and initrd images to /boot on the other linux installation.

Edited grub.conf, adding the Arch entry...

And it worked.

Keep in mind that if you make a mistake, this will completely destroy all your data. But if you are careful, and have enough space, you can install your next Linux while you use your current one.

Now, is that cool or what? :-)

Looks like the programming tutorial was well received. About 5000 hits in 12 hours :-)

Today's ranings:

Lateral Opinion (link) 2355 SPE - Stani's Python Editor (link) 86

If it continues, Lateral Opinion will reach 50K hits today. I'll have a beer for that!

Rosario is pregnant. Yipee! :-)

She is now about 6 weeks pregnant, and all three of us are doing fine (but she's having **bad** morning nausea... should go away in another 6 weeks... but that doesn't seem to be any help ;-)

The first echography is scheduled for Sept. 15, expect pictures of the tiny bean-like alien invader.

It's the first grandkid on her side of her family, so it's being preemptively spoiled by all her/his/its aunts/grandparents/whatever. Our house is being prepared for a flooding of knitted outfits and redundant childware by strengthening the floorboards.

I am quite at a loss for words, and have not been useful for almost anything for the last two weeks since we found out... I am totally spastic with happiness.

On other news, I was just starting to work on preparing my studio... which will now be made the child's bedroom. Do I care? No! It's just an excuse to actually build the office in the roof I wanted all along (so I can work with my cats, too :-), so everybody wins.

No, do not use gdbm. Probably not bdb either.

TDB allows for concurrent **writes**. So, if you want to store some data in a file in your application/tool/command/whatever, and there is a chance that you need concurrent writes... **TDB** is really the only reasonable choice I know.

You can even have **distributed** TDBs.

I found out about **TDB** while writing a persistent cache for a little tool (a gmail plugin). Since by its nature it would have concurrent writes all the time, I was aiming for something like memcached, or its disk-based version... but **TDB** solves the problem nicely and effortlessly.

So, if you need a flat DB (a.k.a anonymous blobs with a single index, a.k.a a hash in a file)...

Traditionally, you graphed your system's status using [MRTG](#) but nowadays, there are much nicer tools, and I like [Munin](#).

For MRTG+qmail there are many scripts, but I couldn't find one I liked for Munin. Since munin is modular, it's easy to fix, of course.

First, get the excellent [qmrtd](#) which gives you great, quick, awesome multilog-crunching tools.

Then, check the cods, check the munin plugin docs, and it's pretty much a 20-line thingie. In shell.

I would post it, but my blogging tool **hates** shell code :-)

And you get your graphs. I am probably going to write a collection of these and publish it somewhere.

Since I found how to make the articles keep the nice links to the comment pages, I just `_HAD_` to fix all the previous articles so they do that.

I am sure there is some template somewhere that does that, but... well, I didn't find it.

But it was necessary, because I want that if someone runs into any of the old things in the future, he can find the comments about it.

Of course all the articles have more edits than comments right now, but that may change someday ;-)

At least in the sense that they are really full of people with bad intentions...

- The red stripe are connections blocked because the sender is listed in spamcop.
- The blue stripe is connections blocked because the sender is in NJABL.
- The darker green stripe is connections that are rate-limited because they look like a DOS or address spam (client connecting too quickly)

The top green line is SMTP connections.

The thin white stripe is real connections that we allow to send email.

Of course when you look at those...

- 42% are spam (yes, the filter is quite efficient).

So, about 1 in 5 connection attempts actually intends to send a real, good, valid email.

I am amazed this email thingamajig still works.

I have been on [Arch](#) for a while now.

So far, not bad. Not too good, either, but it is a fun distro.

I am thinking of using it as a base for my ever-vaporous personal small server distro.

You can see any progress I make by searching for my packages in [AUR](#) (search for submitter ralsina). AUR is really a nice thing to have for a hobbyist. You put your stuff there, people can find it, they can even try it, comment on it, vote on it... really a very nice community site for a very hobbyist distro.

I posted yesterday that I liked Arch but I called it “not too good”. So, Mark Kretschmann posted a comment asking what I didn’t like.

It’s not too much, but here it goes:

- 1) The upgrades sometimes are a bit painful (switching to udev was a bit hard).
- 2) The policy of deleting the package documentation is evil. Really.
- 3) The startup system is too simplistic. No default order of startup scripts means sometimes it takes trial and error to figure out what goes first. Hal or dbus? hwd?
- 4) The package selection (without unsupported) is somewhat skimpy (no perl-net-server? no perl-html-template?) but that’s probably my POV because I am a bit server-oriented.
- 5) Some basic packages make scary assumptions. For example, if you have a user with UID 89 when you install mysql server, weird things may happen. Same for UID 40 and named.

On the other hand, the good side (at least for an amateur like me) is a bazillion times bigger.

I wrote a short intro to syslog, in order to **stop using it**.

Learn about socklog, svlogd and other alternative logging mechanisms.

Because logs are your best friends.

What happens if you try not to use a shell in Linux?

And no, I don’t mean for interactive use. Many do that.

No... what happens if you try not to use a shell... **to boot your system?**

Read [more](#) to find out.

I have not posted in a long time, not because nothing happened, but because too much happened.

So, here is an update...

Baby

The baby is doing great. The first ultrasound was on Sept. 15th, and here he is:

Birthday

That was on Sept. 15th because Rosario wanted it to be my 35th birthday present.

So happy birthday to me.

Work

The little company I am starting up is doing great

Hobby

system-message

```
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “packaged”.
```

I have decided that I have almost enough [packaged](#) for Arch that it's starting to make sense to master some ISOs. Basically: a somewhat-DJB-way-oriented linux distro.

- Boots using runit
- Maybe someday will use LUA instead of sh for startup scripts
- All services should be managed
- Will introduce genetic diversity to the ecosystem

That means: no sendmail, no postfix, no BIND, no Apache (at least for basic stuff), no many other "leading" programs. That is intentional. If everyone uses BIND, a BIND failure is catastrophic.

If I have learned anything from Outlook/IE it's that having everyone use the same thing is comfortable, but troublesome.

system-message

WARNING/2 in `everything.txt`, line 7810
Duplicate explicit target name: "packaged".

This chaco stuff seems pretty powerful. But it's tkinter/wx based :-P
I think I will take a look at their Kiva toolkit.

I have been procrastinating about creating my own Linux distro for at least three years. Guess what? I will still procrastinate about it for a few more, but that doesn't mean I can't write about how it's supposed to work ;-)

So, here is a first piece of the puzzle...

What do I mean by "Main in not-mail-servers"?

If by mail server we mean a box that has the responsibility to handle sending mail for users, non-mail-servers are all the rest.

And what is it they do with mail? They generate it. Both the users and the processes of those boxes generate mail. They do it for cron jobs, they do it for maintenance processes, they do it for alerts, whatever.

And what is it they do with that email? They send it somewhere.

Usually, they send it to themselves. Which is a pretty useless thing.

Go now and check the root mailbox in your computers. I bet most of you have a bunch of mails in them you never checked. Either it's important, in which case you should have placed it in a mailbox you actually read, or it's not, in which case it's useless to store.

In any case, **it shouldn't be there**.

How does your box send those mails? Using either the sendmail binary, or the mail program (probably mailx), which uses the sendmail binary.

Just because it's called sendmail it doesn't mean it **is** sendmail, of course. Postfix and qmail provide a *sendmail wrapper* to inject mail into their queues.

But the main problem is that using those means you need to have a well configured mail server in every box, even if they are **not mail servers**! Yes, your distro gives you a decent configuration by default which makes things usually work... for local mail delivery at least. Which is probably not really what you want.

Enter [nullmailer](#). A sort of heavily sedated, neutered qmail.

Configuration:

- Default domain name of outgoing mail in `/etc/nullmailer/me`
- List of SMTP servers in `/etc/nullmailer/remotes`:

```
mx1.mydomain.com smtp --user=ralsina --pass=notmyrealpass
```

You can put several, it will try them in order.

And that's that. A tiny service, which uses no TCP ports. The whole thing is 59KB (or less if you use diet libc), has one SUID binary (but it is **not** SUID root), two config files (both one-liners), no need for aliasing the system users.... and you can remove postfix/sendmail/qmail from most of your servers.

Sounds like a good idea to me.

I reorganized the partitions in my notebook. Since it has two different Linuxes, and I didn't touch anything in one of those, I thought I was pretty safe.

Well... I wasn't, because I moved my `/boot`, and forgot to change my GRUB configuration, so I ended with a non-bootable box.

And not just regular non-bootable:

- GRUB gave error 17 before the menu
- The DVD reader is broken
- My external USB HD is on loan on a client
- My Sony-discman/PCMCIA-SCSI-external-CD-reader needs 110v power (and I have no transformer around)
- My original Libretto PCMCIA FD works, at least to boot... but I have no other floppy drive to master a diskette.

Yes, I get into this kind of thing more often than most. I buy second hand hardware. I am not a first-worlder, guys.

So... what do you do in such situation? You go with PXE booting.

First I tried the only thing I knew that worked with PXE, LTSP (Linux Terminal Server Project).

So, two hours, a DHCP/TFTP/NFS config later, it's working.

It worked well enough (although it's quite a bit of work) but... I got no way to access the HD, because the only screen is a remote X display!

I am sure with a little elbow grease I could make it give me a local login with decent rescue tools, but it's not really obvious.

Then I googled for "linux PXE distribution" and found [RIP](#). Specifically the RIP-PXE version.

It is meant for recovery, so it's not surprising that it does a better job than LTSP, which isn't.

One nice thing is that it loads its root FS into RAM from TFTP, so no need for NFSROOT (or NFS at all, really). It works with only DHCP and TFTP. Quite trivial to do... but not documented. However, starting with the LTSP docs about PXE booting, it was pretty obvious :-)

This is what you need to put in your DHCP config (in the subnet chunk):

```
if substring (option vendor-class-identifier, 0, 9) = "PXEClient" {
    filename "/pxelinux.0";
```

```
}
else{
    filename "/kernel";
}
```

And then dump RIP-PXE into your /tftpboot (or /var/tftpboot, or whatever).
As long as you can do this with no errors, you are golden:

```
bash-3.1$ tftp localhost
tftp> get /pxelinux.0
tftp> get /kernel
tftp>
```

On some tftp servers, you need to do “get /tftpboot/pxelinux.0” or similar. Adjust paths accordingly.

So, it boots. You login. You fix your menu.lst/grub.conf/fstab/whatever, and you are out.

Beautiful!

Ever needed a process monitor that runs in a terminal? Have you been using top? Use [Htop](#) instead. Much, *much*, **much** nicer!

One of the things people study when they “learn unix” is shell scripting and usage. Because every system has a shell, and if you learn to use it interactively, you are half way there to automating system tasks!

Let’s consider that for a moment... what are the odds that the same language can be good for interactive use and for programming? I say slim.

Not to mention that learning shell as a way to learn unix is like going to a school that teaches TV production, and studying the remote. While useful, not really the important tool (ok, that analogy doesn’t work at all. But it sounds neat, doesn’t it?).

The first thing is that today’s Linux domination of the unixsphere has caused a serious monoculture in shell scripting: everyone uses bash. The more enlightened ones **may** check that their scripts work on some other Bourne-style shell.

There are no important distributions (or proprietary unices) that use a csh or anything like it. Debian has a policy that things should work without bashisms. That’s about as good as it gets.

Writing a dozen pages on how shell sucks would be trivial. But uninteresting.

So, let’s think it over, and start from the top.

What should a shell scripting language be like?

What doesn’t matter?

Let’s tackle these things. I invite anyone to add extra ideas in the comments section.

What should a shell scripting language be like?

- Interpreted (obvious)
- Dynamic typing (you will be switching ints to strs and viceversa all the time).
- Easy incorporation of other programs as functions/methods/whatever.

That pretty much is what makes it a shell. Is should be indistinguishable from something written using the shell itself.

- Pipes. This is a **must**. Unix has a bazillion tools meant to be used in command pipelines. You can implement a RDBMS using that kind of thing (look for nosql). Leverage that.

But even here, on its strength, the shell is not perfect. Why can't I easily pipe stderr and stdout to different processes? Why can't I pipe the same thing to two processes at the same time (yes, I know how to do it with a neat trick ;-)

- Globbing. *.txt should give you a list of files. This is one of the obvious things where sh is broken. *.txt may be a string or a list, depending on context... and a list is just a series of strings with blanks. That is one of the bazillion things that makes writing shell scripts (at least good ones) hard:

system-message

WARNING/2 in `everything.txt`, line 7989
[Inline emphasis start-string without end-string.](#)

system-message

WARNING/2 in `everything.txt`, line 7989
[Inline emphasis start-string without end-string.](#)

```
[ralsina@monty ralsina]\$ echo *out
a.out
[ralsina@monty ralsina]\$ echo *outa
outa
```

- A list data type. No, writing strings separated with spaces is **notok**. Maybe a python-style dictionary as well?
- Functions (obvious)
- Libraries (and ok, the shell source mechanism seems good enough)
- Standalone. It shouldn't spawn sh for any reason ;-)

What doesn't matter?

- Performance. Ok, it matters that a five-liner doesn't take 50 minutes unless it has to. But 1 seconds or two seconds? not that important.
- Object orientation. I don't see it being too useful. Shell scripts are old-fashioned :-)
- Compatibility to current shells. Come on. Why be like something that sucks? ;-)

Now, the example

Let's consider a typical piece of shell script and a rewrite in a more reasonable syntax.

This is bash (no it doesn't work on any other shell, I think):

```
DAEMONS=( syslog network cron )

# Start daemons
for daemon in "\${DAEMONS[@]}"; do
    if [ "\$daemon" = "\${daemon#!}" ]; then
        if [ "\$daemon" = "\${daemon#@}" ]; then
            /etc/rc.d/\$daemon start
        else
            stat_bkgd "Starting \${daemon:1}"
            (/etc/rc.d/\${daemon:1} start) &>/dev/null &
        fi
    fi
done
```

And since DAEMONS is something the admin writes, this script lets you shoot in the foot in half a dozen ways, too.

How about this:

```
DAEMONS=["syslog", "network", "cron"]

# Start daemons
for daemon in DAEMONS {
    if ( daemon[0] != "!" ) {
        if ( daemon[0] == "@" ) {
            stat_bkgd ("Starting "+daemon[1:])
            /etc/rc.d/+daemon[1:] ("start") &> /dev/null &
        } else {
            /etc/rc.d/+daemon ("start")
        }
    }
}
```

Of course the syntax is something I just made up as I was writing, but isn't it nicer already?

For a few months I have been using an unmanaged virtual private server from [Tektonic](#), and I love it.

What's that? Let's take it one word at a time, and then some more.

- 1) It's a **server**: which means it's a full-ish linux installation. So it is capable of doing lots of things. I can run all sorts of weird python thingies in it if I want. IMAPS and SSMTP? No problemo.
- 2) It's **private**: which means I am root on it. I have the shell. I choose what to install.
- 3) It's **virtual**: it's a Virtuozzo partition in a real server. That means no custom kernel modules, and that since almost everything is shared with other instances, 5GB of disk and 128MB of RAM go a long way.

- 4) It's **unmanaged**: which means I manage it. Which is just the way I prefer it, since that's my job.
- 5) It's **cheap**. I started on a 8 dollars a month plan (which doesn't seem to be there anymore, the current cheapest is a 15 dollars plan).
- 6) It's a **throwaway**. I want to host some client as a favour? I just put it there. I could even rent another of these servers for a while, use it, then close it. Backups? Clicking on a webpage saves the image! Other than that... I back it.
- 7) **Fixed IPs**. All you want (for extra coins).
- 8) A **home** away from home. All my stuff is there. I need it, I get it. Without bothering about having my own server at home via no-ip or somesuch (which of course I still have too ;-)
- 9) It **works**. It hardly ever breaks. And having survived expensive, **managed** servers, this baby is working just as well.
- 10) It's a nice **gift**. Suppose you have a connection to a free software project/LUG/family/whatever, and they need a place on the internet. Why not sponsor them with something like this? I offered one to PyAr (which didn't take it, but it's the thought that counts ;-)
- 11) The ultimate **learning** experience: you can restore the system in 2 minutes. Want to play/learn sysadmining? Do it on the real virtual thing! Much cheaper than hosing your own box ;-)
- 12) They offer a good **service**. So, people should know about it. And of course... if you know a similar, but even better deal... I'm all ears!

I had an emergency. The CPU usage of a certain mail server was raising, and the culprit was clamd.

For some reason, in the last few months, the CPU usage of clamd kept rising, and was now near 70% **average** of the server's CPU.

Removing the antivirus is, of course, not an option. On the other hand, performance was starting to suffer.

The usual response would be a full retooling of the setup, multiple SMTP servers handling the load against a central storage server, clamav running on each SMTP... but switching to that involves a full reimplementatation of the system. Because of the antivirus??? Hell no.

So, I started investigating how I could move clamd to another box, like I did with spamassassin. It was not pretty.

- clamav has a protocol defined for connecting to remote servers.
- clamav doesn't have a client for it.
- clamd-stream-client doesn't seem to work.

So, I thought... let's be original. What do I actually need?

I need to be able to call clamscan, and have it scan the current folder. Based on its exit status code (0/1/2) the mail is accepted, rejected, temporarily rejected.

Having the same folders structure available to two boxes is trivial. I have NFS, lots of bandwidth and another computer.

Running clamdscan in the second box, scanning those folders is trivial too.

The missing piece is a way to tell the second box's clamd to scan, and get the exit code in the mail server.

Enter netpipes!

Netpipes is software to “make TCP sockets usable from the shell”. You can find it at <http://web.purplefrog.com/~thoth/netpipes/netpipes.html>.

And here's a replacement clamdscan which works the way I wanted it:

```
#!/bin/dash
exit `echo \$PWD | hose 192.168.1.53 9000 --slave `
```

This version takes the folder you want to scan as an argument:

```
#!/bin/dash
exit `echo \$* | hose 192.168.1.53 9000 --slave `
```

And here is the “server side”. First netclam.sh:

```
#!/bin/dash -x
read args
/usr/bin/clamdscan \$args >/dev/null 2>&1
echo \$?
```

Then the “network code”:

```
faucet 9000 --in --out /usr/bin/netclam.sh
```

And there you have it. ClamAV moved to another server. With 5 lines of shell code.

I read Zack Rusin's blog about benchmarking vector graphic APIs... then I see a comment mentioning [Antigrain](#). Then I check the antigrain examples, and they are gorgeous, **and pretty fast!** Even on a lame Sis630!

Then it hit me... I am never going to do anything with it (or with Qt's Arthur). Maybe I am getting old, but I see a swirl of cool software... [dparser](#)... [asymptote](#)... [txt2tags](#) ... (and those are only the ones I saw in the last week).

All of them are about something that interests me, but I simply can't do anything. I mean, would it be cool to write a vector-app-for-kids with antigrain (or Arthur?) Sure!

Would I like to implement this shell-style language I have floating in my head for a year using dparser (or pyparsing?) Yeah! Would I like to hack a Trac plugin using txt2tags (or restructured text?) Sure!

But when can I do that? I have my business, my wife, her pregnancy, my other projects... maybe that's what happens when you become old. You gather enough baggage that you can't lift any more backpacks in your trek.

But what can I do with all the ideas swirling in my head? Really! What?

Bought an [Eusso](#) (No, I had never heard of them either) Wifi USB dongle.

Why?

- It says “linux driver” on the blister
- It's the cheapest 802.11g thing on the local ebay-like place
- My ancient pcmcia 802.11b card sucks.

I am thinking of buying half a dozen more and getting rid of all the cables for all my boxes, all of Rosario's office and the guest computers (yes, I do have guest computers. They are there so my guests have their own computers :-).

Plugged it and it worked (ok, I had to install the zd1211 driver which took me 40 seconds). Only problem: it's hot. **HOT**.

So, need a USB/WiFi thingie that works well in Linux? You can do worse than this baby.

Yesterday I wrote that I have too many ideas. Ok, here's another one:

A word processor for writers. And when I say writers, I mean novelists, technical book writers, script writers, playwrights...

Word is not very good for a writer. OpenOffice is not good. KWord is probably worse (because of the emphasis on page layout). LyX is probably as good as it gets, and it's not exactly perfect.

A writer actually needs a simple-ish word processor with a bunch of ancillary gadgetry.

For example:

- Statistics:
 - How many words/chars/pages a day is he writing
 - A **live** word/char counter
 - A live word frequency monitor (put the cursor on a word and see how often it's used)
 - Live counter of document/chapter/section/scene size.
- Outlining
 - Real live outlining. The kind where you drag stuff around and the text follows.
 - An editable full-text outline view
- Collaboration
 - Multiple editors
 - Versioning control
- Projects
 - Multiple files per project
 - Linking files to places on the text in other files
- Index cards
 - Associating index cards to places on the text
 - Grouping index cards (for example, per character, or per location)
 - Placing them on a timeline or a storyboard
- Live Thesaurus / Dictionary
 - Show definitions and alternatives as the pointer crosses a word.
 - One click replacement

- Styling
 - Per fragment/paragraph styles
 - User defined
 - Predefined styles

There are a bazillion things he does not need, though, like detailed page layouting, or grammar checking.

It would be nice if it could later be easily imported (styled!) into something like Scribus so a decent page layout could be done, but it doesn't need to be in the same app at all.

The text engines in Qt4 are good enough for all this app needs graphically.

RestructuredText is good enough to provide a backend, a parser, an exporter, a reader, a transformer, whatever.

So there it is, another idea I will most likely not implement. Someone please run with it, you can probably make it a rather expensive GPL shareware on Mac ;-)

Last night I saw an "investigative news" program on the TV. It's called "Informe Central", and their headline story was about an abandoned factory in San telmo (where tourists go to see typical BA and locals go to see tourists).

The thing is, that factory has been taken over by poor people who live there. It's conveniently located, and they don't pay anything.

On the other hand, it's a nest of drug, rape, poverty and violence, but that's not the only thing these "journalists" said.

They said they lived in inhumane conditions, up to 2.6 persons per square meter.

They also said about 300 people live there, which would mean there are roughly 115 square meters in the factory.

Which is, actually, closer to 1200 square meters. or maybe 5000. But they kept on saying those numbers.

Do you know that in order to have 2.6 persons per square meter so that each of them has a small (double) bunk bed, you would have to put the bunk beds one next to the other with 20cm-wide spaces in between?

How the **hell** did they get that number? Is that a sign of their regular investigative quality? Probably.

... get one of these?

I mean, I *do* have the money. But noone is going to sell them here. Any kind reader willing to buy one and mail it to me?

I am a great fan of Restructured Text. I write my blog using it. I write my business proposals using it, I write my documentation using it, I think you should write almsot everything you write now using it. I have even blogged many times about it.

RST is a minimal markup language. You can figure it out in a couple of hours, and then use it to produce pretty HTML pages, PDF docs, man pages, LaTeX documents, S5 slides, and other things.

Plus, the source works as a plain text version, and is very readable:

```
This is a title
=====
```

```
Some text in a paragraph
```

```
A subtitle
```

A list

More items

1. A numbered sublist
2. Another item
 - a) A sub-sub-list
 - b) With more items

A table	With two columns
And Two	rows

See? Nice.

RST has another great thing that is not so well known: there is a parser for it, which turns the document into a tree of nodes representing different parts of the document.

You can manipulate this node tree, modifying the document, and **then** generate the output.

But there is no way, right now, to generate RST from the tree. Which means it's a one way road.

Well, I am hacking to fix that.

Right now, I handle titles, sections, all sorts of lists, transitions, quotes, emphasis, italics, and a few other elements.

The only ones that seem difficult to implement are tables, but I still think I can do it. Although the produced RST doesn't look the same as the original, it is functionally identical.

How do I test if it works? With a test suite. If it works, it should be invariant this way:

RSTsample -> rst2html produces the exact same output as RSTsample -> rst2rst -> rst2html

If anyone wants a copy, email me.

What is it? A program that takes a docutils document tree (parsed from a RST document or programatically generated) then dumps as close as I can guess to reasonable RST back.

This lets Restructured Text be a saveable data format, which is nice.

It's not done as a docutils writer. Sorry, I couldn't make that work.

What works? Most of it.

What doesn't? A dozen directives, custom interpreted text roles, and **tables**.

Yes, all of those are important. But the rest seems to work ok!

Look: a 804 line RST document containing almost every feature of the language, and the only difference in the generated HTML output between the original and rst2rst's is an invisible difference in continuation lines in line blocks.

```

[ralsina@monty wp]$ python rst2rst.py t1.txt > t2.txt
[ralsina@monty wp]$ /usr/bin/rst2html.py t1.txt t1.html ; /usr/bin/rst2html.py t2.txt t2.html
[ralsina@monty wp]$ diff t1.html t2.html
468,469c468,469
< <div class="line">But I'm expecting a postal order and I can pay you back
< as soon as it comes.</div>
---
> <div class="line">But I'm expecting a postal order and I can pay you back
> <div class="line">as soon as it comes.</div>
[ralsina@monty wp]$ wc -l t1.txt
804 t1.txt

```

You can get [rst2rst.py](#) and the [testfile](#).

Anyone knows of a real docutils test suite I could borrow?

rst2rst was going along just fine, but I have run into the real problem: TABLES.

To refresh your memory (if you know) or let you know anyway, RST is an ascii markup language. And it supports tables. Like this example:

```

+-----+-----+
|         |         |         |
+-----+-----+-----+
|         |         |         |
+-----+-----+         +
|         |         |         |
+-----+-----+-----+

```

The parser gives me a tree structure:

row (entry, entry with morecols=1) row (entry,entry,entry with morerows=1) row (entry with morecols=1)

Now, each cell has a minimum size set by its contents.

For example, a cell containing "Hello" has a minimum size of 5x1 (wxh).

It is a good idea to surround the contents with a blank row/column, so make that 7x3. I can figure this out from the contents.

And here's the trick question... anyone knows an algorithm to do this that is not incredibly difficult? Or, if it is incredibly difficult, care to help me? ;-)

Como todos los años, voy a dar una charla en CafeConf 2006, el tema de este año es "Booteando más rápido y mejor usando runit".

Si buscan runit en este blog se van a enterar mas o menos de las mismas cosas, pero la charla debería ser más prolija :-)

La primera persona que mencione el nombre de este blog en la parte de preguntas y respuestas se gana un regalo sorpresa!

Like every year, I will be speaking at CafeConf, Sat. Nov 11. This year's topic is "Botting faster and better using runit".

If you look for runit in this blog, you will find out more or less the same things, but this should be more organized.

The first guy to mention the blog in the Q&A session gets a surprise gift.

My Restructured Text to Restructured Text converter can now handle tables!

If you don't understand a word of what I just wrote... well, skip this. It's really quite esoteric.

I managed without linear programming by the usual route: make it ugly but functional. Basically, I overdimension things badly.

Here's the source table:

```
+-----+-----+
| f2,c1 | f2,c2 longer longer still |
|       +-----+
| f3,c1 | 1. a                       |
| f3,c1 |                             |
| f3,c1 | 2. b                       |
+-----+-----+
```

Here's what rst2rst generates:

```
+-----+-----+
| f2,c1 | f2,c2 longer longer still |
|       |                             |
| f3,c1 |                             |
| f3,c1 |                             |
| f3,c1 |                             |
|       +-----+
|       | 1. a                       |
|       |                             |
|       | 2. b                       |
+-----+-----+
```

Seriously non-optimal, but it is functional, until a braver soul improves the algorithm.

It can't handle table headers correctly, but other than that, it works just fine.

Update: It now supports all the RST table syntax, AFAIK.

Version 0.2.9 of ra-plugins, my qmail-spp plugin collection is coming soon.

Including lots and lots of new plugins, a real build system, and even two patches by someone else :-)

So, now is a good time to let me know if you are using ra-plugins, if you have any problems with it, and if you have any ideas for cool plugins. I can write them.

One of [these](#) I mean.

Slightly old, but it can get a nice Qtopia setup on the CF card. And, with a PCCard->USB gadget, I expect it to be useful (wifi/external disk/whatever).

Used. Cheap. Should replace my faithful Clie for ebook reading, and my heavy notebook for meetings. More if I can get a NX client in it :-)

... who own a HP Jornada 720 **and** are using Opie on it **and** they have the spanish/latin-american keyboard... here is your [keymap](#).

I will write something about how to get Linux going right on it soon, but here's the status report, 48 hours in.

This baby (unnamed yet) has:

- 32MB of RAM

- 1 GB of Flash
- Wifi (802.11b pcmcia) + IRDA + Ethernet (pcmcia) + Anything once I find a 16-bit pcmcia-USB card (anyone has a spare and wants to **recycle** it? ;-)
- Decent battery life (6 hours use **with wifi**, 9 without)
- A keyboard
- A decent screen (640x240)
- A decent Linux-based GUI (Opie)
- A somewhat erratic touchscreen

So, what can I do with it:

- Email
- Web browsing (With Konqueror goodness)
- Programming (Python, even PyQt2!). The keyboard and screen are surprisingly decent.
- eBook reading. This is the most important one. In my work, I spend a lot of time waiting. Waiting for the train to arrive, for the trip to end, for someone to come to a meeting, for the waiter to bring my meal, for stuff to compile, for stuff to download... maybe I wait 3 hours a day. So I read. And this screen (long and somewhat thin) is quite spectacular for reading. Opie-reader is pretty good.
- MP3 and Video player (haven't used it yet). I have streaming TV at home, courtesy of CherryTV (check the links at the left). This should work great when Rosario wants to see Montecristo and I'd rather see Penn & Teller's show.
- General PIM stuff. Although I tend to keep that stuff in my head and my phone.

The bad side:

- The bizarre screen aspect ratio confuses many configuration dialogs.
- Almost no game works unless you rotate the screen.
- The keyboard configuration took a while, and is not perfect yet (I can't make dead_acute work for some reason)
- The extra buttons don't work (external audio recorder, and alarm light-button)
- I can't find a way to bind the function keys to apps in Opie
- The reset button doesn't work (it's now a **hang** button)
- Suspend is not really suspend on Linux (for unavoidable hardware reasons), so it spends battery when suspended (may last 12 hours or so, I think).
- The only way to really turn it off is to take out the battery (not as bad as it sounds).
- If you do that, it takes about one minute to boot.

So, I am using it more as a laptop (although a really, really small one, with very, very good battery life ;-) than as a PDA.

The small memory and CPU means I can't run very demanding stuff, but I never seem to do that, anyway.

And of course, the really bad thing: it's so much fun to hack with, I have trouble working!

All in all, a great toy, lots of fun, and rather useful.

A few people have asked me for the code. Ok, here it goes.

First, get qmrtg.

Then you need this: [qmunin.tar.bz2](#)

Then build and install it. You may need to modify the sources, depending on just how your qmail works. There are some **example** munin plugins included (in spanish, I am not translating them ;-), which 99% surely will not work for you, but they are simple shell scripts, so you should be able to hack them.

Creating a real release of this is just useless, because everyone's qmail logs look different, so take it and hack it.

And that's it.

UPDATE: There is *another* *better* test suite It is in YAML, though, so I need to parse it before I can use it, but that's my problem.

It's having a test suite that makes no sense.

I have written, for my ra-plugins project (you don't have to know what it is for this post anyway) a piece of code that tries to check mail senders using SPF.

SPF is an open standard. It has standard implementations. It has a test suite (<http://www.schlitt.net/spf/tests/>).

The test suite says this: spfquery -ip=192.0.2.1 -sender=05.spf1-test.mailzone.com -helo=05.spf1-test.mailzone.com result /*/ fail smtp-comment /*/ explanation header-comment /*/ spfquery: domain of 05.spf1-test.mailzone.com does not designate 192.0.2.1 as permitted sender received-spf /*/ Received-SPF: fail (spfquery: domain of 05.spf1-test.mailzone.com does not designate 192.0.2.1 as permitted sender) client-ip=192.0.2.1; [envelope-from=postmaster@05.spf1-test.mailzone.com](#); helo=05.spf1-test.mailzone.com;

So, yeah:

```
$ spfquery -ip=192.0.2.1 -sender=05.spf1-test.mailzone.com -helo=05.spf1-test.mailzone.com
fail
Please see http://www.openspf.org/why.html?sender=05.spf1-test.mailzone.com
spfquery: domain of 05.spf1-test.mailzone.com does not designate 192.0.2.1
Received-SPF: fail (spfquery: domain of 05.spf1-test.mailzone.com does not
192.0.2.1 as permitted sender) client-ip=192.0.2.1;
envelope-from=05.spf1-test.mailzone.com; helo=05.spf1-test.mailzone.com;
```

So, the standard implementation does what the test suite says.

Too bad that, if you bother checking the URL you are told to "please see"...

The domain 05.spf1-test.mailzone.com has published an SPF policy, however the policy is neutral on whether 192.0.2.1 is authorized to send mail on its behalf.

Either both the test suite and the sample implementation are wrong, or the site is wrong. And I am leaning towards "the test suite is wrong", because...

```
$ host -t txt 05.spf1-test.mailzone.com
05.spf1-test.mailzone.com descriptive text "v=spf1 default=deny"
```

If you check the record syntax (http://www.openspf.org/SPF_Record_Syntax) default is an unknown modifier, and should be ignored, so the record is simply “v=spf1”, and indeed the result is neutral and there is no reason why this should be a fail.

Lots of people, when they heard of this merger, started thinking like this:

Ok, so Novell bought Ximian. Ximian is a GNOME company. They also bought Suse. Suse is a distro. Ergo, Suse will become a Ximian-oriented distro.

Well, that makes some sense. But they seem to be forgetting of a change that should happen much sooner, because it:

- a) Is easier to do
- b) Makes more sense
- c) Costs less money

What change? Well, the changes that Ximian will have to face for now being part of a company that owns a dsitro!

Let’s look at Ximian’s products:

- Ximian connector: I don’t see this changing, except for they adding a groupwise connector.
- Evolution: I can see Novell pushing Evolution as a Groupwise client, for example. So, this has some legs. I don’t see it becoming a major revenue source for Novell, though, so it’s not going to get a large push.
- Gnumeric: I see an axe in its future. Novell is not in the spreadsheet business. They owned Quattro Pro once, didn’t they?
- Red Carpet. I have heard Novell really wanted this. They have ZenWorks, I suppose some integration is in order, and this will be productized. However, I don’t think Novell will want to push it as a software distribution mechanism for their competition, at least not while it’s free. If they did, Novell will be giving away what they bought. My guess? It will lose support for Red Hat, and will become more and more proprietary as time passes.
- Ximian setup tools: I am doubting between the axe and proprietary lockdown. Why would Novell want to support configuration tools for their competition? On the other hand, YaST is better for their own product, so why keep this at all?
- Mono: I can see they wanting this for their own tool development, and to make a push into the development tool market, but Novell has really no foothold there. So, while they will keep itr and push it, it’s not exactly a guaranteed success (then again, nothing is).
- Ximian Gnome: I have no idea.

Also, polishing the crystall ball a little, if Novell has some cashflow problems in the next year, they will focus harder in what they really care about.

My guess is setup tools will become SuSE-only or SuSE-first, Red Carpet will be absorbed or rebranded into ZenWorks or something, Mono will be marketed as .NET

platfor for Linux as soon as they get Forms working, the rest will be “liberated” fedora-style.

Now, let’s put a reminder to read this again in 2005 ;-)

Ok, the SPF implementation situation is kinda pathetic.

There seems to be exactly one maintained C implementation. And it’s windows-only.

- libspf’s website seems to have disappeared
- libspf2’s not RFC-compliant (verified for 1.2.5) and their issue reporting system bounces.

So, I have taken the most compliant one I found whose code I can actually follow (that would be the python one) and am reimplementing it in C (using bstrlib and libdjbdns).

It will probably not come to a good end, but hey, it **may** work ;-)

I am using ra-plugins as a toy to do things I never bothered in other projects.

I am doing unit-testing. And now... some literate programming!

Ok, not much, and not very well, but at least I am playing with [Lp4all](#) which is a nice, simple tool to generate nice HTML from slightly wiki-marked sources.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “here”.

You can see some little things in my code [here](#). My veredict so far? A nice way to keep the code documented in a fashion that ocasinal browsers can follow.

The main thing missing is automatic cross-referencing.

In general, I am finding that this (and unit testing) helps me express explicitly to myself what the heck I am trying to do, and see if the code actually does it. Which is a really good thing.

system-message

WARNING/2 in `everything.txt`, line 8690
Duplicate explicit target name: “here”.

If so, what is the C POSIX regex (you know regcomp & friends) equivalent of this python regular expression:

```
re.compile(r'^([a-z][a-z0-9_-\.\.]*)=', re.IGNORECASE)
```

Because it sure isn’t this:

```
regcomp(&re, "^([a-z][a-z0-9_-\.\.]*)=", REG_ICASE)
```

I have been playing with it for two hours and am bored :-)

I am porting pypf to C (long story, and I am stupid for trying). But of course, C is not python.

So you don’t have anything nearly as nice as `re.compile("whatever").split("somestring").`

What is that good for, you may ask? Well, to do things like splitting email addresses while validating them, or in this specific case, to validate SPF mechanisms (nevermind what those are).

But hey, you can always do this (excuse me while I weep a little):

```

struct bstrList *re_split(const char *string, const char *pattern)
{
    int status;
    regex_t re;
    regmatch_t pmatch[20];

    if (regcomp(&re, pattern, REG_ICASE|REG_EXTENDED) != 0)
    {
        return(0);        /* Report error. */
    }

    bstring tmp=bfromcstr("");
    char *ptr=(char *)string;

    for (;;)
    {
        status = regexec(&re, ptr, (size_t)20, pmatch, 0);
        if (status==REG_NOMATCH)
        {
            break;
        }
        bcatblk (tmp,ptr,pmatch[0].rm_so);
        bconchar (tmp,0);
        bcatblk (tmp,ptr+pmatch[0].rm_so,pmatch[0].rm_eo-pmatch[0].rm_so);
        bconchar (tmp,0);
        ptr=ptr+pmatch[0].rm_eo;

    }
    regfree(&re);
    bcatblk (tmp,ptr,strlen(string)-(ptr-string));
    struct bstrList *l= bsplit(tmp,0);
    return l;
}

```

And that is probably wrong for some cases (and it doesn't split the exact same way as Python, but that's what unit testing is for).

I **must** be missing something that makes regcomp & friends nicer to use. Right? Right?

I understand them now :-)

Apparently, the poster wants to use authentication on Squid against a LDAP server, and use LDAP groups to decide whether the user is allowed or denied.

Apparently, there is no way to do that with Squid itself. But it sounds like a simple thing to do using an external program.

Who knows, I may write it, too :-)

Other comments showed concern about this passing cleartext passwords. That is true. Simple modifications to the scripts should prevent the POP3 or IMAP access from doing that, but I know no way to make Squid use an encrypted link for the authentication (At least not without using Squid v3).

It now can do a bunch of things like expanding macros and (in some cases) validat-

ing mechanisms.

I am making very heavy use of unit testing, because it's a pretty complex piece and each function **needs** to do exactly the right thing or everything else fails (it's pretty hard to figure out where it will fail ;-)

You can check the 947 LOC thing at <http://code.google.com/p/raspf> (the Code tab). If you do check it, jeep in mind the following:

- It uses a few libs, and they are included in the source code for simplicity.
- I do sometimes commit code that doesn't compile
- I do sometimes commit code that fails tests
- You need cmake
- I am not giving a damn about memory management right now, so don't bother worrying about leaks: **everything** leaks in this code. I want to make it functional first, then I can plug it one function at a time (simply by running the unit testing code with a memory checker).

Enjoy (although it's not precisely enjoyable code right now ;-)

After many years, this is the first radical change of look.

It's not very nice, because my HTML skillz sux0rz, but hey, it's darker!

There are many little things wrong (like the color of visited links) but I like it.

It's also a bit simpler, and I did the banner using inkscape (I had it done much nicer using Karbon, but then I couldn't figure out how to do the gradient. Oh, well).

Update1: It looks incredibly awful on IE6, from the untransparent PNG to the unsupported overflow:auto I think I caught every damn thing that doesn't work :-)

Update2: Both IE and Konqueror will not use `overflow: auto` if the object is inside a table. In a blog that posts code and logs, and so on, that's a big problem.

So, to make this kinda work, I had to get rid of (almost) all tables.

It looks ok now, except on IE the sidebar is at the bottom right, which is probably because it calculates element widths differently.

As of right now, my customers owe me more than I billed in the second half of last year, and more comes due each month.

I suppose that's bad because I am sucking at collecting. On the other hand, it also means I am not sucking at billing. Or maybe yes, but much less than last year.

[RaSPF](#), my attempted port of [PySPF](#) to C is now at a very special point in its life:

The provided CLI application can check [SPF](#) records and tell you what you should do with them!

Here's an example:

```
[ralsina@monty build]$ ./raspfquery --ip=192.0.2.1 --sender=03.spf1-test.ma
Checking SPF with:

sender: 03.spf1-test.mailzone.com
helo:   03.spf1-test.mailzone.com
ip:    192.0.2.1

response:      softfail
```

```
code:          250
explanation:    domain owner discourages use of this host
```

Is that correct? Apparently yes!

```
[ralsina@monty pyspf-2.0.2]$ python spf.py 192.0.2.1 03.spfl-test.mailzone.
('softfail', 250, 'domain owner discourages use of this host')
```

Is it useful? Surely you jest!

There are still the following problems:

- The memory management is unexistent
- I need to hack a way to run the official SPF test suite so I can see how well it works and that it works exactly as PySPF
- It probably will segfault on many places
- I am changing the error handling to be exception-based, thanks to [EXCC](#)
- The IPv6 support is between iffy and not there
- There is no support for SPF (type 99) DNS records, only TXT records (need to hack the udns library)

But really, this should be about 60% of the work, and it does work for some cases, which is more than I really expected at the beginning.

Here's the whole source code of the sample application (except for CLI option processing):

Here are the results as of right now:

- Give the expected results: **82 tests**
- Give the wrong result: **48 tests**
- Give a correct but not preferred result (mostly because of SPF records and IPv6): **6 tests**
- Fail (crash): **9 tests**

So, depending on how you look at it, RASPF passes between 61% and 56% of the tests.

Not bad so far :-)

Update: As of 20:52 ART, it's 105/0/35/5 and 72-76%. The bad news is that that was all the low hanging fruit, and now it gets much harder.

Ok, not SCO, but Exo Training.

You see, I teach Linux courses for a couple of companies (steady job, bad pay, but steady, you meet people), and we use the Exo Training facilities.

Well, one of the guys there contacted me to see if I was capable of teaching the SCO courses.

I said no, of course, but hell, doesn't it surprise anyone that there are still such things as:

- SCO LINUX System Administration

- SCO LINUX Network Infrastructure
- SCO LINUX Network Services
- SCO LINUX Enterprise Administration
- SCO LINUX Enterprise Security

And these are offered right now!

Working on my SPF library, I ran into a problem. I needed to validate a specific element, and the python code is a little hairy (it splits based on a large regexp, and it's tricky to convert to C).

So, I asked, and was told, maybe you should start from the RFC's grammar.

Ok. I am not much into grammars and parsers, but what the heck. So I check it. It's a ABNF grammar.

So, I look for the obvious thing: a ABNF parser generator.

There are very few of those, and none of them seems very solid, which is scary, because almost all the RFC's define everything in terms of ABNF (except for some that do worse, and define in prose. Did you know there is no formal, verifiable definition of what an Ipv6 address looks like?).

So, after hours of googling...

Anyone knows a **good** ABNF parser generator? I am trying with [abnf2c](#) but it's not strict enough (I am getting a parser that doesn't work).

Anyone knows why those very important documents that rule how most of us make a living/work/have fun are so ... hazy?

In my previous post, I mentioned how PySPF does something using a regular expression which I couldn't easily reproduce in C.

So, I started looking at parser generators to use the original SPF RFC's grammar.

But that had its own problems.... and then came [ragel](#).

Ragel is a finite state machine compiler, and you can use it to generate simple parsers and validators.

The syntax is very simple, the results are powerful, and here's the main chunk of code that lets you parse a SPF domain-spec (it works, too!):

```

machine domain_spec;
name = ( alpha ( alpha | digit | '-' | '_' | '.' ) * );
macro_letter = 's' | 'l' | 'o' | 'd' | 'i' | 'p' | 'h' | 'c' | 'r' | 't';
transformers = digit* 'r'?;
delimiter = '.' | '-' | '+' | ',' | '|' | '_' | '=';
macro_expand = ( '%' { macro_letter transformers delimiter* } ) |
                '%%' | '%_' | '%-';
toplabel = ( alnum* alpha alnum* ) |
            ( alnum{1,} '-' ( alnum | '-' )* alnum );
domain_end = ( '.' toplabel '.'? ) | macro_expand;
macro_literal = 0x21 .. 0x24 | 0x26 .. 0x7E;
macro_string = ( macro_expand | macro_literal ) *;
domain_spec := macro_string domain_end 0 @ { res = 1; };

```

And in fact, it's simpler than the ABNF grammar used in the RFC:

```

name          = ALPHA *( ALPHA / DIGIT / "-" / "_" / "." )

```

```

macro-letter      = "s" / "l" / "o" / "d" / "i" / "p" / "h" /
                  "c" / "r" / "t"
transformers      = *DIGIT [ "r" ]
delimiter        = "." / "-" / "+" / "," / "/" / "_" / "="
macro-expand     = ( "%{" macro-letter transformers *delimiter "}" )
                  / "%%" / "%_" / "%-"
toplabel        = ( *alphanum ALPHA *alphanum ) /
                  ( 1*alphanum "-" *( alphanum / "-" ) alphanum )
domain-end       = ( "." toplevel [ "." ] ) / macro-expand
macro-literal    = %x21-24 / %x26-7E
macro-string     = *( macro-expand / macro-literal )
domain-spec      = macro-string domain-end

```

So, thumbs up for [ragel](#)!

Update:

- The code looks very bad on python or agregators.
- This piece of code alone fixed 20 test cases from the SPF suite, and now only 8 fail. Neat!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "pyspf".

[RaSPF](#), my C port of [PySPF](#), is pretty much functional right now.

Here's what I mean:

- It passes 75 internal unit tests (ok, 74 , but that one is arguable).
- It passes 137 of 145 tests of the [SPF](#) official test suite.
- It agrees with PySPF in 181 of the 183 cases of the [libspf2](#) live DNS suite.
- It segfaults in none of the 326 test cases.

So, while there are still some corner cases to debug, it's looking very good.

I even spent some time with valgrind to plug some leaks (the internal test suite runs almost leakless, the real app is a sieve ;-)

All in all, if I can spend a little while with it during the week, I should be able to make a release that actually works.

Then, I can rewrite my SPF plugin for qmail, which was what sent me in this month-log tangent.

As a language wars comparison:

- The sloccount of raspf is 2557 (or 2272 if we use the ragel grammar source instead of the generated file)
- The sloccount of PySPF is 993.

So, a 2.6:1 or 2.28:1 code ratio.

However, I used 4 non-standard C libraries: bstrlib, udns, and helpers for hashes and exceptions, which add another 5794 LOCs.

So, it could be argued as a 8:1 ratio, too, but my C code is probably verbose in extreme, and many C lines are not really “logic” but declarations and such.

Also, I did not write PySPF, so his code may be more concise, but I tried my best to copy the flow as much as possible line-per-line.

In short, you need to write, according to this case, between 2 and 8 times more code than you do in Python.

That’s a bit much!

system-message

WARNING/2 in `everything.txt`, line 9025
Duplicate explicit target name: “pyspf”.

I have been able to work some more on [RaSPF](#) and the results are encouraging.

Thanks to valgrind and test suites, I am pretty confident it doesn’t leak memory, or at least, that it doesn’t leak except on very rare cases.

I think I found a neat way to simplify memory management, though, and that’s what I wanted to mention.

This is probably trivial for everyone reading, but I am a limited C programmer, so whenever something works unexpectedly right, I am happy ;-)

One problem with C memory management is that if you have many exit points for your functions, releasing everything you allocate is rather annoying, since you may have to do it in several different locations.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “this”.

I compounded this problem because I am using exceptions (yeah, C doesn’t have them. I used [this](#)).

Now not only do I have my returns but also my throws and whatever uncaught throw something I called has!

Hell, right?

Nope: what exceptions complicated, exceptions fixed. Look at this function:

It doesn’t matter if `spf_query_expand` or `spf_query_dns_txt` throw an exception, this will not leak.

Nice, I think :-)

system-message

WARNING/2 in `everything.txt`, line 9055
Duplicate explicit target name: “this”.

Hope it wasn’t any serious trouble!

Ok, time to go a little more public with this.

Here’s a page for it (click on “read more”) and I will ask the openspf guys to put it on the implementations list (let’s see how that goes).

I spent another couple of hours tweaking the blog templates, and the latest is half-reasonable. Still a few problems:

- The banner is very ugly on IE

- Stuff is misaligned (too much margin/padding tweaking)

But hey, it works, and it looks ok, which is more than the usual.

I am also trying feedburner's site stats service, which is pretty decent.

I have this agreement with Rosario: we will go to the movies together. But not, under any circumstances, to see something like [Crank](#).

So, on wednesdays (cheap tickets, yipee!) if I am downtown, I will go see that kind of things.

Two weeks ago, it was [Rocky Balboa](#). Yesterday it was [Crank](#).

Crank

This is a special movie. It may seem on the surface to be an ordinary brainless action/violence/sex spectacle, the kind many take for granted nowadays. However, there are some things noone thinks about when watching it.

When I was a kid, this movie's cinematography would have been impossible. It shows how action films made 20 (maybe 10) years ago look cheap today, because technical advance has been massive.

The script may be dumb, and medically incomprehensible (like the alternative uses of [ephedrine](#) and [epinephrine](#) for the same thing. They are not.), but you know what? It's an excuse to have a guy that can't stop. The movie is about a bad guy that can't stop, not about "Haitian hardcore shit. Made of plant shit, man!". Come on, is *radioactive spider bite* any better? Or *brain cloud* in [Joe versus the Volcano](#) any better? (Ok, bad example there).

No one complains that Spidey makes no sense. Or that Kong is too large. And that's OK. Then why is the idea of a guy that needs constant adrenaline stupid?

I think that this movie is seen as stupid speaks more about how many things we have already seen. This is not even the first movie about a poisoned character guaranteed to die looking for revenge I saw (That would be [D.O.A.](#) which is actually a pretty decent movie).

Look at action movies like Die Hard. Is it any less stupid? Maybe (and maybe not). But it was there first.

And at least Crank doesn't take itself too seriously.

As a bonus, it contains the best spoken-portrait-that-looks-just-like-the-guy-scene ever.

BTW: it's really **not** a good movie.

Rocky Balboa

Warning: this below is a little personal, rambling, and probably not a movie review.

You can make a case for this actually being a good movie. It may not get all the way there, but it's a decent job, done by competent craftsmen. You have to consider the movie as part of a larger thing, the sports movie genre. In particular, the boxing movie.

Everyone has seen them ... Underdog ... Long Odds ... Training Montage ... Final Fight .. Last Round ... it's almost like the conventions lie besides conventional narrative, just a series of cliches waiting for you, coming at you like a river. You can hear the upper case initials like waves hammering at your temples.

But I have many soft spots. And one of them is for boxing, and it's for purely personal, subjective, undefensible reasons.

My dad had a stroke many years ago. He was always a dry man. Affection with him was not exactly a simple matter. We had very little in common.

But on late friday nights we would watch the fights. We would talk about them, he with the difficulty of not being used to talk to me, later that of the hemiplegic, me with the difficulty of talking to him.

After a while, I started living alone, but kept watching the fights, and when I visited him, we would do it together. Until Benn-McClellan, which cured me from any enjoyment of boxing I ever had.

But I can still appreciate exactly how hard boxing is. Any sport at a high level requires skills I don't have, and is hard. Now think about practicing a sport (Any sport. Figure skating, if you will!) while another guy, just as strong and quick as you is trying to beat the crap out of you. That's boxing. You are not going to convince me there is anything harder in sports (except maybe some other combat sport).

And Rocky Balboa is a tale about a guy whose heart is so broken he can only think about that skill he used to have, and he gets a miraculous chance at doing that again, and it's a goddamn fairy tale, and nothing else, but I know how the eyes of a guy who would die, who would kill for a chance at doing something he can't possibly do look like.

And that kills me. And this movie killed me. And I got shivers at the end. And I am a goofball.

Back in october of 1996, when everyone was saving for the flying car we would buy in 5 years, and KDE was starting, slowly, to take shape in the minds of a few... there was a generous offer..

```
Matthias Ettrich (ettr...@ti-ibm03.informatik.uni-tuebingen.de) wrote:*
>
> -----
> New Project: Kool Desktop Environment (KDE)
> -----
>
>
> Programmers wanted!
```

```
Freedom Software would be willing to contribute with
the source code of Freedom Desktop Light for this effort.
Please don't subestimate the task of building a
desktop manager. Several Years have been spent building
Freedom Desktop. We could also contribute with
other pieces of technology (i.e Freedom Rt - Object oriented
toolkit). For more information about Freedom Desktop,
please visit http://www.fsw.com
```

```
Freedom Software is about to announce a free version
of the software for Linux (personal use only). This version
is called Freedom Desktop Light for Linux.
```

```
If I were you, I wouldn't restrict the project to a specific
toolkit (at least for now). There are many pieces of public
software that can be reused easily. It could take a long
time to rebuild everything from scratch. Try to reuse
the more you can now. You can standarize on a single
toolkit later.
```

```
Also keep in mind that Motif is the defacto standard.
Most Unix platform ship with Motif. It would be nice
if your desktop work on all the versions of Unix
```

Edgar Galvis
Freedom Software
<http://www.fsw.com/motif.html> - Home of Freedom Desktop for Motif
supp...@freedom.lm.com

UPDATE: I had not bothered checking, but freedom desktop's site has been available until very recently. There is still something in the internet archive, [too](#).

I have been posting this blog using PyDS for over 4 years now. Sadly, the PyDS author seems to have abandoned it. Which is sad, because it's nifty software.

However, keeping it working is getting harder every year, and I don't expect to be able to do it soon.

Also, the data is in a Metakit database, which is the most annoying DB ever (no real schema! columnar instead of record oriented! gouge my eyes with a breadstick!)

So, since I have all the data, and my blogging needs are modest, and no tool does exactly what I want, I decided to write my own.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "raspf".

I could make it a web app, maybe using TurboGears, but what the heck, I haven't done a decent GUI app in ... ok, arguably, I **never** have done a decent one, and my PyQt4 needs some work, and I am kinda in a groove for actually finishing things lately (I am rather proud of [RaSPF](#)).

system-message

WARNING/2 in `everything.txt`, line 9196
Duplicate explicit target name: "raspf".

And I have a neat name (BartleBlog) reserved from another aborted app. So, here's the mandatory screenshot after a couple hours hacking:
And here are the goals:

- Generate static pages, so it can be used by anyone with a little web space (I am a gipsy)
- Simple templating (Using cherrytemplate right now, but should be modular)
- Restructured Text as input mechanism (again, modular)
- Good support for code snippets
- Should support static pages (like the ones I have in the Stories link)
- Integrate with Flickr for images
- Integrate "chunks" in the templating, where you can do things like setting the right Haloscan comment/trackback links easily
- Simple category mechanism, with a regex-based autotagger *without creating per-category copies of everything*.

- RSS feed generation, global and per-category.
- A way to import all my PyDS blog (and maybe my older advogato things)
- Use sqlite and SQLAlchemy for sane storage.

So far, it's doing some things, I can import, edit, save (by instant application, there is no "save" here).

I can't yet generate the site, or create a new post, and it should take months to make it useful, but let's see how it goes.

Eric Bina's Xaw Mosaic.

Just read it.

Warning: foul language

Severe Warning: rampant programmer elitism

Extreme Warning: he's SO right.

After a few more hours hacking, it's got the following working:

- CherryTemplate templates that do about the same as the Cheetah templates in PyDS
- Generates the whole site and it looks just the same
- Advogato import (my blog should go all the way back to 2000 when I switch!)
- PyDS import

The main missing things are:

- Do a decent templating system (right now they are embedded in the code)
- Do a decent config system (right now, global variables)
- Do uploading (or just trust lftp)
- Do post/story creation
- Port the RSS template
- Flickr integration
- Integration with all those neat little gadgets: feedburner flares, HaloScan comments which are currently kinda grafted (only work for my account ;-)
- Look into Yahoo UI toolkit for things like the calendar and menus.
- Add the extra stuff to Restructured Text so it:
 - Fixes automatically links to posts/stories in the blog
 - Pretty-prints code using SilverCity
- Lots of UI stuff

All in all, not really a huge amount of work, but I am taking it easy.

When KDE4 is out, a version with a full-fledged KHTML in it will be a whole lot nicer.

A couple more hours of hacking, and the templates are all new, and more functional than ever.

I am making heavy use of Yahoo's UI library, which makes lots of things much simpler:

- Layout using *Yahoo Grids*

I spent hours making the layout you see now, and the one with Grids works better and was done in minutes. Avoid reinventing the wheel works for webpages, too.

- Calendar using *Yahoo Calendar*

Isn't it neat? And it works, too. Since the linking is handled by javascript I may make it so it loads the posts for a month without reloading the page.

- Styling using their *reset.css* stylesheet.

That stylesheet removes **all styling** from your page. That way, if there's something there, you put it.

I used that, added a slightly simplified stylesheet based on Firefox's default, Restructured Text's and Silvercity's, and all the customizing I needed to do to achieve a simple but functional layout were 30 lines of CSS, compared to the rather monstrous pyds.css my blog currently uses.

- Modular thingies.

I turned all Technorati/HaloScan/FeedBurner/Talkr thingies into macros that take as configuration your personal data (for example, HaloScan ID) and if necessary a post.

If the styling was a little more done and a few bugs were ironed, I may even start uploading the site using bartleblog instead of PyDS soon :-)

Another morning, another feature: archive

Now I'm working on the image tool, importing PyDS's images and uploading to flickr, etc.

Added a Yahoo! Ui menubar generator. It turns this:

```
a
b
c
 1
 2
  x
  y
d
```

Into this:

Or this if you are using links:

Or rather into this, which explains why it's necessary (consider this is a very simple menu!):

Still some missing features, but a neat thingie already.

I had updated to PyDS 0.5.2 and wondered why I saw no change... well, it turns out one has to update the templates in ~/.PyDS, of course.

Now, I will try to do that and then port back the changes I had made.

Curious internet fact:

Since I posted my "review" of [Crank](#) I have had no less than 5 visitors daily from google, searching for variations on "haitian shit plant shit".

In fact, the review is almost the top result for that. Somehow I don't expect this to attract any sponsors, though ;-)

I added a Restructured Text directive that takes as argument the title of one of your (my) flickr pictures, like this:

```
.. flickr:: pa120071
```

And produces this:

Which is much nicer than my images have looked for a while.

It can still be improved, probably add a caption, link to image comments if available, etc.

The bad side is that if you don't have internet access the rendering fails.

A lot of new features going on, like [openomy](#) based file uploading and links that work like simple rst directives:

```
.. openomy:: my-file.ext
```

And you should get a link to the file, stored in the 1GB the [openomy](#) guys kindly provide.

Also, you can actually see how the test site looks at <http://lateral.netmanagers.com.ar/weblog> (using the exact same data as this one).

I actually upload almost every change I do, so it breaks often ;-)

Now that the backend is still iffy but generates a blog, I finally started showing some love to the UI.

Here's the brand new post editor:

The far-right two buttons should let you choose your flickr photo/openomy file respectively and add the link/thumbnail easily (the UI is not done yet).

The format to write is Restructured Text or HTML (non wysiwyg) but other things like Textile or wiki markup should be easy to add.

The main window uses a Qt model to create the tree with views (right now) by date, but with alternative organizations soon.

I always get this weird high when I'm coding. I have no idea why. I just do.

Last night, I didn't sleep, I was so hyper.

Brand new article, just finished writing it.

This article describes the queue system of qmail, and the tools you can use to manage it. Also, it introduces a new one, called qmail-cleaner, to find messages in the queue that match a given regular expression.

Nice post at mechanicalcat.net, about why python makes good programming easier, or at least not harder.

They are regular PAL TV cameras, 704x576, and that is a regular CD.

I am developing a security camera solution based on Linux and motion, and well, the gadgets are just too nifty :-)

One of those is infrared and can shoot in the dark. It even has LEDs to illuminate (in infrared, so you can't see).

Paranoids must hate technology nowadays!

In the python listing at the end of the article, I applied py2html so it would get syntax highlighting.

In mozilla it looks ok, in Konqueror it flows over the box.

I wonder if that's how it's supposed to work and it only looks well in Mozilla by accident.

It's only about grain and sepia, but if people try this kind of things, they start getting ideas.

This is the first article in what will hopefully will become a series explaining how a little creativity can enrich the KDE desktop experience, maybe using a not-very-popular feature.

This time, I explain how to make listening to Internet radio a more pleasant experience, using quickbrowser buttons.

I can't find any editor (outside of emacs) that has a mode for it. And it needs a mode ;-)

In fact, because of the way it's marked, it seems a mode for scintilla or kate simply can't be done.

The problem is that, in order to understand what a paragraph is, you need to see *the end* of it, sometimes (for example, headings)

So, since most highlight parsers switch when they trip onto something (say, a quote), it doesn't work.

I *could*, I suppose, do highlighting using something more evil, like detecting when the user ends a paragraph to set the paragraph's style, in addition to the usual stuff, but it seems pretty hard to get right.

Well, what the title says, mostly :-)

How to integrate qmail and spamassassin, so your mail gets filtered, trying to keep it simple.

So, I'm only posting out of discipline.

Looks like I will finally deliver a product that has been finished for a long time. Let's hope it doesn't crash and burn too soon.

So, Linuxtoday picks it, and it had a serious mistake: ifspamh is not a perl script, but a korn shell script.

Fixed now, hopefully noone screwed his system because of that ;-)

New tutorial: Enabling remote blogging securely in PyDS, using authentication and a HTTPS reverse proxy.

Useful if you want to, for example, put PyDS on the Internet to blog from anywhere!

If this works, then I have something here!

Ok, klog is just a temporary name, but it's 4AM, and I am not creative. So, what is it? It's a PyQt client to PyDS.

Look at the picture in the link, if you want!

Now, I am not sure if I will work on it much, since I only did it to learn a bit about XMLRPC, but it DOES work, or else, you won't be able to read this anyway ;-)

Rationale: While PyDS is a great piece of work, editing text through a web page feels like scratching wearing gloves. It is doable, but it's not satisfying. So, I tried this just to see how hard it would be... answer: not very hard, really.

Just to keep me honest, and keep the software somewhat designed instead of piled, I am doing a advogato module that replaces the PyDS one (of course with much less functionality).

This is a recipe at the Python CookBook showing how to add a new directive to reStructuredText, called code-block, which syntax-highlights the block!

I just implemented it in Notty, and now you can have the neatest code snippets. I did have to hack the output a fair bit since the displaying widget doesn't support (or CSS for that matter), so I had to do massive search-replaces on the output, to convert to . But it took only half an hour to do, since SilverCity's output is very regular.

I think I will suggest this to Hugo for PyDS, considering the density of programmers among its users :-)

The link goes to a message by Havoc Pennington describing D-BUS, a IPC technology being developed for use in Red Hat, and perhaps as a future (think 18 months) replacement for DCOP and maybe Bonobo's IPC mechanism.

Sounds nice, at least what I can understand of the specs, and adopting it shouldn't cause any big trauma on the KDE side, since our DCOP classes are autogenerated by macros anyway. Hopefully, it may even be source compatible.

Lots more information, including the specs and an interesting take on KDE-GNOME cooperation by Miguel de Icaza if you follow the links a little.

This guy posts 1200 word magazine-quality information-packed articles so casually...

Two hard days of work, including migrating a server at midnight. A server that noone really knew exactly what it did!

You see, lots of apps went through it. Weird business custom proprietary apps for several companies.

So, what had to be done? make another one of those evil contraptions work. But apparently noone found a way to pile another hack on the hacktower, and my job was to clean up the mess.

Well, they haven't called me yet, so either it worked, or the guy that I gave my number was fired ;-)

What satire has become! I was reading <http://barrapunto.com> and...

Some quotes:

"However, these propagandists aren't just targeting the young. Take for example Apple Computers, makers of the popular Macintosh line of computers. The real operating system hiding under the newest version of the Macintosh operating system (MacOS X) is called... Darwin! That's right, new Macs are based on Darwinism! While they currently don't advertise this fact to consumers, it is well known among the computer elite, who are mostly Atheists and Pagans. Furthermore, the Darwin OS is released under an "Open Source" license, which is just another name for Communism."

I mean, this guy is a comic genius!

Part II of the series. In this article, we replace qmail-smtpd with qpsmtpd, which is more flexible and spam-aware.

I wonder if bartlebog is a good name. It sure is better than klog, and I like XIXth century lit references.

I also suspect almost noone will get it, and that it just can't be pronounced ;-)

I wanted it to be bartleby, but there is some tool using that name in sourceforge.

On other issues: it's HOT here!.

...and still lots to do. Well, not really late, just 8PM, but lots and lots and lots to do.

I really should start saying no to some jobs. Or hire someone.

As of now, I am teaching 3 courses, total 21 hours/week, plus consulting (about 30 hours/week). Call me lazy, but 50hs/week is just too much.

Sure, I get paid for it, but what do I want money for if I can't do anything I want?

Of course what I want is to play "Driver 2"[1] for hours and then code a little. But hey, to each his own taste.

[1] Yes, I am playing Playstation 1 games. I own a PS1. It's nice, it hooks to the TV, so the image is big, and the sound is loud. That's about all I ask of the gaming experience :-)

Just to see if anyone can figure this out:

How do you install Linux on a box with the following handicaps:

- No ethernet (or token ring or arcnet, etc)

- No CD (or DVD)
- A pcmcia floppy that has no Linux driver
- No other OS installed
- You can't add any hardware
- You can't move the HD to another box or open the box at all.
- It must be a real distro, like Red Hat, SuSE or Debian, not a toy or floppy distro.
- Available hardware in the box: keyboard, mouse, serial port, parallel port, monitor, a toshiba libretto floppy drive (assume it doesn't work with linux, yes, I know there's a "driver", if you can make it work, I want a copy ;-)
- Use of another PC with whatever software or hardware you want is allowed, except the software must all be Linux (no DOS floppies or stuff like that)
- Any cables you want can be used.
- Any media you want can be used for the install.

I will give the answer in a week, hopefully someone will answer first.

Yes, it *can* be done. In fact, I have already done it, a while ago (see the "Small Linux" article in this very site)

Found a few references to the trick question I posted a few days ago, and I really don't understand them.

I mean, some guys say "you throw the box away" or "you are a masochist". Well, as it turns out, I had good reasons not to throw it away, and good reason why those constraints were in place.

The good reason not to throw it away is, of course, that it's a sweet computer. It's smaller than almost any other notebook sold nowadays, rugged and it works just fine. And it can be useful. So why throw it away?

As for masochism... I have no licenses for DOS, so I couldn't use DOS floppies. I had no PCMCIA gadgets and no money, so I didn't use a PCMCIA CD, or USB or flash disk.

Finally, I didn't open it and move the HD to another box because it's a notebook. They are quite tricky to open, and the Libretto more than most. I had just got it, if there was any chance of not breaking it, I wanted not to break it.

And of course... I expected most readers to be, well, hackers, in the old sense. You know, complication for complication's sake is the basis for technical fun sometimes, as long as it's not overdone. Since I did this in one afternoon, I just had fun for a few hours.

Saying that this is masochism is about as stupid as complaining that basketball is masochism because it would be easier if you built a conveyor belt to drop the balls in the basket!

It's a game, folks! Part of the game is that rules restrict you!

The rules I gave were the constraints I faced. Of course other situations would make it easier.

And if the San Antonio Spurs were allowed to use a conveyor belt, they would have won the NBA finals in 4 games!

For different clients, I am having to install programs that for some reason distros don't include.

Courier-imap, a dozen perl modules, netatalk... why aren't they in Red Hat or other mainstream distros? Sure, Debian has them, but I don't like Debian.

Oh, well. They are easy to rpm-ize, so I just do it myself. But then I have no place to put them so others can use it.

Does it mean I should become part of something like fedora or somesuch? I don't want to maintain them forever!

I did some small code cleanup in bartleblog, the advogato plugin now works ok, I will start work in a blogger plugin.

After that:

- A semi-decent editor. Maybe HTML, maybe structured-text, who knows
- GUI cleanup and prettification
- Plugins for a few more PyDS tools
- Public announcement? Who knows.

Or I am missing something. I mean:

- No way to list the posts?
- No way to fetch a post text?

So, if you migrate to another server... you are screwed? If you lose your client's local version, you are screwed?

Well, at least it's easy to write ;-)

I opened a blogspot account to test:

<http://bartleblog-test.blogspot.com/>

Who's Hugo? Why am I calling Georg Hugo? Is Alzheimer coming for me? ;-)

Excuse me, I have to go polish my chrome!

Here, no reference to getRecentPosts:

http://www.blogger.com/developers/api/1_docs/

Here, the explanation of getRecentPosts:

<http://groups.yahoo.com/group/bloggerDev/message/225>

I try it... it works. So... am I using MetaWeblog by accident, or the API docs are just incomplete?

Who knows!

Ok, so editPost, which `_is_` in the 1.0 API doesn't work. I am obviously using the wrong API reference. Anyway, the plugin works, except you can't edit stuff that's already posted. Uploaded to CVS and everything.

Someone who bothered picking up all the docs of the `_real_` blogger API. I hope ;-)

...and I have done nothing important :-)

Oh, sure, I played games, and drank beer, and had fun, and called my mom (mother's day was today), but no real advance on anything at all.

I start teaching another course tomorrow, so I am copying CDs and browsing... really I am a boring dude nowadays.

Since Red hat 8.0 is aging a little, I am updating it via apt-get into Red Hat 9.

Surprisingly, it seems to be working ;-)

Looks like the kind fellows at kde-redhat.sf.net have a working PyKDE, and I got it while I was upgrading.

That's useful because I really wanted to hack some into KrSN (the RSS aggregator tool I once wrote), and it needs PyKDE.

I hadn't done it because building PyKDE on a PII, 233MHZ, 128MB of RAM (the only one I have on the Internet) is really NOT fun. But once I got it on RPM, using it works just fine :-)

Now, how is a RSS aggregator supposed to sort these things by date?

- Tue, 07 Oct 2003 23:09:52 GMT
- 20010627T12:40:33

the first ok for a human, but for a computer it is not, viceversa for the second :-P

Sadly, that is what PyCS and advogato return in the date field of RSS feeds.

So, the client is forced to assume that:

- That thing must be readable somehow for the user, and cant format it (I try to parse for the advogato format, when it doesnt work I leave it as is)
- Items cant be sorted by date cross-feed (ie: give me the latest posts from these ten feeds) because I have no reliable posting date, and must use instead the date it was fetched.

Not nice.

And I mean it literally this time.

I just had 6 hours of straight solid coding (KRsn, if you are interested), and I feel great.

I can't even remember the last time I did this. But hey, at least now KRsn has a semi-decent data handling, shouldn't lock on the DB much (I still have some concept problems there), and should pretty much work, as long as you don't take its multi-threading too seriously.

...all you need is either a very strong magnet, or some fire and a sieve.

But finding a post I agree with in Slahdot... now *that* is a hard thing.

But here's one!

And someone posted it on the internet.

So, the RSS date stuff I mentioned? Mark Pilgim already fixed it with `rssfinder's parse_http_date` function.

Even though it's not really ready, I am switching my blog to BartleBlog.

The idea is that I do blog. So I should be able to focus on what's needful for my own kind of blog. For example, right now, the idea of not being able to tag this post is driving me nuts.

- I did the tag editor:

The "Guess Tags" button in the composer uses the "Magic words" defined for each tag to give you a starting set, which you can edit later. Still has a few implementation issues (thus the taglessness of this post) but it's almost there.

- Using [HTMLTidy](#) I made all the HTML validate (as transitional, at least).
- Using [browsercam](#) I made sure this page works on every decent browser (it looks bad on IE 4.0, and Netscape 6):

There is a nasty problem with IE and JavaScript, where if you have a <script> that does some specific things inside of a <div> it tends to give a weird “Operation aborted” dialog after showing part of the page, and then **delete what you were seing**. That is a known, confirmed bug, and has been there for three major versions already.

- Added a feedBox widget that can display as a sidebar any feed that uses [FeedBurner’s BuzzBoost](#) (see “Latest comments” on the sidebar).

I am now toying with using FeedBurner’s API to (optionally) automatically publish all the RSS feeds BartleBlog generates there, thus making this kind of thing more transparent.

I had a few minutes waiting for yum to do its thing and added a couple of easy features:

The SilverCity-based code-block syntax highlighting directive was replaced with a version using [Pygment](#) which supports more languages and is (I think) nicer looking... and comes with several styles, as you can see above in the configuration dialog.

Other than that, I added support in the backend to:

- Regenerate HTML from RST everywhere
- Do the same only on items marked “dirty”
- A model thingie that shows if there are posts or stories with RST errors
- Some GUI love

I need to make the config dialog support more than one gadget at the same time, though ;-)

Is there a name for that? It’s like picking oneself by one’s bootstraps.

But anyway, I have been writing a bunch of little PyQt projects in the last year or so, and they all seem to share something with each other.

Writing KrSN, I worked on making a pseudo/browser using QTextBroser... which is the note-displaying element in Notty... which now is getting a reStructuredText editor... which will be used in Bartleblog... for which I learned how to use XMLRPC.. which I will use to write a Notty server... and so on.

Of course, since I am doing all this as a hobby, a reasonable release of all these apps is still months in the future (KRsn is usable, although incomplete, Notty will probably be out in a decent shape first), but I am only coding for fun, any use anyone gets of it is strictly a happy side-effect, so... it’s amusing in its own terms.

Nice read. Haven’t seen the ads, though.

Since I uploaded Notty to [Sourceforge](#) (get it from CVS), I thought it would be nice to show what it looks like nowadays.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “screenshot”.
```

So here’s a [screenshot](#). If that link doesn’t work, try [this one](#).

It is mostly a working app, although still pretty basic. You can use it, you can create notes, you can even export them to HTML (working fairly well, except for links to other notes, which don’t get fixed).

I am even using it to document itself (see docs/help.notty in the sources, it's a notty file)

Specially nice is Silvercity's code-block directive, which you can see used in the screenshot (look at the yellow fragment).

I'd say it is *right now* a pretty decent documentation writing tool, although the editor component is too featureless.

```
system-message
WARNING/2 in everything.txt, line 10844
Duplicate explicit target name: "screenshot".
```

It seems I can't just stop working, so I will now be teaching three courses (24 hours/week) plus the consulting, plus whatever.

I wonder why I do that.

Phil's support is great. That's good enough news, but he helped me make images-and-http-in-qtextbrowser work again.

I think I will package that widget neatly, because it's of general usefulness.

Now, I didn't expect this to work at all!

A QListView is a sort of tree widget, and is drag&drop enabled.

The trick is, you have to reimplement dragObject() so it returns a draggable thingie after you drag something.

But, since I am using designer to draw the forms, I have to use a real QListView, not a child class.

Conundrum? Custom widget in designer? Not with python!

```
def myDragObject(self): return QTextDrag("something",self)
```

```
self.tree.__class__.dragObject=myDragObject
```

And that works! I know it should work in python, but QListView is implemented in a C++ module!

The guys that did PyQt really made it well.

Here's the code for a modified QTextBrowser I am using. I hope it's useful for someone else, too :-)

The main differences:

- It can open URLs
- Images referenced from the text work
- It prints (stolen shamelessly from helpviewer)

It's in the public domain, enjoy!

Yeah, I fixed the same bug that was making images/whatever not work in Notty, (and that is, of course, a feature of pybrowser), and it works!.

So, I can now skim all my news quickly again :-)

As soon as I release Notty (it's pretty close), KR'sN is the next in the pipeline to get a scrubbing and a release.

Of course, anything can happen, like a really good date, then everything goes on hold for a couple of weeks or months :-)

It's better than I remember. Specially because the SF CVS version is newer than all the copies I had on disk (I wonder how *that* happened).

In fact, it has a nice trick to create error dialogs from any exception I am lifting right away for other apps :-)

The only problem it does have is that python's urllib and urllib2 support for FTP seems to suck.

You can't choose the username and password?

Sailor (Gutenberg's main server) is picky about the password you send, and ralsina@roberto, which urllib insist on using ([username@hostname](#)) doesn't pass muster.

And ftplib, which can do the right thing, doesn't let me give progress of the download, which would make the program awful.

Oh, well.

Just that. Click in the title for details and a screenshot.

I imported my old advogato diary so now this blog goes back to 2000/1/17 17:19:57.

For some reason I didn't blog anything in 2002, though.

<http://noonebelongsheremorethanyou.com/>

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "this one".

I was going to write one, but there's absolutely no need when there is [this one](#)

system-message

WARNING/2 in `everything.txt`, line 12638

Duplicate explicit target name: "this one".

Isn't this neat?

```
\ :kn: 'import '\ \ :nn: 'urllib'\
\n :k: 'def '\ \ :nf: 'makeTiny'\ \ :p: '('\ \ :n: 'url'\ \ :p: ')': '\
\n :n: 'url'\ \ :o: '='\ \ :s: 'http://tinyurl.com/api-create.php?' '\ \ :o:
\n :k: 'return'\ \ :n: 'urllib'\ \ :o: '.'\ \ :n: 'urlopen'\ \ :p: '('\ \ :n:
\n :o: '>>>'\ \ :n: 'makeTiny'\ \ :p: '('\ \ :s: 'http://www.kde.org' '\ \ :p:
\n :s: 'http://tinyurl.com/3cnthx' '\
```

It was pretty hard, but I finally managed to make bartleblog re-render only the necessary pages.

In fact, now my blog has a few *less pages* because some random stuff was generated in the past, and never got deleted.

There are still some bugs but, with this, it's a reasonably useful app.

Todo:

- Alternative comment system based on JS-Kit
- Flickr photoset directive
- Nicer Openomy tag directive
- Flickr and Openomy uploading

- Many config pages and wizards
- Background rendering
- Lots of UI love
- FeedBurner support to automatically use their version of the RSS feeds

Other than that, it's pretty much feature-complete, or at least it has enough features a nerd can use it to post a blog just like this one, for whatever that's worth.

This is something I wrote in March of 2000, about how to write a cooperatively multithreading app using Qt and GNU Pth.

Amazingly enough, I think it still may have some little value :-)

The syntax is probably wrong considering Qt has gone from 2.0 to 3.2 and Pth from 1.0 to 2.0, but it should work as a rough guide.

One funny bit: I had no copies of this. So, it's a good thing the Internet never forgets :-)

The nerdiest feature of all time... simple SVN logs.

Here's the RST code:

```
\ :p:`..`\ \ :ow:`svnlog`\ \ :p:`::`\ https://ra-blog.googlecode.com/svn/
\ :nc:`:limit:`\ \ :nf:`3`\
```

And here is what it does:

system-message

ERROR/3 in `everything.txt`, line 12705

At least one body element must separate transitions; adjacent transitions are not allowed. backrefs:

You can see the entries are nicely formatted. That's because this restructured text directive connects to your SVN server, retrieves the changelog, parses it, tries to process the individual entries as restructured text, and then formats and spits it out again.

I decided to add the recursive parsing because I tend to format everything I write as valid restructured text anyway. But that means I can include images, tables, files, and even other logs in my entries.

How is it useful? Well, I intend to have my projects' pages written using this. This way I don't need to keep a changelog.

Also, I bet the title is right ;-)

First, the message is important. Then, the guy is a very good public speaker, but ... awesome graphics. I have never seen anyone use such awesome statistical displays. If I was still into maths, I would love that kind of gizmos!

The main thing is that it's not about the gizmo, but about new ways to look at data, which should really be the main job of statistics.

Update: And it turns out there is an online version of it: <http://tools.google.com/gapminder/>

A new gadget: **Mootools** based menus.

It takes the same data format I used for the Yahoo UI menubar, and creates a dynamic, sliding, animated, pretty, works-on-lynx menu.

No screenshot because it makes no sense without the movement. You can see it at the top of the sidebar (click on the darker items, outside the links... that needs work).

And yes, that means this blog has, right now, two menus with exactly the same data in them. And on a non-JS browser both are fully expanded.

But that's just temporary ;-)

Now, I don't know if this is useful, but I do think it's way neat.

Inspired by [S5](#), [rst2s5](#), and [mootools](#) I took an hour (or two) and hacked this neat little slide tool.

The goals differ from S5 in that I intend to write a frontend, so that you have a sort of very-poor-man's powerpoint, but also in that the output should be simple to embed in other pages so that I can eventually make this a part of bartleblog.

Check it out (click to go to next slide, move mouse to the top of the slideshow for controls):

There may be an artifact when you slide out the syntax-highlighted boxes, but I have no idea how to fix it.

Also, I have not tested it at all in IE, so if it fails there, don't worry, that's to be expected!

Next time I speak in public, I may use this :-)

And, as a teaser... this is the source for the presentation you just saw:

```
\ :gh:`Why use BartleBlog`\
\ :gh:`=====`\

\ :gh:`(If you are a nerd) `\
\ :gh:`~~~~~`\

\ :nc:`:author:`\ \ :nf:`Roberto Alsina <ralsina@kde.org>`\

\ :gh:`It's nerd-oriented`\
\ :gh:`-----`\
```

It's trivial to display source code, with proper syntax highlighting.

```
\ :p:`..`\ \ :ow:`code-block`\ \ :p:`::`\ python

from base64 import *
def myFirstFunction():
    print b64decode ('YnllIHdvcmxkIQ==')
```

Even for things like shell sessions.

Dammit, I am a nerd, I will try to add every nerd feature I deem cool.

```
\ :gh:`It does things simply`\
\ :gh:`-----`\
```

Wanna show a flickr photo?

```
\ :p:`..`\ \ :ow:`code-block`\ \ :p:`::`\ rst
\n :p:` ..`\ \ :ow:`flickr`\ \ :p:`::`\ myPhotoTitle
```

How about using openomy.com to share your files?
And whatever else you can think of, it probably
\ :gs:`**can**`\ be done.

```
\ :gh:`It's very easy to extend`\
\n :gh:`-----`\
```

Really. It's simple python code.

```
\ :m:`*\` The module to do syntax highlighting has 41 lines.
```

```
\ :m:`*\` The module to do dynamic animated menus has 103.
```

```
\ :m:`*\` The module to do calendars has 72.
```

If there is a python module or web service to do what you want, hooking it into BartleBlog is simple.

```
\ :gh:`And it has kickass features`\
\n :gh:`-----`\
```

Like online, embedded, animated slideshows! Like this one!
Done with 65 lines of simple almost-plain-text markup!
Ok, it's not fully implemented yet, because it needs some manual code, but the hard part is done!

Or automatic SVN changelog display (I admit that's \ :ge:`*somewhat niche*

Now, who else has that kind of thing?

...surely is a phrase I never expected to hear. But then the rats started laughing.
Check it out in this video (thanks [Videosift!](#))

As a result of my playing with mootools and creating powerpoint-style slideshows on webpages from restructured text, I am pretty much decided about turning this into a standalone, real application.

I am doing a PyQt4 frontend, and restructured text addons to generate the slides.

The native file format is restructured text itself, and it supports basic transitions between slides.

It still needs a ton of work, but here is a video showing how it looks after a couple of days hacking. You can see the current UI, a document's source, and a demonstration using firefox (only because Konqy doesn't suppose opacity, the rest of the transitions work even better on Konqueror)

It's Ogg/Theora and 3MB (thanks [openomy.com!](#)): [the video](#)

... bueno, el gurú dice muchas cosas, pero en particular, César Ballardini acaba de inaugurar su propio blog, <http://katra.blogsavy.com/>

Ni idea de qué va a hablar, pero es una persona que programa Forth para divertirse,

así que es seguramente algo interesante.

This post is in spanish because so is César's blog.

...because in exactly 8 hours Rosario and I will enter the operation room for a c-section, which will deliver our first son, Juan Francisco Alsina Guerrero.

I don't expect to be available for the next 48 hours, of course.

Pictures, maybe video, coming up in a few days.

<p>Bilingual post for everyone that reads this Juan Francisco was born on April 18 at 8:51 AM He was born weighing 3.700Kg As expected, he's a very pretty baby, and both him and the mother are doing just fine. He was born with a high red blood cell count, but that has been fixed, and is in the mother's room and feeding normally. I am pretty much shocked to be a father, but I'm loving it (and him) so far.</p>	<p>Mensaje bilingüe para los que lean esto Juan Francisco nació el 18 de abril a las 8:51 AM Nació pesando 3 kilos 700. Como esperábamos es un bebé hermoso, y tanto él como la madre están muy bien. Nació con un exceso de glóbulos rojos, pero ya está bien, en la habitación con la madre y alimentándose normalmente. Estoy sorprendido de ser padre, pero hasta ahora la estoy pasando bien y enamorado de mi hijo. Fotos aquí</p>
<p>system-message WARNING/2 in everything.txt, line 3 Duplicate explicit target name: "here".</p>	
<p>Some pictures here.</p>	

<p>system-message WARNING/2 in everything.txt, line 13051 Duplicate explicit target name: "here".</p>

Another bilingual post / Otro post bilingüe

<p>Last post about JF for a while at least He had to spend a couple of days under a lamp to decrease his bilirubin but he's at home breastfeeding well, and doing just fine. The mother is in some pain because of the surgery, but getting better. And I don't even mind changing diapers. Even heavy ones. Which was something I was not really sure about ;-)</p>	<p>Ultimo post sobre JF por un tiempo. Tuvo que pasar un par de días bajo una lámpara para bajar la bilirrubina pero ya está en casa mamando bien, y está bárbaro. La madre está un poco dolorida por la cirugía, pero mejorando. Y no me molesta cambiar pañales, ni siquiera los cargados. Que no estaba demasiado seguro ;-)</p>
<p>system-message WARNING/2 in everything.txt, line 3 Duplicate explicit target name: "here".</p>	<p>system-message WARNING/2 in everything.txt, line 3 Duplicate explicit target name: "aquí".</p>
<p>More pictures here.</p>	<p>Mas fotos aquí.</p>

system-message

WARNING/2 in `everything.txt`, line 13082

Duplicate explicit target name: "here".

system-message

WARNING/2 in `everything.txt`, line 13084

Duplicate explicit target name: "aquf".

Mind you, when I say morons, I don't mean the following:

- those using Clarux are morons
- people who don't like current KDE are morons
- people who can't handle UNIX are morons
- people who prefer Windows are morons
- people who prefer Mac are morons

I am specifically saying that those who put in their webpage stuff like

Directory Free kde-libs and Directory Free kde-base have been released (screenshot). Both of these libraries have been modified to support **higher quality users** (emphasis mine)

are morons. Of a very high quality. Or at least they look like morons until they correct it.

On other news: those developers you clarux guys seem to dislike so much are the ones who wrote all the fucking software you are distributing, and they were kind enough to let you spend a weekend changing strings and two icons, then distribute it in such a stupid condescending manner from your website, which is, of course, powered by *more* software you got as a gift from developers.

Got it? Without those developers, you would be using ground plants to draw antelopes on the roof of your filthy caves, Clarux kids.

Now, that feels better. Or was it a little over the top? Nah.

- [MonkeyStudio](#) is a rather neat Qt4 IDE.
- [Katapult](#) is a pretty launcher (but I wonder if it could display more than one match as I type?).

I will try to write more detailed reviews in my [copious free time](#)

Some of you may be familiar with weird programming languages. Some of you may even know about bidimensional languages. Some of you may have seen [Flip](#), where computation is performed by balls bouncing off things.

Now you can see Flip using Qt. My [Ra-Flip](#) interpreter. I wrote it in a lark after finding out about Flip in [Good Math, Bad Math](#) and shares no code with the original, but it's a simple language ;-)

It's almost feature-complete. Missing are:

- Levels (to make it threedimensional)
- Input

- Start-stop-pause-delay controls

Here is the mandatory screenshot:

But that really doesn't do it justice, so here's a video:

The fun starts around 35 seconds in. The program prints an infinite series of even numbers and stores odd numbers in a ... I am not sure how to call that. A circuit?

I may think about writing a something-to-Flip compiler, so you can write your programs in some sort of simplistic imperative language and then see them as a pinball machine.

It may be cute if an artist could spend a few hours making nicer gadgets, before I do it (it won't be pretty ;-)

And yes, this is probably the most useless program written using Qt, right there with kyes.

...nothing.

Ok, here's what I have been doing:

- Daddy work. Lots of it. Fun!
- Actual work. Lots of it. Fun, but not so much!
- Worked a little more in RA-Flip. Play/Stop/Pause is done. The editing mode and opening/saving files is pretty much done, so it will be fully functional as soon as I can spend an afternoon in it.
- Tobogan (My slideshow software). It works nicely. There are many rough spots and it needs a few features:
 - Some way to edit the CSS for individual slides in a nice manner. Like setting backgrounds and colours (and not much else)
 - A nice way to save the slideshow so it ends all in a single folder.
 - Many UI details
 - Docs
 - A site
- BartleBlog: it has worked well for me for a long while, since I have been posting this blog using it. And really, it's a nice blog, isn't it? ;-) But of course, there is stuff to be done:
 - Many UI details
 - Decent flickr and openomy uploaders/managers
 - Some extra gadgets I have in mind
 - Docs
 - A site

And that's all I have in mind right now. Except maybe a graphical Befunge interpreter, but that's not really important ;-)

Got commissioned a small job for a customer:

- Take a form done in ancient HTML using dreamweaver.
- Redo it in PHP+MySQL

- Make it work.
- Add validation.
- Make it look the same.

Ok, it's not terribly hard, but it's not terribly simple, either, specially if you are not all that up-to-date on PHP.

But hey, using [HTML_QuickForms](#) and [Renderer_Tableless](#), it's kinda done.

But boy, do I now remember how I don't like PHP!

I bet you have seen before some article about how some economist calculates the value of things.

For example, if you prefer not to have health insurance worth \$100 a year, and that means you are X% more likely to die in 20 years, then some number-massaging will tell you you value your life at \$Y (the numbers are not really important).

Here is an [example in Slate magazine](#) which includes a rather clear explanation of how this process works.

It shows that a Chicago drug dealer values its life at between \$50000 and \$100000.

Now, this is all based on the fact that that person chooses to go on his life in a certain way for a certain amount of money. However, I am pretty sure if offered \$100000 or life, he would choose life.

Since that choice is just as his as the one used to reach that number, it's clear that human beings are incoherent, or maybe that the difference in value between a 10% chance of death and a 100% chance of death is way larger than economists estimate.

Personally, I can't even think of an amount of money that would make me take a 50% chance of death. But I am aware that, for example, my diet causes me a certain risk, maybe 1% or 10% (no idea).

How about you? Do you believe in economists?

I found a couple of hours to hack, and decided to spend them on BartleBlog.

Here's what got done:

- UI for blog preferences (still need to do maybe ten of these little modules).
- Search for files in reasonable places (instead of using the DB in the current folder, for example).
- Started work on a first-use wizard
- Started working on a original and simple help system (resource files, yay! ;-).
- Implemented (lamely) posting of "stories" which are static pages that are not part of the blog's flow.

I intend to spend a couple of hours on it every day for the next ten days or so. After that, who knows, maybe a release will be reasonable.

Today I was fiddling with my desktop and decided to check what KDE styles were available in [Arch Linux](#).

Then I tested [Domino](#).

It's very, very, very nice.

I have no idea how good it is performance-wise (my notebook is pretty brawny) but hey, I am a boring guy, and I'm wowed.

I probably should have charged more.

Things I learned in this project:

- If they tell you the design is important, it doesn't matter that said design includes:
 - Fixed pixel-height text inputs (so when you resize fonts you can't see what you write)
 - Misaligned fields and labels

So, I needed to use Smarty to generate the **exact** same HTML, working just as bad as the original. And it is thrice the work of doing it right.

- If deployment is on Windows, regardless of having the same version of PHP, you are going to get in trouble one way or another.

In order to do it quick & cheap & good, I used PEAR modules extensively. And now I have to make sure all of them are in the server... where there is no remote admin solution usable. So I have to be onsite to deploy. I can't even remember the last time I had to do such a thing :-)

So, I was cheaper than the other proposals, and maybe I was a bit too cheap. Live and learn :-)

It's remarkably easy to turn your QTextBrowser into a limited web browser, at least good enough to show images from the web.

Here's all the code:

```

\ :kn:'from' \ \ :nn:'PyQt4' \ \ :kn:'import' \ \ :n:'QtCore' \ \ :p:', '\ \
\ :kn:'import' \ \ :nn:'urllib' \ \ :o:', '\ \ :nn:'os' \ \ :o:', '\ \ :nn:'n

\ :k:'class' \ \ :nc:'PBrowser' \ \ :p:'('\ \ :n:'QtGui' \ \ :o:'. '\ \ :n:'QT
  \ :k:'def' \ \ :nf:'loadResource' \ \ :p:'('\ \ :bp:'self' \ \ :p:', '\ \
    \ :n:'url' \ \ :o:'='\ \ \ :nb:'unicode' \ \ :p:'('\ \ :n:'name' \ \ :o:
    \ :n:'ret' \ \ :o:'='\ \ \ :n:'QtCore' \ \ :o:'. '\ \ :n:'QVariant' \ \ :
    \ :k:'if' \ \ :n:'url' \ \ :o:'. '\ \ :n:'startswith' \ \ :p:'('\ \ \ :s
      \ :n:'dn' \ \ :o:'='\ \ \ :n:'os' \ \ :o:'. '\ \ :n:'path' \ \ :o:'.
      \ :k:'if' \ \ :ow:'not' \ \ \ :n:'os' \ \ :o:'. '\ \ :n:'path' \ \ :
        \ :n:'os' \ \ :o:'. '\ \ :n:'mkdir' \ \ :p:'('\ \ \ :n:'dn' \ \ \ :
        \ :n:'m' \ \ :o:'='\ \ \ :n:'md5' \ \ :o:'. '\ \ :n:'new' \ \ :p:'()
        \ :n:'m' \ \ :o:'. '\ \ :n:'update' \ \ :p:'('\ \ \ :n:'url' \ \ :p:
        \ :n:'fn' \ \ :o:'='\ \ \ :n:'os' \ \ :o:'. '\ \ :n:'path' \ \ :o:'.
        \ :k:'if' \ \ :ow:'not' \ \ \ :n:'os' \ \ :o:'. '\ \ :n:'path' \ \ :
          \ :n:'urllib' \ \ :o:'. '\ \ :n:'urlretrieve' \ \ :p:'('\ \ \ :n
          \ :n:'ret' \ \ :o:'='\ \ \ :n:'QtGui' \ \ :o:'. '\ \ :n:'QTextBrowse
    \ :k:'else' \ \ :p:': '\ \
      \ :n:'ret' \ \ :o:'='\ \ \ :n:'QtGui' \ \ :o:'. '\ \ :n:'QTextBrowse
    \ :k:'return' \ \ \ :n:'ret' \ \

```

And here's bartleblog taking advantage of it:

It even has a primitive cache and everything ;-)

- Done with the main blog config dialog.
- Fixed a dozen bugs
- Generate the blog in a reasonable place

- Fixed a lot of UI bugs (tab orders, sizes)

Still lots and lots of things to be done, tho.

Ilan, darling, you are paranoid. What you did *is* a patch. When you modify a few strings (my guess is about 25 of them), that's not a software project.

I also missed the part where anyone forced anything on your government (whatever that government is). Must be the power of the KDE guerrillas, subverting the countryside.

... has been all about UI.

I have always had a problem when writing PyQt apps: stock icons.

Which ones should I use? Where are they?

I usually fished through the crystalsvg icon set until I found one that seemed to be what I needed, and then copied it to my app.

Sadly, that's annoying in several ways:

1. Since those are PNG icons, you need to find the right size.
2. Not all icons are there for all sizes!
3. Because of 2, I need to check three or four folders to see **all** the icons.

So, I decided to cut my losses, and see what else could be done. And here it is:

I am now using all SVG icons, from [the reinhardt set](#) that will look equally out of place in all OSs, but which **I** like (and I think look awesome with this relaxed Domino theme). And because they are all SVG, I don't care about sizes, and they are all in the same place, and all is good.

And whenever Oxygen is released, all I need to do is switch the files around and that's that. Which is nice, too.

Of course there is a catch... it **does** look out of place, and I expect many to find it ugly. So what, since I am the only user of this app! ;-)

The Python [config objects](#) are convenient and simple, but they have a problem: you can only save strings. That means you need to store numbers as strings and remember to use the `getInt()/getfloat()` methods (or coerce by hand!), which is error prone and anti-pythonic. Storing a list is even uglier.

You could store ascii pickles, but those are pretty unpleasant to read in some cases.

Here's my solution: Encode it using a JSON encoder first! (I am using [demjson](#))

Silly obvious code fragment:

```
\ :k: 'def' \ \ :nf: 'getValue' \ \ :p: '(' \ \ :n: 'section' \ \ :p: ',' \ \ :n: 'ke
  \ :k: 'try' \ \ :p: ':' \
    \ :k: 'return' \ \ :n: 'JSON' \ \ :p: '()' \ \ :o: '.' \ \ :n: 'decode' \ \
  \ :k: 'except' \ \ :p: ':' \
    \ :k: 'return' \ \ :n: 'default' \

\ :k: 'def' \ \ :nf: 'setValue' \ \ :p: '(' \ \ :n: 'section' \ \ :p: ',' \ \ :n: 'ke
  \ :n: 'value' \ \ :o: '=' \ \ :n: 'JSON' \ \ :p: '()' \ \ :o: '.' \ \ :n: 'encode
  \ :k: 'try' \ \ :p: ':' \
    \ :n: 'r' \ \ :o: '=' \ \ :n: 'conf' \ \ :o: '.' \ \ :n: 'set' \ \ :p: '(' \ \
  \ :k: 'except' \ \ :n: 'ConfigParser' \ \ :o: '.' \ \ :n: 'NoSectionError' \ \
    \ :n: 'conf' \ \ :o: '.' \ \ :n: 'add_section' \ \ :p: '(' \ \ :n: 'section
    \ :n: 'r' \ \ :o: '=' \ \ :n: 'conf' \ \ :o: '.' \ \ :n: 'set' \ \ :p: '(' \ \
```

```

\ :n:'f'\ \ :o:'='\ \ :nb:'open'\ \ :p:'('\ \ :n:'os'\ \ :o:'.'\ \ :n:
\ :n:'conf'\ \ :o:'.'\ \ :n:'write'\ \ :p:'('\ \ :n:'f'\ \ :p:')'\
\ :k:'return'\ \ :n:'r'\

```

With just a little effort you can have a readable ascii typed python config file.

Took a while to implement, but BartleBlog finally got a functional menu editor:

Right now, it only works with the mootools-based menu gadget, but I will start working on the yahoo menu version in a moment.

The only thing not working is the preview button, because it needs more support on the backend side.

... isn't releasing software called exactly the same as another that exists previously, like, wrong?

[Adobe Flex](#)

[GNU Flex](#)

I am working on changing BartleBlog so it can be used from scratch. That may sound odd but because I have been using it since day 2 to post this blog, it has grown very organically, meaning there are things that only work because of the way I used it while developing it.

So, I created a test user, and created a test blog there, and I am working, and decide to do another from-scratch test, and...

I deleted my production copy.

Yes. The one that generates this blog. So this blog disappeared. Because I used the wrong terminal window.

And I had one-week old backups.

So I felt very very stupid.

Because undeleting in Linux is a joke.

So I was thinking how to spend a few hours recreating the last week of posts, and whatever, when I noticed on the taskbar... bartleblog was still running.

Which means that the DB was still open by a process. Which means...

```

\ :o:'[\ \ ralsina@monty bartleblog\ :o:']\ \ :nv:'$ '\ ps ax | grep python
17063 pts/1    S      24:33 python bartleblog.py
17161 ?        S      0:04 konqueror \ :o:'[\ \ kdeinit\ :o:']\ \ -mimetypes
17454 pts/1    D+     0:00 grep python
\ :o:'[\ \ ralsina@monty bartleblog\ :o:']\ \ :nv:'$ '\ su
Password:
\ :o:'[\ \ root@monty bartleblog\ :o:']\ \ \ :c:'# cd /proc/17063/fd
'\ \ :o:'[\ \ root@monty fd\ :o:']\ \ \ :c:'# ls
'\ 0 1 10 11 12 2 3 4 5 6 7 8 9
\ :o:'[\ \ root@monty fd\ :o:']\ \ \ :c:'# ls -l
'\ total 0
lrwx----- 1 ralsina users 64 2007-05-13 21:07 0 -> /dev/pts/1
lrwx----- 1 ralsina users 64 2007-05-13 21:07 1 -> /dev/pts/1
lrwx----- 1 ralsina users 64 2007-05-13 21:07 10 -> socket:\ :o:'[\ \ 15948
lrwx----- 1 ralsina users 64 2007-05-13 21:07 11 -> socket:\ :o:'[\ \ 15948
lrwx----- 1 ralsina users 64 2007-05-13 21:07 12 -> /mnt/centos/home/ralsin
lrwx----- 1 ralsina users 64 2007-05-13 21:07 2 -> /dev/pts/1
lr-x----- 1 ralsina users 64 2007-05-13 21:07 3 -> /mnt/centos/home/ralsin
lr-x----- 1 ralsina users 64 2007-05-13 21:07 4 -> pipe:\ :o:'[\ \ 159481\

```

```

l-wx----- 1 ralsina users 64 2007-05-13 21:07 5 -> pipe:\ :o:'[\ 159481\
lr-x----- 1 ralsina users 64 2007-05-13 21:07 6 -> pipe:\ :o:'[\ 159482\
l-wx----- 1 ralsina users 64 2007-05-13 21:07 7 -> pipe:\ :o:'[\ 159482\
lr-x----- 1 ralsina users 64 2007-05-13 21:07 8 -> pipe:\ :o:'[\ 159485\
l-wx----- 1 ralsina users 64 2007-05-13 21:07 9 -> pipe:\ :o:'[\ 159485\
\ :o:'[\ root@monty fd\ :o:']\ \ :c:'# cp 12 /root/db
'\ \ :o:'[\ root@monty fd\ :o:']\ \ :c:'# ls -l ~/db
'\ -rw-r--r-- 1 root root 3582976 2007-05-13 21:07 /root/db
\ :o:'[\ root@monty fd\ :o:']\ \ :c:'# sqlitebrowser ~/db
'\ \ :o:'[\ root@monty fd\ :o:']\ \ :c:'# cp ~/db /home/ralsina/.bartlebl

```

And I got the database back.

If you don't understand how that worked.... here's the explanation:

- On unix, files are **really** unlinked (removed from directories) when no process has them open. Even then, the data is not deleted, but finding it is much harder.
- On /proc/PID you can see the file descriptors each process has open.
- You can actually copy a file descriptor.

So I went and copied the open file. And got it back. And this blog didn't go away. So I am lucky! Stupid. But lucky!

Since the very beginning, BartleBlog has been using [CherryTemplate](#) for its output formatting needs. I like it, because it's very simple.

However, it had grown rather cumbersome.

Specifically, most pages in a blog are sort of a page template with a body template inside (the main content).

To do that on CherryTemplate, I used a two-pass approach: generate the body, then pass it as parameter to the page template.

Which is a pain in some cases because you end basically having to do a rendering function for each kind of page, or some crazy-evil function (what I did).

Exploring the different python template engines, I ran into [Mako](#) and decided to give it a whirl. It looks good.

The approach is a bit different, it is much more powerful, but you can still use it simply if you can.

And the main feature was template inheritance. Using that, no more inner and outer templates, baby!

Oh, and performance is better:

Cherry

```

real    31m44.732s
user    21m18.336s
sys     2m7.628s

```

Mako

```

real    24m54.472s
user    19m9.508s
sys     1m56.375s

```

This is for completely re-rendering the whole 7 years, 574 posts, 40 static articles, 14 category blog, and there is tons of optimizations to be done.

BTW: this is how you re-render the whole blog:

```
\ :kn:'from' \ \ :nn:'BartleBlog.backend.blog' \ \ :kn:'import' \ \ :n:'Blog' \ \ :n:'Blog' \ \ :p:'()' \ \ :o:'. ' \ \ :n:'renderFullBlog' \ \ :p:'()' \
```

I was thinking: how can I implement page previews in BartleBlog?

The obvious way is to render the page and open the local file. However, the page may link or include pieces that are not updated yet in the static version, so that can give confusing results.

Then it hit me... generate the page on the fly and serve it. And do the same for everything else the browser asks for.

So, after searching for 15 minutes for the simplest python “web framework” that let me use the code already in Bartleblog, and deciding for [Colubrid](#)...

Now, this is cute: bartleblog as a dynamic web app in 34 lines.

```
\ :kn:'from' \ \ :nn:'colubrid' \ \ :kn:'import' \ \ :n:'RegexApplication' \ \ :n:'Blog' \ \ :n:'Blog' \ \ :p:'()' \ \ :o:'. ' \ \ :n:'renderFullBlog' \ \ :p:'()' \ \ :kn:'import' \ \ :nn:'BartleBlog.backend.blog' \ \ :kn:'import' \ \ :n:'Blog' \ \ :kn:'import' \ \ :nn:'BartleBlog.backend.dbclasses' \ \ :kn:'as' \ \ :nn:'db' \ \ :kn:'import' \ \ :nn:'os' \ \ :o:', ' \ \ :nn:'codecs' \ \ :k:'class' \ \ :nc:'webBlog' \ \ :p:'(\ \ :n:'Blog' \ \ :p:'):' \ \ :k:'def' \ \ :nf:'__init__' \ \ :p:'(\ \ :bp:'self' \ \ :p:'):' \ \ :n:'Blog' \ \ :o:'. ' \ \ :n:'__init__' \ \ :p:'(\ \ :bp:'self' \ \ :n:'basepath' \ \ :o:=' ' \ \ :s:'http://localhost' \ \ :bp:'self' \ \ :o:'. ' \ \ :n:'basepath' \ \ :o:=' ' \ \ :n:'os' \ \ :o:'. ' \ \ :k:'if' \ \ :ow:'not' \ \ :n:'os' \ \ :o:'. ' \ \ :n:'path' \ \ :o:'. ' \ \ :n:'os' \ \ :o:'. ' \ \ :n:'mkdir' \ \ :p:'(\ \ :bp:'self' \ \ :o:'. ' \ \ :k:'class' \ \ :nc:'MyApplication' \ \ :p:'(\ \ :n:'RegexApplication' \ \ :n:'blog' \ \ :o:=' ' \ \ :n:'webBlog' \ \ :p:'()' \ \ :n:'urls' \ \ :o:=' ' \ \ :p:'[\ \ :p:'(\ \ :s:'r'^(.*?)$' \ \ :p:', ' \ \ :s:'page' \ \ :p:')', ' \ \ :p:'(\ \ :s:'r'^(.*?) / (.*?)$' \ \ :p:', ' \ \ :s:'page' \ \ :p:')', ' \ \ :p:'(\ \ :s:'r'^(.*?) / (.*?) / (.*?)$' \ \ :p:', ' \ \ :s:'page' \ \ :p:')', ' \ \ :p:'(\ \ :s:'r'^(.*?) / (.*?) / (.*?) / (.*?)$' \ \ :p:', ' \ \ :s:'page' \ \ :p:')' \ \ :p:']' \ \ :k:'def' \ \ :nf:'page' \ \ :p:'(\ \ :bp:'self' \ \ :p:', ' \ \ :o:'*' \ \ :n:'path' \ \ :o:=' ' \ \ :s:'/' \ \ :o:'. ' \ \ :n:'join' \ \ :p:'(\ \ :n:'page' \ \ :o:=' ' \ \ :n:'db' \ \ :o:'. ' \ \ :n:'pageByPath' \ \ :p:'/' \ \ :bp:'self' \ \ :o:'. ' \ \ :n:'blog' \ \ :o:'. ' \ \ :n:'renderPage' \ \ :p:'/' \ \ :k:'return' \ \ :n:'HttpResponse' \ \ :p:'(\ \ :n:'codecs' \ \ :o:'. ' \ \ :n:'app' \ \ :o:=' ' \ \ :n:'MyApplication' \ \ :n:'app' \ \ :o:=' ' \ \ :n:'StaticExports' \ \ :p:'(\ \ :n:'app' \ \ :p:', ' \
```

```
\ :s: '/static' \ \ :p: ': ' \ \ :s: './static' \ \
\ :p: ')) \ \
```

```
\ :k: 'if' \ \ :n: '__name__' \ \ :o: '==' \ \ :s: '__main__' \ \ \ :p: ': ' \ \
\ :n: 'execute' \ \ :p: '(' \ \ :n: 'app' \ \ :p: ') ' \ \
```

Kmail searches slowly.

I have been a kmail user for a couple of years, since I settled in my own home office with my own computer.

I like the thing.

However, it annoys me that it takes sooooo long to find a message in my mailstore. Hey, it's only 13000 messages!

So, while I wait for KDE4 to bring all its searching goodness, I decided to see if I could hack something quickly.

Enter [mairix](#): a mail indexer/search thing.

Convincing mairix to index all my mail was rather simple (here is my ~/.mairixrc):

```
base=/home/ralsina/
maildir=Mail/*...
maildir=.kde/share/apps/kmail/dimap...
omit=Mail/mairix
database=~/.mairix_db
```

What does it do?

- It indexes mail stored at ~/Mail and everywhere in my kmail imap folders.
- It stores search results in ~/Mail/mairix and ignores its contents when searching. The results are stored as links, so they waste no disk space.

After running mairix so it builds its DB (took about a minute, which is less than most kmail searches), you can search for things like this:

```
[ralsina@monty ~]$ time mairix b:bartleblog
Matched 6 messages

real    0m0.232s
user    0m0.012s
sys     0m0.204s
```

And the result can be seen in kmail, in the mairix folder:

However, there is a problem. It will work for the first search, but not for the second one. On the second search, you get the same content listing, but all messages appear empty.

That's because kmail saves an index file of each folder. To work around that, I wrote a little shell wrapper, kmairix:

```
#!/bin/sh
rm ~/Mail/.mairix* -f
mairix $*
```

And you use that instead of calling mairix directly.

But there are still improvements to be done. If your kmail is currently displaying the mairix folder, searching doesn't update the message list.

DCOP to the rescue! We can switch to the inbox, then back to mairix (adjust as needed for yourself):

```
dcop kmail KMailIface selectFolder /Local/inbox
dcop kmail KMailIface selectFolder /Local/mairix
```

Missing pieces:

- How about switching to the kmail window? Sadly, the kwin DCOP interface seems incomplete. Maybe assigning kmail a hotkey and work from there? Let me know if you have any ideas.

UPDATE as suggested by Anno Heimburg: just call kmail.

- A GUI (of course!) probably with a tray icon...
- A way to auto-update the Mairix DB when new mail arrives. I am thinking about doing it with [incron](#) but have not done it yet.

So, here is the final version, put it somewhere in your path, and use ALT+F2 to search your mails :-)

```
#!/bin/sh
rm ~/Mail/.mairix* -f
mairix $*
dcop kmail KMailIface selectFolder /Local/inbox
dcop kmail KMailIface selectFolder /Local/mairix
kmail
```

That's the way those movies are meant to be seen.

On monday I had my first IMAX experience (the first IMAX in Argentina opened not long ago).

The rumble in my guts in the first big action scene... having to move the head to follow the action on the last one... really an intense cinematic experience.

I liked the second movie better, and everyone seemed a bit tired of their roles, but a fun movie.

If you are going to do popcorn, you better do great popcorn, and that's what Raimi does.

Ok, I give up. I need help.

The magical sometimes-it-works-on-IE-sometimes-doesn't overflow:auto CSS property has kicked my butt.

It worked until a while ago, and now it doesn't.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

Can anyone tell me why the grey preformatted thing [here](#) has scrollbars on Firefox and Konqueror, but not on IE?

In the past, I thought that was because it didn't work inside tables, but now there are no tables in the layout.

So, I am clueless :-)

system-message

WARNING/2 in `everything.txt`, line 13724
Duplicate explicit target name: "here".

UPDATE Thanks for all the suggestions, it kinda works now :-)

At least for me, since I have very little of it.

Today I went to a meeting with a prospective client, and we were talking with two people, one guy, apparently manager of the company, and a girl, in charge of the security area.

Blah, blah, yes, I teach for peanuts, no, no, more peanuts, and so on, then she tells me, she knows me from somewhere.

La Plata... CACIC?

Ok, she had been, in 1996 or 1997, in the audience for a lecture I gave on hierarchical proxying, in a congress called CACIC, in La Plata.

She even remembered I was at the Universidad Nacional del Litoral at the time.

I just remember that I did give that lecture at that time, and that it didn't work out very well (I redid it the previous night because I lost my notes, it was a mess).

How the hell can she remember that? I have no idea.

I just released a wee piece of software, called [RA-WebPass](#) which is simply a web-page that you can use to change linux system passwords.

Basically, I wrote it today out of my frustration with customers asking me "how can I change the password for the FTP server". Ok, this is how you can.

It's simple and doesn't have too many dependencies, so it should be rather secure, but don't trust me on that, and you need to run it as root, so be very very careful.

[RA-WebPass home page](#)

On a related note: it's quite satisfying to write something in two hours and just release it :-)

According to [Alexa.com](#) this site (<http://lateral.netmanagers.com.ar>) is the 682768th most popular site in the internet this week.

Also, it's the 50998th most popular site in Argentina and most suprisingly... the 10180th most popular site in Slovenia!

In fact, a whole **18.5%** of my readers come from Slovenia. Which is not surprising. It's stunning!

I hereby salute all my slovenian readers and declare myself a Beno Udrih fan.

[All the data.](#)

Ignore this link below, please.

I have been an [Arch Linux](#) for a while now, and I am still liking it.

Here's the good side of it:

- It's small (one CD)
- It's simple (it comes with very little)

system-message

• **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "aur".

It has a decent package selection (if you consider [AUR](#), more about that later)

- It uses pretty much unpatched upstream software

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "aur".

It's a binary distro (except for [AUR](#). Again, more about it later)

- It's pretty stable (no crashes I can remember)
- It has rolling releases (unlike, say, Fedora or Debian)
- It's easy to keep updated (like all of them nowadays)
- It's not ideologically dogmatic, but pragmatic (yes, there are NVidia drivers, and test-drive games, and whatever)
- It doesn't seem to be a one-guy joint

And the bad side:

- Updates sometimes break things (about twice a year)
- Admin tools are between unexistent and disjointed

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "aur".

And of course, there is the very very good side: [AUR](#)

[AUR](#) is a community repository. And there is a rather large community. And packaging things for Arch is so easy, and putting things in [AUR](#) is so simple, even I find time to contribute ([my packages](#)).

And it's a calm community, and pretty much, instead of compiling my random unknown packages for myself, I save the steps to build them and stick them in a PKG-BUILD and upload them. Takes two minutes for most things.

It's a throwback to the old days of Linux: quiet, competent (or learning) people doing things, sharing, you use them, you give back... I had not felt that way with a distro for years.

system-message

- **WARNING/2** in `everything.txt`, line 13823
Duplicate explicit target name: "aur".

Since I am trying to make my blog into a serious site^[1] I decided to take a serious look into web typography. It should be useful if BartleBlog ever gets a second user (which is not precisely coming soon ;-)

Specially since [this link](#) appeared in [Ned Batchelder's blog](#).

Right now, I am into basic things, like vertical rythm and such, but you can already see a difference:

Now, let's go into more [serious material](#) to read on the subject.

On 2004 I wrote a [spreadsheet](#) in python, which was about a 25KB [download](#) (compressed). It was pretty [functional!](#).

1

The main problem that would never let that program be a *real* application was that it used python as a formula language instead of the traditional Lotus/Excel thing.

Yesterday I decided to look at it again, and see if there was a way around it[1]. My main asset was that I know a fair bit more about parsers now than I did then, and *how hard could it be?*

Well, it turns out that language is a whole lot harder than I knew!

I found this page [about a parser](#)

Here's an example from it:

```
=IF (R[39]C[11]>65,R[25]C[42],ROUND((R[11]C[11]*IF(OR(AND(R[39]C[11]>=55,R[40]C[11]>=20),AND(R[40]C[11]>=20,R11C3="YES")),R[44]C[11],R[43]C[11]))+(R[14]C[11]*IF(OR(AND(R[39]C[11]>=55,R[40]C[11]>=20),AND(R[40]C[11]>=20,R11C3="YES")),R[45]C[11],R[43]C[11])),0))
```

Yes, that's a valid Excel formula. No, I have no idea what it does.

And here's some extra nuggets about the language (quoted from the same page):

- For some reason Excel uses a comma as the range union operator. This means that its only unambiguous use between ranges is outside of a function call or within a subexpression (i.e., between parentheses).
- For some other reason Excel uses a space (or multiple spaces) as the range intersection operator. While not as ambiguous as the comma, it does require some consideration.

And it goes on, and on, and on....

But hey, he wrote a nice parser!

Just for kicks, here's [the best BNF I could find](#).

I really don't envy the KSpread authors if they try to be compatible to this garbage!

But *why* is this language so gnarly?

- Because it has been growing organically for 30 years?
- Because of backwards compatibility?
- Because it was never designed?
- All of the above?

I mean, it can't be intentional! Noone actually *meant* it to be like this, right?

In 2004 I saw a [recipe](#) about how to make a "spreadsheet" in python in 10 lines of code:

```
system-message
WARNING/2 in everything.txt, line 13847
Duplicate explicit target name: "1".
```

as in "excessive or impressive in quality, quantity, extent, or degree", not "not joking or trifling". This site is all about trifling.

1

```
system-message
WARNING/2 in everything.txt, line 13907
Duplicate explicit target name: "1".
```

Yes, I routinely look at things I wrote and haven't touched in 2.4 years.

```

\ :k:'class'\ \ :nc:'SpreadSheet'\ \ :p:':'\
\ :n:'_cells'\ \ :o:'='\ \ :p:'{'}\
\ :n:'tools'\ \ :o:'='\ \ :p:'{'}\
\ :k:'def'\ \ :nf:'__setitem__'\ \ :p:'('\ \ :bp:'self'\ \ :p:','\ \
\ :bp:'self'\ \ :o:'. '\ \ :n:'_cells'\ \ :p:'['\ \ :n:'key'\ \ :p:
\ :k:'def'\ \ :nf:'getformula'\ \ :p:'('\ \ :bp:'self'\ \ :p:','\ \
\ :k:'return'\ \ :bp:'self'\ \ :o:'. '\ \ :n:'_cells'\ \ :p:'['\ \
\ :k:'def'\ \ :nf:'__getitem__'\ \ :p:'('\ \ :bp:'self'\ \ :p:','\ \
\ :k:'return'\ \ :nb:'eval'\ \ :p:'('\ \ :bp:'self'\ \ :o:'. '\ \

```

It's shocking. And it works, too:

```

\ :o:'>>>'\ \ :kn:'from'\ \ :nn:'math'\ \ :kn:'import'\ \ :n:'sin'\ \ :
\ :o:'>>>'\ \ :n:'SpreadSheet'\ \ :o:'. '\ \ :n:'tools'\ \ :o:'. '\ \ :n:'up
\ :o:'>>>'\ \ :n:'ss'\ \ :o:'='\ \ :n:'SpreadSheet'\ \ :p:'() '\
\ :o:'>>>'\ \ :n:'ss'\ \ :p:'['\ \ :s:'a1'\ \ :p:'] '\ \ :o:'='\ \ :s:
\ :o:'>>>'\ \ :n:'ss'\ \ :p:'['\ \ :s:'a2'\ \ :p:'] '\ \ :o:'='\ \ :s:
\ :o:'>>>'\ \ :n:'ss'\ \ :p:'['\ \ :s:'a3'\ \ :p:'] '\ \ :o:'='\ \ :s:
\ :o:'>>>'\ \ :n:'ss'\ \ :p:'['\ \ :s:'a3'\ \ :p:'] '\
\ :mi:'210 '\
\ :o:'>>>'\ \ :n:'ss'\ \ :p:'['\ \ :s:'b1'\ \ :p:'] '\ \ :o:'='\ \ :s:
\ :o:'>>>'\ \ :n:'ss'\ \ :p:'['\ \ :s:'b1'\ \ :p:'] '\
\ :mf:'0.70710678118654746 '\
\ :o:'>>>'\ \ :n:'ss'\ \ :o:'. '\ \ :n:'getformula'\ \ :p:'('\ \ :s:'b1'\
\ :s:'sin(pi/4)'\

```

I was so awed, I wrote [a PyQt version](#) . Of course there is a catch in that code: it totally sucks if you are trying to write a spreadsheet with it.

Why? Because it doesn't store results, but only formulas.

For example:

```

A1=2
A2=A1*2

```

If you ask for the value of A2, you get 4. If you set A1 to 7, what's the value of A2?

Well, it's nothing yet, because it's only calculated when you ask for it. But suppose you are trying to **display** that sheet... you need to know A2's value changed when you set A1!

That's cell dependencies, and while that simple code handles them in a way, it totally sucks in another.

So, I went ahead and [coded around it](#) successfully. Of course [the code was not so pretty anymore](#) (although a large part of the ugliness is just for making it work with Python 2.3 and relative cells).

Then yesterday, while looking at the [excel formula parser madness](#) I saw a reference to [PyCells](#), a python port of [Cells](#) from CLOS.

Here is a [blog commenting on Pycells](#):

It basically takes the concept of a cell in a spreadsheet that get updated automatically to programming where there are a lot of internal data states

that are dependent on one another in a chain, or a complex graph of dependencies. Like, the color of a button depends on whether you selected a radio button or not. Or, shut down the motor if the sensor reads above 100 degrees (example given in text).

Almost everyone uses that analogy... however, no matter how hard I looked, I couldn't find anyone who had actually tried writing a spreadsheet using PyCells! Not even as an example!

So here it is:

```
\ :kn:'import' \ \ :nn:'cells' \

\ :k:'class' \ \ :nc:'Cell' \ \ :p:'(' \ \ :n:'cells' \ \ :o: '.' \ \ :n:'Model' \ \ :n:'formula' \ \ :o:'=' \ \ :n:'cells' \ \ :o: '.' \ \ :n:'makecell' \ \ :p:'@cells.fun2cell' \ \ :p:'()' \
\ :k:'def' \ \ :nf:'value' \ \ :p:'(' \ \ :bp:'self' \ \ :p:', ' \ \ :n:'pre' \ \ :k:'print' \ \ :s:'"eval "' \ \ :p:', ' \ \ :bp:'self' \ \ :o: '.' \ \ :k:'return' \ \ :nb:'eval' \ \ :p:'(' \ \ :bp:'self' \ \ :o: '.' \ \ :k:'def' \ \ :nf:'__init__' \ \ :p:'(' \ \ :bp:'self' \ \ :p:', ' \ \ :n:'ss' \ \ :bp:'self' \ \ :o: '.' \ \ :n:'ss' \ \ :o:'=' \ \ :n:'ss' \ \ :n:'cells' \ \ :o: '.' \ \ :n:'Model' \ \ :o: '.' \ \ :n:'__init__' \ \

\ :k:'class' \ \ :nc:'ssDict' \ \ :p: ':' \
\ :k:'def' \ \ :nf:'__init__' \ \ :p:'(' \ \ :bp:'self' \ \ :p:')' \
\ :bp:'self' \ \ :o: '.' \ \ :n:'ss' \ \ :o:'=' \ \ :p:'{' \

\ :k:'def' \ \ :nf:'__getitem__' \ \ :p:'(' \ \ :bp:'self' \ \ :p:', ' \
\ :k:'return' \ \ :bp:'self' \ \ :o: '.' \ \ :n:'ss' \ \ :p:'[' \

\ :k:'def' \ \ :nf:'__setitem__' \ \ :p:'(' \ \ :bp:'self' \ \ :p:', ' \
\ :k:'if' \ \ :ow:'not' \ \ :bp:'self' \ \ :o: '.' \ \ :n:'ss' \
\ :n:'c' \ \ :o:'=' \ \ :n:'Cell' \ \ :p:'(' \ \ :bp:'s' \
\ :n:'c' \ \ :o: '.' \ \ :n:'formula' \ \ :o:'=' \ \ :n:'ss' \ \ :bp:'self' \ \ :o: '.' \ \ :n:'ss' \ \ :p:'[' \ \ :n:'else' \ \ :p: ':' \
\ :bp:'self' \ \ :o: '.' \ \ :n:'ss' \ \ :p:'[' \ \ :n:'

\ :k:'if' \ \ :n:'__name__' \ \ :o:'==' \ \ :s:'"__main__"' \ \ :p: ':' \
\ :n:'ss' \ \ :o:'=' \ \ :n:'ssDict' \ \ :p:'()' \
\ :n:'ss' \ \ :p:'[' \ \ :s:'a1' \ \ :p:']' \ \ :o:'=' \ \ :s:'5' \

\ :n:'ss' \ \ :p:'[' \ \ :s:'a2' \ \ :p:']' \ \ :o:'=' \ \ :s:'2*a1' \
\ :k:'print' \ \ :s:'"a1: "' \ \ :p:', ' \ \ :n:'ss' \ \ :p:'[' \ \ :s:'a2: "' \ \ :p:', ' \ \ :n:'ss' \ \ :p:'[' \ \ :s:'a1' \ \ :p:']' \ \ :o:'=' \ \ :s:'7' \
\ :k:'print' \ \ :s:'"a1: "' \ \ :p:', ' \ \ :n:'ss' \ \ :p:'[' \ \ :s:'a2: "' \ \ :p:', ' \ \ :n:'ss' \ \ :p:'[' \ \ :s:'
```

And here you can see it running:

```
[ralsina@monty cells]$ python ctest.py
a1: eval 5
5
a2: eval 2*a1
10
eval 7
eval 2*a1
a1: 7
a2: 14
```

See how when I set a1 to 7 I get “eval 7” and “eval 2*a1”? That’s because it’s propagating changes the right way. And that’s why this would work as a basis for a spreadsheet.

UPDATE

It seems there is a bug wither in PyCells, or in my example, or something, because it breaks pretty easily, if the dependency chain is even two cells:

```
a1: eval 5
5
a2: eval 2*a1
10
a3: eval 2*a2
20
eval 7
eval 2*a1
a1: 7
a2: 14
a3: 20
```

In this example, I am setting A3 to 2*A2, and when I update A1, A3 is not updated. Further research is needed.

UPDATE 2

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “comments”.

Please check the reddit [comments](#). They are very educational!

system-message

WARNING/2 in everything.txt, line 14082
Duplicate explicit target name: “comments”.

I apologize in advance for any ugly amateurism in this post. It’s my first attempt at a domain specific language :-)

Yesterday I posted about using PyCells to write a spreadsheet in Python.

Sadly, I can't figure out the problem with my code, and the PyCells mailing list seems to be pretty much dead.

So, I started thinking... what other ways are to achieve my goal? And decided to go medieval on this problem.

By that I mean that I will do it the most traditional way possible... with a twist.

The traditional way is, of course, to write one or more of lexer/parser/interpreter/compiler for the formula language.

Mind you, I don't intend to do anything complete, much less Excel-compatible (see [Excel formula parsers are hell](#) in this same blog).

So, let's start with a toy language, supporting the following:

- Assignment to a variable
- Classic 4-op arithmetics.
- Function calls
- Cell ranges

That's enough for a toy spreadsheet, and it should be easy to extend.

Here's a description of the grammar for such a language, written using [Aperiot\[1\]](#):

```
# This is a simple language for arithmetic expressions

numbers
  number

operators
  plus "+"
  times "*"
  minus "-"
  div "/"
  equal "="
  colon ":"
  comma ","
  semicolon ";"

brackets
  lpar "("
  rpar ")"

identifiers
  label

start
  LIST

rules
LIST          -> ASSIGNMENT          : "[${1}]"
```

```

| ASSIGNMENT semicolon LIST : "$1]+$3"
| ASSIGNMENT semicolon : "$1]"

ASSIGNMENT -> label equal EXPR : "($1,$3)"

ARGLIST -> ARG comma ARGLIST : "$1]+$3"
| ARG : "$1]"

ARG -> RANGE : "$1"
| EXPR : "$1"
| label : "$1"

EXPR -> TERM : "$1"
| TERM plus EXPR : "(\'+\',$1,$3)"
| TERM minus EXPR : "(\'-\',$1,$3)"

TERM -> FACTOR : "$1"
| FACTOR times TERM : "(\'*\',$1,$3)"
| FACTOR div TERM : "(\'/\',$1,$3)"

FACTOR -> number : "$1.val()"
| lpar EXPR rpar : "(\`group\',$2)"
| FUNCALL : "$1"
| label : "$1"
| minus FACTOR : "-$2"

FUNCALL -> label lpar ARGLIST rpar : "(\`funcall\',$1,$3)"

RANGE -> label colon label : "(\`range\',$1,$3)"

```

This transforms this:

```

A1=SUM(A1:A7)*2;
A3=2+2;

```

Into this:

```

[(<aperiot.lexer.Identifier instance at 0xb7af10ac>,
 ('*',
 ('funcall',
 <aperiot.lexer.Identifier instance at 0xb7af142c>,
 [ ('range',
 <aperiot.lexer.Identifier instance at 0xb7af15cc>,
 <aperiot.lexer.Identifier instance at 0xb7af144c>)]),
 2)),
 (<aperiot.lexer.Identifier instance at 0xb7b4c72c>, ('+', 2, 2))]

```

Which is sort of a tree with all the expressions in prefix notation in them.

Now, here is the twist: I will “compile” this tree into.... python code. So I can use eval to do the evaluation, just like in the original [python spreadsheet recipe](#).

So this is sort of a preprocessor:

- The user writes excel-like formulas.
- The spreadsheet stores python code obtained through compilation.
- The spreadsheet evals the python code.

Of course we have the same problem as usual: cell dependencies, which is the reason why I started playing with PyCells in the first place!

But... well, here's another trick: since I am compiling, I know whenever there is a variable referenced in the code. And I can remember them :-)

So, I can turn this:

```
A1=SUM(A1:A3)*2;
A3=2+2;
```

Into this:

```
[['A1=SUM(a1,a2,a3)*2;', set(['a1', 'a3', 'a2'])],
 ['A3=2+2;', set([])]]
```

The “compiled” python code and a dependency set. And voila, this spreadsheet **will** propagate correctly.

Here's the compiler... in about 60 lines of python^[2]. And since the whole point of this language is to track dependencies... let's call it Traxter.

Of course, this is a toy right now. But it's a toy with potential!

```
\ :kn:'from'\ \ :nn:'pprint'\ \ :kn:'import'\ \ :n:'pprint'\
\ :kn:'from'\ \ :nn:'apertiot.parsergen'\ \ :kn:'import'\ \ :n:'build_par
\ :kn:'import'\ \ :nn:'apertiot'\
\ :kn:'import'\ \ :nn:'cellutils'\
\ :kn:'import'\ \ :nn:'sys'\

\ :n:'dependencies'\ \ :o:'='\ \ :nb:'set'\ \ :p:'()' \

\ :k:'def'\ \ :nf:'addOp'\ \ :p:'(\ \ :o:'*\ \ :n:'args'\ \ :p:'):\
\ :k:'return'\ \ :s:'+' \ \ :o:'.\ \ :n:'join'\ \ :p:'([\ \ :n:
\ :k:'def'\ \ :nf:'mulOp'\ \ :p:'(\ \ :o:'*\ \ :n:'args'\ \ :p:'):\
\ :k:'return'\ \ :s:'*\ \ \ :o:'.\ \ :n:'join'\ \ :p:'([\ \ :n:
\ :k:'def'\ \ :nf:'subOp'\ \ :p:'(\ \ :o:'*\ \ :n:'args'\ \ :p:'):\
\ :k:'return'\ \ :s:'-' \ \ :o:'.\ \ :n:'join'\ \ :p:'([\ \ :n:
\ :k:'def'\ \ :nf:'divOp'\ \ :p:'(\ \ :o:'*\ \ :n:'args'\ \ :p:'):\
\ :k:'return'\ \ :s:'/' \ \ :o:'.\ \ :n:'join'\ \ :p:'([\ \ :n:

\ :k:'def'\ \ :nf:'groupOp'\ \ :p:'(\ \ :o:'*\ \ :n:'args'\ \ :p:'):\
\ :k:'return'\ \ :s:'(' \ \ :si:'%s' \ \ :s:')' \ \ :o:'%' \ \ :n:'

\ :k:'def'\ \ :nf:'funcOp'\ \ :p:'(\ \ :o:'*\ \ :n:'args'\ \ :p:'):\
\ :k:'return'\ \ :s:'\ \ \ :si:'%s' \ \ :s:'(\ \ \ :si:'%s' \ \ :s:')'
\ :s:','\ \ \ :o:'.\ \ \ :n:'join'\ \ :p:'([\ \ \ :n:

\ :k:'def'\ \ :nf:'rangeOp'\ \ :p:'(\ \ :o:'*\ \ :n:'args'\ \ :p:'):\
```

```

\ :n:'c1'\ \ :o:'='\ \ :n:'args'\ \ :p:'[\ \ \ :mi:'0'\ \ :p:]'\ \
\ :n:'c2'\ \ :o:'='\ \ :n:'args'\ \ :p:'[\ \ \ :mi:'1'\ \ :p:]'\ \
\ :k:'return'\ \ :s:','\ \ :o:'.'\ \ :n:'join'\ \ :p:'([\ \ \ :n:

\ :n:'operators'\ \ :o:'='\ \ :p:'{\ \ \ :s:''+' \ \ :p:':\ \ :n:'addOp'\ \
\ :s:''-' \ \ :p:':\ \ :n:'subOp'\ \ :p:','\ \
\ :s:''*'\ \ :p:':\ \ :n:'mulOp'\ \ :p:','\ \
\ :s:''/' \ \ :p:':\ \ :n:'divOp'\ \ :p:','\ \
\ :s:''group'\ \ :p:':\ \ :n:'groupOp'\ \ :p:','\ \
\ :s:''funcall'\ \ :p:':\ \ :n:'funcOp'\ \ :p:','\ \
\ :s:''range'\ \ :p:':\ \ :n:'rangeOp'\ \
\ :p:'}'\ \

\ :k:'def'\ \ :nf:'compile_token'\ \ :p:'(\ \ \ :n:'token'\ \ :p:):'\ \
\ :k:'if'\ \ \ :nb:'isinstance'\ \ \ :p:'(\ \ \ :n:'token'\ \ \ :p:','\ \
\ \ :n:'v'\ \ \ :o:'='\ \ \ :n:'token'\ \ \ :o:'.'\ \ \ :n:'symbolic_
\ \ :n:'dependencies'\ \ \ :o:'.'\ \ \ :n:'add'\ \ \ :p:'(\ \ \ :n:
\ \ :k:'return'\ \ \ :n:'v'\ \
\ :k:'if'\ \ \ :nb:'isinstance'\ \ \ :p:'(\ \ \ :n:'token'\ \ \ :p:','\ \
\ \ :k:'return'\ \ \ \ :nb:'apply'\ \ \ :p:'(\ \ \ :n:'operators'\ \ \ :p:
\ \ :k:'return'\ \ \ \ :nb:'str'\ \ \ :p:'(\ \ \ :n:'token'\ \ \ :p:')'\ \

\ :k:'def'\ \ \ :nf:'compile_assignment'\ \ \ :p:'(\ \ \ :n:'tokens'\ \ \ :p:):'\ \
\ \ :n:'target'\ \ \ :o:'='\ \ \ :n:'tokens'\ \ \ :p:'[\ \ \ :mi:'0'\ \ \ :p:
\ \ :n:'compiled'\ \ \ :o:'='\ \ \ :n:'compile_token'\ \ \ :p:'(\ \ \ :n:'to
\ \ :k:'return'\ \ \ :s:'''\ \ \ :si:'%s'\ \ \ :s:'='\ \ \ :si:'%s'\ \ \ :s:')'

\ :n:'myparser'\ \ \ :o:'='\ \ \ :n:'build_parser'\ \ \ :p:'(\ \ \ :s:''traxter'
\ :n:'t'\ \ \ :o:'='\ \ \ :s:''A1=SUM(A1:A7)*2;A3=2+2;' \ \
\ :n:'assign_list'\ \ \ :o:'='\ \ \ :n:'myparser'\ \ \ :o:'.'\ \ \ :n:'parse'\ \ \ :p:
\ :n:'pprint'\ \ \ :p:'(\ \ \ :n:'assign_list'\ \ \ :p:')'\ \

\ :n:'compiled'\ \ \ :o:'='\ \ \ :p:'[]'\ \
\ :k:'for'\ \ \ :n:'assignment'\ \ \ :ow:'in'\ \ \ :n:'assign_list'\ \ \ :p:':'\ \
\ \ :n:'dependencies'\ \ \ :o:'='\ \ \ :nb:'set'\ \ \ :p:'()\ \ \
\ \ :n:'c'\ \ \ :o:'='\ \ \ :n:'compile_assignment'\ \ \ :p:'(\ \ \ :n:'assi
\ \ :n:'compiled'\ \ \ :o:'.'\ \ \ :n:'append'\ \ \ :p:'([\ \ \ :n:'c'\ \ \ :p:

\ :k:'print'\ \ \ :n:'compiled'\ \

```

1

system-message

WARNING/2 in everything.txt, line 14309
Duplicate explicit target name: "1".

You may be asking yourself: what the heck is Aperiot? Or Why the heck Aperiot? Well... I had never heard of it until 6 hours ago, and I just wrote a DSL using it. That means it's worth knowing.

2

Adding something else to my plate is probably not a very good idea, but what the heck, I can make it sleep another three years if I lose interest.

So: I am writing a *real* spreadsheet in python.

Probably never going to be useful for corporations, but it should be at least as featureful as Google's and it should be amazingly small.

Here are the components:

- Traxter: my spreadsheet-formula-like-language with dependency tracking that compiles to python.
- PyQt (hey, it has a grid widget)
- Python (Of course)

The status right now:

- It's almost as functional as it was 2.5 years ago
- Except for broken relative cells.
- But with the beginning of a real formula language.
- With automatic recalculation and cyclic dependency checks.
- Adding things is dead simple. Here's the implementation of SUM (not uploaded yet, though):

```
\ :k:'def'\ \ :nf:'sum'\ \ :p:'(\ \ :o:'*\ \ \ :n:'args'\ \ :p:'):\ \
\ :n:'ac'\ \ :o:'='\ \ :mi:'0'\ \
\ :k:'for'\ \ :n:'v'\ \ :ow:'in'\ \ \ :n:'args'\ \ :p:':'\ \
\ :n:'ac'\ \ :o:'+='\ \ :n:'v'\ \
\ :k:'return'\ \ \ :n:'ac'\ \
```

All the range stuff happens behind the scenes (although you may get a function called with thousands of args... I wonder how well Python handles that).

You can check it at the [google code project](#) (Use the SVN).

After hacking for about two hours the cell dependencies yesterday using dicts, I found myself saying "how can I check if the dependency graph is cyclical?"

And of course it hit me. That's the dependency **graph**. Use a graph library!

I looked for about 2 minutes before finding one that's probably not optimal, but is damn simple and in the public domain: [graph_lib.py](#) by Nathan Denny.

First interesting data point, the first two lines on the file (yes, I found out there is a later version):

```
!--Version 1.0.0
!--Nathan Denny, May 27, 1999
```

system-message

WARNING/2 in `everything.txt`, line 14311
Duplicate explicit target name: "2".

cellrange() is left as an exercise for the reader because my current implementation is shameful ;-)

Yup. In two days this piece of code is turning 8 years old and untouched. But it works just fine and dandy!

Here's the piece of code that makes the engine run which has grown from a [humble 10 LOC](#) to almost a whooping 40 LOC:

```
\ :k:'class'\ \ :nc:'SpreadSheet'\ \ :p:'('\ \ :n:'QtCore'\ \ :o:'. '\ \ :n:
\ :n:'_cells'\ \ :o:'='\ \ \ :p:'{'}\ \
\ :n:'tools'\ \ \ :o:'='\ \ \ :p:'{'}\ \
\ :n:'graph'\ \ \ :o:'='\ \ \ :n:'Graph'\ \ \ :p:'() '\ \

\ :k:'def'\ \ \ :nf:'__init__'\ \ \ :p:'('\ \ \ :bp:'self'\ \ \ :p:', '\ \ \ :n:
\ \ :n:'QtCore'\ \ \ :o:'. '\ \ \ :n:'QObject'\ \ \ :o:'. '\ \ \ :n:'__init__'\ \
\ \ :k:'for'\ \ \ :n:'name'\ \ \ :ow:'in'\ \ \ :nb:'dir'\ \ \ :p:'('\ \ \ :n:
\ \ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'tools'\ \ \ :p:'['\ \ \ :n:'name

\ :k:'def'\ \ \ :nf:'__setitem__'\ \ \ :p:'('\ \ \ :bp:'self'\ \ \ :p:', '\ \ \
\ \ :n:'key'\ \ \ :o:'='\ \ \ :n:'key'\ \ \ :o:'. '\ \ \ :n:'lower'\ \ \ :p:'()
\ \ :n:'c'\ \ \ :o:'='\ \ \ :n:'traxcompile'\ \ \ :p:'('\ \ \ :s:'' '\ \ \ :si:
\ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'_cells'\ \ \ :p:'['\ \ \ :n:'key'\ \ \ :p:

\ :c:'# Dependency graph'\ \
\ :k:'if'\ \ \ :ow:'not'\ \ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'graph'\ \ \ :
\ \ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'graph'\ \ \ :o:'. '\ \ \ :n:'add_r
\ :k:'for'\ \ \ :n:'edge'\ \ \ :ow:'in'\ \ \ :bp:'self'\ \ \ :o:'. '\ \ \ :
\ \ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'graph'\ \ \ :o:'. '\ \ \ :n:'delet
\ :k:'for'\ \ \ :n:'cell'\ \ \ :ow:'in'\ \ \ :n:'c'\ \ \ :p:'['\ \ \ :n:'k
\ \ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'graph'\ \ \ :o:'. '\ \ \ :n:'add_e
\ :k:'try'\ \ \ :p:': '\ \
\ \ :k:'print'\ \ \ :s:'' GRAPH(TOPO): '\ \ \ :p:', '\ \ \ :bp:'sel
\ \ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'_cells'\ \ \ :p:'['\ \ \ :n:'key
\ \ :k:'print'\ \ \ :s:'' GRAPH(BFS) : '\ \ \ :p:', '\ \ \ :bp:'sel
\ \ :k:'for'\ \ \ :n:'cell'\ \ \ :ow:'in'\ \ \ :bp:'self'\ \ \ :o:
\ \ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'emit'\ \ \ :p:'('\ \ \ :n:
\ :k:'except'\ \ \ :n:'Graph_topological_error'\ \ \ :p:': '\ \
\ \ :c:'# We made the graph cyclic'\ \
\ \ :c:'# So, mark this cell as evil'\ \
\ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'_cells'\ \ \ :p:'['\ \ \ :n:'key
\ \ :c:'# And remove all incoming edges to go back to'\ \
\ \ :c:'# status quo'\ \
\ \ :k:'for'\ \ \ :n:'edge'\ \ \ :ow:'in'\ \ \ :bp:'self'\ \ \ :o:
\ \ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'graph'\ \ \ :o:'. '\ \ \

\ :k:'def'\ \ \ :nf:'getformula'\ \ \ :p:'('\ \ \ :bp:'self'\ \ \ :p:', '\ \ \ :
\ \ :n:'key'\ \ \ :o:'='\ \ \ :n:'key'\ \ \ :o:'. '\ \ \ :n:'lower'\ \ \ :p:'()
\ \ :k:'return'\ \ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'_cells'\ \ \ :p:'['\ \ \
\ :k:'def'\ \ \ :nf:'__getitem__'\ \ \ :p:'('\ \ \ :bp:'self'\ \ \ :p:', '\ \ \
\ \ :k:'print'\ \ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'_cells'\ \ \ :p:'['\ \ \ :
\ \ :k:'if'\ \ \ :bp:'self'\ \ \ :o:'. '\ \ \ :n:'_cells'\ \ \ :p:'['\ \ \ :n:
```

```

        \ :k:'return'\ \ :s:'"ERROR: cyclic dependency"'\
\ :k:'else'\ \ :p:':'\
    \ :k:'print'\ \ :s:''evaluating ['\ \ :si:'%s'\ \ :s:']:\
\ :k:'return'\ \ :nb:'eval'\ \ :p:'('\ \ :bp:'self'\ \ :o:'\

```

This engine supports:

- User defined functions (just add them in formula.py)
- Cell dependencies with detection of cycles, of any length
- Unlimited size spreadsheets
- Notification of cell changes to other modules
- Automatic recalculation
- A custom formula language (ok, that's not in this piece of code ;-)

The [whole project](#) is now 1167 LOC, of which 591 are generated or graph_lib, which means everything including the traxter “compiler” and the UI is less than 600 lines of code.

Not too shabby.

My goal is a functional spreadsheet with a working GUI and supporting the most common functions and pieces of the formula language in 2000 LOC, and I am willing to work on it for about two weeks.

Let's see how it turns out. Of course anyone with 5 free minutes can contribute his favourite spreadsheet function (I already [have sum](#) ;-)

It promises:

- Native look&feel for your apps on all major platforms
- Works for graphical and terminal apps
- Open Source
- Low overhead
- Universal binaries that work on Windows, Mac and Linux

Seems too good to be true, but it is supposed to be released next month.

- [The website](#)
- [The press release](#)
- [The videos](#)
- [The whitepaper](#) (which reads mostly like an ad)

It seems my [Hinges and Integration](#) article is the first result when you search for Hinges in google.

I should get a hingemaker to sponsor me.

For my spreadsheet project, I had to redo something I had forgotten about: cell displacement. I did that [once](#) when the formula language was python.

At the time, I parsed the python using the [tokenize module](#) and Ka-Ping Yee's [regurgitate](#).

Python->Tokens->Displace cells->regurgitate->Python

Since I have lots of other things to do, I decided to do it the same way, and wrote the equivalent of regurgitate for Traxter, my formula language.

It turns out it was not really hard, but I had to redo parts of the parser so it kept more information about the source.

After that it was simple, you see, Traxter compiles to Python. That means all I had to do was another (very similar) backend, and there it is, a Traxter-to-Traxter compiler. Or decompiler. Or something. And relative cell references are working again (and now with the right syntax).

I was able to hack a bit at StupidSheet this morning, and there are some real advances.

In fact, it's a barely functional spreadsheet already! Hell, it has at least one feature OOcalc lacks, and one both OOcalc and KSpread missed.

Check it out in this video (9.5MB):

Sorry, dead file

Sorry if the audio is out of sync and/or too low. And you probably can't stream it, you need to download it first. And maybe it will 404 on you. If that's the case, wait a few minutes and insist.

... asking the president of Kazakhstan how to patch it under FreeBSD (which I must confess I have no idea, unless you answer "using patch"?).

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

More info [here](#).

system-message

WARNING/2 in `everything.txt`, line 14524
Duplicate explicit target name: "here".

You may have noticed no posts about StupidSheet for about a week.

Well, I ran into the limitations of the formula parser I was doing using Aperiot. I just couldn't make it parse this:

```
A1=IF (A2=B2, 1, 0)
```

So, I spent the next week trying one parsing package for Python a day until I could find one I understood and could make it parse that example.

I must say it was educational.

So, now the parser is based on **PLY** which is pretty much Lex+YACC with a (slightly more) pythonic syntax, and it works.

Yes, it's a bit harder, but by trying to do things simply I was limiting myself too much, and, perhaps underestimating myself.

I am a pretty smart guy, there is no reason I can't understand these things.

Anyone who has no idea who the spurs are, ignore this, [from Cary Clack](#).

My favorite bits:

And what are some of the thuggish things the team does?

Well, you know those allegations about Michael Vick?

The Spurs are into pit bull fighting?

No, no. Panda bears.

What?

Panda bear fighting. Most of the guys own panda bears, so we get together, put soft and fluffy pillows on the paws of the bears and let them go at it.

How dirty are you going to be towards LeBron?

Real dirty. The fans are going to help us mess that kid up. You know how he spells his name with a capital 'B'?

Yes.

Well. We're going to have posters spelling it with a small 'b.' And when they call out his name they're going to scream 'Lebron!' instead of 'LeBron.' It's dirty, but that's how we roll in San Antonio.

On [April of 2006](#) I wrote this about CentOS/RHEL/Fedora installers:

I actually like anaconda a lot. But I would like it even better if I could just get some sort of tool that, given a list of packages, would create an anaconda-based CD with customizable install scripts.

And finally, here it is: [revisor](#). Now all I need is that someone makes it work with CentOS and I'm fulfilled :-)

Here are the two largest mail servers I manage. They handle over 20K users, with full mail service.

```
[root@nmail ~]# uptime
11:02:29 up 217 days,  3:09,  3 users,  load average: 1.54, 5.79, 7.40
```

```
[root@correol ~]# uptime
11:02:25 up 227 days, 21:58,  1 user,  load average: 6.78, 4.61, 3.90
```

Ok, the load is a little high, but I know the reason and it's not a big problem.

I have been exploring embeddable languages for the last month or so. I have learned forth and some of its many *many* variants^[1] and while exploring [one of the most obscure ones](#) called Atlas^[2], I found a very interesting [README](#) which I will quote liberally in this post.

Virtually every industry analyst agrees that open architecture is essential to the success of applications. And yet, even today, we write program after program that is closed--that its users cannot program--that admits of no extensions without our adding to its source code. If we believe intellectually, from a sound understanding of the economic incentives in the marketplace, that open systems are better, and have seen this belief confirmed repeatedly in the marketplace, then the only question that remains is why? Why not make every program an open program?

And this is indeed very important. Any app worth developing is probably worth developing in an extensible manner. After all, you are probably never going to figure out what the user really wants to do.

But why is not every program "open" in this manner?

Well, because it's HARD! Writing a closed program has traditionally been much less work at every stage of the development cycle: easier to design, less code to write, simpler documentation, and far fewer considerations in the test phase. In addition, closed products are believed to be less demanding of support, although I'll argue later that this assumption may be incorrect.

This is true, although it was much more true in 1991 when Atlast was released to the public domain. Nowadays the generalized adoption of more dynamic languages has made this much easier. After all, you can write a plugin system for your python app in perhaps ten lines of code.

However, there is a strong backlash from the pains of MSOffice scripting, which involved loveliness like function names that changed according to your locale, and basic as the language of choice.

On static languages this is harder, but there are technological solutions. For example, in KDE there is already a limited scriptability culture based on how easy it is to provide external DCOP interfaces (I suppose I should say DBUS nowadays). Also for KDE4 the Kross scripting framework promises language-neutral scripting.

Most programs start out as nonprogrammable, closed applications, then painfully claw their way to programmability through the introduction of a limited script or macro facility, succeeded by an increasingly comprehensive interpretive macro language which grows like topsy and without a coherent design as user demands upon it grow. Finally, perhaps, the program is outfitted with bindings to existing languages such as C.

Noone provides straight custom C bindings for applications now, right? But yes, I expect scripting is not the first thing that gets implemented.

An alternative to this is adopting a standard language as the macro language for a product. This approach has many attractions. First, choosing a standard language allows users to avail themselves of existing books and training resources to learn its basics. The developer of a dedicated macro language must create all this material from scratch. Second, an interpretive language, where all programs are represented in ASCII code, is inherently portable across computers and operating systems. Once the interpreter is gotten to work on a new system, all the programs it supports are pretty much guaranteed to work. Third, most existing languages have evolved to the point that most of the rough edges have been taken off their design. Extending an existing language along the lines laid down by its designers is much less likely to result in an incomprehensible disaster than growing an ad-hoc macro language feature by neat-o feature.

Unfortunately, interpreters are slow, slow, slow. A simple calculation of the number of instructions of overhead per instruction that furthers the execution of the program quickly demonstrates that no interpreter is suitable for serious computation. As long as the interpreter is deployed in the role of a macro language, this may not be a substantial consideration. However, as soon as applications try to do substantial computation, the overhead of an interpreter becomes a crushing burden, verging on intolerable. The obvious alternative is to provide a compiled language. But that, too, has its problems.

Interesting point here is that again, the speed of interpreters is not so big a deal nowadays. After all, [LuaJIT](#) is pretty fast for most uses.

Using a compiler as an extension language is not something I have run into but I could do some pretty things using, for example, [TCC](#), Python and [python-instant](#).

Now we get to ATLAST itself:

ATLAST is a toolkit that makes applications programmable. Deliberately designed to be easy to integrate both into existing programs and newly-developed ones, ATLAST provides any program that incorporates it most of the benefits of programmability with very little explicit effort on the part of the developer. Indeed, once you begin to “think ATLAST” as part of the design cycle, you’ll probably find that the way you design and build programs changes substantially. I’m coming to think of ATLAST as the “monster that feeds on programs,” because including it in a program tends to shrink the amount of special-purpose code that would otherwise have to be written while resulting in finished applications that are open, extensible, and more easily adapted to other operating environments such as the event driven paradigm.

The idea of a portable toolkit, integrated into a wide variety of products, all of which thereby share a common programming language seems obvious once you consider its advantages. It’s surprising that such packages aren’t commonplace in the industry. In fact, the only true antecedent to ATLAST I’ve encountered in my whole twisted path through this industry was the universal macro package developed in the mid 1970’s by Kern Sibbald and Ben Cranston at the University of Maryland. That package, implemented on Univac mainframes, provided a common macro language shared by a wide variety of University of Maryland utilities, including a text editor, debugger, file dumper, and typesetting language. While ATLAST is entirely different in structure and operation from the Maryland package, which was an interpretive string language, the concept of a cross-product macro language and appreciation of the benefits to be had from such a package are directly traceable to those roots.

This concept was later adopted by Lua, Tcl, and you **could** use Python in this manner, although usually it’s done the other way around. Which means that there is not so much sharing of extension languages as there could be.

And onto the conclusions:

Everything should be programmable. EVERYTHING! I have come to the conclusion that to write almost any program in a closed manner is a mistake that invites the expenditure of uncounted hours “enhancing” it over its life cycle. Further tweaks, “features,” and “fixes” often result in a product so massive and incomprehensible that it becomes unlearnable, unmaintainable, and eventually unusable.

Amen.

Far better to invest the effort up front to create a product flexible enough to be adapted at will, by its users, to their immediate needs. If the product is programmable in a portable, open form, user extensions can be exchanged, compared, reviewed by the product developer, and eventually incorporated into the mainstream of the product.

This prefigures the current FLOSS ecosystem, which is pretty impressive for a product that was **mature** in 1991, although of course there was already a community of EMACS hacking and similar niches.

It is far, far better to have thousands of creative users expanding the scope of one's product in ways the original developers didn't anticipate--in fact, working for the vendor without pay, than it is to have thousands of frustrated users writing up wish list requests that the vendor can comply with only by hiring people and paying them to try to accommodate the perceived needs of the users.

True if a little too cynical for my current taste ;-) Of course in FLOSS there is no such conflict between users and developers because the developers can usually just ..
raw:: html

~~whine~~

explain to

system-message

WARNING/2 in `everything.txt`, line 14762

Block quote ends without a blank line; unexpected unindent. backrefs:

the users the realities of free software development. But hey, a low barrier to entry is always a nice thing to have.

Open architecture and programmability not only benefits the user, not only makes a product better in the technical and marketing sense, but confers a direct economic advantage upon the vendor of such a product--one mirrored in a commensurate disadvantage to the vendor of a closed product.

The chief argument against programmability has been the extra investment needed to create open products. ATLAST provides a way of building open products in the same, or less, time than it takes to construct closed ones. Just as no C programmer in his right mind would sit down and write his own buffered file I/O package when a perfectly fine one was sitting in the library, why re-invent a macro language or other parameterisation and programming facility when there's one just sitting there that's as fast as native C code for all but the most absurd misapplications, takes less than 51K with every gew-gaw and optional feature at its command enabled all at once, is portable to any machine that supports C by simply recompiling a single file, and can be integrated into a typical application at a basic level in less than 15 minutes?

And then proceeds to throw a Forth variant at you ;-) Good concept, perhaps not the nicest language to use, although the choice is very understandable.

Am I proposing that every application suddenly look like FORTH? Of course not; no more than output from PostScript printers looks like PostScript, or applications that run on 80386 processors resemble 80386 assembly language. ATLAST is an intermediate language, seen only by those engaged in implementing and extending the product. Even then, ATLAST is a chameleon which, with properly defined words, can look like almost anything you like, even at the primitive level of the interpreter.

Again and again, I have been faced with design situations where I knew that I really needed programmability, but didn't have the time, the memory, or the fortitude to face the problem squarely and solve it the right way. Instead, I ended up creating a kludge that continued to burden me through time. This is just a higher level manifestation of the nightmares perpetrated by old-time programmers who didn't have access to a proper dynamic memory allocator or linked list package. Just because programmability is the magic smoke of computing doesn't mean we should be spooked by the ghost in the machine or hesitant to confer its power upon our customers.

Oh yes. Libraries rule, and when there is no such thing as the library you need, it's the worst position to be in.

Don't think of ATLAST as FORTH. Don't think of it as a language at all. The best way to think of ATLAST is as a library routine that gives you *programmability*, in the same sense other libraries provide file access, window management, or graphics facilities. The whole concept of "programmability in a can" is odd--it took me two years from the time I first thought about it until I really got my end effector around it and crushed it into submission.

I am probably going to use ATLAST or something similar in a program, but the user will not see a forth-like language at all, but a classical Excel-like formula language which would get compiled to the forthish language behind the scenes.

Open is better. ATLAST lets you build open programs in less time than you used to spend writing closed ones. Programs that inherit their open architecture from ATLAST will share, across the entire product line and among all hardware platforms that support it, a common, clean, and efficient means of user extensibility. The potential benefits of this are immense.

Indeed. Probably not using ATLAST, but Lua or Python or something else instead. And still, 16 years after ATLAST was released to the public domain, we are still walking down this road, and not even close to the goal.

Quick, answer this:

What was the last time a basic piece of the Linux system was redesigned and replaced by everyone?

1

```
system-message
WARNING/2 in everything.txt, line 14852
Duplicate explicit target name: "1".
```

As they say, "Once you see a forth, you have seen a forth"

2

```
system-message
WARNING/2 in everything.txt, line 14854
Duplicate explicit target name: "2".
```

The "Autodesk Threaded Language Application System Toolkit", by John Walker. Version 1.0 was released in August 1995, 1.1 in July 2002... I expect 1.2 around August 2009 :-)

And the new piece was not a drop-in replacement or evolutionary development for the old garbage?

Please post the answers on comments, because the best I can come up with is one of the following:

- Postfix replacing Sendmail
- Everything else replacing Wu-ftpd
- GRUB replacing LILO ? (not that GRUB is all that great, but at least you have a decent chance of fixing it when it breaks)
- OpenSSH replacing telnet and rlogin

There are still distros shipping Wu-imap and its offspring!

There are still distros shipping the old syslog!

Let's consider a basic, tty Linux first.

- 1) GRUB (ok)
- 2) Linux kernel (ok I guess)
- 3) Ancient SysV init (unless you use pardus/gobo/some other radical distro)
- 4) Services, which probably include
 - 1) Syslog-NG (which is marginally less broken than old syslog)
 - 2) Sendmail (even if only for loopback addresses, it's still lame)
 - 3) OpenSSH (ok, although I think the client sucks because I can't figure out how to store passwords and passphrases in KWallet)
- 5) A getty

At least here there is not much room for innovation because we are trying to start something that is a lot like a 30-year-old Unix box.

So, let's go server-ish. What would you normally use?

- BIND

Ancient software with a terrible security history. Yes I know it's rewritten lately. They did that before, too, you know.

- Apache

For all the good things Apache has, it has some bad ones, too.

- It's overkill for most servers.
- As the A in LAMP it has lead people to believe PHP4 is the right language to develop applications in, and MySQL a good place to store their data.
- If it fails to do what you want, you **may** get an error. Or not.
- The configuration is in some sort of pseudo-XML

Let's get real. For most modern web apps what you want is a decent, high performance WSGI thingie for python apps, and whatever you use for Rails. Apache may or may not be that or have that inside, but who needs the rest of it? What for? PHP pages? mod_perl web apps?

No, really, I'm asking a question here. What pieces of Apache do you use nowadays?

- Samba
 - It does what it does.
 - Noone else does it.
 - Ergo, it's the best at what it does.
 - That doesn't mean that losing its TDB every once in a while while doing a "RPC vampire" is not annoying.

But actually, I am pretty happy about Samba. I mean, what's the alternative, here? NFreakingS?

- CUPS

Ok, not too many new print servers out there, but hey, it's better than LPRng!

And if I had written this rant three years ago, I would have used the exact same examples.

Where's the vibrant new server app?

Who's going to write a cool, performing, easy to configure HTTP+WSGI server in D?

Who's going to implement a fast, secure, simple, zeroconf-enabled, file serving daemon?

Who's going to replace BIND?

Who's going to create a Linux server distro with **only** decent software in it?

Me? No way, I have diapers to change. And there used to be smarter and more driven people around to do this stuff. Are they **all** changing diapers now?

Come on, stop rehearsing with your band that plays "metal with medieval influences"! Stop growing your stamp collection! Stop

Come on, it's only going to consume **at most** a year or two of your life. It's not going to harm you more than a budding alcoholism, or a poetry hobby, or attending furry conventions, young man (or woman)!

You don't need to be all that knowledgeable (look at the BIND4 sources) or brilliant, all you need is to be **industrious**.

Grow a spine and get cranking! Show us old hacks what you've got!

I stumbled onto a site where you write and they are supposed to share the advertising revenue with you.

So, I decided to do an experiment. I created an account and am posting there using a character.

This character is the most annoying, cranky, acid guy I can be. It's my evil side. For example, my first post was about how everything on the site is garbage. My second one is a specious argument about how noone is special.

Let's see [where it leads...](#)

Warning: flammable.

It seems Ilan has found a feature some developer wrote, called referers log, and thus this blog.

Well, Ilan, I will make sure to follow this link so you notice this reply :-)

Since you intend to reinforce your dermal structure through exposition to high temperatures, let me be the first to assist you.

I am fully aware of everyone's right to criticism. In fact, I fully endorse it.

I also endorse my right to tell any critic that his criticism has the value of manure. Call it metacriticism.

And yes, a 10000 LOC piece of code is worth more than your lame patch. Want to verify it? Check market value of both things. You'll be amazed.

In other issues,

- It's really a bad thing to use quotes on things that are not quoted, as you did. If you want to show emphasis, use cursive or bold, not quotes.
- I am happy to challenge **your** notions, instead of challenging mine, because I think your notions are stupid. want details?

Your notions are stupid because:

- You make the point in a condescending manner. What is that garbage about "better quality users"? if not condescension you would hate on a developer? At least the developers are providing value. You aren't.
- Your project is tainted by paranoia. Noone is out to get you. Not even I am. I just don't suffer fools gladly.
- TANSTAAFL. You get so called free software, but you still pay for it. You don't want to pay the price, don't use it. The price, Ilan, is that you have no control over it, **because** you didn't pay for its development.

Oh, you may say, I am getting involved. I released Directory Free KDE! No, if you had gotten involved, KDE would have those directory references removed, not your patch that noone will ever use.

- Your writing style is stupid, and an idea wrapped in stupidity is a stupid idea. I almost fainted with the reference to the "wonderfully subversive path". Stupidity.
- You obviously have no idea of the effort involved in actual software development. I do. I know it so well, I stopped. You just have a chip in your shoulder and an attitude. Your ignorance drives your paranoia. Your paranoia drives your work. Your work drives nothing, because it's lame.

In your site, you say "thinking differently, thinking dangerously".

Amazing as it is that you ape a multi-billion dollar corporation's slogan to pretend to show your disconformism, I agree.

You think differently. You think wrongly, too, but at least yes, you think differently. As for the dangerously part... well, I suppose that if you are too busy thinking differently you may fall into a drain or something.

In short:

- Your so-called software project isn't.
- It's a patch, and it's a lame one.

- Your attitude, if you saw it in a developer (which you aren't), you would denounce. So you are also a hypocrite. (note: I am not. I fully enjoy flaming people sometimes, and support everyone's right to flame and be ornery at will).
- Your impulse comes from sheer ignorance, paranoia and conceit.

I am happy to encourage you, Ilan, in your path. Please go and prove me wrong, by developing a much nicer KDE, and then making it the default in many, many distros.

On the other hand, I am willing to bet USD 20 that in six months you will have nothing to show.

Go ahead, make my day. Or at least earn me a dinner.

A modest one, though. I got a 802.11b card, and a nice USB storage thingie. Nice toys, should be useful, too.

Coding in KRsN is quite nicely moving forward. The design of the app is almost competent now, and it works well (a couple silly mistakes I just fixed had broken feed addition, though)

Said [Giles Bowkett](#)

The Perl community's starting to look more and more like the Lisp community every day. The combination of incredible power, reclusive wizards, and antisocial Slashdotters gives it the vibe of a lava-filled wasteland dotted with towers where strange men with white beards obsess over unspeakable knowledge. I spoke to someone once who compared programming in Lisp to studying Kabbalah, in that it does strange things to your head. Parts of Perl are like that. Still, source filtering's kind of cool. Unnecessary, but cool.

So, now we know. Saruman used too much Perl.

Monday was a very special day:

- Holiday (Independence day)
- Anniversary (3 years as Rosario's boyfriend)
- The first snowfall in Buenos Aires in 89 years.

Besides that, this week my brother is getting married so the whole family (including 2.5 month-old JF) is leaving for my ancestral lands tomorrow.

And I started a new small project, which should be finished soon.

This is something that seems useful to me in the context of mail servers, but maybe it will also find its uses elsewhere.

I call it rater, and it tells you if things are happening faster than a specific rate.

For example, I intend to use it to figure out if a specific IP is connecting to a server more than X times every Y seconds, or if a user is sending more than Z emails every T minutes.

The only thing I found for this is relayd, which is old, unmaintained and whose site has vanished.

The config file is something like this (thanks to libconfig):

```
limits : {
    user: (
        ("rosario", 90, 20),
        ("ralsina", 90, 10),
```

```

        ("*",2,10)
    );
    ip: (
        ("10.0.0.*",90 , 20),
        ("10.0.1.*",90 , 20),
        ("*",2 , 10)
    );
};

```

You can define as many classes of limits as you want (that would be ip and user in this example) and as many limit keys as you want, that will be matched using something like fnmatch.

I am using an in-memory SQLite DB for the accounting, and an interesting library called [libut](#) for the sockets, logging, and event loop.

This library has a very interesting feature: your app gets an administrative interface for free!

```

[ralsina@monty rater]$ telnet localhost 4445
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
libut control port interpreter
Type 'help' for command list.

```

```

help
command          description
-----
mem              - memory pool usage summary
var              - Display or set config variables
log              - Change log file or verbosity
  fds           - list selected file descriptors
  tmr           - show pending timers
  uptime       - show uptime
prf              - Performance/profiling stats
cops             - List coprocesses
help            - view command help
exit           - close connection

```

Commands preceded by * have detailed help. Use help <command>.

```

Ok
var
  name          description          value
-----
ut_log_level    log level                          Debugk
ut_log_file     log file                            /dev/stdout
ut_jobname      job name                             job1
ut_control_port control port IP/port                 127.0.0.1:4445
ut_basedir     shl base directory                  /mnt/centos/home/ralsin

```

Variables prefixed with '*' can be changed.

```
Ok
var ut_log_level Debug
```

```
Ok
var
  name                description                value
-----
ut_log_level          log level                Debug
ut_log_file           log file                 /dev/stdout
ut_jobname            job name                 job1
ut_control_port      control port IP/port     127.0.0.1:4445
ut_basedir            sh1 base directory      /mnt/centos/home/ralsin
```

Variables prefixed with '*' can be changed.

Ok

Pretty neat.

Beyond this, there will be a small client-side library that hides all the network stuff behind a couple of blocking calls (or you can do your own because the protocol is silly simple).

Really, I can imagine having one of these as my main computer (with external HD and monitor).

I probably won't and use my full-size notebook or a desktop when at home/office, but it should take me back to the days when I simply carried my Libretto everywhere because it was light enough.

Here's the best [review](#) I found so far.

People worry about the 4GB or 8GB "disk". If you keep your media (ISOs, movies, music) on an external HD, you probably can have everything else there.

Using rsync to keep two boxes synced... lots of potential.

I am probably buying two of the cheap ones.

I am no longer on planetKDE so this will probably not be read by many in the KDE community but...

Please, please, please start making big noise about the Asus eee. It ships with KDE and is expected to sell "up to 500,000 by middle of next year and 5,000,000 by 2009"^[1].

Is that number not big enough? It's probably more than our current installed base.

One of the ways your linux system sucks, and what you can do about it: system notifications.

I am hacking a bit on [rater](#) my daemon/client to see if things are happening more often than they should (in other words, generic rate limiting).

1

system-message

WARNING/2 in `everything.txt`, line 15184
Duplicate explicit target name: "1".

<http://asuseehacks.blogspot.com/2007/07/asus-eee-pc-pricing-availability-updates.html>

I had to take a few days off, since my brother got married and we all went back to Santa Fe for that and a weekend, and then everyone else has sore throats and I am the only one healthy.

But hey, it works well enough already:

- The simplistic protocol is done
- The server works
 - It can take hours of [gibberish](#) without problems.
 - It can take hours of valid input without problems.
 - It does what it's supposed to do.
- It's staying below 300SLOC, which was my goal.

Missing stuff:

- Valgrind it.
- Client library.
- Generic CLI client.
- A [qmail-spp](#) plugin that uses it.

And then, I can forget all about it.

Second (or third) article on this series. Now we tackle why and how cron/at suck and what we can do about it.

In [2005-04-04](#) I whined about a site called Section 6 plagiarizing one of my articles (the always popular [Linux Booting Process Unveiled](#)).

Well, yesterday I found in my inbox an apology from Kristopher Zentner, owner of the site!

So, I can say Section6 1z n0t 7h3 5ux0rz n0vv!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "before".

As I posted [before](#) I am pretty hyped about the Asus eee, the \$199 laptop.

Ok, maybe the \$250 laptop, according to latest rumours.

Then again, I am not the only one. Here's an [interesting thread](#) about what you could do with one of them.

Now, the random venting:

A fairly large portion of Linux users are very annoying. It's normal, since a very large portion of humanity is annoying too.

However, consider this post, which I will quote because I refuse to link to it, about [my cron/at article](#):

GreyGeek - Subject: Another "I hate Linux" article.... (Jul 31, 2007, 17:18:34)

Oh, yeah. That's what the article looks like. A linux hater's rant. Sure, I spent three hours doing research and writing telling you half a dozen ways to make Linux better (IMNVHO) because I *hate* Linux.

as if we aren't getting a FLOOD of them these days. I often wonder if Microsoft has began "donating" Ferrari laptop to bloggers who rant on Linux.

At least the ferrari laptop matches my car! Accesoring is important!

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "here".
```

I wonder if it's because this guy didn't understand a word I wrote? Or is he just paranoid? I know he's not [the author of cron](#) (he once posted a comment [here](#)[1]_) so it's not a wounded ego or something like that[2]_.

```
system-message
WARNING/2 in everything.txt, line 15265
Duplicate explicit target name: "here".
```

```
system-message
WARNING/2 in everything.txt, line 15271
Duplicate explicit target name: "before".
```

Which is usually the point where I switch to new versions...

BTW: Cnuth? Best release name for KDE ... ever!

And now for something at least somewhat constructive: A modest, and hopefully realistic proposal on how UI designers can coexist with free software coders.

Nowadays, the very little time I have for personal projects is spent doing things like fixing little things and adding little features to BartleBlog[1]_ and thinking how I could use [GLE](#) and [mako templates](#) to create a cool nerdy tool to create charts.

However, I will be having a little time for personal projects in a couple of weeks, and having stumbled onto [chipmunk](#) today really got me thinking.

It's a seriously nifty 2D physics library. Consider this demo video:

Now, what could possibly be done with it... I need to really think.

1

```
system-message
WARNING/2 in everything.txt, line 15275
Duplicate explicit target name: "1".
```

At least I think that was him. Why would anyone *pretend* to be Paul Vixie? Now how's that for name dropping. Has anyone else with a Wikipedia entry left a comment in this blog?

2

```
system-message
WARNING/2 in everything.txt, line 15277
Duplicate explicit target name: "2".
```

And I don't expect that article to be hurtful, really, or I wouldn't have written it.

1

I joined Orkut many moons ago. I hated it. It was boring, vapid, and useless. That was my whole experience in social networking sites[1].

A few days ago I was invited to LinkedIn. Since I had heard some good things about how it was useful and let you network your contacts and leads to business opportunities, I took care to fill in some data, start some links.

So far... if the Questions page (helpfully labelled Answers, of course) is a representative sample of the average LinkedIn fellow...

Here are a few examples (In the technology area):

Does anyone has a doc named stauto.pdf (SecurID API (SOFTID 2.0)) ?

I can make you one if you want it!

Has anyone else recently lost Sprint wireless coverage? Suddenly, our Sprint Moto-Q doesn't get coverage at home.

MMMMMkey...

Sure, there are some that are interesting, even if most of those are just too vague or locale-specific or both (What's the future of digital media in India?)

I did find a few people I had lost track of, and am surprised it keeps on suggesting I know two people I actually **do** know but am not interested in re-finding.

But, all things considered: it is looking like a very slightly less useless Orkut.

Is it a problem with me, or would I find something like **Ohloh** more interesting?

It seems that during the big SVN conversion some data were lost in commits.

So, if you check KDE's svn for the really old stuff, it has no author information.

For example, check this out:

<http://websvn.kde.org/branches/KDE/1.1/kdenetwork/krn/>

I know I wrote pretty much everything there, but you are probably not going to notice it, and I spent a good couple of years working really hard on that thing.

Sure, it was crap, but it was hard-worked crap, and there is at least one thing I am slightly not ashamed of: TypeLayout, which is still nicer than most toolkits have, even if it was shamelessly copied from LinuxConf.

But what the heck, it's old stuff.

Always remember to do this:

Or else, your code will break in misterious ways.

That's because sys/quota.h has this:

Why? I have no idea. But this is true at least on Centos4, I have no clue of this is also true for your distro, but it is sooooo wrong :-)

```
system-message
```

```
WARNING/2 in everything.txt, line 15307
Duplicate explicit target name: "1".
```

What I use to post this.

1

```
system-message
```

```
WARNING/2 in everything.txt, line 15336
Duplicate explicit target name: "1".
```

Unless you count how I met my wife.

I got a mail from a reader telling me that he couldn't download the sources for Notty, the toy app I developed in my [Rapid Application development using PyQt and Eric3 ... in realtime!](#) tutorial.

So, I checked and was shocked at just how much the server moves had wrecked that article, which is one of my favourites!

No images, broken link to the sources, no syntax highlights!

So, I rejiggered the thing quickly with some search and replace (thanks restructured text!) and now it should be up to standards, except that... it's still about Qt3 and I am not even sure it works nowadays.

Normally that would be simple to fix: change the code as needed, make it work, and be happy.

But the fun thing about that article was that it was written in 3 hours, and it talks about how it was written in 3 hours. So, I think I may have to do keep that and add a note with a link to a corrected/updated one, someday.

I have no idea how, but I ran into this in [Linuxtoday](#):

KDE Programming Tutorial 0.2 (Dec 22nd 1998, 00:09:36)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

Roberto Alsina announced that he uploaded to ftp.kde.org the version 0.2 of his excellent programming tutorial. It is also available [here](#). Hopefully, this must-have material will be soon included in the kdesdk package and in CVS.

system-message

WARNING/2 in `everything.txt`, line 15392
Duplicate explicit target name: "here".

Now, this was pretty shocking because I don't remember writing it (anyone knows where a copy may be?). I have no idea what was in it, and in fact, the idea of people learning C++ from me in 1998 is so stupid it's shocking, since I knew very little.

Then it hit me, the URL! Ultra7? That was my 486 at college[1].

It had a webserver? What on earth had I been drinking back then? Why can't I remember this???

Wayback machine, come here! Fetch!

Here's [my first home page](#). I must say it's pretty good. It's not garish. Very Web2.0 in the abundant white space.

First interesting thing: there is at least one extant copy of [PyXForms](#), my first free software project, used by absolutely noone.

And there are even two cool screenshot of a functional program noone ever used[2].

In fact, I have no idea how I did that highlighted message display.

Then all that proto-good taste goes away when you reach the page for my most "popular" project ever... [Krn](#) in my [1999 home page](#)[4]. It' so 1999 free software style. All that's missing is an electric blue background.

It seems I had discovered [The Gimp](#) and it's banner scripts!

BTW: I wonder what's Magnus Reftel doing nowadays? We used to exchange dozens of mails a day and for some reason we lost contact.

There was also a mailing list archive for the very very very first posts of our proto LUG (which would later become [LUGLI](#))

And here's the best part, and the one that makes me kinda sad... [Themes for Qt1](#).

You can read a little about them in [this thread](#).

I have written about them once or twice before but here's the short version:

Rasterman was starting to hack themes for Gtk. I decided they couldn't be the only ones with this *cool new technology* but I had the big disadvantage of not being able to touch Qt's code because of licensing issues. So I intercepted the drawing events using a LD_PRELOAD hack and implemented a theme full of gradients *without modifying or recompiling Qt or KDE*.

Now that's one of the most impressive hacks I have ever performed, even if it was a hack done with the most evil code ever, and... **there seems to be no extant copy of the code or the screenshots.**

Which is kinda sad, indeed.

Allow me to enter trash-talking mode...

Be *very* afraid because this year, my mighty skills are entering [PyWeek](#) for the first time.

Just to make it interesting and somewhat challenging:

- I will be entering as an individual, not in a team.
- I have not written a line of PyGame code in my life.

1

```
system-message
WARNING/2 in everything.txt, line 15462
Duplicate explicit target name: "1".
```

UNL, where every computer had a public IP address!

2

```
system-message
WARNING/2 in everything.txt, line 15464
Duplicate explicit target name: "2".
```

The idea was a sort of GUI Pine, all handled with single key strokes. I still pine for that program![\[3\]](#)

3

```
system-message
WARNING/2 in everything.txt, line 15466
Duplicate explicit target name: "3".
```

And for the fjords.

4

```
system-message
WARNING/2 in everything.txt, line 15468
Duplicate explicit target name: "4".
```

AKA the buggiest program ever to be part of KDE proper.

- I have written barely any game code at all.
- I will submit my final code one day earlier, just because **I can**.

Nonetheless, I intend to kick asses and ask no questions. I intend to overwhelm you all with a game of such awesomeness you will all wither like lettuce in Lapland. I will write code of such ingenuity you will try to steal it and fail.

...exiting mode.

Should be fun and my entry will probably suck. But we'll see...

Trying to create a more serious looking infrastructure for my consulting business, I decided I needed a non-mobile phone number, and a way to receive Faxes[1]. While I would use this as a secondary business number, it's just my home phone.

Since the costs for old-fashioned phone service in Argentina are stupid[2] I started investigating VoIP alternatives.

On a subway ticket, I found an ad for [Metrotel](#) which is quite interesting.

First, there is the numbers:

- No installation fee
- \$35 for 1000 minutes
- Cheaper long distance service

What they do is provide you a simple [Handytone 486](#) box, and you plug it to your Internet-connected LAN on one side, and a phone on the other.

You can also plug in your land line and use the phone to call on either one, and another ethernet device (example, a computer or switch). If you have only one computer, you just stack it onto your cablemodem or whatever.

Absolutely trivial installation... unless they ship you the wrong gadget, as they did to me, sending me one that was disabled. But that's ok, they fixed it in a couple of days.

I have three small issues:

You have to sign a 12-month contract. However, if you take the smallest plan, the 12-month contract is cheaper than the connection fee for Telefónica de Argentina! And you still get 1200 free minutes in that year!

You must use the gadget. While it's a simple SIP device, they won't give you the user and password, so you can't use a softphone or a real VoIP phone with this service (unless you hack it, but why bother).

Because of that, it's kinda tricky that the ads say you can use this phone line "everywhere". I mean, it's true... if you have broadband internet connections with ethernet connections "everywhere".

However, if you do your due diligence and figure out what you are getting into, it's a very nice service, which works well, and has great cost advantages over the traditional alternative.

It's specially great if you want to have a Buenos Aires phone number and work from somewhere else (for example, labour costs in other provinces are usually below

75% of those in Buenos Aires).

Found on reddit, from [Lens Culture](#)

A mindblowing presentation of image-browsing software by Microsoft Research at TED. Please watch it.

And then there is [this research from CMU](#).

Really awesome stuff.

Now, this is really scifi stuff. Or should have been. While science fiction was promising jetpacks and virtual worlds, computer science is providing new and incredible ways to manipulate information.

I suppose it happens to everyone once in a while, and it has happened to me often in the past, but I am thinking if I should keep on writing this blog, or if some large change is needed.

Here are some random things from my head:

1. Almost noone reads it. Really. It has less than 40 subscribers. That's pathetic for a blog that has content for over 7 years :-)
2. Maybe I should post in spanish, or at least bilingually.
3. Maybe I should write more features. When I write a longish piece and announce it, there is a respectable traffic surge.
4. On the other hand, I enjoy writing it. And it's really very little effort (specially now, with BartleBlog ;-)
5. Maybe it should be more focused in one area, make it a python programming blog, or a tutorials blog, or something like that.
6. But I am not a focused person. I am a generalist. This week I have worked in the following things:
 - VoIP
 - Django
 - PyQt
 - Linux sysadmining
 - Firewall/Proxy integration with windows clients

1

```
system-message
WARNING/2 in everything.txt, line 15530
Duplicate explicit target name: "1".
```

Although Faxes make no sense whatsoever in 2007, some companies still ask me for a fax number.

2

```
system-message
WARNING/2 in everything.txt, line 15532
Duplicate explicit target name: "2".
```

How about \$181.50 to install and \$36.50 monthly **with no minutes included**? And that's for home service. Commercial service is more expensive still. Those are pesos, but trust me, they look just as high to me as dollars would for you. The beauties of local monopoly!

- Consulting in the most generic sense, sitting with a company's IT staff and thinking about their situation.
- Learning PyGame

And this was in 5 days of work. If I listed what I have done this year, it would take me 500 items. I am *broad*, how could my blog be *narrow*?

7. Maybe it's just not interesting? Or badly done?
8. Is it too nerdy? Is it not nerdy enough?
9. I have had a blog with a small readership for 7 years, why is it bothering me now?
10. If I stop, it doesn't matter, I can always pick it up again later when I feel like writing.

So, there. You, the 40 guys, comment on it if you want :-)

Been busy. Work is back in full throttle, so time for coding and blogging is low.

Hopefully, I will earn enough in the coming month to take it easy in march, then go crazy in april and so on.

This blog has been much more fun than I expected and I want to thank the guys at pyds for hosting it for free, they provide excelent service :-)

Life is good. I have been playing with a flyvideo2000 I will install for a client: sweet TV board. I think I will buy myself one.

- It works with linux (after some learning, which is good, too)
- MythTV turns it into a lovely recorder, TV, etc, except you need internet for it to work at all (or else it doesn't get a channel list!).
- For a few dollars I get another TV and VCR, which I don't need but is cool anyway
- With a little effort and some cool free code, it can monitor surveillance cameras, saving a client of mine from buying a digital recorder... and giving me some more money :-)
- It has a remote that is supported by lirc (haven't made it work yet) so I am supposed to be able to remotely control mplayer/kaffeine/whatever (even when not playing TV :-)

So, all in all, fun stuff.

The only bad piece is, I haven't coded much (or at all), but I will get to it on due time.

In the last few days I have been learning Django in perhaps the hardest way possible: by being hired to work on a site someone else wrote.

I already had the view from 10000 feet. And since I had to get to this thing rather quickly, I jumped on my parachute from those 10000 feet, and learned it on the way down.

Here's what I knew:

- Python Web framework
- Regexp-based URL dispatching

- Its own template language
- Its own ORM and form stuff

I have hacked stuff based on TurboGears, Colubrid, pure CherryPy, Mako/Kid/Cheetah/CherryTemplate templates, Routes, Paste and about half a dozen other frameworks or pieces that are used for frameworks, so how new could it be? Well, not very new. I am starting to notice a sort of sameness in these things. They are all alike.

First, the conclusion: I liked it, I could work with it.

Now for some little detail:

- The URL dispatching is nice ,if not really interesting. there seem to be two ways to do this, all frameworks use one or the other, and almost everyone likes regexps better.
- The ORM+newforms is quite nice! Of course everything was done with old-forms, which is... not quite so nice. But you can switch pieces as you go, and the code actually simplifies as you hack, so it's good.
- The template language I could live without. It doesn't seem to be specially featureful, and it didn't seem as expressive to me as my current favourite, Mako. Luckily you can replace it easily. It's not that it's **bad**, it's just average.

So, I see no reason to learn it instead of Turbogears, or viceversa. On the other hand, if you know one, you can learn the other in perhaps a weekend, so there's no point on not having at least a basic knowledge of both.

Well, it seems I am in trouble for PyWeek.

Why? Because it's wednesday and I have done nothing. Nothing! It's because I have been working a lot, really, and I have a 4 month baby, too.

So, I am upping the ante.

I will do a PyDay.

I am taking tomorrow off (yeah, right!) and I'm doing the game in one day. Maybe I will scrounge a few hours on sunday, too.

It will probably not be fit for the contest because:

- I will use PyQt
- I won't test it in any platform other than my Linux box

But here's the game concept (BTW: Twisted **sucks** as a theme. It sucks really, really, really hard!):

According to the dictionary, Twisted also means perverted. So, this game, *Twisted Little Boy* is about a bad boy. A really bad boy. But a clever one. He creates machines using random equipment he finds to do evil, really mean things.

I will probably do a live-blog thing like those tutorials I wrote years ago about PyQt.

There's a Google code project (obviously empty): <http://code.google.com/p/twistedlittleboy/>
See you all tomorrow.

Suddenly I was having a calm day at work, and Rosario is taking care of the baby, so I spent a few hours on the PyWeek project.

I have integrated Chipmunk with QGraphicsScene.

What does it mean? That I can now...

- Create a scene
- Create a view onto that scene
- Create balls, walls and polygons as scene items
- Watch said balls/walls/polygons bounce around happily under Chipmunk direction.

For example, here's enough code to create a few balls a box and a staircase:

```

\ :bp: 'self' \ \ :o: '.' \ \ :n: 'scene' \ \ :o: '=' \ \ :n: 'ChipScene' \ \ :p: '('
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'ui' \ \ :o: '.' \ \ :n: 'view' \ \ :o: '.' \ \ :n: 's

\ :k: 'for' \ \ :n: 'x' \ \ :ow: 'in' \ \ :nb: 'range' \ \ :p: '(' \ \ :mi: '0' \ \
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'scene' \ \ :o: '.' \ \ :n: 'addBall' \ \ :p: '('
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'scene' \ \ :o: '.' \ \ :n: 'addBall' \ \ :p: '('
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'scene' \ \ :o: '.' \ \ :n: 'addWall' \ \ :p: '('
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'scene' \ \ :o: '.' \ \ :n: 'addWall' \ \ :p: '('
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'scene' \ \ :o: '.' \ \ :n: 'addWall' \ \ :p: '('
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'scene' \ \ :o: '.' \ \ :n: 'addWall' \ \ :p: '('
\ :k: 'for' \ \ :n: 'i' \ \ :ow: 'in' \ \ :nb: 'range' \ \ :p: '(' \ \ :mi: '0' \ \
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'scene' \ \ :o: '.' \ \ :n: 'addWall' \ \ :p: '('
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'scene' \ \ :o: '.' \ \ :n: 'addWall' \ \ :p: '('

\ :bp: 'self' \ \ :o: '.' \ \ :n: 'scene' \ \ :o: '.' \ \ :n: 'addPoly' \ \ :p: '('

```

I declare that *nifty*.

Not a game yet, but the engine is starting to look good.

The ball drops, follows both ramps, bounces down the stairs, hits the dominoes, the last one falls on the pivoting ramp, slides down and to the left, falls standing and leans left, then hits the rest of the dominoes.

A bit of progress, although not much time to work on it anymore so I will probably not make it.

Objects now can do things when they are hit by other kinds of things.

Example: If a ball hits the bottom of the catapult cup, the catapult shoots. If something hits the target object, you win the level.

- 1) I don't have the time for this kind of deadlines anymore. Not even one all-nighter? I did nothing on Saturday except real work and family reunion?
- 2) It's really easy to write games with Python. It's mostly just a matter of having a good game concept. The coding is the easy part.
- 3) Chipmunk is cool. Qt's graphic scene stuff is cool. My ChipmunkScene is coolest ;-) I should rethink the API but the concept is killer stuff. With a little work this thing is like LEGO!
- 4) I will try again in PyWeek6.

Here's the scenario:

- A Linux+Samba server with 40GB of data.

- A SMB-only small storage server.

Your mission? Backup the thing. You should do full backups, and keep the last three.

In another age, I would have cobbled a 10-line script using tar (and split, see below) and be done with it. But now I want to use backup software.

So, I tried, and I ran into the following limitations:

- 1) 2GB filesize limit on the storage server. Have no idea why, assume can't fix it.
- 2) Weird unicode characters in filenames. There must be some encoding issue, but when a Windows client saves a file with accented characters, the clients see it all right. On the server, though, they are weird-looking. This is enough to make mc unable to delete some folders, for example.

So far I have tried:

- 1) rdiff-backup: breaks with the unicode chars.
- 2) flexbackup: breaks with filesize limit
- 3) rsync: breaks with the unicode chars
- 4) synbox: using rsync backend, see above. Using tar backend, breaks with filesize limit.

Here's what I want:

A simple backup software, where I can tell it "take this, back it there, keep last three backups, do it in files smaller than 2GB, give me a report".

Bonus points if restoring it is doable from windows.

Any suggestions?

I am doing this as a public service for USA readers, because you are likely to be in contact with someone who believes evolution of life is incredibly unlikely. After all, many of your presidential candidates [don't believe in evolution](#). Which would be sad if it wasn't because I expect some of them do but are afraid of losing the ignorant religious fanatic vote (which makes it just lame).

You will sometimes see someone say something like "that's unlikely!. The odds of A are 1 in 1000 and the odds of B are 1 in 500, so the odds of A **and** B are 1 in 500000!".

This argument is very, very, very wrong. But for those with an intuitive knowledge of probability it looks good, because they know that the odds of a coin flip coming heads is 1/2 and the odds of 2 consecutive heads is $1/2 * 1/2 = 1/4$ and so on.

Now, let's construct me this way (warning: made up numbers ahead).

I am argentinian. There's roughly 35M argentinians out of 6000M humans. That's about 1 in 171.

I am male. That's about 1 in 2.

I am a bit over median height. That's also 1 in 2.

I have a beard. I am guessing 1 in 10.

I wear size 41 shoes. That's about 1 in 4.

I use glasses: I'd say 1 in 5.

I get dizzy riding buses: About 1 in 100.

I am married: 1 in 4.

I have a child: 1 in 2.
I have a son: 1 in 4.
He is 4 months old: 1 in 200

So, I have a 1 in 87 552 000 000 chance of existing. And I could make those odds much lower. After all, I was born in 1971, mi favourite colour is green and my name is Roberto.

There are two reasons why you should never trust this kind of number manipulation unless you really, *really* know what you are doing.

- 1) You can't just multiply anything. These odds have to be about *statistically independent variables*.

For example, I get 1 in 8 from having a child, and having a son.

Yes, maybe the odds of someone having a child are 1 in 2, and the odds of someone having a son is 1 in 4, but those are **correlated**. The odds of having a son *if you have a child* are about 1 in 2. The odds of having a child if you have a son are 100% :-)

If two variables are correlated you can **not** multiply their probabilities.

A much sillier example. Imagine there are ten countries, each with 1 tenth of the population.

The odds of being from A is 1/10. But he is also **not** from B,C,D,E,F,G,H,I, **or** J! There's a 9/10 chance for each one of those facts!

So, the odds of being from A and not from anywhere else is 1/10 times 9/10 ... that's almost 1 in 26!

And since the same reasoning applies to each country, it turns out there is almost a 60% chance the next baby will not be born in any one of the ten countries.

See how stupid that sounds? Of course when correlation is more subtle, it's harder to figure out, and you will not be able to do this while arguing.

- 1) Add random unlikely data.

This is slightly trickier, because it may be what you really wanted to do. Yes, me being **exactly** the way I am is extremely unlikely. However, something more or less like me is not.

Yes, I am unlikely but that's trivial. However, if you are going to apply this kind of reasoning in other cases it gets silly quick. Here's an example:

Imagine a lottery with 6-digit numbers. Today, the winner is 123456. Yesterday it was 654321.

The odds of those numbers being the winners is 1 in 1 000 000 000 000. But it's obvious that if you make two draws, **some** two numbers will come up! And whatever they are, they will be just as unlikely!

That something **specific** is unlikely doesn't always matter, because the important thing is the chance of some **kind** of thing happening, not of **one specific** thing.

And when/if you have a kid, and he/she/it asks you why he should study math, show him his kind of thing and tell him why:

Math makes it harder for people to lie to you.

This is the real outcome of my PyWeek failure: a neat library.

Take [Chipmunk](#) and [Qt](#) mash them up, and what do you get?

A OpenGL-accelerated, multiplatform, easy-to-use playground!

But watch the silly video that shows no interesting features instead:

You can't see it but there are 29 balloons, numbered, that bounce around happily.

Performance in the video sucks because I had to disable OpenGL in order to capture it correctly (and video recording kills my notebook, anyway). That demo normally runs in 3 seconds, not 57.

Here's the non-boilerplate code for that demo:

```

\ :k:'def' \ :nf:'fillWorld' \ :p:'(' \ :n:'scene' \ :p:'):' \
\ :n:'items' \ :o:'=' \ :p:'[]' \
\ :k:'for' \ :n:'x' \ :ow:'in' \ :nb:'range' \ :p:'(' \ :mi:'1' \
\ :n:'b' \ :o:'=' \ :n:'cs' \ :o:'. ' \ :n:'CPBodyItem' \ :p:'(' \
\ :n:'s' \ :o:'=' \ :n:'cs' \ :o:'. ' \ :n:'CPCircleShapeItem' \
\ :n:'t' \ :o:'=' \ :n:'QtGui' \ :o:'. ' \ :n:'QGraphicsSimpleText' \
\ :n:'t' \ :o:'. ' \ :n:'setPos' \ :p:'(' \ :o:'-' \ :mi:'5' \
\ :n:'items' \ :o:'. ' \ :n:'append' \ :p:'(' \ :n:'b' \ :p:')' \
\ :n:'items' \ :o:'. ' \ :n:'append' \ :p:'(' \ :n:'cs' \ :o:'. ' \
\ :n:'items' \ :o:'. ' \ :n:'append' \ :p:'(' \ :n:'cs' \ :o:'. ' \

\ :k:'for' \ :n:'i' \ :ow:'in' \ :n:'items' \ :p:':' \
\ :n:'scene' \ :o:'. ' \ :n:'addItem' \ :p:'(' \ :n:'i' \ :p:')' \

```

Neat, isn't it?

You can get it at the [ChipScene google code project](#) including, of course, the [source](#).

For a project I am doing for one of my customers, I needed a mailing list archive. I looked, and it seems the nicer one is [Lurker](#).

Sadly, there are no RPM packages, so I had to build it manually... and ran into mimelib. Which, looking closer, is the same mimelib from Doug Sauder I used in KRN back in the late 90s.

And which I had a [hand](#) in GPLing, too. And it's still in use. One of the things I always liked about FLOSS: nothing ever *really* gets thrown away.

This is something I have not read about much so:

Well, I teach Linux courses for several companies. In the last year or so, almost every course has one or two or five students coming from the IT area of some government office or another.

That includes local and national government, by the way.

Some of them are already on deployment stages on servers, a few even on desktops.

Here's what I know about this stuff:

- There's no master plan. Every little place is doing its own thing, all in the same direction. That sucks a bit.
- They are going to be in trouble. Lack of planning means people suddenly find their desktop is now Linux, for instance :-). That's not going to be too nice.
- They will do it anyway: The governemnt can't buy a dead rat at a sale, much less MS Office
- Being a semi-known Linux consultant around here is going to be interesting in 2004
- Some people are peddling Linux and are clueless. Others are clued-in, but they are not the ones getting the boss's ears.
- If you are in gov. IT, you gonna have to learn Linux quick.

- Xandros is making a big push here: it seems some of the development for it is [local](#) (they are overselling the local share, though, but this is government, that was to be expected).
- Red Hat and SuSE are not selling themselves too well in that area
- Lots of stuff is happening, in the quiet. For example, it seems that several provincial Education Ministries are shipping all new computers to schools with Linux in it. Of course they probably get reformatted in 48 hours.
- Most servers, specially mail/web/file servers should be switched to Linux before 2006 (this is mostly a guess)
- Expect MS to offer a busload of free licenses to the government soon (a guess, too, but want to bet?)

... or why flexibility is good sometimes.

After some dabbling with Lurker, I decided it's just too much trouble for the prototype site and went looking for alternative mailing list archivers.

I found few, and decided to try [MHonArc](#).

Then I tried to make it fit the rest of the site. Ouch.

It's not that MHonArc doesn't provide a way to customize the generated HTML. It provides one.

A terribly convoluted one where instead of writing a template for MHonArc to fill it has its template and it's filled from the data and from a resource file you write.

For example, I expected writing a template like this (stupid example):

```
<h1> my archives </h1>
${toolbar}
${contents}
```

So that MHonArc would put a toolbar and the contents where I tell it to.

In reality, MHonArc has a hardcoded template like this:

```
IDXPGSSMARKUP
IDXPGBEGIN
  LISTBEGIN
    (AUTHORBEGIN |
     DAYBEGIN |
     SUBJECTBEGIN) ?
  LITEMPLATE+
    (AUTHOREND |
     DAYEND |
     SUBJECTEND) ?
  LISTEND
  DOC?
IDXPGEND
```

And you tell it what IDXPGEND should be, or it uses its default value. And there is one of these things, with similar but different variables, for each kind of page it generates.

So, you can customize your layout by putting things like this in a *resource file*:

```
<IdxPgEnd>
$MY-FOOTER-LINK$
</body>
</html>
</IdxPgEnd>
```

Which is doable. But you end with static pages. How can you make that work well with a modern site without duplicating the whole base layout? Not to mention all the niceties like having your username in the toolbar and whatever.

Generate page templates instead.

I customized the MHonArc layout to generate neat Mako templates that inherit the “real” base layout template.

I only had to be careful to escape the \$ signs as \$\$ if I wanted them to survive until Mako saw them.

Then create a view that processes the templates instead of serving the static files and it works like a charm.

A simple, but neat hacklet, I think.

This has always been possible, going via LaTeX.

However, LaTeX being what it is, you either need to learn it, or you end up with rather plain-looking documents.

While that’s ok for a manual, I want to use Restructured Text for everything involving documents.

So, I looked for another solution. Sadly, I could not make the existing rlpdf writer work, sooooo I decided to write my own tool.

Since I intensely dislike the Visitor pattern involved in writing a regular docutils writer, I adapted my old rst2rst.py and ended with rst2pdf.py which just traverses the tree recursively and writes the PDF using ReportLab.

And it took me about 3 hours to make it work:

- For a limited subset of RST (no footnotes, no links, no decoration)
- For some subset of tables (no col/row spanning)
- With limited “styling” (it’s mostly there, but I need to write a lot of ReportLab styles.

How well does it work... rather well.

Here’s [a generated PDF](#) of [The RestructuredText Primer](#)

Ignore aesthetics, and consider function, it’s pretty good.

It works much better than it did last night.

The main missing/broken things are:

- Tables
- Links
- Footnotes/citations
- Headers/Footers
- References
- Table HEaders/Spanning cells

- Lists that don't start at 1

The rest seems to be in working order and producing decent output already. It can process [the ReST demo](#) and it doesn't look bad: [check it out](#)

I say, good for half a day of hacking, and will now stop, because it's a weekend and I have a family :-)

But that doesn't mean you can't [try it for yourself](#). Just run it thus:

```
python rst2pdf.py myrstfile.txt
```

And you will get (hopefully) a `myrstfile.txt.pdf`

Let me know how it works, and if it doesn't and it's not related to one of the things I mentioned above as broken, send me a test file!

Un post no-nerd, y bilingüe.

Esto es acerca de mi vida personal, nada nerd, si no les interesa, al próximo ;-)

El sábado a la noche mi hijo [Juan Francisco](#) tenía un poquito de fiebre (38°). Lo llevamos al hospital, y le hicieron un análisis de sangre que mostró que tenía bajos los glóbulos blancos.

En resumen: quedó internado 3 días mientras los análisis descartaban todo tipo de enfermedades muchísimo peores que lo que en realidad tenía (Exantema viral). El ya está perfecto, es más, salvo por la erupción típica de esta enfermedad no tuvo ningún síntoma de nada, ni siquiera fiebre, mientras estuvo internado.

Algo importante: el Hospital Materno Infantil de San Isidro es excelente. Doctores, enfermeras, las voluntarias, todos se desviven por los chicos, y es un hospital público. Por suerte, mi cobertura médica va a reembolsar todos los costos de la atención, lo que va a ayudar al hospital, pero el servicio es buenísimo para todos los vecinos del municipio.

This is about my personal life, not nerdy, if you are not interested, read the next entry :-)

Saturday night my son [Juan Francisco](#) had a little fever (100.4F). We took him to the hospital, and blood tests showed a low white cell count.

Short story: he stayed as a patient 3 days while analysis discarded all sort of diseases much worse than what he actually had (viral exanthem). He's all right now, in fact he had no symptoms at all except for the typical rash of this disease (not even a fever anymore).

Something important: The Hospital Materno Infantil de San Isidro is excellent. Doctors, nurses, volunteers, everyone give everything for the kids, and it's a public (free) hospital. Fortunately, my insurance will pay all the costs of caring for him, which will help the hospital, but the attention is awesome (and free) for anyone in the district.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "rst2pdf".

I did a little (very little) more work on [rst2pdf\[1\]](#)

Better:

- Headers

- Footers
- Footnotes (as endnotes, real footnotes are too much work because you have to reflow the text)
- External links (http, email)
- Some styling improvements (meaning: the output is not **so** painful in the eyes)
- Paper size support
- Real separators

Then there are some things you just get because it's done using reportlab:

- TrueType font embedding (check the example below, and look at the monospaced font)
- No hyphenation (although there is a project on the web that claims to have done it, I should check it out)

Still broken:

- No way to put things like page numbers or section names in headers/footers
- The footnotes are not linked to their contents and viceversa
- In fact, no internal links work, including title references
- Tables are pretty broken

And here is the usual version of [the rst demo](#) showing improvements.

```
system-message
WARNING/2 in everything.txt, line 16151
Duplicate explicit target name: "rst2pdf".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "spreadsheet".
```

Really. I already have a functioning [slideshow program](#), a [spreadsheet](#) and I could write a word processor in a week (for some values of "word", "processor", "write", "week" and "a".)

Of course it would be a very awful office suite, but at least it would be **small**. I think I can put all that in 1MB including icons.

I could make stupidsheet pretty good, and tobogan is actually useful (if rough).

1

```
system-message
WARNING/2 in everything.txt, line 16153
Duplicate explicit target name: "1".
```

My tool to convert Rstructured Text to PDF

system-message

WARNING/2 in `everything.txt`, line 16164
Duplicate explicit target name: "spreadsheet".

Yup. Figured out column/cell spans in reportlab, found old rst2rst code that regularizes the table row/columns to be all the same number of elements, and voilá, tables work, including examples like this:

```
+-----+
| header |
+=====+
|      | tr |
+  tl  +-----+
|      | c  | br |
+-----+-----+
|      | bl |    |
+-----+-----+
```

Which generates nicely on PDF now. [See it](#)

The bad side is that it's now not a single file anymore, so I am needing a place to host SVN and whatever, because it's turning into a real project... and googlecode doesn't let me start more stuff.

It's the second coming of the Toshiba Libretto, only this time it's:

- a) Cheaper
- b) Filled with Linux goodness.

It's [the Asus eee](#). And the default UI is some sort of weirdized KDE. And it is going to sell a whole lot.

Really, lugging my current notebook is tiring, not having my own tools when I go to a customer's is annoying. This baby should fix that.

Sure, you Word/Excel/Powerpoint jockeys are going to be annoyed with a 800x600 screen, but it's awesome for ssh sessions.

Because it's probably never going to work, but if it does, I can point out how ugly it was at first ;-)

No, do not use the alternatives. Use the Yahoo User Interface [Rich Text Editor](#). Why?

- It's maintained
- It at least tries to work with all browsers
- You can use it hosted at yahoo
- It's nice and easy to use
- It has an [image upload extension](#) already.

This is something I run into often, and it really is a way to ruin reading. It's jarring. Basically: no book in english ever gets quotes in spanish right.

For example, I was reading *The Eldorado Network* which is a silly but amusing spy-thriller book with subpar pseudo-Heinlein dialog, when out of the blue, I read this: "viva el muerte".

The book's first half is set on civil war Spain, and "viva el muerte" is said to be the motto of a company of moorish riflemen or whatever.

Obviously:

- Noone who speaks spanish has ever proofread this book
- That is *not* anyone's motto
- It took me 5 minutes to stop feeling annoyed at the carelessness of the publishers and the writer.

You see, in spanish, almost every word has a gender, and you have to keep what's called "concordancia de género", let's call it gender agreement.

What does it mean? That when you use a word of feminine gender, you use feminine pronouns, articles, adjectives, etc.

So, well, muerte is feminine, and el is masculine. So, it disagrees, and it's a mistake no spanish-speaker could ever made, not once, not drunk, not drugged, not asleep.

Sure, there are exceptions, both to avoid unpleasant sounds.

For example, agua (water) is f., so it would have to be "la agua" but that sounds ugly, so it's "el agua", just like in english you use "an" sometimes instead of "a".

So, it would have to be "viva **la** muerte".

There is a weird corner case in which "viva el muerte" would be correct, if there was a guy whose nickname was "muerte", since he's a dude, if you cheered for him you would say "viva el muerte", but that's unlikely.

And this happens in every damn book that has quotes from allegedly spanish-speaking characters, and 9 out of every 10 times, it's the same mistake, because english lacks the concept.

Annoying as all hell.

The largest telco in Latin America. Over 16.5M lines. And they can't configure their freaking DNS correctly.

```
[ralsina@monty ~]$ host -t any telmex.com.ar
telmex.com.ar has SOA record ns2.telmex.com.ar. root.telmex.com.ar. 3 3600
telmex.com.ar name server ns2.telmex.com.ar.
telmex.com.ar name server ns3.telmex.com.ar.
[ralsina@monty ~]$ host ns2.telmex.com.ar.
Host ns2.telmex.com.ar not found: 3(NXDOMAIN)
[ralsina@monty ~]$ host ns3.telmex.com.ar.
Host ns3.telmex.com.ar not found: 3(NXDOMAIN)
```

Before you ask, yes, their domain is really telmex.com, but they did register telmex.com.ar and have a lame web redirector at www.telmex.com.ar.

IN fact, according to www.nic.ar, that should be ns2.telmex.net and ns3.telmex.net so it's plain and lame misconfiguration (BTW: what happened to ns1?).

People of Telmex, you can hire me to fix it for you. I won't overcharge you. I will however, snicker a little.

This is just lazy, guys! Anyone who can guess change when buying a pack of mints can guess this better than what you wrote!

Quote:

Sales figures since the release have been astounding, with 200 pieces snapped up in 20 minutes on Taiwan's shopping channel, ETTV Shopping - aver-aging an Eee PC sold every two seconds.

Let's see, that would be 200 pieces in 20 minutes, 10 a minute, I guess that's one every 2 seconds plus/minus 4 seconds. Or rather, just plus 4 seconds.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "here".

Read the whole Asus press release [here](#).

system-message

WARNING/2 in `everything.txt`, line 16295

Duplicate explicit target name: "here".

Maybe I was spoiled by Pine.

Here's how you read your mail using it:

- Press I to go to the inbox message list.
- You will see the first unread message.
- Press R to reply, F to forward, D to delete, I to go back to the inbox.
- Repeat until you have no more mail.

Yes, this is simplistic, and it's nicer to be able to see the mailbox at the same time as the messages, but you can go through **a lot** of mail this way.

It's old fashioned and crappy in many ways, but it seems done by people who actually read mail using it.

I am liking kmail well enough for the last year or so, but I decided to check alternatives, just in case I was missing something.

I had heard good things about Claws, a fork of Sylpheed, and it does the same crappy thing most other GUI mail clients do:

- Open the INBOX.
- Click on the mail you want to read.
- You can forward or reply.
- But if you want to delete... the message list shows that you shifted to the next message, but it won't display. You need to press enter, then wait until it shows (no disconnected IMAP?).

Dude, let me go through my mail fast, please? It's perhaps my most important task of the day, I need to do it every 30 minutes, I need it to be as fast as possible. I want a queue. I want triage.

I see a message, I must either:

- Delete it because it's useless
- Reply
- Forward

- Tag as important/for later/todo/whatever.
- Save to another folder

That's it. It's not a difficult workflow!

And I need to do this without waiting for the messages to download one at a time!

If I wanted to work that way I would use a freaking webmail.

Kmail lets me do these things. So far, it's the only GUI IMAP client that does.

Anyone has a suggestion for a lightweight IMAP client that works that way?

And no, I am not going to install evolution just to see it I can do the same thing using a larger program that drags half of gnome along :-)

One of my customers has about 15000 email accounts.

I use extensively RBLs to keep the unwanted connections to a minimum, and many other spam detection techniques, many of which involve DNS lookups. Plus all the lookups caused by the email of 15000 users.

It turns out that according to our ISP, we were one of the top ten DNS users in their network, and we were killing their servers.

That got fixed, but it seems lately we were killing our own DNS servers, too.

BIND would stop answering to lookups every once in a while and had to be restarted. Since that was annoying, and the software was up to date, and there seemed to be nothing wrong with the configuration, I did what I could think of, and started looking for an alternative.

PowerDNS's recursor works well, is lighter, it's easier, and works like a charm (but you really should use their multi-forward patch).

So, thumbs up for PowerDNS, and I am still keeping BIND for our authoritative zones, at least for a while, since that part seems to still be working correctly.

I have managed to create the most bizarre way to write a spreadsheet engine in Python.

I still need to polish some things, but here are the highlights:

- Your formulas compile to C
- C is inlined using [Instant](#)

Yes, that means you edit a cell in the GUI and you need to wait until gcc **compiles the thing**.

Is it going to be useful? Probably not. Is it cool? I say yeah.

I will polish it somewhat, create a standalone engine, and show it here.

Can you be a computer consultant and generally a computer guy without ever installing windows?

I managed for over 12 years... [until this week](#).

He was famous, he wrote 30 books, he got to sleep with Marilyn Monroe, and he still had time to provide some of the best anecdotes *ever*.

Check them out at [10 Zen Monkeys](#)

Or at least my article about it.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "reddit".

Also, check it out at [reddit](#).

system-message

WARNING/2 in `everything.txt`, line 16406

Duplicate explicit target name: "reddit".

Since this blog just broke the 100K visitors barrier yesterday (although it had about 150K more when it was `lateral.pycs.net`), it's a good time to revisit some of the old stuff that was somewhat good.

So here are (IMVHO) the best ten things I remember writing in this blog in the last 7 years.

1) [Data-aware widgets in PyQt](#)

This article describes a cool (again, IMVHO) way to implement DB-backed apps using PyQt. It's short, working code and you end being able to create neat stuff. I liked it, noone else did.

1) [Be a good lamarckian froggy](#)

It has it all! Evolution theory (theories)! It pretends to provide insights into FLOSS! Movie-critic-like quotes in the comments!

best blog i've read in a long, long time.

—Aaron Seigo

1) [Rapid Application development using PyQt and Eric3 ... in realtime!](#)

An original premise, a semi-useful app written, got good reviews. I still like it, but sadly it's not a format that ages well, since you can't update the tutorial for newer versions of PyQt.

1)

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "squid authentication via pop or imap".

[Squid authentication via POP or IMAP](#)

It solves a real problem, does it elegantly, and I still am installing it.

1) Shared: [Narnia](#), [The Da Vinci Code is Broken.](#), [Kong at dawn](#), [Matrix Revolutions](#), [Troy \(not McClure\)](#), [Double feature at the Electric](#)

I sometimes try to "review" movies in an oddball way. Please read them if you saw the movies. I think I made sense.

1) [Skeletons of stories that won't ever be written.](#)

I have no idea why I wrote it, but I still like it.

1) [The world cup and I](#)

Too sentimental, but hey, I **did** feel that way.

1) [Frodo as a Hacker](#)

The subtitle is "Shameless explosion of nerditude." and it is that. I am at the same time very ashamed of writing it, and rather amazed by it.

1) [The Linux Booting Process Unveiled](#)

One of my most popular articles. It's even cited as a reference on [Linux booting](#) in Wikipedia! (I edited it because they had the link wrong, though). It was even copied without attribution a couple of times.

1) [A Modest Usability Improvement](#)

Other articles had more links, more views, or more comments, but this article inspired the creation of two new apps that are much better than what was around before I wrote it, which makes me feel all warm and fuzzy inside. So, check it out, then use [Speedcrunch](#) or [Abakus](#) instead of kcalc (or wincalc).

system-message

WARNING/2 in `everything.txt`, line 16463

Duplicate explicit target name: "squid authentication via pop or imap".

Try this and be amazed:

```
$ cat < /dev/tcp/gsmtp163.google.com/25
220 mx.google.com ESMTP 12si345086nzn
```

Be honest: did you know bash could do that? I didn't until finding it in the [man page](#)

Now try this and be amazed it doesn't work (yes, it's in the docs):

```
$ cat /dev/tcp/gsmtp163.google.com/25
cat: /dev/tcp/gsmtp163.google.com/25: No such file or directory
```

And remember, on unix everything is a file, but maybe that file is only there in some very specific circumstances.

UPDATE: There is a chance this will not work in your distro, specifically Debian.

Well, if you are one of the 6 who actually checked the link in the previous entry, you will know I am writing a news aggregator ;-)

It is progressing quickly. Now it is almost a real application, and only 3 or 4 important features are still missing.

The only thing I wish is that there was a KDE 3.1 python binding, since I would need it to use KHTML :-)

Qt's QTextBrowser is nice, but it is not a HTML rendering engine, and it would give it a nice integrated feel. Ok, eventually PyKDE 3.1 will come out, I suppose :-)

PS: if you follow the Pictures link on the left side, you can see a new screenshot!

... I was totally wrong. Read [Dabblers and Blowhards](#) for the real thing.

Inspired paragraph:

Great paintings, for example, get you laid in a way that great computer programs never do. Even not-so-great paintings - in fact, any slapdash attempt at splashing paint onto a surface - will get you laid more than writing software, especially if you have the slightest hint of being a tortured, brooding soul about you. For evidence of this I would point to my college classmate Henning, who was a Swedish double art/theatre major and on most days could barely walk.

Dear fictional character that oppresses the workers of the North Pole:

This christmas, I want an Asus eee PC, an Everex gPC, and some bare white box with a nice Phoenix PC 3.0 BIOS.

Why am I asking the red menace from the north for these items?

Well, they do have one thing in common: **Linux**. Another is that they are consumer boxes, not servers.

For many years, one of the huge advantages windows had was that it came preloaded with most PCs. This enabled people to turn a blind eye to [windows installation and configuration](#) since it was done by Someone Else (TM).

Since getting Linux has become much easier in the last 10 years[1]_ this has been very frustrating. Imagine you had something you gave away for free, but people kept using something more expensive because they *had to pay for it anyway!*

That itches. If Linux was not chosen because it was inferior for the task at hand, that's one thing, but not even being able to be tested because the other product was bundled and paid for? Annoying.

Of course on servers this worked differently. The OS was not the expensive part, and was preloaded less often. Corporations have prearranged licensing terms, and adding things to the mix is simpler.

But for consumers, preloading has been a huge problem[2]_

So, if the jolly trespasser brings me what I ordered, I will find the following:

- **Asus eee**: A cheap subnotebook with Linux and KDE preloaded.
- Everex **gPC**: A cheap Desktop with Linux and Enlightenment(!?) preloaded.
- **Phoenix PC 3.0 BIOS**: an embedded hypervisor and Linux OS.

The eee is probably the most appealing. It's ideal for many uses:

- Salesmen who are now using some ungodly Blackberry app (or worse)
- System and network admins. Really. I would love to have a cheap notebook I won't hesitate bringing to a roof, a bar, the beach, whatever. It would live in my bag. My current notebook? Besides weighting 8 pounds, it's expensive and large. All I need are webpages email and SSH sessions!
- Kids and students (it's *cheap!* You can buy a replacement if he drops coffe on it!)
- Basic users and old people. Really, an office-like thing and a web browser? And I can use it wherever there's wifi? Neat.

And it is going to get a lot cheaper, and it's going to get a lot better. I expect there will be a 32GB, 10" model by the end of next year for \$350, and the current model available for \$250 (after all, half the components are cheap as dirt already, only flash is expensive, and that's a fluke)

And so on and so forth. If Asus creates a decent dock[3]_ and a nice rdiff-backup-based backup solution (it should be **at least** as nice as Apple's Time Machine), this box

turns into my main computer whenever I am at home, and is a useful tool on the road. I really **can** live with those specs.

The **gPC** is a bit harder to grasp.

First, it's even cheaper. \$200 is cheap. The CPU is slowish, but there are a whole range of tasks that are not CPU bound. I really want one of those as a **home server**. This is the first time I can see one of these ITX boxes as actually **cheap** not just small (in fact this one is not small at all).

- I have a TV capture card, I could make a PVR out of it using **LinuxMCE**? It does have enough CPU for that (since I am doing it with a slower box already)
- A file server? More than good enough for that.
- A houseguest computer?
- A **MPD** server?
- All of the above?

And do all this while being quiet and power-efficient? Neat!

And the **Phoenix PC 3.0 BIOS** simply would be cool because I can virtualize without jumping through **any** hoops. This one is still fuzzy for me, but I only found out about it today. I need time for things to grow.

Why do I think these boxes mark a trend? Because they are definitely low-end products. These are meant to be made by thousands and hundreds of thousands, and make small money on each.

The makers are being smart about providing as little functionality as they can and making them simple, niche, consumer products instead of monstrously powerful Linux monsters (sorry for how ugly that sounds).

Another factor is the huge growth of web apps *that work well on non-IE browsers*. This is making the OS irrelevant just like Netscape hoped in 1996. **If the OS is invisible, Linux won.**

So, Santa, for this christmas I ask for all these toys,
and if it has to be only one, please make it the Asus eee.

Roberto Alsina

PS: and if you don't do your part, the reindeer's a goner!

Consider this:

- My Windows installation [article](#):
 - Digg: [3 diggs, no comments](#)
 - Reddit: [137 points, 345 votes, 154 comments](#)
- A post about [linux preloading](#):
 - Reddit: [-2 points, 8 votes, no comments](#)
 - Digg: [21 diggs, 2 comments](#)

I expected both sites to have more or less the same audience, and if anything, digg to be larger, thus give more votes.

Anyone has even a wild guess at the divergence?

This has to be the slowest argument in history. But it's starting to heat up!

Some things happened (or not, or maybe some did) in [the 70s](#) at MIT. Then in 2002, RMS gives [a speech](#) about them. Then near the end of 2007, comes [the rebuttal](#).

I expect RMS to respond sometime around april 2008.

Today I found [Paleo Future](#) via reddit.

- 1) Something like Wernher von Braun's [Shuttle](#) images look a ton better than the CG used today
- 2) The future is, of course, always going to be different from what we think now
- 3) I still really want a flying car
- 4) The nice things we have, noone guessed (like cellphones which work as phones always did instead of as radios)

1

```
system-message
WARNING/2 in everything.txt, line 16633
Duplicate explicit target name: "1".
```

Look, no need to create 30 floppies! You can add a driver without recompiling the kernel! You don't need to know what a Modeline is!

2

```
system-message
WARNING/2 in everything.txt, line 16635
Duplicate explicit target name: "2".
```

The other one is probably games, but that's a different problem. There are whole markets where gaming is not an issue.

3

```
system-message
WARNING/2 in everything.txt, line 16637
Duplicate explicit target name: "3".
```

A wireless dock with place for a IDE disk or two, an optical drive, a powered USB hub, a card reader. Hardware costs? Maybe US\$ 50 + disks?

5) This blog is very readable! Welcome to my RSS feed collection :-)

And that's sad, because I don't ask for much!

I have read everything for the last three years in some sort of Sony Clie. First a [SJ30](#), and after its sad demise in a coke accident, a [SJ20](#) and am very **very** happy with them.

Now, as usual I may be a bit eccentric in that my favourite format for ebooks is TXT files. They are easy to manage, you can read them on anything, and they are gorgeous to look at if you use [Palm Fiction](#), the best etext reader by far for any platform I've seen.

Now, what do I want?

- e-ink display
- A display that doesn't take a second to refresh
- One week of battery life
- A way to put a bazillion texts in it using some sort of card (I have about 3500 books in zTXT format in a memory stick right now)
- Light (below 300 grams)

That's all. OK, I also want it to be cheap and at least open-ish, since I would prefer to be able to write apps for it, but that's secondary (I never wrote a Palm app anyway). And sadly, the Kindle fails in:

- Not cheap enough (\$400????) Plus a subscription for most useful stuff??
- Not fast enough refreshing the screen, according to the demo video. And even if you can live with the slowness (I probably could) it flashes in an incredibly annoying way.
- Not open enough (they charge you to upload files to it by email?)
- It's incredibly ugly. It looks like a gadget from Star Trek: TOS. That's seriously not good. I know that was not a requirement, but baby jebus cried a little when he saw it.

Of course the e-Ink display looks great as usual, and the battery life is excelent, and we are getting closer to a decent product, but not there yet :-)

I have decided to make this blog available also in Spanish starting yesterday.

Since it's generated by my own software called BartleBlog, that meant I had to implement everything so each post can have multiple translations.

And it's *starting* to work. You may notice several things:

- Below the banner there is an "Available in English - Español" thing.
- Each Post has links to the different translations for which it's available.

The location and styling of these elements **will** change, and they **may** be broken in different ways for a few days. Sorry about that.

There will also be a "Lateral Opinion en Español" RSS feed soon.

Update: [here it is](#)

You start a blog, you like something another guy wrote, what do you do?

- 1) Link and write a nice comment
- 2) Add it to del.icio.us, have feedburner show it in your "Links of the day" feature?
- 3) Copy the whole thing, including the name of the author, not following his very liberal license, ignore his request to fix it, and piss him off?

If you chose option 3, then you will get along just fine with [this guy!](#)
 I implemented story translation today, and why not start with this one.
 Here it is [Windows: mi eXPeriencia](#)

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "rater".
```

Checking on my semi-dead projects, I found that one was almost finished but I had forgotten about it: [rater](#)

In order to make it really useful, however, I need a C programmer that can turn this python program:

```
\ :c:#!/usr/bin/env python\
\ :c:# -*- coding: utf-8 -*-\

\ :kn:'import' \ \ :nn:'sys' \
\ :kn:'from' \ \ :nn:'socket' \ \ :kn:'import' \ \ :o:'*' \
\ :n:'serverHost' \ \ :o:'=' \ \ :s:'localhost' \
\ :n:'serverPort' \ \ :o:'=' \ \ :mi:'1999' \

\ :n:'s' \ \ :o:'=' \ \ :n:'socket' \ \ :p:'(\ \ :n:'AF_INET' \ \ :p:', \ \
\ :n:'s' \ \ :o:'. \ \ :n:'connect' \ \ :p:'(( \ \ :n:'serverHost' \ \ :p:', \ \
\ :k:'print' \ \ :s:'"Sending: "' \ \ :p:', \ \ :s:' ' \ \ :o:'. \ \ :n:'join' \
\ :n:'s' \ \ :o:'. \ \ :n:'send' \ \ :p:'(\ \ :s:' ' \ \ :o:'. \ \ :n:'join' \
\ :n:'data' \ \ :o:'=' \ \ :n:'s' \ \ :o:'. \ \ :n:'recv' \ \ :p:'(\ \ :n:'mi:' \
\ :n:'sys' \ \ :o:'. \ \ :n:'stderr' \ \ :o:'. \ \ :n:'write' \ \ :p:'(\ \ :n:' \
\ :n:'sys' \ \ :o:'. \ \ :n:'stderr' \ \ :o:'. \ \ :n:'flush' \ \ :p:'()' \
\ :n:'sys' \ \ :o:'. \ \ :n:'exit' \ \ :p:'(\ \ :nb:'int' \ \ :p:'(\ \ :n:' \
```

Into a nice function that never fails and never leaks memory (of course, it should return instead of exit, this is just example code ;-).

If that's done, I can release rater as a useful tool, which should find a home in many gmail installations (and maybe other uses).

```
system-message
WARNING/2 in everything.txt, line 16806
Duplicate explicit target name: "rater".
```

I mean, can you do this on a GUI?

```
grep 23/Nov/2007 /var/log/httpd/lateral-access_log | \
grep html | cut -d" " -f1 | sort |uniq | \
while read IP; do host $IP; done
```

The KDE project is relicensing some code. There is a tool to verify if any SVN account holder is blocking the relicensing of some file.

Hey, looks like [I was!](#)

I don't expect to have any actual code there since I have not editd those files in perhaps 6 years, but it costed me nothing to approve it ;-)

I have said Linux sucks. Here's my new project: Make it suck less, one bit at a time.

And I start with one of the big chunks: configuration.

[Part I](#) in a longish series (I expect 5 parts at least).

- If you don't know what [qmail-spp](#) is, please check it out. It makes qmail much much better.
- If you know qmail-spp, then maybe [my plugin collection](#) will be handy for you.
- My most useful plugin is probably ipthrottle, which you can use to make overea-ger IPs connect less often.
- The version currently in SVN will autoblock those IPs for a configurable amount of time if you are using [ipsvd](#) which is like tcpserver, only much better.
- I really need someone to help me test the SVN version, which should be way, way better than the releases on the page.
- The SVN repo is [at googlecode](#)

Ok, not really, but if everything works right, I get a "Galaxy Black" Asus eee 4G Surf in a couple of weeks :-)

Ok, I wanted the 4G, not the surf, but my mother in law is bringing it as a favour and I have no use for the webcam anyway, so the only difference is the soldered RAM, which I was not planning to upgrade anyway.

According to [Slashdot](#), at least.

That means there can be a real community project to integrate all the patches float- ing around!

That means qmail will **stop** sucking without need for manual labour!

I am just thrilled by this.

I had to do a vry late night job yesterday, so I am not going to do it right today. Probably wednesday.

I am still anxiously waiting for my Asus eee (still a week to go or so), and I was thinking about web browsing in the 7", 800x480 screen.

Yes, you can do the usual things, go to full screen mode, whatever, but for many sites, 800 pixels is just too narrow.

You can make pages fit better in a limited width by reducing the font size. A good 8pt font is probably readable on that screen, since you will be pretty close to the screen.

However, that does nothing for layouts that are pixel-based, and for the size of images.

However, after reading about Qt 4.4's support for webkit and widgets in QGraph- icsView, I started thinking... that's *doing it wrong*.

What you want in limited screen space is text in the usual size and the page in a smaller size.

So, my idea is:

- Make the font larger. Say, 12pt.
- Render the page in a webkit widget that's 1200x720 pixels.
- Put the widget in a QGraphicsScene, and reduce it 33%.

The result? a 800x480 web page view that contains the whole page.

Of course you should be able to change all these parameters with a single control.

Is the page too wide to see in 800x480 with 8pt fonts?

Then try 1000x600 with 10pt fonts, and reduce it to fit the screen.

Still too wide? Then try 1200x720 and a 33% reduction.

Possible tips:

- Should work better with aliased fonts, since the size reduction should make anti-aliased fonts too blurry. A wide, open font will look better,
- The scrollbars will look squashed, I expect. Probably checkboxes and radio buttons will look funky. Here, more advanced wizardry is needed.

Of course it would have to be tested, but I am willing to bet this will work better than other alternatives. If I had a working PyQt 4.4 I would try to do it myself ;-)

The day is arriving when Net Managers SRL will emerge from its cocoon of bureaucracy.

Net Managers is a company. Of which I own a piece. And it will have a website, at <http://www.netmanagers.com.ar> ... and I have no idea what to put there.

So, what should it be like?

Here's some data:

- It's formed by Linux/Unix geeks of a rather high level, IMHO
- Our business is fixing your Linux/Unix problems, and explaining to you how you can really use Linux/Unix to do cool things (which means, corporate Linux consulting and support)
- We have a relaxed corporate culture. Every partner would get the joke about how Christmas and Thanksgiving are the same, since 25dec==31oct. We would prefer to actually show it. We are usually hired by IT grunts, not suits.
- I have a blog and it's really not [enterprisey](#) , but it shows (I think) that I know about this. Should it be linked?
- We have no physical offices. Legally we do, but really, we operate out of each partner's homes, on two different provinces. So, no pictures of our corporation's digs available ;-)
- We are sort of a startup, but more like a formalization of a way of working that has been going on for a while.

Ideas, suggestions, are welcome.

Of course, questions about our services are welcome, too ;-)

I finally got my eee PC last saturday. It's the 4G Surf in Galaxy Black[1].

Everyone says the same thing, and so do I: you can't understand how small the thing is until you see it.

And then everyone takes a picture of it sitting inside its previous notebook. So will I, 2 times.

Here's the eee with a HP Pavillion zd7000, which has a 17" widescreen:

Here's the eee with a Toshiba Satellite with a very unusual 16.6" 4:3 ratio screen:

But is it the smallest notebook I ever had? Nope.

Here you can see the eee, a Toshiba Libretto and a HP Jornada 720 laying over the HP notebook, so you can get an idea of how much smaller all are. The Libretto is smaller but thicker and feels heavier.

Regarding construction quality, the screen is decent, if you can live with the low resolution (I can). The keyboard is ok, even though I have large fingers[2] and the general construction feels good (not creaky, no flex[3]), but nothing remarkable.

The software... it works. But I am looking to replace it with another distro ASAP. Let's get into some detail...

- 1) Xandros package availability is abysmal. There's little, what's there is old, what I like is usually missing, if you start pulling Debian packages it will break, and if you don't want to use the Xandros File Manager you may have to do evil stuff[4]
- 2) KDE 3.4 is worse than 3.5. There's no kopete?
- 3) The menus are incomplete (in both the simple and advanced modes). There are a bunch of things installed but not showing.
- 4) If you have only 4GB of storage, little RAM, and a slowish CPU, building from source is probably not a good idea, so I can't install that way even if I felt like it.
- 5) No PyQt4? That means I can't blog from it :-)
- 6) On the other hand, everything in the eee works using xandros, and I don't know if it will on another distro.

The only changes I made so far are:

- 1) Switched to full desktop (KDE) mode.
- 2) I got rid of the silly unionfs situation (BTW: I did it using the instructions at <http://wiki.eeeuser.com>, but used RIPLinux as the USB bootable distro, it's the easiest of them all)
- 3) I removed a lot of garbage (got 2.2GB free now)
- 4) Moved logs to a tmpfs

Other than that, it's still the original stuff, and I have been using it to work around the house while watching the baby, and from bars, and such.

Happyness-meter: 8 out of 10 so far.

I got myself a [keyboard from BenQ](#) today. I had bought one for a client yesterday, and it's just too cute to pass. It's roughly a notebook keyboard, only USB, and thin as a crêpe :-)

It has very nice touch, and it's... well, small and cute:

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

You can read an interview with the big boss at BenQ [here](#) , which is weird in that eastern-guys-are-very-different way.

system-message

WARNING/2 in `everything.txt`, line 17041
Duplicate explicit target name: "here".

1

system-message

WARNING/2 in `everything.txt`, line 17018
Duplicate explicit target name: "1".

A huge thank you to Feray Girgin, my mother in law who brought it from the US :-D

2

system-message

WARNING/2 in `everything.txt`, line 17020
Duplicate explicit target name: "2".

My hands look like crippled obese octopuses (octopii?)

3

system-message

WARNING/2 in `everything.txt`, line 17022
Duplicate explicit target name: "3".

Being so small, it's not supposed to flex even if it were badly built, I guess.

4

system-message

WARNING/2 in `everything.txt`, line 17024
Duplicate explicit target name: "4".

I ended symlinking konqueror to xandrosfilemanager

As I said before, the bastardized Xandros[1] sucked. In fact, I managed to destroy it. So I decided to switch to another distro. **Any** distro.

Enter [eeeXubuntu](#), a eee-tuned Xubuntu[2].

Since I prefer KDE to XFCE, I started hacking at it, and produced, IMVHO, a much nicer thing.

As you can see, it takes slightly less space than eeeXubuntu's default, and a whooping 1GB less than the default distro.

How did I do it? First, by deciding what I wanted.

- No OpenOffice. Koffice is enough to read docs, then there is Google docs, and I produce my texts using [docutils](#)
- No printing. I have not used a printer in 5 years. What are the odds I will use one from the eee?
- wicd for network configuration. It works for wireless/wired/encrypted and has no problem whatsoever reconnecting after suspend.
- yakuake for terminal. It uses the least screen-space possible: none if you are not using it.
- Firefox + Flash (I would use Konqueror, if flash9 had not broken it)
- Kmail + Akregator for mail and RSS

It works great, boots in 35 seconds (and I have not tuned that, yet).

Here's the [list of installed packages](#) and here's the [list of repos](#) I am using

Maybe someone more enterprising will make a real distro out of it.

UPDATE: you can get this program now [at google code](#)

I have been using my eee for a while already with a sort of Kubuntu in it.

However, my favourite wireless/wired network management app is wicd, which is a Python/GTK application. Or was, since version 1.4.1 requires python-gnome2-extras.

Which depends on ...

```
libart-2.0-2 (>= 2.3.18), libaspell15 (>= 0.60), libatk1.0-0 (>= 1.13.2), libbonoboui2-0 (>= 2.15.1), libc6 (>= 2.6-1), libcairo2 (>= 1.4.0), libfontconfig1 (>= 2.4.0), libfreetype6 (>= 2.3.5), libgconf2-4 (>= 2.13.5), libgda3-3, libgdl-1-0, libgdl-gnome-1-0, libgksu1.2-1 (>= 1.3.3), libgksu2-
```

1

system-message

WARNING/2 in `everything.txt`, line 17085
Duplicate explicit target name: "1".

Which is itself a bastardized Debian etch.

2

system-message

WARNING/2 in `everything.txt`, line 17087
Duplicate explicit target name: "2".

Which is a XFCE-tuned ubuntu

```
libgksuui1.0-1, libglade2-0 (>= 1:2.6.1), libglib2.0-0 (>= 2.14.0), libgnome2-0 (>= 2.17.3), libgnomecanvas2-0 (>= 2.11.1), libgnomeui-0 (>= 2.17.3), libgnomevfs2-0 (>= 1:2.17.90), libgtk2.0-0 (>= 2.12.0), libgtkspell0 (>= 2.12.0), libice6 (>= 1:1.0.0), libnspr4-0d (>= 1.8.0.10), liborbit2 (>= 1:2.14.8), libpango1.0-0 (>= 1.18.2), libpng12-0 (>= 1.2.13-4), libpopt0 (>= 1.10), libstartup-notification0 (>= 0.8-1), libx11-6, libxcomposite1 (>=1:0.3-1), libxcursor1 (>> 1.1.2), libxdamage1 (>= 1:1.1), libxext6, libxfixed3 (>= 1:1.2.0), libxml2 (>= 2.6.29), libxrandr2 (>= 2:1.2.0), libxrender1, zlib1g (>= 1:1.2.11), python-support (>= 0.3.4), python (<< 2.6), python (>= 2.4), python-gtk2, python-pyorbit, python-gnome2-desktop
```

In short: a 87MB download. That can't be good. In fact, there are almost no changes from 1.3.1 which didn't require all that! Except for one change that makes all the difference on a eee PC: vertical resizing to under 400px. :-(

So, because I am who I am, I did this:

It's a replacement for wicd's gui.py and tray.py. Only needs PyQt4 which I already had and depends on:

```
libc6 (>= 2.6-1), libgcc1 (>= 1:4.2.1), libqt4-core (>= 4.3.2), libqt4-gui (>= 4.3.2), libstdc++6 (>= 4.2.1), python-central (>= 0.5.8), python (<< 2.6), python (>= 2.4), python-sip4 (>= 4.7), python-sip4 (<< 4.8)
```

See a difference there?

Took me about 3 hours to hack together, and works (except for wired config, the prefs dialog, static IP and scripts) but the hard work is done.

If anyone wants a copy, just ask. I expect KUbuntu could use something like it?

My rst2pdf script has had several things happen to it.

- a) It got another guy working on it: Christoph Zwerschke
- b) It's on googlecode now: <http://rst2pdf.googlecode.org>
- c) Christoph made a number of improvements:
 - bulleted and enumerated list simplified, use same font as text for bullets and numbers
 - links in table of contents work
 - compress literal sections horizontally so that they always fit on the page
- d) I have integrated hyphenation using [wordaxe](#) (works only with reportlab 2.1)

The output using hyphenation is really ugly right now (for example, I get a black square instead of an hyphen) but it's a small step forward.

If you are in the Buenos Aires subway, this is the number you dial if you have an emergency:

Good thing they limited it to 5 figures and phones don't let you use punctuation. I really, really, **really** would like to know why they decided to use this number. It's not easy to type, it's not particularly easy to remember, it's not used for emergencies anywhere else, as far as I know. Is it a case of a math nerd with power?

BTW: this is the first time I buy a cellphone with a camera. It also happens to have bluetooth, and it works flawlessly with Linux.

The eee is small. It has very little storage. So, why should I use dozens of MB on a word processor? Because I get word documents in the mail every once in a while.

In fact, the only word documents I get are error messages from windows users. Here's the procedure **every one of them** seems to have found to tell me what's inside explorer's error pages:

- 1) Capture the screen
- 2) Paste it into Word
- 3) Mail it to me

I suppose copy&paste of the page **contents** was too hard. But anyway, I usually manage by having kword or abiword handy, but I was thinking...

I am reading my email. That means I am on the net. That means google docs is right there. And they have a python API! 15 minutes later... `open_with_gdocs.py`

```
\ :c: '#!/usr/bin/env python'\
\ :kn: 'import '\ \ :nn: 'gdata.docs.service'\ \ :o: ', '\ \ :nn: 'gdata'\ \ :o:

\ :c: '# Create a client class which will make HTTP requests with Google Docs'
\ :n: 'client'\ \ :o: '='\ \ :n: 'gdata'\ \ :o: '.'\ \ :n: 'docs'\ \ :o: '.'\ \
\ :c: '# Authenticate using your Google Docs email address and password.'\
\ :n: 'client'\ \ :o: '.'\ \ :n: 'ClientLogin'\ \ :p: '('\ \ :s: 'joe@gmail.com'

\ :n: 'ms'\ \ :o: '='\ \ :n: 'gdata'\ \ :o: '.'\ \ :n: 'MediaSource'\ \ :p: '('
\ :n: 'entry'\ \ :o: '='\ \ :n: 'client'\ \ :o: '.'\ \ :n: 'UploadDocument'\ \
\ :n: 'os'\ \ :o: '.'\ \ :n: 'system'\ \ :p: '('\ \ :s: '"firefox '\ \ \ :si: '%s'
```

Try it, as long as you have firefox, a decent version of [python](#) and [gdata](#) it should open the doc you pass as first argument on google docs in firefox.

Remember that you need to delete it later if you don't want it, and rename it if you want to keep it with a decent name.

I am not turning it into a real app, but it's good enough for me. Put it in your path and associate it to your .doc files.

A similar thing for .xls is trivial. A script that would handle both, also.

Could please someone take this and make it a real app?

Its dependencies seem to have reverted to something sane lately:

```
Package: wicd
Version: 1.4.2-1
Section: extras/web
Priority: optional
Architecture: all
Essential: no
Depends: python, python-gtk2, python-dbus | python2.4-dbus,
wpasupplicant, python-glade2, wireless-tools
```

I may still keep on working on my Qt frontend but just because I enjoy it. Good job wicd guys!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "1".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "2".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "3".

There have been a few posts on planet KDE about text-based presentation tools (1, 2, 3) and while their solutions are all probably better than mine, I have to say I did something in the area about a year ago.

Since I like [restructured text](#) and PyQt that's what it's based on, but you can just use it with vi and a `wyse60`, if you want.

My project is called [Tobogan](#) and it has a working version.

It has a GUI, but let's ignore it for this post about text-based software ;-) and show the text behind it:

```
\ :gh: `===== `\
\ :gh: `Why use Tobogan `\
\ :gh: `===== `\

\ :gh: `----- `\
\ :gh: `(If you are a nerd) `\
\ :gh: `----- `\

\ :nc: `:transitions: `\ \ :nf: `from_left,to_left,from_top,to_top,from_right

\ :gh: `It's nerd-oriented `\
\ :gh: `----- `\

It's trivial to display source code, with
proper syntax highlighting.

\ :p: `..` \ :ow: `code-block` \ :p: `::` \ python

    from base64 import *
    def myFirstFunction():
        print b64decode ('YnllIHdvcmxkIQ==')
```

Even for things like shell sessions.
Dammit, I am a nerd, I will try to add every nerd feature I deem cool.

```

\ :gh:`It does things simply`\
\ :gh:`-----`\

\ :gh:`This is a subtitle;\
\ :gh:`=====`\

Wanna show a flickr photo?

\ :p:`..`\ \ :ow:`code-block`\ \ :p:`::`\ rst

\ :p:` ..`\ \ :ow:`flickr`\ \ :p:`::`\ myPhotoTitle

How about using openomy.com to share your files?
And whatever else you can think of, it probably
\ :gs:`**can**`\ \ be done.

\ :p:`..`\ \ :ow:`footer`\ \ :p:`::`\ Smart thing goes here

```

And here is the [result](#) after running `rst2sl.py` on it.

Hints:

- If you move the cursor to the top, you will get (broken?) previous/next arrows
- If you click on the window, you get to the next slide.

I have not touched this in almost a year, and it would need lots of love, but it's not a bad concept, IMVHO.

system-message
WARNING/2 in `everything.txt`, line 17310
Duplicate explicit target name: "1".

system-message
WARNING/2 in `everything.txt`, line 17312
Duplicate explicit target name: "2".

system-message
WARNING/2 in `everything.txt`, line 17314
Duplicate explicit target name: "3".

Walking near the train station of Carupá, heading for a customer's I saw a rather shabby house in the industrial area, white and with green painted doors made of metal. And on top of it, standing as if about to jump, there's a statue of a chubby guy wearing a white shirt and black pants. Why is there a statue in the roof of that house? Why is it so creepy?

The awful picture is courtesy of my phone.

I am finishing my first large-ish Django app from scratch [1].

To login a user you can do something like this:

```

\ :p:['\ \ :n:'root'\ \ :nd:'@wally'\ \ :n:'app'\ \ :p:]'\ \ :c:'# pytho
\ :n:'Python'\ \ :mf:'2.3'\ \ :o:'.'\ \ :mi:'4'\ \ :p:'('\ \ :c:'#1, May
\ :p:['\ \ :n:'GCC'\ \ :mf:'3.4'\ \ :o:'.'\ \ :mi:'6'\ \ :mi:'20060404'\
\ :n:'Type'\ \ :s:'"help"'\ \ :p:','\ \ :s:'"copyright"'\ \ :p:','\ \ :s
\ :p:'('\ \ :n:'InteractiveConsole'\ \ :p:')'\
\ :o:'>>>'\ \ :kn:'from'\ \ :nn:'pyodb'\ \ :kn:'import'\ \ :o:'*'\
\ :o:'>>>'\ \ :kn:'import'\ \ :nn:'django.contrib.auth'\
\ :o:'>>>'\ \ :k:'print'\ \ :n:'django'\ \ :o:'.'\ \ :n:'contrib'\ \ :o:
\ :n:'user'\

```

And then you use that user object for `django.contrib.auth.login`

Some of the authentication data is available on a Microsoft SQL Server, and I get it via `pyodb`. And this happens when I try to authenticate the user (assume user and pass are valid, this is the smallest snippet that triggers the bug):

```

\ :p:['\ \ :n:'root'\ \ :nd:'@wally'\ \ :n:'app'\ \ :p:]'\ \ :c:'# pytho
\ :n:'Python'\ \ :mf:'2.3'\ \ :o:'.'\ \ :mi:'4'\ \ :p:'('\ \ :c:'#1, May
\ :p:['\ \ :n:'GCC'\ \ :mf:'3.4'\ \ :o:'.'\ \ :mi:'6'\ \ :mi:'20060404'\
\ :n:'Type'\ \ :s:'"help"'\ \ :p:','\ \ :s:'"copyright"'\ \ :p:','\ \ :s
\ :p:'('\ \ :n:'InteractiveConsole'\ \ :p:')'\
\ :o:'>>>'\ \ :kn:'from'\ \ :nn:'pyodb'\ \ :kn:'import'\ \ :n:'Connect
\ :o:'>>>'\ \ :kn:'import'\ \ :nn:'django.contrib.auth'\
\ :o:'>>>'\ \ :n:'c1'\ \ :o:'='\ \ :n:'Connect'\ \ :p:'('\ \ :s:'"sqlserve
\ :o:'>>>'\ \ :k:'print'\ \ :n:'django'\ \ :o:'.'\ \ :n:'contrib'\ \ :o:
\ :bp:'None'\

```

As you can see, the `pyodb` stuff should not interfere with the `django` stuff at all. I am only importing one function and connecting to the DB, I don't even execute any SQL or use any data from the SQL connection. But the calls to authenticate fail.

How are you supposed to debug this? I worked around it by moving the `PyODB` stuff to another module, but it's *weird*.

I moved to Buenos Aires (BA) almost exactly 8 years ago. For those who have never been here, let me tell you some things about it. It's large. Do you know Sao Paulo? A bit smaller. Much smaller than Mexico DF. About the same size as New York. Twice the population of the Randstad. About the same as greater Paris or Istanbul. So figuring out a way to move around it was important.

The way most natives do it is by bus. There is a pretty extensive and efficient network of buses which will take you anywhere. There are maybe 150 different lines, but if you don't know the city, specially the place you are trying to reach, they are a recipe for getting lost, because you can (will) miss your stop and end anywhere else.

1

```

system-message
WARNING/2 in everything.txt, line 17366
Duplicate explicit target name: "1".

```

It's a list server. Discussion groups, moderation features, subscriptions, the works. Like google groups, but yours and not ajaxy. It's not Open Source software, but if you want something similar, I am sure my customer will license it to you :-D

To make it worse, I get dizzy on buses. The braking and starting makes me really sick. I can control it, as long as I look out the window, or straight forward, and breath really carefully.

So, since I don't drive, and cabs are relatively expensive, I always preferred the subway, or, as it is called here, the subte. Plus, on trains and subways I can even read and not get dizzy. I always tried to live close to a station, I had almost one hour to read while traveling and we got along great.

The subte is pretty old. The first in Latin America, and still the only one in a few million nearby square miles. But it's also ... quirky.

For instance, I lived in Belgrano, close to the D line. Which had japanese cars. How did I know they were japanese cars? Well, they had all these things written on the windows in Japanese. Sadly, I can't find pictures of that, and in a recent trip I didn't see them, so it may be that after maybe 20 years someone decided to rub them off. That's a pity. I always imagined they said interesting stuff, even if they probably said "keep your hands inside the car, you idiot".

There's also the boletería-kiosco. A kiosco is a sort of mini drugstore, where you can buy candy, a soda, maybe a comb, or condoms. A boletería is a palce where you buy tickets to ride the subte. And in some places, you can do both things. Because they turned the ticket booths into kioscos.

Tere is the line at Retiro station, in the C line. There's 4 or 5 boleterías. When you get there, often there's 40 or 50 people in line on the first one. And you can walk just beside them and buy a ticket in the 4th or 5th booth, where there's noone waiting.

And of course, a classic, the one every tourist sees. The A line. The original BA subte, opened 90 years ago or so... and still using the same cars. Yes, you can ride antique, wooden cars to work on that line. With incandescent bulbs on glass tulips. With manual doors (manual opening only, they close automatically with bone crushing force).

Sure, it's hot. There's no air conditioning, and BA can get pretty hot in summer. But it's nice in winter! It's fast, you can't get lost, and it's just so BA.

I am powerless. I mean, I have no power. I mean I have no electricity!

It seems I forgot to pay a bill (although I do have the latest one, and it says I owe nothing), and the power guys came friday and shut it off.

No big deal, I can pay it on monday, but what happens in the weekend... well, I am not going to let a USD 12 bill get me annoyed, so I decided to move to a hotel for the weekend.

Mind you, a cheap hotel, but it's clean, has cable tv and breakfast is included. And **someone else** changes the sheets for clean ones.

So, I did. What do you know, the hotel is full of interesting people one meets over breakfast. And I have the right way to get a conversation rolling:

Me: Where are you from?

Tourist: Colombia/Mexico/Brazil/Netherlands/Whatever. How about you?

Me: I live 5 blocks from here, but I think they cut my electricity by mistake, and am staying here for the weekend.

You can, I suppose, imagine where that goes.

Coming tomorrow or next day, a picture of the hotel's wall-embedded TV remote. You won't believe it.

It should not happen but it does: Your qmail server is overloaded. Maybe you are under a DOS attack, or there is a reason why you are getting 10x your usual amount of mail.

But then you start seeing how your “not preprocessed” queue starts growing, and growing...

This can also mean things like clamav or spamassassin, which need to check the mail before it gets queued are not keeping up with the mail flow, or maybe some IO performance issue.

But what can you do right now to fix it?

Well, you can disable spamassassin, or, in extreme cases, shutdown SMTP so the system has a chance to *catch its breath* so to speak.

Of course, closing SMTP means your own users can’t send email either, which sucks.

Now there is a lighter alternative: shutdown SMTP for those who are *not* your users.

Here’s the trivial code, implemented as a [SPP](#) plugin, fit to be used in the [mail] section:

```
\ :c: `#!/bin/dash
  `
\ :k: `if` \ \ :o: `[` \ -f /var/qmail/control/overloaded \ :o: `]` \
\ :k: `then
    if` \ \ :o: `[` \ -z \ :s2: `"$SMTPAUTHUSER" ` \ \ :o: `]` \
    \ :k: `then
        \ \ :nb: `echo ` \ R451 Temporary Failure: Server overload
        \ :nb: `echo ` \ overload: \ :nv: `$PPID` \ Temporary Failure: S
    \ :k: `fi
fi` \
```

And if you are daring and want to make your system self-correcting, maybe you should cron something like this:

```
* * * * * \ :k: `if` \ \ :o: `[` \ \ :sb: ` ` \ qmail-qstat | tail -1 | cut -c
  ` \ \ :k: `then ` \ touch /var/qmail/control/overloaded ; \ :se: `
  ` \ \ :k: `else ` \ rm -f /var/qmail/control/overloaded; \ :k: `fi` \
```

I will probably code it again in C and make it part of ra/plugins.

The guys at <http://github.com> have been nice enough to add me to their beta program, so I am doing a little project there, to figure out if I like git or not.

Since everyone raves about it, I suppose I will, and then will have to turn my numerous googlecode SVN repos into git mirrors or whatever the correct terminology is.

I was in one of my customer’s datacenters the other day, and while I do most work remotely, I had to take this opportunity to take some snapshots of their proxy server.

This post is just a bit of fun. This is not supposed to be their server, it was just a stopgap measure because of multiple hardware failures. This is a large, well managed company, and this irregular situation will be fixed soon. And anyway, it is working just fine.

As you know, Linux can do that kind of job easy without much hardware requirements. After all, it has to handle at most about 3Mbps of data, and this box has 2GB of ram, so there’s plenty of room for a speedy cache.

So, let’s go for some good news/bad news.

Good news: It’s a true-blue IBM Server! Bad news: It’s a IBM Netfinity 5000 (model [3Ry](#))!

Here is [some tech info from IBM](#) about it.

Good news: It has 2 CPUs! Bad news: They are two 450Mhz Pentium II CPUs.

Good News: it has hot-swappable SCSI discs! Bad news: you have no discs for that controller, so we will use this 8GB IDE (PATA) disk!

And will leave it just sitting there by the CD unit, besides the huge gaping hole in the front where the SCSI discs would be.

Bad News: it has a tendency to overheating! Good News: You have a place to keep your coffee warm!

And you can see it at <http://www.netmanagers.com.ar>.

It's spanish only right now, but that will change in a week or so.

It was an interesting job, because we are using three web apps written in three different languages (php/python/perl) and I wanted to provide our customers with a single login for all three.

It was a bit of effort, but educational :-D

On other news, there is now a "Tipit" link on each post. What's that for? Well, you can use it to give me money. You get to be listed as "guy that gave Roberto money", too (as soon as there is one ;-).

If you delete the empty directory /var/lib/xkb, then enable the KDE keyboard layout switcher, the up-arrow key works like print-screen.

At least on my box it does. It took me a month to figure it out.

Just a snippet of code because every once in a while I need something like the classic memoize decorator but am working on a CentOS 4 box (with python 2.3!)

I am still testing it, and am not even sure it really works, but it should be close.

```
\ :n:'cache'\ \ :o:'='\ \ :n:'memcache'\ \ :o:'. '\ \ :n:'Client'\ \ :p:'([\n :n:'cachetimeout'\ \ :o:'='\ \ :mi:'30'\n\n \ :k:'def'\ \ :nf:'memoize'\ \ :p:'(\ \ :n:'fun'\ \ :p:'):\n \ :k:'def'\ \ :nf:'inner'\ \ :p:'(\ \ :o:'*\ \ :n:'args'\ \ :p:', '\n \ :n:'key'\ \ :o:'='\ \ :nb:'repr'\ \ :p:'(\ \ :n:'fun'\ \ :p:')'\n \ :n:'cached'\ \ :o:'='\ \ :n:'cache'\ \ :o:'. '\ \ :n:'get'\ \ :p:\n \ :k:'if'\ \ :n:'cached'\ \ :ow:'is'\ \ :bp:'None'\ \ :p:': '\n \ :n:'val'\ \ :o:'='\ \ :n:'fun'\ \ :p:'(\ \ :o:'*\ \ :n:'a\n \ :k:'print'\ \ :s:'"Setting: "'\ \ :p:', '\ \ :n:'key'\ \ :p:\n \ :n:'cache'\ \ :o:'. '\ \ :n:'set'\ \ :p:'(\ \ :n:'key'\ \ :p:\n \ :k:'return'\ \ :n:'val'\n \ :k:'return'\ \ :n:'cached'\n \ :k:'return'\ \ :n:'inner'\
```

And later inside a class:

```
\ :k:'def'\ \ :nf:'myfun'\ \ :p:'(\ \ :bp:'self'\ \ :p:', '\ \ :n:'arg'\ \ :p:\n \ :p:': '\n \ :p:': '\n \ :n:'myfun'\ \ :o:'='\ \ :n:'memoize'\ \ :p:'(\ \ :n:'myfun'\ \ :p:')'\
```

And that's it. The basic idea I stole from a blog who was inspired by a Paul Graham book. It can be trivially turned into a decorator, of course (but then only works on 2.4 and later).

Suppose you want to send SMS messages from windows through a bluetooth connection to a phone.

I am sure you can make it work. On the other hand, I already had it working on Linux... so you can just use this on a friendly Linux box, and send SMS messages by accessing a special URL:

```
\ :c: '#!/usr/bin/env python'\
\ :kn: 'from'\ \ :nn: 'colubrid'\ \ :kn: 'import'\ \ :n: 'BaseApplication'\
\ :kn: 'import'\ \ :nn: 'os'\

\ :k: 'class'\ \ :nc: 'SMSApplication'\ \ :p: '('\ \ :n: 'BaseApplication'\ \

\ :k: 'def'\ \ :nf: 'process_request'\ \ :p: '('\ \ :bp: 'self'\ \ :p: ')': '\
\ :n: 'numero'\ \ :o: '='\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'request'\ \
\ :n: 'mensaje'\ \ :o: '='\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'request'\ \
\ :p: '['\ \ :n: 'entrada'\ \ :p: ','\ \ :n: 'salida'\ \ :p: ']'\ \ :o: '='\
\ :n: 'entrada'\ \ :o: '.'\ \ :n: 'write'\ \ :p: '('\ \ :n: 'mensaje'\ \
\ :n: 'entrada'\ \ :o: '.'\ \ :n: 'flush'\ \ :p: '()'\
\ :n: 'entrada'\ \ :o: '.'\ \ :n: 'close'\ \ :p: '()'\
\ :n: 'msg'\ \ :o: '='\ \ :n: 'salida'\ \ :o: '.'\ \ :n: 'read'\ \ :p: '()'\
\ :n: 'response'\ \ :o: '='\ \ :n: 'HttpResponse'\ \ :p: '('\ \ :n: 'msg'\
\ :n: 'response'\ \ :p: '['\ \ :s: 'Content-Type'\ \ :p: ']'\ \ :o: '='\
\ :k: 'return'\ \ :n: 'response'\

\ :k: 'if'\ \ :n: '__name__'\ \ :o: '=='\ \ :s: '__main__'\ \ :p: ':'\
\ :n: 'execute'\ \ :p: '('\ \ :n: 'SMSApplication'\ \ :p: ','\ \ :n: 'debug'\
```

If someone opens `http://mybox.domain.internal:8080/?numero=1234&mensaje=hola%20mundo` it sends “hola mundo” to the 1234 number.

I suppose I could call this a web telephony service or somesuch, but it’s actually just the 5’ solution that came to mind.

It uses a silly little not-a-web-framework called [colubrid](#) instead of something you may know, because I wanted to keep it simple, and it doesn’t get much simpler than this.

Q: What happens if you have a SMTPA server, and you have a user called info, and then set its password to info?

A:

```
[root@victim ~]# qmail-qstat
messages in queue: 151369
messages in queue but not yet preprocessed: 9
```

Lesson:

- Yes, there **is** a reason why those passwords are not available by default.
- I really should limit the per-account outgoing mail by default in all servers. Let’s start working on it.

I am trying to become more organized in my work. And it seems to be working.

Things I do/notice/try now:

- 1) There is a ticketing system. Use it.

Really, there is no reason not to. If the customer says creating tickets is annoying/boring/whatever, I do them myself.

It takes very little time and it lets me see/describe what I do/did in the day. Then, take what's in the ticket, clean it up, and...

- 2) There is a wiki. Use it.

Document every thing I do. No "this is just a little thing". There are no little things. If it was so little, I wouldn't have to do it. Put everything in the docs. The client needs it. I need it.

- 3) There is an email. Use it.

I got an email. Then either it's something that's broken (create a ticket), or something that needs explaining (create/refer a doc), or something that needs doing (create a todo), or a meeting (put it in the phone), or a piece of data I need to remember (archive it, mairix will find it when I need it). I have had an inbox with no mail in it for three days, and it's quite liberating.

- 4) The phone is the first PDA that actually works.

I never managed to use desktop PIM suites or PDAs. I *do* keep my appointments on the phone. I always have my phone with me. I listen to it when I have an alarm. I can't say the same thing about the notebook, or about any PDA I owned. And it's a cheap phone. Using a smartphone makes no sense. I lose the things. Now I can sync it via bluetooth, too! I still use a Sony CLIE to read ebooks, though.

- 5) I like HiveMinder.

It's todo lists I can actually use. Done? Click, gone. New task? Fill a line and it's there, tagged, grouped, dated, prioritized. Want to do it from the CLI? Yup. I do need to write a Trac plugin that converts my tickets into todos, though.

Sure, I don't have it with me all the time (maybe sync it with my phone's todo list is possible somehow...) but I just need to do one full task review a day, add tasks when I'm on a computer, close tasks at the end of the day or while working.

- 6) Read email/news on fixed time periods.

I can't focus on working for over one hour nonstop. So I stop every hour for 10 minutes of news, and 5 minutes of mail (which is still working anyway, since I need to know about stuff). No mail app running in the meantime. If it's urgent, they have my phone number. If they don't, it's not urgent.

It is making a difference, I am becoming more productive without spending more hours at it, which was my previous "strategy".

I will probably write a comments app to replace my usage of haloscan.

Why? Because I can't connect to haloscan from home.

I updated it... the archives work well, and remove the need for the previous hack... haven't found how to add a search form to the blog yet, though.

I am assuming it is being indexed, but how do I use it?

Yesterday I saw this in a news stand and I just had to share it. It's a picture of Cosmopolitan magazine, and it says is:

Test: am I a virgin?

It does pose an interesting problem, though. What's more logical:

- If you have to ask, you are a virgin?
- If you have to ask, you are **not** a virgin?

Since I got my invitation and am tired of HaloScan not being reachable from home (not their fault, probably), I decided that my first project would be a comment hosting app.

In other words, something a bit HaloScan-like.

Since I have very limited resources, it will probably not be useful for many people, but I am learning about App Engine, and at the same time probably making my blog a wee bit more comfortable.

Some random thoughts:

- Can I put Google ads in app engine apps?
- Does anyone else need this kind of app? I intend to make it open, so anyone can register its blog in it and use it. 500MB (the max DB size) are **a lot of comments**. Like a million of them.
- I intend to use Yahoo's YUI RTE for editing. So my app will be hosted in Yahoo **and** Google. Cool :-D
- It's basically just Django. Sure, no UNIQUE, no CRUD (ok, there is Google's, which is kinda lame... hire one of the Django guys and mke him work on it ;-), but it's the same thing, give or take a few bytes, specially using djangoforms.
- webapp is... ok, it's rather ugly. Routing the requests is annoying, you can't do things like passing parts of the URL as parameters...
- The User/DataStore APIs are ok, they *feel* a bit limited but they have a lot of scope in other ways (as in, there are a few million registered users and many TB of data stored ;-)

All things considered, a nice thing to use, specially at the cost.

I had stopped trying to run stuff in WINE a few years ago, because pretty much nothing worked.

You know what? They are right about approaching 1.0, I had to try a few windowsy things lately, and each one has worked just fine.

Kudos wine people!

PS: can you make the widgets look less ugly, though? I heard rumours of a theming engine, but have no clue as to where one would look. Even something like GTK's clearlooks would be bearable.

Since I am the guy that "works with computers" everyone asks me the same things over and over. So here's at least one answer.

Question: What computer should I buy?

Answer:

If you never need a computer outside your home/office, a [eee PC](#) will suck for you, but maybe a [Shuttle KPC](#) will do the trick.

But suppose you are a “Office+Web+Mail+IM+Youtube” guy, as are about 90% of teenage girls and maybe 95% of the over-50.

Then the KPC will work wonderfully for you, if you don’t need to travel with it.

And the eee could work great for you if you travel a lot, or prefer your internet access on a cafe[1], although you could be happier with a light-ish, ordinary 13“ or 14” notebook with a large disk (you know the size of your data, right?)

Do you use it for work on the road, only? Then the eee is awesome. I dare anyone to need over 3GB of data for work, outside of some specific data-intensive niches.

Do you see a pattern here? I never tell you to buy a 4-core 4GB RAM box. You do **not** need it.

If you needed it, you would not have to ask, you would **know** you need it.

- If you are a gamer, you need a gaming rig. But you knew that, so you were not asking me.
- If you edit video, you need what you need. But you knew that too.
- If you build KDE from sources daily, well, you know you need more than one computer anyway ;-)

A Celeron 900 (what’s on the eee, probably the slowest non-VIA CPU sold today) is enough to play any video, except full-HD (which makes no sense, since it has higher resolution than your screen anyway ;-).

A 240GB disk can hold everything you own. Everything you may need on the road probably fits on 8GB, or 40GB if you want a huge music collection.

So, your data can fit in one external disk (buy 2, do backups) , and what you use often **will** fit on any notebook you buy, even a eee PC, modulo a few SD cards.

So, if it’s not about the CPU, it’s not about storage, it’s not about the video board, why would you need a very fast computer?

Buy cheap. Buy last year’s model. Are you willing to spend some extra money? Then spend it on nicer peripherals for your desktop computer, or spend it on a quiet computer, or spend it on a little RAM, but do NOT spend it on CPU.

Here’s my current hardware:

- Main traveling notebook: Asus eee PC 4G. Can’t be happier about it.
- Main desktop computer: A HP Pavillion zd7000. Got it for nothing, used.
- Rosario’s desktop computer: a Intel box with no fans, a 600Mhz P3, got it as surplus from a client for free.

Also: a couple of monitors, a bunch of external IDE disks (with el-cheapo USB-IDE adapters) for backups.

Total cost: probably U\$S 500. You will probably pay U\$S 800 for it, since you probably have less people willing to dump their old computers on you ;-)

If I were to buy all new today, I would probably get two KPC’s along with a 9“ eee.

Expected responses:

- I **need** a faster computer
Then buy one ;-)

- The eee/P3/KPC would be too slow!
For what? Tell me something specific, I'll try it on my eee and let you know how it goes.
- You manage with that hardware because you are a low-end user.
Maybe I am! But maybe you are, too. Let me know why you are high end.

I have used my eee PC for a few months already, and here are a few tips I gathered along the way.

- Remove Xandros, get *Ubuntu on it

system-message

WARNING/2 in `everything.txt`, line 17790
[Inline emphasis start-string without end-string.](#)

Yes, it boots a bit slower, but everything still works (I admit, with some manual tweaks), and you will use less disk space, and be generally happier.

- If you use KDE, check the advanced window menu.
Did you know that any app can be fullscreen? Yes it can. However, I prefer the "No Border" option. Yes, your windows will have no borders, buuuuut:
 - You are probably not going to use windows side by side on a 800x480 screen, are you?
 - You can still move and resize windows: Alt+LMB and Alt+RMB are your friends.
- Yes you can play games on the eee: get psx, the psx emulator. I had a bunch of CDs for my long-dead PS1. Need for Speed looks pretty good, and Tekken 3 is a lot of fun :-D
- There's enough streaming music to avoid carrying your MP3 around. Last.FM and all the shoutcast radios on Amarok do the trick for me.
- Get SD cards.
- The black model seems to be pretty scratch resistant.
- Yes, you can throw it on the sofa when you arrive at home. No, nothing will break (at least nothing broke on mine). SSDs are nice.
- A 1yo boy with clean hands can slap at the keyboard for a minute or two and it may live.

1

system-message

WARNING/2 in `everything.txt`, line 17785
[Duplicate explicit target name: "1".](#)

I get at least one question about my eee everytime I use it on a cafe. Guaranteed.

Suppose a Thing (TM) costs \$100. Now, you are offered a 20% discount. You expect to pay \$80 for it, right?

That means you expect to pay, for X discount, 100-X % of the cost.

That's why this picture is full of FAIL.

BTW: Real picture from a shop in Mar del Plata, Argentina, a few days ago.

Reading a [post](#) on planetkde, I saw an apparently out-of-nowhere reference to blackboxqt...

Not all that interesting, I suppose, but I am fond of ol'BB, and this Qt4 remake could be pretty cool (although I can't test it because it doesn't build for me). Here's [a link to the code](#).

I am a user of [Argus](#) as a monitoring software. Since it's very flexible and easy to extend, I wanted to add MSN alerts, the same way I had added SMS alerts a while ago. It was easier than I thought!

1) Install [msnlib](#)

2) Install the example msnbot, modified so do_work is like this:

```
\ :k: `def` \ :nf: `do_work` \ :p: `():` \
  \ :sd: `""`
  Here you do your stuff and send messages using m.sendmsg()
  This is the only place your code lives
  "" \

\ :c: `# wait a bit for everything to settle down (sync taking effect
\ :c: `# basically)` \
\ :n: `time` \ :o: `.` \ :n: `sleep` \ :p: `( \ :mi: `15` \ :p: `)` \

\ :n: `msg` \ :o: `=` \ :n: `sys` \ :o: `.` \ :n: `stdin` \ :o: `.` \
\ :k: `for` \ :n: `d` \ :ow: `in` \ :n: `sys` \ :o: `.` \ :n: `arg
  \ :k: `print` \ :n: `m` \ :o: `.` \ :n: `sendmsg` \ :p: `( \

\ :c: `# give time to send the messages` \
\ :n: `time` \ :o: `.` \ :n: `sleep` \ :p: `( \ :mi: `30` \ :p: `)` \

\ :c: `# and then quit` \
\ :n: `quit` \ :p: `()` \
```

3) Define a custom notification in argus like this:

```
Method "msn" {
  command: /usr/local/bin/msnbot alerts@mycompany mypass %R >/tmp/XXX
  send: %M\n
}
```

4) Wherever you want MSN notifications, add this (on notify or escalate directives, using as many guys MSN addresses as you need):

```
msn:admin1@hotmail.com+admin2@hotmail.com
```

That's it.

I decided I needed an automatic report of some things on my email every day, and I wanted it to look nice both in plain text and HTML. Here's what I came up with.

Let's assume you created the HTML version using whatever mechanism you wish, and have it in a variable called "report".

Here's the imports we will use:

```
\ :kn:'import' \ \ :nn:'smtplib' \ \ :o:', '\ \ :nn:'email' \ \ :o:', '\ \ :nn:
\ :kn:'from' \ \ :nn:'email.MIMEText' \ \ :kn:'import' \ \ :n:'MIMEText' \
\ :kn:'from' \ \ :nn:'email.MIMEMultipart' \ \ :kn:'import' \ \ :n:'MIMEMul
\ :kn:'from' \ \ :nn:'email.Charset' \ \ :kn:'import' \ \ :n:'Charset' \
```

And here's the code:

```
\ :c:'# Create a HTML mail part' \
\ :n:'hpart' \ \ :o:'=' \ \ :n:'MIMEText' \ \ :p:'('\ \ :n:'reporte' \ \ :p:',

\ :c:'# Create a plain text mail part' \
\ :c:'# Ugly and requires links, but makes for a great-looking plain text v
\ :n:'tf' \ \ :o:'=' \ \ :n:'tempfile' \ \ :o:'. '\ \ :n:'mkstemp' \ \ :p:'()' \
\ :n:'t' \ \ :o:'=' \ \ :nb:'open' \ \ :p:'('\ \ :n:'tf' \ \ :p:', '\ \ :s:'w'
\ :n:'t' \ \ :o:'. '\ \ :n:'write' \ \ :p:'('\ \ :n:'report' \ \ :p:') \
\ :n:'t' \ \ :o:'. '\ \ :n:'close' \ \ :p:'()' \
\ :n:'tpart' \ \ :o:'=' \ \ :n:'MIMEText' \ \ :p:'('\ \ :n:'os' \ \ :o:'. '\ \ :
\ :n:'os' \ \ :o:'. '\ \ :n:'unlink' \ \ :p:'('\ \ :n:'tf' \ \ :p:') \

\ :c:'# Create the message with both parts attached' \
\ :n:'msg' \ \ :o:'=' \ \ :n:'MIMEMultipart' \ \ :p:'('\ \ :s:'alternative' \
\ :n:'msg' \ \ :o:'. '\ \ :n:'attach' \ \ :p:'('\ \ :n:'hpart' \ \ :p:') \
\ :n:'msg' \ \ :o:'. '\ \ :n:'attach' \ \ :p:'('\ \ :n:'tpart' \ \ :p:') \

\ :c:'# Standard headers (add all you need, for example, date) \
\ :n:'msg' \ \ :p:'[\ \ \ :s:'Subject' \ \ :p:'] \ \ :o:'=' \ \ :s:'Report
\ :n:'msg' \ \ :p:'[\ \ \ :s:'From' \ \ :p:'] \ \ :o:'=' \ \ :s:'support
\ :n:'msg' \ \ :p:'[\ \ \ :s:'To' \ \ :p:'] \ \ :o:'=' \ \ :s:'you@yo

\ :c:'#If you need to use SMTP authentication, change accordingly' \
\ :n:'smtp' \ \ :o:'=' \ \ :n:'smtplib' \ \ :o:'. '\ \ :n:'SMTP' \ \ :p:'('\ \ :
\ :n:'smtp' \ \ :o:'. '\ \ :n:'sendmail' \ \ :p:'('\ \ :s:'support@yourcompar
```

This blog was hosted for a long time in <http://lateral.pycs.net> ... until the site kinda died. And there are hundreds of links around, all pointing to URLs in pycs.net for the older stories. Now they all work again!

So, thanks Philip!

.... there **is** a search form in the blog already. It even works!

I have been using Bruce Guenter's nice Qmail rpms for years, but he doesn't seem to be working on them much.

So, I just fixed a couple of issues:

- TLS Support
- errno fix (for glibc 2.3)

- svc-add smtpd (silly)

So, it now builds out-of-the-box on RH8 and RH9.

I should open a sf.net site for them, probably. Add CRAM-MD5 authentication (I have it working with a plain file, LDAP would be nicer), a working courier RPM (tricky to build sometimes), a little extra stuff (antivirus, spamassassin, autoresponder) and it's really a nice mail server in a box.

I started this as an experiment to see how hard it was to build apps using QT and Elixir. This is how it looks after two hours of coding:

And here's the code: <http://urssus.googlecode.com>

You will need:

- PyQt 4.4 or later
- Mark Pilgrim's feedparser
- Python 2.5 (or whenever elementtree got included)
- A OPML file
- The Python SQLite bindings
- Elixir (the declarative layer over SQL Alchemy)

You can find the real code at

And then you can use these 131 LOC ;-)

```

\ :c:`# -*- coding: utf-8 -*-`

\ :c:`# Mark Pilgrim's feed parser`
\ :kn:`import` \ \ :nn:`feedparser` \ \ :kn:`as` \ \ :nn:`fp`

\ :c:`# DB Classes`
\ :kn:`from` \ \ :nn:`elixir` \ \ :kn:`import` \ \ :o:`*`

\ :n:`metadata` \ \ :o:`.` \ \ :n:`bind` \ \ :o:`=` \ \ :s:`"sqlite:///urssus`
\ :n:`metadata` \ \ :o:`.` \ \ :n:`bind` \ \ :o:`.` \ \ :n:`echo` \ \ :o:`=`

\ :k:`class` \ \ :nc:`Feed` \ \ :p:`(` \ \ :n:`Entity` \ \ :p:`):`
\ :n:`htmlUrl` \ \ :o:`=` \ \ :n:`Field` \ \ :p:`(` \ \ :n:`Text` \ \ :p:`)`
\ :n:`xmlUrl` \ \ :o:`=` \ \ :n:`Field` \ \ :p:`(` \ \ :n:`Text` \ \ :p:`)`
\ :n:`title` \ \ :o:`=` \ \ :n:`Field` \ \ :p:`(` \ \ :n:`Text` \ \ :p:`)`
\ :n:`text` \ \ :o:`=` \ \ :n:`Field` \ \ :p:`(` \ \ :n:`Text` \ \ :p:`)`
\ :n:`description` \ \ :o:`=` \ \ :n:`Field` \ \ :p:`(` \ \ :n:`Text` \ \ :p:`)`
\ :n:`children` \ \ :o:`=` \ \ :n:`OneToMany` \ \ :p:`(` \ \ :s:`"Feed`
\ :n:`parent` \ \ :o:`=` \ \ :n:`ManyToOne` \ \ :p:`(` \ \ :s:`"Feed`
\ :n:`posts` \ \ :o:`=` \ \ :n:`OneToMany` \ \ :p:`(` \ \ :s:`"Post`
\ :k:`def` \ \ :nf:`__repr__` \ \ :p:`(` \ \ :bp:`self` \ \ :p:`):`
\ :k:`return` \ \ :bp:`self` \ \ :o:`.` \ \ :n:`text`

\ :k:`def` \ \ :nf:`update` \ \ :p:`(` \ \ :bp:`self` \ \ :p:`):`
\ :n:`d` \ \ :o:`=` \ \ :n:`fp` \ \ :o:`.` \ \ :n:`parse` \ \ :p:`(` \ \ :bp:`

```

```

\ :k:'class'\ \ :nc:'Post'\ \ :p:'('\ \ :n:'Entity'\ \ :p:'):\
\ :n:'feed'\ \ :o:'='\ \ :n:'ManyToOne'\ \ :p:'('\ \ :s:''Feed'\
\ :n:'title'\ \ :o:'='\ \ :n:'Field'\ \ :p:'('\ \ :n:'Text'\ \ :p
\ :n:'post_id'\ \ :o:'='\ \ :n:'Field'\ \ :p:'('\ \ :n:'Text'\ \ :p
\ :n:'content'\ \ :o:'='\ \ :n:'Field'\ \ :p:'('\ \ :n:'Text'\ \ :p

\ :c:'# This is just temporary'\
\ :n:'setup_all'\ \ :p:'()'\
\ :n:'create_all'\ \ :p:'()'\

\ :c:'# UI Classes'\
\ :kn:'from'\ \ :nn:'PyQt4'\ \ :kn:'import'\ \ :n:'QtGui'\ \ :p:','\ \
\ :kn:'from'\ \ :nn:'Ui_main'\ \ :kn:'import'\ \ :n:'Ui_MainWindow'\

\ :k:'class'\ \ :nc:'MainWindow'\ \ :p:'('\ \ :n:'QtGui'\ \ :o:'.'\ \ :n:
\ :k:'def'\ \ :nf:'__init__'\ \ :p:'('\ \ :bp:'self'\ \ :p:'):\
\ :n:'QtGui'\ \ :o:'.'\ \ :n:'QMainWindow'\ \ :o:'.'\ \ :n:'__init__'\

\ :c:'# Set up the UI from designer'\
\ :bp:'self'\ \ :o:'.'\ \ :n:'ui'\ \ :o:'='\ \ :n:'Ui_MainWindow'\ \ :p:
\ :bp:'self'\ \ :o:'.'\ \ :n:'ui'\ \ :o:'.'\ \ :n:'setupUi'\ \ :p:'('\

\ :c:'# Initialize the tree from the Feeds'\
\ :bp:'self'\ \ :o:'.'\ \ :n:'model'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'.'\

\ :c:'# Internal function'\
\ :k:'def'\ \ :nf:'addSubTree'\ \ :p:'('\ \ :n:'parent'\ \ :p:','\ \
\ :n:'nn'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'.'\ \ :n:'QStandardItem'\ \
\ :n:'parent'\ \ :o:'.'\ \ :n:'appendRow'\ \ :p:'('\ \ :n:'nn'\ \ :p:
\ :n:'nn'\ \ :o:'.'\ \ :n:'feed'\ \ :o:'='\ \ :n:'node'\
\ :k:'if'\ \ :ow:'not'\ \ :n:'node'\ \ :o:'.'\ \ :n:'children'\ \ :
\ :k:'return'\
\ :k:'else'\ \ :p:':'\
\ :k:'for'\ \ :n:'child'\ \ :ow:'in'\ \ :n:'node'\ \ :o:'.'\ \ :
\ :n:'addSubTree'\ \ :p:'('\ \ :n:'nn'\ \ :p:','\ \ :n:'child'\

\ :n:'roots'\ \ :o:'='\ \ :n:'Feed'\ \ :o:'.'\ \ :n:'query'\ \ :o:'.'\
\ :n:'iroot'\ \ :o:'='\ \ :bp:'self'\ \ :o:'.'\ \ :n:'model'\ \ :o:'.'\
\ :n:'iroot'\ \ :o:'.'\ \ :n:'feed'\ \ :o:'='\ \ :bp:'None'\
\ :k:'for'\ \ :n:'root'\ \ :ow:'in'\ \ :n:'roots'\ \ :p:':'\
\ :n:'addSubTree'\ \ :p:'('\ \ :n:'iroot'\ \ :p:','\ \ :n:'root'\ \

\ :bp:'self'\ \ :o:'.'\ \ :n:'ui'\ \ :o:'.'\ \ :n:'feeds'\ \ :o:'.'\ \

\ :n:'QtCore'\ \ :o:'.'\ \ :n:'QObject'\ \ :o:'.'\ \ :n:'connect'\ \ :p:
\ :n:'QtCore'\ \ :o:'.'\ \ :n:'QObject'\ \ :o:'.'\ \ :n:'connect'\ \ :p:

\ :k:'def'\ \ :nf:'openFeed'\ \ :p:'('\ \ :bp:'self'\ \ :p:','\ \ :n:'f
\ :n:'item'\ \ :o:'='\ \ :bp:'self'\ \ :o:'.'\ \ :n:'model'\ \ :o:'.'\
\ :n:'feed'\ \ :o:'='\ \ :n:'item'\ \ :o:'.'\ \ :n:'feed'\

```

```

\ :k:'if'\ \ :ow:'not'\ \ :n:'feed'\ \ :o:'. '\ \ :n:'xmlUrl'\ \ :p:'\
\ :k:'return'\

\ :n:'d'\ \ :o:'='\ \ :n:'fp'\ \ :o:'. '\ \ :n:'parse'\ \ :p:'('\ \ :n:
\ :n:'posts'\ \ :o:'='\ \ :p:'[]'\
\ :k:'for'\ \ :n:'post'\ \ :ow:'in'\ \ :n:'d'\ \ :p:'[\ \ \ :s:'entr
\ :k:'print'\ \ :n:'post'\
\ :k:'print'\ \ :s:'-----'\ \ :se:'\n\
\ :k:'if'\ \ :s:'content'\ \ :ow:'in'\ \ :n:'post'\ \ :p:': '\
\ :n:'posts'\ \ :o:'. '\ \ :n:'append'\ \ :p:'('\ \ :n:'Post'\ \ :p:
\ :k:'elif'\ \ :s:'summary'\ \ :ow:'in'\ \ :n:'post'\ \ :p:': '\
\ :n:'posts'\ \ :o:'. '\ \ :n:'append'\ \ :p:'('\ \ :n:'Post'\ \ :p:
\ :k:'elif'\ \ :s:'value'\ \ :ow:'in'\ \ :n:'post'\ \ :p:': '\
\ :n:'posts'\ \ :o:'. '\ \ :n:'append'\ \ :p:'('\ \ :n:'Post'\ \ :p:
\ :n:'session'\ \ :o:'. '\ \ :n:'flush'\ \ :p:'() '\

\ :bp:'self'\ \ :o:'. '\ \ :n:'ui'\ \ :o:'. '\ \ :n:'posts'\ \ :o:'. '\ \
\ :k:'for'\ \ :n:'post'\ \ :ow:'in'\ \ :n:'posts'\ \ :p:': '\
\ :n:'item'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QStandardItem'\
\ :n:'item'\ \ :o:'. '\ \ :n:'post'\ \ :o:'='\ \ :n:'post'\
\ :bp:'self'\ \ :o:'. '\ \ :n:'ui'\ \ :o:'. '\ \ :n:'posts'\ \ :o:'. '\
\ :bp:'self'\ \ :o:'. '\ \ :n:'ui'\ \ :o:'. '\ \ :n:'posts'\ \ :o:'. '\ \

\ :k:'def'\ \ :nf:'openPost'\ \ :p:'('\ \ :bp:'self'\ \ :p:', '\ \ :n:'f
\ :n:'item'\ \ :o:'='\ \ :bp:'self'\ \ :o:'. '\ \ :n:'ui'\ \ :o:'. '\ \ :
\ :n:'post'\ \ :o:'='\ \ :n:'item'\ \ :o:'. '\ \ :n:'post'\
\ :bp:'self'\ \ :o:'. '\ \ :n:'ui'\ \ :o:'. '\ \ :n:'view'\ \ :o:'. '\ \

\ :k:'if'\ \ :n:'__name__'\ \ :o:'=='\ \ :s:'"__main__"\ \ :p:': '\
\ :kn:'import'\ \ :nn:'sys'\
\ :c:'# For starters, lets import a OPML file into the DB so we have some
\ :kn:'from'\ \ :nn:'xml.etree'\ \ :kn:'import'\ \ :n:'ElementTree'\
\ :n:'tree'\ \ :o:'='\ \ :n:'ElementTree'\ \ :o:'. '\ \ :n:'parse'\ \ :p:
\ :n:'current'\ \ :o:'='\ \ :bp:'None'\
\ :k:'for'\ \ :n:'outline'\ \ :ow:'in'\ \ :n:'tree'\ \ :o:'. '\ \ :n:'f
\ :n:'xu'\ \ :o:'='\ \ :n:'outline'\ \ :o:'. '\ \ :n:'get'\ \ :p:'('\ \
\ :k:'if'\ \ :n:'xu'\ \ :p:': '\
\ :n:'f'\ \ :o:'='\ \ :n:'Feed'\ \ :p:'('\ \ :n:'xmlUrl'\ \ :o:'='\ \
\ :n:'htmlUrl'\ \ :o:'='\ \ :n:'outline'\ \ :o:'. '\ \ :n:'get
\ :n:'title'\ \ :o:'='\ \ :n:'outline'\ \ :o:'. '\ \ :n:'get'\
\ :n:'text'\ \ :o:'='\ \ :n:'outline'\ \ :o:'. '\ \ :n:'get'\
\ :n:'description'\ \ :o:'='\ \ :n:'outline'\ \ :o:'. '\ \ :n:
\ :p:') '\
\ :k:'if'\ \ :n:'current'\ \ :p:': '\
\ :n:'current'\ \ :o:'. '\ \ :n:'children'\ \ :o:'. '\ \ :n:'append'\
\ :n:'f'\ \ :o:'. '\ \ :n:'parent'\ \ :o:'='\ \ :n:'current'\
\ :k:'else'\ \ :p:': '\
\ :n:'current'\ \ :o:'='\ \ :n:'Feed'\ \ :p:'('\ \ :n:'text'\ \ :o:'=
\ :n:'session'\ \ :o:'. '\ \ :n:'flush'\ \ :p:'() '\

```

```

\ :n: 'app' \ \ :o: '=' \ \ :n: 'QtGui' \ \ :o: '.' \ \ :n: 'QApplication' \ \ :p:
\ :n: 'window' \ \ :o: '=' \ \ :n: 'MainWindow' \ \ :p: '()' \ \
\ :n: 'window' \ \ :o: '.' \ \ :n: 'show' \ \ :p: '()' \ \
\ :n: 'sys' \ \ :o: '.' \ \ :n: 'exit' \ \ :p: '()' \ \ :n: 'app' \ \ :o: '.' \ \ :n:

```

When you upgrade a piece of software on Linux, there are two paths it can go when there are incompatible changes in the config files (ok, 3 paths, Debian asks you what to do):

- 1) The "rpmnew" way: install the new config file as "whatever.rpmnew", which means the software will break immediately, but that's ok, because you are doing upgrades, so you should be watching already.
- 2) The "rpmsave" way: replace the old file and save a copy as "whatever.rpmsave".

This has two problems:

- a) The software may fail or not, or fail in a subtle way, and you will not notice right away.
- b) Maybe the old file will be lost anyway:

```

lrwxrwxrwx 1 root root 32 jul 15 22:41 /etc/named.conf -> /var/named/c
lrwxrwxrwx 1 root root 32 jul 15 22:36 /etc/named.conf.rpmsave -> /var

```

In this case the "file" was a symlink, so by "saving a copy" it only saved **another** symlink to the soon-to-be-overwritten file.

And that's why, ladies and gentlemen, the rpmnew way is the good way.

I worked on [uRSSus](#) for a couple of hours again, and it's working pretty nicely.

- Really uses the ORM
- Multiprocessing for a non-blocking UI (python-processing is awesome)
- Adapts to the quirks of some feeds (why on earth would someone do a feed without dates? It's AOL FanHouse!)

I intend to keep working like this for a couple of weeks, and see how far I can get in feature parity to akregator.

No, I don't expect to *reach* feature parity, I only want to strive for it. Since I lack the focus and/or energy for a multi year commitment it requires to write the average free software, I want to see how far a sprint gets me.

So far, it's fun.

Today's 2 hours:

- An about dialog (not wired to the UI yet)
- Icons on the feed tree (not favicons, though)
- Handle a few more feed quirks
- A status message queue, so subprocesses can post their progress in the status bar.
- UI for import OPML

- Implemented "Next Article" and "Next Feed". This Model/View Qt thing is kinda painful (but powerful!)
- Use of Mako templates to display the articles pretty and neat
- Menu & Shortcuts following Akregator
- Added several fields to the Post and Feed models to make them more useful (which means the DB changed, but that's to be expected at this stage). These include things like "unread" and "author" and "link" ;-)

Still fun!

Another day, another two hours of work on it.

What's new?

- Some UI elements without code behind them (the "filter article" thingie), and a feed properties dialog.
- A (IMVHO) better UI distribution than akregator. Consider this screenshot:

The article list and feed tree are the same size, but removing the tabs and moving the filter into a toolbar really makes the actual reading area quite larger.

- Delegate all links from the reading area to the desktop's web browser (this is not a web browser yet ;-)
- Progress report for web page loading (more useful once you can declare a feed as "load link directly")
- Finished feed/article navigation (next/previous feed/article unread/any), and fixed bugs in the part that was already done.
- Zoom in/out for the web view
- Show/hide statusbar
- Several bug fixes

And yes, still fun!

I confess I cheated and kept working on it yesterday after the blog post. OTOH, I will not touch it today ;-). Big functionality added, too.

- Feed items (The things on the tree) now are updated when the background process checks them.
- They are also updated when you read articles from a feed.
- The filter thingie works, you type some text, and only the articles with that text are shown (see screenshot).
- Added a widget (not dialog) for searching within the page, firefox-like (see screenshot)

The bad news is that the "next unread article" code and a few others is garbage. It's quite inefficient because I tried to be cheap and not create a coherent model for feeds.

However it works and you will never tell the difference unless you have 2000 articles between where you are and the next unread (in which case the window goes kinda nuts for a couple of seconds).

Still fun!

Yes, I am now using uRSSus instead of Akregator.

On the programming side, I did very little work because it's sunday...

- Implemented the Find widget backend code and it works forward and backward (but not "Match case" which is pretty trivial).
- Fixed a few focus issues. When you have a multi-widget main window, it can get kinda tricky, and you may end needing to move focus around manually (not in this case yet).

I have found a few bugs, mainly related to traversing the feed tree, but they are all fixable with some effort (or by doing it correctly the second time ;-)

Today it was a bit more than two hours, but spread in 10 minute chunks, which is not exactly efficient.

Improvements:

- Rewrote the whole next/next unread/article/feed spaghetti in a decent way. It's broken regarding filtered/sorted article lists, but it's trivial to fix.
- A few UI features:
 - Show only feeds with unread articles google-reader style
 - Show in dark red the unread articles
 - Show title / date of articles in separate columns
- Implemented "Match case" for the search
- Implemented dumb systray icon (need to make it useful)
- Implemented article counts in folders (but not article lists which is, of course, harder)

And of course, a screenshot:

A bad side is that the main window takes about 3 seconds to appear, but I am pretty sure that's fixable reordering the startup code.

TODO:

- Feed CRUD
- Post mark as (read/unread/important/whatever)
- Filter by post status
- Search in all feeds
- Virtual feeds
- Persistent settings

- More aggressive feed fetching (now it's a single queue, should be a configurable number)
- Lots of minor fixes

And it will be pretty much in feature parity with Akregator, and ready to start the packaging work. Not bad for a week of work (BTW: if anyone has any experience packaging PyQt stuff for windows/Mac I can use the help ;-).

Still fun (although tree traversing is starting to get quite annoying)

So I did work on it a little last night:

- Feed/Folder context menus
- Open Feed Homepage action
- New Feed (uses Mark Pilgrim's feedfinder.py, which is awesome)
- New Folder
- Delete Feed/Folder
- Some UI fixes (finally figured out how to make a QTreeView's headers look right!)
- Some bugfixes in tree traversing (where else ;-)
- Implemented "fetch the web page instead of the feed contents" option for feeds (because of your annoying feed, DZone!)

Sadly no visible changes, so a screenshot would be useless.

When I started this project I wrote:

I intend to keep working like this for a couple of weeks, and see how far I can get in feature parity to akregator.

No, I don't expect to reach feature parity, I only want to strive for it. Since I lack the focus and/or energy for a multi year commitment it requires to write the average free software, I want to see how far a sprint gets me.

To my surprise, only 9 days into the project, feature parity is at hand.

What's not done?

- Some feed properties (custom archiving modes, mark as read on arrival) (Update: "mark as read on arrival" implemented at 22:40)
- Fix article filtering (it's not quite broken, but interacts badly with many things) (Update: done as of 19:20)
- Implement app preferences dialog
- Filtering by article status (Update: implemented as of 19:01)
- Folder-as-feed (Update: done, as of 22:20, even if not really efficiently)
- Export as OPML (Update: just implemented it :-)
- Favicons

And some things in the backend, which are not proper features.
What have I done today?

- New articles notification via systray
- Custom update intervals per-feed
- Made Feed.unreadCount() fast enough
- Article sorting (not as trivial as it sounds)
- Fixed a lot of bugs (none of which crashed uRSSus, take that Akregator! ;-)

And still I have written only 1066 lines of code, according to sloccount.

Update: 1127 lines with the extra features marked above as "update"

I just spent the afternoon at a client's hacking qmail+courier to work storing mail on a NT server.

Why? Well, migration issues, from NT to Linux.

And what had to be hacked? Lots.

- You can't chmod over SMB
- You can't link() over SMB (you can unlink(), though), so in order to have atomic message moves from Maildir/tmp to Maildir/new and whatever, rename()
- Good thing this was a single-UID qmail system: you can't chown either ;-)
- Courier's metadata is created on Maildir/tmp and moved: same problem as above
- Cranky permission checks fixed via uid,gid,fmask mount options.
- It had to work with a canned webmail thing, which expects all mail to be in **one** folder on disk, so Maildir/cur and Maildir/new have to be symlinks to a single folder.
- Since you can't have symlinks on SMB, Maildir is local, but subdirs of it are symlinks to the SMB-mounted thing
- Then courier's pop doesn't work because you can't rename from Maildir/tmp (over SMB) to maildir/courierpopwhatever (local), so hack courier to use mkstemp or whatever
- Then courier got each mail twice. Looks like it sees two UIDLs for the same file, depending on whether it's read or not (my guess: because we are mixing Maildir/cur and Maildir/new on one folder). Make courier use alternative UIDL format and it works.

And so on. About two hours, and I am pretty sure noone has ever done something like this before. On the other hand, when noone has done something like this before, they usually had a very good reason!

My stated goal of Akregator feature parity is looking even better. With screenshot!

If you didn't read (or care about) last night's updates to yesterday's post, here they are:

- Implemented "Mark posts as read on arrival for this feed"

- Implemented customupdate intervals for feeds
- Fixed article filtering
- Implemented filtering by article status
- Implemented folder-as-feed (not perfect)
- Export as OPML

And today I did these:

- Switched from Mako to Tenjin (which I can just bundle)
- Implemented "Combined View" mode (you can see it in the screenshot above)
- Implemented "Import Technorati top 10"
- Organized the code correctly
- Wrote a Paver script so it can be installed nicely (someday)
- Removed all generated files from the repo (now you generate them using paver and/or setup.py)

So, on the TODO we have:

- Favicons
- Implement app preferences dialog
- Bugfixes
- Pagination for combinedView (so you don't have a page with 90000 posts in it which takes a month to load)
- Akregator DB import script, so I can compare performances reasonably.

And still fun!

Current LOC count: 1295

So you need to install SQLAlchemy 0.4.6 and Elixir 0.5.2 and python and pyqt and paver, and whatever manually right now... *it does* work.

I had to change about 10 lines, specifically stuff about temporary files.

On Linux, I create a NamedTemporaryFile, write to it, then read it from another place of the code. As long as I keep a handle to the object that's ok.

On Windows, it seems I can't read the file because it's locked by the writer, but if I close it it's deleted, so I switched to mkstemp and that's it.

Also, it doesn't run well from inside IDLE for some reason.

Last post for the day, and probably until monday.

- Finished expiration implementation (and it works just fine)
- Fixed a tough bug and now the background fetcher updates the UI correctly
- Made the template location code work correctly before and after installation (thanks to help from PyAr mailing list)

- And the pretty one: custom CSS file used for all templates. The default webkit (and khtml, and firefox) CSS is horrid. This one is just not pretty, but still an improvement.

And of course, a screenshot. Akregator on the left, uRSSSus on the right:

I spent the day bugfixing. Then, when I realized I was fixing stull like "when you delete a feed, the unread count in the parent feed is not updated correctly" and "if you go to *Next Unread article* and there are none, make the search wrap" I decided this thing must really be working right, since both things failed me in at least one similar program that claims to be production-ready (not akregator! ;-). So I am releasing it as 0.1.0

And I am doing it for real:

- [PyPi page](#) for easy_install
- Working installer! (Not quite working for windows, though)
- Most known bugs fixed (at least 15 today)
- Uploaded to [Arch's AUR](#), so we Arch users can get it painlessly ;-)

It has been a nice experiment, and I will try to write something coherent tomorrow about lessons learned and such as we reach my self-imposed two week deadline for Akregator feature parity.

Sure, new search engine Cuil serches "121,617,892,992 web pages". I believe it... maybe.

However, I tried a simple test. My name. I know more or less where it appears:

- My blog
- My linkedin pages
- Several FLOSS hosting sites
- A bazillion mailing list archives

On google this search gives the expected results:

On Cuil? Not so much:

In other words, I managed to be skipped by Cuil completely. All traces of my last 12 years on the Internet? Not seen. That kinda makes me not too confident ;-)

I did a fair amount of work in urssus today. The highlight? database schema migration.

Here is the (not really) full list:

- Fixes on the installer
- Fix in importOPML
- Raise the window when you click on the systray popup
- Fixed several issues after feed deletion
- Fixed several issues with feeds not updating the UI's unread count
- Fixed the look of buttons on searchWidget and filterWidget

- And several more...

And the big one:

- Implemented database schema versioning using sqlalchemy-migrate.

What does that mean? That I can change the database and the user will not notice anything.

On startup, uRSSus checks if you are using the correct schema, and updates your database accordingly.

For example, I can now implement manual feed sorting just adding a "position" column to the Feed class. If I had not implemented this, that would require a much greater hack.

Now? I just do it, write an upgrade/downgrade script, and that's all. Migrate is really a very cool tool, and shows the strength of using SQLAlchemy (not so much that of using Elixir, because they get along grudgingly ;-)

Current LOC count: 1491

Current status: still fun! (and educational!)

Yes, I released 0.2.0, and announced it on PyPI, Freshmeat and kde-apps.org, we'll see what happens.

But why is it outdated? Because I am doing nicer UI work, already.

- The search widget is now a bottom-locked toolbar, which is much cleaner.
- There are now 3 window layouts (normal, widescreen and combined) and each has a "long feed list" and a "short feed list" variant.

Here is widescreen-long:

And here are widescreen-short, widescreen-long, normal-short, and combined-short, all scrunched in a single screen (yes, you can run multiple urssus (urssuses? urssi?):

Tired of chasing after other program's features, I took a detour on the original feature road: the combined view is now much nicer than Akregator's.

Why? Because you can use it and still flag individual articles as read/unread/important.

Using a tiny bit of mootools and some rudimentary javascript, plus a wee Tenjin template effort...

And yes, the "buttons" appear and disappear as needed, and update the GUI just like they should.

I may have gotten a bit carried away with mootools: Combined view now has an accordion-like thing google-reader style.

Not too pretty, but pretty's not my thing.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "this guy".

So, I needed a python script to create tinyurls. Google gets me [here](#) where credit is given to [this guy](#). Of course *that guy's* solution is a bit broken, check the comments ;-)

system-message

WARNING/2 in `everything.txt`, line 18608

Duplicate explicit target name: "here".

system-message

WARNING/2 in `everything.txt`, line 18610

Duplicate explicit target name: "this guy".

On this morning's newspaper. So I am ahead of the curve by about 48 hours. That's not much :-)

Interesting: it seems Windows was mandatory in official usage. Weird, since I have seen hundreds of Linux servers on official usage :-)

I have a twitter account. I never found a use for it.

Now I have one. I will post there whatever interesting thing I find. Just right-click, "Post to Twitter" and voilà:

Here you can see the window, which is filled automatically with the post title and URL (made tiny, 140 chars is not much!), with the classic char countdown.

Also in the background, some news and my twitter account.

BTW: Uwe was awesome.

At least three people have tried urssus (including me ;-) and over 80 have downloaded it. The two that told me anything say it's a nice app.

I hope it keeps improving.

Right now, it has about a dozen less bugs (or more features) thanks to the [issue reports](#) by vgarvardt!

Helped by the testing by vgarvardt, many issues and possible features have been fixed or implemented in the last two days. Here's an incomplete list:

- Implemented (and later fixed) proper close-to-tray
- Marking a folder as read marks all its children
- The reset filter button works like Akregator's
- The tray changes its icon when there are unread items (and shows a count in the tooltip). Icon work needed, though.
- Double click on a post opens in external browser
- Default refresh period is configurable (no UI yet)
- The feed tooltip shows unread count
- Window size and position is memorized
- Position of the splitters is memorized
- Toolbar titles are now reasonable (never tried to right-click a Qt toolbar before)
- Better templates for feed and post layouts
- Unread items/feeds are now shown bold
- The about window has the right title and icon, and doesn't appear in the taskbar.

There is almost certainly going to be a release tomorrow.

And here's a picture:

I had planned to release a new version today, but there are too many untested features. A list:

- Feed reordering via drag and drop
- Unique app. The second uRSSus just make the first one operate.
- Lots of refactoring to cleanup View/Model related things.

Since it's a lot of code, it will need a day or two of testing. Or a week.

Mostly, I spent my two hours today refactoring. But also, testing how uRSSus looks on a small screen. Take a look:

This is urssus in the configuration I liked best on my 7", 800x480 eeePC's screen.

As you can see, a more compact template for the combined view is a must, and I will implement a compact template. Or maybe just make a configuration option where you choose what template the combined view uses, to avoid the multiplication of "modes" that are the exact same thing under the hood.

Using combined view with all bars hidden and fullscreen makes for a rather large reading area, which is nice.

The performance in that limited computer was good... until the memory usage balloons. While uRSSus is not *too* CPU hungry, it's pretty memory intensive. There are obvious ways to improve that, of course, and I am exploring them.

BTW, I have not measured this in a while: 1957 lines of code. And still quite fun!

It was rather easy because I didn't do the hard part (thanks authors of pyrfeed!) and now uRSSus has a simple "import my subscriptions from Google Reader" action.

All I can say is "it works for me" and "Google sure strips data from the feeds!".

It will import your tags as top-level folders (merged with whatever you already had).

Should be smart enough not to duplicate feeds or folders.

If a feed is in more than one folder, it will show in the first listed by google (no idea what the order is).

Version 0.2.9 of [uRSSus](#), my news agregator is out. Many new features including:

- Different layouts (regular/widescreen/combined, aka river-of-news)
- Twitter about things you see on [uRSSus](#) with a single click
- Many many bugfixes
- Lots of UI improvements

And so on.

Truly a paperbag release. But there's still hope!

While 0.2.9 had a number of bugs and awful performance, I have to say that the current svn revisions are much, much, better.

- Snappy performance (ok, not quite snappy, but snappier than 0.2.0)
- Finally a coherent nex/previous/whatever mechanism. The only broken piece is that posts may vanish from view when the feed is updated (and I can fix that, too).

- Fixed issues 16,14 and 22

So, 0.2.10, coming tonight, should be a pretty good one.

I felt that a “user-friendly” configuration dialog for uRSSSus was not a good way to spend effort, because I expect options to change often for a while, and I didn’t want an outdated dialog, either. The solution? Autogenerate a mediocre configuration dialog!

First, define what options uRSSSus supports (Update: suggestion by Alejandro Cura, using tuples works much better!):

```
\ :n:'options' \ \ :o:'=' \ \ :p:'('\
 \ :p:'('\ \ :s:'ui' \ \ :p:', '\
 \ :p:'('\
 \ :p:'('\ \ :s:'alwaysShowFeed' \ \ :p:', '\ \ :p:'('\ \ :s:'bool'
 \ :p:'('\ \ :s:'hideOnTrayClick' \ \ :p:', '\ \ :p:'('\ \ :s:'bool'
 \ :p:') \
 \ :p:'), \
 \ :p:'('\ \ :s:'options' \ \ :p:', '\
 \ :p:'('\
 \ :p:'('\ \ :s:'defaultRefresh' \ \ :p:', '\ \ :p:'('\ \ :s:
 \ :p:'('\ \ :s:'maxPostsDisplayed' \ \ :p:', '\ \ :p:'('\ \ :s:
 \ :p:'('\ \ :s:'defaultExpiration' \ \ :p:', '\ \ :p:'('\ \ :s:
 \ :p:') \
 \ :p:'), \
 \ :p:'('\ \ :s:'twitter' \ \ :p:', '\
 \ :p:'('\
 \ :p:'('\ \ :s:'username' \ \ :p:', '\ \ :p:'('\ \ :s:'string' \ \
 \ :p:'('\ \ :s:'password' \ \ :p:', '\ \ :p:'('\ \ :s:'password' \
 \ :p:') \
 \ :p:') \
 \ :p:') \
```

Then a little magic, and this comes out:

How magic? This magic:

```
\ :k:'class' \ \ :nc:'ConfigDialog' \ \ :p:'('\ \ :n:'QtGui' \ \ :o:'. '\ \ :n:
 \ :k:'def' \ \ :nf:'__init__' \ \ :p:'('\ \ :bp:'self' \ \ :p:', '\ \ :n:'p
 \ :n:'QtGui' \ \ :o:'. '\ \ :n:'QDialog' \ \ :o:'. '\ \ :n:'__init__' \ \ :p:
 \ :c:'# Set up the UI from designer' \
 \ :bp:'self' \ \ :o:'. '\ \ :n:'ui' \ \ :o:'=' \ \ :n:'UI_ConfigDialog' \ \
 \ :bp:'self' \ \ :o:'. '\ \ :n:'ui' \ \ :o:'. '\ \ :n:'setupUi' \ \ :p:'('\
 \ :n:'pages' \ \ :o:'=' \ \ :p:'[]' \
 \ :n:'sections' \ \ :o:'=' \ \ :p:'[]' \
 \ :bp:'self' \ \ :o:'. '\ \ :n:'values' \ \ :o:'=' \ \ :p:'{' \

 \ :k:'for' \ \ :n:'sectionName' \ \ :p:', '\ \ :n:'options' \ \ :ow:'in
 \ :c:'# Create a page widget/layout for this section:' \
 \ :n:'page' \ \ :o:'=' \ \ :n:'QtGui' \ \ :o:'. '\ \ :n:'QScrollArea' \ \
 \ :n:'layout' \ \ :o:'=' \ \ :n:'QtGui' \ \ :o:'. '\ \ :n:'QGridLayout' \ \
 \ :n:'row' \ \ :o:'=' \ \ :mi:'2' \
 \ :k:'for' \ \ :n:'optionName' \ \ :p:', '\ \ :n:'definition' \ \ :ow:
```

```

\ :n:'row'\ \ :o:'+='\ \ :mi:'2'\
\ :k:'if'\ \ :n:'definition'\ \ :p:'[\ \ \ :mi:'0'\ \ :p:]'\ \ :o:
\ :n:'cb'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QCheckBox'\ \
\ :n:'cb'\ \ :o:'. '\ \ :n:'setChecked'\ \ :p:'(\ \ \ :n:'config'\
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'(\ \ \ :n:'cb'\ \
\ :bp:'self'\ \ :o:'. '\ \ :n:'values'\ \ :p:'[\ \ \ :n:'sectionName'

\ :k:'elif'\ \ :n:'definition'\ \ :p:'[\ \ \ :mi:'0'\ \ :p:]'\ \ :o:
\ :n:'label'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QLabel'\ \
\ :n:'label'\ \ :o:'. '\ \ :n:'setAlignment'\ \ :p:'(\ \ \ :n:'QtCo
\ :n:'spin'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QSpinBox'\ \
\ :k:'if'\ \ :n:'definition'\ \ :p:'[\ \ \ :mi:'3'\ \ :p:]'\ \
\ :n:'spin'\ \ :o:'. '\ \ :n:'setMinimum'\ \ :p:'(\ \ \ :n:'defin
\ :k:'else'\ \ :p:':'\
\ :n:'spin'\ \ :o:'. '\ \ :n:'setMinimum'\ \ :p:'(\ \ \ :o:'-\ \
\ :k:'if'\ \ :n:'definition'\ \ :p:'[\ \ \ :mi:'4'\ \ :p:]'\ \
\ :n:'spin'\ \ :o:'. '\ \ :n:'setMaximum'\ \ :p:'(\ \ \ :n:'defin
\ :k:'else'\ \ :p:':'\
\ :n:'spin'\ \ :o:'. '\ \ :n:'setMaximum'\ \ :p:'(\ \ \ :mi:'9999
\ :n:'spin'\ \ :o:'. '\ \ :n:'setValue'\ \ :p:'(\ \ \ :n:'config'\
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'(\ \ \ :n:'label'
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'(\ \ \ :n:'spin'
\ :bp:'self'\ \ :o:'. '\ \ :n:'values'\ \ :p:'[\ \ \ :n:'sectionName'

\ :k:'elif'\ \ :n:'definition'\ \ :p:'[\ \ \ :mi:'0'\ \ :p:]'\ \ :o:
\ :n:'label'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QLabel'\ \
\ :n:'label'\ \ :o:'. '\ \ :n:'setAlignment'\ \ :p:'(\ \ \ :n:'QtCo
\ :n:'text'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QLineEdit'\ \
\ :n:'text'\ \ :o:'. '\ \ :n:'setText'\ \ :p:'(\ \ \ :n:'config'\
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'(\ \ \ :n:'label'
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'(\ \ \ :n:'text'
\ :bp:'self'\ \ :o:'. '\ \ :n:'values'\ \ :p:'[\ \ \ :n:'sectionName'

\ :k:'elif'\ \ :n:'definition'\ \ :p:'[\ \ \ :mi:'0'\ \ :p:]'\ \ :o:
\ :n:'label'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QLabel'\ \
\ :n:'label'\ \ :o:'. '\ \ :n:'setAlignment'\ \ :p:'(\ \ \ :n:'QtCo
\ :n:'text'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QLineEdit'\ \
\ :n:'text'\ \ :o:'. '\ \ :n:'setEchoMode'\ \ :p:'(\ \ \ :n:'QtGui'
\ :n:'text'\ \ :o:'. '\ \ :n:'setText'\ \ :p:'(\ \ \ :n:'config'\
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'(\ \ \ :n:'label'
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'(\ \ \ :n:'text'
\ :bp:'self'\ \ :o:'. '\ \ :n:'values'\ \ :p:'[\ \ \ :n:'sectionName'

\ :n:'help'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QLabel'\ \ :p:
\ :n:'help'\ \ :o:'. '\ \ :n:'setWordWrap'\ \ :p:'(\ \ \ :bp:'True'\
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'(\ \ \ :n:'help'\
\ :n:'separator'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QFrame'\
\ :n:'separator'\ \ :o:'. '\ \ :n:'setFrameStyle'\ \ :p:'(\ \ \ :n:'Q
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'(\ \ \ :n:'separat

```

```

\ :n: 'page' \ \ :o: '.' \ \ :n: 'setLayout' \ \ :p: '(' \ \ :n: 'layout' \ \ :
\ :n: 'pages' \ \ :o: '.' \ \ :n: 'append' \ \ :p: '(' \ \ :n: 'page' \ \ :p: ')'
\ :n: 'sections' \ \ :o: '.' \ \ :n: 'append' \ \ :p: '(' \ \ :n: 'sectionName'

\ :k: 'for' \ \ :n: 'page' \ \ :p: ',' \ \ :n: 'name' \ \ :ow: 'in' \ \ :nb:
\ :c: '# Make a tab out of it' \
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'ui' \ \ :o: '.' \ \ :n: 'tabs' \ \ :o: '.' \ \
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'ui' \ \ :o: '.' \ \ :n: 'tabs' \ \ :o: '.' \ \
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'ui' \ \ :o: '.' \ \ :n: 'tabs' \ \ :o: '.' \ \

```

My first ever thing that I just don't see how it would work without a lambda :-)

Yes friends, my desktop feed agregator uRSSus has a brand new release.

You can get it at <http://urssus.googlecode.com> or <http://pypi.python.org> or via easy_install.

There is a rather large bug in that the configuration dialog may or may not work for you. Don't worry, there is no life-or-death option there, just ignore it.

As always, I am looking for a hand making it work well on Windows and Mac.

Let me know if you like it (or if you don't).

I am reading comments about ars technica's KDE 3.2 review in /., osnews, etc.

Here's the stupid argument I mentioned in the title: "GNOME is better than KDE because GNOME does X and KDE does Y"[\[#id3\]](#)

That argument is absolutely stupid unless you can show incontrovertible proof that X is better than Y. And for 99% of the instances of the argument, you can't. And even in that case, you are isolating **one** feature or decision in a very complex system.

Is "yes/No" button ordering better? Maybe. Maybe not. I know "No/Yes" dialogs piss me off, but that doesn't mean one is absolutely better.

And besides, suppose KDE decided to do all things like GNOME. Minimalistic UI, GConf-editor for anything more advanced than changing the date.

Then what's the point? Why would **another** bland, boring UI be useful at all?[\[#id4\]](#)

There's a saying: "Taste is in variety".

In an older version of this, I said footnotes were broken. They aren't, as you can see :-)

It's running pretty stable (at least for me) so I decided to fix some issues and add some features. First a screenshot:

And now text:

- A new feature that also fixes an issue:
 - vgarvardt asked about sorting by feed, for example when reading a folder. Implemented it using a "feed" column that you can enable/disable. Since your choice is persistent, there's no need to add a configuration option to the dialog.
- Important is now Starred. I really like google's idea of using stars at hand to mark things important. So, there they are. The star icon I did myself, in Inkscape, trying to respect the Reinhardt style. I think it looks nice.
- Google News feeds! Yes, I know I could just go to <http://news.google.com>, make a search and subscribe to the results page. Can my mom? Nope. So, when adding

⁸The opposite is of course just as stupid.

⁹Yes, that is KDE-centric position, I am a KDE user, and I prefer KDE, live with it :-)

new feed, just choose “Create google news feed” enter some keywords, and it’s ready.

I need to add “custom fields” to the feed config dialog, though, since once it’s created it’s just a regular feed with a long URL.

- Also did some performance and reliability work, and current SVN (r476) is actually quite a bit nicer than 0.2.10 already. So there may be another release soonish.

When trying to serialize python data, often the first thing we look at is pickle. Well, sometimes pickle can be very expensive!

Here Post is an elixir class, which means it’s an object mapped to a relational database, and comparing its properties to something is pretty much a piece of a SQL WHERE statement.

```
\ :o: '>>>' \ :k: 'print' \ :n: 'Post' \ :o: '.' \ :n: 'important' \ :o: '\n' \ :n: 'posts' \ :o: '.' \ :n: 'important' \ :o: '=' \ :err: 'z'
```

What happens when we pickle it?

```
\ :o: '>>>' \ :k: 'print' \ :nb: 'len' \ :p: '(' \ :n: 'pickle' \ :o: '.' \ :mi: '27287' \ :o: '>>>' \ :k: 'print' \ :n: 'pickle' \ :o: '.' \ :n: 'loads' \ :p: '(' \ :n: 'posts' \ :o: '.' \ :n: 'important' \ :o: '=' \ :p: ':' \ :n: 'important'
```

Yikes. What’s a better solution?

```
\ :o: '>>>' \ :k: 'print' \ :nb: 'eval' \ :p: '(' \ :s: 'Post.important==' \ :n: 'posts' \ :o: '.' \ :n: 'important' \ :o: '=' \ :err: 'z'
```

As long as we are sure no weird data will get into the eval...

I have been delaying this functionality because I expected it to be hard to do, then... it was very easy! Now you can create “metafeeds” based on queries. Here’s the first example: “Post.important==True”.

Of course, beyond a set of standard metafeeds, a UI to allow the user to create them will be *hard* but at least the basic functionality is there (BTW, it’s 15 (fifteen!) lines of code)

After 27 days of development, uRSSus is finally a really usable app. I miss nothing from Akregator, although there are a few unimplemented things. It was broken the alst couple of days, because I was rewriting large chunks of its guts, but the new guts work much better ;-)

The missing pieces (I know about):

- Favicons
- Save column width in post list
- Show total posts in feed list
- Drag&Drop is not reliable (weird things happen every few tries)

- Metafeeds (ex. your unread posts/ your starred posts) are not updated automatically
- Del.icio.us sync is not working at all
- No help
- No decent config dialog (but a functional replacement!)
- **Very** occasional DB locking.
- Logging is broken
- Need to implement backup of the DB

That's not really more than a day or two of work.

Right now (revision 570) it's working pretty good, so it's a good revision to try, if you feel up to it.

Yes, sometimes I think a feature is much harder than it really is.

So, since revision 571, every once in a while a new favicon will appear in your feed tree when you use uRSSus. No, they won't all appear at once. They will appear whenever:

- 1) You actually fetched new posts from a feed
- 2) You restart the app

Once they appear they will never change, either.

Just sent it to Python Argentina:

<http://mx.grulic.org.ar/lurker/message/20080813.155700.6b988bc8.es.html>

A little over a month ago, on July 15th, I opened a Google Code project called uRSSus. Here's [the commit](#). My goal was to try building a desktop application like if I were building a web application, using a ORM, templating, generic views, and other things.

The first thing I learned is that it was more fun to just write the application and see it grow than spending time writing the framework needed to do what I wanted, so I just kept the ORM, and the rest is pretty traditional code.

The second thing I learned is that for a hobbyist programmer, this is a golden age. I am not exactly an awesome programmer myself, and with today's tools, I could almost *wish* my app into existence. When I started programming on a PC, I had to swap floppies to change from the IDE to the compiler[1]. And if I made a mistake, the computer crashed. No, not the program. The *computer* crashed.

Now? I get a pretty dialog, a link to the position, a stack dump, etc, etc, etc. Not missing the old days at all.

Another way this is a golden age is that there is a lot of code out there. I literally had to learn my code from books. I first "got" C by reading the help for a pirated copy of Autodesk Animator's POCO extension language. There were no collections of code I could look at and learn. There were not even any large libraries of code I could legally use!

And that's another reason why this is a golden age: Open Source and Free Software. You really can be a programmer just by willing it and effort. You will not lack tools, you will find users (if you are good), you will find helpers (if you are lucky), you will

find free infrastructure (svn repos, free wikis, free file hosting, free everything), you will find libraries you can use!.

The third thing I learned is that [Python](#) does come with batteries included. Many things that would be annoying effort in other languages are just there, ready to be used. Add the internet, and it's a [Mr. Fusion](#) instead of a battery.

The application I developed is a [News aggregator](#) and thanks to Mark Pilgrim I had [Feed Parser](#) and thanks to Troll Tech (Now Nokia) I had Qt for the UI, and many many other things. I could focus on application logic, not on parsing and drawing.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "download".
```

The fourth thing I learned is that a month is a long time when you have productive tools. [Urssus](#) (that's my application) was functional (but awful) in a day or two. It was not awful in 2 weeks. It was pretty good in 3. In a month? [Download](#) it and see for yourself, I like it, the SVN version is much better most of the time, try revision 619 ;-)

The fifth thing I learned is that Python performance is good enough. I don't see much performance difference between [uRSSus](#) and, say, [Akgator](#), which is C++, except on places which are obviously broken. Sure, the database is C, the UI toolkit is C++... they are all black boxes to me here. I code Python. My pieces do well.

The last thing I learned is that I can still code free software. I had not written a useful/usable *large* free software application in perhaps 8 years. I am 36.9 years old... excuse me if I feel middle-aged, surrounded by youngsters which are faster, more dedicated and actually have free time.

Because of the productivity of the tools, I managed to code just a couple of hours a day for the first weeks, and progress was still good, so I didn't get discouraged, which is the worst enemy of free software.

It has been a fun experiment, hopefully it will be a fun *ongoing* hobby.

```
system-message
WARNING/2 in everything.txt, line 19070
Duplicate explicit target name: "download".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "rst2pdf".
```

Because of a thread in [the PyAr list](#) about generating reports from Python, I suggested using [ReST](#) and my [rst2pdf](#) script.

This caused a few things:

1

```
system-message
WARNING/2 in everything.txt, line 19080
Duplicate explicit target name: "1".
```

Can you guess what I was using?

- 1) I decided it's a pretty decent piece of code, and it deserves a [release](#). Making a release means I needed to fix the most embarrassing pieces of it. So...
- 2) Implemented the class directive, so it can have custom paragraph styles with very little effort.
- 3) Did proper command line parsing.
- 4) Did proper setuptools script
- 5) Uploaded to PyPI
- 6) Created a release in Google Code.

So, if you want the simplest way to generate PDF files from a program in the entire pythonic universe... give it a look.

system-message

WARNING/2 in `everything.txt`, line 19107
Duplicate explicit target name: "rst2pdf".

This article is inspired by a [thread](#) in the PyAr mailing list. Here's the original question (translated):

From: Daniel Padula

I need some advice. I need to create an application for schools that takes student data (personal information, subjects, grades, etc) and produces their grade report. I need to create a printed copy, and keep a historic record.

As a first step, I thought on generating them in PDF via reportlab, but I want opinions. For example, I can generate the PDF, print it and regenerate it if I need to reprint it. What other options do you see? It's basically text with tables. Reportlab? LaTeX? Some other tool?

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "restructured text".

To this I replied I suggested [Restructured Text](#) which if you follow my blog should surprise noone at all ;-)

In this story I will try to bring together all the pieces to turn a chunk of python data into a nice PDF report. Hope it's useful for someone!

Why not use reportlab directly?

Here's an example I posted in that thread: how to create a PDF with two paragraphs, using restructured text:

```
This is a paragraph. It has several lines, but what it says does not matter
I can press enter anywhere, because
it ends only on a blank
line. Like this.
```

```
This is another paragraph.
```

And here's what you need to do in reportlab:

```
# -*- coding: utf-8 -*-
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer
from reportlab.lib.styles import getSampleStyleSheet
from reportlab.lib.units import inch
styles = getSampleStyleSheet()
def go():
    doc = SimpleDocTemplate("phello.pdf")
    Story = [Spacer(1,2*inch)]
    style = styles["Normal"]
    p = Paragraph(''This is a paragraph. It has several lines, but what it s
I can press enter anywhere, because
it ends when the string ends.'', style)
    Story.append(p)
    p = Paragraph(''This is another paragraph.'', style)
    Story.append(p)
    Story.append(Spacer(1,0.2*inch))
    doc.build(Story)

go()
```

Of course, you could write a program that takes text separated in paragraphs as its input, and creates the reportlab Paragraph elements, puts them in the Story and builds the document... but then you are reinventing the restructured text parser, only worse!

Restructured text is data. Reportlab programs are code. Data is easier to generate than code.

So, how do you do a report?

system-message

```
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "rst2pdf".
```

You create a file with the data in it, process it via one of the many rst->pdf paths (I suggest my [rst2pdf](#) script, but feel free to use the other 9 alternatives).

Suppose you have the following data:

```
frobtimes = [[1,3],[3,5],[9,8]]
```

And you want to produce this report:

```
Frobniz performance
=====

1 frobniz: 3 seconds

3 frobniz: 5 seconds

9 frobniz: 8 seconds
```

You could do it this way:

```
print '''Frobniz performance
====='''

for ft in frobtimes:
    print '* %d frobniz: %d seconds\n'%(ft[0],ft[1])
```

And it will work. However, this means you are writing code again! This time, you are reinventing templating languages.

What you want is to use, say, Mako (or whatever). It's going to be better than your homebrew solution anyway. Here's the template for the report:

```
#{title('Frobniz Performance')}

% for ft in frobtimes:
    #{ft[0]} frobniz: #{ft[1]} seconds

% endfor
```

This uses a function `title` defined thus:

```
title=lambda(text): text+'\n'+'='\*len(text)+'\n\n'
```

You could generalize it to support multiple heading levels:

```
title=lambda(text,level): text+'\n'+'=-~_#%^' [level]*len(text)+'\n\n'
```

Trickier: tables

One very common feature of reports is tables. In fact, it would be more natural to present our frobniz report as a table. The bad news is how tables look like in restructured text:

```
+-----+-----+
| Frobniz | Time (seconds) |
+-----+-----+
|         1|                 3 |
+-----+-----+
|         3|                 5 |
+-----+-----+
|         9|                 8 |
+-----+-----+
```

Which is very pretty, but not exactly trivial to generate. But don't worry, there is a simple solution for this, too: CSV tables:

```
.. csv-table:: Frobniz time measurements
   :header: Frobniz,Time(seconds)

   1,3
   3,5
   9,8
```

Produces this:

Frobniz time measurements

Frobniz	Time(seconds)
1	3
3	5
9	8

And of course, there is python's csv module if you want to be fancy and avoid trouble with delimiters, escaping and so on:

```
def table(title,header,data):
    head=StringIO()
    body=StringIO()
    csv_writer = csv.writer(head, dialect='excel')
    csv_writer.writerow(header)

    head=':header: %s'head.getvalue()

    csv_writer = csv.writer(body, dialect='excel')
    for row in data:
        csv_writer.writerow(row)
    body=body.getvalue()

    return '''.. csv-table:: %s
       :header: %s

       %s
       '''%(title,head,body)
```

will produce neat, ready for use, csv table directives for restructured text.

How would it work?

This python program is really generic. All you need is for it to match a template (an external text file), with data in the form of a bunch of python variables.

But how do we get the data? Well, from a database, usually. But it can come from anywhere. You could be making a report about your del.icio.us bookmarks, or about files in a folder, this is really generic stuff.

What would **I** use to get the data? I would use JSON in the middle. I would make my report generator take the following arguments:

- 1) A mako template name.
- 2) A JSON data file.

That way, the program will be completely generic.

So, put all this together, and there's the superduper magical report generator.

Once you get rst, pass it through something to create PDFs, but store only the rst, which is (almost) plain text, searchable, easy to store, and much smaller.

I don't expect such a report generator to be over 50 lines of code, including comments.

Missing pieces

- While restructured text is almost plain text, there *are* special characters, which you should escape. That is left as an exercise to the reader ;-)
- Someone should really write this thing ;-)

```
system-message
WARNING/2 in everything.txt, line 19353
Duplicate explicit target name: "rst2pdf".
```

```
system-message
WARNING/2 in everything.txt, line 19355
Duplicate explicit target name: "restructured text".
```

Since revision #17_ you can display Page numbers in headers and footers (only!) by using this syntax:

```
\ :p: '..' \ \ :ow: 'header' \ \ :p: ':': '\
    This is the header. Page ###Page###

This is the content

\ :p: '..' \ \ :ow: 'footer' \ \ :p: ':': '\
    This is the footer. Page ###Page###
```

It has some issues if your page number is bigger than 9999999999 or your header/footer is a little longer than one line when using the placeholder, because the space required is calculated **with the placeholder** instead of with the number, but those are really marginal cases.

Next in line, a decent way to define custom stylesheets.

As for “time-based releases”, I intend to release a new version of **something** every friday.

Since I have about a dozen projects in different stages of usability, I expect this will push me a bit more towards showing this stuff instead of it rotting in my hard drive and unknown svn repos.

I just finished reading “Into Deepest Space” by Fred and Geoffrey Hoyle. It is the worst science fiction book I have ever read. And I have read hundreds.

Mind you, perhaps the translation is to blame for the flat tone and the dreadful dialogue. However, the stupid storyline and the undescribed location and characters surely come from the original.

In this book, there are three aliens, who come from Betelgeuse (although they are genetically almost-humans), and their names are Betelgeuse, Rigel and Alcyone. I have no idea how they got to be named as starts, but anyway, it took me 50 pages to figure out Alcyone was female, and 120 to see a mention of her being romantically linked to the human (called, I kid you not, Dick Warboys).

But enough of that. After I finished this garbage (which is at least short), I started thinking, what was my favourite scifi book, since this one is surely the opposite.

I have to say “The dispossessed” by Ursula K. LeGuin.

As I have discussed with [Uwe](#) a few times (although we are both fans of UKLG, so it’s no argument :-), the reason her books are so good is that they are about people.

She is one of the biggest humanists writers I know. While some authors write about people who are more than human in some sense, like Heinlein, whose alter ego is usually trivial to recognize, and is super-competent and really, pretty super-human, LeGuin’s characters are more human than us.

They are good, or evil, or something in between, but they are what they are for human reasons, with human motivation, for human goals, with human purposes.

You can understand the guys. You can hate them or like them, but you can understand them.

For example, Shevek, the main character in *The Dispossessed*, is, for human eyes, supremely uninterested in power, wealth, material possessions, which would seem to make him superhuman or inhuman.

After all, we all consider ambition and desire of possession, material or personal as natural, because we experience them, we **embrace** them!.

But he is not described as a character who simply is like that. He **became** like that through extensive and deep education, almost brainwashing, although brainwashing with a seemingly positive goal in the described circumstances.

Now, is it good? He seems to be happy. His society, formed of likeminded folks is described as happy, and good, and he even describes the wealthy earth-like capitalist world he visits as “hell”.

Would we be better off if we were brainwashed into being good? Aren’t we, only inefficiently?

Is there moral value on Shevek’s being ethical after he has been educated to be **unable** to be unethical?

Does it matter?

Those are questions a Heinlein book will not make you ask, and they are worthy questions, and they have no easy answers. And they are all deeply human problems.

And although the competent men of scifi are fun to read, and cause me to think of problems of different nature, which are worthwhile in themselves, I think Ursula K.

LeGuin's books actually make me learn and think about more important things. I learn about people from them.

They are good for you. And they are well written. And they are a fun read. And what more can one ask?

Following my new policy of one release every friday, in 6 days you will see a rst2pdf release. But not any release: a *great* release.

What will be new?

- Support for page number/section names/section numbers in headers and footers.
- Custom interpreted text roles (that means inline styling ;-)
- Stylesheets defined in external files. The syntax is JSON which may look a bit strange, but it works great.
- A Manual!
- Easy True Type font embedding.
- Maybe: syntax highlighting directive via pygments. I know I could make it work using the ImageFormatter, but then you can't copy the code. There is a docutils-sandbox project that does exactly what I want.

I intend to call this release 0.3.0, but maybe I will jump higher, since there is not much more left to implement.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

This mini-sprint is doing wonders for rst2pdf. Now on SVN: pygments-based syntax highlighting. Example [here](#): rst2pdf's code, in a PDF by rst2pdf.

system-message

WARNING/2 in `everything.txt`, line 19447
Duplicate explicit target name: "here".

I did a release yesterday, and another today of my rst-to-pdf-without-latex tool. What's new? Here's an incomplete list:

New in 0,4

- Fixed bullet and item lists indentation/nesting.
- Implemented citations
- Working links between footnotes and its references
- Justification enabled by default
- Fixed table bug (demo.txt works now)
- Title and author support in PDF properties
- Support for document title in header/footer

- Custom page sizes in stylesheet

New in 0.3

- Font embedding (use any True Type font in your PDFs)
- Syntax highlighter using Pygments
- User's manual
- External/custom stylesheets
- Support for page numbers in header/footer

Of course, since I said I would release **something** every friday, this means I need to find something else to release? ;-)

I am a big Alexandre Dumas fan. He's the direct ancestor of Neal Stephenson, so many of you should like him too. So I used one of his best books to try some automatic typesetting of project gutenber texts.

No, the whole book did not convert without errors, and yes, there is *some* manual work in what you are about to see, but hey, take a look.

Here's a far look of the first two pages:

And here's some detail of the typesetting:

Yes, the typesetting is not really LaTeX quality, but it's not bad, either.

Compare it with [the HTML version at project Gutenberg](#). The typesetting is a thing of **beauty** compared to that :-)

The image is a picture of Chateau d'If from flickr, released under Creative Commons. The title font is Scriptina, I chose it because it looks 19th century but modern.

How good? Let me tell you...

- Support for PDF table of contents
- Section names and numbers in headers/footers
- Compressed PDFs (or not)
- Guess image sizes. Specially if you meant to use them in a web page and declared just “:width: 50%”
- Gutter margin support
- Raw directive (insert pagebreaks and vertical space manually)
- Offers a docutils-compliant API (and another API, too)
- Include full or partial files for code-block. That means you can extract code and show it in your document!
- Huge code cleanup lead by Nicolas Lurance.
- Working multilingual hyphenation. You can have a per-paragraph language and hyphenate it correctly.

Besides everything I mentioned yesterday, today I implemented two rather important features: cascading stylesheets, and user-defined page layouts. Here is a screenshot:

That neat two-column layout is done by adding this to the stylesheet:

```

"pageTemplates" : {
  "firstPage": {
    "frames": [
      ["0cm", "0cm", "49%", "100%"],
      ["51%", "0cm", "49%", "100%"]
    ]
  }
},

```

The name “firstPage” is magical right now, and there’s no way to change from one template to another (yet), and until I do that, I won’t be releasing.

Here’s what cascading stylesheets does. Suppose you want to use A5 paper and size 12 Times New Roman fonts? Here’s all the stylesheet you need:

```

{
  "pageSetup" : {
    "size": "A5",
  },
  "fontsAlias" : {
    "stdFont": "Times-Roman",
  },
  "styles" : [
    ["base" , {
      "fontSize": 14,
      "leading": 16
    }]
  ]
}

```

Also, you can specify as many stylesheets as you want in the command line. So you can have one that sets the paper size, one for page layout as above, one for font “sets”, etc.

Neat, isn’t it?

Many new features. Custom page layouts! Multiple frames per page! Multiple layouts per document! Cascading stylesheets! Not very buggy! Get it at <http://rst2pdf.googlecode.com> or via PyPI.

Today I started a branch called SmartFrames. The main goal is to achieve a better text flow in the document (for example, for sidebars), and it is starting to get there, slowly.

Let’s consider how ReST sidebars are rendered in the different writers.

We’ll work with an ordinary lorem ipsum that has a sidebar declared just before it.

Here’s HTML:

And here’s LaTeX:

Each one has its good side and its bad side.

The HTML sidebar is a real sidebar, while the LaTeX one is some sort of insert.

OTOH, the ragged text against the HTML sidebar is ... horrid.

So, I wanted something at least a bit better than that for rst2pdf. In the best of all possible worlds, it would be the neat text alignment of LaTeX with the floating HTML sidebar.

Here’s how it looks now:

There are some minor problems with the current implementation, such that the sidebar is always aligned to the top of a paragraph, and some spacing issues.

How is it done? Let me tell you: it was not trivial :-)

In fact it's pretty evil, but here's a quick explanation:

When I get a sidebar flowable, I insert a **new frame** in the page template where the sidebar should go, then call a framebreak, insert the "real" sidebar, a "framecutter" and another framebreak.

The framecutter is a flowable that does nothing visible, but inserts another two frames, one at the right of the sidebar with the same height, and another below the sidebar, full width.

I need to use the framecutter because I don't know the height of the sidebar until after it's drawn.

So, we now have 4 frames instead of one:

- 1) The original frame, covers the whole page, but has a framebreak above the sidebar.
- 2) The sidebar frame, which is very tall, but has a framebreak below the sidebar text.
- 3) A beside-the-sidebar frame, short and wide, starting at the right of the sidebar.
- 4) A below-the-sidebar frame, wide and tall, starting below the sidebar.

The text should flow from 1 to 3 to 4 neatly and the seams shouldn't show.

Here's a picture that MAY make it clear (there are some odd displacements: those were bugs):

So, I'm not calling it a success yet, but it *is* looking decent.

Yes, it is getting better. Now there are left-floating or right-floating elements, and you can float pretty much anything (except literal blocks, that's a problem) by using the class directive. Here's how it looks:

Keeping with my new time-based release schedule, tomorrow is again rst2pdf release day! What will be new in 0.7? Several things!

- Fixed several serious bugs, specially with the handling of literal blocks (preformatteds and code blocks).
- Implemented a rather neat sidebar/floating block mechanism. It only lets you go float-left or float-right but it looks better than what you get in HTML, at least:

You can even use it to float *almost* arbitrary objects, so you can have floating images or floating figures.

- Fixed the look of hyperlinks. Reportlab had a bug about making *some* hyperlinks underlined, and those with a black **thick** underline. Not when you use rst2pdf! Monkey-patched the heck out of that.
- Table styling. Let me show you:

The first row is a header row. It automatically takes the table-heading style.

The following rows are regular, and they take the table style, which has support for *zebra* tables with alternating colors (white and gray here).

The lone red cell at the right is special. Its content is this:

```
.. class:: red
```

```
red
```

If you don't know ReST, that means "red" is a paragraph with *class* red, so it will be styled whatever way that means (here: red background).

Usually that would mean you have a white (or gray) cell with a red paragraph in it. That looks incredibly ugly.

So, rst2pdf tries to be clever: If there is a single element in a cell, it will try to guess the cell background from it.

And as you saw above, it works :-)

As mentioned before, here's a picture of the remote control in a hotel room I stayed this weekend. Notice the problem? Click on it for a larger version.

I am not sure what the lesson is here. It can't be that a user interface **must** be good. This guy equipped a full hotel with these.

Right on schedule, everything I promised, plus much simpler/smarter font embedding, and maybe a bag of chips. More at <http://rst2pdf.googlecode.com>.

On my ongoing something-released-every-friday rampage, this friday I will release a working version of Uqbar, a Gutenberg project e-texts interface.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "rst2pdf".

It may use use pandoc and [rst2pdf](#) and [makeztxt](#) and other things to make e-books that are pretty to read, and work on my Sony Clie. We'll see.

This project has been "sleeping" since 2005, I think I can now, with my new focus in limited, quick development, get a reasonable version ready in two weeks.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "rst2pdf".

I may slip a little [rst2pdf](#) update, too.

After that, back to [Urssus](#).

system-message

WARNING/2 in `everything.txt`, line 19730
Duplicate explicit target name: "rst2pdf".

I had one too many problems with kmail from KDE4 in my eee with Kubuntu, and [sylpheed-claws](#) is just unusable in a small screen (the huge widgets! the non-hidable things all over the interface!) I decided to get old fashioned and try a console mail reader.

I was a pine user for many years, and a mutt user for a while, and I was deeply disappointed that the last three years have been bad for these programs.

Just because you run in a terminal, there's no reason to be hard to configure! After spending 20 minutes trying to get a decent IMAP account setup (just two IMAP accounts) in [alpine](#) and another 10 wondering if [mutt](#) really had no place in the UI for

account configuration (and whether the Debian/KUbuntu default “commented” config file is the product of hard drugs), I tried [Cone](#).

It was bliss. It was all I remembered from pine years ago (on local accounts) only over IMAP.

Easy to configure, easy to use, quick, capable. I was in love all over again.

While I will keep using KMail on my main notebook under Arch Linux where I had no reliability problems whatsoever, I have Cone configured in my server’s shell account and in the eee.

One of the big limitations of reportlab is that it has no support for vector-based images. You can’t insert SVG, EPS or any other vector-based format in your documents.

Until now.

By hijacking another app called [uniconvertor](#), I have managed to insert *as vectors* SVG images in a reportlab document.

Here’s the hackish code:

```
\ :kn:'import' \ \ :nn:'sys' \ \ :o:', ' \ \ :nn:'os' \
\ :kn:'from' \ \ :nn:'app.io' \ \ :kn:'import' \ \ :n:'load' \
\ :kn:'from' \ \ :nn:'app.plugins' \ \ :kn:'import' \ \ :n:'plugins' \
\ :kn:'import' \ \ :nn:'app' \
\ :kn:'from' \ \ :nn:'reportlab.platypus' \ \ :kn:'import' \ \ :o:'*' \
\ :kn:'from' \ \ :nn:'reportlab.lib.styles' \ \ :kn:'import' \ \ :n:'getSampleStyleSheet' \
\ :kn:'from' \ \ :nn:'reportlab.lib.units' \ \ :kn:'import' \ \ :n:'inch' \

\ :n:'app' \ \ :o:'. ' \ \ :n:'init_lib' \ \ :p:'()' \
\ :n:'plugins' \ \ :o:'. ' \ \ :n:'load_plugin_configuration' \ \ :p:'()' \

\ :k:'class' \ \ :nc:'SVGImage' \ \ :p:'(\ \ \ :n:'Flowable' \ \ :p:'):' \
\ :k:'def' \ \ :nf:'__init__' \ \ :p:'(\ \ \ :bp:'self' \ \ :p:', ' \ \ :n:'in' \
\ \ :bp:'self' \ \ :o:'. ' \ \ :n:'doc' \ \ :o:'=' \ \ :n:'load' \ \ :o:'. ' \
\ \ :k:'for' \ \ :n:'k' \ \ :ow:'in' \ \ :nb:'dir' \ \ :p:'(\ \ \ :bp:'self' \
\ \ :k:'print' \ \ :n:'k' \
\ \ :bp:'self' \ \ :o:'. ' \ \ :n:'saver' \ \ :o:'=' \ \ :n:'plugins' \ \ :o:'. ' \
\ \ :n:'Flowable' \ \ :o:'. ' \ \ :n:'__init__' \ \ :p:'(\ \ \ :bp:'self' \ \ :o:'. ' \

\ :k:'def' \ \ :nf:'wrap' \ \ :p:'(\ \ \ :bp:'self' \ \ :p:', ' \ \ :n:'aW' \ \ :n:'br' \
\ \ :n:'br' \ \ :o:'=' \ \ \ :bp:'self' \ \ :o:'. ' \ \ :n:'doc' \ \ :o:'. ' \ \ :n:'br' \
\ \ :k:'return' \ \ :n:'br' \ \ :p:'[\ \ \ :mi:'2' \ \ :p:], ' \ \ :n:'br' \

\ :k:'def' \ \ :nf:'drawOn' \ \ :p:'(\ \ \ :bp:'self' \ \ :p:', ' \ \ :n:'canv' \
\ \ :n:'canv' \ \ :o:'. ' \ \ :n:'translate' \ \ :p:'(\ \ \ :n:'x' \ \ :p:', ' \
\ \ :bp:'self' \ \ :o:'. ' \ \ :n:'saver' \ \ :p:'(\ \ \ :bp:'self' \ \ :o:'. ' \
\ \ :n:'os' \ \ :o:'. ' \ \ :n:'unlink' \ \ :p:'(\ \ \ :s:'".ignoreme.pdf"' \

\ :n:'styles' \ \ :o:'=' \ \ :n:'getSampleStyleSheet' \ \ :p:'()' \
\ :k:'def' \ \ :nf:'go' \ \ :p:'():' \
\ \ :n:'doc' \ \ :o:'=' \ \ :n:'SimpleDocTemplate' \ \ :p:'(\ \ \ :s:'"phel' \
\ \ :n:'Story' \ \ :o:'=' \ \ :p:'[\ \ \ :n:'Spacer' \ \ :p:'(\ \ \ :mi:'1' \
\ \ :n:'style' \ \ :o:'=' \ \ :n:'styles' \ \ :p:'[\ \ \ :s:'"Normal"' \ \ :n:'style' \
```

```

\ :n: 'p' \ \ :o: '=' \ \ :n: 'SVGImage' \ \ :p: '(' \ \ :s: 'warning.svg' \ \
\ :n: 'Story' \ \ :o: '.' \ \ :n: 'append' \ \ :p: '(' \ \ :n: 'p' \ \ :p: ')' \ \
\ :n: 'doc' \ \ :o: '.' \ \ :n: 'build' \ \ :p: '(' \ \ :n: 'Story' \ \ :p: ')' \ \

\ :n: 'go' \ \ :p: '()' \ \

```

It has several problems (see where the second paragraph ended), but it *does* work. To run it, you need a `warning.svg` file (edit as needed) and you run it this way (replacing the path as proper for your setup):

```
PYTHONPATH=/usr/lib/python2.5/site-packages/uniconvertor/ python use_unicon
```

In fact, this is not limited to SVG files. You should be able to use the following formats:

- CorelDRAW ver.7-X3,X4 (CDR/CDT/CCX/CDRX/CMX)
- Adobe Illustrator up to 9 ver. (AI postscript based)
- Postscript (PS)
- Encapsulated Postscript (EPS)
- Computer Graphics Metafile (CGM)
- Windows Metafile (WMF)
- XFIG
- Scalable Vector Graphics (SVG)
- Skencil/Sketch/sK1 (SK and SK1)
- Acorn Draw (AFF)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “`rst2pdf`”.

I just committed into trunk of [rst2pdf](#) a nicely working `SVGImage` flowable for reportlabs.

The code is self-contained in two files:

- [svgimage.py](#)
- [uniconvsaver.py](#)

You can use them in your app if you want (I’d like to know, though). One of them is GPL/LGPL (it’s basically copied from uniconvertor) the other is MIT-Licensed.

The output is astonishing: [svg.pdf](#) is only 32KB and completely resolution-independent. Graphics in your PDF That look good printed and on screen! And are not huge! Yay!

This will be the flagship new feature in tomorrow’s `rst2pdf` 0.8.

system-message

WARNING/2 in `everything.txt`, line 19856
Duplicate explicit target name: “`rst2pdf`”.

So, I failed to release uqbar this week. However, I am releasing rst2pdf 0.8, with SVG support! Get its vectorial goodness from <http://rst2pdf.googlecode.com>

Of course I could be misunderstanding this, but it seems they included uRSSus 0.2.10 in a DVD for some reason :-)

[The link](#) and I will *not* make a “in soviet russia” joke.

I announced it on all other places except here: rst2pdf 0.8.1 is out. What’s new?

- Complex headers and footers
- Optional inline links
- Wordaxe 0.2.6 support
- Many bugfixes

Get it at [the usual place](#)

If I manage to make it work well: Math!

Here’s how it looks:

It does **not** require TeX. It is **not** an image. It does require external software, and it does require you to write TeX syntax.

If you are a reportlab user: this is a platypus flowable.

As usual, it’s not me that’s doing the heavy lifting. I am just good at using other tools.

With a little luck, it should be usable this friday.

Well, what the title says. Get it at [the usual places](#). No, it doesn’t require actual LaTeX. Just Matplotlib.

SCO is becoming Intergraph.

For those who don’t know it, Intergraph was once a workstation maker. Last year they had 10M in business income. And 385M in lawsuit income.

Basically the company sits in a few patents, does nothing, and milks everyone else (not that it’s illegal to do so).

And why I say SCO is becoming intergraph?

Let’s look at the possible income sources for SCO, shall we?

- a) SCO OpenServer and UnixWARE: pretty much just upgrading old servers.
That is not going to pay for R&D for new versions, and then this dries up, because no new version == no upgrades == no income. Unless of course, this gets subsidies from somewhere else.
- b) Caldera Linux == If this suit killed something, this is what got whacked
- c) Licensing of IP, either friendly or forceful a.k.a the Intergraph path
- d) There is no d!

And about c)... who is going to license the UNIX stuff from SCO anymore? This suit means if you do it, you can NEVER AGAIN develop a unix system or a linux system, or SCO will sue your ass. To do that to get a technology that is not too much better than FreeBSD... it makes no sense. Specially considering there will be no R&D on it anymore (see a).

The only ones with UNIX licenses will be those who got them from AT&T, and they don’t pay a dime to SCO. BTW: this basically means Sun, since HP-UX and IRIX and the rest is on their death bed anyway (and AIX... well, just think about it ;-).

So c), the only visible income source for SCO becomes “We will sue people, and live on what we get out of settlements”.

Because SCO can’t even do a GOOD Intergraph and actually win the lawsuits. Not this one, and not against IBM, at least, and they don’t have a good patent portfolio!

Tu put it another way, SCO will become a bully that will try to look impressive and steal milk money from smaller companies. That is, they will, if IBM lets them.

If IBM caves in and settles, SCO will look impressive enough for this to work. Which is why I hope IBM beats the crap out of them, just to make SCO have to work to earn an honest buck.

I spent a few hours today round [Code Golf](#) and here’s a neat thing I did. I think this is python’s shortest possible factorial:

```
f=lambda x:x<2or x*f(x-1)
```

You may say it’s not correct, because f(1) returns True, but int(True) is 1, so it’s almost there ;-)

And here’s a surprisingly readable (as long as you know pascal’s triangle) and not too long answer to their [choose](#) challenge:

```
\ :n:'n'\ \ :p:', '\ \ :n:'k'\ \ :o:'='\ \ :nb:'input'\ \ :p:'()\ '\
\ :n:'a'\ \ :p:', '\ \ :n:'c'\ \ :o:'='\ \ :mi:'l'\ \ :p:', '\ \ :mi:'0'\
\ :k:'while'\ \ :n:'c'\ \ :o:'<'\ \ :n:'k'\ \ :p:': '\ \ :n:'a'\ \ :o:'*='\
\ :k:'print'\ \ :n:'a'\
```

First a huge announcement: I will not release anything tomorrow. For the first time in 6 weeks, no friday release from me!

Now, let’s see what I’m planning.

- 1) I will be at [LatinoWare 2008](#) speaking about PyQt. The usual free beer offer: if you mention this blog, I will buy you 1 (one) not very good beer. Being in Brazil, that’s probably a Brahma.

2)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “`rst2pdf`”.

`rst2pdf` development is slowing down. The easy stuff is all done. There are some things I can do, though:

- Expose transitions to the writer (good for presentations)
- “Background” forms (again, for presentations)
- A mini-language exposing the canvas, so there is a useful “raw” mode.
- Using `svglib` instead of `uniconvertor` to handle SVG (but keep `uniconvertor` for other vector formats)
- Redoing lists, which have a shaky code and bad layout
- Fix half a dozen bugs
- Make it work more like a unix app when in unix (respect environment vars, for example)

However, I am taking a little time off from it, and going to try reviving other projects. Focusing in rst2pdf for two months did wonders for it. Let's see if it can be repeated!

- 3) Try some more [code golfing](#). I helped Nubis from [PyAr](#) reach the lowest-score of 74 in the grid challenge. This is really fun stuff.
- 4) I need more money. We have spent lots in ... *plumbing*. Yup. Two years of savings spent on that. So, I need money. If you read this blog and want to make me happy, get me freelance work. It can be sysadmin stuff, it can be programming, it can be babysitting. I am good at all three things. Contact me. I'm easy to find, and I am cheap(ish).
- 5) No, I am not starving. No, I am not in money problems. Yes, [my company](#) is doing very well, but it's an investment, not a cash cow (yet).

system-message

WARNING/2 in `everything.txt`, line 19998
Duplicate explicit target name: "rst2pdf".

If you read this blog on the website (and you do, because my feed is garbage ;-) you may have noticed a [goodreads.com](#) widget on the right.

It's not that my reading habits are interesting to anyone, since I have become a very shallow reader in the last two years or so (a terrible, terrible involution caused by all the free sci-fi ebooks floating around), but I love that site. In the past I posted some book-related stuff here, but it was always sort of guilty about it, posting things my readers (few as they are) probably don't give a damn about.

With goodreads, I have a way to vent my shallow opinion about my shallow readings. I need the same thing for movies.

Also, it's a much more reasonable thing than all that "now listening" idea. Who cares what you are listening? It lasts for 3 minutes! A book is something you have in your head for at least a couple of days! It is so much more meaningful.

I have been thinking about using rst2pdf to create presentations. I finished exposing PDF transitions today, and that part works nicely.

Still needed:

- Some sort of "page decoration" feature.
- Handout support? I think this would work nicely with some sort of "conditional" directive. I need to think about it some more.

I intend to have that done for next friday, and release then.

Many of my clients have been spammed by [La Capitana Real Estate](#) lately. And I mean *many*. Hundreds.

However, they seem to have found a way to spam that works. And that sucks.

They have created a Google [Group](#), added all their victims there, and let google do the dirty work.

What's the problem?

- 1) Google group mails are not blockable at SMTP-level because their senders contain a sort of hash and the recipient address, **and no group name**. That's incredibly stupid in google's part.

- 2) The messages they send are huge (6MB and up) so spamassassin can not process them. The SA docs say this will not happen because of “the economics of spam”. Well, it happens when you make google do it!
- 3) I don’t want to go back to the old days of keeping a local queue-level address blacklist. That’s awful!

I have complained to google, I have complained to the spammers, even by phone. They use the standard defense of “we are just inviting people”. “They can unsubscribe if they want to”. “This is not spam”

Noone does anything.

What’s the next step? I can’t blacklist google groups!

Just finished speaking about PyQt programming in LatinoWare.

It went ok, but I ran out of time, so no Q&A, and I couldn’t show about 20% of the code. :-(-

system-message

WARNING/2 in `everything.txt`, line 3
 Duplicate explicit target name: “here”.

Anyway, [here](#) is the code in its different stages, so you can see the “evolution” of a simple PyQt app. I will probably turn it into a real tutorial in a few days and post it.

system-message

WARNING/2 in `everything.txt`, line 20056
 Duplicate explicit target name: “here”.

Here is the guide I was using for myself when doing the presentation:

1.
 - a) App backend (1/todo.py)
 - b) Basic window with a treewidget Columns Task/Date/Tags
 - c) Compile archivo .ui, show how it works
 - d) Template for main.py
2.
 - a) Load backend data in the task list
 - b) Show interaction is broken
3.
 - a) Explain signals/slots
 - b) Show difference between 1/main.py 2/main.py
 - c) Explain autoconnect, show on_lista_itemChanged
 - d) Show task persistence working
4.
 - a) Fix tree decorations
 - b) Make items editable
 - c) Explain changes in on_lista_itemChanged
 - d) Explain the problem with editing date and tags inline
 - e) New idea: use a dialog to edit tasks (editor.ui)
 - f) Explain layout managers

5. a) Show changes in main.py
6. a) Actions
 - b) Show actions in designer
 - c) Show actions in main.py
7. a) Tray
 - b) Show tray class in main.py
 - c) Show tray behaviour
 - d) Show differences in main.py

Last night while eating one of the Python Brazil guys told me about the Lightning Talks session taking place today.

Since I am basically insane, I was convinced to do my first lightning talk:

- In spanish in front of a purely brazilian audience
- Creating it while having breakfast 1 hour before the event
- After exceeding the allotted 50! minutes 2 days ago

So, in short, I was terrified. I got there, sat at the back, start listening.

I was 4th in order.

1: doctests (nice)

2: pronounceable passwords (nice)

3: MVC in gtk... I didn't understand a word, and ran out of time.

So, I am now thinking nobody will understand me at all, so I will slow down, run out of time, and will then give the worst lightning talk ever.

I'm next up. My topic: how to build a spreadsheet in 5 minutes.

Amazingly, it worked rather well, the "MC" asked me for a copy of the code, and said it was a neat example. Noone looked horrified or ROFLing, so I am at least somewhat happy about it.

Here is the code for you, too: [ssin5min.tgz](#)

I would do it again. But if I am doing it again here, I am learning portuguese first.

Yes, again, two months passed and nothing on the blog. What can I say, my life takes a lot of my time.

Anyway: I will try to actually write my PyQt tutorial based on what I showed at LatinoWare (I said "will write it in a few days". Ja!), and start pushing things out every once in a while.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "rst2pdf".

The only FLOSS I did was some bugfixes for [rst2pdf](#) which is desperately needing a new release.

Hopefully march will mean a bit more time for free software and writing.

On other areas:

- KDE 4.2 is *really* good. Beats 3.5 for me, like for most anyone else that was not convinced yet.

- I want to try my hand at a couple of plasmoids (yes, I do have at least one semi-original idea)

So, see you soon, dear readers. Maybe. Probably.

system-message

WARNING/2 in `everything.txt`, line 20159
Duplicate explicit target name: “`rst2pdf`”.

So, I must confess something. I have not been using [uRSSus](#) for a couple of months. It all started on my vacations. Since I had only my eee, I ran into some problems and was forced to use google reader.

And it was NOT terrible. Mind you, it was not great, either, but really, not terrible at all.

Since switching RSS apps is a pain (because you need to REALLY catchup on your reading before switching), it took me a long time to get back.

And now I find all sorts of problems in [uRSSus](#) which I will have to fix. Which means I better start cranking.

Not that it looks bad, it's a handsome program!

I don't expect to do anything like last year's "A release of *something* every friday" madness, but I will do something.

Also, I need to rewrite BartleBlog, the app that generates this blog you are reading. It's broken in many ways.

On my latest disappearance, this blog turned 9 years old. I started it (in [advogato](#), where you can still [read](#) it, BTW), on January 17th, 2000.

It is currently my longest-running project, since at the time I lived in a different province, have had several girlfriends (and a wedding), I had two or more jobs, started one company, went through about 9 presidents, and of course, there's my boy Juan:

BTW: his grandfather took that picture. Looks a bit dangerous, doesn't it? It probably was. But that's ok, there's like an **inch** of sand in case he fell. I need to talk to him ;-)

All things considered, awesome 9 years. I need to actually remember for the 10th anniversary.

Adds tarpitting to `qmail-smtpd`, which is almost mandatory for a modern large SMTP server.

Too bad it's not progressive tarpitting, but that's easy to fix :-)

Now all I need is a patch to make `qmail-smtpd` check local rcpt's validity and I'm completely trouble-free with it.

Yes, after months of doing nothing, I hve made some changes in [uRSSus](#), my RSS aggregator.

Basically, I removed the Qt MVC tree for feeds and replaced it with a old, reliable item-based tree.

And now, magically you can drop feeds into folders, sort feeds by unread count, and it all seems to work.

I am probably the only [uRSSus](#) user in the world, so I am not bothering with a release until I kick the beast into shape with at least some nice features, but it's something.

Since I did some neat coding on it yesterday and today, I decided it was a good moment to release [uRSSus](#) 0.2.11 into the wild.

It doesn't look much different from 0.2.9 but it works much, much better.

Gone are the crazy disappearing/duplicate feeds when dragging/dropping in the feed tree.

Gone is the broken update scheduler, and now feeds update smoothly.

Gone are the inconsistent displays (like All feeds <> Unread items)

Gone is the CPU guzzling

Gone seems to be the DB locking and IO churning.

Sadly, gone is the DB schema migration because I can't get miruku to work with current alchemy/migrate/Elixir. Since I don't remember if the DB schema changed since 0.2.9 this may be bad ... or not. Backup, friends. Export to OPML and back. Whatever, this is alpha stuff ;-)

So, head to the [uRSSus homepage](#) and take a look. Maybe you will like it!

I mean, if google can get it wrong, I have an excuse, right?

See how at the same time you see:

- At the left: "All items (8)"
- At the top-center: "8 new items"
- In the tree: 8 items
- In the list: **14** new items

Each of those numbers can get out of sync with each other if you do something the "wrong" way.

But what the heck, it seems as of [r678](#) urssus does it right. The list of unread posts even updates when you have it open adding the new posts in the right places without disturbing you, and keeping the right numbers everywhere, AFAICS!

So here, in uRSSUs:

- 13 unread articles in the "Unread Articles" item
- 13 in "All Feeds"
- 13 adding each folder
- 13 in the article list

Sadly, this is all post-0.2.11 release.

Reading DZone I ran into an interesting post titled [Screw all GUI builders](#) which advocates dropping all GUI builders and instead coding your UI by hand.

I always advocate using Qt [Designer](#) instead of coding by hand, so I wanted to see, even it's talking about Java, I thought, "why don't I feel that way"?

My conclusion? I don't see it because in PyQt we are just lucky because our tools *don't suck quite as much*.

It gives several arguments:

- You don't know how exactly the generated code works. You don't need to. You start not to care and GUI application development becomes a process of drawing and adding simple event handlers here and there.

To that, my reaction was yes, indeed I don't care, as long as it works, which it has 99.99% of the time. When it didn't, I *did* understand the generated code, though.

The reason for this will be more obvious once you see the code in both cases, I think.

- Most GUI builders force you to use single class for single window, so generated classes tend to have thousands of lines of code.

Hmmm... I really don't know what this means in context, but that's just my java ignorance. OTOH, yes, one class per window, but almost never *thousands* of lines of code.

For example, the UI for [uRSSus](#) main window is quite complex, and it is only 530 lines of code. Here's how it looks, so you see it's not a trivial window:

Indeed Designer's code is not meant to be modified. That's what inheritance is there for. Overload whatever you want changed. Maybe this is easier in PyQt because Python is more dynamic than Java? Not sure.

- GUI builders force you to use an IDE, mostly one you started coding with. So if you start with NetBeans, you most likely be forced to stay with it for the whole project.

Just not true for Designer. I can see how that would suck, though.

- The generated code is far from being optimal. It's not resize-friendly, not dynamic enough, it has many hard-coded values, refactoring is most likely impossible, because builder would not allow that.

Designer does generate resize-friendly dialogs if you use it correctly. The hard-coded values are runtime-editable, refactoring is pretty simple (take whatever you want, create a widget with it?)

And since I wanted to compare apples to apples...

Here is his dialog, done via PyQt and Designer:

No, I didn't bother inserting text in the dialog so it looks the same ;-)

The "close" button closes the window, the URL opens in your system browser.

The text widget actually supports a subset of HTML, so there is a valid HTML document there, instead of just plain text.

Also, another thing is ... Designer's (or rather pyuic's) generated code is straightforward stuff.

Here's the code, which I think is roughly equivalent to his, only it's just 123 LOC, instead of 286 (his generated version) or 279 (his hand-made version). And I didn't delete the comments.

If I were doing this for real, all widgets would have descriptive names instead of PushButton1 and whatever.

Also, it's i18n-ready, unlike the Java versions, unless I missed something.

You can also get [hawkscope.ui](#) from this site to play with, it's done with Designer from Qt 4.4.

```
\ :c:`# -*- coding: utf-8 -*-`
\n
\n :c:`# Form implementation generated from reading ui file 'hawkscope.ui'`
\n :c:`#`
\n :c:`# Created: Fri Feb 13 22:39:47 2009`
\n :c:`#      by: PyQt4 UI code generator 4.4.4`
\n :c:`#`
\n :c:`# WARNING! All changes made in this file will be lost;`
\n
\n :kn:`from` \ \ :nn:`PyQt4` \ \ :kn:`import` \ \ \ :n:`QtCore` \ \ \ :p:``,`` \ \`
```

```

\ :k:'class' \ :nc:'Ui_Dialog' \ :p:'(\ \ :nb:'object' \ :p:'):\
  \ :k:'def' \ :nf:'setupUi' \ :p:'(\ \ :bp:'self' \ :p:', \ \ :n:
    \ :n:'Dialog' \ :o:'. \ \ :n:'setObjectName' \ :p:'(\ \ :s:'"Dia
    \ :n:'Dialog' \ :o:'. \ \ :n:'resize' \ :p:'(\ \ :mi:'444' \ \ :p:
    \ :bp:'self' \ :o:'. \ \ :n:'verticalLayout_2' \ :o:'=' \ \ :n:
    \ :bp:'self' \ :o:'. \ \ :n:'verticalLayout_2' \ :o:'. \ \ :n:'se
    \ :bp:'self' \ :o:'. \ \ :n:'horizontalLayout_2' \ :o:'=' \ \ :n:
    \ :bp:'self' \ :o:'. \ \ :n:'horizontalLayout_2' \ :o:'. \ \ :n:
    \ :bp:'self' \ :o:'. \ \ :n:'label_2' \ :o:'=' \ \ :n:'QtGui' \ \
    \ :bp:'self' \ :o:'. \ \ :n:'label_2' \ :o:'. \ \ :n:'setPixmap' \
    \ :bp:'self' \ :o:'. \ \ :n:'label_2' \ :o:'. \ \ :n:'setAlignmen
    \ :bp:'self' \ :o:'. \ \ :n:'label_2' \ :o:'. \ \ :n:'setObjectNa
    \ :bp:'self' \ :o:'. \ \ :n:'horizontalLayout_2' \ :o:'. \ \ :n:
    \ :bp:'self' \ :o:'. \ \ :n:'verticalLayout' \ :o:'=' \ \ :n:'Qt
    \ :bp:'self' \ :o:'. \ \ :n:'verticalLayout' \ :o:'. \ \ :n:'setC
    \ :bp:'self' \ :o:'. \ \ :n:'label_3' \ :o:'=' \ \ :n:'QtGui' \ \
    \ :n:'font' \ :o:'=' \ \ :n:'QtGui' \ :o:'. \ \ :n:'QFont' \ \ :p:
    \ :n:'font' \ :o:'. \ \ :n:'setWeight' \ :p:'(\ \ :mi:'75' \ \ :p:
    \ :n:'font' \ :o:'. \ \ :n:'setBold' \ :p:'(\ \ :bp:'True' \ \ :p:
    \ :bp:'self' \ :o:'. \ \ :n:'label_3' \ :o:'. \ \ :n:'setFont' \ \
    \ :bp:'self' \ :o:'. \ \ :n:'label_3' \ :o:'. \ \ :n:'setObjectNa
    \ :bp:'self' \ :o:'. \ \ :n:'verticalLayout' \ :o:'. \ \ :n:'addV
    \ :bp:'self' \ :o:'. \ \ :n:'label_4' \ :o:'=' \ \ :n:'QtGui' \ \
    \ :bp:'self' \ :o:'. \ \ :n:'label_4' \ :o:'. \ \ :n:'setObjectNa
    \ :bp:'self' \ :o:'. \ \ :n:'verticalLayout' \ :o:'. \ \ :n:'addV
    \ :bp:'self' \ :o:'. \ \ :n:'gridLayout' \ :o:'=' \ \ :n:'QtGui' \
    \ :bp:'self' \ :o:'. \ \ :n:'gridLayout' \ :o:'. \ \ :n:'setObjec
    \ :bp:'self' \ :o:'. \ \ :n:'label_5' \ :o:'=' \ \ :n:'QtGui' \ \
    \ :n:'font' \ :o:'=' \ \ :n:'QtGui' \ :o:'. \ \ :n:'QFont' \ \ :p:
    \ :n:'font' \ :o:'. \ \ :n:'setWeight' \ :p:'(\ \ :mi:'75' \ \ :p:
    \ :n:'font' \ :o:'. \ \ :n:'setBold' \ :p:'(\ \ :bp:'True' \ \ :p:
    \ :bp:'self' \ :o:'. \ \ :n:'label_5' \ :o:'. \ \ :n:'setFont' \ \
    \ :bp:'self' \ :o:'. \ \ :n:'label_5' \ :o:'. \ \ :n:'setObjectNa
    \ :bp:'self' \ :o:'. \ \ :n:'gridLayout' \ :o:'. \ \ :n:'addWidge
    \ :bp:'self' \ :o:'. \ \ :n:'label_10' \ :o:'=' \ \ :n:'QtGui' \ \
    \ :bp:'self' \ :o:'. \ \ :n:'label_10' \ :o:'. \ \ :n:'setObjectM
    \ :bp:'self' \ :o:'. \ \ :n:'gridLayout' \ :o:'. \ \ :n:'addWidge
    \ :bp:'self' \ :o:'. \ \ :n:'label_9' \ :o:'=' \ \ :n:'QtGui' \ \
    \ :n:'font' \ :o:'=' \ \ :n:'QtGui' \ :o:'. \ \ :n:'QFont' \ \ :p:
    \ :n:'font' \ :o:'. \ \ :n:'setWeight' \ :p:'(\ \ :mi:'75' \ \ :p:
    \ :n:'font' \ :o:'. \ \ :n:'setBold' \ :p:'(\ \ :bp:'True' \ \ :p:
    \ :bp:'self' \ :o:'. \ \ :n:'label_9' \ :o:'. \ \ :n:'setFont' \ \
    \ :bp:'self' \ :o:'. \ \ :n:'label_9' \ :o:'. \ \ :n:'setObjectNa
    \ :bp:'self' \ :o:'. \ \ :n:'gridLayout' \ :o:'. \ \ :n:'addWidge
    \ :bp:'self' \ :o:'. \ \ :n:'label_8' \ :o:'=' \ \ :n:'QtGui' \ \
    \ :bp:'self' \ :o:'. \ \ :n:'label_8' \ :o:'. \ \ :n:'setObjectNa
    \ :bp:'self' \ :o:'. \ \ :n:'gridLayout' \ :o:'. \ \ :n:'addWidge
    \ :bp:'self' \ :o:'. \ \ :n:'label_7' \ :o:'=' \ \ :n:'QtGui' \ \
    \ :n:'font' \ :o:'=' \ \ :n:'QtGui' \ :o:'. \ \ :n:'QFont' \ \ :p:

```

```

\ :n:'font' \ \ :o:'. \ \ :n:'setWeight' \ \ :p:'(\ \ :mi:'75' \ \ :p
\ :n:'font' \ \ :o:'. \ \ :n:'setBold' \ \ :p:'(\ \ :bp:'True' \ \ :p
\ :bp:'self' \ \ :o:'. \ \ :n:'label_7' \ \ :o:'. \ \ :n:'setFont' \ \
\ :bp:'self' \ \ :o:'. \ \ :n:'label_7' \ \ :o:'. \ \ :n:'setObjectName'
\ :bp:'self' \ \ :o:'. \ \ :n:'gridLayout' \ \ :o:'. \ \ :n:'addWidget'
\ :bp:'self' \ \ :o:'. \ \ :n:'label_6' \ \ :o:'=' \ \ :n:'QtGui' \ \
\ :bp:'self' \ \ :o:'. \ \ :n:'label_6' \ \ :o:'. \ \ :n:'setOpenExtensio
\ :bp:'self' \ \ :o:'. \ \ :n:'label_6' \ \ :o:'. \ \ :n:'setObjectName'
\ :bp:'self' \ \ :o:'. \ \ :n:'gridLayout' \ \ :o:'. \ \ :n:'addWidget'
\ :bp:'self' \ \ :o:'. \ \ :n:'verticalLayout' \ \ :o:'. \ \ :n:'addLay
\ :bp:'self' \ \ :o:'. \ \ :n:'horizontalLayout_2' \ \ :o:'. \ \ :n:'ad
\ :bp:'self' \ \ :o:'. \ \ :n:'verticalLayout_2' \ \ :o:'. \ \ :n:'ad
\ :bp:'self' \ \ :o:'. \ \ :n:'label' \ \ :o:'=' \ \ :n:'QtGui' \ \ :
\ :n:'font' \ \ :o:'=' \ \ :n:'QtGui' \ \ :o:'. \ \ :n:'QFont' \ \ :p
\ :n:'font' \ \ :o:'. \ \ :n:'setWeight' \ \ :p:'(\ \ :mi:'75' \ \ :p
\ :n:'font' \ \ :o:'. \ \ :n:'setBold' \ \ :p:'(\ \ :bp:'True' \ \ :p
\ :bp:'self' \ \ :o:'. \ \ :n:'label' \ \ :o:'. \ \ :n:'setFont' \ \ :
\ :bp:'self' \ \ :o:'. \ \ :n:'label' \ \ :o:'. \ \ :n:'setObjectName'
\ :bp:'self' \ \ :o:'. \ \ :n:'verticalLayout_2' \ \ :o:'. \ \ :n:'ad
\ :bp:'self' \ \ :o:'. \ \ :n:'textBrowser' \ \ :o:'=' \ \ :n:'QtGui' \ \
\ :bp:'self' \ \ :o:'. \ \ :n:'textBrowser' \ \ :o:'. \ \ :n:'setObje
\ :bp:'self' \ \ :o:'. \ \ :n:'verticalLayout_2' \ \ :o:'. \ \ :n:'ad
\ :bp:'self' \ \ :o:'. \ \ :n:'horizontalLayout' \ \ :o:'=' \ \ :n:'
\ :bp:'self' \ \ :o:'. \ \ :n:'horizontalLayout' \ \ :o:'. \ \ :n:'se
\ :n:'spacerItem' \ \ :o:'=' \ \ :n:'QtGui' \ \ :o:'. \ \ :n:'QSpace
\ :bp:'self' \ \ :o:'. \ \ :n:'horizontalLayout' \ \ :o:'. \ \ :n:'ad
\ :bp:'self' \ \ :o:'. \ \ :n:'pushButton' \ \ :o:'=' \ \ :n:'QtGui' \ \
\ :bp:'self' \ \ :o:'. \ \ :n:'pushButton' \ \ :o:'. \ \ :n:'setObje
\ :bp:'self' \ \ :o:'. \ \ :n:'horizontalLayout' \ \ :o:'. \ \ :n:'ad
\ :bp:'self' \ \ :o:'. \ \ :n:'pushButton_2' \ \ :o:'=' \ \ :n:'QtGui' \ \
\ :bp:'self' \ \ :o:'. \ \ :n:'pushButton_2' \ \ :o:'. \ \ :n:'setObje
\ :bp:'self' \ \ :o:'. \ \ :n:'horizontalLayout' \ \ :o:'. \ \ :n:'ad
\ :bp:'self' \ \ :o:'. \ \ :n:'verticalLayout_2' \ \ :o:'. \ \ :n:'ad

\ :bp:'self' \ \ :o:'. \ \ :n:'retranslateUi' \ \ :p:'(\ \ :n:'Dialog
\ :n:'QtCore' \ \ :o:'. \ \ :n:'QObject' \ \ :o:'. \ \ :n:'connect' \ \
\ :n:'QtCore' \ \ :o:'. \ \ :n:'QMetaObject' \ \ :o:'. \ \ :n:'conne

\ :k:'def' \ \ :nf:'retranslateUi' \ \ :p:'(\ \ :bp:'self' \ \ :p:', \ \
\ :n:'Dialog' \ \ :o:'. \ \ :n:'setWindowTitle' \ \ :p:'(\ \ :n:'QtC
\ :bp:'self' \ \ :o:'. \ \ :n:'label_3' \ \ :o:'. \ \ :n:'setText' \ \
\ :bp:'self' \ \ :o:'. \ \ :n:'label_4' \ \ :o:'. \ \ :n:'setText' \ \
\ :bp:'self' \ \ :o:'. \ \ :n:'label_5' \ \ :o:'. \ \ :n:'setText' \ \
\ :bp:'self' \ \ :o:'. \ \ :n:'label_10' \ \ :o:'. \ \ :n:'setText' \ \
\ :bp:'self' \ \ :o:'. \ \ :n:'label_9' \ \ :o:'. \ \ :n:'setText' \ \
\ :bp:'self' \ \ :o:'. \ \ :n:'label_8' \ \ :o:'. \ \ :n:'setText' \ \
\ :bp:'self' \ \ :o:'. \ \ :n:'label_7' \ \ :o:'. \ \ :n:'setText' \ \
\ :bp:'self' \ \ :o:'. \ \ :n:'label_6' \ \ :o:'. \ \ :n:'setText' \ \
\ :s:''<html><head><meta name='\ \ :se:''" \ \ :s:'qrichtext' \ \ :se:''" \ \
\ :s:''p, li { white-space: pre-wrap; } \ \ :se:''n \ \ :s:'' \ \

```

```

\ :s: ""</style></head><body style='\ \ :se:"\ \ :s: ' font-family:\ \ :s:
\ :s: ""<p style='\ \ :se:"\ \ :s: ' margin-top:0px; margin-bottom:0px; ma
\ :bp: 'self'\ \ :o: '.'\ \ :n: 'label'\ \ :o: '.'\ \ :n: 'setText'\ \ :
\ :bp: 'self'\ \ :o: '.'\ \ :n: 'textBrowser'\ \ :o: '.'\ \ :n: 'setHtml
\ :s: ""<html><head><meta name='\ \ :se:"\ \ :s: ' qrichtext'\ \ :se:"\ \
\ :s: "p, li { white-space: pre-wrap; }'\ \ :se:"\n'\ \ :s: ""\
\ :s: ""</style></head><body style='\ \ :se:"\ \ :s: ' font-family:\ \ :s:
\ :s: ""<p style='\ \ :se:"\ \ :s: ' margin-top:0px; margin-bottom:0px; ma
\ :s: ""<p style='\ \ :se:"\ \ :s: '-qt-paragraph-type:empty; margin-top:0
\ :bp: 'self'\ \ :o: '.'\ \ :n: 'pushButton'\ \ :o: '.'\ \ :n: 'setText
\ :bp: 'self'\ \ :o: '.'\ \ :n: 'pushButton'\ \ :o: '.'\ \ :n: 'setShort
\ :bp: 'self'\ \ :o: '.'\ \ :n: 'pushButton_2'\ \ :o: '.'\ \ :n: 'setTex

```

Yay!

Announcing release 0.2.12 or [uRSSus](#) a RSS/Atom aggregator.

This release fixes the big crashy updater bug in 0.2.11, and makes some minor improvements, like nicer date display, and a xdg resource installer.

So all the latest releases of [uRSSus](#) have a horrible bug: if you had never used it, it would not work at all.

Good way to get new users, moron!

system-message

WARNING/2 in `everything.txt`, line 3
 Duplicate explicit target name: "release".

Thanks to a bug report on the Arch package, this is fixed in today's 0.2.13 [release](#).

system-message

WARNING/2 in `everything.txt`, line 20496
 Duplicate explicit target name: "release".

I just finished implementing my first public planet, [Planeta PyAr](#) using [rawdog](#). It contains the blogs of members of [Python Argentina](#)

To avoid the classical argument about whether this is "the planet of people in PyAr" or "posta about python from people in PyAr", I decided to hack rawdog into producing both.

I didn't want to have two separate rawdog configurations, so I started looking at plugins.

The basis was the [selectfeeds](#) plugin and the [rss](#) plugin.

Then I merged selectfeeds into rss to make things simpler.

Here is my [hacked version](#) which takes the selectfeed argument and uses only those feeds when creating the HTML, FOAF, OPML and RSS files.

The bad side of this is that selectfeed requires you post the feed URLs all in one line!

With just a couple dozen blogs in the planet, that means several hundred characters. And then when you decide to change one URL... ick.

So, I wrote a tiny script to take separate feed files and create separate config files, [merge-config.py](#).

Now, I create a feeds-full file with the non-python-specific feeds in rawdog format:

```
#####
# Feeds full
#####

# Anthony Lenton
feed 20m http://anthony.lenton.com.ar/feed/es/
# Rivendel
feed 20m http://www.ceportela.com.ar/?feed=rss2
# Facundo Batista
feed 20m http://feeds2.feedburner.com/com/wEER
:
:
```

And the same for a feeds-python:

```
#####
# Feeds solo python
#####

# Anthony Lenton
feed 20m http://anthony.lenton.com.ar/?feed=rss2&cat=4&language=es
# Rivendel
# No tiene RSS solo python
# Facundo Batista
# No tiene RSS solo python
# Gabriel Patiño
feed 20m http://gepatino.blogspot.com/feeds/posts/default/-/Python
:
:
:
```

Put all the common configurations in config-base.

And merge-config.py produces a config, a config-full and a config-python.

Then run rawdog like this:

```
cd ~/.rawdog ; LANG=es_ES rawdog -c config-full -w ; LANG=es_ES rawdog -c c
```

And that's it, two planets for the cost of one and a half. I wonder why other planets don't provide this.

I decided to add a manual for [uRSSus](#). Since it's a Qt app, I checked how to do one of those neat help files, like the ones that come with Qt apps.

It turns out it's [not so simple](#) to create one of those.

You need to create your help in one or more HTML files, then create a XML file that describes what each file is, references for each section, and references for each keyword you want in the index.

For any real-life-size document, that's going to be incredibly annoying.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "docutils".

So, I took my usual escape route when I don't want to do grunt work on docs: [Docutils](#).

Specifically, I wrote [rst2qhc](#) which takes one (or more) restructured text files, and creates a nice and clean Qt Help Project file from them, including section titles, references and keywords, which you mark on the text using the 'keyword' role.

What does this mean? Let me be graphical:

The one on the right is nice to read. The one on the left is nice to write. The arrow in the middle is [rst2qhc](#) :-)

And here's how the generated qhp file looks for a trivial outline of a manual (and the reason why I don't want to do this manually ;-):

Also, as a bonus, you can create PDF, LaTeX, HTML and ODT files from the same source (heck, you can create freaking man pages).

```
system-message
WARNING/2 in everything.txt, line 20611
Duplicate explicit target name: "docutils".
```

Today I woke up at 6AM. I had nothing to do until 9, but I woke up and couldn't sleep again.

So, what can I do? I hacked [rst2qhc](#) a bit more :-D

Since this is a small project, it is quickly approaching feature complete status (there is really only so much this thing can do).

So, what's new?

- Passing options to `rst2html`, so you can do things like set stylesheets.
- Support for a manifest file, so you can list what files should be embedded in the helpfile.
- Fixed project support. Now you can *optionally* ask `rst2qhc` to create a qhcp file that will build your document.
- Better, more complete example that displays most features.

I have only one thing left in my TODO, which is calling the HTML writer via API instead of via `system()` because that way there is no need to define the keyword role in the document, and I can add syntax highlighting via pygments.

Since I now *do* have work to do, I will just release 0.2 now ;-)

You can get it from the usual place: <http://rst2qhc.googlecode.com>

And here's a sample of the input/output:

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "planeta pyar".
```

I am using [rawdog](#) for [Planeta PyAr](#) and I am very happy with it. One thing I really didn't like was the templating.

```
system-message
WARNING/2 in everything.txt, line 20654
Duplicate explicit target name: "planeta pyar".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "1".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "2".
```

Why? It's basically undocumented, it looks ugly and it doesn't support template inheritance, which in this case is very useful, because I am actually doing two very similar planets: 1 2.

```
system-message
WARNING/2 in everything.txt, line 20658
Duplicate explicit target name: "1".
```

```
system-message
WARNING/2 in everything.txt, line 20660
Duplicate explicit target name: "2".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "mako".
```

So, since I saw a plugin to use Vellum templates, which means the templating is pluggable, why not use my favourite templating library ([Mako](#)) instead?

```
system-message
WARNING/2 in everything.txt, line 20664
Duplicate explicit target name: "mako".
```

It turns out to be *very* easy to do!

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "this".
```

Just put [this](#) in your plugins folder and start using Mako templates (yes, the planet's config and stuff is in [github](#). Why not?).

```
system-message
WARNING/2 in everything.txt, line 20672
Duplicate explicit target name: "this".
```

```

\ :c:'# -*- coding: utf-8 -*-\
\ :kn:'import'\ \ :nn:'rawdoglib.plugins'\

\ :kn:'from'\ \ :nn:'mako.template'\ \ :kn:'import'\ \ :n:'Template'\
\ :kn:'from'\ \ :nn:'mako.lookup'\ \ :kn:'import'\ \ :n:'TemplateLookup'\
\ :kn:'from'\ \ :nn:'mako'\ \ :kn:'import'\ \ :n:'exceptions'\

\ :k:'class'\ \ :nc:'MakoPlugin'\ \ :p:':\
  \ :k:'def'\ \ :nf:'__init__'\ \ :p:'(\ \ :bp:'self'\ \ :p:'):\
    \ :bp:'self'\ \ :o:'.''\ \ :n:'lookup'\ \ :o:'='\ \ :n:'TemplateLo
      \ :n:'output_encoding'\ \ :o:'='\ \ :s:'utf-
      \ :n:'encoding_errors'\ \ :o:'='\ \ :s:'repl

  \ :k:'def'\ \ :nf:'fill_template'\ \ :p:'(\ \ :bp:'self'\ \ :p:','\ \
    \ :bp:'self'\ \ :o:'.''\ \ :n:'lookup'\ \ :o:'.''\ \ :n:'put_string'\
    \ :n:'t'\ \ :o:'='\ \ :bp:'self'\ \ :o:'.''\ \ :n:'lookup'\ \ :o:'.''\
    \ :k:'try'\ \ :p:':\
      \ :n:'r'\ \ :o:'='\ \ :n:'t'\ \ :o:'.''\ \ :n:'render_unicode'\
    \ :k:'except'\ \ :p:':\
      \ :n:'r'\ \ :o:'='\ \ :n:'exceptions'\ \ :o:'.''\ \ :n:'html_err
    \ :n:'result'\ \ :o:'.''\ \ :n:'value'\ \ :o:'='\ \ :n:'r'\
    \ :k:'return'\ \ :bp:'False'\

\ :n:'p'\ \ :o:'='\ \ :n:'MakoPlugin'\ \ :p:'()\
\ :n:'rawdoglib'\ \ :o:'.''\ \ :n:'plugins'\ \ :o:'.''\ \ :n:'attach_hook'\ \
  \ :n:'p'\ \ :o:'.''\ \ :n:'fill_template'\ \ :

```

Yup, 20 lines of code!

There is a patch to check recipients... have to see if it works, of course :-)

The uRSSus doc is slowly growing, so I hooked assistant to the UI. Not difficult, but I have not seen it elsewhere.

Here's how:

Assume the "Handbook action" is called `action_Handbook`. Set `window.assistant` to `None` in `__init__`.

```

\ :k:'def'\ \ :nf:'on_action_Handbook_triggered'\ \ :p:'(\ \ :bp:'self'\
  \ :k:'if'\ \ :n:'i'\ \ :o:'=='\ \ :bp:'None'\ \ :p:':\ \ :k:'return

  \ :k:'if'\ \ :ow:'not'\ \ :bp:'self'\ \ :o:'.''\ \ :n:'assistant'\ \
    \ :ow:'not'\ \ :bp:'self'\ \ :o:'.''\ \ :n:'assistant'\ \ :o:'.''\ \

    \ :n:'helpcoll'\ \ :o:'='\ \ :n:'os'\ \ :o:'.''\ \ :n:'path'\ \ :o:
      \ :s:'help'\ \ :p:','\ \
      \ :s:'out'\ \ :p:','\ \
      \ :s:'collection.qhc'\ \ :p:')'\
    \ :n:'cmd'\ \ :o:'='\ \ :s:'assistant -enableRemoteControl -collec
    \ :bp:'self'\ \ :o:'.''\ \ :n:'assistant'\ \ :o:'='\ \ :n:'subproces
      \ :n:'shell'\ \ :o:'='\ \ :bp:'True
      \ :n:'stdin'\ \ :o:'='\ \ :n:'subpr

  \ :bp:'self'\ \ :o:'.''\ \ :n:'assistant'\ \ :o:'.''\ \ :n:'stdin'\ \ :o:

```

And that's it. Now I need to figure out context help.

I am finally publishing my LatinoWare 2008 tutorial, in revised and expanded form. It will probably be a 10-part series, and here is [session 1](#)

I am finally publishing my LatinoWare 2008 tutorial, in revised and expanded form. It will probably be a 10-part series, and here is [session 2](#).

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "session 1".

See also [session 1](#).

system-message

WARNING/2 in `everything.txt`, line 20748
Duplicate explicit target name: "session 1".

Following the path of the venerable `tinyurl`, <http://Tinyarro.ws> takes advantage of unicode to make things very, very, very short.

For example, this blog's URL would be <http://\protect\begingroup\immediate\write\@unused\def\MessageBreak\le>

As you can see there is a *tiny arrow* (thus the name of the service) and a seemingly korean character.

The domain name is legal, that's just a one-character unicode name, which seems to be legal according to the rules of the `ws` TLD. Then a unicode path, again just one character.

Since unicode has several thousand glyphs, it is possible that about a trillion URLs could be done using only a 3-letter path, which is pretty awesome.

Of course you probably can't **type** the damn things, but you can copy&paste them.

If any of this doesn't work for you, get on the unicode wagon already!

I am not sure if this is good or bad:

On one hand, I have more subscribers than ever, even considering when this blog was in [planetkde](#) (BTW: maybe I should add my `pyqt` feed there again? Nah, I don't qualify as "active KDE contributor")

The dip in the last three months was because I just posted nothing, and now I am posting again, it's doing well.

On the other hand, I am at 50-some subscribers, which is a bit pathetic for a blog that has existed for over 9 years ;-)

On the gripping hand (Larry Niven FTW!) I am having more fun with the blog than I had in quite a while, so I would post even if noone read it.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "this".

And in case you are wondering what that peak of over 6000 reach (the average is under 100!) it's [this](#). That post had over 12000 visitors. My second most popular story had only 3800.

Here's the most popular content in the last 2 years or so, selected from 783 posts (784 with this one) and 47 longer stories:

1. 12228 visitors: [Windows: My eXPerience](#)
So I wanted to see what windows looked like. Don't worry, I feel better now.
2. 3797 visitors: [Making Linux systems that don't suck. Part II](#)
A rant on cron and at. I never imagined this would be number 2.
3. 2968 visitors: [BOP: Ball Oriented programming](#)
I am rather proud of this one: a pyqt-based graphical, animated interpreter for [FLIP](#), a language based on balls :-)
4. 2888 visitors: [Good News: Linux gives life to old hardware. Bad News: Maybe in some cases it shouldn't.](#)
Silly, yes, but a cool picture :-)
5. 1870 visitors: [PyCells: The Python SpreadSheet redux](#)
My second or third attempt at writing a toy spreadsheet using python. I have been at it for about 5 years, apparently.
This is popular... even when the code it describes is based on a completely broken library!

6.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "squid authentication via pop or imap".
```

- 1841 visitors: [Squid authentication via POP or IMAP](#)
This was already about 4 years old when I started counting, so I have no idea how many visitors it *really* had. It is a handy script, I still use it sometimes!

7.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "the linux booting process unveiled".
```

- 1813 visitors: [The Linux Booting Process Unveiled](#)
I really expected this one to be much higher. It's even linked from wikipedia! Every day it has 2 or 3 hits. Then again, the first 3 years are not being counted ;-)
8. 1706 visitors: [Queue Management for Qmail](#)
While the tool it introduces is lame nowadays, the ideas are sound, and it explains a real problem.
 9. 1548 visitors: [Custom widgets using PyQt](#)
Very obsolete, noone should read that.
 10. 1505 visitors: [How to make your own distro in 3 not-so simple steps](#)
Oh, this one. It is **wrong**. It gives bad advice. Yet noone seems to notice ;-)
Not my best idea, not my best effort, still get email about it every month or so.

system-message

WARNING/2 in `everything.txt`, line 20840

Duplicate explicit target name: “the linux booting process unveiled”.

system-message

WARNING/2 in `everything.txt`, line 20842

Duplicate explicit target name: “squid authentication via pop or imap”.

system-message

WARNING/2 in `everything.txt`, line 20850

Duplicate explicit target name: “pycells”.

system-message

WARNING/2 in `everything.txt`, line 20854

Duplicate explicit target name: “windows”.

system-message

WARNING/2 in `everything.txt`, line 20856

Duplicate explicit target name: “this”.

I am finally publishing my LatinoWare 2008 tutorial, in revised and expanded form. It will probably be a 10-part series, and here is [session 3](#).

See also:

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “session 1”.

[session 1](#).

- [session 2](#).

system-message

WARNING/2 in `everything.txt`, line 20873

Duplicate explicit target name: “session 1”.

Yes, the blog was looking old and ugly. So I changed it. Hope you like the new style, it's based on [plaintxtBlog](http://demo.plaintxt.org/), from <http://demo.plaintxt.org/>

Not everything is adapted yet, but most places should look much better now.

The licensing of all original content has been changed to a Creative Commons license (non-commercial, share-alike, see the badge at the left).

Also, fewer ads, and a new Donate button I strongly encourage you to use ;-).

Oh, and I will try my best to also post the spanish translations of, at least, the posts. The code and long stories will probably stay english-only for a while.

I am finally publishing my LatinoWare 2008 tutorial, in revised and expanded form. It will probably be a 10-part series, and here is [session 4](#).

This session introduces “Actions”, which means we can now do menus and tool-bars.

See also:

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "session 1".

[session 1.](#)

- [session 2.](#)
- [session 3.](#)

system-message

- **WARNING/2** in `everything.txt`, line 20905
Duplicate explicit target name: "session 1".

Yesterday I was spammed by a new age portal called [Portalmico](#). The name is strange, since in spanish (their language) it means "Monkey Portal", but that the heck, after adding me to the Facebook group they wanted me to join, posting they are spammers, then leaving, I started to think...

What exactly bothers me about these guys? And decided to enter their portal and see what was awaiting me behind the gates.

Apparently these guys are heavily into [2012](#). However, they don't expect the end of the world, but a "cosmic awakening" and they call this event "important to the whole galaxy". This seems to be caused because of a peculiarity of the mayan calendar. Whoa. And I thought the guys at [Stop, unix time!](#) were overdoing it.

My views on religion

I am an atheist. I know it's more polite to say agnostic, but there are real problems with that concept. If we consider agnosticism to believe there is no evidence of god's existence one way or the other, then pretty much all catholics would have to declare themselves agnostics or heretics.

Why is that? Because religions requires faith. Knowledge banishes faith. If you know god exists, you can't have *faith* in god's existence. Faith is *defined* as believing *without proof* [\[5\]](#). So religion *requires* agnosticism. Yes, I know it makes no sense to you. That's your own problem. See here for a [counterargument](#).

On the other hand, I have been accused of not being tolerant to other's religious beliefs. That is not a fair accusation. I tolerate them. On the other hand, they often believe there is a place called "hell" where I will spend eternity in pain and suffering.

Mind you, I don't believe such a thing exists or that I will be there, but just for the sake of argument, let's say it does. That means they expect their god to torture me for eternity.

And I am being intolerant because I say they believe in childish fantasies? I see some unbalance here between what my alleged intolerance is, and their alleged tolerance.

If I am right, they are condescended to. If they are right, I burn in hell. On the other hand, if I am right, I will cease existing, so I can't gloat. If they are right, they can drink their ambrosia forever while I am tortured.

What sounds more tolerant to you?

One would be tempted to ask, if mayans were so awesome, how come their civilization was such a mess?[1]

But that's not it. That's not what bothers me. Brainless adoration of poor people who did their best to live their lives 600 years ago is not such a terrible thing, (even if it is not the smartest thing ever). What bothers me is the credulity.

Let's see...

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "karma?".

Karma?.. karma?: <http://www.portalmico.com/?p=244>

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "traditional reiki?".

Traditional Reiki?.. traditional reiki?: <http://www.portalmico.com/?p=229>

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "angels?".

Angels?.. angels?: <http://www.portalmico.com/?p=10>

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "classic platonism?".

Classic Platonism?.. classic platonism?: <http://www.portalmico.com/?p=184>

[2]

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "atlantis? lemuria?".

Atlantis? Lemuria?.. atlantis? lemuria?: <http://www.portalmico.com/?p=80>

[3]

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "reincarnation?".

Reincarnation?.. reincarnation?: <http://www.portalmico.com/?p=80>

[4]

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "a kid that spoke at 8 months, and psychologists(!?) say he was trying to read".

A kid that spoke at 8 months, and psychologists(!?) say he was trying to read people's auras?..
a kid that spoke at 8 months, and psychologists(!?) say he was trying to read people's auras?: <http://www.portalmico.com/?p=282>

You will be hard pressed to find a post there that's scored less than 8 in a 1/10 scale, so almost everyone seems to agree that all these things exist/ are true.

In fact, the only thing that came to mind that noone there seems to believe in is hidden messages in the Torah and bigfoot, but maybe I just didn't look correctly.

So, I think I now know what bothers me. The lack of taste. If you believe everything, you believe nothing in particular. If you accept everything, you are not looking for truth. If you hold all things equal, nothing is special.

I find the perspective of a world where everything can be true, everything has the same value, everything is *the same*... desperate.

These people sound desperate. They sound like they are adrift at sea and try to hold to any passing twig, be it a crystal, a drowned continent, or a charlatan.

Dear fellows at portalmico: the only thing that make life worth living is people. Yes, people. Not god, not mayan prophecies, not angels. People.

Go out, meet someone, have a nice life. What happens after that, if it is to happen at all, will do so anyway. Be good to people, and trust that god (if he exists) is worth anything and will welcome you to his best condo, because if he doesn't... well, he was not really worth adoring anyway.

So, I think I come out of this diving into nonsense with a clearer, if a bit sadder mind. Yes, they are spammers. Yes, they believe childish fantasies. I hope they get

better.

1

system-message
WARNING/2 in `everything.txt`, line 20985
Duplicate explicit target name: "1".

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

Come on, 80% of skeletons showing signs of severe anemia? That's **bad!** See [here..](http://www3.interscience.wiley.com/journal/120142003/abstract?CRETRY=1&SRETRY=0) here:

2

system-message
WARNING/2 in `everything.txt`, line 20989
Duplicate explicit target name: "2".

"This process allows us to bypass the illusion of our tridimensional reality..."

3

system-message
WARNING/2 in `everything.txt`, line 20991
Duplicate explicit target name: "3".

"... the lemurians lived in a frequency matching the Fifth Dimension (5D)..."

4

system-message
WARNING/2 in `everything.txt`, line 20993
Duplicate explicit target name: "4".

"...cleansing and clearing can include memories and traumas from previous reincarnations..."

5

system-message
WARNING/2 in `everything.txt`, line 20995
Duplicate explicit target name: "5".

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "dictionary".

From the [dictionary..](http://dictionary.reference.com/browse/faith?qsrc=2888) dictionary: <http://dictionary.reference.com/browse/faith?qsrc=2888>
: "belief that is not based on proof: He had faith that the hypothesis would be substantiated by fact." . At least I think they mean that definition of faith. I surely hope they are not saying they have "confidence" in god?

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

Thanks to Sebastián Bassi, there is now a spanish translation of the first part of the series. You can read it [here](#).

Also, because of a mistake I made, `todo.py` has lacked meaningful comments for a while. That's fixed now.

Also, there is a new [RSS feed](#) specifically for this tutorial series.

system-message

WARNING/2 in `everything.txt`, line 21014
Duplicate explicit target name: "here".

First footnote[#id3]_ Second footnote[#id4]_

Footnotes weren't broken, I was!

It's

.. [#]

Not

.. _[#]

And of course, you can only do this if you disable cheetah, which I hadn't done.

I planned to have it ready for today, but my current consulting job (migrating a firm from Exchange to FLOSS) is getting a bit gnarly (mostly because of Outlook 2000) (Yes, I know).

It may be done this saturday, maybe next tuesday, I just don't know yet.

xkcd 0xFF:

Figure 9: See this strip at [xkcd](#), the gretest comic... ever!

10

system-message

WARNING/2 in `everything.txt`, line 21019
Duplicate explicit target name: "id3".

First one

11

system-message

WARNING/2 in `everything.txt`, line 21021
Duplicate explicit target name: "id4".

Another one

Real life (he is not even 2 years old yet):

I need to find a **really** tall one in the next 4 years.

Ok, I will post this just in case some poor soul needs to try doing it. You tell me if it makes any sense.

- When installing Outlook 2000, you can choose support for “Groupware only”, “Internet only” or “both”.
- Internet Only lets you create IMAP or POP3 accounts, but no Exchange accounts.
- Groupware Only lets you use Exchange, but no IMAP or POP3.
- Both gives you only Exchange and POP3, but no IMAP.

Here is how you can have Exchange and IMAP at the same time:

- 1) Close Outlook.
- 2) Go to control panel.
- 3) There you have a “Mail” icon that was installed by Office.
- 4) Use that icon to create the IMAP account.

And some people *dare* say windows is easy to manage :-P

Yes, I know, old software, whatever. It’s not my fault if upgrading the freaking mail client costs so much money they won’t do it because they have to upgrade the whole puddlejumping office suite.

PS: Thanks to the guy that told me how it’s done ;-)

PS2: How to make Outlook show more than 100 results from LDAP: <http://support.microsoft.com/kb/262848>

Anyone actually **buying Tree**?

Because I just have to wonder: I am pretty sure I could clone it in a weekend.

It’s very basic, only for Mac...

And yet the author asks for \$40 for the full version. Tell you what, readers, pledge me \$400 (american dollars, that is ;-)) and I promise I can deliver:

- An app of similar features
- Open Source
- Multiplatform (Mac/Win/Linux at least)

In **one week**. While doing my job and taking care of my boy.

In fact, I don’t expect it to take more than one full day of coding, and 400 bucks is more than my usual rate anyway.

If I don’t release it, don’t give me the money and keep the code anyway, which would be in a public repo for progress monitoring.

I am finally publishing my LatinoWare 2008 tutorial, in revised and expanded form. It will probably be a 10-part series, and here is [session 5](#).

This session does heavy designer work, including promoted widgets, custom slots, and much more.

See also:

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “session 1”.

[session 1.](#)

- [session 2.](#)
- [session 3.](#)
- [session 4.](#)

system-message

- **WARNING/2** in `everything.txt`, line 21143
Duplicate explicit target name: “session 1”.

Doing a routine update in my desktop, I was surprised to see Qt 4.5 is now in extra!
Way to go [Arch!](#)

I had a bad start, because all fonts were broken, but after running `fc-cache -fv` as root and my user it looks ok again.

In fact, it fixes a [bug](#) that has been annoying me for a year.

Performance seems to be snappier. That may be a bit subjective, but things like the animations in krunner now are really quick and smooth while before they were jumpy.

And... I can try to make flash work in [uRSSus!](#) Yay!

UPDATE: it just started working:

I just saw [this post](#) about the [SheevaPlug](#). I need one. First I need a reason to need it.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “rdiff-backup”.

Backup server? Add a USB disk enclosure, put everything in a metal box, install [bacula](#) and-or [rdiff-backup](#)?

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “mpd”.

Same hardware, Home [MPD](#) server?

- Idem, home SAN?
- Portable demo server? I could install some solutions on SD cards and show them for clients with this and my netbook.

Suggestions accepted...

system-message

- **WARNING/2** in `everything.txt`, line 21184
Duplicate explicit target name: “mpd”.

system-message

WARNING/2 in `everything.txt`, line 21190

Duplicate explicit target name: "rdiff-backup".

The spanish translations of most sessions of "PyQt by Example" are almost done, thanks to several volunteers.

Tomorrow I will upload at least one.

If anyone wants to translate them to other languages, I will be happy to include those too.

I am currently in dire need of creating a nice admin page for a LDAP addressbook, which should do some slightly unusual things with the data (like manipulating Postfix's virtual table).

So, what the heck, let's also learn something new while I'm at it. The victims: [Pyjamas](#) and [Juno](#).

Pyjamas is sort of a Python version of [GWT](#) and is bliss. Finally I can code HTML and JS in python ;-)

And Juno is refreshingly simple. Since the whole frontend is done by Pyjamas, all I needed is a way to route [JSONRPC](#) calls to python code and operate in the backend.

So, here are two useful snippets:

```
\ :c:`# This decorator decodes JSONRPC arguments as sent by Pyjamas to`\
\ :c:`# Juno, and calls the target function with the decoded version.`\
\ :k:`def`\ \ :nf:`JSONRemote`\ \ :p:`(`\ \ :n:`target`\ \ :p:`)`:\
  \ :k:`def`\ \ :nf:`wrapper`\ \ :p:`(`\ \ :n:`web`\ \ :p:``,`\ \ :o:`*\`
    \ :n:`postdata`\ \ :o:`='\ \ :n:`web`\ \ :o:`.`\ \ :n:`input`\ \ :p:`
    \ :n:`data`\ \ :o:`='\ \ :n:`json`\ \ :o:`.`\ \ :n:`loads`\ \ :p:`
    \ :nb:`id`\ \ :p:``,`\ \ :n:`method`\ \ :p:``,`\ \ :n:`params`\ \
    \ :n:`kwargs`\ \ :p:`[\ \ \ :s:`method`\ \ :p:`]`\ \ :o:`='\ \ :n:`n
    \ :k:`return`\ \ :n:`target`\ \ :p:`(`\ \ :o:`*\` \ :n:`params`\ \
  \ :k:`return`\ \ :n:`wrapper`\
```

Using this, any plain junos method works as a JSONRPC method!

For example:

```
\ :nd:`@route`\ \ :p:`(`\ \ :s:`'/user'\ \ :p:`)`\
\ :nd:`@JSONRemote`\
\ :k:`def`\ \ :nf:`list`\ \ :p:`(`\ \ :n:`web`\ \ :p:``,`\ \ :n:`startswith
  \ :k:`try`\ \ :p:`:`\
    \ :n:`result`\ \ :o:`='\ \ :n:`search`\ \ :p:`(`\ \ :nb:`filter`\ \
    \ :n:`resp`\ \ :o:`='\ \ :n:`JSONResponse`\ \ :p:`(`\ \ :n:`result
  \ :k:`except`\ \ :n:`ldap`\ \ :o:`.`\ \ :n:`LDAPError`\ \ :p:``,`\ \ :
    \ :n:`resp`\ \ :o:`='\ \ :n:`JSONResponse`\ \ :p:`(`\ \ :bp:`None`\
  \ :k:`return`\ \ :n:`resp`\
```

See? No JSON decoding. And no encoding either, because I am creating the response using this:

```
\ :c:`# This class creates a correct JSON response from Juno as required by
```

```

\ :k: 'class' \ \ :nc: 'JSONResponse' \ \ :p: '(' \ \ :n: 'JunoResponse' \ \ :p: ')'
  \ :k: 'def' \ \ :nf: '__init__' \ \ :p: '(' \ \ :bp: 'self' \ \ :p: ', ' \ \ :n:
    \ :n: 'JunoResponse' \ \ :o: '.' \ \ :n: '__init__' \ \ :p: '(' \ \ :bp: 'se
    \ :bp: 'self' \ \ :o: '.' \ \ :n: 'headers' \ \ :p: '[' \ \ :s: 'Content-Ty
    \ :bp: 'self' \ \ :o: '.' \ \ :n: 'append' \ \ :p: '(' \ \ :n: 'json' \ \ :o:

```

This is probably not a great implementation but it's good enough for me right now.
 This is how you add basic authentication to your [Juno](#) app:

```

\ :n: 'init' \ \ :p: '{' \ \
  \ :s: 'middleware' \ \ :p: ': [' \ \ :s: 'paste.auth.basic.AuthBasicHandl
    \ :p: '{' \ \ :s: 'realm' \ \ :p: ': ' \ \ :s: 'Admin' \ \
    \ :s: 'authfunc' \ \ :p: ': ' \ \ :n: 'valid_user' \ \ :p:
  \ :p: '}' \ \

```

Where `valid_user` is the obvious function that takes `user/pass` and returns `true/false`.
 Not exactly a sophisticated role-based auth, but I don't need that for my app (LDAP's access controls take care of it).

After a while without any original content, here's a description of the Linux booting process (SystemV variant).

The idea was for it to be both simple and comprehensive. Let me know if it isn't!

Ok, here it goes. If you are not fluent in Qt programming the following entry will make no sense whatsoever :-)

If you set the contents of a `QTextBrowser`, and the content is HTML, it will call its `QMimeSourceFactory`'s `data()` method to get `MimeObjects` representing, for example, images in the HTML.

So far so good.

However, I am mystified by this: While it's doing that, you can change the contents, and it will go along it's merry way, except that old `data()` calls are still in progress.

In other words, it acts as if it's multithreaded. Usually, in Qt, this is because someone is calling `QApp::processEvents`, and then you have multiple functions running and yielding to each other in the Qt event loop.

That is surely what's happening, but... how the hell does one stop the old `data()` calls???

The `processEvents` call is not made by me, it's done internally by some method in the `QTextBrowser`... so, right now, in `KRsN`, sometimes you are downloading images for old stuff you saw minutes ago!

Thanks to Nicolás Pace, session 2 is now [available in spanish](#).

If anyone wants to translate these things to other languages, please, go ahead and contact me!

system-message

WARNING/2 in `everything.txt`, line 3
 Duplicate explicit target name: "available in spanish".

Thanks to Nicolás Miyasato, session 3 is now [available in spanish](#).

If someone wants to translate this to other languages, please contact me.

system-message

WARNING/2 in `everything.txt`, line 21309
 Duplicate explicit target name: "available in spanish".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “available in spanish”.

Thanks to Leonardo De Luca, session 4 is now [available in spanish](#)
Also, thanks to:

- ZeD who pointed out a bug in session 2 and wrote a sphinx config file (works very well!)
- Emanuele Rampichini who pointed out a bug (which I have not fixed yet)
- Simon Edwards who posted about this series in [his blog](#) and pointed out some differences if you want to use PyKDE instead of PyQt.

system-message

WARNING/2 in `everything.txt`, line 21324
Duplicate explicit target name: “available in spanish”.

Because of special circumstances, my boy had the whole cake-and-candles thing done three times in three days.

So here they are:

And now for something completely different:

In 1997 I was one of the founding members of a LUG, called LUGLi.

I left the group a few years ago, but I have rejoined their mailing list recently.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “rawdog”.

Since I officially Know Everything About [Rawdog](#) (TM) I offered to create a planet for member blogs. And here is [Planeta LUGLi](#).

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “planeta pyar”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “github”.

It’s heavily based on my work for [Planeta PyAr](#) and, just like it, its configuration and data is hosted in [github](#) in case of my sudden death (and because there is no point in making it difficult to help me ;-).

system-message

WARNING/2 in `everything.txt`, line 21356
Duplicate explicit target name: “github”.

system-message
WARNING/2 in `everything.txt`, line 21358
Duplicate explicit target name: “`planeta pyar`”.

system-message
WARNING/2 in `everything.txt`, line 21362
Duplicate explicit target name: “`rawdog`”.

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “`rst2pdf`”.

After an awfully long time, I set aside one hour for coding, and spent it on bugfixing for `rst2pdf`.

- Fixed Issue [112](#) with [r513](#)
- Fixed Issue [107](#) with [r516](#)
- Started implementing Issue [111](#)

Not much work (the fixes were very simple) but at least I am doing something. Important since some people actually use this program ;-)

system-message
WARNING/2 in `everything.txt`, line 21387
Duplicate explicit target name: “`rst2pdf`”.

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “`sphinx`”.

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “`rst2pdf`”.

Looks like someone is finally [working](#) on getting [sphinx](#) to generate PDFs using `rst2pdf`

Happy happy joy joy! Good Luck!

system-message
WARNING/2 in `everything.txt`, line 21396
Duplicate explicit target name: “`sphinx`”.

system-message
WARNING/2 in `everything.txt`, line 21398
Duplicate explicit target name: “`rst2pdf`”.

A bit of time today, too:

- Started looking at issue reported by msoulier (not reproduced yet)
- Implemented improvement requested in Issue [111](#) in [r520](#) (including docs and sample config file). Now it needs to be verified by [adk2012](#)
- Verified Issue [106](#) isn't happening anymore and closed it.
- Closed Issue [76](#) as WontFix because I am not keeping RL2.1 installed :-)
- Closed Issue [99](#) because it's fixed in wordaxe 0.3.2

UPDATE: The [changelog](#) in SVN is getting more than long enough to deserve a new release!

After 8 months (!) it's finally out: rst2pdf 0.10 ready for downloading from its [homepage](#).

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "restructured text".
```

If you don't know what I'm talking about, rst2pdf lets you create PDF files from a simple, very lightly marked pure text format called [restructured text](#) and is the easiest way to create nice documentation for your free software project and more!

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "this".
```

Here is the [manual](#) which is created using rst2pdf itself from [this](#).

```
system-message
WARNING/2 in everything.txt, line 21436
Duplicate explicit target name: "this".
```

```
system-message
WARNING/2 in everything.txt, line 21438
Duplicate explicit target name: "restructured text".
```

And here's a (maybe partial) list of the changes, which should convince you that you really want this version:

- Issue 87: Table headers can be repeated in each page (thanks to Yasushi Masuda)
- Issue 93: Line number support for code blocks (:linenos: true)
- Issue 111: Added --no-footnote-backlinks option
- Issue 107: Support localized directives/roles (example: sommaire instead of contents)

- Issue 112: Fixed crash when processing empty list items
- Issue 98: Nobreak support, and set as default for inline-literals so they don't hyphenate.
- Slightly better tests
- Background colors in text styles work with reportlab 2.3
- Issue 99: Fixed hyphenation in headers/footers (requires wordaxe 0.3.2)
- Issue 106: Crash on demo.txt fixed (requires wordaxe 0.3.2)
- Issue 102: Implemented styles for bulleted and numbered lists
- Issue 38: Default headers/footers via options, config file or stylesheet
- Issue 88: Implemented much better book-style TOCs
- Issue 100: Fixed bug with headers/footers and Reportlab 2.3
- Issue 95: Fixed bug with indented tables
- Issue 89: Implemented --version
- Issue 84: Fixed bug with relative include paths
- Issue 85: Fixed bug with table cell styles
- Issue 83: Fixed bug with numeric colors in backColor attribute
- Issue 44: Support for stdin and stdout
- Issue 79: Added --stylesheet-path option
- Issue 80: Send warnings to stderr, not stdout
- Issue 66: Implemented "smart quotes"
- Issue 77: Work around missing matplotlib
- Proper translation of labels (such as "Author", "Version" etc.) using the docutils languages package. (r473)
- Fixed problems with wrong or non-existing fonts. (r484)
- Page transition effect support for presentations (r423)

system-message

WARNING/2 in `everything.txt`, line 3
 Duplicate explicit target name: "rst2pdf".

So, `rst2pdf` 0.10 was not ok. The `setup.py` that worked on 0.9 simply didn't work for the new version.

Good thing is, Christoph actually **understands** `setuptools`, so he could fix it and deliver a neat **0.10.1** for your PDF-generating pleasure.

Post 0.10.1, I have already fixed one bug, Issue **117**.

system-message

WARNING/2 in `everything.txt`, line 21512
Duplicate explicit target name: "rst2pdf".

Do you miss the old LinuxToday.com of 4,5 years ago? Do you miss the old LWN.net of a 2 years ago?

Do you know what they had in common? Dave Whiting. Do you know what he's doing now?

Well, a Linux news site!

It's <http://www.lxer.com> and it's really **really** good.

For one thing, unlike LinuxToday.com, the page layout doesn't make you read the articles in a 2-inch strip between 6 inches of ads and boxes (But I still like LinuxToday, I just submitted an article there minutes ago! :-)

Dear <http://www.pythonprogramminglanguage.info/> I have no problem whatsoever with my blog being republished. I would prefer if you could link back here from somewhere, mmmmkay?

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "rst2pdf".

Since the last release, there have been *many* changes, improvements and fixes in `rst2pdf`.

system-message

WARNING/2 in `everything.txt`, line 21531
Duplicate explicit target name: "rst2pdf".

At the bottom of this post you will find (maybe incomplete) list, but I wanted to share a couple of relatively large advances.

Presentations

I have wanted to make `rst2pdf` useful for presentations for quite a while. And now it's getting close.

I am writing a small tutorial about it, but you can see a [demo](#) already.

The changes that make this possible are:

- Fixed bugs regarding page layout
- Added support for page backgrounds (including SVG!)

This was in large part because of [César Rodas](#), who wanted something simple and text-based to do his ZendCon presentation, so he asked me about it, and I was ashamed to give him a broken tool.

See how that works? I am driven by embarrassment. That's embarrassing.

Major stylesheet improvements

The related changes are way too many (see below), but it also happened for a reason: I wanted to make prettier documents.

So, since most people seems to like Sphinx's style, I wanted to look similar.

Did I achieve it? Yes and no, take a look:

But it's just a draft, I didn't look at fonts, or weight of the headings, or even at what pygments color scheme sphinx is using.

But anyway, even [this rough stylesheet](#) made me find a dozen bugs or problems in rst2pdf.

So, the main lesson here is, please tell me what you can and can't do with rst2pdf, because there are whole areas I am not even looking at, day-to-day.

The final thing is... the next release is going to be awesome. But then again, I always say that, don't I?

Changes as of today:

- Workaround for issue 103 so you can use borderPadding as a list (but it will look wrong if you are using wordaxe <= 0.3.2)
- Added fieldvalue style for field lists
- Added optionlist tstyle, for option lists
- Added collection of utility stylesheets and documented it
- Improved command line parsing and stylesheet loading (guess extension like latest rst2latex does)
- Fixed Issue 67: completely new list layouting code
- Fixed Issue 116: crashes caused by huge images
- Better support for %width in images, now it's % of the container frame's width, not of the text area.
- Fixed bug in SVG scaling
- Better handling of missing images
- Added missing styles abstract, contents, dedication to the default stylesheet
- Tables style support spaceBefore and spaceAfter
- New topic-title style for topic titles (obvious ;-)
- Vertical alignment for inline images (:align: parameter)
- Issue 118: Support for :scale: in images and handle resizing of inline images
- Issue 119: Fix placement of headers and footers
- New background property for page templates (nice for presentations, for example)
- Default to px for image width specifications instead of pt
- Support all required measurement units ("em" "ex" "px" "in" "cm" "mm" "pt" "pc" "%" "")

- New automated scripts to check test cases for “visual differences”
- Respect images DPI property a bit like rst2latex does.
- Issue 110: New --inline-footnotes option
- Tested with reportlab from SVN trunk
- Support for Dinu Gherman’s svglib. If both svglib and uniconvertor are available, svglib is preferred (for SVG, of course). Patch originally by rute.
- Issue 109: Separate styles for each kind of admonition
- For Issue 109: missing styles are not a fatal error
- Issue 117: TOCs with more than 6 levels now supported (raised limit to 9, which is silly deep)

It’s my pleasure to announce the release of rst2pdf version 0.11, available at <http://code.google.com/p/rst2pdf/download>. Rst2pdf is a tool to generate PDF files directly from restructured text sources via reportlab.

This version includes many bugfixes and some new features compared to the previous 0.10.1 version, including but not limited to embedding PDF images, much improved image sizing, nicer list layouts, better styling, page backgrounds, and more than 15 bugs fixed. A full changelog is included at the bottom of this message.

Rst2pdf aims to support the full restructured text feature set, and is very close to that goal, while also including some of the more experimental features, like a source code directive with syntax highlighting and math notation support with LaTeX-like syntax.

It supports embedding arbitrary fonts, both True Type and PS Type 1, both raster and vector images (including SVG), page transition effects, multiple, flexible page layouts, cascading styles, and much, much more.

You can find more information about rst2pdf in its home page (<http://rst2pdf.googlecode.com>), and ask anything you want in the rst2pdf- discuss mailing list (<http://groups.google.com/group/rst2pdf-discuss>)

Here are the changes in this version compared to 0.10.1:

- Degrade more gracefully when one or more wordaxe hyphenators are broken (currently DWC is the broken one)
- Fixed issue 132: in some cases, with user-defined fontAlias, bold and italic would get confused (getting italic instead of bold in inline markup, for instance).
- New stylesheet no-compact-lists to make lists... less compact
- SVG images now handle % as a width unit correctly.
- Implemented issue 127: support images in PDF format. Right now they are rasterized, so it’s not ideal. Perhaps something better will come up later.
- Fixed issue 129: make it work around a problem with KeepTogether in RL 2.1 it probably makes the output look worse in some cases when using that. RL 2.1 is not really supported, so added a warning.
- Fixed issue 130: use os.pathsep instead of “:” since “:” in windows is used in disk names (and we still pay for DOS idiocy, in 2009)

- Fixed issue 128: headings level 3+ all looked the same
- Ugly bugfix for Issue 126: crashes when using images in header + TOC
- New tstyles section in the stylesheet provides more configurable list layouts and more powerful table styling.
- Better syntax highlighting (supports bold/italic)
- Workaround for issue 103 so you can use borderPadding as a list (but it will look wrong if you are using wordaxe <= 0.3.2)
- Added fieldvalue style for field lists
- Added optionlist tstyle, for option lists
- Added collection of utility stylesheets and documented it
- Improved command line parsing and stylesheet loading (guess extension like latest rst2latex does)
- Fixed Issue 67: completely new list layouting code
- Fixed Issue 116: crashes caused by huge images
- Better support for %width in images, now it's % of the container frame's width, not of the text area.
- Fixed bug in SVG scaling
- Better handling of missing images
- Added missing styles abstract, contents, dedication to the default stylesheet
- Tables style support spaceBefore and spaceAfter
- New topic-title style for topic titles (obvious ;-)
- Vertical alignment for inline images (:align: parameter)
- Issue 118: Support for :scale: in images and handle resizing of inline images
- Issue 119: Fix placement of headers and footers
- New background property for page templates (nice for presentations, for example)
- Default to px for image width specifications instead of pt
- Support all required measurement units (“em” “ex” “px” “in” “cm” “mm” “pt” “pc” “%” “”)
- New automated scripts to check test cases for “visual differences”
- Respect images DPI property a bit like rst2latex does.
- Issue 110: New --inline-footnotes option
- Tested with reportlab from SVN trunk

- Support for Dinu Gherman’s `svglib`. If both `svglib` and `uniconvertor` are available, `svglib` is preferred (for SVG, of course). Patch originally by `rute`.
- Issue 109: Separate styles for each kind of admonition
- For Issue 109: missing styles are not a fatal error
- Issue 117: TOCs with more than 6 levels now supported (raised limit to 9, which is silly deep)

Yesterday I had one of those moments where I feel very happy about my distro of choice, [Arch Linux](#). Since the last time I posted about Arch seems to have been over [two years ago](#) (time flies when you are having fun!), I think it’s time to explain it.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "rst2pdf".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "docutils".
```

I wanted to test `rst2pdf` against `reportlab` from SVN, `wordaxe` from SVN and `docutils` from SVN, and I wanted it to be simple.

```
system-message
WARNING/2 in everything.txt, line 21780
Duplicate explicit target name: "rst2pdf".
```

```
system-message
WARNING/2 in everything.txt, line 21786
Duplicate explicit target name: "docutils".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "aur".
```

Solution: I just packaged them in [AUR](#)!

```
system-message
WARNING/2 in everything.txt, line 21790
Duplicate explicit target name: "aur".
```

Now, whenever I need to check `rst2pdf` against `wordaxe` trunk, I just need to `yaourt -S python-wordaxe-svn` and I can go back to stable `wordaxe` with `yaourt -S python-wordaxe`.

The `svn` package will always be the current trunk without any modifications, and I can switch back and forth in about 45 seconds, without messing up my system’s packages.

Also, I can keep my installed SVN packages updated by doing `yaourt -Su --devel` every now and then.

How would I have done that using Debian or a RPM distro? I suppose by going around the packaging system (which I hate) or by doing a private repo (which is so ... lame?) or by doing a public repo (which is freaking work).

Really, if you are a coder, I can't think of a Linux distro that makes life easier than Arch. Pretty much everything is there (12K packages in unsupported!) and if it isn't, it's a 5-minute job to slap it into AUR and help the community.

Suppose you are doing a KDE app. On most distros you need to install your own from-source copy of kdelibs to have the latest and make sure it's not screwed by distro-specific patches.

On Arch? Patching upstream is frowned upon. Not having the latest version is frowned upon. So it's pretty much the ideal environment to develop against KDE, or GNOME, or PyQt or whatever.

If my life was not 150% committed already, I would try to become an Arch developer, or at least a TU (Trusted User). Maybe next life!

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "yesterday".
```

My post about Arch Linux from [yesterday](#) got posted at [tuxmachines](#) and there was a comment there.

```
system-message
WARNING/2 in everything.txt, line 21813
Duplicate explicit target name: "yesterday".
```

I can't reply at the site because:

- It requires login.
- I can't find a place to get an account.
- It has **freaking google ads popups**

So I reply here because:

- Hey, it's my own blog, so I can do whatever I want.

Here's the comment by hussam with my (admittedly ranty) response:

I've been using ArchLinux as my only distribution/operating system since early 2006. It is really a good distribution but lately there have been a lot of really bad choices which I call bad compromises:

1. Too many ArchLinux users think gnome/kde are bloat and instead install some half developed window manager and some terminal emulator and call it a "minimalist" desktop.

Why is that any of your business? And what "compromise" is there?

2. Optional dependencies are the worst idea ever. If a package is linked against libsomething.so then libsomething should be a dependency not an optional dependency. Making libsomething an optional dependency just because “minimalist” users don’t want to install dependencies is plain stupid.

That’s not what optional dependencies are for. For example, consider the example I mentioned, rst2pdf. It can use pythonmagick. It can also not use it. You will lose one small feature that AFAIK only one person ever used. If you need that feature, the manual tells you what to do: install pythonmagick.

Maybe there should be a pacman option “install optdepends” which turns optional dependencies into regular ones. That would make you happy and keep others happy too.

3. Bad leadership. Aaron is fantastic guy but I know at least two Arch-Linux developers who can do a much better job.

That’s just stupid *and* mean.

4. Too many ArchLinux users now like badly automation scripts like yaourt or whatever it is called.

Parse error. And then again: yaourt is great. You don’t like it? Act as if it doesn’t exist and be happy. You seem to have a big problem ignoring people who disagree with you. That’s a really, really serious personal flaw. I suggest you grow up.

5. Too many noobs who do dumb things like people adding their users to hal, disk and dbus groups.

Sure, they should add themselves to optical and storage. So what? It’s a simple problem and it has a simple solution.

Then again, the adduser scripts probably should do that for regular user accounts. After all, who wants to create a regular user that *can’t use removable storage*? And if said use case exists, that should be doable by removing the user, and not viceversa!

On the other hand, I don’t give a damn, because I can fix it trivially.

The main reason why I don’t think I will switch to another distribution soon is that creating Archlinux packages from scratch is very easy and the initscript system is really fantastic.

All in all, ArchLinux is a really strong distribution now and it’s constantly growing.

I expect you, like most elitist poseurs, will run away when you feel Arch is too popular and accessible to “too many noobs” or some similar nonsense.

Which, like the title says, is why you are a big part of what’s wrong with Linux users.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “rst2pdf”.

As of today, [rst2pdf](#) supports Chris Liechti’s awesome [afigure](#).

system-message

WARNING/2 in `everything.txt`, line 21887
Duplicate explicit target name: “rst2pdf”.

What does it mean? It means you can put this in your document:

```
+-----+           +-----+
|   A   |----->|   B   |
+-----+           +-----+
```

And get this (and as a PDF drawing, **not** as an image, so you can copy the text from it):

So it's an awesome way to add simple diagrams to your restructured text documents that look good in plain text, on the web, and on PDF.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “here”.

Aafigure can create lots of interesting drawings, check their docs for examples [here](#)

system-message

WARNING/2 in `everything.txt`, line 21907
Duplicate explicit target name: “here”.

It currently requires aafigure from bazaar (0.3 won't work).

While randomly clicking today I saw [this article](#) with the fun title of “2009: software installation in GNU/Linux is still broken -- and a path to fixing it” by Tony Mobily.

As I don't have an account at FSM and don't intend to open one today, I will post a response here, where noone will ever notice it. Because that's how self-confident I am. I warn you, this response is full of proof-by-assertion and invalid generalizations, but hey, it's opinion.

It starts with an interesting claim:

And yet, software installation in GNU/Linux is broken. No, not broken... it's terribly broken. Why is that, and what can be done to fix it?

I intend to show how there is one little problem with that statement: reality. As in “what's the most successful software delivery platform for 'normal' users in 2009, and why that matters here” reality (hint: it's not windows, it's nothing on a Mac).

I mean the iphone, and its App Store. And what he is proposing is abandoning something that works *a lot* like it, and go back to the cave-age “download this from the app's site” model windows uses.

He describes the usual installation procedure, be it apt, yum or pacman, almost all distros do the same thing nowadays, then starts describing the problems.

- Users need to have root access in order to install a piece of software; no per-user installation is allowed

This would have been a problem in the old days of shared computers. Nowadays, you *are* root, or (in some cases) root’s mom. All modern distros use sudo so you don’t **need** root access, you just use your own password (or no password at all).

So, this is a non-issue.

- It’s very tricky to install several versions of the same piece of software. Just think of the poor graphic designer who needs to install several versions of Opera and Firefox; Users are stuck with the piece of software installed system-wide;

How do you install iFart 1.0 and 2.0 in an iphone? Can you? Special users (like graphics designers) should be provided by special solutions, like [portable firefox](#). Think there’s a need for it? Go ahead, it’s *maybe* one day of work.

- The software needs to be downloaded from the official repositories. Well, it doesn’t need to, but an average user wants to stay well away from unofficial repositories for technical reasons;

The iphone software needs to be downloaded from the app store. That has not hurt anyone (other than app developers, I mean)

- In some cases (especially when the user adds repositories or installs packages directly), the dependency-checking mechanism often fails and users end up with circular dependencies. They are nasty;

I have not seen a circular dependency since 2003. But hey, maybe they do exist. In that case, don’t do that.

- A piece of software is bound to a specific distribution, and — what’s worse — to a specific version of that distribution too. It’s not trivial to install Openoffice 3.1 on Ubuntu 8.10. You can argue that you can install the bunch of .deb packages from OpenOffice’s web site. Tell that to your grandmother or your average inexperienced computer user.

Some iphone apps will not work without the 3.0 firmware. Solution? Upgrade your firmware. How’s that different in Linux?

- It’s not trivial to “give” a program to a friend. To the end user, giving a program to a friend should be as simple as dragging an icon onto a memory stick; instead, files are scattered all over the system.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "rst2pdf".
```

It’s even simpler than that: here, let me give you [rst2pdf](#) for ubuntu: “Tony, install rst2pdf, it’s in ‘universe’”. I don’t even need to give you a floppy disk or whatever kids use these days.

Oh, and “how do you give someone an iphone app?”

Then, he proposes solutions, which I think are worse than the status quo:

- Users should be able to install software even without being root

Users are root, or root's mom. This is not a problem.

- Users should be able to install different versions of the same software immensely easily

The only way to do this is to make everything static or freeze the systems libraries. Do this for "portable" versions of specific apps if you want. Don't make the system suck, please.

- Users should be able to run either their own version of the software, or the one installed on the system (if any)

Fine by me. OTOH, how do you do that on an iphone? You don't. Please, stop diverging from what is working in 2009.

- It should be possible to download and install software even though it doesn't come from an official repository

Again, the iphone doesn't, and **noone cares**. If you care, you are not the normal user you claim to advocate.

- Software just needs to work — unchanged — even if it's a bit old and it's run on a newer distribution

Yes, [I want Quick Shot for iphones with 3.0 firmware!](#). Again, almost noone cares, except app developers.

You are acting like a **Linux** user, dude!

- It should be possible to "give" a piece of software to a friend by simply dragging an icon onto a memory stick

Just tell the guy the name of the app already!

All this is true with Apple's OS X. They got software installation just right — although a few programs, lately, seem to come with an ugly installation process.

Noone chooses a mac for this. They choose it because it's pretty, or because they are told they are better, or whatever, but anyone actually saying "hey, on a mac I can give you Office by sharing this 800MB dmg!"? Not only noone does it, it's freaking illegal.

Besides, you give the dmg to your friend, and 92% of the time, **your friend has windows or Linux** and can't use it. So your chance of success is about 8%.

```
system-message
```

```
WARNING/2 in everything.txt, line 3  
Duplicate explicit target name: "here".
```

Imagine if it only worked if your friend's computer was an Acer? There you would have a 10.5% chance! (see [here](#)) Noone would say that's **good**!

```
system-message
```

```
WARNING/2 in everything.txt, line 21999  
Duplicate explicit target name: "here".
```

No, sharin apps like that is not the problem.

Please, dear linux distro developers, look around you, and see what *works* in 2009.

Amazingly, what works looks an awful lot like what we have been enjoying since 1998 or whatever, so let's enjoy it and make it better, instead of trying to become like a mac, or like windows, old style platforms in the age of the smartphone and the netbook.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "reddit".

UPDATE: you can check more comments about this story in [reddit](#), [lxxr](#) and [OSNews](#).

system-message

WARNING/2 in `everything.txt`, line 22011
Duplicate explicit target name: "reddit".

system-message

WARNING/2 in `everything.txt`, line 22017
Duplicate explicit target name: "rst2pdf".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "pyar".

Some guys at [PyAr](#) are translating the [python tutorial](#) to spanish.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "sphinx".

That tutorial, like all the new python docs, uses [sphinx](#).

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "rst2pdf".

Since sphinx is very [popular](#), I want [rst2pdf](#) to work as well as possible as Sphinx's PDF producer. So, two different things itching, I started scratching.

Well, progress has been made. While the sphinx changes currently reside only on my computer, I will publish them soon, and the [rst2pdf](#) changes are already committed. Here's a [sample](#) to whet your appetite.

system-message

WARNING/2 in `everything.txt`, line 22032
Duplicate explicit target name: "rst2pdf".

system-message

WARNING/2 in `everything.txt`, line 22036

Duplicate explicit target name: “sphinx”.

system-message

WARNING/2 in `everything.txt`, line 22040

Duplicate explicit target name: “pyar”.

Ok, it *kinda* works *a little*.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “here”.

I got commit rights in the mercurial repo, so now you can get the very very alpha version [here](#).

This triggered several bugs in rst2pdf, so you need [SVN rst2pdf](#) for it to work.

What works:

- Glossary
- Plain restructured text ... with some caveats (example: footnotes are semi-broken)
- Index generation (not 100%, but maybe 75%)

I don't know what other pieces of sphinx require specific rst2pdf support, but I will be hitting them as they come.

system-message

WARNING/2 in `everything.txt`, line 22061

Duplicate explicit target name: “here”.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “here”.

Read all about it [here](#).

system-message

WARNING/2 in `everything.txt`, line 22066

Duplicate explicit target name: “here”.

You can read my latest article in [vietnamese](#) ... I think.

Too bad they removed my name from it. At least the link was kept. I am pretty liberal regarding reproduction of what I post, but please:

- Keep my name in it
- Put a link back

- Make changes clear

I just stumbled into this video in [The argentine post](#), which is a “documentary” from 1932 about traveling in Argentina.

There you can see: a bandoneón and guitar duo playing (for some reason) on a speeding boat.

Folkloric dancers in high heels. Very, very very bad ones, too, specially the women wearing said high heels.

A real life **crazy law**: coats were mandatory, so men wore pyjama coats in the streets (no, they were not mandatory **on the streets** although you wouldn’t get into almost any bussiness without one).

A man selling milk from a cow in the street. Riiiiight. Ok, that is *almost* plausible. Could happen sometimes.

The world’s worst mate, with about as much sugar as yerba (believe me, that’s not even good for tourists).

A comment like this “It is seldom that an argentine woman is troubled about the rights of her sex”.

If this was paid for by some tourism promotion agency: they really pushed it. If it was paid by MGM: crappy production values in the staged scenes, dudes!

The really really bad side of this kind of thing you can see in the comments by people that seem to be argentinians who say things like “It was such a country. What have they done with it?”.

You know what? Argentina in the 1930s **sucked** as a place to live in for most people. It was ok if you were a member of the Jockey Club, I suppose, but the average guy lived like crap.

Sure, the per capita income was high... compared to a world ravaged by the great depression, and even then 70% of the people lived in poverty. And trust me, poverty in the 30s was a bad place to be! Read some freaking [Arlt](#).

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “here”.
```

What was the life expectancy of the average argentine? How about 55? (see [here](#)) Yes, this “rich” place had a life expectancy 20 years lower than today.

```
system-message
WARNING/2 in everything.txt, line 22115
Duplicate explicit target name: “here”.
```

Of course you had almost no political rights because less than 35% of the population over 18 years old had the right to vote.

Those who did vote could just as well not bother, since there was rampant fraud.

There was no public health system.

There were no retirements. You worked until you could work no more, then you lived with your kids, if you had them. If you didn’t, they bad luck for you.

You lived under laws that would now seem draconian, the police could arrest you without cause and hold you almost forever.

But yeah, Palermo was a nice racetrack (Hell, it still is!) where the dissolute rich could waste the money they earned as absentee landlords of vast ranches, and you could swim in the Costanera (now polluted).

The gist? We are better off now. The average argentine citizen has more rights, lives better and longer than the average 1930s argentine.

The rest is touristy crap and false memories.

I am tired as a dog, back home after day 1 of PyCon Argentina 2009.

So, what happened:

- I was late with my brochures because the train got delayed (doors were not closing, so we had to change trains)
- Anyway, I got there kinda in time (about 30 minutes after registration started)
- Saw [Leito Monk's lecture](#)
- Saw [Alejandro Cura's lecture](#)

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "rst2pdf".

Saw the final printed version of [the python tutorial in spanish](#). He is experimenting with *very* low cost printing to use for educational purposes. It looks nice, and I am a very proud uncle, since it was typeset using [rst2pdf](#)!

I hope I can get a copy tomorrow!

- Then the lights went out, so had a coke with Leito
- Then lunch
- Saw Karucha's [Juegos educativos con PyGame](#)
- Grabbed a precious seat near a power outlet and worked in...
- My [A spellchecker in 5 minutes](#) lightning talk! Based in [this Norvig code](#)

I think it went fine. At least it lasted EXACTLY 5 minutes and covered all the slides :-)

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "rst2pdf".

Talked a minute with [Jacob Kaplan-Moss](#), told him I wrote [rst2pdf](#), he said he likes it (blush) and maybe something about using it to generate custom books to sell via [lulu.com](#)

system-message

- **WARNING/2** in `everything.txt`, line 22168
Duplicate explicit target name: "rst2pdf".

Now, to finish writing my slides for tomorrow!

Ha, I thought I was very tired after Day 1... not even close. Not after how tired I was after day 2!

First a short recap:

- Got there (in time!)
- Watched Facundo Batista's [Entendiendo Unicode](#)
- Finally presented [PyQt + WxPython + PyGtk](#). And here are my [slides](#).

A huge thank you to Mariano Reingart, Mariano Guerra, Daniel Moisset and Javier Castrillo who were there in representation of WxPython and PyGtk. It was a very cordial affair and lots of fun for everyone involved, I hope. Here are the [Wx](#) and [PyGtk](#) slides.

On the other hand, this kind of thing is dead as future talk subjects, at least to me. There isn't much sizzle in it anymore.

- Lunch
- A bit late for [Multiprocesamiento en Python](#) by Claudio Freire
- Then I completely didn't follow [Behind the scenes: Python bytecode](#) by Matías Bordese because...
- I had yet another lightning talk to give!

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "xkcd".
```

It was an old trusty friend of mine, Raymond Hettinger's [spreadsheet thingie](#) and I even did slides, there was even time to get a neat [xkcd](#) reference in it... and the netbook I was using crapped out in slide #3.

I could tell the joke (Marcos Dione had a shirt with the exact needed xkcd strip in the back), but I really had a thing going and .. well, it sucked.

What the heck, I will try again next time.

And yes, here are [the slides I couldn't show](#)

- Then, last conference, [Unladen Swallow](#) by Collin Winter, which was very nice.
- Then conference closing, gifts for the lucky, some talking, and home to rest.

In other areas:

I finally got my copy of the python tutorial printed with rst2pdf. It's the first time ever a book has been typeset using rst2pdf.

While the printing and binding are (intentionally!) cheap, the typesetting was pretty good. Here are some pictures:

And finally, I congratulate all the organizers, it was a beautifully handled event!

```
system-message
WARNING/2 in everything.txt, line 22260
Duplicate explicit target name: "xkcd".
```

Here I am speaking...
And here's the audience...
I really, **really** *really* hope they were laughing **with** me.
Pictures by [Beatrice Murch](#) licensed under [creative commons](#).

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

And one more. This was the first day, when I was speaking about Norvig's 20-line spellchecker (slides [here](#))
Photo by Kant (Ariel Kanterewicz), CC:BY

system-message
WARNING/2 in `everything.txt`, line 22307
Duplicate explicit target name: "here".

The latest iteration in "you know, a python spreadsheet makes sense", now from [Mission: Cognition](#).. `_mission: cognition: http://www.missioncognition.net/`
, with a cool idea he calls Code Objects.

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

Read it [here](#).

system-message
WARNING/2 in `everything.txt`, line 22320
Duplicate explicit target name: "here".

The guy at [Mission: Cognition](#).. `_mission: cognition: http://www.missioncognition.net/`
actually released his spreadsheet!
I havenot tested it, but it looks sweet.

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

See it [here](#)

system-message
WARNING/2 in `everything.txt`, line 22331
Duplicate explicit target name: "here".

Since this is the first post that appears in Planet Python, a small intro is in order.
Hi, my name is Roberto Alsina, I am a former KDE developer, amateur programmer [\[1\]](#)
, python fan.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “rst2pdf”.

This post (and many you will see from me) is about [rst2pdf](#), a tool to convert re-structured text to PDF. For example, it turns the python library reference into this [PDF](#) (not production quality yet).

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “rst2pdf”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “sphinx”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “docutils”.

If you are a [rst2pdf](#) 0.11 user, or a [sphinx](#) or [docutils](#) user who wants to try an alternative way to create PDFs, then I would like your help testing the SVN trunk code.

These are some of the new features since 0.11:

- PDFBuilder, a Sphinx extension to generate PDFs.
- Kerning (requires wordaxe from SVN)

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “afigure”.

[AAFigure](#) support (diagrams using ASCII Art)

- PageBreaks can be odd/even/any
- Support for advanced page counters (manual numbering, roman/arabic/letters)
- Much better styling, specially for tables.
- About two dozen bugfixes.

So, please, take a look, you may like it!

system-message

WARNING/2 in `everything.txt`, line 22362
Duplicate explicit target name: “sphinx”.

system-message

WARNING/2 in `everything.txt`, line 22364
Duplicate explicit target name: “docutils”.

system-message

WARNING/2 in `everything.txt`, line 22366
Duplicate explicit target name: “rst2pdf”.

system-message

WARNING/2 in `everything.txt`, line 22368
Duplicate explicit target name: “afigure”.

I never remember how to do footnotes right using ReStructured Text.
If that, or something similar, happens to you, get my [rst cheatsheet](#) and print it, it's a two-sided A4 page.

This is an early version, the left-hand columns need lots of love.

Also, you may want to take a look at this one-page [python cheatsheet](#), too.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “rst2pdf”.

Both are created using restructured text and [rst2pdf](#), and licensed under Creative Commons.

I would **love** to see translations to other languages. If you want to help, [contact me!](#)

system-message

WARNING/2 in `everything.txt`, line 22387
Duplicate explicit target name: “rst2pdf”.

I really, really hate to see myself on pictures, but my wife asks for evidence that I spent two days locked up with 400 nerds, so here they are:

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “pyqt.pdf”.

These are from the PyQt+WxPython+PyGtk panel ([pyqt.pdf](#).. [pyqt.pdf](#): <http://lateral.netmanagers.com.ar/static/pyqt.pdf>):

1

system-message

WARNING/2 in `everything.txt`, line 22370
Duplicate explicit target name: “1”.

That just means I love programming.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “`hoja.pdf`”.

And this one is from the failed lightning talk about Hettinger’s tiny “spreadsheet” ([hoja.pdf](http://lateral.netmanagers.com.ar/static/hoja.pdf).. `hoja.pdf`: <http://lateral.netmanagers.com.ar/static/hoja.pdf>):
All pictures by Ariel Kanterewicz (Kant) licenced CC-by.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “`thread`”.

There is a very funny `thread` currently in the PyAr (Python Argentina) mailing list. It all started when “Galileo Galilei” asked about how to do a very simple thing. He presented this code:

```
\ :n: 'age' \ \ :o: '=' \ \ :nb: 'int' \ \ :p: '(' \ \ :nb: 'raw_input' \ \ :p: '(' \ \
\ :k: 'if' \ \ :n: 'age' \ \ :o: '<' \ \ :mi: '18' \ \ :p: ':' \ \
\ :k: 'print' \ \ :s: 'You are underage' \ \
\ :k: 'else' \ \ :p: ':' \ \
\ :k: 'print' \ \ :s: 'You are a grownup' \ \
```

Ok, the original was... not quite as polite, but the code is the same. So far, nothing strange. But then he asked this:

How can I make it so that if the user enters something that is not a number, it does something like this:

```
\ :k: 'print' \ \ :s: 'I have no supercow powers' \ \
or maybe
\ :k: 'print' \ \ :s: 'Typing error' \ \
```

You can probably imagine that asking this kind of thing should produce maybe two answers. Right?

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “`facundo batista`”.

That is indeed the case, and you can see that in the answer by [Facundo Batista](http://mx.grulic.org.ar/lurker/message/20090916.112740.d31142f7.es.html).. `facundo batista`: <http://mx.grulic.org.ar/lurker/message/20090916.112740.d31142f7.es.html>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “`ezequiel`”.

or `Ezequiel`.. `ezequiel`: <http://mx.grulic.org.ar/lurker/message/20090916.125324.03d28e87.es.html>

Except... what if we wanted it to **keep asking** in case the user entered not-a-number?

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "fault".
```

Then, my friends... it's about **taste**, and it's all Juan Pedro Fisanotti's **fault**.. fault: <http://mx.grulic.org.ar/lurker/message/20090916.155741.32e208e7.es.html>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "my take".
```

Here's **my take**.. my take: <http://mx.grulic.org.ar/lurker/message/20090916.162906.2726401a.es.html>

:

```
\ :k: 'while' \ :bp: 'True' \ :p: ': ' \
  \ :n: 'edad' \ :o: '=' \ :nb: 'raw_input' \ :p: '(' \ :s: '¿Cuantos años' \
  \ :k: 'if' \ :n: 'edad' \ :o: '.' \ :n: 'isdigit' \ :p: '(): ' \
    \ :k: 'break' \
  \ :k: 'print' \ :s: 'No ingresaste un numero!' \
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "manuel aráoz".
```

Yes, I admit, a bit old fashioned. And there was a cry of “no, break sucks, it's not *right*”, which leads to this by **Manuel Aráoz**.. manuel aráoz: <http://mx.grulic.org.ar/lurker/message/20090916.173028.f>

```
\ :n: 'age' \ :o: '=' \ :nb: 'raw_input' \ :p: '(' \ :s: 'Your age?' \ \
\ :k: 'while' \ :ow: 'not' \ :n: 'age' \ :o: '.' \ :n: 'isdigit' \ :p: '()' \
  \ :k: 'print' \ :s: '"That's not a number!"' \
  \ :n: 'age' \ :o: '=' \ :nb: 'raw_input' \ :p: '(' \ :s: 'Your age?' \
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "(again by manuel aráoz)".
```

Which caused cries of “Having raw_input twice is ugly!”, which leads to (**again by Manuel Aráoz**).. (again by manuel aráoz): <http://mx.grulic.org.ar/lurker/message/20090916.174135.4ffa551.es.html>

:

```
\ :n: 'get_age' \ :o: '=' \ :k: 'lambda' \ :p: ': ' \ :nb: 'raw_input' \ \
\ :n: 'age' \ :o: '=' \ :n: 'get_age' \ :p: '()' \
\ :k: 'while' \ :ow: 'not' \ :n: 'age' \ :o: '.' \ :n: 'isdigit' \ :p: '()' \
  \ :k: 'print' \ :s: 'Not a number!' \
  \ :n: 'age' \ :o: '=' \ :n: 'get_age' \ :p: '()' \
```

Here Patricio Molina digs up [PEP 315](#).

And then Alejandro Santos says something like “This is easier in C, because we can assign a value to age *in the while’s condition*”. Please remember this.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “pablo zilliani”.
```

Now Pablo Zilliani.. pablo zilliani: <http://mx.grulic.org.ar/lurker/message/20090917.004645.ec6fc9c5.es.html>

gives his version, which is, I must say, perfect in some ways:

```
\ :n:'age' \ :o:'=' \ :n:'reset' \ :o:'=' \ :n:'msg' \ :o:'=' \
\ :k:'while' \ :ow:'not' \ :n:'age' \ :o:'. ' \ :n:'isdigit' \ :p:'(
  \ :n:'age' \ :o:'=' \ :nb:'raw_input' \ :p:'( \ :n:'msg' \ :p:
  \ :n:'msg' \ :o:'=' \ :s:'' \ :si:'%r' \ :s:' is not a number!,

\ :k:'print' \ :n:'age'
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “knuth”.
```

Here Gabriel Genellina decides to defend break by hitting everyone in the head using Knuth.. knuth: <http://mx.grulic.org.ar/lurker/message/20090917.040728.dc141f3c.es.html>

which should have a much stronger effect than Hitler.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “news”.
```

And now, we start veering into weird waters. Here is what news.. news: <http://mx.grulic.org.ar/lurker/message/2009>

proposes, which I must say, I admire... from a respectful distance.

First, the relevant code:

```
\ :n:'edad' \ :o:'=' \ :s:''0'' \ :c:'# Entra igual la primera vez'

\ :k:'while' \ :n:'firstTrue' \ :p:'( \ :ow:'not' \ :n:'edad' \ :o:
  \ :n:'edad' \ :o:'=' \ :nb:'raw_input' \ :p:'( \ :s:''¿Cuantos
  \ :k:'if' \ :ow:'not' \ :n:'edad' \ :o:'. ' \ :n:'isdigit' \ :p:
  \ :k:'print' \ :s:''No ingresaste un nro!''
```

But what, exactly, is firstTrue?

```

\ :kn:'import' \ \ :nn:'inspect' \

\ :k:'def' \ \ :nf:'firstTrue' \ \ :p:'(' \ \ :n:'cond' \ \ :p:'):' \
\ \ :sd:'""' devuelve True siempre la primera vez que se la ejecuta,
\ \ las veces subsiguientes evalua la condicion "" \
\ \ :n:'stack' \ \ :o:'=' \ \ :n:'inspect' \ \ :o:'. ' \ \ :n:'stack' \ \ :p:
\ \ :n:'line' \ \ :o:'=' \ \ :n:'stack' \ \ :p:'[' \ \ :mi:'2' \ \ :p:']' \
\ \ :k:'del' \ \ :n:'stack' \

\ :k:'if' \ \ :ow:'not' \ \ :s:'"line"' \ \ :ow:'in' \ \ :n:'firstTrue
\ \ :c:'# Primera vez que llamo a la funcion' \
\ \ :n:'firstTrue' \ \ :o:'. ' \ \ :n:'line' \ \ :o:'=' \ \ :n:'line' \
\ \ :k:'return' \ \ :bp:'True' \
\ :k:'elif' \ \ :n:'firstTrue' \ \ :o:'. ' \ \ :n:'line' \ \ :o:'!=' \ \ :
\ \ :c:'# Llame a la funcion desde otro punto del programa' \
\ \ :n:'firstTrue' \ \ :o:'. ' \ \ :n:'line' \ \ :o:'=' \ \ :n:'line' \
\ \ :k:'return' \ \ :bp:'True' \

\ :k:'return' \ \ :n:'cond' \

```

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "claudio freire's".

Then, I bring up generators, which leads to [Claudio Freire's..](http://mx.grulic.org.ar/lurker/message/20090917.182523.416ae5e3.es.html) [claudio freire's:](http://mx.grulic.org.ar/lurker/message/20090917.182523.416ae5e3.es.html)
<http://mx.grulic.org.ar/lurker/message/20090917.182523.416ae5e3.es.html>
, which **almost** works, too:

```

\ :n:'age' \ \ :o:'=' \ \ :s:'' \
\ :k:'def' \ \ :nf:'invalidAge' \ \ :p:'():' \
\ \ :k:'yield' \ \ :bp:'True' \
\ \ :k:'while' \ \ :ow:'not' \ \ :n:'age' \ \ :o:'. ' \ \ :n:'isdigit' \ \ :p:
\ \ :k:'print' \ \ :s:'"Not a number"' \
\ \ :k:'yield' \ \ :bp:'True' \
\ \ :k:'yield' \ \ :bp:'False' \

\ :k:'for' \ \ :n:'i' \ \ :ow:'in' \ \ :n:'invalidAge' \ \ :p:'():' \
\ \ :n:'age' \ \ :o:'=' \ \ :nb:'raw_input' \ \ :p:'(' \ \ :s:'"Age please'

\ :k:'print' \ \ :n:'age' \

```

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "fabian gallina".

And then [Fabian Gallina..](http://mx.grulic.org.ar/lurker/message/20090917.193059.0b80da01.es.htm) [fabian gallina:](http://mx.grulic.org.ar/lurker/message/20090917.193059.0b80da01.es.htm)
is the second one to bring up C's assignments inside conditions.

You know, I can't accept that. I will not accept C being easier for this.
So, with a little help from the python cookbook...

```

\ :n:'age'\ \ :o:'='\ \ :p:'[\ \ :mi:'1'\ \ :p:]'\
\ :k:'while'\ \ :ow:'not'\ \ :n:'age'\ \ :o:'|\ '\ \ :n:'asig'\ \ :o:'|\ '\
  \ :k:'print'\ \ :s:''Not a number!'\
\ :k:'print'\ \ :s:'u'You are '\ \ :si:'%s'\ \ :s:' years old'\ \ :o:'%'\

```

You may ask, what's |asig|? Glad you asked!

```

\ :k:'class'\ \ :nc:'Infix'\ \ :p:':'\
  \ :k:'def'\ \ :nf:'__init__'\ \ :p:'(\ \ :bp:'self'\ \ :p:','\ \ :n:
    \ :bp:'self'\ \ :o:'. '\ \ :n:'function'\ \ :o:'='\ \ :n:'function
  \ :k:'def'\ \ :nf:'__ror__'\ \ :p:'(\ \ :bp:'self'\ \ :p:','\ \ :n:
    \ :k:'return'\ \ :n:'Infix'\ \ :p:'(\ \ :k:'lambda'\ \ :n:'x'\ \
  \ :k:'def'\ \ :nf:'__or__'\ \ :p:'(\ \ :bp:'self'\ \ :p:','\ \ :n:'
    \ :k:'return'\ \ :bp:'self'\ \ :o:'. '\ \ :n:'function'\ \ :p:'(\
  \ :k:'def'\ \ :nf:'__rlshift__'\ \ :p:'(\ \ :bp:'self'\ \ :p:','\ \
    \ :k:'return'\ \ :n:'Infix'\ \ :p:'(\ \ :k:'lambda'\ \ :n:'x'\ \
  \ :k:'def'\ \ :nf:'__rshift__'\ \ :p:'(\ \ :bp:'self'\ \ :p:','\ \ :
    \ :k:'return'\ \ :bp:'self'\ \ :o:'. '\ \ :n:'function'\ \ :p:'(\
  \ :k:'def'\ \ :nf:'__call__'\ \ :p:'(\ \ :bp:'self'\ \ :p:','\ \ :n:
    \ :k:'return'\ \ :bp:'self'\ \ :o:'. '\ \ :n:'function'\ \ :p:'(\

\ :k:'def'\ \ :nf:'opasigna'\ \ :p:'(\ \ :n:'x'\ \ :p:','\ \ :n:'y'\ \ :
  \ :n:'x'\ \ :p:'[\ \ :mi:'0'\ \ :p:]'\ \ :o:'='\ \ :n:'y'\
  \ :k:'return'\ \ :n:'y'\ \ :o:'. '\ \ :n:'isdigit'\ \ :p:'()'\

\ :n:'asig'\ \ :o:'='\ \ :n:'Infix'\ \ :p:'(\ \ :n:'opasigna'\ \ :p:')'\

```

And then, Pablo posts this gem:

```

\ :kn:'import'\ \ :nn:'inspect'\

\ :k:'def'\ \ :nf:'assign'\ \ :p:'(\ \ :n:'var'\ \ :p:','\ \ :n:'value'\
  \ :n:'stack'\ \ :o:'='\ \ :n:'inspect'\ \ :o:'. '\ \ :n:'stack'\ \ :p:
  \ :n:'stack'\ \ :o:'. '\ \ :n:'f_locals'\ \ :p:'[\ \ :n:'var'\ \ :p:]'\
  \ :k:'del'\ \ :n:'stack'\
  \ :k:'return'\ \ :n:'value'\

\ :k:'while'\ \ :ow:'not'\ \ :n:'assign'\ \ :p:'(\ \ :s:''age''\ \ :p:','\
  \ :k:'print'\ \ :s:'u'Not a number!'\

\ :k:'print'\ \ :s:'u'You are '\ \ :si:'%s'\ \ :s:' years old'\ \ :o:'%'\

```

Which is, IMVHO, about as far from trivial as you can get here. Of course the thread is not dead yet ;-)

system-message
WARNING/2 in everything.txt, line 22643
Duplicate explicit target name: "thread".

Gentium. Very readable. Free usage.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “a post by andré roberge”.

Inspired by a post by André Roberge.. a post by andré roberge: <http://aroberge.blogspot.com/2009/09/live-preview->

I wanted to see if `rst2pdf` was too slow to be used for real-time previews in a restructured text editor.

It would also be very useful, for example, as a way to test stylesheet changes, making `rst2pdf` much more useful in general.

And after a couple of hours of gentle hacking, you know... it doesn't suck **at all**. I implemented the (still very primitive) PDF viewer using a python/poppler/Qt binding I found via google, the UI is PyQt.

Here's the video:

A note: the video was recorded using `qt-recordmydesktop` and that program is awesome. It was **trivial** to do.

I expect this will not be good enough when long documents are processed, but the `rst2pdf` manual (about 25 pages) renders in 5 seconds.

yak shaving (idiomatic) Any apparently useless activity which, by allowing you to overcome intermediate difficulties, allows you to solve a larger problem.

And boy, are my yaks hairy!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “rst2pdf”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “see here”.

I started trying to do a `rst2pdf` stylesheet editor ([see here](http://lateral.netmanagers.com.ar/weblog/2009/09/).. see here: <http://lateral.netmanagers.com.ar/weblog/2009/09/>).

One thing lead to another, and I have now at least three interesting mini-projects because of it.

Here's today's: abuse `pygments` to use it as a generic syntax highlighter in a Qt interface.

Why `pygments`? Because it's the only reStructured Text highlighter I found. That's probably because `reSt` is pretty damn hard to highlight!

AFAIK, this is the first time anyone has managed to use `pygments` for this, in an editable window. And there are good reasons for that:

- It's pure python, so maybe you expect it to be too slow
- It doesn't do partial or progressive lexing, so you need to lex the whole thing (again, maybe you expect it to be too slow)

- It has a file-oriented API, it generates a file with all the formatting in it, and for this kind of thing we need to access stuff in the middle of the data.

So, of course, it turns out it works pretty well, as you can see in this video:

Lesson learned: computers are **fast** nowadays.

Here's the code for [highlighter.py](#) with extensive comments.

You can just run it and get the same demo you saw on the video (minus the typing ;-)

```
system-message
WARNING/2 in everything.txt, line 22711
Duplicate explicit target name: "rst2pdf".
```

yak shaving (idiomatic) Any apparently useless activity which, by allowing you to overcome intermediate difficulties, allows you to solve a larger problem.

This yak is starting to look better.

For my second pile of yak shavings: turning [QPlainTextEdit](#) into a decent editing widget for programmers.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "rst2pdf".
```

As work advanced in my [rst2pdf](#) editor (BTW: need a name!), it became obvious that the piece of the UI the user will use most is just a couple of plain text editors.

Qt comes with a widget for that, of course, called [QPlainTextEdit](#). However, it's a very, very bad widget for programmers.

Here's the least I wanted:

- 1) Syntax highlighting for two languages: restructured text and javascript.
This yak is already [shaved](#).
- 2) Line numbers
- 3) Current line highlight
- 4) Error highlight when it makes sense (like, in the stylesheet)

One way to achieve this would be to dump [QPlainTextEdit](#) and use [QSciScintilla](#) which is the basis for the code editor in [eric](#) and (another version) in [scite](#).

However, I experienced a bad bug in [QSciScintilla](#), where I can't type accented characters! Without that, (decent) spanish is impossible, and the bug seems to be at least two years old, so... no go.

So, did I get those features? I say yes!

Here is the video (yes, I am getting addicted to making these, since qt-recordmydesktop makes them so easy ;-):

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "code editor example".
```

The basis for this is the [Code Editor example](http://doc.trolltech.com/4.5/widgets-codeeditor).. code editor example: <http://doc.trolltech.com/4.5/widgets-codeeditor>

that comes with Qt itself, plus a bit of my own handywork.

First, I ported Code Editor from C++ to Python, which was very simple and took a few minutes. That takes care of points 2 and 3.

Then, the syntax highlight was plugged in, which was point 1.

Then, how about realtime javascript validation? Easy using simplejson! Just make sure to run this whenever you want validation (I do it on every keystroke).

Replace `self.ui.style.toPlainText` with whatever your widget is called, of course:

```
\ :k: 'def' \ \ :nf: 'validateStyle' \ \ :p: '(' \ \ :bp: 'self' \ \ :p: ')': \
\ :n: 'style' \ \ :o: '=' \ \ :nb: 'unicode' \ \ :p: '(' \ \ :bp: 'self' \ \ :o:
\ :c: '#no point in validating an empty string' \
\ :k: 'if' \ \ :ow: 'not' \ \ :n: 'style' \ \ :o: '.' \ \ :n: 'strip' \ \ :p: '
\ :k: 'return' \
\ :n: 'pos' \ \ :o: '=' \ \ :bp: 'None' \
\ :k: 'try' \ \ :p: ':': \
\ :n: 'json' \ \ :o: '.' \ \ :n: 'loads' \ \ :p: '(' \ \ :n: 'style' \ \ :p:
\ :k: 'except' \ \ :ne: 'ValueError' \ \ :p: ', ' \ \ :n: 'e' \ \ :p: ':': \
\ :n: 's' \ \ :o: '=' \ \ :nb: 'str' \ \ :p: '(' \ \ :n: 'e' \ \ :p: ')': \
\ :k: 'print' \ \ :n: 's' \
\ :k: 'if' \ \ :n: 's' \ \ :o: '==' \ \ :s: 'No JSON object could be o
\ :n: 'pos' \ \ :o: '=' \ \ :mi: '0' \
\ :k: 'elif' \ \ :n: 's' \ \ :o: '.' \ \ :n: 'startswith' \ \ :p: '(' \ \ :
\ :n: 'pos' \ \ :o: '=' \ \ :nb: 'int' \ \ :p: '(' \ \ :n: 's' \ \ :o: '.'
\ :k: 'else' \ \ :p: ':': \
\ :k: 'print' \ \ :s: 'UNKNOWN ERROR' \

\ :c: '# This makes a red bar appear in the line' \
\ :c: '# containing position pos' \
\ :bp: 'self' \ \ :o: '.' \ \ :n: 'ui' \ \ :o: '.' \ \ :n: 'style' \ \ :o: '.' \ \
```

`highlightError(pos)` simply stores `pos` in the Code Editor, which will draw a red bar in that line, the same way it highlights the current line.

And that's it. Here is the code for [codeeditor.py](#)

system-message

WARNING/2 in `everything.txt`, line 22800
Duplicate explicit target name: "rst2pdf".

First: I really, really need a name for this thing. I am tired of saying "my rst2pdf previewer/editor app".

Now, here's a video of how it looks nowadays after all the yak shaving (sorry about my english, I **write** lots of it, but never **speak** it):

As you can see, the basic app is fairly complete, even if it lacks all the amenities that would make someone use it (like, search? save? ;-).

It has one **big** problem, though: I can't publish it yet.

Why? Because I need to use [poppler](#) from PyQt, and the code I found to do it has no license (see the code).

I am trying to contact the author (Rajeev J Sebastian), so there should be news soon!

As soon as that's cleared, the PDF widget is a whole post by itself, because it's pretty neat, if I may say so myself.

UPDATE: the binding is now under MIT license, thanks to Rajeev Sebastian!

I am in the middle of that honeymoon you get starting a new app. Every new feature seems to be just 50 lines of code away, there is no legacy code (in fact, you are creating that legacy code), and you learn new tricks all the time.

So, I did a new feature today.

A day or two ago, my editor started showing a yellow bar highlighting the current line.

But then I thought... wouldn't it be more useful to have a similar bar following you **in the PDF?**

That way, when you are on a given line, you can immediately see where you are in the output. Neat, right?

Here is a video showing it:

Sadly it's not perfect, and probably never will be because of docutils limitations, but it's pretty nice!

These are the visitors of this site in the last 10 minutes or so:

Let's see: Belgium, the Netherlands, USA, Turkey, Brazil, Thailand, Austria.

I know most people are not impressed by this kind of thing anymore, but I am old enough to have had to **buy** ftp sites on CD (Remember the Infomagic collection? I had like 10 of those).

In 1994 I was one of the first ten or so to have a mail account in my university (in fact, since I had root, I could have been the first one ;-)) and the web was a wasteland ... you could not even find **porn!**

So, knowing that my blog has *some* sort of a following, even if it's a handful of people and that it's close to being **10 freaking years old**, and it comes from a time when they were not even called blogs... is making me feel strange today.

This blog is one of the specific things I have done longer.

- I have been a husband for 4 years.
- I have been a dad for 3.
- I was active in KDE for 4.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "company".

My **company**.. company: <http://netmanagers.com.ar>

is about 2 years old

Of course there are obvious and/or fuzzy things I have done much longer:

- I have been involved in free software 15 years.
- I have worked with computers 18 years
- I have programmed 30 years (started on paper, kids!)
- I have inhaled and exhaled regularly for 38 years

But a specific thing, I can point at and say “that’s mine, I did it”? This blog is probably my longest-lived project.

I don’t see myself not having it. I see myself not posting for a month or so, I do that every now and then, but *not having a blog*? I can’t see it.

I have added things though, so it’s not a static deadend thing.

I am **loving** screencasts (thanks to qt-recordmydesktop and youtube). I enjoy twitter and identi.ca very much.

So, this rambling post is just my unfocused way of saying that what I love about having a blog is that someone reads it. So, let’s do something worth blogging about ;-)

Yes, the program known so far as “my rst2pdf editor/previewer application” is now called Bookrest.

What’s a bookrest? It’s a thing you put a book on.

Why Bookrest? I hope someday people will have books open in bookrest. Plus, it ends with “rest”, which is the preferred abbreviation for reStructured Text.

And what’s the outline view? It’s a clickable tree with the outline of the document, of course.

As usual, let’s go to the video:

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “multiprocessing”.

The background rendering was done using python’s awesome [multiprocessing](http://docs.python.org/library/multiprocessing.html)..

[multiprocessing](http://docs.python.org/library/multiprocessing.html): <http://docs.python.org/library/multiprocessing.html>

module.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “rst2pdf”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “rst2pdf”.

In my original post about it I was referring to Bookrest as a [stylesheet editor](http://rst2pdf.googlecode.com) for `rst2pdf`.. `rst2pdf`: <http://rst2pdf.googlecode.com>

, because that’s what I wanted, a way to test style changes and see what they did.

Of course, one thing lead to another and it’s starting to look more like a word processor than anything else, but ... well, how about a stylesheet editor?

Sure, you can use the “Style” tab, and edit at will, but that’s not exactly fun for everyone.

So, let’s work on one. Here’s the video of the current status:

Of course, this is about 1/20th of the stylesheet editor, but at least the dialog is there, and most of the remaining work is wiring dialogs, which is quick using designer.

Here’s what `fc-match` does:

```
$ fc-match "Droid Sans"  
DroidSans.ttf: "Droid Sans" "Regular"
```

Or even:

```
$ fc-match "Droid Sans" -v | grep file:  
file: "/usr/share/fonts/TTF/DroidSans.ttf"
```

So, how does one do that, going from a font family name or font name to a font *file*, where there's no fontconfig?

I found code for this in matplotlib's [font_manager](#) module but it looks hard to untangle, and requiring matplotlib is a bit over the top.

If there's no portable solution, I would be happy enough with three standalone solutions instead, and promise to publish an abstraction layer over them ;-)

So, dear windows and mac pythonistas, any pointers?

Thanks to Nicolás Pace, I got a video of the Pycon Argentina 2009 lightning talks.

So, after fighting my ignorance of video editing, here is my own talk, about Peter Norvig's spellchecker, spliced with the slides.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

If you want a better quality version of the slides, they are [here](http://lateral.netmanagers.com.ar/static/spellch).. here: <http://lateral.netmanagers.com.ar/static/spellch>

There is a problem in that no matter how I cut the original hour-long video, regardless of what tool I use, the sound gets out of sync, so I look kinda odd. I will try to do another version from the original video later on.

So, here it is:

Edited with kdenlive: crashy but cool and easy

While I won't claim to be a Ph D. in mathematics, and not event a self proclaimed Platonist Philosopher as [Dr. Hatcher](#)

But I know what's wrong with his proof.

While it doesn't fall into the infinite regress, it falls into something similar.

Since the universe is a composite and can not be the cause of its components, there must be an external cause... ok.

By definition, if there is the universe and Dr. Hatcher's G element, then there is a second metauniverse, containing G and our universe.

In that metauniverse, G can not be created by the metauniverse, since it's a component of it. So there must be a G' which created G.

If we decide, as he did, that G is self-created instead, why can't every quark be self-created?

Why can't there be just a dozen quarks, moving through time, and we see him many times because he crossed our position in time many times?

Are you willing to have a quark as your god? :-)

As usual, a video showing it:

The current status is that the infrastructure required for the stylesheet editor is in place:

- I figured out how to turn the StyleSheet object back into a JSON stylesheet.

- The changes are done and merged back into the same file.
- There is UI for 3 of the 6 required pieces. The missing ones are text styles, embedded fonts, and config options (like, break subsections to odd pages)
- The UI for the whole app is pretty rough, I am in full make-it-functional mode right now. It **will** be rethought later on.

It's my pleasure to announce the release of rst2pdf version 0.12, available at <http://code.google.com/p/rst2pdf/download>
 Rst2pdf is a tool to generate PDF files directly from restructured text sources via reportlab.

Rst2pdf aims to support the full restructured text feature set, and is very close to that goal, while also including some of the more experimental features, like a source code directive with syntax highlighting and math notation support with LaTeX-like syntax.

It supports embedding arbitrary fonts, both True Type and PS Type 1, both raster and vector images (including SVG and PDF), page transition effects, multiple, flexible page layouts, cascading styles, and much, much more.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "sphinx".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "sphinx".
```

This version includes many bugfixes and **MANY** new features compared to the previous 0.11 version, including but not limited to better styling, integration with [sphinx](http://sphinx.pocoo.org).. sphinx: <http://sphinx.pocoo.org>

, a very raw, preliminar graphical frontend called bookrest, kerning support (via wordaxe), and a much more powerful table implementation.

In fact, this release has so much new code it needs testers! I would consider it beta quality, and may (probably will) have some rough spots.

You can find more information about rst2pdf in its home page (<http://rst2pdf.googlecode.com>), and ask anything you want in the rst2pdf-discuss mailing list (<http://groups.google.com/group/rst2pdf-discuss>)

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "changelog".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "changelog".
```

A list of improvements and changes in this version is just too long, but is available at the [changelog](http://code.google.com/p/rst2pdf/source/browse/branches/0.12/CHANGES.txt).. changelog: <http://code.google.com/p/rst2pdf/source/browse/branches/0.12/CHANGES.txt>
 I hope you enjoy this program!

In other news, Google just found me two nice things related to rst2pdf:

1. A tutorial to use sphinx and rst2pdf for japanese texts: <http://d.hatena.ne.jp/MiCHiLU/20091009/1255065687>

2.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “pythonxy”.

It looks like a rst2pdf 0.11 binary for windows is shipped by pythonxy..
pythonxy: <http://code.google.com/p/pythonxy/>

here: http://code.google.com/p/pythonxy/downloads/detail?name=rst2pdf-0.11_py26.exe
I will link to that!

I love when people do things like these :-)

I just uploaded rst2pdf 0.12.1 to <http://rst2pdf.googlecode.com>

Rst2pdf is a tool to generate PDF files directly from restructured text sources via reportlab.

This release has no new features, just a few bugs fixed.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “changelog”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “changelog”.

For more information, see the [changelog](#).. changelog: <http://code.google.com/p/rst2pdf/source/browse/branches/0.12.1/>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “likes rst2pdf”.

In another rst2pdf-related note: someone at the Universitat Politecnica de Catalunya likes rst2pdf.. likes rst2pdf: <http://lsi.epsem.upc.edu/signatures/informatica-1/faq/com-i-on-he-dentregar-els-exercicis/>

:-)

I just uploaded rst2pdf 0.12.2 to <http://rst2pdf.googlecode.com> Rst2pdf is a tool to generate PDF files directly from restructured text sources via reportlab.

This release fixes a major bug (incompatibility with reportlab 2.1) as well as several minor ones and adds a minor feature (better styling of list items).

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “changelog”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “changelog”.

For more details, you can see the [changelog](#).. changelog: <http://code.google.com/p/rst2pdf/source/browse/branches>

Most people nowadays have more than one computer. Often, you are using one, and would like to do something in another. In this video, I will explain how trivial it is to do that without leaving your seat in a modern Linux using KDE.

We will use the following:

- Avahi, a zeroconf implementation to let you find your computers in your network without worrying about IP addresses, DNS, etc.
- krfb, the KDE Remote Frame Buffer. This is a program to share your desktop over the network.
- krdc, the KDE Remote Desktop Client, a VNC, RDP client, which is what you use to see a desktop shared via krfb.

I am sure users of other operating systems or desktop environments will say they can do it just as easily. In that case, feel free to do your own videos ;-)

Keep in mind that accessing remote desktops over the internet is a whole different beast, and this solution is not meant for that case.

As usual, this video was recorded using qt-recordmydesktop. There was minor editing using mencoder.

The computer used is the original Asus eee PC 701 4G, so you can see this is not exactly a hardware-intensive operation. I find the eee's small screen is great for this kind of full-screen demo, because it's not big enough to drown the important parts.

I have long known that application development is an arduous process. I have also long suspected one of the reasons it's arduous is the developer. I should be more specific, **I** am one of the reasons.

That's because I don't know what I am doing, and I don't mean that in the "I am a lame programmer" sense (even if that's also true somewhat), but in the sense that I **literally** don't know what the app should look like, or what its feature set should be.

So, I have decided to embark on a series of experiments I will call 24-hour apps.

Here are the rules:

- I shall create a neat application, stable, useful, usable and decent-looking.
- I shall do it in no more than 24 hours. After that time, it should be at least good enough for a preview release, if not a beta.
- Those 24 hours can be split in two or three sessions
- Time spent doing icons, docs, etc, counts.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "github".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "github".

All development shall be public (I am using [github](#).. [github: http://github.com/ralsina/24-hour-apps](http://github.com/ralsina/24-hour-apps)
)

- I **must** have a use for the resulting application, and it should be at least an adequate solution for that problem.

So, what's the first project? I call it Die Schere (The Scissors in German) and it's a video editor.

It's not a Kdenlive replacement, it's just the video editor I wish I had when I needed to glue a piece of one video with a piece of another.

In the old, pre-digital world, that was done using a cutter and scotch tape. I want Die Schere to be as useful and comprehensible as that was, but useful for clumsy people like myself.

Here is a video after today's session, which lasted 8 hours:

The basic functions are there, even if lots of work is still needed.

- You can load clips to work with them
- You can cut clips (like using a cutter!)
- You can choose the cut points interactively or by editing a time
- You can arrange them (like using scotch tape!)
- You can generate the output video

As a backend it's using mencoder, but there's no reason it shouldn't work with ffmpeg or melt if someone writes 20 lines of code.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "sabotaged from the future".

You may have heard about the possibility that the Large Hadron Collider is being [sabotaged from the future](http://io9.com/5380647/is-the-large-hadron-collider-being-sabotaged-from-the-future..).. sabotaged from the future: <http://io9.com/5380647/is-the-large-hadron-collider-being-sabotaged-from-the-future..>

That may sound preposterous, but it really is quite reasonable, once you think about it a little.

Let's start with the obvious:

Suppose you travel back in time and kill your grandfather. Then you wouldn't exist. And you couldn't travel back in time. And that's a paradox.

Then there is the anti-paradox:

Suppose you can't travel back in time. Because you didn't save him, your grandfather died. Ergo you don't exist.

That's not a paradox, but suppose time travel is possible: that means you can go back in time, save your grandfather, and thus make your own existence possible, ergo allowing you to save him.

As soon as you allow for time traveling to be possible, the solution for paradoxes is obvious: you can't change the past because the future won't let you. The future "faction" that tries to preserve the past will **always** win, because the past is fixed.

So, if you travel back in time and try to kill your grandfather by shooting him in the head, another person from the future will appear behind you and disarm you. Or appear behind you and kill you as soon as you had the idea to travel in time in the first place.

Or your kid will drop a coke bottle on your time machine just before you used it to kill his great-gramps.

The exact mechanism used to prevent you from becoming a paradox is unknown to you because you need to go further “forward” to know it.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “drops a piece of bread in the machinery”.

Now, back to the LHC: if in the future there is knowledge about time travel, and there is knowledge about how the LHC may destroy the world[1]_ then maybe some time tourist decided to see how humanity dressed in the pre-ice-age era, comes to 2009 and while eating a sandwich, somehow spooks a bird which then [drops a piece of bread in the machinery..](http://www.theregister.co.uk/2009/11/05/lhc_bread_bomb_dump_incident/)
drops a piece of bread in the machinery: http://www.theregister.co.uk/2009/11/05/lhc_bread_bomb_dump_incident/

Which of course, brings me to the best time-travel show on TV: “Mickey Mouse Clubhouse”.

If you don’t have an under-4 kid, grandson or nephew, you probably don’t know it, but let me give you a summary of **each darn episode**.

- 1) Something “bad” happens (Goofy has a cold)
- 2) A solution is proposed (Give Goofy some Minniestrone soup)
- 3) A flying ideogram of a mouse called Toodles shows a collection of tools:
 - Giant candy cane
 - Roller skates
 - Picnic basket
 - Seesaw
- 4) Something else “bad” happens (Pete wants the soup)
- 5) The tools are exactly what’s needed to solve the problems presented to accomplish the proposed solution (You can use the seesaw to move giant rocks blocking your path).

This is **obviously** a case of the future preventing something from happening. How else would Toodles know what tools to choose *before they are needed?* There was no way to guess that a giant rock would block the ravine through which Mickey and Minnie would try to escape from Pete![\[2\]](#)

So, whenever you need to think about paradox prevention, remember Mickey and his friends and just call Toodles.

1

system-message

WARNING/2 in `everything.txt`, line 23249

Duplicate explicit target name: “1”.

No, I don’t think the LHC would destroy the world.

2

As a former future mathematician, it drives me mad when I see an ad that looks like a great opportunity... as long as you don't crunch the numbers.

Not that they are hard to crunch at all, in fact they are pretty much pre-crunched, but consider this one:

Frávega, a large electronics/appliances store is advertising that in november, for one day, EVERYTHING WILL BE FREE.

That's right, for one day in november, they will not charge one cent for anything. Everything has a price tag of \$0.

Of course we don't know what day that is. It will be decided randomly after the month ends.

And... ok, and you don't get the money back, you get a voucher for the same amount, and you can use it in Frávega.

So, while this may look like a big promotion, it's actually doubtful Frávega will spend any money whatsoever in it, except the money spent printing the banners.

How much does this cost?

Assuming a random day, and that this campaign brings no new sales whatsoever, it's 1/30th of the monthly sales.

Also, they are not giving you money, they are giving you store credit. Since you can only spend it at Frávega, in reality will be getting a "2x1" deal. You spent in november, got the second one for "free" in december.

So, what is **really** happening is that 30 people spent, say, \$1000 each in an oven, and Frávega has to give them 31 ovens.

That would mean this is the equivalent of a 3.33% discount.

What? A 3.33% discount doesn't sound so *appealing*? Of course not! They are taking advantage of maths not being intuitive.

So, always keep it in mind: whenever you see numbers in advertising, they are there to convince you to buy. More often than not, it will *not* be as good a deal as it seems. This is an extraordinary case in that "one day everything for free!" actually means "a 3.33% discount in average for a month!".

BTW: the cost for frávega if you win is more probably less than \$600 for a \$1000 oven:

Frávega buys two ovens (let's be generous and say they paid \$1800 for them?) and gets \$1000.

But the \$800 deficit goes against earnings tax, which is 35% so \$280 come back.

And if they bought the ovens for \$900 each, that means \$314 are VAT, and since they are selling you two ovens for \$1000, there's a VAT refund of \$140.

So, those \$1000 actually are only \$580 after taxes.

This part about taxes can be terribly wrong, though ;-)

I will be doing a brand-new never seen introduction to PyQt programming at the "Jornadas de Software Libre y Open Source" in Mar del Plata tomorrow or the next day.

More info at <http://softwarelibre.mdp.utn.edu.ar/>

If you mention this blog and ask nicely, you get a can of cheap national beer tomorrow night (limit: 2 cans ;-)

```
system-message
```

```
WARNING/2 in everything.txt, line 23251
```

```
Duplicate explicit target name: "2".
```

Also, that part shows a complete ignorance of how leverage works.

No, this is not a post announcing I just wrote my first public python code. This is a post about my first public python code... from 1996!

In 1996, the soon-to-be-here year of the Linux desktop was fueled by one of the marquee open source applications of the time: LyX.

LyX was (is) a sort of word processor where you wrote and generated LaTeX which then produced whatever you used to print. But I am digressing: LyX was cool because it used one of the first *good free* graphical toolkits: XForms.

Ok, it was not really free, because you couldn't distribute patches.

And it was not all that good either, but we were comparing it with Motif, so it was much more free and much better than *that* monstrosity.

BTW: The latest release of XForms is from august of 2009.

At the time, a 25-year-old me was in love with Python 1.3. Here's how I described it:

Python 1.3 It's a beautiful ,free, language. Get it from <ftp://ftp.python.org>
<http://www.python.org>

Yes, Python 1.3. So, I wanted to use this C GUI toolkit used in this cool app, and this neat language I was learning and use them at the same time.

I ran (not walked) to my faithful *Slackware 3.0 ELF* in my 486DX2 PC and started hacking. In a weekend or so I had a working binding.

I even started writing the holy grail of desktop applications, a GUI version of Pine, using python and its IMAP module (python mailer, or PyM):

I released version 0.1 alpha in 1996, May 13 ... and a few months later Matthias Ettrich started KDE and I found Qt and never thought about XForms again.

Until this month.

For reasons that don't matter, I mentioned PyM in the PyAr mailing list the other day, and ... well, would pyxforms still work?

Why, pretty much, yeah!

I got the pyxforms-0.1-alpha sources from somewhere in the internet, installed XForms 1.0.92sp2 (yes, the latest release, from three months ago), of course I already had python 2.6.4 installed, added a setup.py, edited 10 lines of code and...

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

Yes, it works. You can get this 0.2 version (codename "Cthulhu was here") [here](#) just 13 years after 0.1.

No, I don't understand the weird rounded corners, or why the cursor looks weird and old when it's inside the window.

It's a REALLY small and fast toolkit, though.

Honestly, is it useful for anyone? Almost certainly not. Am I amazed something I wrote in 1996 still works? Oh, yeah I am.

system-message

WARNING/2 in `everything.txt`, line 23363
Duplicate explicit target name: "here".

Good article at IBM Developerworks.

However, I would use `dcop "konq*"` instead of `dcop konq*`, or else stuff may fail :-)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here it is".

This is not really part of my PyQt by Example series but since it's a totally unrelated topic that would be impossible to connect to it, but is still a PyQt tutorial and shares the concept so, [here it is...](#) here it is.: <http://lateral.netmanagers.com.ar/stories/BBS53.html>

I have been a customer of Fibertel for my broadband needs for several years. Their service has not sucked bad, even if I had a few problems in the past.

However, a month ago, I decided to switch to a competing service for a good reason: *it's much much much cheaper*[\[1\]](#).

Since I am not stupid, I canceled *late* with Fibertel, so I would have a full month of overlapping service. I thought hey, maybe I would come back if the new service sucks[\[2\]](#).

What happened next has been increasingly annoying.

First Fibertel called the person who contracted the service (not me). That's reasonable.

Then they called me to ask me why I was cancelling. I explained that the other service is (all things considered) about 75% cheaper for the same performance. I suppose that's reasonable too.

What started becoming less and less reasonable are the next 5 calls, each offering cheaper rates, but still more expensive than my new service.

What was not reasonable was that they started calling to tell me technically unsound bullshit to scare me into their service again.

What is definitely not reasonable is that they called me 6 times, even after I told them to stop calling me in the third call.

What is incredibly *stupid* and *annoying* was that they decided to block my internet access in order to show me an ad about the same offers I *ALREADY DECLINED* and until I clicked that ad I could not use the Internet.

What is *freaking nuts* is that because I have an access point I couldn't see the satan-damned ad, so I was left without internet until:

- I removed the access point
- Hooked my notebook to the cable modem
- Got rejected by it
- Hooked back the AP
- Got the MAC from it
- Faked my notebooks MAC
- Clicked the Cthulhu-pissed ads
- Hooked back the AP
- Un-faked my MAC

So, I needed to spend an hour doing this.

Therefore, dear Fibertel, your company also provides cable TV. I am pissed at you. I have decided to buy service from DirecTV *exclusively* to piss you off.

Yup, I, who have been paying you every month \$230 for years, will pay you \$0 next month. I could have paid you \$100, but no, you had to be asses about it. You couldn't let me go. Well, I do not like being pestered. I do not like that you acted like a spurned 17-year old[3].

Fibertel, I kinda liked you, but my opinion of you has gone lower.

Also, any Fibertel customers out there: **if you ask to cancel the service saying you switch because of the price, they will offer you 3Mbps for a year for \$79. If you say no, they will offer you 3 free months.**

That's a \$600 saving, so go ahead and take their money.

1

```
system-message
WARNING/2 in everything.txt, line 23428
Duplicate explicit target name: "1".
```

\$130 a month versus \$79 a month for the same bandwidth, plus it includes phone service with free local phone calls.

2

```
system-message
WARNING/2 in everything.txt, line 23430
Duplicate explicit target name: "2".
```

Since it still may suck, I am not saying who they are, they have not earned an endorsement yet.

3

```
system-message
WARNING/2 in everything.txt, line 23432
Duplicate explicit target name: "3".
```

Would you go out with me? How about we go see Twilight? No? How about we go to the fair? How about I pick you tonight? How about if I ask my dad for the car? How about if I pay dinner? -- Fibertel: that whiny teenager is you.

I wrote a while ago a RSS program called [uRSSus](#). I expect I am the only user of it because it has some problems (all of them my fault ;-)) but *I* really like it.

For a while now I have wanted it to have podcast support. The thing is... that always seemed like a lot of work. Sure, using phonon I can create an audio player and everything, but...

I am using a HTML widget to display the posts, so I would have to find a way to add the audio player to the UI and ... too much work.

So, today I woke up and thought... wait a minute... Qt's HTML widget is based on Webkit. And Webkit supports HTML5. And HTML5 has an "audio" tag.

So, if I fixed [uRSSus](#) to fetch the enclosure links, and added them in the database, and then added this to the post template:

```
<?py for enclosure in post.enclosures: ?>
  <audio autobuffer="Yes" controls="controls" src=#{enclosure.href}></audio>
<?py #end ?>
```

Wouldn't that actually work? Well, yeah!

So there you have it, I was right not to implement it, because the easiest way is to let Qt do it ;-)

How to write a tiny python app (less than 50 lines) that reacts to events on the DBUS buses. For example, displaying a notification when you press one of your keyboards' special keys.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "read all about it".

[Read all about it..](http://lateral.netmanagers.com.ar/stories/BBS54.html) read all about it: <http://lateral.netmanagers.com.ar/stories/BBS54.html>

No, not unique in the sense "oh, this app is a special snowflake", but unique in the sense "you can only run one copy of this application".

I tried googling for it and I always found the same answer, "use dbus, try to own the name, if it exists already, then a copy is already running".

What I could not find is one working example of this, or at least not something conveniently labeled "here is how you do a unique application using dbus and python".

So, here is how you do a unique application using dbus and python:

Supposing your application is called `uRSSus` (mine is):

```
\ :n:'session_bus'\ \ :o:'='\ \ :n:'dbus'\ \ :o:'.'\ \ :n:'SessionBus'\ \
\ :k:'try'\ \ :p:':'\
  \ :n:'session_bus'\ \ :o:'.'\ \ :n:'get_object'\ \ :p:'('\ \ :s:'"org.u
  \ :c:'# This is the second copy, make the first one show instead'\
  \ :c:'# TODO: implement'\
\ :k:'except'\ \ :n:'dbus'\ \ :o:'.'\ \ :n:'DBusException'\ \ :p:':'\ \ :
  \ :c:'# This will 'take' the DBUS name'\
  \ :n:'name'\ \ :o:'='\ \ :n:'dbus'\ \ :o:'.'\ \ :n:'service'\ \ :o:'.
  \ :c:'# Now, start your app:'\
  \ :n:'window'\ \ :o:'='\ \ :n:'MainWindow'\ \ :p:'()'\
  \ :nb:'object'\ \ :o:'='\ \ :n:'UrssusServer'\ \ :p:'('\ \ :n:'window
  \ :p:':'\
  \ :p:':'\
```

```
\ :p: '\ \
\ :n: 'etc' \ \ :p: ', '\ \ :n: 'etc' \
```

And that's it. No, it's not hard, but since the DBUS docs seem to be... rather they seem almost *not to be* sometimes, every little bit may help.

First a short explanation:

24-hour apps are small, self-contained projects where I intend to create a decent, useful application in 24 hours. The concept is that:

- a) I will think about this app a lot for a while
- b) I will design it in my head or in written notes
- c) I will code, from scratch, for 24 hours.
- d) That's not one day, really, but 24 hours of work. I can't work 24 hours straight anymore.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "last time around".

The [last time around](http://lateral.netmanagers.com.ar/weblog/2009/10/23.html#BB845).. last time around: <http://lateral.netmanagers.com.ar/weblog/2009/10/23.html#BB845>

this didn't quite work as I intended, but it was fun and educational (for me at least ;-)) and the resulting app is really *not bad*!

So, what's foley going to be? A note-taking app aimed at students and conference public.

In your last geeky conference, did you notice *everyone* is using a computer?

And what are they taking notes on? Vi? Kwrite? OpenOffice? Whatever it is they use, it's not meant to be used for this purpose.

So, what will foley do different? I don't quite know yet, but I have some ideas:

- 1) A *strong* timeline orientation. Every paragraph will be dated.
- 2) Twitter/Identica support. Want to liveblog your notes? Just click.
- 3) Multimedia incorporated in the timeline.
 - Webcam/Audio recording synced to your notes?
 - Images imported and added in the timeline?
 - Attach files to the timeline? (Useful for slides?)
- 4) If provided with a PDF of slides, attach each slide to the right moment in the timeline
- 5) Easy web publishing: find a way to put this on a webpage easy and quick (single-click publishing is the goal)

I have only thought about this for about 10 minutes, but I see potential here.

The bad news is... I have a ton of paying work to do. So this will probably only happen in January. However, I wanted to post it so I can take input while in this planning phase.

So, any ideas?

Introduction

If you are a Haloscan user, and are starting to wonder what can you do... this page will explain you a way to take your comments to Disqus, another free comment service.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "haloscan".
```

A few days ago, [Haloscan.. haloscan: http://haloscan.com](http://haloscan.com)
announced they were stopping

```
system-message
WARNING/2 in everything.txt, line 23556
Block quote ends without a blank line; unexpected unindent. backrefs:
```

their free comment service for blogs. Guess what service has in it the comments of the last 9 years of this blog? Yes, Haloscan.

They offered a simple migration to their Echo platform, which you have to pay for. While Echo looks like a perfectly nice comment platform, I am not going to spend any money on this blog if I can help it, since it already eats a lot of my time.

Luckily, the guys at Haloscan allow exporting the comments (that used to be only for their premium accounts), so thanks Haloscan, it has been nice!

So, I started researching where I could run to. There seems to be two large free comment systems:

```
system-message
• WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "disqus".
```

[Disqus.. disqus: http://disqus.com](http://disqus.com)

```
system-message
• WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "intense debate".
```

[Intense Debate.. intense debate: http://intensedebate.com](http://intensedebate.com)

Keep in mind that my main interest lays in not losing almost ten years of comments, **not** on how great the service is. That being said, they both seem to offer roughly the same features.

Let's consider how you can import comments to each service:

- Disqus: It can import from blogger and some other hosted blog service. Not from Haloscan.
- Intense Debate: Can import from some hosted services, and from some files. Not from the file Haloscan gave me.

So, what is a guy to do? Write a python program, of course! Here's where Disqus won: they have a public API for posting comments.

So, all I have to do then is:

- 1) Grok the Disqus API
- 2) Grok the Haloscan comments file (it's XML)
- 3) Create the necessary threads and whatever in Disqus
- 4) Post the comments from Haloscan to Disqus
- 5) Hack the blog so the links to Haloscan now work for Disqus

Piece of cake. It only took me half a day, which at my current rates is what 3 years of Echo would have costed me, but where's the fun in *paying*?

So, let's go step by step.

1. Grok the Disqus API

```
system-message
```

```
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "disqus python client library".
```

Luckily, there is a reasonable [Disqus Python Client library](http://code.google.com/p/disqus-python-client/).. `disqus python client` library: <http://code.google.com/p/disqus-python-client/>

```
system-message
```

```
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "docs for the api".
```

and `docs for the API`.. `docs for the api`: <http://groups.google.com/group/disqus-dev/>
so, this was not hard.

Just get the library and install it:

```
hg clone https://IanLewis@bitbucket.org/IanLewis/disqus-python-client/
cd disqus-python-client
python setup.py install
```

```
system-message
```

```
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "pybloxsom import script".
```

The API usage we need is really simple, so study the API docs for 15 minutes if you want. I got almost all the tips I needed from this [pybloxsom import script](http://freshfoo.com/blog/disqus-import-script).. `pybloxsom import script`: <http://freshfoo.com/blog/disqus-import-script>

Basically:

1. Get your API Key
2. You login
3. You get the right "forum" (you can use a disqus account for more than one blog)
4. Post to the right thread

2. Grok the Haloscan comments file

Not only is it XML, it's pretty simple XML!

Here's a taste:

```
\ :cp: '<?xml version="1.0" encoding="iso-8859-1" ?>'\
\ :nt: '<comments>'\
  \ :nt: '<thread'\ \ :na: 'id='\ \ :s: '"BB546"'\ \ :nt: '>'\
    \ :nt: '<comment>'\
      \ :nt: '<datetime>'\ 2007-04-07T10:21:54-05:00\ :nt: '</datetime>'\
      \ :nt: '<name>'\ superstoned\ :nt: '</name>'\
      \ :nt: '<email>'\ josje@aaaaaa.nl\ :nt: '</email>'\
      \ :nt: '<uri></uri>'\
      \ :nt: '<ip>'\ 86.92.111.236\ :nt: '</ip>'\
      \ :nt: '<text>'\ \ :cp: '<![CDATA[that is one hell of a cool website
    \ :nt: '</comment>'\
    \ :nt: '<comment>'\
      \ :nt: '<datetime>'\ 2007-04-07T16:14:53-05:00\ :nt: '</datetime>'\
      \ :nt: '<name>'\ Remi Villatel\ :nt: '</name>'\
      \ :nt: '<email>'\ maxilys@aaaaaa.fr\ :nt: '</email>'\
      \ :nt: '<uri></uri>'\
      \ :nt: '<ip>'\ 77.216.206.65\ :nt: '</ip>'\
      \ :nt: '<text>'\ \ :cp: '<![CDATA[Thank you for these rare minutes of
    \ :nt: '</comment>'\
  \ :nt: '</thread>'\
\ :nt: '</comments>'\
```

So, a comments tag that contains one or more thread tags, which contain one or more comment tags. Piece of cake to traverse using ElementTree!

There is an obvious match between comments and threads in Haloscan and Disqus. Good.

3. Create the necessary threads and whatever in Disqus

This is the tricky part, really, because it requires some things from your blog.

- You *must* have a permalink for each post
- Each permalink should be a separate page. You can't have permalinks with # in the URL
- You need to know what haloscan id you used for each post's comments, and what the permalink for each post is.

For example, suppose you have a post at <http://lateral.netmanagers.com.ar/weblog/posts/ADV0.html> and it has a Haloscan comments link like this:

```
<a href="javascript:HaloScan('ADV0');" target="_self"><script type="text/javascript">postCount('ADV0');</script></a>
```

You know where else that 'ADV0' appears? In Haloscan's XML file, of course! It's the "id" attribute of a thread.

Also, the title of this post is “Advogato post for 2000-01-17 17:19:57” (hey, it’s *my* blog ;-)

Got that?

Then we want to create a thread in Disqus with that *exact same data*:

- URL
- Thread ID
- Title

The bad news is... you need to gather this information for your entire blog and store it somewhere. If you are lucky, you may be able to get it from a database, as I did. If not... well, it’s going to be a lot of work :-)

For the purpose of this explanation, I will assume you got that data nicely in a dictionary indexed by thread id:

```
{
  id1: (url, title),
  id2: (url, title)
}
```

4. Post the comments from Haloscan to Disqus

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “download”.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “download”.

Here’s the code. It’s not really tested, because I had to do several attempts and fixes, but it should be *close* to ok (download.. download: <http://lateral.netmanagers.com.ar/static/haloscan2disqus.py>).

```
\ :c: `#!/usr/bin/python`\
\ :c: `# -*- coding: utf-8 -*-`\

\ :c: `# Read all comments from a CAIF file, the XML haloscan exports`\

\ :kn: `from`\ \ :nn: `disqus`\ \ :kn: `import`\ \ :n: `DisqusService`\
\ :kn: `from`\ \ :nn: `xml.etree`\ \ :kn: `import`\ \ :n: `ElementTree`\
\ :kn: `from`\ \ :nn: `datetime`\ \ :kn: `import`\ \ :n: `datetime`\
\ :kn: `import`\ \ :nn: `time`\

\ :c: `# Obviously these should be YOUR comment threads ;-)`\
\ :n: `threads`\ \ :o: `='\ \ :p: `{`\
\ :s: `ADV0'\ \ :p: `'\ \ :p: `(\ \ :s: `http://lateral.netmanagers.com.ar/static/haloscan2disqus.py`
```

```

\ :s:'ADV1' \ \ :p:'' \ \ :p:'(\ \ :s:'http://lateral.netmanagers.o
\ :p:'' \ \

\ :n:'key' \ \ :o:'=' \ \ :s:'USE YOUR API KEY HERE' \ \
\ :n:'ds' \ \ :o:'=' \ \ :n:'DisqusService' \ \ :p:'()' \ \
\ :n:'ds' \ \ :o:'. ' \ \ :n:'login' \ \ :p:'(\ \ :n:'key' \ \ :p:') \ \
\ :n:'forum' \ \ :o:'=' \ \ :n:'ds' \ \ :o:'. ' \ \ :n:'get_forum_list' \ \ :p:'

\ :k:'def' \ \ :nf:'importThread' \ \ :p:'(\ \ :n:'node' \ \ :p:'):\ \
\ :n:'t_id' \ \ :o:'=' \ \ :n:'node' \ \ :o:'. ' \ \ :n:'attrib' \ \ :p:'[ \ \

\ :c:'# Your haloscan thread data' \ \
\ :n:'thr_data' \ \ :o:'=' \ \ :n:'threads' \ \ :p:'[ \ \ :n:'t_id' \ \ :p:

\ :c:'# A Disqus thread: it will be created if needed' \ \
\ :n:'thread' \ \ :o:'=' \ \ :n:'ds' \ \ :o:'. ' \ \ :n:'thread_by_identifie

\ :c:'# Set the disqus thread data to match your blog' \ \
\ :n:'ds' \ \ :o:'. ' \ \ :n:'update_thread' \ \ :p:'(\ \ :n:'forum' \ \ :p:

\ :c:'# Now post all the comments in this thread' \ \
\ :k:'for' \ \ :n:'node' \ \ :ow:'in' \ \ :n:'node' \ \ :o:'. ' \ \ :n:'fi
\ :n:'dt' \ \ :o:'=' \ \ :n:'datetime' \ \ :o:'. ' \ \ :n:'strptime' \ \
\ :n:'name' \ \ :o:'=' \ \ :n:'node' \ \ :o:'. ' \ \ :n:'find' \ \ :p:'(
\ :n:'email' \ \ :o:'=' \ \ :n:'node' \ \ :o:'. ' \ \ :n:'find' \ \ :p:'(
\ :n:'uri' \ \ :o:'=' \ \ :n:'node' \ \ :o:'. ' \ \ :n:'find' \ \ :p:'(
\ :n:'text' \ \ :o:'=' \ \ :n:'node' \ \ :o:'. ' \ \ :n:'find' \ \ :p:'(

\ :k:'print' \ \ :s:'-' \ \ :o:'*' \ \ :mi:'80' \ \
\ :k:'print' \ \ :s:'Name:' \ \ :p:', ' \ \ :n:'name' \ \
\ :k:'print' \ \ :s:'Email:' \ \ :p:', ' \ \ :n:'email' \ \
\ :k:'print' \ \ :s:'Date:' \ \ :p:', ' \ \ :n:'dt' \ \
\ :k:'print' \ \ :s:'URL:' \ \ :p:', ' \ \ :n:'uri' \ \
\ :k:'print' \ \
\ :k:'print' \ \ :s:'Text:' \ \
\ :k:'print' \ \ :n:'text' \ \

\ :k:'print' \ \ :n:'ds' \ \ :o:'. ' \ \ :n:'create_post' \ \ :p:'(\ \
\ :n:'created_at' \ \ :o:'=' \ \ :n:'dt' \ \
\ :n:'time' \ \ :o:'. ' \ \ :n:'sleep' \ \ :p:'(\ \ :mi:'1' \ \ :p:') \ \

\ :k:'def' \ \ :nf:'importComments' \ \ :p:'(\ \ :n:'fname' \ \ :p:'):\ \
\ :n:'tree' \ \ :o:'=' \ \ :n:'ElementTree' \ \ :o:'. ' \ \ :n:'parse' \ \ :p:
\ :k:'for' \ \ :n:'node' \ \ :ow:'in' \ \ :n:'tree' \ \ :o:'. ' \ \ :n:'fi
\ :n:'importThread' \ \ :p:'(\ \ :n:'node' \ \ :p:') \ \

\ :c:'# Replace comments.xml with the file you downloaded from Haloscan' \ \
\ :n:'importComments' \ \ :p:'(\ \ :s:'comments.xml' \ \ :p:') \ \

```

Now, if we are lucky, you already have a nice and fully functioning collection of comments in your Disqus account, and you should be calm knowing you have not lost your data. Ready for the final step?

5. Hack the blog so the links to Haloscan now work for Disqus

You may not need to do anything beyond what the Disqus install guide suggests. If you have to do the custom generic install editing HTML manually, it's something like this:

First I changed my comment links. Here's the Haloscan version:

```
<a href="javascript:HaloScan('ADV0');" target="_self">
<script type="text/javascript">postCount('ADV0');</script></a>
```

And here's the Disqus version:

```
<a href="http://lateral.netmanagers.com.ar/weblog/posts/ADV0.html#disqus_thread">
```

Add the monkeypatching javascript at the bottom, before </body> like the Disqus install guide says (here's mine, it's not the one you want!):

```
<script type="text/javascript">
//
(function() {
    var links = document.getElementsByTagName('a');
    var query = '?';
    for(var i = 0; i &lt; links.length; i++) {
        if(links[i].href.indexOf('#disqus_thread') &gt;= 0) {
            query += 'url' + i + '=' + encodeURIComponent(links[i].href) + '&amp;';
        }
    }
    document.write('&lt;script charset="utf-8" type="text/javascript" src="http://disqus.com/forums/' + query + 'comments.js"&gt;&lt;/script&gt;');
})();
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="204 662 789 691" data-label="Text"><p>Also, in the post's own webpage, add the "embed code" from the Disqus install guide wherever you want the comments to appear.</p></div><div data-bbox="204 690 789 733" data-label="Text"><p>If you want more than one page to share a set of comments (for example, I use the same comments for the spanish and english versions of a post), you will have to use something like this before the embed code:</p></div><div data-bbox="246 743 1000 785" data-label="Text"><pre>&lt;script type="text/javascript"&gt;
    var disqus_url = "http://the_url_where_the_comments_really_belong.com";
&lt;/script&gt;</pre></div><div data-bbox="228 794 683 809" data-label="Text"><p>And that's about it, it should be enough to get you going. However:</p></div><div data-bbox="231 817 789 847" data-label="List-Group"><ul><li>• If you found this guide useful, and it saved you money, why not be a nice guy and give it to me instead? There's a donate link at the left, you know ;-)</li></ul></div><div data-bbox="479 878 513 893" data-label="Page-Footer"><p>374</p></div>
```

- If you can't manage to fix it, ask me for help. I have very reasonable consulting rates!

Make sure to export your haloscan comments while you still can!

When I was migrating the comments I noticed a page looked wrong in the site, and started fixing it.

While I was reading, I noticed a couple of things:

- It was published almost exactly 5 years ago
- It may be the smartest thing I ever wrote

Sometimes, you get an idea, and you can give it shape. Since I really think it's not completely stupid, I translated it to spanish (there was only an english version) y "reprint" it today.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "being a good lamarckian froggie!".

It's about programming, it's about evolution, and I hope you like [Being a good Lamarckian froggie!](http://lateral.netmanagers.com.ar/stories/34.html).. being a good lamarckian froggie!: <http://lateral.netmanagers.com.ar/stories/34.html>

Many programs require passwords from the user.

It's nice when a program can remember the password you give it.

It's nicer when it stores said password safely. However, it's not trivial to do that if you care for cross-platform support.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "python keyring".

Or at least it wasn't until Kang Zhang wrote [python keyring](http://home.python-keyring.org/).. `python keyring`: <http://home.python-keyring.org/>

, a module that abstracts the password storage mechanisms for KDE, GNOME, OSX and windows (and adds a couple of file-based backends just in case).

So, how does it work?

Install it in the usual way. If it's not packaged for your distro/operating system, just use `easy_install`:

```
easy_install keyring
```

You could also get it from mercurial:

```
hg clone http://bitbucket.org/kang/python-keyring-lib/
```

The API is simplicity itself. This is how you save a secret:

```
\ :kn: `import` \ :nn: `keyring` \
 \ :n: `keyring` \ :o: `.` \ :n: `set_password` \ :p: `(` \ :s: `keyring_den
```

You may get this dialog (or some analog on other platforms):

And here's the proof that it was saved correctly (this is KDE's password manager):

And how do you get the secret back?

```
\ :kn:'import' \ \ :nn:'keyring' \
\ :k:'print' \ \ :n:'keyring' \ \ :o:'. '\ \ :n:'get_password' \ \ :p:'('\ \
```

This is how it runs:

```
$ python load.py
thisisabadpassword
```

As you can see, the API is as easy as it could possible get. It even chose the KWallet backend automatically because I am in KDE!

Python-keyring is a module that fixes a big problem, so a big thank you to Kang Zhang and Tarek Ziadé (who had the idea)

So, what do you *do* for a living?

—Hardest question ever

Whenever I am speaking with people who don't know me[1] that's the question I dread.

If someone asks my wife what she does, all she has to do is say "I'm a lawyer". If someone asks my mother, she'd say "I am a retired teacher". *Everyone* understands what a lawyer does, or what a retired teacher did.

If someone asks me... oh, boy, that's hard. I usually weasel out by saying "I work with computers" but that has several problems:

- They assume I repair PCs
- They start telling me how their windows box was slow until they installed some kropotkina which supergarbled their frobnozzles[4], then ask me my opinion on frobnozzle garbling. For or against?

It's really hard to explain that yes, I work with computers every day, but I almost never open one (in fact, I have a policy of *not touching* my customers computers), and I have no idea what a frobnozzle is.

I have tried saying "I work on server side things, like mail servers and such. I install them, support them and also consulting work, explaining companies what the best ways to improve their services are."

That one usually gets glassy eyes and a general "what?" look.

I could lie and say I program for a living, but that's not true. While I program a lot, it's usually not for money, and what little I do for money is just using programming as a sysadmin tool.

I could say "I'm a sysadmin" but most people have no idea what that is. It does tend to end conversations, though, so it has one thing going for it.

Nowadays I could say "I have a company", which is true (we are **awesome**, you should hire us to do whatever it is we do, more details at <http://www.netmanagers.com.ar>)

So, I usually manage to work around this question, but I have a problem: I'm not telling the truth, or if I am, I am not telling the truth *in spirit* because I am not conveying what my work *is*, but only what I *do*.

So, this post is about trying to explain what the hell I do for a living, in another way, which is more ... internally true, so to speak. This is really hard to do, so I am trying to just let the writing flow, maybe you can understand what I do even if it's not clearly explained.

I work with computers. I make them do what I want them to do. Whenever a regular user sits before his keyboard, he tries to make his computer follow his orders, which variable rates of success. I always succeed.

Sometimes, I am logged into a computer that manages data for thousands of people. They all are on my care. No, it's not their lives at stake, but a little part of their fun, or work is under my care. I help them. I *care* about them, and I want their fun, their work to be smooth and pleasant.

Often the computer will not do what they need. I will try with my craft to make it happen. I will write little programs, search for others on the Internet, carefully piece together a puzzle and make their needs be fulfilled.

I will write or install and configure those programs and do it well, because I am skilled, I have literally decades of training and experience, but I will mostly do it because I like order and function. I like when things flow unimpeded, I like when serendipitous accidents make things just click together.

I do those things for a living, yes, because I *need* to make a living. And later, when I'm off the clock and my boy is asleep and I have my own time, you know what I do? I do the same things because they are fun. And I will bother writing a 1300 word post about how I migrated my blog's comments from one site to another because it was *fun*.

Yes, I know, to most people that would not be fun at all, it would be a boring job, and they would hate doing it. And that's one of the many reasons I am a lucky man[5]: I have fun doing unusual things. That's really lucky, because if my idea of fun was watching "Gossip Girl" I would never have found anyone to pay me to do *that!*

But going back to what I do for a living, I create things. I don't create large, impressive things, I am not a bridge builder, an architect, I create small, useful things and try to do it with a certain taste or elegance. I am more like a silversmith doing cutlery. Sure, I'll try to make it nice to look at, but it must cut a chunk of beef first.

Yes, I work with computers, but how does that convey what I feel when after a solid day of work I can see that what was a lot of stupid computers and cables are now a working machine that can make 50000 phone calls a day?

How can I make anyone see the beauty in 3 hard lines of code that do nothing but print a bunch of numbers?

How can someone who makes a living any other way understand that I think things and they become real? No, not real as in a puff of smoke and there they are, but they become real through work and effort and thinking and cursing, which is what makes them *really* real.

I know most of this will sound like mysticism, but it's not, it's my honest truth, I really feel all these things as I work, all these things *are* my work. Sometimes when I crack a hard problem I want to fucking *sing*[7]_ that's how awesome it feels.

So, that's what I do for a living. I work with computers.
I implemented a (hopefully) reasonable open/save logic, supporting encryption,
compression, (and encrypted compression ;-) atomic saves.
Since almost every app can use this, why not make it a module?

1

system-message
WARNING/2 in `everything.txt`, line 23976
Duplicate explicit target name: "1".

Given my shyness problem, that's not too often[2]_

2

system-message
WARNING/2 in `everything.txt`, line 23978
Duplicate explicit target name: "2".

To those who only know me from public speaking[3]_: I am *painfully* shy.

3

system-message
WARNING/2 in `everything.txt`, line 23980
Duplicate explicit target name: "3".

Yes, it's perfectly possible to be a decent public speaker and be shy.

4

system-message
WARNING/2 in `everything.txt`, line 23982
Duplicate explicit target name: "4".

At least that's how windows users sound to me half the time

5

system-message
WARNING/2 in `everything.txt`, line 23984
Duplicate explicit target name: "5".

The others are of course y wife and boy, and if they ever read this: kisses for both.[6]_

6

system-message
WARNING/2 in `everything.txt`, line 23986
Duplicate explicit target name: "6".

He's just 2.9 years old but you know, the Internet keeps things forever.

7

system-message
WARNING/2 in `everything.txt`, line 23988
Duplicate explicit target name: "7".

If you know my voice, you know why I don't. My own son says "no, don't sing, daddy", except for his good night song, which is the only one he lets me sing. Oh, and sorry for the cursing, but no other word fits.

This has been a pet peeve of mine for years: programming shell scripts suck. They are ugly and error prone. The only reason why we still do it? There is no real replacement.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "python.reddit.com".
```

Or at least that *was* the case, until today I met iterpipes at python.reddit.com.
python.reddit.com: http://www.reddit.com/r/Python/comments/ahpou/iterpipes_library_for_running_shell_pipelines/
Iterpipes is “A library for running shell pipelines using shell-like syntax” and guess what? It’s *brilliant*.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "pypi page".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "pypi page".
```

Here’s an example from its [PYPI page](http://pypi.python.org/pypi/iterpipes).. pypi page: <http://pypi.python.org/pypi/iterpipes>
:

```
\ :go:`# Total lines in *.py files under /path/to/dir,
# use safe shell parameters formatting:

`\ \ :n:`\ \ :gp:`>>> \ \ :n:`total`\ \ :o:`=\ \ \ :n:`cmd`\ \ :p:`(`\
\ :gp:`... \ \ :s:`find {} -name {} -print0 | xargs -0 wc -l | tail -
\ :gp:`... \ \ :s:`/path/to/dir' \ \ :p`,` \ \ :s:`\*.py' \ \ :p`,`
\ :gp:`>>> \ \ :n:`run`\ \ :p:`(`\ \ :n:`total`\ \ :o:`|\ \ \ :n:`strip`\
\ :go:`315`\
```

Here’s how that would look in shell:

```
find /path/to/dir -name '*.py' -print0 | xargs -0 wc -l | tail -1 | awk '{p
```

You may say the shell version looks better. That’s an illusion caused by the evil that is shell scripting: the shell version is buggy.

Why is it buggy? Because if I control what’s inside /path/to/dir I can make that neat little shell command fail[1], but at least in python I can *handle errors*!

Also, in most versions you could attempt to write, this command would be unsafe because quoting and escaping in shell is insane!

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "prepared statements".
```

The iterpipes version uses the equivalent of SQL [prepared statements](http://dev.mysql.com/tech-resources/articles/4.1/prepared-statements.html): <http://dev.mysql.com/tech-resources/articles/4.1/prepared-statements.html>

which are much safer.

It's nearly **impossible** to do such a command in pure shell and be sure it's safe.

Also, the shell version produces a string instead of an integer, which sucks if you intend to do anything with it.

And the most important benefit is, of course, not when you try to make python act like a shell, but when you can stop pretending shell is a real programming language.

Consider this gem from Arch Linux's `/etc/rc.shutdown` script. Here, `DAEMONS` is a list of things that started on boot, and this script is trying to shut them down in reverse order, unless the daemon name starts with "!":

```
# Shutdown daemons in reverse order
let i=${#DAEMONS[@]}-1
while [ $i -ge 0 ]; do
    if [ "${DAEMONS[$i]:0:1}" != '!' ]; then
        ck_daemon ${DAEMONS[$i]#@} || stop_daemon ${DAEMONS[$i]#@}
    fi
    let i=i-1
done
```

Nice uh?

Now, how would that look in python (I may have inverted the meaning of `ck_daemon`)?

```
\ :c: '# Shutdown daemons in reverse order'\
\ :k: 'for'\ \ :n: 'daemon'\ \ :ow: 'in'\ \ :nb: 'reversed'\ \ :p: '('\ \ :n:
\ :k: 'if'\ \ :n: 'daemon'\ \ :p: '['\ \ :mi: '0'\ \ :p: ']\ \ :o: '=='\ \
\ :k: 'continue'\
\ :k: 'if'\ \ :n: 'ck_daemon'\ \ :p: '('\ \ :n: 'daemon'\ \ :p: '):\ \
\ :n: 'stop_daemon'\ \ :p: '('\ \ :n: 'daemon'\ \ :p: ')\ \
```

Where `stop_daemon` used to be this:

```
stop_daemon() {
    /etc/rc.d/$1 stop
}
```

And will now be this:

```
\ :k: 'def'\ \ :nf: 'stop_daemon'\ \ :p: '('\ \ :n: 'daemon'\ \ :p: '):\ \
\ :n: 'run'\ \ :p: '('\ \ :n: 'cmd'\ \ :p: '('\ \ :s: '/etc/rc.d/{ } stop'\ \
```

So, come on, people, we are in the 21st century, and shell scripting sucked in the 20th already.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "ten years old".

1

system-message

WARNING/2 in `everything.txt`, line 24084
Duplicate explicit target name: "1".

I leave that as exercise for the reader.

Since yesterday this blog is [ten years old](#).. ten years old: <http://lateral.netmanagers.com.ar/weblog/posts/ADV0.htm>
so, time for some history.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "advogato".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "advogato".
```

It all started in [advogato](#).. advogato: <http://advogato.org>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "read it today".
```

where you could still [read it today](#).. read it today: <http://advogato.org/person/ralsina/>

! (Please read it here instead ;-)

Then it moved to PyDS an early python desktop blog platform with a web interface, and was hosted in PyCS, a free service.

Then PyCS kinda died, and I started generating a static blog and hosting it in my ISP's free hosting. That sucked *bad*.

Then I started my own company, and I had my own servers, so I started hosting it there (even today this blog is completely static HTML!)

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "my own blogging software".
```

Then PyDS started acting weird, so I wrote [my own blogging software](#).. my own blogging software: <http://lateral.netmanagers.com.ar/stories/BBS41.html>

, which is a real mess, perhaps 25% finished, but it does things *exactly* the way I like them.

Currently, this blog is syndicated in Planeta PyAr, Planet Python, Planet Qt, Planeta LUGLI, and a couple other places.

This year, I decided to make the blog completely bilingual (English and Spanish), but I hate translating it.

According to the stats I have available, the blog is in average more popular now than ever (but yes, my most popular posts were years ago ;-)

These are the most popular pages in the last year:

```
system-message
• WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "pyqt by example".
```

[PyQt By Example](#).. pyqt by example: <http://lateral.netmanagers.com.ar/stories/BBS47.html>

Each of the parts of this series is in the top ten most visited pages. So, I really need to write more in this area since it seems to be what brings people to the site.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “responding to random blog posts”.

Responding to random blog posts... responding to random blog posts.: <http://lateral.netmanagers.com.ar/weblog/...>

A post explaining how another blogger is wrong. I think I make a good point, even if it's a bit cheap as a way to bring readers.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “queue management for qmail”.

Queue Management for Qmail.. queue management for qmail: <http://lateral.netmanagers.com.ar/stories/8.html>

Oldie but goodie... not really useful nowadays.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “how to make your own distro”.

How to make your own distro.. how to make your own distro: <http://lateral.netmanagers.com.ar/stories/38.html>

Yikes, people, stop reading this, it's just wrong :-)

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “python keyring is seriously nice”.

Python Keyring is seriously nice.. python keyring is seriously nice: <http://lateral.netmanagers.com.ar/weblog/pos...>

Finally something recent!

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “what i do for a living”.

What I do for a living.. what i do for a living: <http://lateral.netmanagers.com.ar/weblog/posts/BB859.html>

Another new post, I like it a lot.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “the linux booting process unveiled”.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “the linux booting process unveiled”.

The Linux Booting Process Unveiled.. the linux booting process unveiled: <http://lateral.netmanagers.com.ar/stor...>

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “current distro”.

A didactic post. I think it's still the better explanation on the subject I've seen around, if I may say so myself. For extra irony points, my [current distro](#).. current distro: <http://archlinux.org>

doesn't follow this.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "creating presentations using restructured text".

[Creating presentations using restructured text](#).. creating presentations using restructured text: <http://lateral.netmanagers.com.ar/stories/BBS52.html>

Another not-so-old post, I do this a lot :-)

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "custom widgets using pyqt".

[Custom widgets using PyQt](#).. custom widgets using pyqt: <http://lateral.netmanagers.com.ar/stories/27.html>

This is almost 6 years old, for a deprecated version of PyQt, and is completely broken for current versions. Stop reading this.

Lessons:

- 1) I need to write more about Qt and/or start flamewars with clueless IT writers
- 2) I need to search for ancient material and deprecate it
- 3) Having your own hosting and blogging software is neat
- 4) 10 years is a lot of time: 860 posts (or 913, depending on how you count)

Announcement:

Marave (nothing, in guaraní) is a relaxed text editor. Its goal is to let you focus in your writing, free of visual distractions and clutter.

It is written using PyQt, so it should work in all major platforms, and it is licensed under the GPLv2 so it is free software.

You can find the current SVN (no release yet) at <http://marave.googlecode.com>

Screenshots:

Status:

It's not finished yet, but it has the following features:

- You can edit text
- Minimalistic, "vanishing" user interface
- (Optional) audio feedback for the keyboard
- (Optional) relaxing music (requires internet access)
- You can customize the background, font, colours, and sounds
- Live spell checking (requires pyenchant)

There are also some major missing features:

- Search and Search/Replace is not implemented
- UI customizations are not stored
- UI has to be cleaned up a lot more
- It doesn't ask to save when closing
- Autosave not implemented

And at least one known bug:

- In windows the widgets are not well placed **Fixed in SVN**

A litte history:

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "ommwriter".
```

A few days ago, I saw [ommwriter.. ommwriter: http://www.ommwriter.com](http://www.ommwriter.com) mentioned in a tweet or something similar.

I was thinking "nice thing", but in the back of my mind I was also thinking "that can't be too hard to do". After all, the hard part of creating a program is *making it do things*, right?

Well, yes and no. I did manage to create a somewhat reasonable facsimile in a day, but tweaking the looks of it is driving me nuts :-)

What the Ipad means

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "by mark pilgrim".
```

Everything I will say here was probably better said [by Mark Pilgrim.. by mark pilgrim: http://diveintomark.org/archives/2010/01/29/tinkerers-sunset](http://diveintomark.org/archives/2010/01/29/tinkerers-sunset)

but what the heck, let's give it a shot.

Here's what's wrong with the iPad: it sucks *for me*.

Before anyone says "don't buy it then!" I'll say it first: I don't intend to buy one.

I think there is a place for iPads and it would go against most of my beliefs to say it shouldn't exist, but I also expect it to make our world poorer, if it's popular enough.

Yes it's hostile to tinkering. Yes, to read about that, see Mark Pilgrim's article, he's a much better writer than I.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "readme file".
```

I once posted the [README file](http://lateral.netmanagers.com.ar/weblog/2007/06/20.html).. readme file: <http://lateral.netmanagers.com.ar/weblog/2007/06/20.html>

for a piece of software called Atlast. It said things like “make everything programmable” and “[It is] far better to invest the effort up front to create a product flexible enough to be adapted at will, by its users, to their immediate needs.”

The iPad and most other Apple products are the antithesis of that. They are products meant to be exactly as they are, and that’s all they are going to be. If you want to change the way it acts, you need to pay and be subject of Apple’s whim, or “break into” your own device.

That hurts me. I see people give up even the possibility of changing what a (let’s say it) pretty, useful, powerful device is capable of, just because they don’t want that freedom. I can understand that from, say, a used car salesman, or whatever, someone without any inclination for that craft.

But I see *freaking programmers* buying apple kit. And I see them happy with their iPhones and iPods and (soon, surely) iPads, buying apps from the only source enabled to sell them, buying the apps that are allowed by a single party, that decides, hey, this app? you won’t need it!

I see them and I say to myself, dude, that thing you hold in your hand is more powerful than anything we had 10 years ago, there *must* be something cool you could do with it that no one else is doing.

What’s the *vision* a programmer has of his future if he endorses Apple’s closed kit? A future where he can program something only if Apple approves? A future where a “real” computer is a SDK for the things “real people” use in their everyday lives?

What is *wrong* with you? What happened to you? Are you now the kind of guy that’s just happy with what he’s given? Are you now a freaking utilitarian? Are you *old* now?

Have you noticed the trend in Apple’s new products is towards *less* control by the user? First it was just handhelds, now there’s a tablet. What was the last new interesting Apple product that wasn’t locked up?

Here they had a device which could have OSX or iPhone OS, and they went with iPhone OS. There is a reason for that: it makes them more money.

For OSX, they make money of the hardware, the OS upgrades, and some apps. On the iPad, they make money every time you buy any app, every time you buy a book to read in it, every time you use 3G with the prepaid plan, and I am sure they are going to find other ways too.

And what’s the key to making money that way? Control. If they had no exclusive control of the App store, they lose a source of revenue. If they allowed for easy development of hobby apps, they would lose revenue. If they could let you replace the freaking battery, they would lose revenue.

And if there’s one thing companies hate is losing revenue. Apple saw two paths ahead, one leading to huge money, the other to just big money. They have taken the huge money path, and it’s working for them. They are *not* going back.

If everyone goes along for the ride, it will be a sad thing.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “marave”.

The first “good” version of [Marave](http://marave.googlecode.com).. marave: <http://marave.googlecode.com>

my relaxing text editor is out!

What is Marave?

Marave is an editor that doesn't distract you. It has a fullscreen interface, and most of the time, while you write, you will only see your text, and maybe a background:

Of course it is also quite configurable:

Some of the features:

- Custom backgrounds (images or colors)
- Font and font size are configurable
- Resizeable editor of configurable opacity
- “Vanishing” UI, when you type, it all goes away
- Optional media player (right now aimed at streaming audio, maybe soundscapes someday)
- Optional audio feedback for the keyboard (just in case you miss the old typewriter)
- Theme support
- Multilingual spellchecking (requires pyenchant)

Marave is implemented using PyQt, so it should work in all major platforms, Windows, Linux and Mac OSX, as long as you have PyQt installed.

In the future, easy to use binaries will be provided for Windows and Mac.

This version is not feature complete. Specifically:

- Search+Replace is not done
- There may be some customizations not implemented

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “marave 0.1”.

Download [Marave 0.1](http://marave.googlecode.com/files/marave-0.1.tar.gz). marave 0.1: <http://marave.googlecode.com/files/marave-0.1.tar.gz>

UPDATE 0.2 is out, at <http://code.google.com/p/marave/downloads/list>

Version 0.2 of Marave, a distraction-free fullscreen editor is out at <http://marave.googlecode.com>

This version includes several bugs fixed and features implemented since 0.1.1:

- A corrupted Right-click menu (Issue 20)
- Flickering on background changes
- More detailed licensing information
- More tested on Windows
- Added help (F1)
- Search & Replace (but replace all is not done)

- New artwork
- Status notifications
- Document Info (Ctrl+I)
- Better feedback in the UI elements (specially the buttons)
- Save font size correctly
- Fix “Starts in the background” problem (Issue 17)

Marave is free software released under the GPL, and should work in all major desktop platforms.

I would love feedback on this release, as well as ideas for Marave’s future, so a mailing list for Marave has been opened:

<http://groups.google.com/group/marave-discuss>

system-message
WARNING/2 in `everything.txt`, line 3
 Duplicate explicit target name: “give me money”.

Of course, if you like Marave, feel free to [give me money](https://www.paypal.com/cgi-bin/webscr).. give me money: <https://www.paypal.com/cgi-bin/webscr>
 Version 0.3 of Marave, a distraction-free fullscreen editor is out at <http://marave.googlecode.com>
 This version includes several bugs fixed and features implemented since 0.2:

- New ‘Styles’ support, you can change the look of Marave with CSS syntax
- Debugged themes support, a few themes included
- Fixed bug saving text color
- Fixed font changing bug
- Use the document name in window title
- “Now playing” notification

Marave is free software released under the GPL, and should work in all major desktop platforms.

I would love feedback on this release, as well as ideas for Marave’s future, so if you want to help, please join the mailing list:

<http://groups.google.com/group/marave-discuss>

system-message
WARNING/2 in `everything.txt`, line 3
 Duplicate explicit target name: “give me money”.

system-message
WARNING/2 in `everything.txt`, line 3
 Duplicate explicit target name: “give me money”.

Of course, if you like Marave, feel free to [give me money](https://www.paypal.com/cgi-bin/webscr).. give me money: <https://www.paypal.com/cgi-bin/webscr>
Version 0.4 of Marave, a distraction-free fullscreen editor is out at <http://marave.googlecode.com>
This version includes several bugs fixed and features implemented since 0.4:

- It works better with WindowMaker (dialogs appeared behind main window)
- Works better with some Qt styles (combo boxes were unusable)
- Added support for SVG backgrounds
- Code cleanups

Marave is free software released under the GPL, and should work in all major desktop platforms.

I would love feedback on this release, as well as ideas for Marave's future, so if you want to help, please join the mailing list:

<http://groups.google.com/group/marave-discuss>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "give me money".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "give me money".

Of course, if you like Marave, feel free to [give me money](https://www.paypal.com/cgi-bin/webscr).. give me money: <https://www.paypal.com/cgi-bin/webscr>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "marave".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "marave".

I have worked really hard on [Marave](http://marave.googlecode.com).. marave: <http://marave.googlecode.com>, a full screen editor in the style of ommwriter, DarkRoom, WriteRoom, pyRoom, etc. I have worked very hard and I want users to *use* it.

Or not even that, I want them to have a *chance* of using it.

That means I want it to work on Windows (and maybe OSX some day, too, if someone helps me). Which means, I have to package it for windows.

Let's do a quick comparison here from the points of view of the user and the developer.

The User, In Linux

This is in Arch Linux, which is what I use, in other Linux variants it will be pretty much the same once Marave is a bit more well known.

```
yaourt -S marave-svn --noconfirm
```

That gets the code from SVN (right now it's the best way, later I will package releases, too), all required dependencies, builds and installs. It takes all of 15 seconds in my notebook.

After that, you have a fully working Marave.

In case it's not packaged for your distro, just install PyQt (which surely is) and run this command:

```
easy_install marave
```

The User, in Windows

You go to <http://marave.googlecode.com>, click on "Marave-0.5.win32.exe" (Not linked yet, it's not finished), then download a 10MB program. That is a 10MB program because windows doesn't believe in packages and dependencies. On Linux, a Marave package could be under 1MB (most of it images), and not be executable, just data.

Of course nowadays web browsers don't actually *run* programs on download, so... let's see it as a gallery!

Yes, save it.

Double click to open it

Yes, I agree

Sure, whatever

Nice...

Good to hear!

Now, this Marave that just got installed may or may not currently work because of a missing MSVCR90.DLL but that's for the next section...

The Developer, in Linux

First, here's the biggest problem a Linux packager can have:

Since Marave is a new app, and I develop it in the rather cutting-edge Arch Linux, it uses some newish features only available in recent versions of Qt. In fact, it doesn't work with PyQt < 4.6, which is not available in some slow distros, like Debian, or even in a not-latest Ubuntu.

Solution? Well, I **could** just ignore it, but what the heck, let's fix it instead!

```
system-message
```

```
WARNING/2 in everything.txt, line 3
```

```
Duplicate explicit target name: "pyinstaller".
```

Thanks to PyInstaller.. pyinstaller: <http://www.pyinstaller.org/>

it's not even hard to do, here's the spec file:

```

\ :n:'a'\ \ :o:'='\ \ :n:'Analysis'\ \ :p:'([\ \ :n:'os'\ \ :o:'.'\ \ :n:'pathex'\ \ :o:'='\ \ :p:'[\ \ :s: '/home/ralsina/trunk/t

\ :n:'pyz'\ \ :o:'='\ \ :n:'PYZ'\ \ :p:'(\ \ :n:'a'\ \ :o:'.'\ \ :n:'pur
\ :n:'exe'\ \ :o:'='\ \ :n:'EXE'\ \ :p:'(\ \ :n:'pyz'\ \ :p:','\
\ :n:'a'\ \ :o:'.'\ \ :n:'scripts'\ \ :p:','\
\ :n:'exclude_binaries'\ \ :o:'='\ \ :mi:'1'\ \ :p:','\
\ :n:'name'\ \ :o:'='\ \ :n:'os'\ \ :o:'.'\ \ :n:'path'\ \ :o:'.'\
\ :n:'debug'\ \ :o:'='\ \ :bp:'False'\ \ :p:','\
\ :n:'strip'\ \ :o:'='\ \ :bp:'False'\ \ :p:','\
\ :n:'upx'\ \ :o:'='\ \ :bp:'True'\ \ :p:','\
\ :n:'console'\ \ :o:'='\ \ :mi:'0'\ \ :p:')'\

\ :n:'coll'\ \ :o:'='\ \ :n:'COLLECT'\ \ :p:'(\ \ :n:'exe'\ \ :p:','\
\ :n:'a'\ \ :o:'.'\ \ :n:'binaries'\ \ :p:','\
\ :p:'([\ \ :s: 'radios.txt'\ \ :p:','\ \ :s: 'marave/radios
\ :n:'Tree'\ \ :p:'(\ \ :s: 'marave/icons'\ \ :p:','\ \ :s: '
\ :n:'Tree'\ \ :p:'(\ \ :s: 'marave/backgrounds'\ \ :p:','\ \ :s: '
\ :n:'Tree'\ \ :p:'(\ \ :s: 'marave/clicks'\ \ :p:','\ \ :s: '
\ :n:'Tree'\ \ :p:'(\ \ :s: 'marave/stylesheets'\ \ :p:','\ \ :s: '
\ :n:'Tree'\ \ :p:'(\ \ :s: 'marave/themes'\ \ :p:','\ \ :s: '
\ :n:'a'\ \ :o:'.'\ \ :n:'zipfiles'\ \ :p:','\
\ :n:'a'\ \ :o:'.'\ \ :n:'datas'\ \ :p:','\
\ :n:'strip'\ \ :o:'='\ \ :bp:'False'\ \ :p:','\
\ :n:'upx'\ \ :o:'='\ \ :bp:'True'\ \ :p:','\
\ :n:'name'\ \ :o:'='\ \ :n:'os'\ \ :o:'.'\ \ :n:'path'\ \ :o:

```

Use this, and PyInstaller will produce a nice folder full of everything Marave needs to run on any Linux.

OTOH, if you can rely on a recent PyQt being available, it's also simple. Here's a packaging configuration for a similar package in Arch Linux (I must confess not having done one for Marave yet). For other distributions it should be about as simple, if more verbose, and **someone else probably does it for you**:

```

# Contributor: Roberto Alsina <ralsina@kde.org>
pkgname=python-rst2pdf
pkgver=0.12.1
pkgrel=4
pkgdesc="Create PDFs from simple text markup, no LaTeX required."
arch=('i686' 'x86_64')
url="http://rst2pdf.googlecode.com"
license=('custom')
depends=('python' 'setuptools' 'docutils' 'pygments' 'python-reportlab' 'py
source=(http://rst2pdf.googlecode.com/files/rst2pdf-$pkgver.tar.gz LICENSE
optdepends=('unicontor: vector images support'
            'python-svglib: SVG support'
            'python-wordaxe: hyphenation'
            'pythonmagick: PDF images support')
build() {
cd $startdir/src/rst2pdf-$pkgver

```

```

python setup.py install --root=$startdir/pkg || return 1
install -D ../LICENSE.txt $startdir/pkg/usr/share/licenses/python-rst2pdf/C
install -D doc/rst2pdf.1 $startdir/pkg/usr/share/man/man1/rst2pdf.1
}
md5sums=(' ea6beda9a46f34ba42c4c94d48cc607a'
         ' 416f8046c66b9476cdbacda69a673afe')

```

And that's all you need to know about the process of packaging your app for Linux. It's **easy** to do, and most of the time, easy to do right! Now, let's go to our final section...

Windows for the developer

First, remember that of relying on the system's version of Qt? Forget it, there is no system version available. And no python either. And noone is going to install it or your app, so it's "ship everything yourself" mode, or nothing.

But anyway, PyInstaller works for Windows too! So, using the same spec file, it works. Right?

Well, no beause of two problems.

Problem 1: You need an installer

Users are not going to open a zip somewhere, then do a shortcut to the binary on Windows, so you need to do some operations, and that means an installer.

Here's what I came up with to use NSIS, a free installer creator for Windows:

```

;-----
;Include Modern UI

!include "MUI2.nsh"

;-----
;General

;Name and file
Name "Marave"
OutFile "Marave-0.5.win32.exe"

;Default installation folder
InstallDir "$LOCALAPPDATA\Marave"

;Get installation folder from registry if available
InstallDirRegKey HKCU "Software\Marave" ""

;Request application privileges for Windows Vista
RequestExecutionLevel user

;-----
;Interface Settings

```

```

!define MUI_ABORTWARNING

;-----
;Pages

!insertmacro MUI_PAGE_LICENSE "LICENSE"
!insertmacro MUI_PAGE_DIRECTORY
!insertmacro MUI_PAGE_INSTFILES

!insertmacro MUI_UNPAGE_CONFIRM
!insertmacro MUI_UNPAGE_INSTFILES

;-----
;Languages

!insertmacro MUI_LANGUAGE "English"

;-----
;Installer Sections

Section "Install"

SetOutPath "$INSTDIR"
File /r "dist\marave"

;Store installation folder
WriteRegStr HKCU "Software\Marave" "" $INSTDIR

;Create uninstaller
WriteUninstaller "$INSTDIR\Uninstall.exe"

;Create shortcuts
CreateDirectory $SMPROGRAMS\Marave
CreateShortCut "$SMPROGRAMS\Marave\Marave.lnk" "$INSTDIR\marave\marave.exe"
CreateShortCut "$SMPROGRAMS\Marave\Uninstall Marave.lnk" "$INSTDIR\Uninstall

SectionEnd

;-----
;Uninstaller Section

Section "Uninstall"

Delete "$INSTDIR\Uninstall.exe"
RMDir /r "$INSTDIR"

DeleteRegKey /ifempty HKCU "Software\Marave"

```

SectionEnd

It's comparable to the effort of building a packaging file, really, except every time you want to test it... you install it. There is no way (AFAICS) to see what's inside the installer except running it!

When things fail, you get no error messages, at least not the kind that is useful for a developer, the guy that **needs to know what went wrong**.

And after it's finished, you may end with a non-working program because of...

Problem 2: system libraries don't exist

Python 2.6 binaries are built using Visual Studio. That means they require the Visual Studio Runtime, specifically MSVCR90.DLL. That contains what on Linux would be considered part of libc. (linux guy: imagine apps that depend on a specific libc... hard to do!)

On Linux that's part of the system. Further, if you wanted, you can redistribute it. On Windows... well, it's a bit different.

- 1) It's part of the "Visual C++ redistributables"
- 2) Installing that doesn't guarantee it will work (yes, I have tried)
- 3) The license for those 'redistributables' says you can't make them available for download.

I have been told that including that in your installer is fine and dandy, but how is that not making them available for download?

So what can you do when you need a library and can't ship it and the user won't install it?

Well, that's why there is no Windows binary of Marave yet. Of course if anyone can help, I'd be really, really happy!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "the usual place".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "the usual place".

Just uploaded Marave 0.5 to [the usual place](http://marave.googlecode.com).. the usual place: <http://marave.googlecode.com>. Marave is a relaxed, fullscreen text editor that tries not to distract you.

It even includes a simple music player just so you don't have to think about switching to another application!

This release has several bugs fixed, and looks a bit nicer.

The main new feature is ... internationalizacion. It now includes a spanish translation only, but if you want to help translating it to any other language, please step forward!

There are only 60 phrases to translate, so it shouldn't be more than one hour of work.

Here's a screenshot of this release:

Marave is free software under the GPLv2, and should work on any platform where PyQt works, which means Windows, Mac, and Unix-like operating systems, at least.

After some network hiccups (48 hour long hiccups, too), I have Internet at home again.

Man, am I pissed at the dumbasses who run velocom.com.ar :-P

This is not interesting for almost noone, but since my google-fu didn't let me find it and it was a bit of a pain to do:

This is how you implement 'replace all' in a QPlainTextEdit (or a QTextEdit, for that matter) using PyQt (similar for C++ of course).

```
\ :k:'def'\ \ :nf:'doReplaceAll'\ \ :p:'('\ \ :bp:'self'\ \ :p:'):'\
\ :c:'# Replace all occurrences without interaction'\

\ :c:'# Here I am just getting the replacement data'\
\ :c:'# from my UI so it will be different for you'\
\ :n:'old'\ \ :o:'='\ \ :bp:'self'\ \ :o:'. '\ \ :n:'searchReplaceWidget
\ :n:'new'\ \ :o:'='\ \ :bp:'self'\ \ :o:'. '\ \ :n:'searchReplaceWidget

\ :c:'# Beginning of undo block'\
\ :n:'cursor'\ \ :o:'='\ \ :bp:'self'\ \ :o:'. '\ \ :n:'editor'\ \ :o:'. '\
\ :n:'cursor'\ \ :o:'. '\ \ :n:'beginEditBlock'\ \ :p:'()' '\

\ :c:'# Use flags for case match'\
\ :n:'flags'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QTextDocument'\ \
\ :k:'if'\ \ :bp:'self'\ \ :o:'. '\ \ :n:'searchReplaceWidget'\ \ :o:'. '\
\ :n:'flags'\ \ :o:'='\ \ :n:'flags'\ \ :o:'|\ '\ \ :n:'QtGui'\ \ :o:'. '\

\ :c:'# Replace all we can'\
\ :k:'while'\ \ :bp:'True'\ \ :p:':'\
\ :c:'# self.editor is the QPlainTextEdit'\
\ :n:'r'\ \ :o:'='\ \ :bp:'self'\ \ :o:'. '\ \ :n:'editor'\ \ :o:'. '\
\ :k:'if'\ \ :n:'r'\ \ :p:':'\
\ :n:'qc'\ \ :o:'='\ \ :bp:'self'\ \ :o:'. '\ \ :n:'editor'\ \ :o:'. '\
\ :k:'if'\ \ :n:'qc'\ \ :o:'. '\ \ :n:'hasSelection'\ \ :p:'()'\
\ :n:'qc'\ \ :o:'. '\ \ :n:'insertText'\ \ :p:'('\ \ :n:'new
\ :k:'else'\ \ :p:':'\
\ :k:'break'\

\ :c:'# Mark end of undo block'\
\ :n:'cursor'\ \ :o:'. '\ \ :n:'endEditBlock'\ \ :p:'()' '\
```

There are other, easier ways to do it, but this one makes it all appear as a single operation in the undo stack and all that.

yak shaving (idiomatic) Any apparently useless activity which, by allowing you to overcome intermediate difficulties, allows you to solve a larger problem.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "implemented a generic syntax highlighter for pyqt using pygments".

A while ago, I wrote how I [implemented a generic syntax highlighter for PyQt using Pygments](http://lateral.netmanagers.com.ar/weblog/posts/BB).

I got a request for such a feature in Marave, so I digged that code and... it's freaking useless. It's just too slow for reasonable use.

So, that yak's hair is all grown up again, and I just got this new pair of scissors!

The goal is a way to highlight syntax in a QPlainTextEdit that:

- Doesn't require programming to add a new highlighter
- Doesn't require programming to add a new color scheme
- Doesn't require me to spend a year writing highlighters for existing languages
- Is fast enough

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "source highlight qt".

A quick google shows that for C++ you can use [Source highlight qt](http://srchiliteqt.sourceforge.net/). source highlight qt: <http://srchiliteqt.sourceforge.net/>

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "gnu source highlight".

which is based on [GNU source highlight](http://www.gnu.org/software/src-highlight/). gnu source highlight: <http://www.gnu.org/software/src-highlight/>

Alas, no python binding that I could find. So, let's write one!

Here it is: <http://marave.googlecode.com/svn/trunk/marave/highlight/>

And here's a screenshot of the demo program running, showing itself in its entirety:

You can create a color scheme using CSS, a language definition is a text file, there are a bazillion already written, it seems to be fast enough.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "marave".

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "marave".

So, another yak shaved, another feature (not finished!) for [Marave](http://marave.googlecode.com). marave: <http://marave.googlecode.com>

Making the Plugin Configurable

This plugin supports different modes of operation. To make this reachable to the user, you need to implement a few extra methods.

The `addConfigWidgets` method takes a dialog argument and adds whatever you want there:

```
\ :nd: '@classmethod'\
\ :k: 'def'\ \ :nf: 'addConfigWidgets'\ \ :p: '('\ \ :bp: 'self'\ \ :p: ', '\
\ :k: 'print'\ \ :s: 'Adding widgets to smarty config'\
\ :n: 'l'\ \ :o: '='\ \ :n: 'dialog'\ \ :o: '.'\ \ :n: 'ui'\ \ :o: '.'\ \ :n:
\ :bp: 'self'\ \ :o: '.'\ \ :n: 'q'\ \ :o: '='\ \ :n: 'QtGui'\ \ :o: '.'\ \ :
\ :k: 'if'\ \ :s: 'q'\ \ :ow: 'in'\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'mod
\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'q'\ \ :o: '.'\ \ :n: 'setChecked'\ \ :p
\ :bp: 'self'\ \ :o: '.'\ \ :n: 'b'\ \ :o: '='\ \ :n: 'QtGui'\ \ :o: '.'\ \ :
\ :k: 'if'\ \ :s: 'B'\ \ :ow: 'in'\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'mod
\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'b'\ \ :o: '.'\ \ :n: 'setChecked'\ \ :p
\ :bp: 'self'\ \ :o: '.'\ \ :n: 'd'\ \ :o: '='\ \ :n: 'QtGui'\ \ :o: '.'\ \ :
\ :k: 'if'\ \ :s: 'd'\ \ :ow: 'in'\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'mod
\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'd'\ \ :o: '.'\ \ :n: 'setChecked'\ \ :p
\ :bp: 'self'\ \ :o: '.'\ \ :n: 'e'\ \ :o: '='\ \ :n: 'QtGui'\ \ :o: '.'\ \ :
\ :k: 'if'\ \ :s: 'e'\ \ :ow: 'in'\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'mod
\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'e'\ \ :o: '.'\ \ :n: 'setChecked'\ \ :p
\ :n: 'l'\ \ :o: '.'\ \ :n: 'addWidget'\ \ :p: '('\ \ :bp: 'self'\ \ :o: '.'\
\ :n: 'l'\ \ :o: '.'\ \ :n: 'addWidget'\ \ :p: '('\ \ :bp: 'self'\ \ :o: '.'\
\ :n: 'l'\ \ :o: '.'\ \ :n: 'addWidget'\ \ :p: '('\ \ :bp: 'self'\ \ :o: '.'\
\ :n: 'l'\ \ :o: '.'\ \ :n: 'addWidget'\ \ :p: '('\ \ :bp: 'self'\ \ :o: '.'\
```

And then the config dialog will look like this:

But then you need to save those options somewhere, which you do reimplementing `saveConfig`:

```
\ :nd: '@classmethod'\
\ :k: 'def'\ \ :nf: 'saveConfig'\ \ :p: '('\ \ :bp: 'self'\ \ :p: ', '\ \ :n: 'o

\ :bp: 'self'\ \ :o: '.'\ \ :n: 'shortcut'\ \ :o: '='\ \ :nb: 'unicode'\ \ :
\ :bp: 'self'\ \ :o: '.'\ \ :n: 'settings'\ \ :o: '.'\ \ :n: 'setValue'\ \ :

\ :n: 'newmode'\ \ :o: '='\ \ :s: '""'\
\ :k: 'if'\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'q'\ \ :o: '.'\ \ :n: 'isChecked
\ \ :n: 'newmode'\ \ :o: '+='\ \ :s: 'q'\
\ :k: 'if'\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'b'\ \ :o: '.'\ \ :n: 'isChecked
\ \ :n: 'newmode'\ \ :o: '+='\ \ :s: 'B'\
\ :k: 'if'\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'd'\ \ :o: '.'\ \ :n: 'isChecked
\ \ :n: 'newmode'\ \ :o: '+='\ \ :s: 'd'\
\ :k: 'if'\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'e'\ \ :o: '.'\ \ :n: 'isChecked
\ \ :n: 'newmode'\ \ :o: '+='\ \ :s: 'e'\
\ :bp: 'self'\ \ :o: '.'\ \ :n: 'mode'\ \ :o: '='\ \ :n: 'newmode'\

\ :bp: 'self'\ \ :o: '.'\ \ :n: 'settings'\ \ :o: '.'\ \ :n: 'setValue'\ \ :
\ :bp: 'self'\ \ :o: '.'\ \ :n: 'settings'\ \ :o: '.'\ \ :n: 'sync'\ \ :p: '
```

And you need to load those settings and put them in your class, too:

```
\ :nd: '@classmethod'\
\ :k: 'def'\ \ :nf: 'loadConfig'\ \ :p: '('\ \ :bp: 'self'\ \ :p: ')': '\
\ \ :k: 'print'\ \ :s: 'SMARTY loadconfig'\ \ :p: ', '\ \ :bp: 'self'\ \ :
\ \ :k: 'if'\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'settings'\ \ :p: ': '\
\ \ :n: 'sc'\ \ :o: '='\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'settings'\ \ :o:
\ \ :k: 'if'\ \ :n: 'sc'\ \ :o: '.'\ \ :n: 'isValid'\ \ :p: '(): '\
\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'shortcut'\ \ :o: '='\ \ :nb: 'unicod
\ \ :n: 'mode'\ \ :o: '='\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'settings'\ \ :
\ \ :k: 'if'\ \ :n: 'mode'\ \ :o: '.'\ \ :n: 'isValid'\ \ :p: '(): '\
\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'mode'\ \ :o: '='\ \ :nb: 'unicode'
```

Making it Work

And yes, you need to make it do something useful. The plugin has access to a “client” which is Marave’s main window. Everything is available there, somewhere ;-)

```
\ :k: 'def'\ \ :nf: 'run'\ \ :p: '('\ \ :bp: 'self'\ \ :p: ')': '\
\ \ :k: 'print'\ \ :s: 'running smarty plugin'\
\ \ :n: 'text'\ \ :o: '='\ \ :nb: 'unicode'\ \ :p: '('\ \ :bp: 'self'\ \ :o:
\ \ :n: 'prog'\ \ :o: '='\ \ :n: 'QtGui'\ \ :o: '.'\ \ :n: 'QProgressDialog'\
\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'client'\ \ :o:
\ \ :mi: '0'\ \ :p: ', '\ \ :nb: 'len'\ \ :p: '('\
\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'client'\ \ :p:
\ \ :n: 'prog'\ \ :o: '.'\ \ :n: 'show'\ \ :p: '() '\
\ \ :n: 'output'\ \ :o: '='\ \ :p: '[] '\
\ \ :k: 'for'\ \ :n: 'i'\ \ :p: ', '\ \ :n: 'l'\ \ :ow: 'in'\ \ :nb: 'enumera
\ \ :n: 'output'\ \ :o: '.'\ \ :n: 'append'\ \ :p: '('\ \ :n: 'unescape'\
\ \ :n: 'prog'\ \ :o: '.'\ \ :n: 'setValue'\ \ :p: '('\ \ :n: 'i'\ \ :p: ')
\ \ :n: 'QtGui'\ \ :o: '.'\ \ :n: 'QApplication'\ \ :o: '.'\ \ :n: 'instan
\ \ :n: 'prog'\ \ :o: '.'\ \ :n: 'hide'\ \ :p: '() '\
\ \ :bp: 'self'\ \ :o: '.'\ \ :n: 'client'\ \ :o: '.'\ \ :n: 'editor'\ \ :o: '
```

And there it is, if you enable the smarty plugin, you can “fix” your quotes, dashes and ellipsis with a key combination :-)

Full source code here: <http://code.google.com/p/marave/source/browse/trunk/marave/plugins/smarty.py>

Still to be done: other ways to integrate plugins into the UI, buttons, panels, etc.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “marave”.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “marave”.

I had a small task today in Marave.. marave: <http://marave.googlecode.com>
. The goal was:

1. Fade in a widget
2. Set a variable
3. Fade in another one

It's important that things are done *in that order* and it's also important that the app remains interactive.

And here's the code to do that (simplified):

```

\ :k: 'def' \ \ :nf: 'fadein' \ \ :p: '(' \ \ :n: 'thing' \ \ :p: ', ' \ \ :n: 'target'
  \ :sd: '"""
  * thing is a QWidget
  * thing.proxy is a QGraphicsWidget
  * thendo is callable
  * target is the desired opacity
  """ \

\ :n: 'thing' \ \ :o: '.' \ \ :n: 'anim' \ \ :o: '=' \ \ :n: 'QtCore' \ \ :o: '.' \
\ :n: 'thing' \ \ :o: '.' \ \ :n: 'anim' \ \ :o: '.' \ \ :n: 'setDuration' \ \ :p:
\ :n: 'thing' \ \ :o: '.' \ \ :n: 'anim' \ \ :o: '.' \ \ :n: 'setStartValue' \ \ :p:
\ :n: 'thing' \ \ :o: '.' \ \ :n: 'anim' \ \ :o: '.' \ \ :n: 'setEndValue' \ \ :p:
\ :n: 'thing' \ \ :o: '.' \ \ :n: 'anim' \ \ :o: '.' \ \ :n: 'start' \ \ :p: '(' \
\ :n: 'thing' \ \ :o: '.' \ \ :n: 'anim' \ \ :o: '.' \ \ :n: 'finished' \ \ :o: '.' \
\ :k: 'if' \ \ :n: 'thendo' \ \ :p: ':' \
  \ :n: 'thing' \ \ :o: '.' \ \ :n: 'anim' \ \ :o: '.' \ \ :n: 'finished' \ \ :

```

And this is how you use it:

```

\ :k: 'def' \ \ :nf: 'later' \ \ :p: '():' \
  \ :n: 'avar' \ \ :o: '=' \ \ :n: 'avalue' \
  \ :n: 'fadein' \ \ :p: '(' \ \ :n: 'widget2' \ \ :p: ') ' \

\ :n: 'fadein' \ \ :p: '(' \ \ :n: 'widget1' \ \ :p: ', ' \ \ :n: 'thendo' \ \ :o: '='

```

Isn't that *lovely*? Having functions as first class objects means I can just take `later` as a closure, along with `widget2` and `avar`, which need only be defined in the local scope, and the call chain will work just as I wanted!

Yes, many other languages can do this, and in Javascript it's one of the most common tricks, but consider that PyQt is a wrapper for a C++ library!

I think this kind of usage shows the real added value PyQt brings to the table, it's not just that python avoids the boring compiles, or that you have the awesome standard library to use, but that the *language itself* enables you to do things that are not practical in C++.

In C++ the only way I can think of is creating a slot that's the equivalent of `later`, then chaining the signals... which means that this throwaway `later` becomes part of the interface of a class!

I would have to define `later` somewhere else on the file, separate from its only usage (maybe even inlined in the header file).

Even then, that's not equivalent: `avalue` may be something that was only available before the first call to `fadein`, (for example, the time of the first `fadein`): I would have

to create a place to store it, make it reachable by `later`... and what happens if you try to do this again while the first fadein is in progress?... it gets hairy.

Programming is like a slap in the face sometimes... you realize that things you use without even noticing are far from trivial.

So, remember young padawan: you can choose your tools. Choose wisely.

Version 0.6 of Marave, my peaceful, fullscreen text editor is now available at the usual place: <http://marave.googlecode.com>

New stuff:

- Syntax highlighter
- Plugins
- Bugs fixed
- Nicer animations
- Code cleanup

Gratuitous screenshot:

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "ohloh".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "ohloh".
```

Here's what Ohloh.. ohloh: <http://www.ohloh.net/p/marave>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "marave".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "marave".
```

has to say about the cost of developing Marave.. marave: <http://marave.googlecode.com>

Really, Marave is *maybe* a month of part-time programming. How could that possible be U\$71355, or "1 Person Years"?

Is this garbage the best we have to estimate costs? If that's the case, then whenever you see something about "Open source program X would take Y years and cost Z dollars to write", cut it down by a factor of 10 or more.

Here's what Marave really costed to develop:

- Nothing.

Ok, here's what it would have costed if I had charged for it:

I am guessing about 100 hours of my time. At my "I hope they pay me this much" rate of U\$S 40/hour , that's U\$S 4000, which means Ohloh is off by 1600%.

OTOH, for that much freelance work I would not charge you the full rate, I would probably end charging you more like U\$S20/hour which would make Ohloh's guess over 3000% too high.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "see it for yourself".

In conclusion: if you like my code (and hey, you can [see it for yourself](#).. see it for yourself: <http://www.ohloh.net/accounts/ralsina>), hire me, I am incredibly cheap, or amazingly fast!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "packt publishing".

The nice fellows at [Packt Publishing](#).. packt publishing: <http://www.packtpub.com/>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "grok 1.0 web development".

have sent me a copy of [Grok 1.0 Web Development](#).. grok 1.0 web development: http://www.packtpub.com/grok-1-0-web-development/book?utm_source=lateral.netmanagers.com&utm_medium=referral
by Carlos de la Guardia to review.

I am reading it and expect to write about it in a few days (BTW: nice so far! My proposed slogan: it's like zope, without the Java [1]), and here is a sample chapter so you can see what it's about. It's easy to read standalone, too:

[Chapter 5: Forms](#).. _chapter 5: forms: <http://www.packtpub.com/files/7481-grok-1-0-Web-development-sample-chapter-5-forms.pdf>

Having had to deal this week with the brokenness that's Django oldforms (legacy site, don't even ask), the Grok way of dealing with forms is refreshing ;-)

[1] Yes, of course I know Zope had no Java in it, it just *felt* like Java ;-)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "marave".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "marave".

Writing an editor is reinventing the wheel. I know that. I tell myself [Marave](#).. marave: <http://marave.googlecode.com>

is a fine wheel, with distinct features, and I think that is true, but, if you are reinventing the wheel, there's no need to reinvent the axle and the spoke, too.

So, I refactored the stuff that I think a text editor *must* provide into a nice library, so the next time someone *must* invent a wheel, he can use Marave's neat spokes and axles:

So, introducing **Editor**, the most-obviously named class ever! It's a text editing widget for PyQt with extra features, which you can use as a drop-in replacement for a QTextEdit or QPlainTextEdit.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "inside marave's svn".

Right now, it lives [inside Marave's SVN](http://code.google.com/p/marave/source/browse/trunk).. inside marave's svn: <http://code.google.com/p/marave/source/browse/trunk>

but it may even move out someday.

Here are its features:

- Syntax highlighting

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "source highlight qt".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "source highlight qt".

And I don't mean "in theory", like QTextEdit and company do! Editor can highlight a bunch of languages, because it uses GNU source highlight via Lorenzo Bettini's [Source Highlight Qt](http://srchiliteqt.sourceforge.net/).. source highlight qt: <http://srchiliteqt.sourceforge.net/>

- Spell checking

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "pyenchant".

If you have [PyEnchant](http://www.rfk.id.au/software/pyenchant/).. pyenchant: <http://www.rfk.id.au/software/pyenchant/>

installed and the right dictionaries, it will do online spellchecking.

- Search and Search+Replace widgets

The Editor class can give you nice widgets for search or search and replace already hooked with the editor widget, so you can add them to your app's UI easily.

- new/open/save/saveas methods:

Don't implement opening/saving, etc yourself! That's *always* the same code!

Hopefully this will be helpful for someone else :-)

I have been thinking on what I *really really* want in a word processor. And then what would it take to create such a thing.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "here if you can't see it".

A few minutes of playing have led me the way of this teaser (video [here if you can't see it..](#) here if you can't see it: <http://www.youtube.com/watch?v=gWOSIIgSIW0>):

Could something come out of it? Who knows.

I just uploaded version 0.7 of Marave, my fullscreen text editor to <http://marave.googlecode.com>

Marave is a "relaxing" text editor inspired by ommwriter, DarkRoom and many others. It combines a spartan fullscreen UI with a vanishing UI, which gets out of the way of your text.

It supports syntax highlighting, inline spellchecking, background music, audible keyboard feedback, themes, is extensible via plugins, and much more.

Here's a screenshot:

There are no major new features in 0.7, but there are important internal changes and some major bugs fixed:

- Fixed bug that broke opening files if you had no spellchecker
- Implemented basic RTL language support
- Several other minor fixes
- Refactored the editor component so it can be reused

It isn't so much, but it's a lot to me :-)

Thanks to all the readers!

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "kuatia".

As mentioned previously, I am hacking a bit on a proof-of-concept word processor. Right now, it's hosted on googlecode and called *kuatia*.. *kuatia*: <http://kuatia.googlecode.com>

Now, it is far from being useful for anything, but... it can do nested itemized and bulleted lists.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "restructured text".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “restructured text”.

Here’s a screenie of the editor and the PDF output it produces via [reStructured Text](#)..
restructured text: <http://docutils.sf.net>

:

Personally I think that’s not too bad.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “why can’t programmers.. program?”.

If you haven’t read Jeff Atwood’s [Why Can’t Programmers.. Program?.. why can’t programmers.. program?](#): <http://www.codinghorror.com/blog/2007/02/why-cant-programmers-program.html>

go ahead, then come back.

Now, are you scared enough? Don’t be, the problem there is with the hiring process.

Yes, there are lots of people who show up for programming positions and can’t program. That’s not unusual!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “something i read by joel spolsky”.

It’s related to [something I read by Joel Spolsky.. something i read by joel spolsky](#):
<http://www.joelonsoftware.com/items/2005/01/27.html>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “stackoverflow.com”.

(amazingly, Jeff Atwood’s partner in [stackoverflow.com](#).. [stackoverflow.com](#):
<http://www.stackoverflow.com>

).

Suppose you are a company that tries to hire in the top 1% of programmers, and have an open position.

You get 100 applicants. Of those, 99 can’t program. 1 can. You hire him.

Then the company next door needs to do the same thing. They may get 100 applicant. 99 can’t program ... and probably 80 of them are the same the previous company rejected before!

So no, hiring the best 1 out of 100 is not a way to get a programmer in the top 1% at all, that’s just statistics intuition getting the worse of you.

You don’t want to hire in the top 1% of applicants, you want to hire in the top 1% of **programmers**. Different universes.

These two things are the two sides of the same coin. 99% of applicants are useless, that’s why they are applicants, because **they can’t get a job** and they can’t get a job **because they are useless as programmers**.

So, judging programmers by the standard of the applicants you get is like judging quality of a restaurant by licking its dumpster.

But now, having taken care of this, how do you find a programmer that can actually program?

Easy! Find one that has programs he can show you!

I would never hire a programmer that can't show me code. There *must be* something wrong with him, because programmers **write programs**.

That's just what we do. If we didn't what kind of programmers would we be?

Let's see some obvious objections to my argument:

- 1) He wrote code for his previous employer and can't show it.

So, he did. What else has he written? Some open source code? Maybe snippets in a blog? Answers in stackoverflow?

Nothing? He has written **nothing** he was not paid to write? He is not who I want. He only programs for money, he lacks passion for programming, he doesn't enjoy it. He is probably not very good at it.

- 2) He is just finishing college, he has not written much code *yet!*

Why? What stopped him? He has been learning to program for years, what has he done with the knowledge he has been receiving? Saving it for his 25th birthday party? He has not **practiced his craft?** Not the programmer I need.

But having him show you code is not enough, of course. It also has to be good code, if you are serious about hiring excellent programmers.

So here's some bonus criteria:

- 1) Check the languages he uses. If he codes COBOL for pleasure, he may or may not be what you want.
- 2) Open source == bonus points: it means he is not ashamed of his code, plus it makes his credentials trivial to verify.
- 3) If he leads a project with multiple contributors and does a good job he is half way to becoming a programmer/manager, so huge bonus points.
- 4) Projects with long commit histories show responsibility and a level head.
- 5) Development mailing lists let you gauge his personality. Is he abrasive? Is he thin-skinned? Is he annoying?

Then there's the obvious stuff, references from previous employers, interviews, exercises, and such. But those are the least important filters, the most important thing is that *he must be able to code*. And showing you his code is the way to do it.

I've just uploaded the 0.13 version of rst2pdf, a tool to convert reStructured text to PDF using Reportlab to <http://rst2pdf.googlecode.com>

rst2pdf supports the full reSt syntax, works as a sphinx extension, and has many extras like limited support for TeX-less math, SVG images, embedding fragments from PDF documents, True Type font embedding, and much more.

This is a major version, and has lots of improvements over 0.12.3, including but not limited to:

- New TOC code (supports dots between title and page number)

- New extension framework
- New preprocessor extension
- New vectorpdf extension
- Support for nested stylesheets
- New headerSeparator/footerSeparator stylesheet options
- Foreground image support (useful for watermarks)
- Support transparency (alpha channel) when specifying colors
- Inkscape extension for much better SVG support
- Ability to show total page count in header/footer
- New RSON format for stylesheets (JSON superset)
- Fixed Issue 267: Support :align: in figures
- Fixed Issue 174 regression (Indented lines in line blocks)
- Fixed Issue 276: Load stylesheets from strings
- Fixed Issue 275: Extra space before lineblocks
- Fixed Issue 262: Full support for Reportlab 2.4
- Fixed Issue 264: Splitting error in some documents
- Fixed Issue 261: Assert error with wordaxe
- Fixed Issue 251: added support for rst2pdf extensions when using sphinx
- Fixed Issue 256: ugly crash when using SVG images without SVG support
- Fixed Issue 257: support afigure when using sphinx/pdfbuilder
- Initial support for graphviz extension in pdfbuilder
- Fixed Issue 249: Images distorted when specifying width and height
- Fixed Issue 252: math directive conflicted with sphinx
- Fixed Issue 224: Tables can be left/center/right aligned in the page.
- Fixed Issue 243: Wrong spacing for second paragraphs in bullet lists.
- Big refactoring of the code.
- Support for Python 2.4
- Fully reworked test suite, continuous integration site.
- Optionally use SWFtools for PDF images
- Fixed Issue 231 (Smarter TTF autoembed)

- Fixed Issue 232 (HTML tags in title metadata)
- Fixed Issue 247 (printing stylesheet)

system-message
WARNING/2 in `everything.txt`, line 3
 Duplicate explicit target name: “pyar”.

system-message
WARNING/2 in `everything.txt`, line 3
 Duplicate explicit target name: “pyar”.

Yesterday in the [PyAr](http://www.python.org.ar). pyar: <http://www.python.org.ar>

mailing list a “silly” subject appeared: how would you translate spanish to rosarino?

For those reading in english: think of rosarino as a sort of pig latin, where the tonic vowel X is replaced with XgasX, thus “rosario” -> “rosagasario”.

In english this would be impossible, but spanish is a pretty regular language, and a written word has enough information to know how to pronounce it, including the location of the tonic vowel, so this is possible to do.

system-message
WARNING/2 in `everything.txt`, line 3
 Duplicate explicit target name: “the thread”.

Here is [the thread](http://mx.grulic.org.ar/lurker/message/20100317.165234.5374d8bb.es.html).. the thread: <http://mx.grulic.org.ar/lurker/message/20100317.165234.5374d8bb.es.html>

It’s looong but, final outcome, since I am a nerd, and a programmer and programmers **program**, I wrote it.

What surprised me is that as soon as I started doing it, this throwaway program, completely useless...I did it cleanly.

- I used doctings.
- I used doctests.
- I was careful with unicode.
- Comments are adequate
- Factoring into functions is correct

A year ago I wouldn’t have done that. I think I am finishing a stage in my (slow, stumbling) evolution as a programmer, and am coding better than before.

I had a tendency to, since python lets you write fast, write fast and dirty. Or slow and clean. Now I can code fast and clean, or at least cleaner.

BTW: this would be an excellent exercise for “junior” programmers!

- It involves string manipulation which may (or may not) be handled with regexps.
- Using tests is very quickly rewarding

- Makes you “think unicode”
- The algorithm itself is not complicated, but tricky.

system-message

WARNING/2 in `everything.txt`, line 3
 Duplicate explicit target name: “`gaso.py`”.

BTW: here is the (maybe stupidly overthought) program, `gaso.py`. `gaso.py`: <http://lateral.netmanagers.com.ar/static>

```

\ :c: `# -*- coding: utf-8 -*-`

\ :sd: `"""
Este es el módulo gasó.

Este módulo provee la función gasear. Por ejemplo:

>>> gasear(u'rosarino')
u'rosarigasino'
"""`

\ :kn: `import` \ \ :nn: `unicodedata`
\ :kn: `import` \ \ :nn: `re`

\ :k: `def` \ \ :nf: `gas` \ \ :p: `( ` \ \ :n: `letra` \ \ :p: `):` \
\ :sd: `'''dada una letra X devuelve XgasX
excepto si X es una vocal acentuada, en cuyo caso devuelve
la primera X sin acento

>>> gas(u'a')
u'agasa'

>>> gas (u'\xf3')
u'ogas\\xf3'

'''`

\ :k: `return` \ \ :s: `u` \ \ :si: `%s` \ \ :s: `gas` \ \ :si: `%s` \ \ :s: `

\ :k: `def` \ \ :nf: `umuda` \ \ :p: `( ` \ \ :n: `palabra` \ \ :p: `):` \
\ :sd: `'''
Si una palabra no tiene "!":
Reemplaza las u mudas de la palabra por !

Si la palabra tiene "!":
Reemplaza las !" por u

>>> umuda (u'queso')
u'q!eso'

```

```

>>> umuda (u'q!eso')
u'queso'

>>> umuda (u'cuis')
u'cuis'

'''\

\ :k:'if'\ \ :s:'!''\ \ :ow:'in'\ \ :n:'palabra'\ \ :p:':'\
\ \ :k:'return'\ \ :n:'palabra'\ \ :o:'.'\ \ :n:'replace'\ \ :p:'('\
\ :k:'if'\ \ :n:'re'\ \ :o:'.'\ \ :n:'search'\ \ :p:'('\ \ :s:''([qg])
\ \ :k:'return'\ \ :n:'re'\ \ :o:'.'\ \ :n:'sub'\ \ :p:'('\ \ :s:''
\ :k:'return'\ \ \ :n:'palabra'\

\ :k:'def'\ \ :nf:'es_diptongo'\ \ :p:'('\ \ :n:'par'\ \ :p:'):\'
\ :sd:''''Dado un par de letras te dice si es un diptongo o no

>>> es_diptongo(u'ui')
True

>>> es_diptongo(u'pa')
False

>>> es_diptongo(u'ae')
False

>>> es_diptongo(u'ai')
True

>>> es_diptongo(u'a')
False

>>> es_diptongo(u'cuis')
False

'''\

\ :k:'if'\ \ :nb:'len'\ \ :p:'('\ \ :n:'par'\ \ :p:')'\ \ :o:'!='\
\ \ :k:'return'\ \ \ :bp:'False'\

\ :k:'if'\ \ :p:'('\ \ :n:'par'\ \ :p:'['\ \ :mi:'0'\ \ :p:]'\ \ :ow:
\ :p:'('\ \ :n:'par'\ \ :p:'['\ \ :mi:'1'\ \ :p:]'\ \ :ow:'in'\ \ :s
\ \ :k:'return'\ \ \ :bp:'True'\
\ :k:'return'\ \ \ :bp:'False'\

\ :k:'def'\ \ :nf:'elegir_tonica'\ \ :p:'('\ \ :n:'par'\ \ :p:'):\'
\ :sd:''''Dado un par de vocales que forman diptongo, decidir cual de l
dos es la tónica.

>>> elegir_tonica(u'ai')

```

```

0

>>> elegir_tonica(u'ui')
1
'''\
\ :k:'if'\ \ :n:'par'\ \ :p:'[\ \ :mi:'0'\ \ :p:]'\ \ :ow:'in'\ \
\ :k:'return'\ \ :mi:'0'\
\ :k:'return'\ \ :mi:'1'\

\ :k:'def'\ \ :nf:'gasear'\ \ :p:'(\ \ :n:'palabra'\ \ :p:'):\
\ :sd:'""'
Convierte una palabra de castellano a rosarigasino.

>>> gasear(u'rosarino')
u'rosarigasino'

>>> gasear(u'pas\xel')
u'pasagas\xel'

Los diptongos son un problema a veces:

>>> gasear(u'cuis')
u'cuigasis'

>>> gasear(u'caigo')
u'cagasaigo'

Los adverbios son especiales para el castellano pero no
para el rosarino!

>>> gasear(u'especialmente')
u'especialmegasente'

'''\
\ :c:'#from pubdb import set_trace; set_trace()'\

\ :c:'# Primero el caso obvio: acentos.\
\ :c:'# Lo resolvemos con una regexp'\

\ :k:'if'\ \ :n:'re'\ \ :o:'.'\ \ :n:'search'\ \ :p:'(\ \ :s:'u'[\ \
\ :k:'return'\ \ :n:'re'\ \ :o:'.'\ \ :n:'sub'\ \ :p:'(\ \ :s:'u'

\ :c:'# Siguiente problema: u muda'\
\ :c:'# Reemplazamos gui que qui que por g!i g!e q!i q!e'\
\ :c:'# y lo deshacemos antes de salir'\
\ :n:'palabra'\ \ :o:'='\ \ :n:'umuda'\ \ :p:'(\ \ :n:'palabra'\ \ :p:

\ :c:'# Que hacemos? Vemos en qué termina'\

```

```

\ :k:'if'\ \ :n:'palabra'\ \ :p:'[\ \ :o:'-\ \ :mi:'1'\ \ :p:]'\
\ :c:'# Palabra grave, acento en la penúltima vocal'\
\ :c:'# Posición de la penúltima vocal:'\
\ :n:'pos'\ \ :o:'='\ \ :nb:'list'\ \ :p:'(\ \ :n:'re'\ \ :o:'.'\
\ :k:'else'\ \ :p:':'\
\ :c:'# Palabra aguda, acento en la última vocal'\
\ :c:'# Posición de la última vocal:'\
\ :n:'pos'\ \ :o:'='\ \ :nb:'list'\ \ :p:'(\ \ :n:'re'\ \ :o:'.'\

\ :c:'# Pero que pasa si esa vocal es parte de un diptongo¿\

\ :k:'if'\ \ :n:'es_diptongo'\ \ :p:'(\ \ :n:'palabra'\ \ :p:'[\ \ :
\ :n:'pos'\ \ :o:'+='\ \ :n:'elegir_tonica'\ \ :p:'(\ \ :n:'pala
\ :k:'elif'\ \ :n:'es_diptongo'\ \ :p:'(\ \ :n:'palabra'\ \ :p:'[\ \
\ :n:'pos'\ \ :o:'+='\ \ :n:'elegir_tonica'\ \ :p:'(\ \ :n:'pala

\ :k:'return'\ \ :n:'umuda'\ \ :p:'(\ \ :n:'palabra'\ \ :p:'[:'\ \ :n

\ :k:'if'\ \ :n:'__name__'\ \ :o:'=='\ \ :s:'"__main__"\ \ :p:':'\
\ :kn:'import'\ \ :nn:'doctest'\
\ :n:'doctest'\ \ :o:'.'\ \ :n:'testmod'\ \ :p:'()' \

```

It's my pleasure to announce that I just uploaded rst2pdf 0.14 to the site at <http://rst2pdf.googlecode.com>. Rst2pdf is a program and a library to convert restructured text directly into PDF using Reportlab.

It supports True Type and Type 1 font embedding, most raster and vector image formats, source code highlighting, arbitrary text frames in a page, cascading stylesheets, the full restructured text syntax and much, much more.

It also includes a sphinx extension so you can use it to generate PDFs from documents built with Sphinx.

In case of problems, please report them in the Issue tracker (<http://code.google.com/p/rst2pdf/issues/list>) or the mailing list (<http://groups.google.com/group/rst2pdf-discuss>)

This release fixes several bugs and adds some minor features compared to 0.13.2. Here are some of the changes:

- Fixed Issue 197: Table borders were confusing.
- Fixed Issue 297: styles from default.json leaked onto other syntax highlighting stylesheets.
- Fixed Issue 295: keyword replacement in headers/footers didn't work if ####Page### and others was inside a table.
- New feature: oddeven directive to display alternative content on odd/even pages (good for headers/footers!)
- Switched all stylesheets to more readable RSON format.
- Fixed Issue 294: Images were deformed when only height was specified.
- Fixed Issue 293: Accept left/center/right as alignments in stylesheets.

- Fixed Issue 292: separate style for line numbers in codeblocks
- Fixed Issue 291: support class directive for codeblocks
- Fixed Issue 104: total number of pages in header/footer works in all cases now.
- Fixed Issue 168: linenos and linenothreshold options in Sphinx now work correctly.
- Fixed regression in 0.12 (interaction between rst2pdf and sphinx math)
- Documented extensions in the manual
- Better styling of bullets/items (Issue 289)
- Fixed Issue 290: don't fail on broken images
- Better font finding in windows (patch by techtonik, Issue 282).
- Fixed Issue 166: Implemented Sphinx's hlist (horizontal lists)
- Fixed Issue 284: Implemented production lists for sphinx
- Fixed Issue 165: Definition lists not properly indented inside admonitions or tables.
- SVG Images work inline when using the inkscape extension.
- Fixed Issue 268: TOCs shifted to the left on RL 2.4
- Fixed Issue 281: sphinx test automation was broken
- Fixed Issue 280: wrong page templates used in sphinx

Enjoy!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "packt publishing".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "packt publishing".

The people at [Packt Publishing](http://www.packtpub.com/).. packt publishing: <http://www.packtpub.com/>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "grok 1.0 web development".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "grok 1.0 web development".

has been kind enough to send me a copy of [Grok 1.0 Web Development](#)..
grok 1.0 web development: http://www.packtpub.com/grok-1-0-web-development/book?utm_source=lateral.net
by Carlos de la Guardia.

I have now read it (sorry it took so long!) and here's my review.
It's a well-written book. The exposition is clear and the author knows the subject and what he wants to say.

I was impressed by the straightforward approach of Grok when compared to what I dimly remember of ancient Zope experiences. The idea of a modern, agile, *simple* framework over Zope's admittedly powerful foundation has a lot of merit.

OTOH, I am not really convinced that I should drop Django for the small web projects I have scheduled, but if you are looking for a framework, and are not heavily invested in another, Grok is certainly worth checking out.

And, if you want to learn Grok, this book does the trick.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "sample chapter".
```

You can read a [sample chapter](#).. sample chapter: <http://www.packtpub.com/files/7481-grok-1-0-Web-development>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "more information about the book".
```

or get [more information about the book](#).. more information about the book:
http://www.packtpub.com/grok-1-0-web-development/book?utm_source=lateral.netmanagers.com&utm_medium

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "a day of blogging to celebrate the achievements of women in technology and science".
```

Today, March 24th is Ada Lovelace day, [a day of blogging to celebrate the achievements of women in technology and science](#) a day of blogging to celebrate the achievements of women in technology and science.:
<http://findingada.com/>

. I am taking the liberty to tag this as python so it appears in the right planets, but that's just to promote Ada Lovelace day. Sorry 'bout that.

I will write about the only person who ever taught me programming, Claudia. I was young, so the earth was still lukewarm, the day we saw the dinosaur.

I was just a green sophomore in the School of Chemical Engineering where, paradoxically I would never take a chemistry class, being an applied math student and all that, and at the time "personal computers" were a novelty, a toy of the upper middle class.

We had spent the first two months of the semester learning how to program the obvious way: writing assembler for a fictional machine on paper by hand, when Claudia broke the news, we were going to see a real computer.

No, not a PC, not even an XT, but a **real** computer, the one **real** computer in all the university, and you could hear the type switching to bold as she spoke about it. Sadly

it was not as **real** as the one at the research facility (A MiniVAX!) but it was a real enough PDP.

We would not be allowed to actually *use it* until the following year, but ... well, it was still something special.

I had been programming for years, even for a year before I saw my first (seriously not real) computer, I had followed BASIC programs in my head for days, imagining the space invaders float on the screen of my mind, and stepped into writing machine code inside REM statements in my Timex Sinclair 1000 onto the luxury of a C64, but never noone had taught me anything.

Our small class (maybe 10 students) spent endless hours doing things like traverse a matrix, first by rows, thn by columns, then in a spiral from the top-left, writing programs that followed our endless source of algorithms, the numerical solutions guide.

First assembler, then Fortran, we learned.

She was my Mr. Miyagi, I was a heterosexual Ralph Macchio, and I figured out the most important thing about programming: I was awful at it.

Over the next 20 years that situation has been slowly improving, but I never again had someone teach me programming. Claudia had already taught me everything I needed to know, that code can always improve, that there's more than one way to skin a cat.

That the dinosaur was **real** and that some day soon *my* computer would be faster and nicer than the dinosaur was then, and that programming was cool, and that if I could find a way to draw a polynomial graph *horizontally* on a printer without ever having the whole graph in memory (it didn't fit), those future computers would do awesome things, and that *I* was one of the many who would help bring that to reality.

That talking about code was fun in itself, that you could make a modest living and be happy about it, that you could in any case make jigsaw puzzles in your spare time and keep on teaching or whatever.

And later the dinosaur's bones were scavenged into a line of racks holding routers, and its glass terminals are destroyed, and the gold in its teeth was stolen and the rare bus cables sold, and its circuits scrapped, but I saw the dinosaur alive, and Claudia taught me how to make it jump, and for that, I will always be grateful.

Since I expect Android on tablets to be a big thing in 2010, I am experimenting with the closest thing I can get: Android in my eee 701 Surf 4G:

I got the testing Android 2.0 image from <http://android-x86.org>. I had the 1.6 "stable" one but it was... well, it worked awful (half the key combos or menu options caused it to crash, reboot or otherwise autocombust).

So... how is it working? Slow, but it has potential!

The bad:

- It boots quite fast... but my tricked full Arch Linux install boots faster.
- It works *sloooooow*, you can see individual letters when you type in the search gadget. I read this is a temporary problem, though.
- I am getting a "castrated" experience because the open android app stores are not as well stocked as the official android marketplace (and come on, why the heck can't I get free apps from there???)

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "radiotray".

I see obvious holes in the app landscape that I suppose are well covered in the market (like, is there a [RadioTray](http://radiotray.sf.net).. radiotray: <http://radiotray.sf.net>

replacement?)

No text editor?

No semi-decent word processor? Not even one that generates HTML?

- The web browser is pathetic. It may be nice for a phone, but for a “real” system? It’s awful. You get the mobile versions of all sites (obviously) and many don’t let you switch to the real ones! (even google does that, for Google Reader), and of course, no flash.
- The email app is *terrible*. You can’t not-top-post!!!! “In-Reply-To” is off-spec!
- The WiFi settings are way too hidden. They should pop if you click on the wifi icon.

The good:

- It shuts down incredibly fast.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “aldiko book reader”.

Some apps are quite nice, specially the [Aldiko book reader](http://www.aldiko.com/).. aldiko book reader: <http://www.aldiko.com/>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “fbreader”.

is awesome (and I can share the ePub books with [fbReader](http://www.fbreader.org).. fbreader: <http://www.fbreader.org> on the arch linux side.

- The included SSH client has great ideas.
- I love the “all your data is in the SD” approach. I do the same thing with Linux. In fact, I have the same exact data organization now on both OSs :-)
- The home screen with the sliding app drawer: nice
- The “grabbable” system notifications on the top bar: very nice
- The “use the menu key to get the menu” thing? genius ;-)
- The “everything fullscreen all the time”, thing? works on this screen.
- App installation is a solved problem here.
- I know I will be able to get Qt working native... can’t wait!

I am not sold yet, Arch is just so much faster right now, and it can do so much more, but...

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "touchscreen".

I am getting a touchscreen.. touchscreen: <http://www.dealextreme.com/details.dx/sku.16124>

for it, so I can experience it more the way it's meant to be experienced.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "makers".

I am using it a lot to read at night in bed (Just finished *Makers..* makers: <http://www.goodreads.com/book/show/6422238-makers>, read it, it's cool!).

- I am using it for casual mail **reading** (I refuse to reply with that broken app).
- It's a pretty nice alarm clock, so it's becoming my bedside OS.

I'll write another report once I have the touch screen or a new (hopefully faster!) version running.

I am just not writing here. I am writing a book instead.

What book am I writing? A book about python programming, of course! It's called "Python No Muerde" (Python Doesn't Bite) and it's in spanish.

Now, I am the first to admin: I am not a great programmer. And I am not a great writer. But I have lots of things to say. If I can organize them correctly, they even make sense sometimes!

So, I am giving this write-long-stuff thing a try.

Of course since I am an open source nerd, I can't do things the usual way, therefore, the book is under Creative Commons. And because I am a programmer, I hacked together a (if I may say so myself) decent structure to handle book-writing.

1. I write in restructured text
2. I use `rst2pdf` to create PDFs both of individual chapters and the whole thing.
3. I use `rest2web` to create a website
4. I use `mercurial` (at `googlecode`) to handle revision control and history.
5. I use `make` to control rebuilding of chapters when code changes, or images get updated, etc.

Of course it's more complicated than that, the PDFs are in the site, the site is uploaded via `rsync`, the uploads and rebuilds are triggered by `hg` push, and so on.

In any case, I may post a few times about how this whole thing works, here is the output of the machinery:

<http://nomuerde.netmanagers.com.ar>

I am writing a book. And I am writing a chapter about UI design. And why not use the Internet?

So, go ahead and tell me all that's wrong with this dialog!

For example, I don't like the dead space at the bottom-left, the different-size of the "Close" button, and the misalignment of the icons.

Are those valid concerns? Are there many more? Would you do it completely different?

The book is open source, and available at <http://nomuerde.netmanagers.com.ar> (In spanish, sorry!)

I need to do it, that is.

I am hacking the restructured text editor out of Notty, so it can be used in other places.

The bad news, of course, is that because I didn't do it at the beginning, the current implementation is like an inkblot running through the rest of the code, tainting everything, which is hard to remove.

After that, I have to do the same thing with the "tree-thing-that drags-drops-and-knows-how-to-save-itself" (TM), and I will have three rather powerful widgets (Py-TextBrowser is the third: a QTextBrowser that does HTTP and images, you can get that one already).

Maybe I should turn these into some sort of package, too.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "my 89 cents bluetooth dongle".
```

A few days ago I finally got [my 89 cents bluetooth dongle](http://www.dealextreme.com/details.dx/sku.11866).. my 89 cents bluetooth dongle: <http://www.dealextreme.com/details.dx/sku.11866>

(now \$1.85, but still with free shipping from china!) and got a bunch of pictures I had in my phone.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "is crap".
```

The quality is crap because my phone [is crap](http://www.flickr.com/photos/ralsina/4476747320/sizes/m/).. is crap: <http://www.flickr.com/photos/ralsina/4476747320/sizes/m/>, but trust me, there *must* be one thing here you have never seen before.

Here they are: weird stuff that made me take out my phone and grab a picture, with explanations.

Titanic

This, from Mar del Plata, is the most badass popup book I ever saw.

Closed:

And open:

I'm Mark Shuttleworth!

In a free software event in Buenos Aires, Canonical's boss and former space cargo was supposed to deliver the keynote. He canceled at the last minute. So Maddog Hall offered to replace him... in character.

Someone found a really, really awesome (and/or crappy!) astronaut costume, and Maddog gave a keynote shouting "I'm Mark Shuttleworth! I'm an astronaut!" and claiming to have come from the future to examine some slides recently found, written by some unknown dude named Maddog. Really funny stuff.

Python vs. Ruby

Same event, take a look:

Yes, I swear they are taken with less than 10 seconds of one another.

Butter

I was buying groceries in San Isidro's Disco supermarket. Yes, usually buying a large package of butter is cheaper per kilo than a small one. But here, a 200g package costed almost the same as a 100g! That's just stealing money from those who don't use much butter. Me? I'm not at risk.

Visa discount!

This was a shop in Avenida Alem in Buenos Aires. It was unusual to see a "VISA is suspended, 20% discount" sign. Much more unusual was to see the small letters: "present your visa card". I mean, wasn't it suspended?

And then I saw the rest:

It says "present your visa card and pay using anything else".

That guy must really have been pissed off at Visa!

I got a bunch more for some other time.

I make a living working with free software (BTW, if you need sysadmins that know what they do, contact us: <http://www.netmanagers.com.ar>)

But that's not that big a deal, I *could* make a living doing something else. I am sure I would be competent at *something* else, even if I have no idea what that would be right now.

It does, however give me freedom to play, which is much more important. Therefore, this post is sort of a status update on things I play with. Not games, those are not really my thing, but things that I do for fun.

Yes, some of these may mean I am a very strange person.

I'll limit myself to the last couple of weeks or so.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "marave".

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "marave".

There's been sort of a bump in interest in [Marave](http://marave.googlecode.com).. marave: <http://marave.googlecode.com>, my distraction free editor and it's because it has been reviewed in Linux Journal! I have read the article (sadly I can't link to it) and it was a super positive review, here are some choice quotes:

"marave makes the dull world of text editing romantic and immersive with beautiful minimalism"

"... it doesn't just have minimalism and simplicity, it has minimalism and simplicity combined with beauty and a palpable design ethic. marave has soul, and I love that."

So thanks for the kind words to the author, and something I noticed: you ran into a big bug in marave and didn't notice :-)

The "cricket bat" icon (it's a screwdriver ;-) should show you the config dialog. However, it seems in Ubuntu (and maybe in other distros, I don't know) the config is not visible, and all you see is the text move around a bit. This is what he should have seen:

I have never been able to reproduce it, but I am going to install a Ubuntu VM just for this, so maybe soon.

On related news, marave was also reviewed in a german magazine a couple of months ago, and I have not been able to get a copy of the article. (BTW, isn't it reasonable to send a copy of these to the author of the program you are reviewing? Neither magazine even mentioned it to me!)

In any case, if anyone has this magazine and can tell me what the article about distraction-free editors say, you will make my day:

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "teaser".

Here's the article [teaser](http://www.linux-community.de/Internal/Artikel/Print-Artikel/LinuxUser/2010/05).. teaser: <http://www.linux-community.de/Internal/Artikel/Print-Artikel/LinuxUser/2010/05>

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "magazine issue".

And this seems to be the [magazine issue](https://shop.linuxnewmedia.de/store/eh01/EH10165/de).. magazine issue: <https://shop.linuxnewmedia.de/store/eh01/EH10165/de>
:

On new projects (yes, I always have new projects), I ran into this awesome blog post by Roger Alsing about *approaching Mona Lisa with just 50 polygons* <<http://rogersaling.com/2008/12/07/genetic-programming-evolution-of-mona-lisa/>> and being a nerd and having awesome programming tools at my command... I wrote a framework to test that kind of algorithms.

I called it *evoluto* and it's at <http://evoluto.googlecode.com>.

I only did a very simple algorithm, based on translucent triangles, but it did work... for some definition of work!

You can even **see** the local minimum that doesn't let her right eye form right :-)
Evoluto has a library of algorithms (currently empty) and you can edit and reapply on-the-fly the one you want, and see the generations change on-screen.

It would take some work to make it a polished program, but it does work.

I fiddled around a bit with creating a nice PDF presentation player but it's still very early:

In what's perhaps my most established project, rst2pdf. I have fixed a bunch of bugs, and a release is a bit overdue:

Issue 186: Text not wrapping around images when specified I fixed this by adding a new CLI option, so behaviour was not changed, but now you can have images with text flowing beside it. It will not look **great** but it works.

Issue 307: Replace directive doesn't replace text in header/footer in some cases This was an interesting problem! It was very entertaining.

Made it work with Sphinx 1.* There is a piece broken still, but what I fixed was not terribly hard.

Unbroken bookrest I need to work much more with this, but at least what worked before works again. If you don't know what bookrest is, it's a rst2pdf graphical frontend / word processor, here's a taste:

```
system-message
```

```
WARNING/2 in everything.txt, line 3  
Duplicate explicit target name: "my aur packages".
```

I also made several releases for [my AUR packages](http://aur.archlinux.org/packages.php?SeB=r).. my aur packages: <http://aur.archlinux.org/packages.php?SeB=r>
Plus I worked, and some other stuff. All in all, not a bad stretch.

```
system-message
```

```
WARNING/2 in everything.txt, line 3  
Duplicate explicit target name: "the usual free sample chapter".
```

The kind fellows at Packt Publishing have sent me a copy of this book to review, and I have been slow with it (flu doesn't help). In any case, while I finally start reading it, here's [the usual free sample chapter](https://www.packtpub.com/sites/default/files/7009-chapter-2-setting-up-shop-in-30-minutes_0.pdf).. the usual free sample chapter: https://www.packtpub.com/sites/default/files/7009-chapter-2-setting-up-shop-in-30-minutes_0.pdf

```
system-message
```

```
WARNING/2 in everything.txt, line 3  
Duplicate explicit target name: "the page at packt publishing".
```

For more information about this book, check [the page at Packt Publishing](https://www.packtpub.com/django-1-2-e-commerce-build-powerful-applications/book?utm_).. the page at packt publishing: https://www.packtpub.com/django-1-2-e-commerce-build-powerful-applications/book?utm_

This post is going to be in spanish. It's about the current public debate in Argentina about gay marriage. Yes, there are similar debates in other countries, but this is about the one in Argentina.

Bueno, hay un proyecto de ley acerca del matrimonio entre personas del mismo sexo¹² y la adopción por parejas homosexuales.

Dado que el nivel del debate por parte de los que se oponen es bastante patético, se me ocurrió escribir un poco acerca de cuáles cosas no deben plantear. ¿Porqué no hay que plantearlas? Porque pasás vergüenza, quedás como un nabo, y no te conviene. Te lo digo de onda.

1. No debés plantear argumentos que dicen cosas como “la palabra matrimonio viene de ...”.

Porqué no te conviene: Porque es una pelotudez. Si las leyes dependieran de la etimología de los términos te pagarían a fin de mes con sal¹³, las mujeres no deberían tener derecho a la propiedad¹⁴ ni ser testigos en juicios¹⁵ y siguen las firmas.

Qué parecés cuando lo decís: Un adolescente al que le encargaron defender una posición que no entiende y arranca buscando la palabra en el diccionario.

2. No debés plantar argumentos como “que los homosexuales tengan su propio matrimonio que se llame distinto”

Porqué no te conviene: Porque estás admitiendo que lo que querés es discriminar. Si la idea es que los homosexuales no son dignos de unirse en matrimonio con la persona que aman, pero los heterosexuales sí, estás planteando que no son dignos *por su condición de homosexuales*. Eso es discriminación.

Ojo, lejos de mí decir que no estás en tu derecho de ser un pajuerano que discrimina a otros porque te dan cosita, pero no es un motivo apto para el estado y la ley. Por lo tanto, no es pertinente al debate.

Ahora bien, si querés que el estado discrimine, no te escondas atrás de esta pavada, y decílo de frente. Poné tus “convicciones” sobre la mesa, y bancáte que te digan que sos un dinosaurio.

Qué parecés cuando lo decís: Una vieja retrógrada de provincia que tiene miedo de adónde iremos a parar si cualquiera puede hacer lo que se le antoje, al ver a su nieta irse a vivir con el novio sin casarse.

3. No debés plantear que no a la adopción porque “nadie me garantiza que los homosexuales sean buenos padres”

Porqué no te conviene: De vuelta, porque estás admitiendo que querés discriminar. ¿A los heterosexuales no les pedís que te garanticen nada, y a los homosexuales sí? Listo, estás discriminando. De nuevo, poné pelotas y/o ovarios y decílo de frente.

Qué parecés cuando lo decís: La esposa del reverendo Alegría en los Simpson: “¿Alguien pensará en los niños?!?”. También parecés un falso que descubre la posibilidad del abuso familiar cuando te conviene para defender una postura que ya tenías de antes.

4. No debés plantear que no al matrimonio porque “dios dice tal cosa”

Porqué no te conviene: Bueno, para empezar dios no existe, y si existiera, resulta que no vivimos en una teocracia¹⁶ así que nada, no importa, así viniera dios y lo dijera en persona en la plaza de mayo, **no importaría** porque él ni corta ni pincha en este tema.

Qué parecés cuando lo decís: Un tipo que no puede aceptar que su religión no es la ley, que no nos regimos por la biblia, y que si vamos a eso, cree que dios hizo que dos osas se morfaran a 42 nenes que le dijeron pelado a un profeta.

O sea, como poder judicial o legislativo, tu dios es flojito flojito, así que guardáelo.

5. No debés plantear que haría homosexuales a los chicos

Porqué no te conviene: Porque estás partiendo de que la inclinación sexual se hereda de los padres. Si es así, la homosexualidad sería rarísima, no? Ok, supongamos que tenés razón. Igual no importa, porque estás partiendo del supuesto de que un chico homosexual es algo malo y/o vergonzoso.

Todo lo que hacés es mover tu discriminación al pobre pibe, y pensás, “claro, va a ser gay y gente como yo le va a joder la vida”. Maestro, lo que tenés que hacer es lo contrario, es dejar de joderle la vida a los demás, y esto no es un problema.

Qué parecés cuando lo decís: Un metido que no puede dejar de preguntarse que hace con su pito y/o su culo cada uno. A ver si entra: cada uno es libre de hacer de su pito un culo y viceversa. Artículo 19 de la constitución y a llorar al zigurath.

6. No debés decir que el matrimonio tiene un fin reproductivo

Porqué no te conviene: Porque es *muy* pavote. ¿O acaso estás a favor de prohibir el matrimonio de hombres infértiles o mujeres postmenopáusicas?

Qué parecés cuando lo decís: Alguien que no se da cuenta de lo que dice.

7. No debés decir algo como “y que van a hacer cuando en el colegio hablen del día de la madre”

Porqué no te conviene: Porque es evidente que estás metiendo el brazo hasta el codo en el frasco de orejones y esto es lo último que hay. ¡Es patético! ¿O sea, que pasa si es huérfano de madre? ¿Y si la madre lo abandonó?

¿En qué caso podría esto ser traumático para el pibe? ¡Obvio! en el caso de que haya paparulos como vos que lo bardeen porque tiene dos papás / dos mamás.

Qué parecés cuando lo decís: Un falso que avisa a los demás que le van a pasar cosas feas... porque gente como él se va a encargar de que así sea.

8. No debés decir cosas como “el matrimonio homosexual degrada el matrimonio tradicional”

Porqué no te conviene: Porque es una pavada importante. ¿En qué te afecta? Yo recuerdo estar haciendo cola para pedir turno para el civil y había 10 parejas. Yo y mi mujer de la manito, dandonos cariño. Los otros 18 como si estuvieran pagando el inmobiliario, alguno discutiendo, todos con -20 de emoción.

Esa gente degrada el matrimonio, aquellos para los que el matrimonio es un trámite, y no la unión con la persona que quieren.

Qué parecés cuando lo decís: Un snob que no quiere que los cabecitas entren a su country.

9. No debés decir cosas como “si se permite esto, porqué no matrimonios con menores, caballos, o de a 4 personas!”

Porqué no te conviene: Porque tener sexo con menores es un delito, los caballos no son ciudadanos y no pueden firmar contratos, y de a 4 es un quilombo porque no hay manera de organizarse para las fiestas de fin de año.

Bah, sabés qué? De a 4 también debería ser legal. No se me ocurre porqué no. Si se quieren los 4, y nadie los obliga, que sean felices.

Qué parecés cuando lo decís: Alguien que opina que es lo mismo un menor que un caballo que un gay. O sea: un pelotudo.

No se me ocurre ningún otro argumento de esta calaña por ahora, pero si quieren, hay comentarios al pie.

Many years ago, I **loved** Homicide: Life on the streets. If you don't know about it, it was a series about Baltimore police detectives. It had some awesome writing, and one particular scene stuck in my mind until this day.

There was a thin detective called Munch (who looked like an Edvard Munch painting anyway), and he was interrogating someone for whatever reason. And then he gets angry, very, very angry, and this is what he says:

“You're saving your really good lies for some smarter cop, is that it? [...] I've been in murder police for ten years. If you're going to lie to me, you lie to me with respect.”

I have never forgotten that line because it speaks to me of one of the biggest annoyances in life, when people tell me transparent, obviously wrong or false things just because they expect me to go along, because calling them out on it is too much work, or impolite, or a waste of time, or make me look bad, or whatever.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “xkcd”.

¹²No, no es matrimonio homosexual. Los homosexuales se han casado toda la vida... con gente del otro género.

¹³Salario viene de “salarium” que sí, viene de sal.

¹⁴Patrimonio viene del latín y significa “los bienes que el **hijo** hereda de su **padre** y **abuelos**”. No, la hija no heredaba.

¹⁵Testimonio tiene la misma raíz que testículo y testamento.

¹⁶A pesar de que la constitución sigue conteniendo ese vergonzoso pasaje acerca de que la Argentina es una nación cristiana.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "xkcd".

And you know what? A big part of the time they get away with it, and that's fine. They get away with it because I also read `xkcd..` xkcd: <http://xkcd.com>, and there is this side of the story, too:

Alternatively, there are those who don't know what they are saying is crap. I am wrong in being pissed at them for treating me like a moron that won't notice the crap, when they themselves *have not noticed it themselves* and deserve guidance.

My last post was about such arguments when applied to the specific debate on gay marriage, but I could do the same thing with almost anything that's worth debating, and I bothered to write down a bunch of oppositions to gay marriage with explanations on why those are useless, and while I did that in part as stating my position *for* gay marriage, it is also because I *want* reasoned debate. I *want* good ideas from the other side. If they don't come hitting hard, what's the point? Where's the thrill in a duel of brains against unarmed adversaries?

I share Munch's indignation! So little do you care about truth that you push lies that don't even *look* like truth? Such feeble fakery that it flies away with a breeze? So little do you care about yourself, that you show the indignities of your mind in public? Respect me when you lie! Use the good lies!

If the other side can't push forward a cogent position, why is there a debate? Are we fighting against nothing, against inertia, against custom, against no rational thought? Are we fighting against *nothing*? That's just sad, because there is people on the other side, and it means **they** are fighting **for** nothing. Are you goddamn pushers of void? Advocates of hot air?

Yes, these people piss me off. Yes, I should take things easier. Yes, I look silly a lot. No, I don't give a damn.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "a book about python (in spanish)".

I am writing [a book about python \(in spanish\)..](http://nomuerde.netmanagers.com.ar) a book about python (in spanish): <http://nomuerde.netmanagers.com.ar>, and it says this:

Language

The arguments about how to write a technical book in spanish are eternal. That in Spain everything is translated. That in Argentina it isn't. That saying "cadena de caracteres" instead of string is bad for the environment.

Fortunately in this book we follow a better method that I hope other books adopt: it's write like I write. Not even a little bit different. I don't think it

even qualifies as spanish, maybe it's written in argentinian. If the readers of our former motherland are bothered by the style... they can translate it.

So today I got this message (sender anonymized because I don't do that):

I was reading your tutorial until I got to the paragraph where you complain or seem to be bothered that in Spain we translate everything and in Argentina nothing. Well, as a reader from the former motherland tell you [sic] that we like writing in our language and that not only am I not going to bother translating from argentinian to spanish, but that I am not even going to read your horrible argentinglish. One favor, learn english well and stop mistreating the old spanish language. You know: an argentinian is an italian that speaks spanish (or so they say) and thinks he is english!. Find yourself a psychoterapist of those you seem to have so much and see if you can overcome your inferiority complex and stop hating so much.

I answered in adequately insulting tone, just wanted to share the fact that there are people *actively bothered* by people not writing a book the way they like to the point of insulting a whole country. Fun!

I sit on a window table at La Farola de San Isidro and the waiter will bring me a *cortado en jarrito* with three *facturas surtidas* without being asked. Then I will read the newspaper (always back to front), pay and go back home to start working.

I would start every day that way if I could. It makes my day start great. I get to work relaxed. It puts a definite boundary between me being dad taking my son to school and me being at work, even if I am doing it at home.

Ritualization is comforting. Rituals are good for most people. On the other hand, rituals suck, are a waste of resources, and hurt you.

Sure, my coffee+newspaper is nice, but it would cost me \$15 a day, which means over \$5700 a year, which is more than half of what my son's school costs. So I only do it once a week, and the rest of the time I just buy the same damn facturas and take the coffee at home, while reading the newspaper on my netbook (BTW: there's just no way to read newspapers back-to-front on the web).

What I did was realize I had fallen into a ritual, decide if it served a useful purpose, estimate the costs, and decide against it. That means I acted rationally, and the choice I made seems correct to me. The best part of doing that is not even saving money, but knowing that I am *paying attention*.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "teaching tea protocol to kids".

I was reading yesterday a newspaper I shouldn't read[1]_ and ran into a fluff piece about [teaching tea protocol to kids](http://www.lanacion.com.ar/nota.asp?nota_id=12817).. teaching tea protocol to kids: http://www.lanacion.com.ar/nota.asp?nota_id=12817

about girls, ages 6 to 13, at a party in the Alvear hotel.

It's meaningless nonsense, but it's the kind of nonsense that can piss me off. Here are some choice quotes, translated:

"A girl asked for sugar, even though the right thing is not to sweeten the tea."

“When it’s time to put jam on your *scone* or toast, you should never cover it. ’Smear jam only on the piece you are eating, and never from the jar, always put some in the plate, then from the plate to the toast”

“... the belly of the fork should be at the bottom if you finished eating cake, or at the top if you ate a piece of meat.”

“Even if it may seem a novelty, protocol for children has 500 years of history. A precursor was the dutch humanist Erasmus of Rotterdam who in 1530 published a treaty on civility aimed to all children, specially those of the court, where he presented a common code of behaviour...”

Where can I start... how about **this is all made up nonsense?** The belly of the fork aiming down or up? Put the jam in the plate first? Bitter tea for 6 year olds? Erasmus of freaking Rotterdam in 1530?

Here’s what this is, it’s ritual. It’s *meaningless* ritual. We don’t live in the dutch court in 1530, why should we feel it’s “right” to act like they did? Why should we not act like 20th century moroccans and eat with our right hand instead?

At least moroccan food tastes good, unlike scones!

Of course I am not against things like using a napkin instead of sucking on your fingers (but hey, I am not going to call you names if you do it, and I will bloody do it if there’s no napkins), but all these random rules without any explanation are the exact kind of things kids should not be exposed to.

Yes, sometimes you have to put your feet down and say “it’s done this way and I can’t explain it to you yet”, but that’s the *exception* not the rule.

Why should you use a napkin? Because if you don’t your fingers are sticky and leave marks. Why your fork should stay on the plate after you use it? Because I don’t want to wash the tablecloth today if I can help it. Why you should put the jam from the jar into the toast? Because I don’t want leftover jam in the plate, thank you.

If you teach your kids that there are arbitrary rules without reasons, even in silly things like tea, you are forming the wrong thing in their brains, you are teaching them that authority is right, that habit is truth, that tradition is law.

And if you do it, allah forbid, then maybe they will do it too, and rituals ossify, and you get a country full of morons that have echo chambers instead of opinions.

The ritualization of everyday things is a sign of decadence in society. The more ritualistic the simple things get, the more those people are not thinking complex thought, the more they waste their mind in the trivial.

So make my day, leave the fork belly up after eating cake today. Even better: don’t look and don’t care.

Today Argentina’s largest newspaper, Clarín, printed a story about e-books titled “[The e-book grows: good news for bad writers?](http://www.clarin.com/sociedad/Crece-electronico-noticia-escritores-malos_0_296970371.html)”.. _“the e-book grows: good news for bad writers?”: http://www.clarin.com/sociedad/Crece-electronico-noticia-escritores-malos_0_296970371.html

and it’s worthy of comment.

1

system-message

WARNING/2 in everything.txt, line 26229
Duplicate explicit target name: “1”.

I have high blood pressure. This newspaper drives me mad, so it’s bad for me.

First, the obvious: yes, of course it's good news for bad writers, just like the arrival of the internal combustion engine was good news for bad riders, but it's not only good news for bad writers, it's good news for almost everyone.

Let's consider some choice quotes from this article:

In the last few months those authors previously damned to self-publishing, either by the low quality of their texts or the shortsightedness of the editors, have found [companies] willing to provide them, in electronic format, the publishing denied to them on paper.

Well, yeah, since publishing e-books is almost free, yet the prices are substantial, it will increase demand of things to publish, because the margin is bigger. You can "print" hundreds of different books for the cost of a single paper book!

So far, obvious but still more or less aligned with reality. It didn't last, though:

This progress has its victims. First the traditional editors, who see weakening their power over the selection, production and distribution of books.

Whoa, how terrible for them. I can only imagine they feel awful because their jobs are becoming more or less obsolete. Then again, so did the copyist monks jobs when editors and publishers started their gig. It lasted them 500 years, that's **a lot**.

The idea that the free access to social networks turns everyone into musicians, journalists or writers makes headway. At the same time, the role of those who, not long ago seemed called, thanks to their education to channel the development of entertainment, information or culture shrinks.

I am completely against the very *idea* of impartial journalism. There isn't *one* impartial journalist, there never has been one, and there never will be one, so why go on with the charade that yes, you are not impartial, but you pretend to be, and we pretend to believe it, and then try to correct your bias when reading... it's tiresome.

So I, a perfectly biased guy will tell you: **OF COURSE, YOU SELF-SERVING NIMROD**. Everyone *is* a writer nowadays. An average teenager writes many times more today than when I was a teenager. The main method of communication with peers has switched from oral to written! Yes, we old creeps hate *how* they write, but what the heck, that's because we *are dinosaurs*.

You are writing about the plight of buggy-whip manufacturer's middle-management instead of writing about *cars*.

Just grow a pair of nuts, say that this scares the shit out of you because you expect journalism to not be a way you can make a living in 20 years and be done with it.

In the meantime, we will all have fun having a wealth of writing (good *and* bad) without peers in the history of civilization.

This article is again, Spanish only, because it's about refuting a site that's in Spanish, and while I would translate my own words, translating the other site is more than I can stand.

Discutiendo en un lugar que no voy a mencionar para no llevarles visitantes, que no lo necesitan ni lo merecen, me pusieron este comentario:

"No sigas con ideas ateas perimidas, Dios existe, y se puede probar científicamente. lee www.torremolinosmalaga.com.ar "

¡Bueno, veamos entonces!

Es un documento increíblemente largo, pero vamos a tratar de hacerle justicia.

Empecemos con el título:

¿Por qué creo en Dios científicamente?

Esto marca la cancha: vamos a tratar esto como un argumento científico. Si el autor se sale de eso, cachetazo para que entre.

Como es **muy** largo, hoy veremos sólo el Capítulo I: Historia

A veces envidio a los agnósticos, pues pueden vivir tranquilamente diciendo “no sé”. Yo soy incapaz de hacerlo. Aun con una formación religiosa tampoco deseo aceptar algo sólo por “fe”, sin ni siquiera cuestionarlo.

Pero tampoco me gusta que me mientan alevosamente con teorías que son totalmente imposibles, y lo hagan con tal caradurez sólo apreciada a algunos políticos. Uno pretende que un científico no trate de ajustar la teoría a sus propias creencias o a situaciones políticas o económicas. Y mucho menos que toda una colección de científicos defiendan teorías sin ninguna base sólida y sin ninguna mínima prueba.

Buenísimo, a mí tampoco.

Una de ellas es el origen de la vida. Poder decidir si hay alguna teoría que tenga la posibilidad de creerse.

El argumento por incredulidad (“no puedo creer que eso sea cierto”) no es científico, es una falacia. Hay montones de cosas que cuesta creer pero son ciertas.

Es sabido que hasta el mas mínimo ser viviente tiene una serie de componentes en su composición química.

Y sí, no hay seres vivos compuestos únicamente de carbono, digamos. Obvio pero cierto.

Esto sugiere que como sea que se haya generado el primer ser viviente, esos componentes debieron estar juntos en la naturaleza para que el supercreador del “tiempo” pudiera completar su tarea. Por supuesto la posibilidad de que todos estos elementos se puedan encontrar en la naturaleza en una cercanía tal que pudiera servir al tiempo es el cero absoluto.

No es lo mismo “posibilidad” que “probabilidad”. El cero absoluto es una temperatura, no una probabilidad. En probabilidad no hay ceros absolutos y relativos, hay ceros no más. Y además, la probabilidad de casi nada es cero. Hay una probabilidad ínfima (pero no cero) de que yo entre en combustión espontánea en este mismo momento, o de que atravesara el piso sin romperlo. Por lo menos si uno acepta la mecánica cuántica, claro.

Ante tal evento un científico ruso propuso que debió haber una sopa química que tendría casualmente las propiedades correctas. No sólo la probabilidad sigue siendo cero de que haya habido alguna vez una sopa química, algo no natural en todo caso y ya situar además ciertas proporciones parece más una tomada de pelo que un postulado científico.

El mar es una sopa química (contiene sodio, potasio y montones de otros elementos disueltos en agua), no sé por qué le parece improbable y antinatural la idea de una sopa química.

Por otro lado, seguramente se está refiriendo a los experimentos en abiogénesis de Miller y otros, en los que se ve la formación espontánea de aminoácidos a partir de amoníaco hidrógeno y metano.

No inventaron las proporciones de materiales de la “sopa”, los toma de modelar la atmósfera primitiva. Es química, que el autor no pueda creer en ella no quiere decir que no exista. ¡Sin ir mas lejos, son algunos de los componentes principales de la atmósfera de Júpiter! A menos que el autor considere improbable la existencia de Júpiter, por supuesto.

Pero aún científicos de antes de la era del ADN, sabían de la extraordinaria complejidad de un ser vivo. Era impensable que aun con una probabilidad de cero, aún si se pudiera admitir la existencia de tal cúmulo de casualidades, el tiempo era incapaz de crear una estructura orgánica desde una sopa inorgánica. La posibilidad que una mera actividad en la tierra como un rayo, una erupción o un meteorito no podrían crear lo que es una compleja vida, o si se quiere con los descubrimientos actuales, crear una cadena ADN.

Nuevamente, que el autor sea incapaz de pensar algo no quiere decir que los demás estén sujetos a las mismas limitaciones (por suerte).

Ningún laboratorio con los mas modernos equipos ha podido jamás crear un ser vivo de algo inorgánico. ¿Cómo el señor tiempo podría hacerlo? Sin ninguna base científica, sin una sola prueba de que alguna parte de la teoría pueda ser verdad, estas teorías no pueden ser consideradas científicas. Competirían en un mismo terreno con las propuestas religiosas, salvos que aquellas proponen generalmente a un Dios creador, algo que no se puede desmentir científicamente y estas proponen a un “tiempo” creador que se puede científicamente demostrar como totalmente incorrecta.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “craig venter”.

Craig Venter.. craig venter: <http://www.guardian.co.uk/science/2010/may/20/craig-venter-synthetic-life-form>

tomó una bacteria y le reemplazó el genoma por uno creado artificialmente. Al reproducirse esa bacteria, las siguientes generaciones son creadas en base a nutrientes (que admitirán, no están vivos) y las instrucciones de ese genoma artificial, utilizando el cadáver reanimado de la bacteria como “fábrica”[#id3]_.

Es debatible si eso es vida artificial o no, pero se parece bastante. Y obviamente, que algo no se haya hecho no es prueba de que sea imposible. Ningún laboratorio ha creado una fusión atómica autosostenida, pero eso no quiere decir que el sol no exista, o no funcione de esa manera.

Esas son perogrulladas, pero son perogrulladas que el autor parece soslayar, obnubilado por el atrevimiento de la existencia de cosas que él no puede crear.

Por otro lado, el autor insiste en decir que puede demostrar científicamente algo pero sigue igualmente sin hacerlo. Cabe aclarar que la afirmación reiterada y vehementemente no es parte del método científico.

Pero la rareza de los científicos es inaceptable no sólo porque la probabilidad de que ciertas circunstancias puedan aparecer es cero, sino que además cualquier científico no puede ignorar que un ser vivo es además de química tiene un componente que se denomina vida, y que ningún científico ha podido jamás explicar como es, cómo se compone. Ése componente de vida que tal vez tiene peso, o energía, no ha podido nunca aislarse ni definirse. La composición química de dos semillas puede ser igual pero la que tiene vida puede germinar y la otra no, No importa lo que se haga. Un cuerpo humano puede tener un paro cardíaco, o una muerte cerebral, pero la vida del cuerpo desaparece en otro momento. Es conocida la medición de ciertos científicos de una pérdida súbita de peso en lo que sería la pérdida de la vida. Es algunas semillas que pueda flotar o no puede significar que ya no tenga vida, o que si es posible que germine.

El autor asigna sus propias convicciones a los científicos en base a que “no pueden ignorar” algo. Pues bien, es perfectamente posible ignorar que exista un componente llamado “vida” en un ser vivo.

Decir que uno está tratando de demostrar algo científicamente y por otro lado poner en juego conceptos esotéricos, como la existencia de una entidad inmedible llamada “vida” es deshonestidad intelectual o ignorancia.

La composición química de dos semillas, una viva y una muerta, no son iguales mas allá de características groseras como la cantidad de carbono o nitrógeno en las mismas. La química trata también sobre procesos, lo digo desde una casi absoluta ignorancia de la química [4], y supongo que la actividad química en células en las semillas vivas no es igual a la de las células de una semilla muerta.

Elaborar una teoría donde la probabilidad es cero de que hubiera ocurrido y soslayando explicar lo que no puede explicarse es definitivamente algo no científico. Darle un carácter diferente de una mera burla o tomada de pelo es ya algo impensable.

Nuevamente el autor supone cosas y le asigna valor de verdad sin dar ningún motivo. ¿Quién dice que la probabilidad de eso es cero? El autor. Entonces es cero y es una burla. La falta de rigurosidad de ese argumento es obvia.

Aún así, aun imaginando que pudiera aparecer una vida una vida incipiente, con todo lo que científicamente se entiende como una criatura con vida, es decir, un complejo mecanismo para alimentarse, reproducirse y volver a formarse. El mecanismo de formación de cualquier ser es tan largo y complejo que no puede despreciarse o tomarse a la ligera. Cuando uno estudia el ADN, no puede menos que maravillarse de la cantidad de información e instrucciones que es necesaria para que una célula fecundada, se divida en dos, en cuatro y así en millones, pero transformándose en muy distintos componentes del ser vivo. Uno ya podía intuir que un programa de computación que quisiera emular el mecanismo de formación de un ser con su carga genética debería ser muy complicado. Cualquier científico lo intuiría. El ADN nos da una medida de cuanto complicado es. Esta

vida incipiente que ni remotamente puede haber sido creada por la casualidad o el tiempo como se nos quiere burdamente hacer creer, debe además caer en un ambiente propicio para cumplir el ciclo de todo ser viviente, alimentarse y reproducirse antes de morir.

Creo ya percibir un patrón en las objeciones del autor. “Tal cosa es improbable, por lo tanto no es posible”.

Tal vez el autor no comprenda en verdad lo grande y viejo que es el universo. Cuando algo es altamente improbable, pero puede suceder en millones y millones de lugares, y se intenta una y otra vez durante millones y millones de años, deja de ser tan improbable.

Es improbable ganar la lotería. Casi siempre alguien la gana.

La excelente frase de Einstein “El hombre encuentra a Dios detrás de cada puerta que la ciencia logra abrir” resume mi pensamiento,

También dijo “El interés compuesto es la fuerza más poderosa de la galaxia”, y se aplica mucho más directamente al tema del comienzo de la vida que la otra.

es tan complejo todo el mecanismo necesario para la creación de la vida, que no hay duda que Dios lo creó. No queda ninguna otra posibilidad. La ciencia ha ido demostrando que cada teoría de Einstein.

¿Ha ido demostrando que cada teoría qué? Ahí falta algún pedazo de frase...

Algunos científicos buscan ante la imposibilidad de demostrar cómo la vida fue creada, inventan otras posibilidades a su gusto. La vida puede haber sido traído por seres extraterrestres o por algún meteorito. Total la ignorancia general puede aceptar cualquier afirmación. Ví un documental que hablaba sobre la importancia de una roca encontrada en la Antártida, tenía componentes orgánico que aseguraban que dicha roca, que casualmente provenía de Marte, podría ser testigo de que la vida en la tierra habría venido desde Marte, según el director de la NASA.

No, la roca no tenía componentes orgánicos, decían que tenía microfósiles. No es lo mismo. Y el director de la NASA no lo dijo, lo dijo David McKay en 1996 en la revista Science.

Y no es casualidad que provenga de Marte, hay un estudio acerca de las burbujas de gas atrapadas en el mismo, y como su composición química coincide con la atmósfera marciana, por ejemplo. Para más datos, se puede leer este informe en el sitio de la NASA: <http://nssdc.gsfc.nasa.gov/planetary/marslife.html>

Me pareció tan ridícula tomada de pelo que me puse a investigar. Lo primero que hice fue entrar al sitio de Internet de la NASA donde pronto ví una afirmación rotunda del organismo. “Jamás se ha descubierto ningún signo de vida fuera de la tierra”. Pero también pude por Internet que realmente el director de la NASA había dicho eso. No me costó mucho darme cuenta que por entonces se estaba gestionando la financiación del envío de la primer sonda a Marte, y que la roca en cuestión la ALH84001 que podía provenir de tiempos de la formación de Marte tenía algún residuo mineral que se tomaron para aseverar una mentira.

El meteorito ALH84001 no es un signo de vida actual, es un signo de una posibilidad de vida hace 4.8 billones de años. Que la investigación sobre ALH84001 la hizo la NASA no es discutible.

Las minúsculas estructuras encontradas en el interior del meteorito que se decía tenían un aspecto similar al de las bacterias fosilizadas que se ven en la Tierra. Hoy se acepta que estos residuos son puramente geológicos.

Pero por entonces la detección de magnetita que es común a determinadas bacterias y ciertos glóbulos de carbonato les permitía decidir que estos eran desechos típicos de alguna forma de vida. Luego han descubierto que tales formaciones son típicas de un volcán. Con el viaje de la sonda a Marte ya nadie afirma que hubo alguna vez vida en Marte.

Es un tema de discusión. Para más datos ver http://en.wikipedia.org/wiki/Mars_meteorite que tiene varias referencias a estudios recientes, por ejemplo “Thomas-Keprta, K., S. Clemett, D. McKay, E. Gibson and S. Wentworth 2009. Origin of Magnetite Nanocrystals in Martian Meteorite ALH84001 journal Geochimica et Cosmochimica Acta: 73. 6631-6677.” que dice que la magnetita es compatible con actividad biológica.

La mentira puede ser justificada como un error científico, pero la realidad es que esperar que en Marte, mucho menos propicio para tener vida, se hubiera generado la vida miles de millones de años antes que en la Tierra era totalmente improbable,

No, no miles de millones de años antes. El meteorito tiene 4800 millones de años. Los microfósiles más antiguos terrestres conocidos tienen más de 3500 millones de años.

y de que esa minúscula vida en Marte hubiera dejado desechos que por casualidad algún meteorito a su vez chocando con Marte hubiera liberado una roca al espacio y que luego de millones de años esta justo cayera en la tierra en la antártida para develarnos la vida en Marte es demasiado poco probable. De probabilidad cero absoluto.

Repetir una y otra vez que se desconoce la ciencia de la estadística no ayuda a la causa del autor.

Es mucho más creíble que ésta fue una teoría forzada para inspirar el financiamiento del proyecto de envío de la sonda a Marte. No tengo dudas que los científicos que suscribieron la teoría no creían en ella sino que les era conveniente. No es la mayoría de los científicos los que hacen este tipo de cosas, pero sin duda son los más publicitados. Los que adquieren renombre en base a teorías falsas o que no cerraban ni en el comienzo de su esbozo. Por caso está todo el espectro de Darwin y sus sucesores que defendieron lo indefendible.

¡Como no puedo creer en sus teorías, debe ser que son fraudes que intentan obtener fama y fortuna! Este haragán ataque *ad hominem* no merece mayor respuesta, ya que no explica nada, no refuta nada, no propone nada, y no sirve de nada.

Entonces, resumiendo, el autor no ha hecho ninguna afirmación refutable, no hace ninguna predicción comprobable, y en general no ha hecho más que parlotear insistentemente en que algunas cosas son improbables y por lo tanto él se niega a creer en ellas.

De qué forma esta tontera pueda llevar a creer en dios, mucho menos a creer en él de forma científica, todavía no tengo la menor idea. Veremos que sucede en capítulos

posteriores.

Once more for any english-reading audience, this is only in spanish because I don't want to translate the nonsense I'm replying to.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "parte 1".
```

Bienvenidos a la segunda parte de "Discutiendo con cualquiera". Para información de qué es esto y qué es esta cosa que estoy comentando, tal vez le convenga leer la [parte 1](http://lateral.netmanagers.com.ar/weblog/posts/BB898.html). parte 1: <http://lateral.netmanagers.com.ar/weblog/posts/BB898.html>

Hoy tenemos un bocado más sustancioso, o por lo menos más grande.

» Capítulo II – Darwin y la mayor falsedad jamás difundida.

Darwin tiene un encanto especial para mí, fue uno de mis primeros héroes "reales" en mi adolescencia. A los 11 años leí "El origen de las especies" y quedé fascinado. Lo que no pensé en ese momento es que lo había entendido completamente. Lamentablemente el autor de este panfleto no ha tenido la misma suerte, al leerlo no sólo creyó haberlo entendido, sino que decidió que casi todo estaba equivocado.

Vamos a revisar lo que dijo, pero recuerden que yo tengo una ventaja injusta: sé muy poco de biología, pero sé que sé muy poco.

Cuando era niño me enseñaron las bases de distintas teorías como la teoría de la relatividad, la teoría de la formación del universo y la teoría de la evolución de las especies. Por supuesto uno tiende a aceptar tales teorías como ciertas, como absolutamente ciertas, como si fueran leyes, sin darle mucha importancia a la diferencia entre leyes y teorías.

Excelente idea.

Sin embargo en la teoría de la evolución de las especies había un montón de detalles que me molestaban de sobremanera. Mucho me parecía más una mala fábula, una torpe novela, que algo con rigor científico. Frases en donde todo se le atribuye casi con un poder más que mágico, al tiempo, como una suerte de comodín que sirve para explicar todo lo inexplicable.

17

```
system-message
WARNING/2 in everything.txt, line 26385
Duplicate explicit target name: "id3".
```

Sí, es una descripción intencionalmente provocativa.

18

```
system-message
WARNING/2 in everything.txt, line 26387
Duplicate explicit target name: "id4".
```

Tengo la ventaja de **saber** que no sé química, a diferencia del autor.

Ya en el capítulo I el autor muestra una desconfianza por el tiempo, lo antropomorfiza y dice que se le asignan poderes. No, el tiempo no tiene poderes, el tiempo es el tiempo, nada más.

Más adelante, cuando veía documentales en donde todo se explicaba con frases como “Los antepasados de los pájaros modernos eran probablemente pequeños dinosaurios emplumados que vivían en los árboles y que desarrollaron alas para planear entre las copas”, siempre me preguntaba cómo había sucedido esto. ¿Las alas se habían desarrollado de a poco? ¿Porque tenían la necesidad se les desarrolló las alas? ¿Cómo se desarrolló algo de a poco?

El autor comienza con el pie izquierdo, la evolución no tiene un porqué. No es un proceso intencional, con una causa detrás y un objetivo delante. La evolución es ciega, avanza a tropezones y falla la mayor parte de las veces. La idea de que hay un objetivo y una mano que guía es más bien una idea religiosa, no de la teoría evolutiva.

Atento a los postulados de Darwin, las formas de vida no son estáticas sino que evolucionan; las especies cambian continuamente, unas se originan y otros se extinguen, el proceso de la evolución es gradual, lento y continuo, sin saltos discontinuos o cambios súbitos.

Más o menos. Sí, Darwin era gradualista. Hoy en día no creo que esa sea la teoría más favorecida por la evidencia o la más popular entre los biólogos.

En aquél tiempo, y aún hoy estoy de acuerdo en los postulados de la selección natural, donde la variabilidad y la supervivencia en la lucha por la vida van favoreciendo el mejoramiento de las especies.

No, esa es una trampa en la que caen los que no entienden a Darwin. No existe el “mejoramiento” existe la adaptación. Podríamos decir que algo “mejora” si se adapta mejor a su ambiente, pero el ambiente mismo es dinámico, con lo que esa mejora puede ser un empeoramiento luego. La teoría de la evolución no habla de una escalera hacia “lo mejor”, es mucho más caótico.

Con simples postulados matemáticos se llega a la misma conclusión. Pero este mismo hecho me hacía imaginar un animal evolucionando con un pequeño ala todavía no desarrollada, así que como ala no le servía. Por una cuestión de inutilidad el pequeño embrión de ala debía haberse atrofiado y mas bien tendido a desaparecer.

Las cosas inútiles no tienden a desaparecer. Las cosas **que disminuyen la adaptación al ambiente** tienden a desaparecer, y a veces ni siquiera eso, porque también está la selección sexual.

Por ejemplo, ¿la cola del pavo real, es útil? Sólo para conseguir pavas. Es molesta, lo hace presa más fácil de los depredadores, etc. ¿Pero de que sirve ser un pavo ágil de cola chiquita si ninguna pava te da su teléfono? De nada, esa mutación desaparece en poco tiempo.

Es un ejemplo sencillo de como algo inútil y hasta desventajoso en muchos aspectos puede no solo propagarse y extenderse en una población, sino aumentar hasta proporciones estrambóticas.

En ése entonces, con lo que entendía del proceso, y aún aceptándolo como verdad, los mismos postulados parecían ilógicos y golpearse entre ellos. Toda el postulado de que los organismos parecidos se hallan emparentados y descienden de un antepasado común, se llevaba muy mal con las clases de genética donde era muy claro que no debía cruzarse hermanos o parientes muy cercanos porque el resultado es la degradación de la especie.

El autor parece pensar que el “parentesco” entre el burro y el caballo (dos especies tan cercanas que son aún capaces de cruzarse) es del mismo orden del parentesco entre él y su hermana. La tontería de ese paralelo es casi difícil de explicar.

Si todos los organismos vivientes pueden remontarse a un origen único de la vida, si todos los hombres descendemos de un teórico adán ¿cómo es que fué sucediendo la descendencia de ése adán?

¿Me repite la pregunta? ¿O necesita la explicación de los pájaros, las abejas y las amebas?

Otro problema por entonces que tenía era que la evolución de las especies se debía dar en períodos largos de tiempos en diferencia de cientos de miles de años.

Los eventos de especiación no tardan tanto. Por ejemplo, una de las especies de pinzones que Darwin vió en las Galápagos está a punto de dividirse en dos especies.

Así con una idea ecologista reducida de sólo dos componentes, conejos y lobos, donde los segundos aparecían como una evolución natural para alimentarse de los primeros, me parecía que con una cantidad enorme de años de diferencia entre la aparición del conejo y el lobo, debía haber habido una saturación tal de conejos, que ciertas capas de la tierra debían estar llenas de estos esqueletos, lo que nadie decía haber encontrado.

Si el autor plantea un modelo simplista al extremo, y ese modelo no refleja la realidad, el problema es el modelo, no la realidad. Sospecho que nunca hubo un mundo poblado de conejos exclusivamente, por lo cual la falta de un registro fósil de ese período conejásico no es sorprendente.

Es decir ecológicamente no me cerraba el postulado de cientos de miles de años de evolución. Me parecía insostenible, pero por supuesto poco lo cuestionaba.

Si no cuestiona lo que le parece insostenible, mal por Ud. señor autor. Claro, estamos nuevamente ante su viejo favorito “no lo puedo creer, debe ser falso!”.

Luego empecé a ver que los postulados de las leyes de Mendel sobre genética, sobre los genes recesivos y dominantes, a obedecer más a un principio de constancia y regularidad que a un principio de cambio, por otra parte De Vries daba testimonio de la aparición súbita de variantes en el proceso de reproducción de ciertos tipo de plantas, cuestionaba tanto el gradualismo como el mecanismo de la selección natural.

La patética gramática de este párrafo me impide entender qué corno está diciendo.

Tardé en darme cuenta que ya había nuevas teorías como la teoría sintética de la evolución o neodarwinismo, que rechaza la herencia de los caracteres adquiridos, pero ratificabas los gradualismo en la evolución y proponían como mecanismo de la selección natural a la producción de mutaciones cromosómicas o variabilidad genética y una selección de los portadores de dotación genética más favorable para hacer frente a las presiones ecológicas; los que matemáticamente tienen una probabilidad de supervivencia y de procreación más alta que el resto de la población.

No, no y no. La variabilidad genética no es el mecanismo de la selección natural. Es la fuente de aquellos cambios sobre los que la selección natural (y la selección sexual, y otras cosas) actúa. Son dos cosas distintas.

Pero éstas teorías me hicieron plantearme cual de las dos sería válida, y si las dos no serían erróneas.

Es muy probable que ninguna de las teorías actuales sea perfectamente correcta. La ciencia avanza por iteraciones. Se tiene una teoría, luego se reemplaza con otra mejor.

También me planteé ¿por qué me habían enseñado una teoría como la de Darwin si ya había otras que trataban de explicar sus fallas?. ¿Cómo es que las fallas nunca están documentadas? ¿Cuál es el sentido de enseñar algo erróneo?

¿Por qué enseñan física newtoniana si se sabe que está mal? Porque es más fácil y a los fines prácticos funciona bien. Se enseña la teoría de la evolución porque es una explicación posible del desarrollo de las especies, y hoy en día se cree que es la más adecuada. Si el día de mañana hubiera una mejor, veremos.

La idea de “Ah, Darwin no estaba 100% en lo correcto! Hay que tirar todo al diablo!” es ridícula.

Si hay evidencia física y de química que contradice la teoría ¿por qué ocultarla?

¿Quién la oculta? ¿Adónde? Por ejemplo, Stephen Jay Gould ha publicado docenas de libros (¡que se han vendido muy bien!) explicando sus diferencias con Darwin. Le recomiendo “El pulgar del panda” que está muy bueno.

Además por ejemplo, una dificultad que es conocida como el «dilema de Haldane argumenta que no se explica la permanencia de una especie cuando parte de sus individuos han evolucionado hacia formas más aptas constituyendo otra.

Ni siquiera Haldane mismo creía que eso fuera correcto. Para algunos es difícil aceptar sus errores. Por suerte no era así para Haldane que en subsiguientes ediciones corrigió un error matemático que provocaba ese efecto.

Más información incluyendo las ecuaciones en http://en.wikipedia.org/wiki/Haldane's_dilemma

Al fin me he dedicado a buscar información de evidencia que ratifique mi impresión de estar ante una gran e inexplicable engaño.

¡Excelente!

Curiosamente, y quizás por ello de más valor, la ratificación de todas mis dudas la obtuve de la lectura de afirmaciones del paleontólogo neodarwinista Gould, quien acepta que a partir del voluminoso registro fósil actual, no hay justificación científica de hablar de proceso evolutivo gradual: la evolución ha avanzado mediante cambios súbitos, a saltos. Por decirlo así, se advierte que de repente, aparecen simultáneamente un sinnúmero de especies.

Digamos que lo dejamos pasar porque en algunos casos es cierto (la famosa explosión cámbrica, por ejemplo).

Este aporte científico y indubitable de la paleontología a eliminar el gradualismo, y en definitiva eliminar al “tiempo” como hacedor mágico me hizo preguntarme cómo yo un simple estudiante podía haber imaginado que la creación gradual de las especies no podía sustentarse y miles de científicos o al menos quienes dicen serlo la hayan aprobado sin la más mínima duda.

¿Eh? ¿Qué dice este tipo? Perdón, me salí de personaje un momento. Gould no dice nada por el estilo. Gould era un *gran* admirador de Darwin, y propuso su teoría que difiere en detalles, pero los conceptos de especies, mutaciones, evolución, selección, etc. son los mismos.

Lo que dice Gould (según recuerdo) es que los cambios no son graduales y uniformes, sino que hay períodos de estabilidad seguidos de un cambio más repentino.

Cómo eso “elimina el tiempo como hacedor mágico” no tengo idea. Bah, no tengo idea de qué demonios quiere decir que el tiempo es un hacedor mágico tampoco.

Pareciera que estoy escuchando a políticos y no a científicos, deformando la verdad o ajustando los hechos a sus conveniencias. Desde allí me he convertido en un escéptico de toda información evolucionista, y cada vez que me lo he propuesto he encontrado otra explicación no evolucionista a los mismos hechos presentados.

Entonces cuando un evolucionista corrige detalles en el trabajo de otro evolucionista eso es motivo para decir que no hay que creerle a ningún evolucionista. Eso es, hablando mal y pronto, una idiotez.

Darwin no era perfecto. Su trabajo no era perfecto, y él lo sabía. Todos los científicos saben que su trabajo no lo es. Cuentan con que después vendrán otros y corregirán lo que dejan, de ahí la frase de “Si veo lejos es porque estoy parado sobre los hombros de gigantes”.

Es completamente anticientífico (y un papelón para alguien que afirma estar tratando de razonar de forma científica), lo que el autor dice. ¿Cuáles son las alternativas que tenía Gould al ver una imperfección en el trabajo de Darwin? Callarse y hacerse el tonto. O corregirlo y que tengamos una explicación mejor.

La primera opción es la típica del pensamiento religioso: ¿Cómo que el sol no parece dar vueltas alrededor de la tierra? ¡Callesé don Copérnico, que sabe ud de estas cosas!

Lo *mejor* del pensamiento científico es aceptar que uno se puede equivocar y que es más importante corregir los errores que mantener la autoridad.

Con fé católica fui instruido a aceptar un conjunto de dogmas religiosos, lo que no me ha impedido también llegar a plantearme la existencia de Dios. Al contrario de aceptar simplemente la religión como un acto de fé, he tratado de demostrar científicamente la existencia de Dios.

Y si este documento es la explicación, has fallado completamente hasta ahora, muchacho.

Realmente admiro y me sorprenden los agnósticos, o quienes pueden conformarse con un “no sé”. Yo soy incapaz de quedarme en ése pensamiento y deseo saber. Por supuesto no dudo que como humanos estamos limitados a conocimientos terrenales, y a una imposibilidad de saber sobre el más allá, pero ello no implica que no podamos intuirlo y finalmente por el razonamiento llegar a probar que Dios existe,

Hay una ley básica de la lógica y la producción de chacinados: metés mierda, sale mierda. Si arrancás con “intuir el más allá”, todo el razonamiento que le pongas arriba es moler mierda y meterla en una tripa: sale una salchicha de mierda.

Muchos filósofos se han planteado lo mismo. Uno de los filósofos que más me gustan es San Agustín, pues al leerlo sus pensamientos son profundos, y con gran sorpresa encontré que definió el mismo método que al final yo he usado: El dijo: 1) en un principio, la razón ayuda al hombre a alcanzar la fe; 2) posteriormente, la fe orientará e iluminará a la razón; 3) la razón, a su vez, contribuirá ulteriormente al esclarecimiento de los contenidos de la fe.

Vaguedades típicas de monje de la edad media. Gente con mucho tiempo libre, evidentemente.

Una de las afirmaciones interesantes que hizo hace 15 siglos es que el tiempo es creación de Dios, antes de crear el cielo y la tierra no había tiempo. Algo que hoy aceptamos como verdad, que el tiempo no existe antes del Big Bang.

Como explicar que un reloj roto tiene razón dos veces al día... vamos a tratar:

La teoría del big bang no dice nada de dios, con lo cual mal puede decir que dios creó el tiempo, el cielo, la tierra o el dulce de leche. Lo que dice la teoría de la relatividad (no la del big bang) es que careciendo de un marco de referencia, medir el tiempo no tiene sentido. De ahí que si no existía nada, mal puede haber habido tiempo.

Sí, si uno admite expresar esas cosas en términos religiosos, entonces lo que dijo Agustín puede con un poco de buena voluntad tomarse como correcto.

Claro, también lo dijeron los indios Lakota (http://en.wikipedia.org/wiki/Creation_myth):

Los Lakota cuentan en su versión de la creación del mundo que los dioses vivían en un submundo y la humanidad los servía. La salida del submundo fue iniciada por Iktomi, el tramposo, que conspiró para separar al dios sol Wi y su esposa Hanwi la luna. Su separación marca la creación del tiempo.

Así que todo bien con Agustín pero no es exactamente original lo suyo.

Hoy, cualquier persona normal puede entender que los postulados de Darwin son definitivamente inaceptables desde el punto de vista científico pero el porqué tal errónea teoría fue difundida a tal punto de ser enseñada en las escuelas es tratado en el siguiente capítulo. Dejemos para este capítulo la incoherencia de la teoría.

Por favor muéstrame la incoherencia.

La primera falla de la teoría Darwinista era la afirmación que una especie habría gradualmente formado otra.

Sí, tal vez fuera así cuando Darwin publicó sus libros. O sea, si este debate fuera en 1865 o algo así, tendría sentido lo que dice. Pero no es así.

La paleontología no ha descubierto ni un mísero hueso que puedan atribuirse algún cambio gradual de alguna especie en otra. No sólo el eslabón perdido, sino ningún eslabón.

¿Ah, si? Ni siquiera me voy a molestar con huesos. Mirá *dientes*:

Pero si ésta es una verdad que sirve para refutar la teoría, también existía como verdad que en miles de años documentados de la humanidad jamás se ha visto ninguna evolución de una especie en otra.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "aquí hay una lista".

Se han observado unos cuantos eventos de especiación, y hasta se ha producido intencionalmente en condiciones de laboratorio. [Aquí hay una lista..](#) aquí hay una lista: <http://www.talkorigins.org/faqs/faq-speciation.html>

Los postulados de genética invalidan que cualquier mutación provechosa pudiera sobrevivir y evolucionar.

¿Eh? ¿Que postulados? ¿De qué estás hablando Willis? ¿Alguna cita? ¿Un nombre, una referencia? ¿Una explicación aunque mas no sea?

Todos los casos de mutaciones o cruza siempre han generado en mecanismos que no pueden evolucionar desde allí. Con lo que hoy sabemos del ADN, es imposible pensar en que una mutación azarosa pudiera alguna vez generar un código nuevo que no esté dañado.

Ah, pero resulta que no sólo sucede sino que ha sido observado en laboratorio. Autor, le presento a *Escherichia Coli*. Resulta que si uno toma bacterias y las examina durante 44000 generaciones, pasan cosas interesantes. Por ejemplo que desarrollan una mutación que les permite digerir citrato.

Es muy interesante y demuestra que el estimado autor no tiene la más pálida idea de qué habla: http://scienceblogs.com/loom/2008/06/02/a_new_step_in_evolution.php

Ninguna especie podría sobrevivir si su código fuera continuamente alterado. En realidad todo funciona exactamente al contrario de las teorías de Darwin, el código o programa genético, es auto-corrector, es parafaseando algunas técnicas de computación y comunicación, el código logra ir anulando toda variación y tiende a restaurar el código original anulando las mutaciones.

No. Hay prueba experimental (ver arriba) de que no es así.

Si un hombre o animal desarrollara un ala incipiente, luego de varias generaciones por atrofia, no uso pero por volver a su código original, el ala desaparecería, por más que los humanos necesitáramos de volar.

No, el ADN no se autocorriga de esa forma.

Que un pájaro pudiera surgir de un huevo de reptil puede parecer tan cierto como que un dinosaurio pueda ser creado de un huevo de rana. Simplemente una ficción sin basamento científico aun cuando se pretendan disfrazarla. Jurassic Park intenta entretener pero de ninguna manera hacer dogma científico. Lo irrespetuoso es que los científicos pretendan engañarnos con teorías que no tienen ni una mínima base científica.

No, lo irrespetuoso es creer que se lo sabe todo y que la gente que se ha pasado una vida estudiando sabe menos que uno. El orgullo malsano del autor es evidente.

El mismo Darwin condicionaba su teoría a la aparición de los eslabones perdidos que nunca aparecieron, por mas que se lo han propuesto millares de científicos.

¿Adónde dijo eso Darwin? En serio, curiosidad mía nomás.

En realidad nada de la teoría ha podido probarse. Desde el punto científico es tan absolutamente improbable que así ocurriera como cualquier fábula de ogros.

Ah, ya estaba extrañando la prueba por aseveración estentórea. No, estimado autor, las cosas no son de una manera porque usted lo diga.

Una de las bases de la teoría es que cuando uno se remonta hacia los estratos fósiles más antiguos donde se observa cada vez formas de vida biológicamente menos complejas. Pero la realidad es que nunca se ha observado ningún eslabón perdido entre ninguna especie.

Y dale con el eslabón perdido. Lo desafío a que lea “El origen de las especies” y encuentre las palabras “eslabón perdido”. Acá lo tiene: <http://www.literature.org/authors/darwin-charles/the-origin-of-species> busque “missing link” a ver si lo encuentra.

Esto tira por completo la teoría de que la evolución fue gradual y como consecuencia de una selección de las especies.

Que la evolución fue tan gradual como decía Darwin ya se han encargado (entre otros) Gould, y con muchos mejores argumentos que los suyos. Si quiere discutir esto, necesita una máquina del tiempo, no una página web.

Es más, los descubrimientos paleontológicos han venido a afirmar que la aparición de las especies en cada época fue un hecho simultáneo, abrupto, prácticamente al mismo tiempo, y luego hay que esperar hasta cien millones de años para que vuelva a haber nuevas apariciones de animales o plantas.

No, “los descubrimientos paleontológicos” no dicen eso. Lo que dice Gould es que **cada especie** pasa largos períodos de estabilidad antes de dividirse en dos. No que **todas las especies** se paran congeladas por cien millones de años. Eso es invento suyo.

¿No le ha molestado siempre que la afirmación de que tal animal es de la familia de ...? Nunca un animal es un descendiente directo. Basta leer a Sermón, especialista en genética de los microorganismos, quien asegura que siempre que se ha estudiado cualquier tipo de organismo y se acompaña su historia paleontológica, siempre se encuentra una repentina interrupción exactamente en el punto en que según la hipótesis del evolucionismo debería encontrarse la conexión genética con la cepa reproductora más primitiva. Documenta que todos los cambios del periodo precámbrico al cámbrico de microorganismos, algas, bacterias se encuentran en el cámbrico con una infinidad de estructuras más complejas como erizos, crustáceos, medusas, etc, sin una mínima prueba de infinidad de organismos que deberían ser transición de esa evolución.

Claro que no hay una infinidad. Porque para que hubiera una infinidad deberían ser un número infinito de animales. Por otro lado admito nunca haber oído de ese tal microbiólogo Sermón. ¿Alguna referencia?

Otro problema de la evolución es también la inteligencia del ser humano. Aunque hay algunas teorías de porqué el ser humano es inteligente, poco sabe la ciencia de cómo y porque el ser humano es inteligente, cualidad que no tiene ningún animal. No es el tamaño pues los elefantes y las ballenas tienen cerebros más grandes que los humanos, ni ciertos tipos de células cerebrales que los seres humanos compartimos con algunos animales. Al no saber de que viene la inteligencia es poco probable decir con algún rigor científico, que estaban dadas las condiciones para que un ancestro se volviera inteligente, y cómo este atributo se pudo transmitir en la nueva especie, sin que ningún otro animal lo lograra.

Supongamos por un momento que el estado actual de las teorías de selección natural y compañía no pudiesen explicar la inteligencia del hombre. ¿Y? Que no lo expliquen ahora no quiere decir que no lo hagan luego con más trabajo.

O tal vez haya que corregir las teorías primero.

Lo que es una zonzera es decir "como no explica esto, entonces no es cierto" porque es un sinsentido lógico. Si explicara la inteligencia, podría decir que no explica el color de ojos de las ballenas, y si explica eso, que no explica el grosor del pelo del caribú canadiense.

La lista de cosas sin explicar no tiene ninguna influencia en la validez de una teoría. Lo que sí la tiene es mostrar que la teoría predice cosas que no se producen.

Por ejemplo, la teoría evolutiva predice que si uno cría E. Coli en ambientes menos ricos en glucosa, van a tender a metabolizarla de manera distinta. Se hizo el experimento, dio bien, es una confirmación.

Si en cambio no se hubiera producido ningún cambio, eso sería un indicio de que la teoría está mal, de que hay algún problema. La falta de explicación de un fenómeno puntual no significa nada.

No ver esta obviedad es lo que me hace pensar que el autor cuando habla de hacer las cosas "científicamente" en realidad cree que eso consiste en usar palabras largas y dar nombres de científicos muertos.

En definitiva no existe una sola prueba de la teoría de la evolución. Ni una. Al contrario, el ADN, la genética, la excavaciones vienen una y otra vez probando su total falsedad. Pero además es falso también para el sentido

común. Si la evolución es un hecho, una especie como los insectos, cuyo cantidad y ciclo reproductivo es totalmente superior al resto de los animales en términos de que debería tener mas posibilidades de observarse claramente una evolución no se dá. Hoy deberíamos tener insectos con tal evolución

Y de hecho los hay. En la lista antes citada de eventos observados de especiación hay varios de insectos. La mosca de la fruta es el animal experimental clásico de la genética.

Nuevamente el autor habla sin saber de qué habla. Trata de probar las cosas en base a su propia ignorancia. “Como nunca oí hablar de esto, debe ser que no existe”. Hace 50 años tal vez eso funcionara, pero hoy en día cualquiera puede averiguar estas cosas en minutos.

La misma fuente de ámbar de la República Dominicana que inspira la película Jurassic Park, nos dá la certeza de que millones de años atrás los mosquitos y otros insectos o animales eran exactamente iguales que los actuales, sin ninguna variación.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “uno de estos”.

¿Ah sí? Menos mal que nunca me crucé con uno de estos.. uno de estos: <http://en.wikipedia.org/wiki/Meganeura>
! Los insectos no me molestan pero una libélula de 75cm de envergadura es mucho para mí.

Los evolucionistas siempre intentarán ver como no científico a todo lo que hoy se llama teoría de la creación inteligente como si la teoría de la evolución tuviera algo de ciencia.

Bastante más ciencia que eso de la “creación inteligente” desde ya.

De ser cierto la teoría de la evolución tendríamos hoy millones de ejemplos, y sólo se arguyen algún ejemplo de ballenas u otro que suena más a intento desesperado de probar lo que ya es improbable que pueda probarse. Para cada “descubrimiento” evolucionista que finalmente no se puede probar hay miles de testimonios de descubrimientos de especies que permanecieron inalterables fiel a su código genético.

Próxima vez que vaya al campo me llevo una raqueta de tenis por las meganeuras.

La única diferencia entre la teoría de un evolucionista y uno que afirme que la creación debió ser inteligente es que el evolucionista usa el tiempo como su Dios. No es mi pretensión decidir cual teoría puede ser más cierta, pero si estoy convencido que la de la evolución es una gran falsedad científica.

Es completamente ilógico decir “No pretendo decir si A o B es más cierta, pero A es falsa”. Es más, es bastante tonto.

Desde el punto de vista matemático me es imposible creer que las cosas puedan producirse por azar. La naturaleza no me va a fabricar un automóvil por error o el paso del tiempo. Mucho menos secuencias muy complicadas que tienen un esquema lógico intrincado. Ningún programa de computadora se produce por un corto circuito.

Bueno, produjo un reactor nuclear, eso no cuenta? http://en.wikipedia.org/wiki/Natural_nuclear_fission_reactor

Las leyes de la ecología tampoco permiten una creación gradual. Hoy desde el punto de vista matemático, es necesario para mi criterio que Dios haya creado especies por etapas en forma más o menos simultánea.

Van dos veces que dice “desde el punto de vista matemático”. La primera la dejé pasar, la segunda ya no. ¿Cuál es el punto de vista matemático? ¿Adónde están sus ecuaciones, sus modelos? ¿De qué matemática estamos hablando?

Incluso desde el punto de vista matemático, puedo aceptar más fácilmente que creó tres o más razas que la teoría que concuerdan tanto religiosos y evolucionistas que desean que haya habido un único adán.

Los “evolucionistas” no creen eso. Es una simplificación.

Con todos los problemas genéticos que ello induciría. No es creíble para mí que un solo hombre haya sido creado por Dios, quien no tenía ninguna restricción para crear un número mayor. Matemáticamente me parece que debe haberse creado varios miembros de una especie en forma simultánea para garantizar genéticamente y realmente la probabilidad de su continuación en un medio hostil como es la misma naturaleza con enfermedades, venenos, depredadores y otros problemas que deben superar las especies.

Y dale con “matemáticamente”.

Aunque esto pueda contradecir libros considerados por la religión como dogma de fé, como el antiguo testamento, creer a pies juntillas lo escrito en él es tan errado como lo es para los evolucionistas creer en el testamento de Darwin y su evolución. Puedo entender que en el relato de la creación y del génesis hay un paralelo apreciable entre lo que está escrito allí y lo que la misma ciencia ha venido afirmando con teorías como el big bang. La Biblia afirma que las mujeres de los descendientes de Adán fueran convenientemente encontradas, pero de ninguna manera me parece lógico hacer un dogma al respecto. Alcanza conocer la intención, del escrito.

La ciencia no cree a pies juntillas en nada. La ciencia prueba, y si no funciona lo tira. Y si funciona lo sigue probando. Toda aceptación de una teoría es provisoria y con la expectativa de que se demuestre lo contrario.

El descartar teorías no es malo, es bueno, es la señal del progreso científico. Cuando Einstein mejora la física de Newton no es porque Newton era un gil!

Por otra parte tampoco acepto los ateos que desean usar las mismas bases de la religión para definir la no existencia de Dios. Frases como “si Dios existe ¿Por qué hacer cosas imperfectas?” “¿Por qué tantas pruebas para crear el ser humano?” “¿Si Dios creó al mundo quien creó a Dios?” Mi respuesta en todos los casos es que la segunda parte de la oración no contradice la primera.

Y sí, no la contradice porque no es la idea. Ahí ya estamos entrando en un área demasiado básica, comprensión de la palabra escrita, pero voy a tratar de explicar.

Tomemos el ejemplo de “¿Si Dios creó al mundo quien creó a Dios?”. Es la primera parte de un argumento por reductio ad absurdum. Desarrollado sería:

A: Dios existe B: Dios creó todo C: El creador precede a lo creado D: Todo lo existente fue creado E: Por A y C, Dios fue creado F: Por E y B, Dios precede a Dios

Como F es contradice C, alguna de las premisas A, B, C o D es falsa. O dios no existe, o no lo creó todo, o es posible crear algo antes de que el creador exista, o existen cosas que no se crearon.

Si es falsa A, dios no existe. Si es falsa B, dios no es el creador. Si es falsa C, no existe la causalidad y estamos en serios problemas. Si es falsa D, dios no es necesario.

Por supuesto que la versión en una frase no contiene una contradicción explícita, **hay que pensar 30 segundos.**

Aun la teoría del big bang que tiene sus problemas enormes del no funcionamiento de ninguna ley física aplicable a los primeros instantes, también arranca con algo hecho, una masa, una energía enorme o una suerte de 4ta dimensión, pero como sabemos en física ningún efecto puede ser creado sin causa, no hay forma de demostrar ni de pensar que un ser que no necesita ser creado no haya dado el puntapié inicial de la creación. Se puede inventar como ateo un movimiento continuo, decir por ejemplo que el pasado es fruto del futuro, de manera de crear la ilusión de un universo que se contrae y vuelve a renacer en el pasado en algún agujero de tiempo sin necesidad de tener un momento inicial, aún así el contenido necesita por ley física haber sido creado. Y ésa es la definición de Dios.

Ver acotación anterior. De todas formas, nuevamente el autor comete un fallo obvio: que una explicación no sea adecuada no es motivo para elegir otra sin ningún fundamento.

Decir “El big bang no sirve, entonces dios existe” es casi tan lógico como “Perdió Excursionistas, dios no existe”.

O una roca se creó a si misma o Dios existe.

O ninguna de las dos cosas.

Por otra parte es una falacia tratar de decidir que por ser Dios lo tiene que crear algo según mi imagen de la perfección. Los Australianos tienen un refrán que aluden a que en la creación Dios todo lo que le salía mal lo ubicaba en Oceanía. Ello tiene más posibilidades de ser cierto que pretender que Dios quisiera hacer las cosas perfectas o de primera intención como a nosotros se nos podría ocurrir que debe ser un Dios. Lo que para un humano algo puede ser aburrido, entretenido o divertido, o lo que es perfecto o imperfecto es sólo una visión humana limitadísima. Pretender entender las razones y la voluntad de Dios y decidir lo que para Dios pueda ser perfecto o entretenido es por lo menos una irreverencia y un pensamiento demasiado limitado por nuestras propias limitaciones. No nos hemos preguntado para que sirven los insectos y después descubrimos que muchos o quizás todos cumplen su función. La primera impresión es que el mundo sería perfecto sin insectos, y no es así.

Aquí ya entramos en otro terreno que no me interesa en demasía. Dado que no se han dado razones para justificar la creencia en la existencia de un dios, mucho menos es válido ponerse a pensar para qué creó al bicho bolita.

Sería lo mismo que afirmar que una hormiga en América puede llegar a conocer los cálculos matemáticos que usó Eiffel en su torre, o que un

gato francés pueda conocer los problemas de las apolos en su decensos en la luna. La visión del ser humano, insignificante en un planeta a su vez insignificante en un universo, respecto a Dios, y a todo lo espiritual es decididamente no alcanzable

Idem anterior.

Es cierto entonces que creo en Dios porque matemáticamente no hay otra explicación.

Y dale con la matemática. El autor no ha mostrado nada que sea siquiera un pariente lejano de un vecino de alguien que oyó hablar de matemática.

Podría aceptar una explicación científica si ésta fuera probablemente correcta. Pero me niego a ser engañado por quienes tuercen la ciencia para justificar lo injustificable. Si matemáticamente llego a la conclusión de que debo creer en un Dios creador ¿Por qué pensar lo contrario?

Porque lo que viene usando no es matemática. En todo lo que leí **no hay matemáticas**. ¿Acaso el autor tiene una prueba matemática de la existencia de dios pero no la muestra? ¿O el autor dice pavadas?

Por supuesto es importante la búsqueda científica de la verdad, y que el motor de ella sea un cierto agnosticismo al tratar de averiguar cómo, cuando, porqué ocurren las cosas como ocurren. En este sentido creo que Dios hizo su creación y la observa como un programador de computadoras ve su programa o un pastor ve a sus ovejas pastar. Pero de ninguna manera puedo avalar matemáticamente las teorías del destino que algunos sustentan.

“Las teorías del destino” son patrimonio de las religiones, no de la ciencia. La ciencia carece del concepto de destino para empezar.

Decidido a creer en Dios científicamente, luego creo en Jesucristo pues mi educación católica me ayuda y por el extraordinario conjunto de enseñanzas que encierra el nuevo testamento. Todo el conjunto de enseñanzas que la palabra de Jesús documenta tiene para mí una extraordinaria coherencia que desde un lado agnóstico diría matemáticamente probable que sea divino. No tanto así lo que denominamos el viejo testamento que tiene para mí algunos altibajos sin por ello dejar de ser importantes

Ah, también tiene priebas matemáticas de que Jesús es dios. Estoy impresionadísimo. Claro, me impresionaría más si las mostrara o aunque sea las explicara, pero el autor parece creer que la matemática consiste en usar la palabra “matemáticamente” en todas sus oraciones.

Admito que por desconocimiento no puedo matemáticamente seleccionar una religión probable de ser la cierta. De haber sido adoctrinado en la religión judía como Einstein o musulmán como Yahya (que le dedico el siguiente capítulo) probablemente tendría como dogma postulados de esas religiones que imagino tan coherentes como la católica.

Quizás la coherencia del nuevo testamento me impide tratar de conocer otras religiones en mi búsqueda de cuál es la correcta.

¿Cosa coherente el nuevo testamento, no? ¿A ver, Judas le dió un beso a Jesús, o no? ¿Qué le dice su coherente y matemático libro?

Creo sin embargo que toda la teoría de la evolución ha provisto de una herramienta para no aceptar a Dios. Es como una teoría sagrada que no debe ser ni objetada.

Ud. mismo cita a Gould, que objeta lo que escribió Darwin. Lea a Dawkins que los objeta a ambos.

Científica y matemáticamente la creación es algo mucho más complejo de lo que puede cualquiera suponer. Es impensable para mí que pueda pensarse con nuestra mentalidad terrestre de que todo efecto no tiene su causa, de que un mísero grano de arena no sea el resultado de la creación de Dios, que por definición, es el ser que no necesita ser creado y que no necesita materia prima para crear algo.

“No puedo creerlo, entonces no es cierto”, la vieja y querida demostración mediante la incredulidad.

Pero necesaria y matemáticamente Dios debe haber hecho su creación por etapas. No hay otra forma matemática ni científicamente de probar que no fue así. En esto disiento de los que ven al tiempo como un elemento creador, de los que pretenden ver en un gas de una piedra a su antecesor. O los que creen en la casualidad como madre de todo lo creado.

Si tomara un trago de ginebra cada vez que dice “matemáticamente” tal vez esto sería mas sencillo.

Los que no pueden ver que la genética decididamente contradice la teoría de la evolución, y al contrario de lo que este postula, la estructura genética del ser humano contiene ya mas de 6000 enfermedades catalogadas fueron localizadas en los genes correspondientes. Es decir no vamos por sucesivas mutaciones convirtiéndonos en hombres perfectos. Al contrario la genética en lugar de corroborar la hipótesis de la evolución de las especies la desacredita aún más confirmando que un organismo no puede dejar de ser lo que es, lo que es a ojos vista además lo que ha pasado con tantas especies de las cuales puede verse como en millones de años no han evolucionado.

Que una especie no evolucione no quiere decir que **otra** no evolucione. La evolución no es el colegio donde se espera que todos pasen de grado.

El atribuir al azar a todo el perfecto orden que existe en el universo es matemáticamente un disparate. Es tirar la ciencia por la ventana e inventar una religión atea que es absurda e indemostrable y tiene como dogmas bases no científicas como cualquier religión.

Tirar la ciencia por la ventana sería una mejora comparado con lo que hace el autor, que es agarrar una muñeca inflable, hacerle caca encima, y decir que eso es ciencia.

La ciencia en definitiva es la mejor prueba de que Dios existe. La frase de Einstein “Dios no juega a los dados” ha sido interpretada tanto por los evolucionistas como los creacionistas de muchas maneras, cada uno quiere

ver lo que no es. O desean abstraerse de todo el conjunto de pensamiento de Einstein para llevarlo como ejemplo aplicado de su propio pensamiento. No es un tema del destino ni de agnosticismo ni de religión. Simplemente Einstein había encontrado que toda teoría física terminaba en un orden perfectamente concebido. Creer que cualquier teoría física es fruto del azar es básicamente un postulado tan incongruente como el evolucionismo mismo.

Einstein estaba frustrado porque no podía juntar la relatividad con la mecánica cuántica. Esa cita surge de un contexto **mu**y específico y me limito a contestar como contestó Pauling: “Albert, dejá de decirle a Dios lo que tiene que hacer”.

There’s a problem often used to show the unintuitive nature of probability, which has become very well known.

In that problem a contestant in a gameshow has to choose between three doors (A,B,C), on one there is a car, on the other two are goats.

After the contestant chooses, the host opens another door and shows a goat.

Then, the host offers the contestant the chance to switch his closed door for the other closed door.

Should he switch?

The intuitive answer is “it doesn’t matter”, because there’s two doors and one car, so it’s a 50-50 chance.

But the real answer is that it does matter, because it’s a 33-67 chance!

While it’s simple to show this to be the case to a statistically-educated dude, it’s somewhat harder for a layman.

In fact, I think most explanations suck.

Here’s my shot at it:

If you were offered the chance to switch between your closed door and **the other two closed doors**, would you take it?

The intuitive answer to that is of course, yes, because it’s 67-33 for the car to be on the other two doors.

Now, regardless of where the car is, can the host open one of those two doors and show a goat? Of course, yes.

So, would you feel your odds went down because the host showed one of your two closed doors had a goat behind it? No, because he could always do that, and you *know* there was (at least) one goat there!

So, what difference does it make if one door is open or not?

I don’t expect this to convince anyone, really, but just in case, I have a python implementation of this problem (goatcar.py :-)) if anyone wants it, if empiricism can convince you ;-)

Working pretty well now. It’s not saved for following sessions, and the folders don’t do what I want (show an aggregate of their inner feeds) yet, but it was pretty easy.

I’m one of the speakers for the free python courses in FM La Tribu, in Buenos Aires.

Every saturday you can learn something from one of the best python programmers in a thousand miles, or from me.

I’ll be teaching about virtualenv, buildout, nose and other things on August 21, about GUI stuff on September 25 and October 2, and about PyQt in October 30.

This is all for free, and I hope lots of people show up!

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "here".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "here".
```

The full schedule is [here](http://python.org.ar/pyar/CharlasAbiertas2010).. here: <http://python.org.ar/pyar/CharlasAbiertas2010>

For a small project I am doing I wanted the capability to see web pages offline. So, I started thinking of a way to do that, and all solutions were annoying or impractical.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "cutycapt".
```

So, I googled and found [CutyCapt](http://cutycapt.sourceforge.net/).. cutycapt: <http://cutycapt.sourceforge.net/>

which uses Qt and WebKit to turn web pages into images. Good enough for me!

Since I wanted to use this from a PyQt app, it makes sense to do the same thing CutyCapt does, but as a python module/script, so here's a quick implementation that works for me, even if it lacks a bunch of CutyCapt's features.

With a little extra effort, it can even save as PDF or SVG, which would let you use it *almost* like a real web page.

You just use it like this:

```
python capty.py http://www.kde.org kde.png
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "download capty.py".
```

And here's the code [[download capty.py](http://lateral.netmanagers.com.ar/static/capty.py).. download capty.py: <http://lateral.netmanagers.com.ar/static/capty.py>]

```
\ :c:\# -*- coding: utf-8 -*-\

\ :sd:``"``This tries to do more or less the same thing as CutyCapt, but as
python module.

This is a derived work from CutyCapt: http://cutycapt.sourceforge.net/

////////////////////////////////////
//
// CutyCapt - A Qt WebKit Web Page Rendering Capture Utility
//
// Copyright (C) 2003-2010 Bjoern Hoehrmann <bjoern@hoehrmann.de>
```

```

//
// This program is free software; you can redistribute it and/or
// modify it under the terms of the GNU General Public License
// as published by the Free Software Foundation; either version 2
// of the License, or (at your option) any later version.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// $Id$
//
/////////////////////////////////////////////////////////////////

"""\

\ :kn:'import'\ \ :nn:'sys'\
\ :kn:'from'\ \ :nn:'PyQt4'\ \ :kn:'import'\ \ :n:'QtCore'\ \ :p:','\ \

\ :k:'class'\ \ :nc:'Capturer'\ \ :p:'('\ \ :nb:'object'\ \ :p:'):\
\ :sd:'""A class to capture webpages as images""\

\ :k:'def'\ \ :nf:'__init__'\ \ :p:'('\ \ :bp:'self'\ \ :p:','\ \ :n:
\ :bp:'self'\ \ :o:'.'\ \ :n:'url'\ \ :o:'='\ \ :n:'url'\
\ :bp:'self'\ \ :o:'.'\ \ :n:'filename'\ \ :o:'='\ \ :n:'filename'
\ :bp:'self'\ \ :o:'.'\ \ :n:'saw_initial_layout'\ \ :o:'='\ \ :n:'b
\ :bp:'self'\ \ :o:'.'\ \ :n:'saw_document_complete'\ \ :o:'='\ \

\ :k:'def'\ \ :nf:'loadFinishedSlot'\ \ :p:'('\ \ :bp:'self'\ \ :p:'):\
\ :bp:'self'\ \ :o:'.'\ \ :n:'saw_document_complete'\ \ :o:'='\ \
\ :k:'if'\ \ :bp:'self'\ \ :o:'.'\ \ :n:'saw_initial_layout'\ \ :
\ :bp:'self'\ \ :o:'.'\ \ :n:'doCapture'\ \ :p:'()\

\ :k:'def'\ \ :nf:'initialLayoutSlot'\ \ :p:'('\ \ :bp:'self'\ \ :p:'):\
\ :bp:'self'\ \ :o:'.'\ \ :n:'saw_initial_layout'\ \ :o:'='\ \ :n:'b
\ :k:'if'\ \ :bp:'self'\ \ :o:'.'\ \ :n:'saw_initial_layout'\ \ :
\ :bp:'self'\ \ :o:'.'\ \ :n:'doCapture'\ \ :p:'()\

\ :k:'def'\ \ :nf:'capture'\ \ :p:'('\ \ :bp:'self'\ \ :p:'):\
\ :sd:'""Captures url as an image to the file specified""\
\ :bp:'self'\ \ :o:'.'\ \ :n:'wb'\ \ :o:'='\ \ :n:'QtWebKit'\ \ :
\ :bp:'self'\ \ :o:'.'\ \ :n:'wb'\ \ :o:'.'\ \ :n:'mainFrame'\ \ :p:
\ :n:'QtCore'\ \ :o:'.'\ \ :n:'Qt'\ \ :o:'.'\ \ :n:'Horizontal'
\ :bp:'self'\ \ :o:'.'\ \ :n:'wb'\ \ :o:'.'\ \ :n:'mainFrame'\ \ :p:
\ :n:'QtCore'\ \ :o:'.'\ \ :n:'Qt'\ \ :o:'.'\ \ :n:'Vertical'\

\ :bp:'self'\ \ :o:'.'\ \ :n:'wb'\ \ :o:'.'\ \ :n:'loadFinished'\ \
\ :bp:'self'\ \ :o:'.'\ \ :n:'wb'\ \ :o:'.'\ \ :n:'mainFrame'\ \ :p:

```

```

        \ :bp:'self'\ \ :o:'. '\ \ :n:'initialLayoutSlot'\ \ :p:') '\
    \ :bp:'self'\ \ :o:'. '\ \ :n:'wb'\ \ :o:'. '\ \ :n:'mainFrame'\ \ :p:') '\
\ :k:'def'\ \ :nf:'doCapture'\ \ :p:'(\ \ :bp:'self'\ \ :p:'):\ \
\ :bp:'self'\ \ :o:'. '\ \ :n:'wb'\ \ :o:'. '\ \ :n:'setViewportSize'\ \ :p:') '\
\ :n:'img'\ \ :o:'='\ \ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QImage'\ \ :p:') '\
\ :n:'painter'\ \ :o:'='\ \ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QPainter'\ \ :p:') '\
\ :bp:'self'\ \ :o:'. '\ \ :n:'wb'\ \ :o:'. '\ \ :n:'mainFrame'\ \ :p:') '\
\ :n:'painter'\ \ :o:'. '\ \ :n:'end'\ \ :p:'() '\
\ :n:'img'\ \ :o:'. '\ \ :n:'save'\ \ :p:'(\ \ :bp:'self'\ \ :o:'. '\ \ :n:'wb'\ \ :p:') '\
\ :n:'QtCore'\ \ :o:'. '\ \ :n:'QCoreApplication'\ \ :o:'. '\ \ :n:'main'\ \ :p:') '\

\ :k:'if'\ \ :n:'__name__'\ \ :o:'=='\ \ :s:"__main__"\ \ :p:': '\
\ :sd: "" "Run a simple capture"" '\
\ :n:'app'\ \ :o:'='\ \ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QApplication'\ \ :p:') '\
\ :n:'c'\ \ :o:'='\ \ \ :n:'Capturer'\ \ :p:'(\ \ :n:'sys'\ \ :o:'. '\ \ :n:'__name__'\ \ :p:') '\
\ :n:'c'\ \ :o:'. '\ \ :n:'capture'\ \ :p:'() '\
\ :n:'app'\ \ :o:'. '\ \ :n:'exec_'\ \ :p:'() '\

```

My previous post explained how to cache whole web pages as images. Now see it in action. This is a lightweight RSS reader, optimized for comic books (but it works for any feed) and for offline use (but it works online too, of course).

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “somewhere”.

Not ready for public use yet, but if you look around you can find the code [somewhere](http://kakawana.googlecode.com).. somewhere: <http://kakawana.googlecode.com>

;-)

Once more for any english-reading audience, this is only in spanish because I don't want to translate the nonsense I'm replying to.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “¿por qué creo en dios científicamente?”.

Bienvenidos al capítulo III de mi serie “Discutiendo con cualquiera”, donde analizo el contenido de [¿Por qué creo en Dios científicamente?.. ¿por qué creo en dios científicamente?:](http://www.torremolinosmalaga.com.ar/) <http://www.torremolinosmalaga.com.ar/>

. El menú del día es...

» Capítulo III – Darwin y porqué se le dió una difusión mundial.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “capítulo ii”.

Ya en el [capítulo II.. capítulo ii: http://lateral.netmanagers.com.ar/tr/es/weblog/posts/BB899.html](http://lateral.netmanagers.com.ar/tr/es/weblog/posts/BB899.html)

el autor demuestra su total ignorancia del método científico y su absoluta incomprensión de la teoría de Darwin y sus modificaciones. Veamos ahora como sigue la cosa...

Si bien desde niño ya intuitivamente descreía la teoría de la evolución y luego fui comprobando su falsedad, siempre me admiró el porqué se difundió tanto.

Claro, si es una cosa tan obviamente falsa para el autor, su popularidad entre el resto de la gente probablemente sea... ¡una conspiración!

Siempre ha habido científicos que la aceptaban y quienes la rechazaban, pero avanzó con tal fuerza para ser una teoría que todo debía decirse en términos de evolución.

No. Hay montones de teorías no evolutivas de las cosas. Aún suponiendo que estemos hablando únicamente de biología, esto no es cierto. Pero démosle el beneficio de la duda al autor y supongamos que está aplicando hipérbole.

Cuando el ganso necesitó volar le crecieron las alas, cuando el ser humano necesitó pensar obtuvo su inteligencia.

Este error viene de arrastre. La evolución no es una fuerza dirigida a satisfacer las necesidades de nadie. Es un enfoque completamente erróneo.

Por ridícula que fuera la frase siempre tenía que aludir a la teoría de la evolución. Para los evolucionistas si un animal tiene un ojo era que lo necesitaba. Así quedaron cosas ridículas de establecer como el radar de los murciélagos. Pero todo el tiempo lo puede según ellos.

Poner palabras en boca de los demás no es buena educación, sobre todo si son inventos suyos.

Para describir a una ballena, se debe decir que tienen extremidades anteriores que han evolucionado hasta convertirse en aletas, aunque sus huesos todavía muestran reminiscencias de elementos óseos articulados en dedos y con extremidades posteriores que se han perdido por completo y sin ninguna conexión entre estas y la cola. Para el humano común bastaba decir que la ballena tiene aletas y cola y nada más.

Para el humano común esta computadora es grande y gris. Para mí es un P4 2.6Ghz con 2GB de RAM corriendo Arch Linux actualizado a ayer con driver legacy nVidia. Las descripciones de las cosas varían de acuerdo a quien las hace y para qué las hace.

Si un biólogo describiera una ballena como un bicho grande con aletas y cola es un ganso (el biólogo, no la ballena).

Siempre me maravilló esta persistencia en describir todo como si alguna vez alguien hubiera desenterrado una ballena con 4 patas y que la comparación algo más real que la imaginación totalmente fuera de la ciencia del relator.

¡Bueno, pero resulta que sí se han desenterrado ballenas con patas!
Veamos algunos ejemplos:

Más información en <http://tinyfrog.wordpress.com/2008/02/01/pz-myers-vs-geoffrey-simmons-discovery-institute>
Que interesante no? Se puede ver como los miembros posteriores se reducen comenzando con Rodhocetus, casi no existen en Dorudon y desaparecen en Basilosaurus.
Esto no fue mas que otro ejemplo de el autor creyendo que su ignorancia es prueba de algo. Él no conoce fósiles de protoballenas, entonces las protoballenas son un mito.

Esta manía de todo explicarlo como si realmente hubiera sucedido, que sólo tiene un paralelo en los políticos que pueden disfrazar la estafa más burda como una necesidad del pueblo.

Demagogia barata.

Cuando leí el excelente libro de Harun Yanya, el engaño del evolucionismo, no sólo encontré un libro que reflejaba la mayor parte de mis ideas sino que a su vez me dio una razón valedera de que la teoría de la evolución que era por lo menos muy discutible y con innumerables problemas que el mismo Darwin no podía resolver, haya evolucionado a tal punto de ser usada como verdad absoluta en gran parte de los escritos.

La teoría de la evolución tiene problemas y antes ha tenido otros. Es la naturaleza de la ciencia que todas las “verdades” son provisorias y sujetas a corrección. Obrar de otra manera o aceptar otra cosa es apartarse de la ciencia y entrar en el dogma religioso, tentación a la que el autor no se resiste en ningún momento.

Algo sin ninguna base científica, con problemas desde el origen, no debería “evolucionar” tan fácilmente en una teoría universalmente aceptada.
¿Porqué los detractores de la teoría no tuvieron difusión?

¿Qué detractor no tuvo difusión? Cuando Darwin publicó su libro, fue ridiculizado en los medios por montones de personas hoy casi olvidadas, excepto por la referencia de “este es el que se reía de Darwin”.

Y siendo una teoría tan difundida también me llamaba la atención que no se difundiera todos los arreglos a la teoría de lo que se denomina neodarwinismo que pretende mantener algunas pocas bases de los postulados y a la vez subsanar los problemas insalvables con nuevas teorías ¿porqué enseñar la teoría que hasta sus defensores consideran equivocada?

¿A qué se refiere con que no se difunden? Dawkins vendió muchísimas copias de “El relojero ciego”, Gould vendió muchísimas de sus libros de divulgación. Que el autor no haya oído acerca de los cambios posteriores a 1870 habla mal de él solamente.

Harun introduce un concepto básico. La difusión de la teoría de la evolución tiene un objetivo claro, apuntalar la filosofía “materialista” que sostiene que lo único que existe es la materia, esencia para ellos de todo lo orgánico e inorgánica negando un Dios creador.

¿A quiénes se supone que se acusa de querer hacer eso? Cuando se firmó ese acuerdo, y porqué no me pagan mi parte para ser miembro de la conspiración, es lo que yo quiero saber.

Al no tener mas que el nivel de materia, el hombre repara solamente en él, y no necesita de valores morales, y va mas allá al abolir cualquier valor básico de ética y sociedad.

Para ser este un escrito que pretende probar la existencia de dios de forma científica, el autor parece simplemente asumir que no sólo dios existe, sino que es la fuente de toda moral y ética. Estas payasadas de primaria no merecen mayor respuesta que: soy ateo y soy moral, chupáte esa mandarina.

El resultado político de la filosofía materialista es el comunismo. El cual entre sus premisas siempre fue netamente antirreligioso.

Uhhhhh los comunistas, yo sabía que los comunistas tenían algo que ver con esto.

Es un hecho que Marx escribiera en el libro para Engels “El origen de las Especies” lo siguiente “este es el libro que contiene los fundamentos de la historia natural para nuestros puntos de vista”. Tanto Marx como Engels, declaradamente ateos, tienen en su base el terminar con las creencias religiosas. El comunismo en su lucha de la guerra fría fue el mayor impulsor de las teorías de evolución dado que éstas confrontan directamente con bases de la religión

Por adonde empiezo... el bloque comunista era anti darwinista. La teoría ”oficial“ de la Unión Soviética era el Lysenkoísmo, o Lysenko-Michurinismo, que es una variante de la evolución Lamarckiana de la heredabilidad de los caracteres adquiridos. Es **muy** distinta de todas las cosas que el autor ha mencionado hasta ahora, y era ya una teoría perimida en occidente a fines del siglo XIX.

Por ejemplo, una de las teorías de Lysenko era que si uno arrancaba las hojas de las plantas de algodón produciría plantas sin hojas por herencia.

O sea, el autor no sólo no tiene idea, no tiene idea de que no tiene idea.

sugiriendo que Dios no es necesario para la creación. Esta es sin duda uno de los pilares de la aprobación de una teoría ridícula y sin bases como la de la evolución.

Desde ya que es imposible hacer ciencia si uno tiene (literalmente) un *deus ex machina* preparado para cada cosa que le cueste. Já, no tenemos una teoría completa para la abiogénesis => Dios lo hizo!

También Harun dá otras razones de la feroz defensa de las teorías de Darwin explicadas en que los científicos son primeros materialistas y luego científicos, por lo que están obligados a crear un aparato de investigación y un conjunto que produzcan explicaciones materialistas sin poder permitirse poner un pie en el umbral divino.

Por supuesto que los científicos deben buscar una explicación materialista a los fenómenos observables, **ese es su trabajo**. Si Mendel hubiera dicho ”estas arvejas son marrones porque dios así lo dispone“ entonces los libros de genética serían muy cortos.

Harun destaca que el evolucionismo no es capaz ni siquiera de explicar una simple proteína y va mas allá en explicar lo que llama las fábulas de transición del agua a la tierra, de la tierra al aire y otras interpretaciones falaces y falsificaciones de los evolucionistas. Recomiendo la lectura que es libre en www.harunyahya.org

Un delirante por vez. Ya le tocará a Harun más adelante.

Es cierto que los mayores defensores de la teoría de la evolución y todas las teorías posteriores que intentan explicar lo inexplicable son ateos o agnósticos, entonces su defensa es también una religión.

No se sigue una cosa de la otra. La compulsión de decir que cualquier cosa que no le gusta a un creyente es la religión de los ateos es irritante.

Desde el punto de vista científico es muy buena la tenacidad de demostrar lo indemostrable pues contribuye al conocimiento. Muchos científicos, paleontólogos, investigadores, etc. se encontrarían sin trabajo si simplemente se aceptara que la creación fue hecha indubitadamente por Dios.

Obviamente. Igual que los médicos, químicos, físicos, y cualquier otro científico. Es más, permítame afirmar que hasta los directores técnicos de fútbol verían su trabajo reducido:

”Vamos perdiendo 1 a 0 y faltan 15 minutos, hay que hacer un cambio“ -- ”No, si vamos perdiendo es porque dios indubitadamente así lo dispuso“.

Así que ya sea por intereses puntuales de desarrollar un proyecto, por premisas del comunismo, por premisas de ateos o agnósticos, por intereses económicos y por la ciencia misma las teorías de la evolución tienen que seguir fingiendo ser una ciencia y festejar todo hallazgo esclarecedor sobre los estudios hechos.

Desde ya que cada hallazgo esclarecedor merece ser festejado. Para eso se estudia y se investiga, para lograr esclarecer cosas. Me pregunto, si el autor desprecia tanto la ciencia, ¿porqué se ha tomado la molestia de escribir toda esta parva de sandeces ”científicas“ según él?

Como el gradualismo es totalmente descartado aún por los mas acérrimos defensores de Darwin, las nuevas teorías quieren mantener todo el esquema de la evolución pero sin gradualismo.

Sí, se descarta lo que parece no funcionar, se mantiene lo que parece funcionar. Eso se llama pensar.

Dicho de manera compleja puede ser atendido los postulados, pero cuando uno los resume para que quede claro la teoría finalmente dice que de un reptil un día nació un ave, o de un ratón nació un día un murciélago.

No, eso es una pavada. Creo distinguir un problema en el entendimiento del autor (bah, uno de tantos). No hay sólo dos puntos en la escala. No es ”cambios infinitesimales o pasamos de un ratón a un murciélago“. Es posible tener cambios grandes, cambios chicos, cambios muy chiquititos y cambios imperceptibles.

Por ejemplo, está determinado como una mutación provocó el desarrollo de la visión tricromática en primates americanos (los africanos no la tienen). Se sabe cual es el gen que provoca el cambio y todo. No es que de golpe hubo un simio volador, es que con algunos cambios menores en el genoma hay un simio que ve distinto.

Todo adornado con una fábula para niños, como que el ratón se veía perseguido así que necesitaba alas, y bueno un día le crecieron, e iba perdiendo la vista y necesitaba no chocarse con los objetos y entonces un día nació un murciélago con radar y de ahí todos los murciélagos aprendieron a nacer u usar el radar. Falta el hada madrina de los cuentos de niños, pero seriamente un científico no puede decir cosas así.

”Un día le crecieron [las alas]“ me suena más a una película de Capra que a biología evolutiva. El autor expresa una caricatura para poder sentirse inteligente mostrando que la caricatura que crea no es correcta. Es un recurso retórico transparente y deshonesto.

Cada vez que aparece un texto de algún hallazgo me tomo el trabajo de leerlo para ver si en algo pudiera cambiar el 0% de probabilidad de las teorías de la evolución puedan alguna vez demostrarse en alguna especie.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "lealo aquí".

Bueno, que opina de una bacteria que de golpe puede digerir un nutriente que antes no podía, eso le gusta como evolución en una especie? [lealo aquí.. lealo aquí: http://myxo.css.msu.edu/lenski/pdf/2008,%20PNAS,%20Blount%20et%20al.pdf](http://myxo.css.msu.edu/lenski/pdf/2008,%20PNAS,%20Blount%20et%20al.pdf)

(cuidado, contiene auténtica ciencia!)

Así, por ejemplo leo un descubrimiento que trata de contestar lo que Darwin se preguntaba, ¿de dónde había salido el murciélago?, una criatura nocturna, voladora y ciega que parecía desafiar las leyes de la selección natural.

El artículo increíblemente sigue dando como cierta la evolución gradual, teoría que ya ha sido abandonada y pasa a preguntarse ¿Qué surgió antes, la capacidad de volar o la de orientarse en la noche por sonidos?, como si ello pudiera ser cierto.

¿Qué artículo? Uno de los conceptos básicos es que si uno tiene referencias debe compartirlas, para que el lector pueda comparar lo que dice realmente con lo que el autor proclama que dice.

Luego relata la aparición en un yacimiento de Wyoming del fósil de una nueva especie de murciélago, y entonces ya define la respuesta buscada: “primero aprendieron a volar, después desarrollaron su peculiar sentido de radar.”

Bueno, si hay un murciélago anterior a los conocidos que volaba y no tenía ecolocación es medio obvio llegar a esa conclusión.

“Lo real es que se encontró casi intacto y con 52 millones de años, el murciélago más antiguo que se conoce. Tenía alas capaces de hacerle volar, lo que se cree que hacía batiéndolas y planeando alternativamente, pero no tenía una cóclea (o caracol) en el oído interno lo bastante grande como para detectar obstáculos mediante el rebote del sonido.”

Parece bastante razonable.

Luego saca conclusiones exquisitas que tienen 100% de fábula y 0% de posibilidad de ajustarse a la realidad, por ejemplo lo nombra antecesor de los murciélagos actuales, aun cuando es una especie distinta y nada hace presumir una relación más que la que a la fábula le conviene.

Bueno, si a priori vamos a asumir que no es posible que sea un ancestro de los murciélagos actuales va a ser difícil mostrar evidencia que te convenza. Veamos por qué podría ser un ancestro de los otros murciélagos.

- 1) Tiene 52 millones de años
- 2) Los otros fósiles de murciélago son más recientes
- 3) Es claramente un murciélago

Entonces o creés que dios creó murciélagos parecidos una y otra vez, y no te vamos a poder convencer sin importar los fósiles que se encuentren.... o es un ancestro de los murciélagos más modernos.

O, por supuesto, no lo era, sino que era otra rama y ambos tienen un ancestro común, pero para el caso es más o menos lo mismo, o es algo que era muy muy muy parecido a un murciélago por evolución convergente, pero eso sería muy raro.

“Agrega que debieron ser diurnos y tener una visión normal en algún momento de su evolución”, la fábula es llevada al desvarío, pues como se verá luego la parte de la visión es una fantasía, y la definición de murciélago, por su origen del latín es de ratón ciego. (al menos sin la visión normal de otros animales)

Ah, bueno, decidir qué es y qué no es un murciélago en base a la etimología de la palabra es nuevo. Es una idiotez, pero es nuevo por lo menos para mí.

Los murciélagos no son todos ciegos. Hay variedades que ven más, y otras que ven menos. Los murciélagos frugívoros suelen tener capacidad de visión similar a la de las ratas, por ejemplo.

Así que según la novedosa clasificación del señor que cito, la mayoría de los murciélagos... no son murciélagos.

Más incoherente es que según su propio relato, al menos en la traducción, se afirma que mirando el esqueleto no se puede deducir científicamente que era diurno.

Y será que no se puede, no es culpa del paleontólogo que no se pueda.

Hasta inventan una mejor teoría, “que algo, probablemente la amenaza de un depredador, les obligó a meterse en las cuevas y fueron perdiendo este sentido (la vista), para aprender a vivir en la oscuridad de dos formas: mediante una potente visión nocturna, o desarrollando el sentido de la eco colocación, es decir, la capacidad de orientarse por sonidos como hace un radar.” Según el artículo esta fue esta última adaptación la que originó a los actuales murciélagos, animales de los que existe una inmensa variedad de especies.

Dado que no tengo acceso al artículo no puedo opinar sobre la paráfrasis presentada.

Además agrega "Era claramente un murciélago, pero no se parecía a ninguno antes conocido. En muchos aspectos, es el eslabón perdido entre los murciélagos y sus ancestros no voladores".

¿Pero no era que no había eslabones perdidos, que en todo el registro fósil no había nada como eso? ¡Seguramente ahora el autor va a demoler esta aseveración!

La realidad fría es que todo es una fábula para niños. Los murciélagos son miles de especies que fueron diseñadas para vivir en las cuevas y volar de noche y regular la cantidad de insectos entre otras funciones.

Ah no, en vez de refutar, simplemente califica de fábula y pasa a otra cosa. Los murciélagos no fueron diseñados. El autor está asumiendo sus propias tesis.

No son ciegos tampoco.

¡¡¡Pero acabás de decir "la definición de murciélago, por su origen del latín es de ratón ciego"!!!! ¡Hace tres renglones lo dijiste!

aunque su mayor visión es acústica. Aunque lo que hayan encontrado sea realmente un murciélago muy diferente, al ser de una familia tan numerosa ¿como probar que es antecesor del resto de las especies y no un primo mas?

Porque el registro fósil no muestra murciélagos con capacidad de ecolocalización anteriores a ése, por ejemplo (no me consta).

¿Sólo los depredadores atacaban los murciélagos, y atacaban a todos? ¿En las cuevas todas las especies perdieron la vista y todas desarrollaron el radar?

La disminución de la capacidad visual de los animales que viven en espacios completamente oscuros está documentada en muchas especies.

Cuando la teoría de la evolución de las especies estaba en sus primeros tiempos, por decirlo de alguna manera en un período romántico, podía ser a primeras luces ser algo posible. Con lo que sabemos ahora de que lo gradual no puede ser posible, cada especie actual debería ser creada por un accidente biológico.

De hecho, por *varios* accidentes.

Un dato interesante de las nuevas teorías es que el dios "tiempo" ya no existe como tal, ya no son necesarios miles de años. Así según las nuevas teorías el ADN de un murciélago con vista y sin radar debería haber mutado a uno con radar sin vista, para transformarlo en una nueva especie.

No, las "nuevas teorías" no dicen eso. El autor está escuchando voces.

Algo así como los cruzamientos dentro de la misma especie, algo como la unión entre yegua y burro para formar una mula, solo que generando una nueva especie.

Los caballos y los burros son *distintas* especies, tienen hasta un número distinto de cromosomas.

Este tipo de accidente tendría que darse para cada especie existente y en una cantidad de casos que permitiera la reproducción sin degradación genética, algo cuya probabilidad es cero. A eso por supuesto hay que sumarle la casualidad de que este accidente biológico justo sólo le sucediera a los murciélagos, porque ya vivían en cuevas y ya estarían perdiendo la vista.

Desvaríos, afirmaciones sin base, refutación de posiciones que sólo él presenta...

La genética no se altera por acciones del individuo. 100% fábula y 0% de ciencia en algo que por supuesto ni podríamos tratar de demostrar de cómo se gana la función de radar justo cuando se la necesitaba.

De nuevo se aprecia gran confusión. La evolución no te da las cosas cuando las quieres. No es "llego tarde al laburo, necesito mutar para mejorar mi glicólisis". No es X-Men. Demuestra ignorancia completa de los mecanismos propuestos para la variabilidad genética y del rol que cumple la selección natural.

Lo real, es que sin base científica, sin demostración, las teorías son sólo fábulas. Teniendo en cuenta que el 20% de las especies vivientes de mamíferos son murciélagos, como lo reconoce el mismo artículo y que aun hoy aparecen frecuentemente nuevas variedades de murciélagos vivos aun no identificadas, poco puede afirmarse sobre una nueva especie fosilizada.

Sí, puede afirmarse que parecía capaz de volar, incapaz de usar ecolocalización, y que parece ser más antiguo que los demás. Que es lo que hicieron. Y con eso uno puede suponer algo más, que es lo que hicieron. No probarlo, pero sí suponerlo.

Lo que descubrieron no tiene que ver con los actuales y de hecho presumen que no tenía radar. Las extremidades traseras más largas y las delanteras más cortas que los actuales murciélagos, "sugieren que vivió en los árboles y que era un buen trepador, parecido en este sentido a los gibones (monos arborícolas) o a los perezosos (mamíferos desdentados de la América tropical)."

El mismo artículo aunque decía que era diurno se contradice afirmando que no ha quedado aún claro es si esta especie vivía de día o de noche. "Un examen de las cuencas de los ojos podría ayudar, ya que los animales nocturnos sin eco-colocación tienen ojos más grandes y, por tanto, cuencas más grandes. Por desgracia, esa zona del cráneo es de lo poco que ha quedado dañado del fósil sin posibilidad de reconstrucción, así que habrá que esperar nuevas evidencias. "

¿Cómo probar que es un antecesor? ¿u otra especie? ¿o simplemente otro individuo de la misma familia?

¿Como probamos que un chimpancé no es un gorila ni un mono tití? ¿Como probamos que el que escribe estas cosas no es un mandril? Con un cuidadoso estudio morfológico, porque leyéndolo no estoy seguro.

Por supuesto que si no encontraran cada tanto algo para hacer un rimbombante anuncio, ¿como seguirían obteniendo presupuesto? En favor de la ciencia, la fábula debe continuar con el mayor agregado posible. Por

supuesto los científicos pueden salir del nivel mundano de la fábula y explicar con razonamientos científicos tan complejos que puede uno lejos perderse sin poder refutar nada, pero bastará que lo haga, para que otros científicos se encarguen de refutar científicamente la propuesta.

“Mami mami, la ciencia es complicada y me hace doler la cabeza, hacé que los hombres malos se vayan”.

Las teorías de la mutación beneficiosa de una especie en otra son hermosas pero nunca abandonan el porcentaje de cero de probabilidad de que ocurran. Ni hay casos observados de la creación de una nueva especie.

Sí, si hay. Ver <http://www.talkorigins.org/faqs/faq-speciation.html>

Por otro lado no hay ninguna teoría que supere el 0% de probabilidad de ser cierta que explique cómo un ser orgánico puede ser hecho de sustancias inorgánicas, ni como una nueva especie sin ningún antecesor puede haber sido creada de especies inferiores.

Eso es un problema distinto. La teoría de la evolución no resuelve el problema de la abiogénesis. Tampoco me ayuda a conseguir donde estacionar en el centro. Son problemas separados.

No hay forma de demostrar o siquiera enunciar una teoría de accidentes biológicos que permitan pasar de una especie a otra.

Sí la hay, y tiene hasta resultados experimentales.

Cuando uno llega a entender la complejidad del ADN es totalmente imposible hoy mantener alguna teoría que permita pensar en que la casualidad, o un accidente realmente pueda crear un ADN nuevo con todas las virtudes de poder ser la guía del nacimiento y reproducción de la nueva especie y que se produzca repetidamente para crear tal cantidad de nuevos organismos que por genética luego logren desarrollarse.

El autor parece creer que es necesario que se produzca la misma idéntica mutación en N individuos para que se cree una nueva especie con una población inicial que le permita mantenerse.

Es un placer para mí aclararle que no funciona así. Se produce una mutación en *un* individuo, que no evita la reproducción con los otros miembros de la población. Si esa mutación es “beneficiosa” entonces se propaga en la población.

Al separarse y aislarse grupos, esta “deriva genética” aumenta hasta que llega a ser imposible la reproducción entre grupos. Ahí se produce uno de los tipos posibles de especiación (hay otros tipos de especiación, por ejemplo por cambios de comportamiento).

Mientras mas se estudia, mayor es la cantidad de eventos simultáneos al azar que debería suceder para que la teoría pudiera ser verdad. Sin aún agregar el componente vida soslayado sistemáticamente, se considera que la naturaleza en su azar debería poder disponer de 100 proteínas diferentes en un mismo lugar para formar una célula.

Formar una proteína tampoco es un evento fácil de imaginar por azar. Si hoy un laboratorio no puede crear una célula viva, aun teniendo todos

los componentes ¿como creer en una teoría que diga que la naturaleza si pudo?. Mucho se ha investigado a partir de los “ladrillos biológicos de Miller” Pero por mucho que se ha intentado no hay una teoría válida.

¿Y? ¿Que hay que hacer? ¿Decir no salió, abandono, debe ser dios? Eso es ser cobarde.

En sus conclusiones, Dean Kenyon dice que es difícil comprender por qué unos científicos están dispuestos a admitir el pensamiento de que una inteligencia de alguna clase (dentro o fuera del universo) estuvo involucrada en la aparición de la primera vida, mientras que otros aparentemente no lo están. Hay un margen de posiciones intermedias que cubren la distancia intelectual entre investigadores como Fox o Miller, y otros como Hoyle, Kenyon o Wilder-Smith.

Por ello, no es necesario salir del mundo racional para dirigirse en dirección de una comprensión científica francamente «teísta» del origen de la vida, a no ser, naturalmente, que uno considere cualquier desviación del «optimismo naturalista» como irracional y anti-intelectual.

Bueno, es que lo es, por lo menos este que vengo leyendo y comentando. Aunque debo agradecer que todavía no haya usado la palabra “matemáticamente” en este capítulo.

Kenyon concluye de que la vida demanda una creación sobrenatural (o al menos una inteligencia creadora dentro del universo) les parecerá prematura a muchos, incluso si se conceden algunos méritos a las principales críticas y argumentaciones positivas creacionistas.

La horrenda gramática me impide opinar. Literalmente no sé que dice ahí.

Y seguirá habiendo los que querrán proseguir con sus actuales enfoques, sin importar lo que muestren los resultados. Siempre podrán decir: «Todavía no hemos encontrado las condiciones experimentales correctas». O, «Sólo hemos estado haciendo experimentos específicamente relacionados con el problema del origen de la vida durante 30 años. Dadnos tiempo. A fin de cuentas, se trata de un problema muy difícil. Al final encontraremos una descripción evolucionista química plausible». O, «Es muy poco lo que se conoce actualmente del sistema genético molecular».

Bueno, sí, si se sabe poco hay que seguir estudiando. Ese siempre es buen consejo. Ojalá el autor lo hubiera seguido en vez de decidir que no valía la pena aprender nada y que su intuición es mejor que la ciencia.

Kenyon finaliza con una lista de seis problemas en el tratar de demostrar el origen de la vida.

Supongamos por un momento que todo lo que dice sea cierto...

- (1) No hay demostración de transición de lo inanimado a materia viviente en ausencia de sustancias preexistentes portadoras de información genética.

¿Y qué?

- (1) No hay demostración en laboratorio de estados precursores realistas.

¿Y qué?

- (1) No hay ninguna descripción teórica convincente de posibles estados precursores.

¿Y qué?

- (1) No hay registro fósil de estados precursores, incluyendo la inexistencia de evidencia geoquímica de un «caldo orgánico prebiótico»

¿Y qué?

- (1) No hay neo-biogénesis (origen de la vida hoy día).

¿Y qué?

- (1) No hay evidencia de vida extra-terrestre ni de estados materiales precursores encontrados en la exploración espacial hasta el día de hoy.

Bueno, está ese meteorito marciano que mencionaste en el capítulo 1...

De nuevo, la ausencia de prueba no es prueba de la ausencia. Decir “no sé” no es lo mismo que “debe haber sido dios”. Una cosa no sigue de la otra, no es lógico, no es científico, es religión maquillada. Y recordemos que se supone que estamos tratando de encontrar a dios “científicamente”.

Comentando acerca de la evidencia de un propósito inteligente inmanente en el universo, y la desgana de muchos científicos a aceptar tal perspectiva, Hoyle a su vez escribía: “Esto es precisamente lo que los científicos ortodoxos no están dispuestos a admitir. Porque podría resultar que hubiera —a falta de una palabra mejor— connotaciones religiosas, y porque los científicos ortodoxos están más interesados en impedir un regreso a los excesos religiosos del pasado que en buscar la verdad”

Hoyle dijo:

”La religión no es más que un intento desesperado para escapar de la horrible posición en que nos encontramos. Aquí estamos, en este universo maravilloso y no tenemos idea de si nuestra existencia tiene algún significado. No es de extrañar que muchos sientan la necesidad de tener alguna creencia que les dé una sensación de seguridad, y no es extraño que se enojen con gente como yo, que les decimos que eso es una ilusión“.

—”The Nature of the Universe“ publicado en 1950.

Ya llevamos tres capítulos y no veo nada de ciencia, mucho menos ciencia aplicada a probar la existencia de dios. Los argumentos del autor parecen reducirse a ignorancia y declamación. Sospecho que no va a mejorar...

I enjoy creating desktop applications. That means I may be a member of a dying breed, since web apps are going to make us all obsolete next week, but I do enjoy doing it.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "urssus".

The bad side of it is, of course that sometimes it's much more convenient to use a web application. For example, I have abandoned my own baby (uRSSus.. urssus: <http://urssus.googlecode.com>

) because google reader is just easier and more convenient to use.

But then I thought... what bothers me of uRSSus? And there are quite a few things!

1. It's not in all computers I may use
That means I will not ever be able to use it exclusively.
2. It's pretty useless without an Internet connection (but so is google reader mostly)
3. Since I can't use it exclusively, I end with feeds on uRSSus that are not on google reader and viceversa.
4. It's freaking slow

So, I decided to see what I could do about that without giving up the good side of uRSSus:

1. It looks much nicer than a web app, because it looks like a desktop app
2. It does things like opening the site instead of showing the feed item (great for partial content feeds)
3. I wrote it (yes, that's a feature for me. I like self-made programs)

So, this attempt at rewriting the desktop RSS reader produced this:

As you can see in the above video, this reader syncs the subscription list to google reader. It will also eventually sync your read/unread posts.

It still can open full sites instead of feed items, it has/will have a heck of an offline mode (full pages captured as images, for example), and... it's very very fast.

It's **much** faster than google reader in Chromium, and **hella** faster than uRSSus. That was done via smarter coding, so it probably means I was braindead before and experienced a minor recovery.

The code is not fit for release (for example, the database schema will change) but you can try it: <http://code.google.com/p/kakawana/source/checkout>

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "'el apache argentino'".

Alejandro Dolina once wrote (and this is from memory that's probably 25 years old) of a round table discussing "What's Tango?", and how after two hours of discussing the nature, characteristics and history of tango, one of the members of the panel picked up a bandoneón, played "El apache argentino".. "el apache argentino": <http://tinysong.com/tmXs>

stood up and left without saying a word.

So, why are Qt and PyQt cool?

Audio player widget:.. _audio player widget:: <http://code.google.com/p/kakawana/source/browse/trunk/kakawana/>

```
\ :c:`# -*- coding: utf-8 -*-`\

\ :kn:`import` \ \ :nn:`sys` \ \ :o:`,` \ \ :nn:`os` \
\ :kn:`from` \ \ :nn:`PyQt4` \ \ :kn:`import` \ \ :n:`QtCore` \ \ :p:`,` \ \
\ :kn:`from` \ \ :nn:`PyQt4.phonon` \ \ :kn:`import` \ \ :n:`Phonon` \
\ :kn:`import` \ \ :nn:`icons_rc` \

\ :k:`class` \ \ :nc:`AudioPlayer` \ \ :p:`(` \ \ :n:`QtGui` \ \ :o:`.` \ \ :n:
  \ :k:`def` \ \ :nf:`__init__` \ \ :p:`(` \ \ :bp:`self` \ \ :p:`,` \ \ :n:

      \ :bp:`self` \ \ :o:`.` \ \ :n:`url` \ \ :o:`=` \ \ :n:`url` \

      \ :n:`QtGui` \ \ :o:`.` \ \ :n:`QWidget` \ \ :o:`.` \ \ :n:`__init__` \
      \ :bp:`self` \ \ :o:`.` \ \ :n:`setSizePolicy` \ \ :p:`(` \ \ :n:`QtGui`
        \ \ :n:`QtGui` \ \ :o:`.` \ \ :n:`QSizePolicy` \ \ :o:`.` \ \ :n:`Pre

      \ :bp:`self` \ \ :o:`.` \ \ :n:`player` \ \ :o:`=` \ \ :n:`Phonon` \ \
        \ \ :n:`Phonon` \ \ :o:`.` \ \ :n:`MediaSource` \ \ :p:`(` \ \ :n:`ur
      \ :bp:`self` \ \ :o:`.` \ \ :n:`player` \ \ :o:`.` \ \ :n:`setTickInter
      \ :bp:`self` \ \ :o:`.` \ \ :n:`player` \ \ :o:`.` \ \ :n:`tick` \ \ :o:

      \ :bp:`self` \ \ :o:`.` \ \ :n:`play_pause` \ \ :o:`=` \ \ :n:`QtGui` \
      \ :bp:`self` \ \ :o:`.` \ \ :n:`play_pause` \ \ :o:`.` \ \ :n:`setIcon
      \ :bp:`self` \ \ :o:`.` \ \ :n:`play_pause` \ \ :o:`.` \ \ :n:`clicked
      \ :bp:`self` \ \ :o:`.` \ \ :n:`player` \ \ :o:`.` \ \ :n:`stateChangee

      \ :bp:`self` \ \ :o:`.` \ \ :n:`slider` \ \ :o:`=` \ \ :n:`Phonon` \ \
      \ :bp:`self` \ \ :o:`.` \ \ :n:`status` \ \ :o:`=` \ \ :n:`QtGui` \ \
      \ :bp:`self` \ \ :o:`.` \ \ :n:`status` \ \ :o:`.` \ \ :n:`setAlignment
        \ \ :n:`QtCore` \ \ :o:`.` \ \ :n:`Qt` \ \ :o:`.` \ \ :n:`AlignVCenter

      \ :bp:`self` \ \ :o:`.` \ \ :n:`download` \ \ :o:`=` \ \ :n:`QtGui` \ \
      \ :bp:`self` \ \ :o:`.` \ \ :n:`download` \ \ :o:`.` \ \ :n:`clicked` \

      \ :n:`layout` \ \ :o:`=` \ \ :n:`QtGui` \ \ :o:`.` \ \ :n:`QHBoxLayout
      \ :n:`layout` \ \ :o:`.` \ \ :n:`addWidget` \ \ :p:`(` \ \ :bp:`self` \ \
      \ :n:`layout` \ \ :o:`.` \ \ :n:`addWidget` \ \ :p:`(` \ \ :bp:`self` \ \
      \ :n:`layout` \ \ :o:`.` \ \ :n:`addWidget` \ \ :p:`(` \ \ :bp:`self` \ \
      \ :n:`layout` \ \ :o:`.` \ \ :n:`addWidget` \ \ :p:`(` \ \ :bp:`self` \ \

\ :k:`def` \ \ :nf:`playClicked` \ \ :p:`(` \ \ :bp:`self` \ \ :p:`,`): \
  \ :k:`if` \ \ :bp:`self` \ \ :o:`.` \ \ :n:`player` \ \ :o:`.` \ \ :n:
    \ :bp:`self` \ \ :o:`.` \ \ :n:`player` \ \ :o:`.` \ \ :n:`pause` \
  \ :k:`else` \ \ :p:`,`: \
```

```

        \ :bp:'self' \ \ :o:'. ' \ \ :n:'player' \ \ :o:'. ' \ \ :n:'play' \ \
\ :k:'def' \ \ :nf:'stateChanged' \ \ :p:'(\ \ \ :bp:'self' \ \ :p:', \ \
\ \ :k:'if' \ \ \ :n:'new' \ \ \ :o:'==' \ \ \ :n:'Phonon' \ \ \ :o:'. ' \ \ :n:
\ \ :bp:'self' \ \ \ :o:'. ' \ \ \ :n:'play_pause' \ \ \ :o:'. ' \ \ \ :n:'setL
\ \ :k:'else' \ \ \ :p:': ' \ \
\ \ :bp:'self' \ \ \ :o:'. ' \ \ \ :n:'play_pause' \ \ \ :o:'. ' \ \ \ :n:'setL

\ :k:'def' \ \ :nf:'tock' \ \ :p:'(\ \ \ :bp:'self' \ \ :p:', \ \ \ :n:'tim
\ \ :n:'time' \ \ \ :o:'=' \ \ \ :n:'time' \ \ \ :o: '/' \ \ \ :mi:'1000' \ \
\ \ :n:'h' \ \ \ :o:'=' \ \ \ :n:'time' \ \ \ :o: '/' \ \ \ :mi:'3600' \ \
\ \ :n:'m' \ \ \ :o:'=' \ \ \ :p:'(\ \ \ :n:'time' \ \ \ :o:'-' \ \ \ :mi:'3600
\ \ :n:'s' \ \ \ :o:'=' \ \ \ :p:'(\ \ \ :n:'time' \ \ \ :o:'-' \ \ \ :mi:'3600
\ \ :bp:'self' \ \ \ :o:'. ' \ \ \ :n:'status' \ \ \ :o:'. ' \ \ \ :n:'setText' \ \ \

\ :k:'def' \ \ :nf:'fetch' \ \ :p:'(\ \ \ :bp:'self' \ \ :p:'):' \ \
\ \ :k:'print' \ \ \ :s:''Should download' \ \ \ :si:'%s' \ \ \ :s:'' \ \ \ :o:

\ :k:'def' \ \ :nf:'main' \ \ :p:'():' \ \
\ \ :n:'app' \ \ \ :o:'=' \ \ \ :n:'QtGui' \ \ \ :o:'. ' \ \ \ :n:'QApplication' \ \ \
\ \ :n:'window' \ \ \ :o:'=' \ \ \ :n:'AudioPlayer' \ \ \ :p:'(\ \ \ :n:'sys' \ \ \ :o:
\ \ :n:'window' \ \ \ :o:'. ' \ \ \ :n:'show' \ \ \ :p:'()' \ \
\ \ :c:'# It's exec_ because exec is a reserved word in Python' \ \
\ \ :n:'sys' \ \ \ :o:'. ' \ \ \ :n:'exit' \ \ \ :p:'(\ \ \ :n:'app' \ \ \ :o:'. ' \ \ \ :n:

\ :k:'if' \ \ \ :n:'__name__' \ \ \ :o:'==' \ \ \ :s:''__main__' \ \ \ :p:': ' \ \
\ \ :n:'main' \ \ \ :p:'()' \ \

```

Video player widget:.. _video player widget:: <http://code.google.com/p/kakawana/source/browse/trunk/kakawana/>

```

\ :kn:'import' \ \ \ :nn:'sys' \ \ \ :o:', ' \ \ \ :nn:'os' \ \
\ :kn:'from' \ \ \ :nn:'PyQt4' \ \ \ :kn:'import' \ \ \ :n:'QtCore' \ \ \ :p:', ' \ \ \
\ :kn:'from' \ \ \ :nn:'PyQt4.phonon' \ \ \ :kn:'import' \ \ \ :n:'Phonon' \ \
\ :kn:'import' \ \ \ :nn:'icons_rc' \ \

\ :k:'class' \ \ \ :nc:'VideoPlayer' \ \ \ :p:'(\ \ \ :n:'QtGui' \ \ \ :o:'. ' \ \ \ :n:
\ \ :k:'def' \ \ \ :nf:'__init__' \ \ \ :p:'(\ \ \ :bp:'self' \ \ \ :p:', \ \ \ :n:

\ \ :bp:'self' \ \ \ :o:'. ' \ \ \ :n:'url' \ \ \ :o:'=' \ \ \ :n:'url' \ \

\ \ :n:'QtGui' \ \ \ :o:'. ' \ \ \ :n:'QWidget' \ \ \ :o:'. ' \ \ \ :n:'__init__' \ \
\ \ :bp:'self' \ \ \ :o:'. ' \ \ \ :n:'setSizePolicy' \ \ \ :p:'(\ \ \ :n:'QtGui
\ \ \ :n:'QtGui' \ \ \ :o:'. ' \ \ \ :n:'QSizePolicy' \ \ \ :o:'. ' \ \ \ :n:'Pre

\ \ :bp:'self' \ \ \ :o:'. ' \ \ \ :n:'player' \ \ \ :o:'=' \ \ \ :n:'Phonon' \ \ \
\ \ :bp:'self' \ \ \ :o:'. ' \ \ \ :n:'player' \ \ \ :o:'. ' \ \ \ :n:'load' \ \ \ :p:
\ \ :bp:'self' \ \ \ :o:'. ' \ \ \ :n:'player' \ \ \ :o:'. ' \ \ \ :n:'mediaObject
\ \ :bp:'self' \ \ \ :o:'. ' \ \ \ :n:'player' \ \ \ :o:'. ' \ \ \ :n:'mediaObject

```

```

\ :bp:'self'\ \ :o:'. '\ \ :n:'play_pause'\ \ :o:'='\ \ :n:'QtGui'
\ :bp:'self'\ \ :o:'. '\ \ :n:'play_pause'\ \ :o:'. '\ \ :n:'setIcon'
\ :bp:'self'\ \ :o:'. '\ \ :n:'play_pause'\ \ :o:'. '\ \ :n:'clicked'
\ :bp:'self'\ \ :o:'. '\ \ :n:'player'\ \ :o:'. '\ \ :n:'mediaObject'

\ :bp:'self'\ \ :o:'. '\ \ :n:'slider'\ \ :o:'='\ \ :n:'Phonon'\ \

\ :bp:'self'\ \ :o:'. '\ \ :n:'status'\ \ :o:'='\ \ :n:'QtGui'\ \
\ :bp:'self'\ \ :o:'. '\ \ :n:'status'\ \ :o:'. '\ \ :n:'setAlignment'
\ :n:'QtCore'\ \ :o:'. '\ \ :n:'Qt'\ \ :o:'. '\ \ :n:'AlignVCenter'

\ :bp:'self'\ \ :o:'. '\ \ :n:'download'\ \ :o:'='\ \ :n:'QtGui'\ \
\ :bp:'self'\ \ :o:'. '\ \ :n:'download'\ \ :o:'. '\ \ :n:'clicked'\ \
\ :n:'topLayout'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QVBoxLayout'
\ :n:'topLayout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'('\ \ :bp:'self'
\ :n:'layout'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QHBoxLayout'
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'('\ \ :bp:'self'
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'('\ \ :bp:'self'
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'('\ \ :bp:'self'
\ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget'\ \ :p:'('\ \ :bp:'self'
\ :n:'topLayout'\ \ :o:'. '\ \ :n:'addLayout'\ \ :p:'('\ \ :n:'layout'
\ :bp:'self'\ \ :o:'. '\ \ :n:'setLayout'\ \ :p:'('\ \ :n:'topLayout'

\ :k:'def'\ \ :nf:'playClicked'\ \ :p:'('\ \ :bp:'self'\ \ :p:'):\ \
\ :k:'if'\ \ :bp:'self'\ \ :o:'. '\ \ :n:'player'\ \ :o:'. '\ \ :n:
\ :bp:'self'\ \ :o:'. '\ \ :n:'player'\ \ :o:'. '\ \ :n:'pause'\ \
\ :k:'else'\ \ :p:':'\ \
\ :bp:'self'\ \ :o:'. '\ \ :n:'player'\ \ :o:'. '\ \ :n:'play'\ \

\ :k:'def'\ \ :nf:'stateChanged'\ \ :p:'('\ \ :bp:'self'\ \ :p:', '\ \
\ :k:'if'\ \ :n:'new'\ \ :o:'=='\ \ :n:'Phonon'\ \ :o:'. '\ \ :n:
\ :bp:'self'\ \ :o:'. '\ \ :n:'play_pause'\ \ :o:'. '\ \ :n:'setL
\ :k:'else'\ \ :p:':'\ \
\ :bp:'self'\ \ :o:'. '\ \ :n:'play_pause'\ \ :o:'. '\ \ :n:'setL

\ :k:'def'\ \ :nf:'tock'\ \ :p:'('\ \ :bp:'self'\ \ :p:', '\ \ :n:'tin
\ :n:'time'\ \ :o:'='\ \ :n:'time'\ \ :o: '/'\ \ :mi:'1000'\ \
\ :n:'h'\ \ :o:'='\ \ :n:'time'\ \ :o: '/'\ \ :mi:'3600'\ \
\ :n:'m'\ \ :o:'='\ \ :p:'('\ \ :n:'time'\ \ :o: '-'\ \ :mi:'3600'
\ :n:'s'\ \ :o:'='\ \ :p:'('\ \ :n:'time'\ \ :o: '-'\ \ :mi:'3600'
\ :bp:'self'\ \ :o:'. '\ \ :n:'status'\ \ :o:'. '\ \ :n:'setText'\ \

\ :k:'def'\ \ :nf:'fetch'\ \ :p:'('\ \ :bp:'self'\ \ :p:'):\ \
\ :k:'print'\ \ :s:'Should download '\ \ :si:'%s'\ \ :s:' '\ \ :o:

\ :k:'def'\ \ :nf:'main'\ \ :p:'()':\ \
\ :n:'app'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QApplication'\ \
\ :n:'window'\ \ :o:'='\ \ :n:'VideoPlayer'\ \ :p:'('\ \ :n:'sys'\ \ :o:
\ :n:'window'\ \ :o:'. '\ \ :n:'show'\ \ :p:'()'\ \
\ :c:'# It's exec_ because exec is a reserved word in Python'\ \

```

```

\ :n:'sys'\ \ :o:'. '\ \ :n:'exit'\ \ :p:'('\ \ :n:'app'\ \ :o:'. '\ \ :n:
\ :k:'if'\ \ :n:'__name__'\ \ :o:'=='\ \ :s:'"__main__"\ \ :p:': '\
\ :n:'main'\ \ :p:'()' \

```

...

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "my book".

I do almost all my business writing (and [my book.. my book: http://nomuerde.netmanagers.com.ar](http://nomuerde.netmanagers.com.ar)

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "rst2pdf".

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "rst2pdf".

) using restructured text. And when I want to produce print-quality output, I tend to use my own tool, [rst2pdf.. rst2pdf: http://rst2pdf.googlecode.com](http://rst2pdf.googlecode.com)

It's popular, surely my most popular program, but very few know it's also extremely easy to extend (hat tip to Patrick Maupin, who wrote this part!). And not only that, but you can make it do some amazing stuff with a little effort.

To show that, let's create the most dazzling section headings known to man (Let's see you do what this baby can do in LaTeX ;-).

First: define the problem.

The titles `rst2pdf` can produce are boring. If you pull every lever and push every button, you may end with the title text, in a Comic Sans, right-aligned, in pink lettering, with avocado-green background and a red border.

system-message
WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "but i have done it".

And that's as far as the customization capabilities go using stylesheets. That's *usually* enough, because `rst2pdf` is not meant for brochures or something like that (but I have done it.. but i have done it: <http://lateral.netmanagers.com.ar/static/machete.pdf>).

The real problem is that when you get all graphic-designer on `rst2pdf`, you lose document structure, because you are not being semantic.

Second: define the goal.

So, imagine you want to make a heading that looks like this:

The image is taken from the library of congress with some light (and bad) gimping by me to leave that empty space at the left, and the title was added using Inkscape.

Can you do that with `rst2pdf`? Hell no you can't. Not without coding. So let's code an extension that lets you create any heading you like within the limits of Inkscape!

system-message

WARNING/2 in everything.txt, line 3

Duplicate explicit target name: "svg template for the headings".

First, we create a [SVG template for the headings](#).. svg template for the headings: /static/titletemplate.svg

(it's a bit big because it has the bitmap embedded).

Three: the image-heading flowable

Suppose you have an image of the heading just like the one above. How would you draw that in a PDF? In reportlab, you do that using `flowables` which are elements that compose the `story` that is your document. These `flowables` are arranged in pages, and that's your PDF.

If you are doing a heading, there's a bit more, in that you need to add a bookmark, so it appears on the PDF table of contents.

So, here's a flowable that does just that. It's cobbled from pieces inside `rst2pdf`, and is basically an unholy mix of `Heading` and `MyImage`:

```
\ :k:'class'\ \ :nc:'FancyHeading'\ \ :p:'(\ \ :n:'MyImage'\ \ :p:'):\ \
\ :sd:'''This is a cross between the Heading flowable, that adds outline
entries so you have a PDF TOC, and MyImage, that draws images'''\ \

\ :k:'def'\ \ :nf:'__init__'\ \ :p:'(\ \ :bp:'self'\ \ :p:','\ \ :o:'\ \
\ :c:'# The initialization is taken from rst2pdf.flowables.Heading'\ \
\ :bp:'self'\ \ :o:'.\ \ \ :n:'stext'\ \ \ :o:'='\ \ \ :n:'kwargs'\ \ \ :o:'\ \
\ :c:'# Cleanup title text'\ \
\ :bp:'self'\ \ \ :o:'.\ \ \ :n:'stext'\ \ \ :o:'='\ \ \ :n:'re'\ \ \ :o:'.\ \
\ :bp:'self'\ \ \ :o:'.\ \ \ :n:'stext'\ \ \ :o:'='\ \ \ :bp:'self'\ \ \ :o:'\ \

\ :c:'# Stuff needed for the outline entry'\ \
\ :bp:'self'\ \ \ :o:'.\ \ \ :n:'snum'\ \ \ :o:'='\ \ \ :n:'kwargs'\ \ \ :o:'\ \
\ :bp:'self'\ \ \ :o:'.\ \ \ :n:'level'\ \ \ :o:'='\ \ \ :n:'kwargs'\ \ \ :o:'\ \
\ :bp:'self'\ \ \ :o:'.\ \ \ :n:'parent_id'\ \ \ :o:'='\ \ \ :n:'kwargs'\ \ \

\ :n:'MyImage'\ \ \ :o:'.\ \ \ :n:'__init__'\ \ \ :p:'(\ \ :bp:'self'\ \ \

\ :k:'def'\ \ \ :nf:'drawOn'\ \ \ :p:'(\ \ :bp:'self'\ \ \ :p:','\ \ \ :n:'canv'\ \

\ :c:'## These two lines are magic.'\ \
\ :c:'#if isinstance(self.parent_id, tuple):\ \
\ :c:'\self.parent_id=self.parent_id[0]\ \

\ :c:'# Add outline entry. This is copied from rst2pdf.flowables.heading'\ \
\ :n:'canv'\ \ \ :o:'.\ \ \ :n:'bookmarkHorizontal'\ \ \ :p:'(\ \ \ :bp:'self'\ \

\ :k:'if'\ \ \ :n:'canv'\ \ \ :o:'.\ \ \ :n:'firstSect'\ \ \ :p:':\ \
\ :n:'canv'\ \ \ :o:'.\ \ \ :n:'sectName'\ \ \ :o:'='\ \ \ :bp:'self'\ \
\ :n:'canv'\ \ \ :o:'.\ \ \ :n:'firstSect'\ \ \ :o:'='\ \ \ :bp:'False'\ \
```

```

\ :k:'if'\ \ :bp:'self'\ \ :o:'. '\ \ :n:'snum'\ \ :ow:'is'\ \ :
\ :n:'canv'\ \ :o:'. '\ \ :n:'sectNum'\ \ :o:'='\ \ :bp:'sel
\ :k:'else'\ \ :p:': '\
\ :n:'canv'\ \ :o:'. '\ \ :n:'sectNum'\ \ :o:'='\ \ :s:'"' \
\ :n:'canv'\ \ :o:'. '\ \ :n:'addOutlineEntry'\ \ :p:'('\ \ :bp:'self
\ :bp:'self'\ \ :o:'. '\ \ :n:'parent_id'\ \
\ :nb:'int'\ \ :p:'('\ \ :bp:'self'\ \ :o:
\ :c:'# And let MyImage do all the drawing'\
\ :n:'MyImage'\ \ :o:'. '\ \ :n:'drawOn'\ \ :p:'('\ \ :bp:'self'\ \ :p

```

And how do we tell rst2df to use that instead of a regular Heading? by overriding the TitleHandler class. Here's where the extension magic kicks in.

If you define, in an extension, a class like this:

```

\ :k:'class'\ \ :nc:'FancyTitleHandler'\ \ :p:'('\ \ :n:'genelements'\ \ :

```

Then that class will handle all docutils nodes of class docutils.nodes.title. Here, I just took rst2pdf.genelements.HandleTitle and changed how it works for level-1 headings, making it generate a FancyHeading instead of a Heading... and that's all there is to it.

```

\ :k:'class'\ \ :nc:'FancyTitleHandler'\ \ :p:'('\ \ :n:'genelements'\ \ :
\ :sd:'''''
This class will handle title nodes.

```

It takes a "titletemplate.svg", replaces TITLEGOESHERE with the actual title text, and draws that using the FancyHeading flowable (see below).

Since this class is defined in an extension, it effectively replaces rst2pdf.genelements.HandleTitle.

```

''''\

```

```

\ :k:'def'\ \ :nf:'gather_elements'\ \ :p:'('\ \ :bp:'self'\ \ :p:', '\
\ :c:'# This method is copied from the HandleTitle class'\
\ :c:'# in rst2pdf.genelements.'\

\ :c:'# Special cases: (Not sure this is right ;-)\ \
\ :k:'if'\ \ :nb:'isinstance'\ \ :p:'('\ \ :n:'node'\ \ :o:'. '\ \ :n:
\ :c:'#node.elements = [Paragraph(client.gen_pdftext(node), '\
\ :c:'#client.styles['title'])]\ \
\ :c:'# The visible output is now done by the cover template'\
\ :n:'node'\ \ :o:'. '\ \ :n:'elements'\ \ :o:'='\ \ :p:'[]'\ \
\ :n:'client'\ \ :o:'. '\ \ :n:'doc_title'\ \ :o:'='\ \ :n:'node
\ :n:'client'\ \ :o:'. '\ \ :n:'doc_title_clean'\ \ :o:'='\ \ :n:
\ :k:'elif'\ \ :nb:'isinstance'\ \ :p:'('\ \ :n:'node'\ \ :o:'. '\ \ :n:
\ :n:'node'\ \ :o:'. '\ \ :n:'elements'\ \ :o:'='\ \ :p:'[]'\ \ :n:
\ :n:'client'\ \ :o:'. '\ \ :n:'styles

```

```

\ :k:'elif'\ \ :nb:'isinstance'\ \ :p:'('\ \ :n:'node'\ \ :o:'. '\ \
\ :n:'node'\ \ :o:'. '\ \ :n:'elements'\ \ :o:'='\ \ :p:'['\ \
\ :n:'client'\ \ :o:'. '\ \ :n:'styles'
\ :k:'elif'\ \ :nb:'isinstance'\ \ :p:'('\ \ :n:'node'\ \ :o:'. '\ \
\ :n:'node'\ \ :o:'. '\ \ :n:'elements'\ \ :o:'='\ \ :p:'['\ \
\ :n:'client'\ \ :o:'. '\ \ :n:'styles'
\ :k:'elif'\ \ :nb:'isinstance'\ \ :p:'('\ \ :n:'node'\ \ :o:'. '\ \
\ :n:'node'\ \ :o:'. '\ \ :n:'elements'\ \ :o:'='\ \ :p:'['\ \
\ :n:'client'\ \ :o:'. '\ \ :n:'styles'
\ :k:'else'\ \ :p:':'\
\ :c:'# Section/Subsection/etc.'\
\ :n:'text'\ \ :o:'='\ \ :n:'client'\ \ :o:'. '\ \ :n:'gen_pdfte'
\ :n:'fch'\ \ :o:'='\ \ :n:'node'\ \ :o:'. '\ \ :n:'children'\ \
\ :k:'if'\ \ :nb:'isinstance'\ \ :p:'('\ \ :n:'fch'\ \ :p:', '\
\ :n:'fch'\ \ :p:'['\ \ :s:'classes'\ \ :p:']'\ \ :o:'=='\
\ :n:'snum'\ \ :o:'='\ \ :n:'fch'\ \ :o:'. '\ \ :n:'astext'\
\ :k:'else'\ \ :p:':'\
\ :n:'snum'\ \ :o:'='\ \ :bp:'None'\
\ :n:'key'\ \ :o:'='\ \ :n:'node'\ \ :o:'. '\ \ :n:'get'\ \ :p:
\ :n:'maxdepth'\ \ :o:'='\ \ :mi:'4'\
\ :k:'if'\ \ :n:'reportlab'\ \ :o:'. '\ \ :n:'Version'\ \ :o:'>
\ :n:'maxdepth'\ \ :o:'='\ \ :mi:'6'\

\ :c:'# The parent ID is the refid + an ID to make it unique for
\ :n:'parent_id'\ \ :o:'='\ \ :p:'('\ \ :n:'node'\ \ :o:'. '\ \ :n
\ :k:'if'\ \ :n:'client'\ \ :o:'. '\ \ :n:'depth'\ \ :o:'>'\ \
\ :n:'node'\ \ :o:'. '\ \ :n:'elements'\ \ :o:'='\ \ :p:'['\
\ :n:'client'\ \ :o:'. '\ \ :n:'styles'\ \ :p:'['\ \ :
\ :n:'level'\ \ :o:'='\ \ :n:'client'\ \ :o:'. '\ \ :n
\ :n:'parent_id'\ \ :o:'='\ \ :n:'parent_id'\ \ :p:', '\
\ :n:'node'\ \ :o:'='\ \ :n:'node'\ \ :p:', '\
\ :p:']]\
\ :k:'else'\ \ :p:':'\ \ :c:'# This is an important title, do ou
\ :c:'# Hack the title template SVG'\
\ :n:'tfile'\ \ :o:'='\ \ :nb:'open'\ \ :p:'('\ \ :s:'titl
\ :n:'tdata'\ \ :o:'='\ \ :n:'tfile'\ \ :o:'. '\ \ :n:'read
\ :n:'tfile'\ \ :o:'. '\ \ :n:'close'\ \ :p:'()'\
\ :n:'tfile'\ \ :o:'='\ \ :n:'tempfile'\ \ :o:'. '\ \ :n:'Na
\ :n:'tfname'\ \ :o:'='\ \ :n:'tfile'\ \ :o:'. '\ \ :n:'name
\ :n:'tfile'\ \ :o:'. '\ \ :n:'write'\ \ :p:'('\ \ :n:'tdata'\
\ :n:'tfile'\ \ :o:'. '\ \ :n:'close'\ \ :p:'()'\

\ :c:'# Now tfname contains a SVG with the right title.'\
\ :c:'# Make rst2pdf delete it later.'\
\ :n:'client'\ \ :o:'. '\ \ :n:'to_unlink'\ \ :o:'. '\ \ :n:'ap

\ :n:'e'\ \ :o:'='\ \ :n:'FancyHeading'\ \ :p:'('\ \ :n:'tf
\ :n:'client'\ \ :o:'='\ \ :n:'client'\ \ :p:', '\ \ :n:
\ :n:'text'\ \ :o:'='\ \ :n:'text'\ \ :p:', '\ \ :n:'leve

```

```

\ :n:'node'\ \ :o:'. '\ \ :n:'elements'\ \ :o:'='\ \ :p:'[ '\
\ :k:'if'\ \ :n:'client'\ \ :o:'. '\ \ :n:'depth'\ \ :o:'<='\
\ :n:'node'\ \ :o:'. '\ \ :n:'elements'\ \ :o:'. '\ \ :n:'insert'\
\ :k:'return'\ \ :n:'node'\ \ :o:'. '\ \ :n:'elements'\

```

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "in svn".

The full extension is in SVN.. in svn: <http://code.google.com/p/rst2pdf/source/browse/trunk/rst2pdf/extensions/fancytitles.py>

and you can try it this way:

```
[fancytitles]$ rst2pdf -e fancytitles -e inkscape demo.txt -b1
```

You need to enable the Inkscape extension so the SVG will look nice. And here's the output:

You can override how **any** element is handled. That's being extensible :-)

Edited

I removed the countdown because it was wrong (timezone problem). Check <http://revista.python.org.ar> for one that works :-)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "pet - python entre todos".

I have been dragged into YAP (Yet Another Project) and in this case it's the first-issue-ever of **PET - Python Entre Todos**.. pet - python entre todos: <http://revistapyar.netmanagers.com.ar>

a python online magazine in spanish.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "pyar".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "pyar".

This is an emergent output of **PyAr**.. pyar: <http://python.org.ar>

the online Python community of Argentina, and it may never have a second issue unless we get a lot of help, so pay attention to the first one :-)

I have been doing webmastering, editing and PDF production, Emiliano Dalla Verde Marcozzi has been helping everywhere and getting articles.

The first issue is some 45 A5 pages (or 23 A4 ones) in small type, so it's not exactly huge, but it has half a dozen articles aimed at different python levels.

It has been a lot of fun so far, I hope it's well received!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "jornadas del sur".

I will be speaking at the [Jornadas del Sur](http://www.jornadasdelsur.org.ar/2010/).. jornadas del sur: <http://www.jornadasdelsur.org.ar/2010/>
in Bahía Blanca this weekend (August 14/15 and 16).

For a change, I will not be giving my old tired talks, but a brand new one, called "The Amateur", and probably a lightning talk or something else.

The usual offer of free beer will not be possible this year, so: If you mention this blog in the QA session, you get... free candy!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "charlas abiertas 2010".

After that, I will be speaking at FM La Tribu on saturday August 21st in the [Charlas Abiertas 2010](http://python.org.ar/pyar/CharlasAbiertas2010).. charlas abiertas 2010: <http://python.org.ar/pyar/CharlasAbiertas2010>

where I will be speaking about a lot of things between virtualenv and nose and tox and other testing-related topics.

Today at 00:00:00 GMT-3 [PET: Python entre todos](http://python.org.ar/pyar/CharlasAbiertas2010).. _pet: python entre todos: <http://revista.python.org.ar>

was indeed launched, in time (arbitrary but forced) and in budget (\$0).

So, what did I learn? I learned a lot!

- The only thing you need to publish an e-mag is time and content.
- Time can be converted into content, but if you write everything yourself it's a blog, not a magazine. Luckily, PET found great contributors.
- If you want utilitarian design, rst2pdf can do the job
- In fact, it can do it better than other tools in some ways
 - I can push a fixed version of the PDFs in 5 minutes for all layouts. How much would it take me using Scribus or other DTP? In a magazine where correctness matters, that's a big deal.
 - TOCs are better than in most amateur PDF magazines I've seen. The in-content-TOC is clickable, and the PDF TOC is perfect.
 - Page numbers in the PDF TOC make sense (no, the cover is not page 1)
 - I am producing 6 PDF versions: A4(bw, colour), A5(bw,colour), Booklet(bw, colour) and I could add any other I want in a few minutes.
- I learned about PDF imposition!

Let's explain the last one:

Suppose you want to print a small booklet, and you have 32 pages of content. How do you do that?

The easiest way is to print it 2-up double-sided in A4 paper so that you can stack the pages, fold them down the middle, staple them, and get a nice A5 booklet.

The problem is that the page ordering is hard to get right. For example, for a 4-page booklet, you need to print one A4 page with pages 4-1 on one side and 2-3 on the other. For an 8 page booklet it's 8-1,2-7,3-6,4-5.

Luckily there's a way to get this done automatically:

1. Install podofdo 3. Get booklet-A4.plan (see below)
2. Run this:

```
podofdoimpose my-A5-pages.pdf my-booklet.pdf booklet-A4.plan lua
```

booklet-A4.plan is this:

```
\ :cl:'---Generic Booklet (A4) '\
\ :cl:'---'\
\ :cl:'---It is said generic as it will try to determine'\
\ :cl:'---automatically how to fit the booklet onto A4'\
\ :cl:'---paper sheets, scaling pages if necessary. '\
\ :cl:'---it is well suited for office documents for'\
\ :cl:'---which you do not care too much about resulting'\
\ :cl:'---imposition artefacts since it manages to save'\
\ :cl:'---paper;\
\ :cl:'---'\
\ :cl:'-- print("Booklet") '\
\ :cl:'-- We output an A4 booklet'\
\ :n:'PageWidth'\ \ :o:'='\ \ :mf:'595.27559'\
\ :n:'PageHeight'\ \ :o:'='\ \ :mf:'841.88976'\

\ :nb:'print'\ \ :p:'('\ \ :s2:'"\ \ :s:'PageCount"\ \ :p:','\ \ :n:'Page

\ :cl:'-- We assume that H > W'\
\ :cl:'-- Argh, we now can do better since we have "if" ;-)\
\ :cl:'-- Scale = PageHeight / (2*SourceWidth)\
\ :k:'if'\ \ :p:'('\ \ :n:'SourceWidth'\ \ :o:'<='\ \ :n:'SourceHeight'\
\ :k:'then'\
\ :cl:'-- If you A5 pages are not really A5, uncomment the next line'\
\ :cl:'-- Scale = PageHeight / (2*SourceWidth)\
\ :n:'Scale'\ \ :o:'='\ \ :mi:'1'\
\ :n:'rot'\ \ :o:'='\ \ :mi:'90'\
\ :n:'xof'\ \ :o:'='\ \ :n:'SourceHeight'\
\ :n:'yofRA'\ \ :o:'='\ \ :mi:'0'\
\ :n:'yofRB'\ \ :o:'='\ \ :n:'SourceWidth'\
\ :n:'yofVA'\ \ :o:'='\ \ :mi:'0'\
\ :n:'yofVB'\ \ :o:'='\ \ :n:'SourceWidth'\
\ :k:'else'\
\ :cl:'-- If you A5 pages are not really A5, uncomment the next line'\
\ :cl:'-- Scale = PageHeight / (2*SourceHeight)\
\ :n:'Scale'\ \ :o:'='\ \ :mi:'1'\
\ :n:'rot'\ \ :o:'='\ \ :mi:'0'
```

```

\ :n:'xof'\ \ :o:'='\ \ :mi:'0'\ \ :p:';'\
\ :n:'yofRA'\ \ :o:'='\ \ :mi:'0'\
\ :n:'yofRB'\ \ :o:'='\ \ :n:'SourceHeight'\
\ :n:'yofVA'\ \ :o:'='\ \ :n:'SourceHeight'\
\ :n:'yofVB'\ \ :o:'='\ \ :mi:'0'\
\ :k:'end'\

\ :k:'do'\
\ :n:'rest'\ \ :o:'='\ \ :n:'PageCount'\ \ :o:'%\ \ \ :mi:'4'\
\ :n:'totp'\ \ :o:'='\ \ :n:'PageCount'\
\ :k:'if'\ \ :n:'rest'\ \ :o:'~='\ \ :mi:'0'\
\ :k:'then'\
\ :n:'totp'\ \ :o:'='\ \ :n:'totp'\ \ :o:'+'\ \ :p:'('\ \ \ :mi:
\ :k:'end'\
\ :n:'inc'\ \ :o:'='\ \ :mi:'0'\
\ :n:'count'\ \ :o:'='\ \ :mi:'0'\
\ :n:'imax'\ \ :o:'='\ \ :n:'totp'\ \ :o:'/'\ \ :mi:'4'\
\ :k:'while'\ \ :n:'count'\ \ :o:'<'\ \ :n:'imax'\
\ :k:'do'\
\ :c1:'-- We assume that podof impose will discard invalid records
\ :c1:'-- such as those with source page greater than PageCount'\
\ :c1:'-- print(totp, inc, rot, xof,yofRA, yofRA, yofVA, yofVB)\
\ :c1:'-- Recto'\
\ :n:'PushRecord'\ \ :p:'('\ \ :n:'totp'\ \ :o:'-\ \ \ :n:'inc'\
\ :n:'PushRecord'\ \ :p:'('\ \ :n:'inc'\ \ :o:'+'\ \ \ :mi:'1'\ \
\ :c1:'-- Verso'\
\ :n:'PushRecord'\ \ :p:'('\ \ :n:'inc'\ \ :o:'+'\ \ \ :mi:'2'\ \
\ :n:'PushRecord'\ \ :p:'('\ \ :n:'totp'\ \ :o:'-\ \ \ :p:'('\ \ \ :n:

\ :n:'count'\ \ :o:'='\ \ :n:'count'\ \ :o:'+'\ \ \ :mi:'1'\
\ :n:'inc'\ \ :o:'='\ \ :n:'inc'\ \ :o:'+'\ \ \ :mi:'2'\
\ :k:'end'\
\ :k:'end'\

```

That code is taken from here: <http://www.oep-h.com/impose/>

And voilà, you get a scrambled PDF with the pages in exactly the right order (and empty pages added as needed).

I never imagined this could work. I am **not** testing it. But hey, you gotta admire the chutzpah!

On the news today (the newspaper is Tiempo Argentino, sorry about the unreadable photo):

It says "In Spain it amounts to less than 340 thousand customers a month, or 0.006% of the total".

This is talking about cellphone users. Now, how many cell phone users are there in Spain?

According to this newspaper, 340000 are 0.006% of the total, so... 5 666 666 667 (aprox.), therefore, Spain has roughly one cell phone for each man, woman and child in the world.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "my old one".

I just got a new phone because [my old one](http://www.flickr.com/photos/ralsina/4476747320/).. my old one: <http://www.flickr.com/photos/ralsina/4476747320/>

disappeared. I got a Samsung Star, which is not a smartphone, but what they call a featurephone, which seems to mean "it does a lot of things but is cheap, can only be coded in Java and doesn't run android".

Here's some of the features of this baby:

- Touchscreen, 400x240
- Fake GPS (gives you a 400 meters radius of your location. Good enough for me in the city)
- Webkit-based browser that works surprisingly well (it comes with the LGPL as a document and you can't delete it :-)
- Accelerometer
- FM radio
- A 2GB micro-SD card
- 3.2MP camera (takes video at 320x240).

The camera can take decent photos in good conditions:

And it even has some "advanced" features (by which I mean: things my cheap dedicated camera doesn't do), like smile shutter and panoramic shots:

Other than that... well, it's a phone, what can I say. It was "cheap", which means "it costs more than I want to pay for it, but less than the alternatives".

So, how am I taking advantage of it...

First: I have not had a decent pocket-sized ebook reader since my last Clie died. So, I looked for software to do that.

It turns out that the world of Midlets, as feature phone apps are often called, is a weird place, where things are quite hard to find for the uninitiated.

There is a sort of "app shop" at <http://getjar.com> but it's by no means comprehensive, and often things are quite hard to find.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "their home page, russian only".

After lots of looking, I found a good (I may even say **very good**) program called Foliant. Here's [their home page, russian only](http://foliant-reader.com/).. their home page, russian only: <http://foliant-reader.com/>

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "russian-only site".

... it's weird, my favourite Palm ebook reader, Palm Fiction also has a [russian-only site](http://palmfiction.sf.net).. russian-only site: palmfiction.sf.net

!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "this tool".

Of course, the fonts sucked (way too large) but it turns out you can convert TTF fonts using [this tool](http://foliant-reader.com/foiant-fonts/). this tool: <http://foliant-reader.com/foiant-fonts/>

so I am now back to the lovely Droid Sans I am used to.

Another nice thing about the new phone is that it can actually play media (yes, my previous phone was so crappy it didn't even play mp3). But... not every kind of media. For audio, just use mp3. For video... it's a bit more complicated.

Here's the short version:

- 1) Convert using HandBrake, ffmpeg MP4 video, AAC audio.
- 2) Always use the .mp4 extension, the phone is not smart enough to know what a .m4v is.
- 3) Don't make the video larger than 320x240.

Number 3 is a problem. For example, you may have a video in a wide screen format, like 640x272, which is a 2.35:1 ratio.

The obvious thing is to cap the width to 320, and that would give you a 320x136 video.

Well, that's **wrong**. What you should do is find the right height, keeping aspect ratio, for a 400px width. In this case, that would be a 400x170.

But you can't use a 400x170 video! which is why you will use 320x170, and on playback tell the phone to stretch it and ignore aspect ratio. And voila, 400x170 and the correct aspect ratio.

The difference? 320x130 has only 41600 pixels, while 320*170 has 54400, which means you get a 30% better picture.

Yes, it's tiny (3 inches) but it looks pretty sharp, and depending on the kind of material you are watching, it works.

UPDATE: Foliant is better than I thought, once you get the Samsung-specific version. It's fullscreen (no silly soft buttons) and the screen rotates automatically using the accelerometer. It's a pleasure to use, and the UI is very nice.

Because it worked once, let's do it again. I have just set a completely arbitrary, and probably too early date for the release of the first english Issue of "PET: Python Entre Todos" magazine.

The english version is called PET, which means "Python Entre Todos: English Translation".

It will have the same contents as the first spanish Issue and... it will be the last Issue done like this.

From now on, both versions will be published at the same time, if we can.

So, there will be a very short gap between the english first Issue and the second one (less than a month, we hope).

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "stay tuned".

So, [stay tuned..](http://revista.python.org.ar/) stay tuned: <http://revista.python.org.ar/>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "goodreads".
```

I am a big fan of [GoodReads..](http://goodreads.com) goodreads: <http://goodreads.com>

a social network for people who read books.

I read a lot, and I like that I can see what other people think before starting a book, and I can put my short reviews, and I can see what I have been reading, and lots more.

In fact, goodreads is going to be a big part of a project I am starting with some PyAr guys.

One thing I have been lazy about is adding my book list to goodreads, because it's a bit of a chore.

Well, chore no more!

Here's how to do it, the hacker way...

1.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "zbar".
```

Get [zbar..](http://zbar.sourceforge.net/) zbar: <http://zbar.sourceforge.net/>

2. Get a cheap webcam
3. Get a book
4. Get a 7-line python program (included below)

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "watch the video...".
```

Now [watch the video.....](http://blip.tv/file/4071008) watch the video...: <http://blip.tv/file/4071008>

Cute, isn't it?

Here's the code:

```
\ :kn: `import` \ \ :nn: `os` \
\n: `p` \ \ :o: `=` \ \ :n: `os` \ \ :o: `.` \ \ :n: `popen` \ \ :p: `( ` \ \ :s: ` /us
\n: `k: `while` \ \ :bp: `True` \ \ :p: `:` \ \
 \ :n: `code` \ \ :o: `=` \ \ :n: `p` \ \ :o: `.` \ \ :n: `readline` \ \ :p: `( )
 \ :k: `print` \ \ :s: `Got barcode:` \ \ :p: `,` \ \ :n: `code` \
 \ :n: `isbn` \ \ :o: `=` \ \ :n: `code` \ \ :o: `.` \ \ :n: `split` \ \ :p: `( ` `
 \ :n: `os` \ \ :o: `.` \ \ :n: `system` \ \ :p: `( ` ` \ :s: ` chromium http://w
```

Warning: rant ahead.

Yesterday the government of Argentina announced that they are giving way 3 million netbooks to students. They also announced that they are giving them the option of Ubuntu or Windows 7.

There was, of course, the typical reaction from the FLOSS side: why are they giving Windows to the students when Linux is better? It's unfair that the government pays for Windows!

I am here to tell you to grow up and stop being a baby. I am here to tell you to **stop** treating others like babies.

I think I can do this because I am immune to criticism from the FLOSS crowd: I am a member of that crowd. I have an awesome FLOSS pedigree, I have used nothing but Linux for over 15 years. And I have a thick skin and I don't care much what other people say, in principle, unless they give me good reasons to care. And I am telling you to stop complaining.

I am telling you that if the only reason to use a specific piece of software is because it's cheaper, you are accepting that piece of software sucks.

I want people not to just use Linux, I want them to **want** to use Linux. I want them to wait anxiously for the next release of Ubuntu or Firefox or whatever.

And the first step towards excellence is wanting to be excellent. If having to pay nothing for Windows or Ubuntu there is a certainty that Windows will win, then Ubuntu freaking **sucks** and needs to improve. People are not adopting it even if it's free? Then something **is** wrong, and figuring out what is important.

But even more important than finding the missing piece is knowing a piece is missing. Open source has grown complacent. It's grown self righteous. It's become adolescent, sure of its awesomeness and immortality.

I don't believe in many things, but I do believe in free will. I believe that people are **not** morons, I believe that if they prefer Windows, it's because it does something better, and I believe that whatever that is (and I don't really know what it may be), it can be found, and can be improved, and can be replaced, and other things can be added, and people will **want** to use the better product.

And if they don't... well, at least we fought an honest fight, and we did our best, and we (hopefully) had fun in the process, and pushed the envelope, and created nice things, and the users are better off in the end even if our babies are not the chosen ones, because we raised the level of everything.

For example, before Linux, Windows sucked much, much more than it does now, and I think many of those improvements were **because** of Linux, and I am happy that today Windows users have a OS that doesn't stink.

I want free and open source software to be used because it's awesome, not because it's cheap. Awesome **and** cheap I can live with. Just cheap? That sucks.

And the constant "they use windows because they don't know better"? That's patronizing and condescending, and very, very annoying. And if it annoys me, who is not the target of the lame condescension, trust me, it annoys the crap out of Windows users.

Grow a spine, get your asses into gear, start making awesome stuff, kick ass with quality. That's why we are here. Not to be the cheapest date in town.

Here's what I'm thinking: how hard could it be to make PyQt app deployment absolutely easy? Well, I am guessing: not very hard.

Here's the trick: see what works in the real world, and adopt it.

Question: what has deployed billions of apps and has its users happy? Answer: phones app stores.

Question: how do they work? Answer: well, that's not **that** short, so let's start explaining.

As I see it, a reasonable app store has the following components:

A Stable Deployment Target

You can't deploy from the store if you don't know what you are deploying into. If the target platform is shaky, you just can't know how to deploy without user assistance, and we are trying to make this **easy** for the user, which means that's not acceptable.

So, what's a stable deployment target we can provide?

- PyQt (so we can deploy GUIs to all major desktop platforms)
- Python standard library
- Selected modules

What can be (and should be) bundled with the app?

- Pure python modules
- Artwork and other resources

What may be bundled:

- Python modules written in C/C++, but you then have to redo the app for each platform, and that kinda sucks.

Deployment Services

- Apps should be able to check if there is a new version of them in the store, to ask for upgrades.
- Apps should be added by the deployment platform nicely into the host system's menus, desktop, etc.

Monetization Services

- Some way to charge for apps. Even for open source apps, you could ask for U\$S0.99 if you install them through the store. Optional, of course, and up to the app owner.
- Ad platform? There must be a good one for desktop apps somewhere?

The Store Itself

- A website that downloads a "package" associated with a local deployment application.
- A app store app. Install things not via web, but via a desktop application.

I don't expect a functional version of this would take me more than a week working fulltime to implement. Of course then there are all sorts of usability, looks, etc. things to consider.

And... I am going to do something I very rarely do. I am going to ask for money.

As an experiment, I have setup a project at <http://www.indiegogo.com/Qt-Shop> and set a funding goal of US\$ 600.

There you can fund me. I promise that if the project is totally funded, I **will** deliver. If it isn't, I may deliver anyway. I would prefer to have the money though.

The platform would be released under GPLv2 or later.

Hell yeah! It has been a lot of work but it's out at <http://revista.python.org.ar>

Some articles:

- PyAr, The History
- from gc import commonsense - Finish Him!
- Painless Concurrency: The multiprocessing Module
- Introduction to Unit Testing with Python
- Taint Mode in Python
- Applied Dynamism
- Decorating code (Part 1)
- Web2Py for Everybody

It's available in pretty much every format anyone can read, and if your favourite is not there, we will make it for you or may I be smote by the flying spaghetti monster's noodly appendage!

AFAIK there is no other Python magazine being published (feel free to correct me), so it's kind of a big thing for us in PyAr (the Argentina Python community) that we are doing one, and in two languages.

But why stop here? Want it to be available in your language? Contact us at revistapyar@netmanagers.com.ar it may be doable!

And of course, very soon there will be a call for articles for Issue 2, and trust me: that one's going to be **epic**: this one was just a warmup.

Sometimes, you see a piece of code and it just feels *right*. Here's an example I found when doing my "Import Antigravity" session for PyDay Buenos Aires: the `progressbar` module.

Here's an example that will teach you enough to use `progressbar` effectively:

```
\ :n: `progress` \ \ :o: `=` \ \ :n: `ProgressBar` \ \ :p: `()` \ \
\ :k: `for` \ \ :n: `i` \ \ :ow: `in` \ \ :n: `progress` \ \ :p: `(` \ \ :nb: `rang
\ :n: `time` \ \ :o: `.` \ \ :n: `sleep` \ \ :p: `(` \ \ :mf: `0.01` \ \ :p: `)` \ \
```

Yes, that's it, you will get a nice ASCII progress bar that goes across the terminal, supports resizing and moves as you iterate from 0 to 79.

The `progressbar` module even lets you do fancier things like ETA or file transfer speeds, all just as nicely.

Isn't that code just *right*? You want a progress bar for that loop? Wrap it and you have one! And of course since I am a PyQt programmer, how could I make PyQt have something as *right* as that?

Here's how the output looks like:

You can do this with every toolkit, and you probably should!. It has one extra feature: you can interrupt the iteration. Here's the (short) code:

```

\ :c: '# -*- coding: utf-8 -*-'
\ :kn: 'import' \ \ :nn: 'sys' \ \ :o: ', ' \ \ :nn: 'time'
\ :kn: 'from' \ \ :nn: 'PyQt4' \ \ :kn: 'import' \ \ :n: 'QtCore' \ \ :p: ', '

\ :k: 'def' \ \ :nf: 'progress' \ \ :p: '(' \ \ :n: 'data' \ \ :p: ', ' \ \ :o: '* '
\ \ :n: 'it' \ \ :o: '=' \ \ :nb: 'iter' \ \ :p: '(' \ \ :n: 'data' \ \ :p: ')' \
\ \ :n: 'widget' \ \ :o: '=' \ \ :n: 'QtGui' \ \ :o: '.' \ \ :n: 'QProgressDialog'
\ \ :n: 'c' \ \ :o: '=' \ \ :mi: '0'
\ \ :k: 'for' \ \ :n: 'v' \ \ :ow: 'in' \ \ :n: 'it' \ \ :p: ':' \
\ \ :n: 'QtCore' \ \ :o: '.' \ \ :n: 'QCoreApplication' \ \ :o: '.' \ \ :n: 'QApplication'
\ \ :k: 'if' \ \ :n: 'widget' \ \ :o: '.' \ \ :n: 'wasCanceled' \ \ :p: '():'
\ \ \ :k: 'raise' \ \ :ne: 'StopIteration'
\ \ :n: 'c' \ \ :o: '+=' \ \ :mi: '1'
\ \ :n: 'widget' \ \ :o: '.' \ \ :n: 'setValue' \ \ :p: '(' \ \ :n: 'c' \ \ :p: ')'
\ \ :k: 'yield' \ \ :p: '(' \ \ :n: 'v' \ \ :p: ')'

\ :k: 'if' \ \ :n: '__name__' \ \ :o: '==' \ \ :s: '"__main__"' \ \ :p: ':' \
\ \ :n: 'app' \ \ :o: '=' \ \ :n: 'QtGui' \ \ :o: '.' \ \ :n: 'QApplication' \ \ :p: '()'

\ :c: '# Do something slow'
\ :k: 'for' \ \ :n: 'x' \ \ :ow: 'in' \ \ :n: 'progress' \ \ :p: '(' \ \ :nb: 'iter'
\ \ :n: 'time' \ \ :o: '.' \ \ :n: 'sleep' \ \ :p: '(' \ \ :o: '.' \ \ :mi: '2'

```

Have fun!

Yes, it's yet another program I am working on. But hey, the last few I started are actually pretty functional already!

And... I am not doing this one alone, which should make it more fun.

It's an eBook (or just any book?) manager, that helps you keep your PDF/Mobi/FB2/whatever organized, and should eventually sync them to the device you want to use to read them.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "the video".

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "the video".

What works now? See [the video](#).. the video: <http://blip.tv/file/4135688>

!

In case that makes no sense to you:

- You can get books from FeedBooks. Those books will get downloaded, added to your database, tagged, the cover fetched, etc. etc.
- You can import your current folder of books **in bulk**.
Aranduka will use google and other sources to try to guess (from the filename) what book that is and fill in the extra data about it.

- You can "guess" the extra data.

By marking certain data (say, the title) as reliable, Aranduka will try to find some possible books that match then you can choose if it's right.

Of course you can also edit that data manually.

And that's about it. Planned features:

- Way too many to list.

The goals are clear:

- It should be beautiful (I know it isn't!)
- It should be powerful (not yet!)
- It should be better than the "competition"

If those three goals are not achieved, it's failure. It may be a fun failure, but it would still be a failure.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "aranduka".

I have been putting lots of love into Aranduka.. aranduka: <http://aranduka.googlecode.com>

an eBook manager, (which is looking very good lately, thanks!), and I didn't want it to also be an eBook reader.

But then I thought... how hard can it be to read ePub? Well, it's freaking easy!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "a good start at stackoverflow.com".

Here's a [good start at stackoverflow.com](http://stackoverflow.com/questions/13).. a good start at stackoverflow.com: <http://stackoverflow.com/questions/13>

but the short of it is... it's a zip with some XML in it.

One of those XML files tells you where things are, one of them is the TOC, the rest is just a small static collection of HTML/CSS/images.

So, here are the ingredients to roll-your-own ePub reader widget in 150 LOC:

- Use python's zipfile library to avoid exploding the zip (that's lame)
- Use Element Tree to parse said XML files.
- Use PyQt's QtWebKit to display said collection of XML/CSS/Images

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: "this recipe".

Use [this recipe](http://www.diotavelli.net/PyQtWiki/Using%20a%20Custom%20Protocol%20with%20).. this recipe: [http://www.diotavelli.net/PyQtWiki/Using%20a%20Custom%20Protocol%20with%](http://www.diotavelli.net/PyQtWiki/Using%20a%20Custom%20Protocol%20with%20)

to make QtWebKit tell you when it wants something from the zipfile.

Plug some things to others, shake vigorously, and you end up with this:

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "the code (as of today)".
```

Here's [the code \(as of today\)](http://code.google.com/p/aranduka/source/browse/src/epubviewer).. the code (as of today): <http://code.google.com/p/aranduka/source/browse/src/epubviewer>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "the ui file you need".
```

and [the UI file you need](http://code.google.com/p/aranduka/source/browse/src/epubviewer).. the ui file you need: <http://code.google.com/p/aranduka/source/browse/src/epubviewer>

Missing stuff:

- It doesn't display the cover.
- It only shows the top level of the table of contents.
- I only tested it on two books ;-)
- It sure can use a lot of refactoring!

Neither should be terribly hard to do.

I recently recovered a Palm m100, because my calendar was getting out of hand.

So, besides using it as a glorified alarm clock, what could I do with it?

Oh, sure, some lame games exist (remember this is an old model, black and white, almost no sound, low resolution, very little memory), but...

Well, I have had for a long time the habit of reading project Gutenberg's books. I even wrote a program to read them on my PC (Uqbar, link at the left).

But the main problem is, of course, that the places where I do most of my reading (bars, clients' offices while waiting for stuff to install, trains, waiting rooms) usually lack computers I can use.

So, I decided to try using the Palm for that.

Of course, I am not being original, since a bazillion people had the same idea.

Therefore, I went and got myself some of the software people recommended for reading PalmDoc documents, and some stuff to create them.

I tried: Iambic Reader, CSpotRun, and a couple of others, and had a nasty surprise, because all of them, when I jumped to a distant place in the book, did one of the following:

- Crash
- Corrupt the text

So, I gave up hope on PalmDoc (although it's nice that KWord can export to it!), and looked harder.

Then I found Weasel Reader. Not only does it work great (it only lacks an option to add a margin, sometimes lighting makes the edge of the screen hard to read), but the format it uses compresses texts more than palmDoc!

So, how's this going:

- The screen is small, but that's not too bad, it is about the same as reading a newspaper column.
- Smooth scrolling doesn't work. Maybe my palm is too old/slow/bad, but it doesn't work well.
- Page-rolling autoscroll does work. The top of the screen is automatically replaced with the next page line-at-a-time.
- Manual page-scrolling works best. Since you only have to tap the screen or press a button it's not annoying at all.
- It doesn't hurt my eyes. Since it's not even backlit, it requires good ambient light, but it's not bad at all.
- It has decent capacity. I can stick up to 3MB of text, along with all the programs I use, in a 2MB device. 3MB of text is a lot of text.

So, I am pretty happy with the result.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "aranduka".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "aranduka".

One of my favourite things about [Aranduka.. aranduka: http://aranduka.googlecode.com](http://aranduka.googlecode.com)

as a project is that it's an endless source of small, limited side projects.

For example, Aranduka is now close to being able to sync my book collection to my phone. But... what if what I want to read on the train is not a book but, say, a blog?

Well, blogs provide their content via a feed. And a feed is a collection of HTML pieces glued into a structure plus some data like author and such.

And there's a great module for parsing them, called `feedparser`. And I have written not one, not two, not three, but *four* RSS aggregators in the past.

So, how about converting the feed into something my phone can handle? [#] Would it be hard to do?

Well... not really hard. It's mostly a matter of taking a small, sample ePub document (created by Calibre) writing a few templates, feeding it the data from `feedparser` and zipping it up.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "as an epub".

For example, this is this blog, [as an ePub.. as an epub: /static/lateral.epub](#)

and here's FBReader reading it:

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "in aranduka's mercurial".

As usual, the code is open, and it's here in [aranduka's mercurial](#).. in aranduka's mercurial: <http://code.google.com/p/aranduka/source/browse/?r=cfccd117ff95bf3098e9d24866ded46ad3ff7705#hg/src>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "templete".

It's not really interesting code, and requires [templete](#).. [templete: http://code.google.com/p/aranduka/source/browse/s](http://code.google.com/p/aranduka/source/browse/s)

feedparser and who knows what else.

The produced ePub doesn't validate, and it probably never will, because it has chunks of the original feed in it, so standard compliance doesn't depend on rss2epub.

Also, you get no images. That would imply parsing and fixing all img elements, I suppose, and I am not going to do it right now.

[#] I first saw this feature in plucker a long time ago, and I know Calibre has it too.

I am one of the speakers in the cycle of open python lectures at La Tribu in Buenos Aires, and here is the one where I speak about virtualenv, buildout, nose, hudson and other things.

Long video!

It would be natural to anyone who doesn't know me to believe I live a life of luxury, international travel and exotic pleasures[1]_ but there is a small thing I am kinda ashamed of:

I hardly ever got paid to code.

Really! Most of the money I live on has absolutely nothing to do with whatever you read about on my blog.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "net managers".

I make my living doing things like installing Asterisk in call centers, or configuring MySQL replication, or configuring VPNs using assorted piece-of-crap routers and by all means if you need that kind of work done, please contact [Net Managers](#).. net managers: <http://www.netmanagers.com.ar>

we are freaky good at it and charge reasonable rates.

But while I like that kind of thing[2]_ I like other things better. I like programming much more than I like configuring Asterisk!

Then again, I am not a really great programmer. I am just about an average coder, except that I am faster than most.

And lately, an interesting phenomenon has taken place.

How much money I earned thanks to my first 14 years of open source and free software development? \$0[3]_

How much money have I earned in the last year of FLOSS development? \$500 (and it is bound to be more than that soon).

To a first-worlder that may seem like peanuts, but down here in the land of cheap beef, that's... about 100 pounds of good meat! Or 10 nice dinners in fine restaurants for me and my wife.[\[4\]](#)

I am wondering, is this part of a trend? Have others been receiving more appreciation in the form of money lately?

In any case, it's a great feeling when a work of love (and trust me, noone does something for 14 years for free if he doesn't love it) is appreciated.

Just in case: this doesn't mean you have to pay me to get a bug fixed, or to get a feature implemented. However, it does mean that if you like or use any of my programs and want me to feel good about the time I spent writing them... well, you **can** just send me a nice email, I will be just as happy. Really!

That a plugin architecture for a complex app is a good idea is one of those things that most people kinda agree on. One thing we don't quite agree is how the heck are we going to make out app modular?

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "yapsy".
```

One way to do it (if you are coding python) is using [Yapsy](http://yapsy.sourceforge.net/).. yapsy: <http://yapsy.sourceforge.net/>

1

```
system-message
WARNING/2 in everything.txt, line 28392
Duplicate explicit target name: "1".
```

I do!

2

```
system-message
WARNING/2 in everything.txt, line 28394
Duplicate explicit target name: "2".
```

And I do like it! It's like a grownup's version of Lego.

3

```
system-message
WARNING/2 in everything.txt, line 28396
Duplicate explicit target name: "3".
```

I did get boxed SuSE CD sets for a while but I actually lost money on them since I had to pay taxes.

4

```
system-message
WARNING/2 in everything.txt, line 28398
Duplicate explicit target name: "4".
```

To all those who have contributed I send my sincerest appreciation, and trust me that all this money has been wisely invested on having fun, cheap chinese gadgetry and chocolate mousse.

Yapsy is awesome. Also, yapsy is a bit underdocumented. Let's see if this post fixes that a bit and leaves just the awesome.

Update: I had not seen the new Yapsy docs, released a few days ago. They are much better than what was there before :-)

Here's the general idea behind yapsy:

- You create a Plugin Manager that can find and load plugins from a list of places (for example, from ["/usr/share/appname/plugins", "~/.appname/plugins"]).
- A plugin category is a class.
- There is a mapping between category names and category classes.
- A plugin is a module and a metadata file. The module defines a class that inherits from a category class, and belongs to that category.

The metadata file has stuff like the plugin's name, description, URL, version, etc.

One of the great things about Yapsy is that it doesn't *specify too much*. A plugin will be just a python object, you can put whatever you want there, or you can narrow it down by specifying the category class.

In fact, the way I have been doing the category classes is:

- Start with an empty class
- Implement two plugins of that category
- If there is a chunk that's much alike in both, move it into the category class.

But trust me, this will all be clearer with an example :-)

I will be doing it with a graphical PyQt app, but Yapsy works just as well for headless of CLI apps.

Let's start with a simple app: an HTML editor with a preview widget.

Figure 10: A simple editor with preview

Here's the code for the app, which is really simple (it doesn't save or do anything, really, it's just an example):

editor1.py

```
\ :kn: `from` \ :nn: `PyQt4` \ :kn: `import` \ \ :n: `QtCore` \ \ :p: `,\`  
\ :kn: `import` \ \ :nn: `os` \ \ :o: `,\` \ \ :nn: `sys` \
```

```

\ :k:'class' \ :nc:'Main' \ :p:'(\ \ :n:'QtGui' \ :o:'. '\ \ :n:'QW
\ :k:'def' \ :nf:'__init__' \ :p:'(\ \ :bp:'self' \ :p:'):\
\ :n:'QtGui' \ :o:'. '\ \ :n:'QWidget' \ :o:'. '\ \ :n:'__init__
\ :bp:'self' \ :o:'. '\ \ :n:'layout' \ :o:'='\ \ :n:'QtGui'
\ :bp:'self' \ :o:'. '\ \ :n:'editor' \ :o:'='\ \ :n:'QtGui'
\ :bp:'self' \ :o:'. '\ \ :n:'preview' \ :o:'='\ \ :n:'QtWeb
\ :bp:'self' \ :o:'. '\ \ :n:'layout' \ :o:'. '\ \ :n:'addWidget
\ :bp:'self' \ :o:'. '\ \ :n:'layout' \ :o:'. '\ \ :n:'addWidget
\ :bp:'self' \ :o:'. '\ \ :n:'editor' \ :o:'. '\ \ :n:'textChan
\ :bp:'self' \ :o:'. '\ \ :n:'setLayout' \ :p:'(\ \ :bp:'self

\ :k:'def' \ :nf:'updatePreview' \ :p:'(\ \ :bp:'self' \ :p:')
\ :bp:'self' \ :o:'. '\ \ :n:'preview' \ :o:'. '\ \ :n:'setHtml

\ :k:'def' \ :nf:'main' \ :p:'():'\
\ :c:'# Again, this is boilerplate, it's going to be the same on'\
\ :c:'# almost every app you write'\
\ :n:'app' \ :o:'='\ \ :n:'QtGui' \ :o:'. '\ \ :n:'QApplication'
\ :n:'window' \ :o:'='\ \ :n:'Main' \ :p:'() '\
\ :n:'window' \ :o:'. '\ \ :n:'show' \ :p:'() '\
\ :c:'# It's exec_ because exec is a reserved word in Python'\
\ :n:'sys' \ :o:'. '\ \ :n:'exit' \ :p:'(\ \ :n:'app' \ :o:'. '\

\ :k:'if' \ :n:'__name__' \ :o:'==' \ :s:'"__main__"' \ :p:':\
\ :n:'main' \ :p:'() '\

```

Note

From now on listings will not include the main function, because it never changes.

But this application has an obvious limit: you have to type HTML in it. Why not type python code in it and have it convert to HTML for display? Or Wiki markup, or restructured text?

You could, in principle, just implement all those modes, but then you are assuming the responsibility of supporting every thing-that-can-be-turned-into-HTML. Your app would be a monolith. That's where yapsy enters the scene.

So, let's create a plugin category, called "Formatter" which takes plain text and returns HTML. Then we add stuff in the UI so the user can choose what formatter he wants, and implement two of those.

Here's our plugin category class:

categories.py

```

\ :k:'class' \ :nc:'Formatter' \ :p:'(\ \ :nb:'object' \ :p:'):\
\ :sd:'""Plugins of this class convert plain text to HTML""'\
\ :n:'name' \ :o:'='\ \ :s:'"No Format"'
\ :k:'def' \ :nf:'formatText' \ :p:'(\ \ :bp:'self' \ :p:', '\

```

```

    \ :sd:``""Takes plain text, returns HTML""`\
    \ :k:'return'\ \ :n:'text'\

```

Of course what good is a plugin architecture without any plugins for it? So, let's create two plugins.

First: a plugin that takes python code and returns HTML, thanks to pygments.

plugins/pygmentize.py

```

\ :kn:'from'\ \ :nn:'pygments'\ \ :kn:'import'\ \ :n:'highlight'\
\ :kn:'from'\ \ :nn:'pygments.lexers'\ \ :kn:'import'\ \ :n:'PythonL
\ :kn:'from'\ \ :nn:'pygments.formatters'\ \ :kn:'import'\ \ :n:'Htm

\ :kn:'from'\ \ :nn:'categories'\ \ :kn:'import'\ \ :n:'Formatter'\

\ :k:'class'\ \ :nc:'Pygmentizer'\ \ :p:'(\ \ :n:'Formatter'\ \ :p:')\
  \ :n:'name'\ \ :o:'='\ \ :s:'"Python Code"'\

  \ :k:'def'\ \ :nf:'formatText'\ \ :p:'(\ \ :bp:'self'\ \ :p:', '\
    \ :k:'return'\ \ :n:'highlight'\ \ :p:'(\ \ :n:'text'\ \ :p:')\

```

See how it goes into a plugins folder? Later on we will tell yapsy to search there for plugins.

To be recognized as a plugin, it needs a metadata file, too:

plugins/pygmentize.yapsy-plugin

```

\ :k:'[Core]'\
\ :na:'Name'\ \ :o:'='\ \ :s:'Python Code'\
\ :na:'Module'\ \ :o:'='\ \ :s:'pygmentize'\

\ :k:'[Documentation]'\
\ :na:'Author'\ \ :o:'='\ \ :s:'Roberto Alsina'\
\ :na:'Version'\ \ :o:'='\ \ :s:'0.1'\
\ :na:'Website'\ \ :o:'='\ \ :s:'http://lateral.netmanagers.com.ar'\
\ :na:'Description'\ \ :o:'='\ \ :s:'Highlights Python Code'\

```

And really, that's **all** there is to making a plugin. Here's another one for comparison, which uses docutils to format reStructured Text:

plugins/rest.py

```

\ :kn:'from'\ \ :nn:'categories'\ \ :kn:'import'\ \ :n:'Formatter'\
\ :kn:'import'\ \ :nn:'docutils.core'\
\ :kn:'import'\ \ :nn:'docutils.io'\

\ :k:'class'\ \ :nc:'Rest'\ \ :p:'(\ \ :n:'Formatter'\ \ :p:'):\
  \ :n:'name'\ \ :o:'='\ \ :s:'"Restructured Text"'\

```

```

\ :k:'def'\ \ :nf:'formatText'\ \ :p:'('\ \ :bp:'self'\ \ :p:','\
\ :n:'output'\ \ :o:'='\ \ \ :n:'docutils'\ \ :o:'.'\ \ :n:'cor
\ :n:'text'\ \ :p:','\ \ \ :n:'writer_name'\ \ :o:'='\ \ \ :
\ :p:')'\
\ :k:'return'\ \ \ :n:'output'\

```

plugins/rest.yapsy-plugin

```

\ :k:'[Core]'\
\ :na:'Name'\ \ \ :o:'='\ \ \ :s:'Restructured Text'\
\ :na:'Module'\ \ \ :o:'='\ \ \ :s:'rest'\

\ :k:'[Documentation]'\
\ :na:'Author'\ \ \ :o:'='\ \ \ :s:'Roberto Alsina'\
\ :na:'Version'\ \ \ :o:'='\ \ \ :s:'0.1'\
\ :na:'Website'\ \ \ :o:'='\ \ \ :s:'http://lateral.netmanagers.com.ar'\
\ :na:'Description'\ \ \ :o:'='\ \ \ :s:'Formats restructured text'\

```

And here they are in action:

Figure 11: reSt mode

Figure 12: Python mode

Of course using categories you can do things like a "Tools" category, where the plugins get added to a Tools menu, too.

And here's the application code:

editor2.py

```

\ :kn:'from'\ \ \ :nn:'categories'\ \ \ :kn:'import'\ \ \ :n:'Formatter'\
\ :kn:'from'\ \ \ :nn:'yapsy.PluginManager'\ \ \ :kn:'import'\ \ \ :n:'Plu

```

```

\ :k:'class'\ \ :nc:'Main'\ \ :p:'('\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'QW
\ :k:'def'\ \ :nf:'__init__'\ \ :p:'('\ \ :bp:'self'\ \ :p:'):''\
\ :n:'QtGui'\ \ :o:'. '\ \ :n:'QWidget'\ \ :o:'. '\ \ :n:'__init__
\ :bp:'self'\ \ :o:'. '\ \ :n:'layout'\ \ :o:'='\ \ :n:'QtGui'
\ :bp:'self'\ \ :o:'. '\ \ :n:'formattersCombo'\ \ :o:'='\ \ :
\ :bp:'self'\ \ :o:'. '\ \ :n:'editor'\ \ :o:'='\ \ :n:'QtGui'
\ :bp:'self'\ \ :o:'. '\ \ :n:'preview'\ \ :o:'='\ \ :n:'QtWeb

\ :bp:'self'\ \ :o:'. '\ \ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget
\ :bp:'self'\ \ :o:'. '\ \ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget
\ :bp:'self'\ \ :o:'. '\ \ :n:'layout'\ \ :o:'. '\ \ :n:'addWidget

\ :bp:'self'\ \ :o:'. '\ \ :n:'editor'\ \ :o:'. '\ \ :n:'textChan
\ :bp:'self'\ \ :o:'. '\ \ :n:'setLayout'\ \ :p:'('\ \ :bp:'self

\ :c:'# Create plugin manager'\
\ :bp:'self'\ \ :o:'. '\ \ :n:'manager'\ \ :o:'='\ \ :n:'Plugi
\ :bp:'self'\ \ :o:'. '\ \ :n:'manager'\ \ :o:'. '\ \ :n:'setPlug

\ :c:'# Load plugins'\
\ :bp:'self'\ \ :o:'. '\ \ :n:'manager'\ \ :o:'. '\ \ :n:'locateP
\ :bp:'self'\ \ :o:'. '\ \ :n:'manager'\ \ :o:'. '\ \ :n:'loadPlu

\ :c:'# A do-nothing formatter by default'\
\ :bp:'self'\ \ :o:'. '\ \ :n:'formattersCombo'\ \ :o:'. '\ \ :n:
\ :bp:'self'\ \ :o:'. '\ \ :n:'formatters'\ \ :o:'='\ \ :p:'{'
\ :k:'print'\ \ :bp:'self'\ \ :o:'. '\ \ :n:'manager'\ \ :o:'. '\
\ :k:'for'\ \ :n:'plugin'\ \ :ow:'in'\ \ :bp:'self'\ \ :o:'. '\
\ :k:'print'\ \ :s:'"XXX"'\
\ :c:'# plugin.plugin_object is an instance of the plugin'\
\ :bp:'self'\ \ :o:'. '\ \ :n:'formattersCombo'\ \ :o:'. '\ \
\ :bp:'self'\ \ :o:'. '\ \ :n:'formatters'\ \ :p:'['\ \ :n:'

\ :k:'def'\ \ :nf:'updatePreview'\ \ :p:'('\ \ :bp:'self'\ \ :p:')
\ :c:'# Check what the current formatter is'\
\ :n:'name'\ \ :o:'='\ \ :nb:'unicode'\ \ :p:'('\ \ :bp:'sel
\ :n:'text'\ \ :o:'='\ \ :nb:'unicode'\ \ :p:'('\ \ :bp:'self
\ :k:'if'\ \ :n:'name'\ \ :ow:'in'\ \ :bp:'self'\ \ :o:'. '\
\ :n:'text'\ \ :o:'='\ \ :bp:'self'\ \ :o:'. '\ \ :n:'form
\ :bp:'self'\ \ :o:'. '\ \ :n:'preview'\ \ :o:'. '\ \ :n:'setHtml

```

In short: this is easy to do, and it leads to fixing your application's internal structure, so it helps you write *better* code.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "full source code for everything".

Full source code for everything.. full source code for everything: <http://lateral.netmanagers.com.ar/static/yapsy/yap>

This is a spanish-only post, it makes no sense for any english-readers: it's about how to buy electronics in Argentina.

Intro

Ser un nerd en Argentina tiene muchos problemas. Uno de ellos es que los precios que nos quieren cobrar por casi cualquier chiche electrónico es ridículo.

Otro de los problemas es que el 99% de la gente no sabe como hacer para que no les afanen de esa manera. Y ojo, no hablo de contrabandear, hablo de usar tus derechos y tus recursos *legales*. Aclaro: no soy abogado, si siguiendo estos consejos vas preso, es cosa tuya, yo digo lo que *creo* que está bien nomás.

Así que estimados lectores, acá está mi guía para comprar chiches sin tener que empeñar un riñón (que también es ilegal, ya que estamos).

Lo primero que necesitás es paciencia. Algunos de los mecanismos que te voy a contar llevan uno o dos meses, o requieren colaboración de terceros. Pero bueno, somos nerds, resolver problemas es lo nuestro.

Chucherías Chinas

Tengo un bonito microscopio de bolsillo de 80x que me compró mi esposa. Costo? \$90. Es un juguete buenísimo, re educativo, y cuando tenía 6 años hubiera dado la figurita difícil (era la tarántula) por uno de estos.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "u\$s 9.39, con envío gratis".

Costo en china? [U\\$S 9.39, con envío gratis.](http://www.dealextreme.com/details.dx/sku) u\$s 9.39, con envío gratis: <http://www.dealextreme.com/details.dx/sku>

O sea, la mitad. Y como sale menos de U\$S 25, de hecho es legal traerlo por correo y no pagás tasa de aduana.

De todas formas, probablemente no tengas que ir a la aduana porque....

Tip

Cosas chicas == no problem

Normalmente los paquetes de menos de medio kilo casi nunca se quedan en la aduana. Ese dongle bluetooth de U\$S 1 te va a llegar a tu casa.

Pero ponéle que pagás la tasa. Con tasa y todo, son U\$S 15, o sea \$60. Si pasás por la aduana a veces y no te molesta hacer cola un rato (llevate un libro)...

Pero es aún mejor. Porque en <http://DealExtreme.com> tenés un modelo de 150x por U\$S 12! De hecho, tenés **36 modelos de microscopios!**

Tip

El mundo es grande

Comprar las cosas directamente significa elegir de una variedad muchísimo mayor de la que tenés acá. Sumáله que encima es más barato...

Por ejemplo, en <http://alibaba.com> hay algo así como 35000 vendedores de tablets con Android. Alguna te va a servir ;-)

Ahora bien, supongamos que te decidiste y te encargaste algo de China o similar, qué tenés que saber:

- Guardá la "factura" que te dá el sitio, y la URL del producto.
La factura suele ser un HTML, imprimílo. Cuando retirás en Aduana lo necesitás. No lo muestres de entrada! Sólo si te lo piden.
- Andá dispuesto a pagar. Es al cuete discutir. Tené en cuenta que aún con lo de la aduana es barato, y no te hagas mala sangre.
- Si vas tipo 10:30 suele haber poca gente.
- Andá tranquilo, llevate algo para entretenerte porque en una de esas lleva un rato. Igual hay sillas, que se yó, no es la muerte de nadie.

Tip

It's a gift!

Casi todos estos sitios tienen algún servicio de "gift". Te sacan el precio y todo lo que haga que parezca una compra por internet. Aumenta la probabilidad de que llegue a tu casa directo.

Cómo Pagar

La más fácil es PayPal. Si tenés tarjeta de crédito es trivial.

Tip

Hablá con nerds amigos!

Si no tenés PayPal ni tarjeta internacional, pero conocés alguien que trabaja para el exterior, es muy probable que esa persona tenga saldo en PayPal que no puede sacar. En Argentina, convertir PayPal en dinero es muy caro. Entonces se puede llegar a un acuerdo: se lo pagás al 95% del valor (o lo que arreglen) tu amigo tiene la plata, vos tenés descuento en dealextrême o donde sea.

También podés hacer la compra con la tarjeta de un amigo/pariente y pagarle a él en efectivo.

El amigo que viene de afuera

Sí tenés un conocido que viaja al exterior con regularidad:

- Podés comprar en USA y que te reciba y traiga las cosas.
- Hasta U\$S 300 (si el que viaja es argentino) no se paga aduana. Se paga el 50% por encima de esa cifra.
- Si es un extranjero que trae una cosa propia "en uso" y te la vende acá, no creo que tengas que pagar importación (no estoy seguro)

Veamos un ejemplo!

- Una HP mini comprada acá: \$1999, o U\$S 500
- Comprada en USA: U\$S 329.
 - Pagando aduana con franquicia: U\$S 345
 - Pagando aduana full: U\$S 495 (ahí no conviene ;-)

Tip

Los productos HP tienen garantía internacional
 Los Dell por ejemplo: **NO**. Así que ojo, comprá una marca que te dé service en Argentina. En el caso de Dell tenés que comprar aparte una garantía que te sirva acá. Vos sabrás si te bancás tener una notebook sin garantía (yo sí, si es barata ;-)

Pero que te parecería tener la **misma** netbook por U\$S 180?

Bueno, resulta que HP (y todas las empresas) venden productos "refurbished" o sea reacondicionados. Normalmente no te vas a dar cuenta de nada, tienen un rayón mínimo, o algo así. Hoy la misma netbook estaba a U\$S 179.90. Normalmente salen más o menos la mitad. Y vienen con garantía.

Y fijáte que la HP mini por U\$S 180, pagando la aduana full... cuesta el 54% de lo que sale la nueva acá. Eso **es negocio**.

Tip

La entrega es allá
 Normalmente las compras de refurbished hay que hacerlas en USA. Por eso lo de que tiene que ir alguien a recibirla.
 Alternativamente hay servicios de reenvío de mercadería, pero no los he usado y no los conozco.

Alternativamente, Ciudad del Este (Paraguay) está muy cerca de Argentina. Y es muy barato comprar electrónicos ahí. Y tenés los U\$S 300 de franquicia.

Pero hay más! Un ciudadano paraguayo que visita argentina puede traer ciertas cosas con ciertos límites. Hay que hablar con él y coordinar!

Si no conocés a nadie... bueno, andá de paseo a cataratas! Pasala bien! Y comprate una cámara o lo que sea en la excursión a Ciudad del Este.

Accesorios

No compres accesorios originales. Por ejemplo: la batería original de repuesto de mi eeePC 701? U\$S 110, 3 elementos. La "trucha" de 6 elementos? U\$S 40.

Tip

Es ilegal importar cosas que se enchufan
 Todo lo que se enchufe a la pared tiene que homologarse y creo que es ilegal traerlo. Si alguien tiene un cargador para su propia notebook por ejemplo, supongo que es legal (así me hice de un cargador para la eee)

Conclusión

Ojalá sirva de algo, si tenés más tips ponelos en los comentarios!

I just saw this on a place I won't bother linking:

This october has 5 fridays, 5 saturdays and 5 sundays. This happens only every 823 years. And I am grounded all month!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "everywhere".

Really, it's not worth linking to because it's [everywhere](#).. everywhere: <http://www.google.com.ar/search?sourceid=c>
There are variations, of course, 5 sunday/monday/tuesdays, different months, etc.

But **why** are people falling for that? I mean, it's pretty obvious if you think 2 minutes that there's no way it can be true!

1. The length of october is fixed (31 days).
2. Since weeks are 7 days, you have 4 weeks + 3 days in october.
3. If october starts on a friday, those last three days are going to be friday/saturday/sunday.

So, for this thing to be true, October 1st would have to be a friday only once every 823 years. Sounds pretty stupid when you say it that way, right?

I really have no idea how anyone with half a brain can fall for it, but I have seen people I know have a functional cortex do it. Comments encouraged :-)

Finally, a new release of rst2pdf!

You can get it at its site: <http://rst2pdf.googlecode.com>

rst2pdf is a tool to convert restructured text (a light, cool markup language) to PDF using reportlab instead of LaTeX.

It has been used for many things, from books, to magazines, to brochures, to manuals, to websites and has lots of features:

- Font embedding (TTF or Type1 fonts)
- Cascading Stylesheets
- Extremely flexible plugin architecture (you can do things like render the headings from arbitrary SVG files!)
- Sphinx integration.
- Configurable page layouts
- Custom cover pages via templates
- And much, much more...

The biggest change in 0.16 is surely the improved support for Sphinx 1.0.x so if you are using Sphinx, you really want this version.

Also, it has a ton of bugfixes, and a few minor but useful new features.

Here's the whole changelog if you don't believe me:

- Fixed Issue 343: Plugged memory leak in the RSON parser.
- Fix for Issue 287: there is still a corner case if you have two sections with the same title, at the same level, in the same page, in different files where the links will break.
- Fixed Issue 367: german-localized dates are MM. DD. YYYY so when used in sphinx's template cover they appeared weird, like a list item. Fixed with a minor workaround in the template.
- Fixed Issue 366: links to "#" make no sense on a PDF file
- Made definitions from definition lists more stylable.
- Moved definition lists to SplitTables, so you can have very long definitions.
- Fixed Issue 318: Implemented Domain specific indexes for Sphinx 1.0.x
- Fixed Index links when using Sphinx/pdfbuilder.
- Fixed Issue 360: Set literal.wordWrap to None by default so it doesn't inherit wordWrap CJK when you use the otherwise correct japanese settings. In any case, literal blocks are not supposed to wrap at all.
- Switched pdfbuilder to use SplitTables by default (it made no sense not to do it)
- Fixed Issue 365: some TTF fonts don't validate but they work anyway.
- Set a valid default baseurl for Sphinx (makes it much faster!)
- New feature: --use-numbered-links to show section numbers in links to sections, like "See section 2.3 Termination"
- Added stylesheets for landscape paper sizes (i.e: a4-landscape.style)
- Fixed Issue 364: Some options not respected when passed in per-doc options in sphinx.
- Fixed Issue 361: multiple linebreaks in line blocks were collapsed.
- Fixed Issue 363: strange characters in some cases in math directive.
- Fixed Issue 362: Smarter auto-enclosing of equations in $...$
- Fixed Issue 358: --real--footnotes defaults to False, but help text indicates default is True
- Fixed Issue 359: Wrong --fit-background-mode help string
- Fixed Issue 356: missing cells if a cell spawns rows and columns.
- Fixed Issue 349: Work correctly with languages that are available in form aa_bb and not aa (example: zh_cn)
- Fixed Issue 345: give file/line info when there is an error in a raw PDF directive.
- Fixed Issue 336: JPEG images should work even without PIL (but give a warning because sizes will probably be wrong)

- Fixed Issue 351: footnote/citation references were generated incorrectly, which caused problems if there was a citation with the same text as a heading.
- Fixed Issue 353: better handling of graphviz, so that it works without vectorpdf but gives a warning about it.
- Fixed Issue 354: make todo_node from sphinx customizable.
- Fixed bug where nested lists broke page layout if the page was small.
- Smarter --inline-links option
- New extension: fancytitles, see <http://lateral.netmanagers.com.ar/weblog/posts/BB906.html>
- New feature: tab-width option in code-block directive (defaults to 8).
- Fixed Issue 340: endnotes/footnotes were not styled.
- Fixed Issue 339: class names using _ were not usable.
- Fixed Issue 335: ugly crash when using images in some specific places (looks like a reportlab bug)
- Fixed Issue 329: make the figure alignment/class attributes work more like LaTeX than HTML.
- Fixed Issue 328: list item styles were being ignored.
- Fixed Issue 186: new --use-floating-images makes images with :align: set work like in HTML, with the next flowable flowing beside it.
- Fixed Issue 307: header/footer from stylesheet now supports inline rest markup and substitutions defined in the main document.
- New pdf_toc_depth option for Sphinx/pdftool
- New pdf_use_toc option for Sphinx/pdftool
- Fixed Issue 308: compatibility with reportlab from SVN
- Fixed Issue 323: errors in the config.sample made it work weird.
- Fixed Issue 322: Image substitutions didn't work in document title.
- Implemented Issue 321: underline and strikethrough available in stylesheet.
- Fixed Issue 317: Ugly error message when file does not exist

You can get a free Issue of [PET: Python Entre Todos](http://revista.python.org.ar).. _pet: python entre todos: <http://revista.python.org.ar>

our python magazine.

400 will be printed thanks to BBLUG, PyConAr and PyDay Rafaela.

I hope I can get a few for myself, being the magazine responsible editor and everything :-)

And if you won't be there: it's CC-by-nc-sa so you can print it too!

Today is 10/10/10 (or rather 10/10/2010 which doesn't look so cute, right?) and again there's a flood of "this only happens every 100 years! OMG!".

At least this time it's correct. However:

How often is 10/11/10 going to happen? How often did yesterday's 10/9/10? Every 100 years, too. In fact, every date written in that format will happen once every 100 years, because that's the way **dates work!**

Oh, you may say, the special part is that the three numbers are the same, silly!

But then again, last year we had 9/9/9 and next year we'll have 11/11/11 (much cooler than 10/10/10!) so ... no, not that special.

OTOH, today (at 10PM) my kid and I will see the first ever episode of Ben10 Ultimate Alien (hey, 5 10s!) for the first time and that... well, that has never happened before, and will never happen again. That, friends, **is** special.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "'what's the most mind-blowing fact you heard/read in your life?'"

There's a really awesome thread at Reddit called "What's the most mind-blowing fact you heard/read in your life?". "what's the most mind-blowing fact you heard/read in your life?": <http://www.reddit.com/r/AskReddit/comments/dlrjs/>

where commenters are posting well... that.

It's a great read! But there are a few here and there that are a bit undeserving of the title, because they are just wrong :-)

Here are a couple from the beginning of the thread (it's got over 4000 posts now).

"There has been no evidence suggesting the universe is not finite. If we assume the universe is infinite, then it is certain that every single possible combination of events has happened somewhere in the galaxy. In an alternate area, all of human history is the same, except you don't decide to post this thread. With certainty."

—MothersRapeHorn

Beyond confusing the galaxy with the universe, let's think this a bit.

It's funny, but I remember reading something like "if the universe is infinite then everything that is possible has **certainly** happened somewhere" in a cheap scifi booklet maybe 30 years ago.

In that story, that reasoning was used to justify a naked cyclops that walked in space, IIRC. It was found by a starship called SKY-11111. If anyone has read that story, he has my sympathy (I should write about how hard it was to read scifi when I was a kid, beggars can't be choosers!)

But the problem here is just a basic misunderstanding of probability.

Let me explain with an example: Consider the odds of someone spending a year flipping a fair coin and getting "heads" every time. That's something very unlikely. Let's call that probability P1.

Now, let's consider the probability of "there has never ever been a person that flipped a fair coin for a year getting only 'heads', and there never will be", and call that P2.

Now, if the universe is large, P1 will grow. As there are more people, there is a larger chance of someone deciding to waste a year and then there **is** a tiny chance of him being incredibly ... (well, not lucky, see how he spent his year!) but let's say, consistent? And getting heads all year.

What the heck, let's say P1 is now 50%, so it's not even unlikely.

But that means P2 is also 50%, since it's true exactly as often as P1 is false.

Now, which one of these two things will happen "With certainty"? See? Likely is not the same as sure, unlikely is not the same as impossible, and adding an infinity doesn't do the trick here.

So, really, even if the universe is infinite, and even if it were crowded with people, there are lots and lots of things that are never going to happen.

I find that more mindblowing than the original, but that's just me.

Here's a simpler one:

If you took all of the silver the Spanish mined from Cerro Rico, you could build a bridge from Potosi, Bolivia, to Spain. If you took the bones of all the Indians (about eight million) who died in the mines, you could build another bridge back to Potosi from Spain. Cerro Rico has been so thoroughly mined that there is an entire mountain next to it of the rubble extracted from it. The Indians call this second one "the mountain that weeps."

—tofkety

Here I am guessing he is greatly overestimating the amount of silver extracted from Cerro Rico.

Spain is roughly 8800km away from Cerro Rico. A 2-meter wide path, 20cm thick would mean 3520000 cubic meters of materials. That's much **much** less than a bridge would use.

Cerro Rico is about 4800 meters tall.

So, this means the material for that path would be enough to build a column with square base 27 meters wide, and as tall as Cerro Rico, of *pure silver*.

That would be, considering silver has a specific weight of 10490... 36924800000000 kilograms of silver. That is... well, that's a lot of silver.

How much silver? Well, in dollars, that would be \$27508976000000000

In any case, the thread is **great** fun, and everyone who has, say, a young kid should read bits of it ... with a skeptic eye open ;-)

Here's a little article about backing up your own personal data safely.

I wrote it because it seems this kind of solutions are not very well known.

While not a real, serious backup strategy, it's pretty safe, as long as the system still lives, you can recover your data.

You are a nerd. I know that because you are reading my blog. You probably don't cook. That is stupid. Let me explain why.

Cooking is easy

You may be intimidated by those who cook and make it look complicated. Don't believe a word they say. You can cook a perfectly fine meal for 1 to 4 people in half an

hour.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "cook it like risotto".

For example, if you have dry pasta, any kind of protein (chicken, seafood, ground beef, sausages, chorizo, fish, whatever), a bouillon cube (vegetables, beef, whatever), garlic, onion and/or anything like it, and a glass of wine (optional) you can [cook it like risotto](#)..
cook it like risotto: <http://www.theatlantic.com/food/archive/10/05/recipe-risottostyle-seafood-pasta/57171/>

And here's the best part: read that recipe, and here's all you need to know to cook it:

- Toast the dry pasta in kinda hot oil (olive oil if you have it, or whatever).
- Add chopped onions, garlic, peppers (or whatever) for 3 minutes.
- Add 1/2 cup of wine (it says white, but red wine will work, it will just make it taste "heavier" and change the colour). If you don't have wine, use broth. Wait two minutes or so until it's almost dry.
- Add broth slowly while stirring.
- When the pasta is still kinda hard, stop adding broth, add protein (chopped small so it cooks completely) and a bit more broth and lemon juice (or not).

Or you can do this:

- Sauteé onions, garlic whatever and protein, then put it aside and save it.
- In the same oil, toast the pasta. This makes the pasta tastier.
- Re-add the protein and vegetables, then add wine and broth slowly
- When the pasta is the way you like it, eat it.

See what I did there? I did it almost exactly the other way around. And you know what? It's still going to taste good. Why? Because cooking is very fault tolerant.

Yes, you will hear all the time about the perfect point for this, and the perfect seasoning for that, and ... it's 90% bullshit.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "a freaking \$4 kitchen timer".

Sure, if you overcook the pasta it's going to be soggy crap, but you can avoid that by being minimally vigilant about it and buying [a freaking \\$4 kitchen timer](#).. a freaking \$4 kitchen timer: <http://www.dealextreme.com/search.dx/search.kitchen%20timer#1013>

Cooking is fun

Once you get over the notion that it's hard, cooking is easy to enjoy.

If you have a kid, he can help. If you have a spouse, he/she will like that you are taking care of the meal.

It's great for unwinding after being stressed. Strangely, chopping garlic relaxes me.

Have a microwave and a 3 year old? Then you can do this:

In one bowl mix:

- Two beaten eggs
- Add vanilla (or don't)
- Add a bit of butter (or vegetable oil)

In another bowl:

- 1 cup of regular all-purpose flour (or whole wheat flour)
- a bit of baking powder
- a bit of salt
- half a cup of sugar

In one of those bowls add something else. If it's moist, add it with the eggs and such, if it's dry, add it with the flour and such.

That "something else" can be chocolate chips, walnuts, shredded carrot, bananas, sliced apples, jam, *whatever*.

Once each bowl is mixed, mix them together a bit until they "mesh". Don't work it too hard or the result will suck.

Then get two (or three, or one, or four) mugs (or cups, or muffin trays, or anything that's microwave-safe) and fill it half-way with the resulting batter.

Microwave it until it looks good, then eat it. If you are not sure if it's cooked, stick a wooden toothpick in it and see if it's dry when you pull it.

Your 3 year old can take care of the dry bowl (he does similar things in kindergarten, you know?) as long as you help with the measures. And most importantly: he can take care of starting the microwave, so technically he can say **he** cooked it (3 year olds are very into technicalities).

Again, see what I did with that recipe? You can replace almost any ingredient with something else (I don't recommend not using eggs, though) And it will probably work.

And you will end with a flour-covered kid, which is a bonus, because then he won't make a fuss about his bath.

Cooking is good for you

I have high blood pressure and I'm fat. But you know what? I have not gained any weight since I started cooking for myself, 10 years ago. I gained a **lot** before that, though.

Now I can cook semi-healthy food for me, and do it with the low sodium I need. Sodium is a habit, so after you are eating low sodium for a few weeks you don't really miss it.

If you are a nerd, you probably are a bit too sedentary, so eating "right" will help you.

And most importantly, you know what you eat. Sure, you also need to take care of what ingredients you buy, but did you know that pizza has a like 400% markup? And that you can do a reasonably healthy pizza in, like, 40 minutes of work?

The only trick is starting 3 or 4 hours early.

But if you use home-made dough, chopped tomatoes for sauce, garlic, peppers, and not too much cheese... it's not a terribly unhealthy meal.

And of course, you can always do a salad. Come on, how hard could that be? It's chopping things up and piling them, dude!

And it **can** be tasty and filling. You just need to do a lot of it ;-)

Cooking is cheap

A milanesa (sort of a breaded steak?) in a cheap restaurant will cost you about \$25. For that money you can make 7 or 8 at home.

A good piece of prime meat in a restaurant? \$50. That buys 2 pounds of prime meat in the market for you to cook.

A reasonable lunch menu for office workers costs about \$30. For that money my family of 3 eats twice. And better. And more.

You can make yourself the most awesome sandwich in the world for under \$10, with anything you imagine in it and take it to the office.

And, most importantly...

Cooking is Applied Nerdiness

Cooking gives you a way to easily experiment with real-life chemistry, gives you something concrete to show for your effort after a whole day coding things you can't really show your family.

Why does ham taste well with apricots and not with apples? Or does it? How about chili? Does it go well with sour cream? How about pepper on a strawberry?

But forget about adding: What can you **remove** from a recipe and still make it work? What can you replace? If you can make banana bread, can you make banana croissants? (no you can't, they taste awesome but they look ghastly)

How can you go through life without wondering if you can really cook fish with just lemon juice? Exactly how long do you cook rice in your microwave so it looks creamy and taste great without removing excess liquid? Did you know you can steam 2 pounds of potatoes in 10 minutes with a plastic bowl and some plastic film, and they will make the most awesome mashed potatoes you ever tasted?

Your kitchen is an awesome place. Instant noodles suck.

A while ago I wrote an article about gay marriage. (I am all for it, BTW). In it I said something like "since god doesn't exist ... " and boy did that bother people.

So, since I have twenty free minutes, let's see if I can explain why I say god doesn't exist.

First of all, a caveat. If you are religious, you have no right whatsoever to whine about me saying this. Why? Because I have no right to whine about people saying god *does* exist. It's called freedom of religion, people. You are supposed to like it.

This was triggered because I noticed there's a grand total of 2 (two) atheist characters on TV shows I watch:

- Gregory House (MD): apparently a narcissistic bastard, but actually a nice guy (who is also a genius) with issues.
- Dr. Brennan: a socially awkward genius.

Well, I am not a genius, so, let's consider some very reasonable arguments for god's lack of existence.

He's Ill-Defined

The first thing you need in order to accept the existence of an entity is a definition for it. If you lack that, how can you say it exists at all? He could ring my doorbell and ask for a cup of sugar, and I still wouldn't be sure, because he could be re-defined at any time.

For example, is god omniscient? Is he allmighty? Is he the guy with the elephant head? Is he immaterial? Does he answer to prayer? Is he a he? Did he have a kid? Did he have his kid by turning into a swan before going on a date?

Since depending on what godist you ask he will answer at least one of those differently, I have to declare his existence impossible.

Of course we could try to accept the definition of *one* godist club and try to see if that specific entity exists, but that doesn't really work either, because god fans have a tendency to move the goalposts. What's "the word of god" becomes later an allegory, depriving us of any evidence on which to base our enquiry.

Mostly, godists say that they know god exists because they feel it in their hearts or something similarly harebrained. Come on, if I told you I feel the easter bunny in my kidney, it would make about as much sense.

The Excluded Middle and Popularity

Either something is true or its opposite is. Either you ate some of that cake, or you didn't. Either god exists or he doesn't. Easy, right?

But why couldn't god exist? Well, let me ask you, why don't the **other** gods exist? You are a zoroastrian: why doesn't Zeus exist? You are a mormon, why doesn't Quetzalcoatl exist?

Every godist is perfectly happy with **the other** gods not existing, so it's not exactly a ground-shaking notion. It's clear that whenever you hear anyone talk about a religious majority, he is full of crap.

Repeat after me: you are not part of a religious majority, because most people believe **your** god doesn't exist. We atheists are just smarter and more consistent.

And no, you can't retreat into "oh, muslims jews and all christians believe in the same god" because that's nuts. Jews believe in a god that doesn't let them eat ham. Mormons believe they are ordered to use magical underwear. Catholics believe they eat meat wafers each sunday, it's just that it looks, feels and tastes like a cracker, but it's "really" (super)human beef. For each group, the other's beliefs are barbaric and (if they are honest) a little nuts.

No, I am not saying you *individually* are nuts, you nutcases, I am saying you are conditioned to believe your *particular* idiosyncrasies are less nuts than average, but

they aren't, just like my dad's habit of putting mayonnaise in the soup was nuts and my belief that Unión de Santa Fe will someday win a tournament is nuts.

IOW: mostly harmless, but nutty anyway. OTOH, some people's beliefs make them believe that killing albino kids is a proper behaviour so some of you godists are really, **really** nuts, ok?. Not **all** of you, but those who aren't should take a good hard look at what believing in invisible friends does to some people.

It's Unethical to Believe in Heaven and Hell

Consider my three year old kid. There is a rule that he has to eat a reasonable amount at dinner, and if he does he can watch **one** TV show before bed as a reward.

In universe A: One night he's very tired, so he doesn't really want to watch TV, he wants to go to bed, but he still eats his dinner because it's good for him.

In universe B: One night he's very tired, so he doesn't really want to watch TV, so he doesn't eat his dinner because there's no reward.

Believers will tell you that human nature is B. That if there was no promise of carrot (heaven) and stick (hell) humans would have no morals and would act like insane hedonists, hurting each other in a frenzy, and that we only avoid such a terrible fate because of the civilizing influence of the churches and the morality induced on us by religion.

I say bullshit. I say I prefer if my kid does what's good for him not because he's expecting a reward or (worse!) because he's scared of punishment, but because he understands that if he eats his dinner he's going to be strong and healthy, and that it makes me happy and that he wants me to be happy because he likes me.

Of course, being a three year old, he sometimes doesn't want to eat his dinner. So I try to convince him. But if he doesn't, he doesn't, and there's no TV, and there's no tantrum, and he gets a kiss good night.

The concept that there is a lot of people who honestly believe that they are moral beings only because there's an invisible guy who will hurt them if they aren't scares me. I find it deeply repulsive. I find religion's promise of eternal (or even temporary) punishment in the afterlife repulsive and creepy.

If you believe in an afterlife, and you believe in heaven and hell, and you act nice because of it, you are a creep. You are, like religious people like to say, a creep in your own heart. You are not good. You are evil but just think you can't get away with acting out your evil. You are a chicken. If the deity you believe in actually exists, he knows it, so you are screwed anyway.

I prefer to be good for my fellow men **because** there's nothing else beyond. If there was a heaven, then we live in a crappy waiting room. No! We live in the real world. What's beyond is fiction or guesswork, you can't count on it, you can't throw away real life for it.

Conclusion

You are a bit nuts and your invisible friend doesn't exist. You don't need to get a life because you already have one, you just need to stop asking for seconds and eat your dinner. Have fun.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "cúspide".

Here's pretty much the only place where you can buy Mark Summerfield's "Python 3" book in Argentina: [Cúspide.. cúspide: http://www.cuspide.com/isbn/8441526133](http://www.cuspide.com/isbn/8441526133)

It costs \$372.50 in pesos which is about \$94 in dollars.
Oh, and you have to go to a book store to pick it up, or add shipping.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "\$16".

How much does it cost to buy that book in **england** and have it shipped to your door? \$16... \$16.: <http://www.abebooks.com/servlet/BookDetailsPL?bi=3605010243&searchurl=an%3Dmark%2Bsum>

That's a whopping 17% of the local cost.

And no, it won't pay import taxes, because books are exempt.
And did I mention that the english version came out a year earlier?
So, if you don't learn english, you pay almost 6 times for the book, and wait a year.
Any programmer that doesn't know enough english is a third class citizen.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "new phone yay!".

I just finished reading Murray Leinster's Space Platform (in my [new phone yay!](http://europe.nokia.com/find-products/devices/nokia-5230/specifications)..
new phone yay!: <http://europe.nokia.com/find-products/devices/nokia-5230/specifications>
).

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "manybooks.net".

You can read it too, if you want, because it's available, for free, from [Manybooks.net..](http://manybooks.net/titles/leinsterm2505125051-8.html)
manybooks.net: <http://manybooks.net/titles/leinsterm2505125051-8.html>

in any format you may need.

It's a very old-fashioned (published in 1953) scifi story, but what really shocked me was that in the 25 years between this and Star Wars (1978) *everything* changed.

Why? Because this is a book written from the perspective of workers building the Death Star.

Specifically, the main character, Joe, is working on building the gyroscopes for a space station which will be the first permanent artificial object in orbit... and is fully loaded with nukes.

Further, it's strictly a USA project (although there is a mention of it "being offered" to the UN) and the whole book is spent showing the courageous workers and soldiers fighting saboteurs in Arizona.

Replace USA by "the empire", workers and soldiers by storm troopers, space platform by death star, communists and anarchists by ewoks and rebels and... well, it's "Return of the Jedi", except the empire wins and all ewoks are killed in the end.

This short novel is completely acritical: US having the power to destroy any city in the world at will is **good**. All other countries being unable to retaliate is **good**. Trying to prevent it by any means? **Bad** and **cowardly**.

In just 25 years, though, films describing the situation exactly from the opposite point of view had every kid cheering for the saboteurs.

It's amazing that this book is closer in time to Star Wars than Star Wars is to today.

I wrote this for a contest at the New Scientist magazine. I thought what the heck, maybe someone will like it. And no, I won't explain it, because that spoils the whole thing.

There is no Such Thing as Free Energy

I wished a cold drink was still a possibility, and looked out, across the baked clay that used to be a swamp. The hatch closed and we started our long trip to the stars, cursing the inventor of the perpetual motion engine all the way.

A few weeks ago I was chatting with my father in law and (since I work with computers and must therefore know everything computer related) he asked me if I had heard of the 9 year old kid that worked for Microsoft as an engineer.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "pea in a maraca".

I said that probably, hiring 9 year olds to work as engineers was illegal, and that in most places to become an "engineer" you need to go to college, but any way it stayed in my head, like a [pea in a maraca](http://es.wikipedia.org/wiki/Archivo:Sambaballen.JPG).. pea in a maraca: <http://es.wikipedia.org/wiki/Archivo:Sambaballen.JPG>

but anyway, I decided to check it out a bit.

First: no, there is no 9 year old working for Microsoft, as far as I know.

And then, a curious pattern appeared: there is not one story about that, there are several. And about different kids. And mostly in spanish-speaking media.

Let's check Mahmud Wael first.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "infobae".

Here's what [InfoBAE](http://www.infobae.com/mundo/544314-101515-0-Tiene-11-anos-y-es-ingeniero).. infobae: <http://www.infobae.com/mundo/544314-101515-0-Tiene-11-anos-y-es-ingeniero>

says about him:

Mahmud Wael, un egipcio de 11 años y aspecto frágil, es el nuevo técnico de Microsoft gracias a su capacidad para resolver complejos cálculos en cuestión de segundos y moverse sin problemas por las redes informáticas

Translated:

Mahmud Wael, a fragile looking 11 year old egyptian, is the latest Microsoft technician thanks to his ability to solve complex calculations in seconds and to move effortlessly through information networks.

If one actually bothers reading the story there's more: apparently Mahmud joined the American University in Cairo at age 9, and is now attending Cairo University for a degree in some computer-related area.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "because".

Now, I don't want to pick on InfoBAE because.. because: <http://www.20minutos.es/noticia/857841/0/nino/egipto/m>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "the".

the.. the: <http://crisolplural.com/2010/10/30/mahmud-wael-de-11-anos-nino-genio-en-microsoft/>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "same".

same.. same: <http://www.midesahogo.com/2010/10/31/increiblemahmud-wael-con-solo-11-anos-ya-trabaja-par>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "thing".

thing.. thing: <http://www.elnuevoherald.com/2010/10/31/829296/un-pequeno-genio-asombra-a-microsoft.html>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "is".

is.. is: <http://www.meneame.net/story/mahmud-wael-3-anos-resolvio-multiplicacion>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "reported".

reported.. reported: <http://vicioconsolas.activoforo.com/noticias-f12/wael-un-nino-de-11-anos-tecnico-de-mic>
in many other places (the previous links are just the first few that google gave me).

In fact, a bunch of those stories even say "With 11 years, he already works at Microsoft", which is somehow not in the InfoBAE story which is taken from the EFE agency.

Now... does he work at Microsoft? I bet he doesn't (or EFE would have mentioned it). It's just that when someone writes "Microsoft Technician" or "Microsoft Engineer" in english, well, that makes no sense in spanish, so the spanish media and readers are lead astray.

A Microsoft Certified Engineer is someone who has taken some Microsoft training courses and exams.

On the other hand, in most of the spanish speaking world, you can't call yourself an engineer unless you get an engineering degree from a university. In fact in Argentina calling yourself an engineer if you don't have one is **illegal**.

So, "Microsoft Engineer" is taken as "an engineer that works at Microsoft", because the alternative simply makes no sense.

What is the real story about Mahmud Wael? Well, let's check some *egyptian sources*, which is what all those newspapers should have done in the first place.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "egypt today's take on it".

Here's [Egypt Today's take on it](http://www.egypttoday.com/article.aspx?ArticleID=7882).. egypt today's take on it: <http://www.egypttoday.com/article.aspx?ArticleID=7882>
from when he was 9.

Did he attend the American University?

Well, he had a scholarship from them to attend the Greenland International Language School, and attended one english course.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "says he got a mcts".

What about the "Microsoft Engineer" thing? He was planning to take the MCSD exams. Did he succeed? Well, Reuters [says he got a MCTS](http://www.itnsource.com/shotlist/RTV/2010/09/30/RTV2483010/).. says he got a mcts: <http://www.itnsource.com/shotlist/RTV/2010/09/30/RTV2483010/>

An MCTS is not an MCSD, or an MCSE. In fact, just by saying someone has an MCTS (very impressive for an 11 year old!) you have no idea of *what* he knows, because a MCTS is about a specific product, and there are MCTSs for almost all of MS products.

So, in short: Mahmud is a very impressive and intelligent kid, but he is not an engineer, have a college degree or work at Microsoft.

And now the second case, Marko Calasan from Macedonia, which is the one actually mentioned to me.

And it's *exactly* the same story, except that he got a much better cert from MS than Mahmud, and he got it earlier.

Again the "works at Microsoft" thing seems to be exclusive to the spanish speaking media, and probably for the same reason.

Now, let's think about what this says of journalism. These stories were *not* hard to check. All you need is passable english skills and google. And if your english sucks, google can help you with that too.

But dozens of newspapers and sites just run with it because the "Microsoft hires (small age) kid" is just too nice and people would accept it because hey, it's in the newspapers.

And you know what? I suspect that it's the same thing with a large part of what you read in the papers. If checking a tiny piece just because I have some peripheral knowledge about it says there are dozens of articles that are just *wrong*, what happens in all the areas where I am clueless?

Because we are all clueless in almost everything, and journalists are probably clueless about 90% of what they write about. It's not even a conspiracy, it's just ignorance amplified by their job description.

Short intro for foreign readers: the largest newspaper in Argentina (Clarín) is in a catfight with the government. Therefore, we are treated daily to stories in the newspaper about how everything is terrible and the government is going to eat our children, and stories in the official TV channel about how Clarin wants to implant dancing contests and biased news into our prefrontal lobes.

The family subsidies are a recent policy that can be easily described: if you have a kid and you send him to school, you get a little money (\$220). That's because having kids in school is a good thing. This has caused school enrollment to increase a lot in one year, meaning a ton of poor kids are now back in school instead of working in the streets or just staying at home.

Since it's hard for Clarín to go ahead and say that's bad, it has to find an angle. How about saying that inflation (which has been raising) is making the subsidies useless? It's an idea.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "this story".

Having said that, it's hard to take [this story](http://www.clarin.com/politica/alimentos-licuo-parte-subsidi).. this story: <http://www.clarin.com/politica/alimentos-licuo-parte-subsidi>

and not say... dudes, you are giving biased journalism a bad name.

Here's the title and intro:

The raise in food prices has eliminated a big part of the family subsidies.

Depending on what indicators you use, the erosion can reach 92%

Taken at face value, that's pure nonsense. In order for that to be right, it would mean that the peso has lost 92% of its value and it has lost between 10% and 20% depending on what you compare it with.

What they did instead is take the cost of a basket of basic goods that has raised 36.2% (according to some estimates) and convert that raise into pesos. That's \$404. Which is 92% of the family subsidies you get if you have two kids.

What's the problem? Well, of course the problem is that it makes no sense, because it's comparing two unrelated things.

Let's consider two moments in time, at the beginning of the subsidies and one year later. The basket of goods has raised from \$712 to \$1116.

Let's consider the case where that family **only** has the subsidies, both parents are unemployed and receive no help at all:

They have gone from covering 62% of their basic needs to covering 40% so they are obviously worse off now than a year ago. But not 92% worse, no matter how you cut it.

A bit more realistic: the family had *some* income other than the subsidies. Imagine only the mother works cleaning houses part time. That means she makes perhaps \$500 discounting travel expenses.

So, a year ago, they made \$940 and covered 132% of their basic needs, and now they cover only 84%.

But that ignores that pretty much everyone has had pay raises in the last year, precisely because of inflation. So assume she got a very modest raise: 10%, and she now gets \$550.

That means she went from covering 132% to 89%.

Of course without the subsidies they would have gone from 70% to 49%! Try telling that mother that the subsidies have lost 92% of their value, and she'll laugh in your face.

Of course that means they are desperately poor, and yes, their salaries are worth less (if you take those numbers at face value, general inflation was much less than 30%).

But those \$440 are *something* that was not there before. It is not a bad thing, and it is not a useless thing. And most certainly it's not a thing that has lost 92% of its value in a year.

Shame on you Clarín for trying to use "math" to confuse people.

Recently I was delighted to read in Boing Boing posts by a modern Stoic. The delight was because it put into words something I had been grappling with for years and never really grasped: people have replaced philosophy with religion.

It used to be that someone would call himself a stoic, or a cynic, or a hedonist, or whatever, and others would understand that he was telling them the principles that rule his life.

A life philosophy! You could choose, from the buffet of the last 3000 years of thought, what you thought made most sense, and try to use it as a beacon to guide you through a (hopefully) happy life.

Nowadays, society seems to have rejected that idea, and the closest thing most people have is religion, following what his sect says, or atheism, defined by rejection of religion.

The main difference (as I see it) between a life philosophy and a religion is that a religion usually implies the others are wrong. If you are not of my sect, you will not be in heaven with me.

If you don't share my philosophy... well, I expect you will take a different path through your life than I would have taken. But if it works for you and doesn't hurt others, why should I give a damn?

So here's my life philosophy as I see it today. It's not how I saw it yesterday, and surely is not the same it will be tomorrow.

From now on, when I say I "believe" something, it's shorthand for "my personal life philosophy implies that". It should be obvious why such a shorthand is needed.

I am a materialist. No, that doesn't mean what you think it does, at least not in this context. What I mean is that I am not a dualist, or a spiritualist, I am not an idealist or a vitalist, and not a phenomenalist.

What it means is that I believe that reality is material. I don't accept that immaterial things have any sort of "reality". Or at least that their reality is of a totally uninteresting kind.

This means that I don't believe in souls. I believe the Turing test is a reasonable test for consciousness. I believe if there was an entity that acted like a human, we ought to treat it like a human. I believe I am not intrinsically different from a robot that could do what I do.

I believe the purpose of life is to have a good time. I believe everyone is as entitled to a good time as I am. I believe part of having a good time is being surrounded by happy people. I believe people that hurt others are a buzzkill and shouldn't be allowed to do it.

I believe in purpose, and I believe I create my own purposes and that makes them better than if they were given to me. I believe in being kind to others because they are all I have.

I believe in learning, because we are surrounded by wonders. I believe the Egyptians piled up lots of very heavy rocks. I believe Saturn is pretty. I believe giving the merit of those things to aliens or gods is an insult to the Egyptians and adds nothing to Saturn.

I believe in making things and fighting against local entropy. I believe that a certain end makes things better and more precious. I believe in love, because I know I feel it and it's precious.

So there.

There is a prejudice that the poor play lotteries because they are lazy, can't save and are generally stupid and are hurting themselves by chasing the fantasy of winning instead of saving pennies. You know what? It's bullshit.

When I was in high school (about 13 years old), I once had a plan to make money: I would play the lottery. Here's the mechanism I had in mind.

I would play \$1 in the quiniela. Quiniela pays \$700 for each \$1 you bet, and you have to choose a number between 000 and 999. My idea was: I can bet the \$1 my parent give me every day, and there's a chance I make \$700. If I had \$700 I could buy anything a 13-year old kid may want. With \$1? Not so much.

Of course you are right now thinking: What a moron! He has a 0.001 chance of winning and it pays 700 to 1, so it's a losing bet! Bzzzzzt!

Let's start with some simple simulation code:

```
\ :kn:'import' \ \ :nn:'random' \
\n :n:'n' \ \ :o:'=' \ \ :mi:'476' \
\n :k:'for' \ \ :n:'tests' \ \ :ow:'in' \ \ :nb:'range' \ \ :p:'(' \ \ :mi:'10
    \ :k:'for' \ \ :n:'w' \ \ :ow:'in' \ \ :nb:'range' \ \ :p:'(' \ \ :mi:'10
        \ :n:'q' \ \ :o:'=' \ \ :n:'random' \ \ :o: '.' \ \ :n:'randint' \ \ :p:
        \ :k:'if' \ \ :n:'n' \ \ :o:'==' \ \ :n:'q' \ \ :p: ':' \
            \ :k:'break' \
```

```
\ :k:'print' \ \ :p:'(\ \ :n:'w' \ \ :p:') \ \
```

Short explanation: run 10000 simulations of this process:

- We play each day for 1000 days.
- If we win, we stop.
- If we don't win in 1000 days we stop.
- We record the number where we stop.

So, I ran it. Here's a graph of the results

So, how many never won anything? In my data set: 3699 players out of 10000 never won anything.

How many actually *lost* money? 5030 players.

And how many *won* money? 4967 players won money.

2910 players won in less than 350 plays.

3 players got exactly even money, winning in their 700th play. For them, this was exactly the same as saving their money.

So, is it a good idea to play a lottery like this? It's a coin toss. Half the time, you end with no money. Half the time, you end with more money than if you had saved.

If you are betting disposable income (a sufficiently low amount that "it doesn't hurt"), it works out. You have a fair chance (50%) of a reward at least as good as saving the money, and a decent chance (25%) of a reward twice as good.

And you have a fair chance (50%) of losing money. But you would lose it very, very slowly and painlessly. ¿How well do you think stocks compare to that? ¿And what are the barriers to entry on both games?

In short: playing the lottery is not irrational, really, it's just a savings plan. It sure was a better idea than buying candy!

I have a strong tendency to be argumentative. That's because I really enjoy a good argument, if you'll pardon the obviousness.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "lamarckian vs. darwinian evolution".

The best thing about a good argument is that you get the most amazing insights from the wrong side of it. For example, [Lamarckian vs. Darwinian evolution..](http://lateral.netmanagers.com.ar/stories/34.html) lamarckian vs. darwinian evolution: <http://lateral.netmanagers.com.ar/stories/34.html>

: Lamarck was wrong. But Lamarckism is a heck of an idea, and once you *get* lamarckism, you can pass it on! (ha!).

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "the chicken and the egg".

Or, [the chicken and the egg..](http://lateral.netmanagers.com.ar/stories/7.html) the chicken and the egg: <http://lateral.netmanagers.com.ar/stories/7.html>

? I actually got into an argument (and I did not start it myself) about this a couple of weeks ago.

If you start with something that's wrong, you can backtrack and see *why* it's wrong. What was the implicit mistaken assumption, the incoming garbage that created the

outgoing crap. And then you can tweak it. And see what new garbage comes out. Wrong stuff, but new stuff.

And that's one of the great things about being a nerd: nerds are the awesome at this. Oh, you may think people in politics would be better? Nah, they never change their minds. Lawyers? Well, they argue for money!

But nerds? We do it for fun. And most of us don't give a damn about looking weird to others because we already *know* we look weird to others.

Spending 4 hours locked in a car with average humans is mostly a chore. Someone will play music, maybe people will talk intermittently about stuff that happened in the last few days, whatever.

But lock 4 nerds in a car for 4 hours and you're going to listen to *stuff*. This happened to me twice in the last few weeks. And in between, we had a dinner with a very high nerd factor (with alcohol assist)... great fun.

Now, I know many experience this when they are with people that have a certain thing in common. I've seen something similar happen between, for example, communists and former communists. They would talk for hours, and it was lots of fun (even for me ;-)) but what they talked about was their common thing: communism.

Nerds apparently don't have such inclination. Nerds will argue about anything. Nerds will argue about *everything*. And that makes me think... why are nerds seen as shy and introverted? Hell, why **are** nerds shy and introverted? How can I reconcile what I see when I'm among fellow nerds and how others see us?

And it's not easy. I am by all standard measures painfully shy. I had great difficulty making friends when I was a kid. I didn't like the things other kids liked. I didn't know many things they knew all along. I continuously was the butt of jokes for being naïve on things I had never heard about. I was always afraid to speak when I was in a group because I thought I would make a fool of myself.

Shy doesn't mean boring. Shy doesn't mean someone who doesn't have anything interesting to say. Shy means someone that has problems *starting*.

On the other hand, I have spoken in front of hundreds and I've been told I look relaxed (I am not, I am faking, guys). I write under my real name and I never felt something was "too weird" so I shouldn't write it (coming soon: an economic explanation of why men like to see women kissing), so I am not really afraid of people thinking I'm weird. What the hell, I *know* I'm weird. Ask me if you see me: Am I weird? Yes!

But I still have trouble when I am in a party with people I don't know (I am lucky my wife is like social WD-40). I still have trouble making small talk. I don't know what happened in the TV show everyone watches. I appear shy and introverted. Until you know me.

I have decided I don't suck at writing technical articles. I am not all that great, but I don't suck.

And I am quick, too.

So, I have decided I would like to get paid. I tried sending email to editors@newsforge.net (they claim to pay for content) but have not gotten even a "no thanks" response. Maybe they are slowish ;-)

Anyone knows any other sources of income for someone who can write decent short technical articles?

I think I will contact LinuxWorld and Linux Journal, but they are a bit too ... high level. You know. Real magazines ;-)

I only mean for this to pay a few bucks, it's not what I intend on living from ;-)

In the meantime, I will continue writing one or two articles a week, anyway!

A few days ago, the last video rental in the neighborhood closed. It was a Blockbuster and since it's bankrupt in the US it's hardly surprising that they killed the Argentina operation.

But Blockbuster had, years earlier, killed all the *other* video rental shops. So now there aren't any, at all.

So, how does anyone with a DVD player actually use it? Well, he can buy DVDs in the newspaper stands. That's expensive because it's *buying* and it's *late*. Who will want to spend \$35 to watch "Due Date" at home 3 months after the theatrical release? There is no NetFlix here!

And of course, you can do what everyone was doing anyway: download it, or buy illegal copies. They are often better than the originals anyway, at least in this sense:

And really it's amazing. The whole movie rental industry has basically ceased to exist, and the competition is illegal. They sucked so much at its business that they couldn't compete with a "business" that can't do commercials, has no quality assurance.

Sometimes you got a very bad illegal copy, filmed from a seat in the last row, full of people talking over the movie you can barely see... and still, people preferred to buy *that* for \$7 instead of paying \$10 for a "quality" rental.

It's a tale of incredible incompetence. Quickly, can someone find me **one** example of an business that got killed by illegal and crappy competition?

There are crappy and/or illegal medicines, but pharmaceuticals still work. There are illegal (unlicensed) places to eat, but restaurants still make money. There are illegal cabs, but real cabs are in business.

This is not even bad, it's just embarrassing.

Neal Stephenson wrote:

There is something new: A globe about the size of a grapefruit, a perfectly detailed rendition of Planet Earth, hanging in space at arm's length in front of his eyes. Hiro has heard about this but never seen it. It is a piece of CIC software called, simply, Earth. It is the user interface that CIC uses to keep track of every bit of spatial information that it owns - all the maps, weather data, architectural plans, and satellite surveillance stuff.

Hiro has been thinking that in a few years, if he does really well in the intel biz, maybe he will make enough money to subscribe to Earth and get this thing in his office. Now it is suddenly here, free of charge...

And of course, I have just that very thing installed in my desktop. Not all the mentioned data is hooked into it, but hey, it *is* free of charge.

Heinlein wrote about private citizens and companies going into space. He thought it was not any government's job. And that is going to happen in my lifetime. I know a guy who knows a guy who knows a guy who went to space paying for it with his own money.

Of course there are no flying cars or rocket backpacks (those were good ideas... *not!*)

What's the difference between Gibson's Idris and Gorillaz, except that it's cheaper to pay musicians than it is to build Artificial Intelligences? Can you tell me what's the *point* in building an AI, anyway? Aren't mechanical turks cheaper and better?

Asimov wrote about a foundation of scholars writing an encyclopedia to be constantly updated, containing the whole of humanity's knowledge (we got wikipedia instead. Good enough!)

Our phones are much nicer than Star Trek's communicators (for example, the loud-speaker is *optional*)

It's as if most of the ideas of scifi got filtered through a purifier and what made sense came out on the other side. I like living in the future. I want to see the next one.

This title came to mind when I saw in the news references to an article in The Lancet about how in 2030 7 out of 10 deaths would be due to cardiovascular, diabetes, cancer, and chronic obstructive respiratory diseases.

The lesson most newspapers get out of this is "whoa, we are a bunch of lazy, salt and fat eating morons and we are all gonna die".

Sure, we are all gonna die, and yes, more people will die of those chronic illnesses in 2030. But that's mostly because we are not going to die of many other things that used to kill us earlier.

So, eat more veggies, stop smoking and don't worry too much.

Oh, and about cigarettes and lightning? I must confess I don't have the numbers to prove it, but I would be **very** surprised if that was not the case. After all, smoking 40 cigarettes a day should reduce your life expectation, and the less you live, the less likely are you to be hit by lightning. It's even a direct causal connection!

Let me share with you a piece from Money Central, but I am not picking on them, you can find the same crap on every newspaper or TV channel's financial section:

The push to dump stocks came in response to news that artillery fire was exchanged overnight between North Korea and South Korea. Though tension between the two nations has been persistent, the latest episode marks a dramatic escalation of the tone between them.

That development added to the distress of market participants, who have long held concern for the fragile state of finances among countries in the European Union periphery. Even Ireland remains a point of worry as it has yet to completely structure a bailout package and implement austerity measures.

Distress and uncertainty moved many into the relative safety of the dollar, which was up 1.3% against competing currencies at the close of trade. The advance marked its biggest one-day bounce in a month and put the currency at its best level in almost two months.

But my title is a lie, these guys are not looking at a magic 8-ball. What they are looking at is the rest of the newspaper. Let me share the recipe:

1. See if the market is up or down.
2. Up? Then look at the news and check for good news.
 - a. What? No important god news? Awesome! Then figure out how something that **looks** bad is actually good. That's a golden chance to be declared *insightful*.
 - b. Good news? Bingo, that's why the market is up.
3. Down? Look at the news and check for bad news.
 - a. Bad news? Ok, that's why.
 - b. No big bad news? Great! Then something that's good needs to be described as "causing uncertainty"

In the real world, the business analysts have only the foggiest notion of why the markets move as they do. In fact, if they had a notion, the market would be predictable. If it were predictable, then savvy investors would beat the market.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "doesn't happen".

And that **doesn't happen**.. doesn't happen: <http://newsblogs.chicagotribune.com/urns-on-business/2009/07/no-on>

Spanish only, since it's about a video in spanish ;-)

Acá está, gracias a la gente de Junín, un video de mi charla "Aplicaciones extensibles usando PyQt", en la que intento mostrar como desarrollar una aplicación con PyQt y yapsy.

No es una charla con la que esté muy contento. La otra salió mejor, pero no se filmó, así que quedará solo en la memoria de los cuatro gatos locos que estábamos ahí ;-)

El resto de las charlas: <http://unnoba.blip.tv/>

Again, spanish only because it's a video... in spanish.

Resulta que me olvidé que sí habían grabado mi charla de docutils y compañía. Gracias a Germán por hacerme acordar y mostrarme adonde estaba!

Y ... acá está:

You surely have seen a million posts like this one. Hacker X starts saying what a great time he had at company Y/college/mom's basement/the circus and how he will always miss the people there but anyway it's great to look forward to the challenge of life at company Z/unemployment/feng shui consulting/elefant excrement shoveling.

Well, this one is pretty much the same thing.

I started working at Canonical today. Yes, Canonical. The Ubuntu guys. You may wonder what a dyed-in-the-wool KDE guy is going to do there. Well, it's a job, dude, the 90s called and they want their flamfest back.

I am the new "Engineering Manager for the Desktop+ group". What the heck is that? Well, my job is to help a bunch of talented people I like (at least the ones I've known so far ;-) deliver cool software.

I will probably not be coding much, since this is a grownup job, the kind where instead of lines of code you are supposed to develop gastric ulcers and receding hairlines while you herd cats to the closest cat shed, but I will probably manage to do something, sometimes.

This position came at a good time for me. My kid is going to be 4 next year and go to school all day. What the heck was I gonna do at home all day then, watch anime? Build killer robots? Plan how to conquer the world?

And what happens to my previous job? Well... I still have it somehow. I own a piece of Net Managers (<http://netmanagers.com.ar>) but I will be stepping away from the daily management and operation of the business.

So, basically, I intend to take the money and dump the work on the backs of my capable partners (just kidding). In any case, the company can work just as well without me since we can now maybe hire an employee instead of paying me, so it's win/win ;-)

On other news, I will still work in the same table as the last 5 years, doing some of the same things, with different people. It doesn't sound so big when said like that, uh? Well, I will travel more, and there *are* interesting challenges in this new job.

In short: canonical, little coding, still own netmanagers, happy guy.

Wearing the software engineer's hat: Code is the most trivial and the least important part of a feature.

—Michael Iatrou

Michael tweeted that, I replied, he replied, but what the heck, I think sometimes things can be better explained in more than 140 characters, thus this post[1].

So, why the mall and the outhouse? Because when we talk about software and code and features, we are not all talking about the same thing. Imagine if I told you that bricks are trivial. After all, they have existed in their current form for thousands of years, they are pretty simple to manufacture, and have no interesting features, really, except a certain resistance.

```
system-message
```

```
WARNING/2 in everything.txt, line 3
```

```
Duplicate explicit target name: "an actual brick outhouse".
```

Now, suppose you are building an outhouse. Since you are a funny guy, you want to build an actual brick outhouse.. an actual brick outhouse: <http://www.urbandictionary.com/define.php?term=Built%20>

so you will use bricks to do it.

Now, since bricks are so boring, you may feel compelled to believe bricks are the least important part of your edifice, and that the overall design is more important. Should you carve a moon-shaped hole in the door? How deep should the latrine be?

However, that position is fatally flawed, since if you ignore those trivial, boring bricks, all you have is shit in a hole in the ground. That is because you are considering the bricks as just a mean to your end. You only care about the bricks insofar as they help you realize your grand outhouse vision. I am here to tell you that you are wrong.

The first way in which you are wrong is in that artificial separation between means and ends. Everyone is familiar with the ethical conundrum about whether the ends justify the means, but that's garbage. That's what you say when you try to convince yourself that doing things haphazardly is ok, because what you do is just the means to whatever other thing is the end. Life is not so easily divided into things that matter and things that don't.

Your work, your creation is not just some ideal isolated end towards which you travel across a sea of dirty means, trying to keep your silver armour clean. It's one whole thing. You are creating the means, you are creating your goal, you are responsible for both, and if you use shoddy bricks, your outhouse should shame you.

In the same way, if you do crappy code, your feature is demeaned. It may even work, but you will know it's built out of crap. You will know you will have to fix and maintain that crap for years to come, or, if you are lucky, ruin your karma by dumping it on the head of some poor sucker who follows your steps.

```
system-message
```

```
WARNING/2 in everything.txt, line 3
```

```
Duplicate explicit target name: "or maybe not".
```

I am pretty much a materialist. If you remove the code, you don't have a feature, or software, you have a concept, maybe an idea, perhaps a design (or maybe not.. or maybe not: http://www.developerdotstar.com/mag/articles/reeves_design_main.html

) but certainly not software, just like you don't have a brick outhouse without piling some damn bricks one on top of the other.

I always say, when I see someone calling himself a software engineer, that I am merely a software carpenter. I know my tools, I care about them, I use them as well as I can according to my lights^[2] and I try to produce as good a piece of furniture as I can with what I am given.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "rst2pdf".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "rst2pdf".
```

This tends to produce humble software, but it's software that has one redeeming feature: it knows what it should do, and does it as well as I can make it. For example, I wrote `rst2pdf`. `rst2pdf`: <http://rst2pdf.googlecode.com>

. It's a program that takes some sort of text, and produces PDF files. It does that as well as I could manage. It does nothing else. It works well or not, but it is what it is, it has a purpose, a description and a goal, and I have tried to achieve that goal without embarrassing myself.

My programs are outhouses, made of carefully selected and considered bricks. They are not fancy, but they are what they are and you know it just by looking at them. And if you ever need an outhouse, well, an outhouse is what you should get.

Also, people tend to do weird stuff with them I never expected, but that's just the luck of the analogy.

But why did I mention malls in the title? Because malls are not outhouses. Malls are not done with a goal by themselves beyond making money for its builders. The actual function of a piece of mall is not even known when it's being built. Will this be a McDonalds, or will it be a comic book store? Who knows!

A mall is built quickly with whatever makes sense moneywise, and it should look bland and recognisable, to not scare the herd. It's a building made for pedestrians, but it's intended to confuse them and make the path from A to B as long and meandering as possible. The premises on which its design is based are all askew, corrupted and self-contradicting.

They also give builders a chance to make lots of money. Or to lose lots of money.

Nowadays, we live in an age of mall software. People build startups, get financing, build crappy software and sometimes they hit it big (Twitter, Facebook) or, more likely, fade into obscurity leaving behind nothing at all, except money lost and sad programmers who spent nights coding stuff noone will ever see or use, and not much else.

Far from me saying startups are not a noble or worthy endeavour. They are! It's just that people who work on them should realize that they are not building software. That's why code doesn't look important to them, because they are actually selling eyeballs to advertisers, or collected personal data from their users to whoever buys that, or captive public for game developers, or whatever your business model says (if you have one!).

They are building malls, where the value is not in the building, which is pretty ghastly and useless by itself, but on the people in it, those who rent space in the mall,

those who will *use* the mall, the software, the social network, whatever it is you are building.

Twitter is not software, Facebook is not software. If they were, identi.ca and diaspora would be bigger! What they are is people in one place, like a mall is not a real building, but a collection of people under a roof.

So, there is nothing wrong with building malls. Just remember that your ends and your means are one and a whole, that code is important, that without code Facebook and Twitter don't work, and that without people they are a badland, and know *what* you are doing.

Because the only hard thing in life is knowing what you want to do. The rest is the easy part. And because malls without toilets suck.

UPDATE: If you read this and all you can say is "oh, he's just embedding WebKit", I have two things to tell you:

- 1) Duh! Of course the 128 lines don't include the rendering engine, or the TCP implementation, or the GUI toolkit. This is about the rest of the browser, the part around the web rendering engine. You know, just like Arora, Rekonq, Epiphany, and everyone else that embeds webkit or mozilla does it? If you didn't get that before this explanation... *facepalm*.
- 2) Get your favourite webkit fork and try to do this much with the same amount of code. I dare you! I double dog dare you!

Now back to the original article

Today, because of a IRC chat, I tried to find a 42-line web browser I had written a while ago. Sadly, the pastebin where I posted it was dead, so I learned a lesson: It's not a good idea to trust a pastebin as code repository

What I liked about that 42-line browser was that it was not the typical example, where someone dumps a Webkit view in a window, loads a page and tries to convince you he's cool. That one is only 7 lines of code:

```
\ :kn:'import' \ \ :nn:'sys' \
\ :kn:'from' \ \ :nn:'PyQt4' \ \ :kn:'import' \ \ :n:'QtGui' \ \ :p:', ' \ \ :
\ :n:'app' \ \ :o:'=' \ \ :n:'QtGui' \ \ :o:'. ' \ \ :n:'QApplication' \ \ :p:'('
\ :n:'wb' \ \ :o:'=' \ \ :n:'QtWebKit' \ \ :o:'. ' \ \ :n:'QWebView' \ \ :p:'()'
\ :n:'wb' \ \ :o:'. ' \ \ :n:'setUrl' \ \ :p:'(' \ \ :n:'QtCore' \ \ :o:'. ' \ \ :n:
```

1

system-message

WARNING/2 in `everything.txt`, line 29989
Duplicate explicit target name: "1".

If you really want to see the whole conversation, it's here: <http://between.com/ralsina/iatrou> (if anyone knows a better conversation tracker please post it in a comment).

2

system-message

WARNING/2 in `everything.txt`, line 29991
Duplicate explicit target name: "2".

Yet here I am, an Engineering Manager at Canonical. Sorry guys!

```

\ :n: 'wb' \ \ :o: '.' \ \ :n: 'show' \ \ :p: '()' \ \
\ :n: 'sys' \ \ :o: '.' \ \ :n: 'exit' \ \ :p: '(' \ \ :n: 'app' \ \ :o: '.' \ \ :n: 'ex

```

And if I wanted to make the code uglier, it could be done in 6.

But anyway, that 42-line browser actually looked useful!

Those buttons you see actually worked correctly, enabling and disabling at the right moment, the URL entry changed when you clicked on links, and some other bits.

So, I have decided to start a small, intermittent project of code golf: put as much browser as I can in 128 lines of code (not counting comments or blanks), starting with PyQt4.

This has a useful purpose: I always suspected that if you assumed PyQt was part of the base system, most apps would fit in floppies again. This one fits on a 1.44MB floppy some 500 times (so you could use 360KB commodore floppies if you prefer!).

So far, I am at about 50 lines, and it has the following features:

- Zoom in (Ctrl++)
- Zoom out (Ctrl+-)
- Reset Zoom (Ctrl+=)
- Find (Ctrl+F)
- Hide find (Esc)
- Buttons for back/forward and reload
- URL entry that matches the page + autocomplete from history + smart entry (adds <http://>, that kind of thing)
- Plugins support (including flash)
- The window title shows the page title (without browser advertising ;-)
- Progress bar for page loading
- Statusbar that shows hovered links URL
- Takes a URL on the command line, or opens <http://python.org>
- Multiplatform (works in any place QtWebKit works)

Missing are tabs and proxy support. I expect those will take another 40 lines or so, but I think it's probably the most featureful of these toy browsers.

The code... it's not all that hard. I am using lambda a lot, and I am using PyQt's keyword arguments for signal connection which makes lines long, but not hard. It could be made much smaller!

Here it is in action:

And here's the code:

```

\ :c: '#!/usr/bin/env python' \
\ :s: '"A web browser that will never exceed 128 lines of code. (not counting
\ :kn: 'import' \ \ :nn: 'sys' \
\ :kn: 'from' \ \ :nn: 'PyQt4' \ \ :kn: 'import' \ \ :n: 'QtGui' \ \ :p: ', ' \ \

```

```

\ :k:'class' \ :nc:'MainWindow' \ :p:'(' \ :n:'QtGui' \ :o:'. ' \ :n:
  \ :k:'def' \ :nf:'__init__' \ :p:'(' \ :bp:'self' \ :p:', ' \ :n:
    \ :n:'QtGui' \ :o:'. ' \ :n:'QMainWindow' \ :o:'. ' \ :n:'__init__'
    \ :bp:'self' \ :o:'. ' \ :n:'sb' \ :o:'=' \ :bp:'self' \ :o:'. '

    \ :bp:'self' \ :o:'. ' \ :n:'pbar' \ :o:'=' \ :n:'QtGui' \ :o:'. '
    \ :bp:'self' \ :o:'. ' \ :n:'pbar' \ :o:'. ' \ :n:'setMaximumWidth'
    \ :bp:'self' \ :o:'. ' \ :n:'wb' \ :o:'=' \ :n:'QtWebKit' \ :o:'. '
    \ :bp:'self' \ :o:'. ' \ :n:'setCentralWidget' \ :p:'(' \ :bp:'s

    \ :bp:'self' \ :o:'. ' \ :n:'tb' \ :o:'=' \ :bp:'self' \ :o:'. '
    \ :k:'for' \ :n:'a' \ :ow:'in' \ :p:'(' \ :n:'QtWebKit' \ :o:'. '
      \ :bp:'self' \ :o:'. ' \ :n:'tb' \ :o:'. ' \ :n:'addAction' \ :o:'. '

    \ :bp:'self' \ :o:'. ' \ :n:'url' \ :o:'=' \ :n:'QtGui' \ :o:'. '
    \ :bp:'self' \ :o:'. ' \ :n:'tb' \ :o:'. ' \ :n:'addWidget' \ :o:'. '

    \ :bp:'self' \ :o:'. ' \ :n:'wb' \ :o:'. ' \ :n:'urlChanged' \ :o:'. '
    \ :bp:'self' \ :o:'. ' \ :n:'wb' \ :o:'. ' \ :n:'urlChanged' \ :o:'. '

    \ :bp:'self' \ :o:'. ' \ :n:'wb' \ :o:'. ' \ :n:'statusBarMessage'
    \ :bp:'self' \ :o:'. ' \ :n:'wb' \ :o:'. ' \ :n:'page' \ :p:'('

    \ :bp:'self' \ :o:'. ' \ :n:'search' \ :o:'=' \ :n:'QtGui' \ :o:'. '
    \ :bp:'self' \ :o:'. ' \ :n:'search' \ :o:'. ' \ :n:'hide' \ :p:'('
    \ :bp:'self' \ :o:'. ' \ :n:'showSearch' \ :o:'=' \ :n:'QtGui' \ :o:'. '
    \ :bp:'self' \ :o:'. ' \ :n:'hideSearch' \ :o:'=' \ :n:'QtGui' \ :o:'. '

    \ :bp:'self' \ :o:'. ' \ :n:'quit' \ :o:'=' \ :n:'QtGui' \ :o:'. '
    \ :bp:'self' \ :o:'. ' \ :n:'zoomIn' \ :o:'=' \ :n:'QtGui' \ :o:'. '
    \ :bp:'self' \ :o:'. ' \ :n:'zoomOut' \ :o:'=' \ :n:'QtGui' \ :o:'. '
    \ :bp:'self' \ :o:'. ' \ :n:'zoomOne' \ :o:'=' \ :n:'QtGui' \ :o:'. '
    \ :bp:'self' \ :o:'. ' \ :n:'wb' \ :o:'. ' \ :n:'settings' \ :p:'('

    \ :bp:'self' \ :o:'. ' \ :n:'sb' \ :o:'. ' \ :n:'addPermanentWidget'
    \ :bp:'self' \ :o:'. ' \ :n:'sb' \ :o:'. ' \ :n:'addPermanentWidget'
    \ :bp:'self' \ :o:'. ' \ :n:'wb' \ :o:'. ' \ :n:'load' \ :p:'('

\ :k:'if' \ :n:'__name__' \ :o:'==' \ :s:'"__main__"' \ :p:': '
  \ :n:'app' \ :o:'=' \ :n:'QtGui' \ :o:'. ' \ :n:'QApplication' \ :p:'('
  \ :k:'if' \ :nb:'len' \ :p:'(' \ :n:'sys' \ :o:'. ' \ :n:'argv' \ :o:'. '
    \ :n:'url' \ :o:'=' \ :n:'QtCore' \ :o:'. ' \ :n:'QUrl' \ :o:'. '
  \ :k:'else' \ :p:': '
    \ :n:'url' \ :o:'=' \ :n:'QtCore' \ :o:'. ' \ :n:'QUrl' \ :p:'('
  \ :n:'wb' \ :o:'=' \ :n:'MainWindow' \ :p:'(' \ :n:'url' \ :p:') '
  \ :n:'wb' \ :o:'. ' \ :n:'show' \ :p:'('
  \ :n:'sys' \ :o:'. ' \ :n:'exit' \ :p:'(' \ :n:'app' \ :o:'. ' \ :n:

```

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "it was just a few days ago".

It seems it was only a few days ago that I started this project. Oh, wait, yes, [it was just a few days ago!](http://lateral.netmanagers.com.ar/weblog/posts/BB948.html).. it was just a few days ago!: <http://lateral.netmanagers.com.ar/weblog/posts/BB948.html>

If you don't want to read that again, the idea is to see just how much code is needed to turn Qt's WebKit engine into a fully-fledged browser.

To do that, I set myself a completely arbitrary limit: 128 lines of code.

So, as of now, I declare it feature-complete.

The new features are:

- Tabbed browsing (you can add/remove tabs)
- Bookmarks (you can add/remove them, and choose them from a drop-down menu)

This is what already worked:

- Zoom in (Ctrl++)
- Zoom out (Ctrl+-)
- Reset Zoom (Ctrl+=)
- Find (Ctrl+F)
- Hide find (Esc)
- Buttons for back/forward and reload
- URL entry that matches the page + autocomplete from history + smart entry (adds <http://>, that kind of thing)
- Plugins support (including flash)
- The window title shows the page title (without browser advertising ;-)
- Progress bar for page loading
- Statusbar that shows hovered links URL
- Takes a URL on the command line, or opens <http://python.org>
- Multiplatform (works in any place QtWebKit works)

So... how much code was needed for this? **87 LINES OF CODE**

Or if you want the PEP8-compliant version, **115 LINES OF CODE**.

Before anyone says it: *yes, I know the rendering engine and the toolkit are huge.*

What I wrote is just the chrome around them, just like Arora, Rekonq, Galeon, Epiphany and a bunch of others do.

It's simple, minimalistic chrome, but it works pretty good, IMVHO.

Here it is in (buggy) action:

It's more or less feature-complete for what I expected to be achievable, but it still needs some fixes.

You can see the code at it's own home page: <http://devicenzo.googlecode.com>

Just saw this at [Reason](#) ... weird.

Funny, though.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "1".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "1".

I have already posted a couple of times (1.. 1: <http://lateral.netmanagers.com.ar/weblog/posts/BB948.html>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "2".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "2".

, 2.. 2: <http://lateral.netmanagers.com.ar/weblog/posts/BB949.html>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "de vicenzo".

) about [De Vicenzo](http://devicenzo.googlecode.com).. de vicenzo: <http://devicenzo.googlecode.com>

, an attempt to implement *the rest* of the browser, starting with PyQt's WebKit... limiting myself to 128 lines of code.

Of course I could do more, but I have my standards!

- No using ;
- No `if whatever: f()`

Other than that, I did a lot of dirty tricks, but right now, it's a fairly complete browser, and it has 127 lines of code (according to `sloccount`) so that's enough playing and it's time to go back to real work.

But first, let's consider how some features were implemented (I'll wrap the lines so they page stays reasonably narrow), and also look at the "normal" versions of the same (the "normal" code is not tested, please tell me if it's broken ;-).

This is not something you should learn how to do. In fact, this is almost a treatise on how *not* to do things. This is some of the least pythonic, less clear code you will see this week.

It is short, and it is expressive. But it is ugly.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "this version".

I'll discuss [this version](http://code.google.com/p/devicenzo/source/browse/trunk/devicenzo.py?spec).. this version: <http://code.google.com/p/devicenzo/source/browse/trunk/devicenzo.py?spec>

It's not terribly weird code, except for the use of the ternary operator in the last line. The use of json ensures that as long as reasonable things are persisted, you will get them with the same type as you put them without needing to convert them or call special methods.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "line 24".
```

So, how do you save/restore the cookies? First, you need to access the cookie jar. I couldn't find whether there is a global one, or a per-webview one, so I created a QNetworkCookieJar in line 24.. line 24: <http://code.google.com/p/devicenzo/source/browse/trunk/devicenzo.py?s>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "line 107".
```

and assign it to each web page in line 107.. line 107: <http://code.google.com/p/devicenzo/source/browse/trunk/de>

```
\ :c:\# Save the cookies, in the window's closeEvent\
\ :bp:\self\ \ :o:'.'\ \ :n:'put\ \ :p:'(\ \ :s:"cookiejar"\ \ :p:',\

\ :c:\# Restore the cookies, in the window's __init__\
\ :bp:\self\ \ :o:'.'\ \ :n:'cookies\ \ :o:'.'\ \ :n:'setAllCookies\ \ :
\ :k:'for\ \ :n:'c\ \ :ow:'in\ \ :bp:\self\ \ :o:'.'\ \ :n:'get\ \ :
```

Here I confess I am guilty of using list comprehensions when a for loop would have been the correct thing.

I use the same trick when restoring the open tabs, with the added misfeature of using a list comprehension and throwing away the result:

```
\ :c:\# get("tabs") is a list of URLs\
\ :p:'[\ \ :bp:\self\ \ :o:'.'\ \ :n:'addTab\ \ :p:'(\ \ :n:'QtCore\ \ :
```

Using Properties and Signals in Object Creation

This is a feature of recent PyQt versions: if you pass property names as keyword arguments when you create an object, they are assigned the value. If you pass a signal as a keyword argument, they are *connected* to the given value.

This is a really great feature that helps you create clear, local code, and it's a great thing to have. But if you are writing evil code... well, you can go to hell on a handbasket using it.

This is all over the place in De Vincenzo, and here's one example (yes, this is one line):

```
\ :n:'QtWebKit\ \ :o:'.'\ \ :n:'QWebView\ \ :o:'.'\ \ :n:'__init__'\ \ :p:
```

```

\ :p:'('\ \ :bp:'self'\ \ :o:'. '\ \ :n:'pbar'\ \ :o:'. '\ \ :n:'show'\ \ :p:
\ :bp:'None'\ \ :p:', '\ \ :n:'loadFinished'\ \ :o:'='\ \ :bp:'self'\ \ :o:
\ :bp:'self'\ \ :o:'. '\ \ :n:'pbar'\ \ :o:'. '\ \ :n:'show'\ \ :p:'()\ \ \ :
\ :n:'t'\ \ :p:': '\ \ :n:'container'\ \ :o:'. '\ \ :n:'tabs'\ \ :o:'. '\ \ :
\ :p:'('\ \ :n:'container'\ \ :o:'. '\ \ :n:'setWindowTitle'\ \ :p:'('\ \ :n:

```

Oh, boy, where do I start with this one.

There are `lambda` expressions used to define the callbacks in-place instead of just connecting to a real function or method.

There are lambdas that contain the ternary operator:

```

\ :n:'loadStarted'\ \ :o:'='\ \ :k:'lambda'\ \ :p:': '\ \ \
\ :bp:'self'\ \ :o:'. '\ \ :n:'pbar'\ \ :o:'. '\ \ :n:'show'\ \ :p:'()\ \ \

```

There are lambdas that use `or` or a tuple to trick python into doing two things in a single lambda!

```

\ :n:'loadProgress'\ \ :o:'='\ \ :k:'lambda'\ \ :n:'v'\ \ :p:': '\ \ \
\ :p:'('\ \ :bp:'self'\ \ :o:'. '\ \ :n:'pbar'\ \ :o:'. '\ \ :n:'show'\ \ :p:
\ :bp:'None'\

```

I won't even *try* to untangle this for educational purposes, but let's just say that line contains what should be replaced by 3 methods, and should be spread over 6 lines or more.

Download Manager

Ok, calling it a manager is overreaching, since you can't stop them once they start, but hey, it lets you download things and keep on browsing, and reports the progress!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "line 16".

First, on [line 16](#).. line 16: <http://code.google.com/p/devicenzo/source/browse/trunk/devicenzo.py?spec=svn52&r=5>

I created a `bars` dictionary for general bookkeeping of the downloads.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "lines 108 and 109".

Then, I needed to delegate the unsupported content to the right method, and that's done in [lines 108 and 109](#).. lines 108 and 109: <http://code.google.com/p/devicenzo/source/browse/trunk/devicenzo.py?>

What that does is basically that whenever you click on something WebKit can't handle, the method `fetch` will be called and passed the network request.

```

\ :k:'def' \ :nf:'fetch' \ :p:'(\ \ :bp:'self' \ :p:', \ \ :n:'reply'
  \ :n:'destination' \ :o:'=' \ \ :n:'QtGui' \ :o:'. \ \ :n:'QFileDialog'
    \ :s:'"Save File"' \ :p:', \ \ :n:'os' \ :o:'. \ \ :n:'path' \ \
      \ :nb:'unicode' \ \ :p:'(\ \ :n:'reply' \ \ :o:'. \ \ :n:'url' \ \
\ :k:'if' \ \ :n:'destination' \ \ :p:': \ \
  \ :n:'bar' \ \ :o:'=' \ \ :n:'QtGui' \ :o:'. \ \ :n:'QProgressBar'
    \ :n:'os' \ \ :o:'. \ \ :n:'path' \ \ :o:'. \ \ :n:'basename' \ \
  \ :bp:'self' \ \ :o:'. \ \ :n:'statusBar' \ \ :p:'()' \ \ :o:'. \ \ :n:'
  \ :n:'reply' \ \ :o:'. \ \ :n:'downloadProgress' \ \ :o:'. \ \ :n:'co
  \ :n:'reply' \ \ :o:'. \ \ :n:'finished' \ \ :o:'. \ \ :n:'connect' \ \
  \ :bp:'self' \ \ :o:'. \ \ :n:'bars' \ \ :p:'[\ \ \ :nb:'unicode' \ \ :
    \ :nb:'unicode' \ \ :p:'(\ \ \ :n:'destination' \ \ :p:')] \ \

```

No real code golfing here, except for long lines, but once you break them reasonably, this is pretty much the obvious way to do it:

- Ask for a filename
- Create a progressbar, put it in the statusbar, and connect it to the download's progress signals.

Then, of course, we need this progress slot, that updates the progressbar:

```

\ :n:'progress' \ \ :o:'=' \ \ :k:'lambda' \ \ :bp:'self' \ \ :p:', \ \ :n:'
  \ :bp:'self' \ \ :o:'. \ \ :n:'bars' \ \ :p:'[\ \ \ :nb:'unicode' \ \ :p:'
  \ :o:'. \ \ :n:'setValue' \ \ :p:'(\ \ \ :mf:'100.' \ \ :o:'*' \ \ :n:'re

```

Yes, I defined a method as a lambda to save 1 line. [facepalm]

And the finished slot for when the download is done:

```

\ :k:'def' \ \ :nf:'finished' \ \ :p:'(\ \ \ :bp:'self' \ \ :p:'): \ \
  \ :n:'reply' \ \ :o:'=' \ \ :bp:'self' \ \ :o:'. \ \ :n:'sender' \ \ :p:
  \ :n:'url' \ \ :o:'=' \ \ \ :nb:'unicode' \ \ :p:'(\ \ \ :n:'reply' \ \ :o:
  \ :n:'bar' \ \ :p:', \ \ \ :n:'_' \ \ :p:', \ \ \ :n:'fname' \ \ :o:'=' \ \
  \ :n:'redirURL' \ \ :o:'=' \ \ \ :nb:'unicode' \ \ :p:'(\ \ \ :n:'reply' \ \
    \ :n:'RedirectionTargetAttribute' \ \ :p:')' \ \ :o:'. \ \ :n:'toStri
  \ :k:'del' \ \ :bp:'self' \ \ :o:'. \ \ \ :n:'bars' \ \ :p:'[\ \ \ :n:'url'
  \ :n:'bar' \ \ :o:'. \ \ \ :n:'deleteLater' \ \ :p:'()' \ \
  \ :k:'if' \ \ \ :n:'redirURL' \ \ :ow:'and' \ \ \ :n:'redirURL' \ \ :o:'!=
    \ :k:'return' \ \ :bp:'self' \ \ :o:'. \ \ \ :n:'fetch' \ \ :p:'(\ \ \ :
  \ :k:'with' \ \ \ :nb:'open' \ \ :p:'(\ \ \ :n:'fname' \ \ :p:', \ \ \ :s:'v
    \ :n:'f' \ \ :o:'. \ \ \ :n:'write' \ \ :p:'(\ \ \ :nb:'str' \ \ :p:'(\

```

Notice that it even handles redirections sanely! Beyond that, it just hides the progress bar, saves the data, end of story. The longest line is not even my fault!

There is a big inefficiency in that the whole file is kept in memory until the end. If you download a DVD image, that's gonna sting.

Also, using `with` saves a line and doesn't leak a file handle, compared to the alternatives.

Printing

Again Qt saved me, because doing this manually would have been a pain. However, it turns out that printing is just ... there? Qt, specially when used via PyQt is such an *awesomely* rich environment.

```
\ :bp:'self'\ \ :o:'. '\ \ :n:'previewer'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :n:'paintRequested'\ \ :o:'='\ \ :bp:'self'\ \ :o:'. '\ \ :n:'print_'\ \ :bp:'self'\ \ :o:'. '\ \ :n:'do_print'\ \ :o:'='\ \ :n:'QtGui'\ \ :o:'. '\ \ :bp:'self'\ \ :p:', '\ \ :n:'activated'\ \ :o:'='\ \ :bp:'self'\ \ :o:'. '\
```

There's not even any need to golf here, that's exactly as much code as you need to hook Ctrl+p to make a `QWebView` print.

Other Tricks

There are no other tricks. All that's left is creating widgets, connecting things to one another, and enjoying the awesome experience of programming PyQt, where you can write a whole web browser (except the engine) in 127 lines of code.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "de vicenzo".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "de vicenzo".

In the spirit of the [De Vicenzo.. de vicenzo: http://devicenzo.googlecode.com](http://devicenzo.googlecode.com)

web browser, I am starting a new program, called Pato Cabrera. Here are the rules:

- Twitter client (no `identi.ca` in the first version, but to be added later)
- Has these features: <http://pastebin.lugmen.org.ar/6464>
- Has to be implemented before April 4th
- Smaller than 16384 bytes (of python code) but may be larger because of artwork.

Let's see how it works :-)

As everyone knows, there was a big quake in Japan, then a Tsunami, then a volcano erupted, then a nuclear plant caught fire. All things considered, a really crappy week.

Then again, if I were japanese and I had to read idiots telling me this was because god is punishing me because of (whatever the idiot doesn't like about Japan), I would be sorely tempted to find the morons and ... ok, considering the japanese are showing they are very reasonable people, probably just tell him something politely.

OTOH, I am not japanese. Which means I can explain in great detail why those who say "maybe it's <whatever> punishing | telling japan <something>" are a complete waste of oxygen.

I will focus on one example, because it's a very special religious moron: a presidential candidate in Argentina, called Lilita Carrió.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "what she said (spanish is the original, of course)".

Here's [what she said \(spanish is the original, of course\)](http://www.infobae.com/notas/570191-Carrio-sobre-Japon-Dios-nos-esta-diciendo-que-no-us).. what she said (spanish is the original, of course): <http://www.infobae.com/notas/570191-Carrio-sobre-Japon-Dios-nos-esta-diciendo-que-no-us>
:

"Dios nos está diciendo que debemos cuidar el planeta, que no sigamos destruyendo la tierra, que vivamos en la verdad, en la decencia, en la justicia, que no usemos la tecnología, aunque sea de manera pacífica. Hay que leer los signos de los tiempos"

"God is telling us that we should take care of the planet, that we should stop destroying the earth, that we should live in truth, in decency, in justice, and stop using technology, even if it's peacefully. We should read the sign of the times".

Let's consider that little by little.

"God is telling us that we should take care of the planet"

I must confess I am amazed that an almighty being is less capable of communicating ideas than my 3.9 year old kid. When he wants me to play ball, he brings the ball and tells me "Dad, let's play ball".

On the other hand, god apparently, to tell us to stop using technology, causes a series of catastrophic events in the other end of the world, then brings us the news over the Internet (a technological miracle), so that Lilita can divine god's intentions and then re-broadcast them to us over the radio (of course, an earlier technological miracle).

Now, does that make sense to anyone? I mean, why doesn't god just, you know, say what he means in a reasonable manner? Because for religious people, the fun is in the divination. They are acting like roman priests divining the future in the entrails of an animal, except they are using the life and suffering of people.

Oh, look, suffering in Japan, that means we should stop using the Wii!

Not only is that approach completely against everything christian doctrine teaches, from with the virtue of charity (if god did it to tell us something, *by definition* they deserved it!) to the injunction against divining god's messages in portents (yes, it is forbidden, go ask a priest).

"[God is telling us] that we should stop destroying the earth"

Oh, gee, ok then! OTOH, maybe a more subtle way than *half-breaking everything in a whole country* to let us know next time? Please?

"[God is telling us] that we should live in truth"

Ok, yes, let's do that. I will start by not believing in god, who truly does not exist. When you catch up to that we'll argue some more, ok?

"[God is telling us to] stop using technology, even if it's peacefully."

I would love if this presidential candidate didn't use technology because it would mean I would not have to see her sanctimonious stupidity ever again. OTOH, if we wouldn't have technology, we would probably not know about the earthquake yet. I suppose she may have been saying "nuclear technology" and this is out of context.

OTOH, number of people killed by peaceful nuclear technology since 1950: 1000?
10000?

number of people killed by earthquakes and tsunami in the last 5 years: 100000?
200000?

Yes, those are numbers I just made up, but I am betting they are more right than wrong, so, basically, god has killed more people this week telling us not to use nuclear power, than nuclear power has killed in the last 50 years. Not exactly good communication skills.

”We should read the sign of the times“

Ok, here it is:

Don't vote for this blithering idiot. She's dangerous, and probably mentally ill.

This is only the first part of a project to create the simplest (for me) software forum possible.

Here are the features I want:

- Login using twitter / Facebook / Google / OpenID
- Unlimited number of threads
- Support for like / dislike both on threads and on posts
- Avatars
- HTML in comments
- Mail the user on replies
- RSS feeds for threads

You can see it in action at <http://foro.netmanagers.com.ar> (for a limited time only ;-)

And here is the code:

```
\ :kn:'import' \ \ :nn:'bottle' \
\ :kn:'import' \ \ :nn:'disqusapi' \ \ :kn:'as' \ \ :nn:'disqus' \
\ :kn:'import' \ \ :nn:'json' \
\ :n:'shortname' \ \ :o:'=' \ \ :s:''magicmysteryforum' \
\ :n:'api' \ \ :o:'=' \ \ :n:'disqus' \ \ :o:'. ' \ \ :n:'DisqusAPI' \ \ :p:'(' \
\ :nd:'@bottle.route' \ \ :p:'(' \ \ :s:''/' \ \ :p:', ' \ \ :n:'method' \ \ :o:
\ :k:'def' \ \ :nf:'index' \ \ :p:'():' \
\ :n:'msg' \ \ :o:'=' \ \ :n:'bottle' \ \ :o:'. ' \ \ :n:'request' \ \ :o:
\ :n:'threads' \ \ :o:'=' \ \ :n:'api' \ \ :o:'. ' \ \ :n:'forums' \ \ :o:
\ :k:'print' \ \ :n:'threads' \ \ :p:'[' \ \ :mi:'0' \ \ :p:']' \
\ :k:'return' \ \ :n:'bottle' \ \ :o:'. ' \ \ :n:'template' \ \ :p:'(' \ \ :
\ :nd:'@bottle.route' \ \ :p:'(' \ \ :s:''/new' \ \ :p:', ' \ \ :n:'method' \ \ :o:
\ :k:'def' \ \ :nf:'new' \ \ :p:'():' \
\ :n:'title' \ \ :o:'=' \ \ :n:'bottle' \ \ :o:'. ' \ \ :n:'request' \ \ :o:
\ :k:'if' \ \ :ow:'not' \ \ :n:'title' \ \ :p:': ' \
\ :n:'bottle' \ \ :o:'. ' \ \ :n:'redirect' \ \ :p:'(' \ \ :s:''/?msg=M
\ :k:'return' \
\ :n:'thread' \ \ :o:'=' \ \ :n:'api' \ \ :o:'. ' \ \ :n:'threads' \ \ :o:
```

```

\ :n:'thread_id'\ \ :o:'='\ \ :n:'thread'\ \ :o:'.'\ \ :n:'__dict__'\
\ :c:'# Redirecting to /thread/thread_id doesn't work'\
\ :c:'# because threads take a few seconds to appear on the listing'\
\ :n:'bottle'\ \ :o:'.'\ \ :n:'redirect'\ \ :p:'('\ \ :s:''/'\ \ :p:')\

\ :nd:'@bottle.route'\ \ :p:'('\ \ :s:''/thread/:id'\ \ :p:')'\
\ :k:'def'\ \ :nf:'thread'\ \ :p:'('\ \ :nb:'id'\ \ :p:'):\
\ :n:'t'\ \ :o:'='\ \ :n:'api'\ \ :o:'.'\ \ :n:'threads'\ \ :o:'.'\
\ :k:'return'\ \ :n:'bottle'\ \ :o:'.'\ \ :n:'template'\ \ :p:'('\ \ :

\ :nd:'@bottle.route'\ \ :p:'('\ \ :s:''/static/:path#.#'\ \ :p:')'\
\ :k:'def'\ \ :nf:'server_static'\ \ :p:'('\ \ :n:'path'\ \ :p:'):\
\ :k:'return'\ \ :n:'bottle'\ \ :o:'.'\ \ :n:'static_file'\ \ :p:'('\ \ :

\ :n:'app'\ \ :o:'='\ \ :n:'bottle'\ \ :o:'.'\ \ :n:'app'\ \ :p:'()\ \
\ :n:'app'\ \ :o:'.'\ \ :n:'catchall'\ \ :o:'='\ \ :bp:'False'\ \ :c:'#M
\ :n:'bottle'\ \ :o:'.'\ \ :n:'run'\ \ :p:'('\ \ :n:'host'\ \ :o:'='\ \ :s:

```

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "bottle".

It requires [Bottle.. bottle: http://bottlepy.org/docs/dev/index.html](http://bottlepy.org/docs/dev/index.html)

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "the disqus python api".

and the [Disqus python API.. the disqus python api: https://github.com/disqus/disqus-python](https://github.com/disqus/disqus-python)

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "main.tpl".

Of course, there is also a bit of templating involved, here is [main.tpl.. main.tpl: http://code.google.com/p/magicforum/source/browse/trunk/views/main.tpl](http://code.google.com/p/magicforum/source/browse/trunk/views/main.tpl)

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "thread.tpl".

and the [thread.tpl.. thread.tpl: http://code.google.com/p/magicforum/source/browse/trunk/views/thread.tpl](http://code.google.com/p/magicforum/source/browse/trunk/views/thread.tpl)

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "bluetrip css".

. Since I suck at HTML, it uses [Bluetrip CSS.. bluetrip css: http://bluetrip.org/](http://bluetrip.org/)

and it's more than simple enough to customize.

OF COURSE I AM CHEATING!

This thing is just a simple veneer around Disqus! More like a blog with comments and without posts than a forum! But ... what's missing to make this a real forum? It works, doesn't it? You could even use Disqus categories to create subforums...

All things considered, I think it's a cute hack.

And if you wait a few days, this will lead to something *much* more magical!

Full source code at <http://magicforum.googlecode.com>

I have not posted in this blog in a long time. One of the reasons is that a whole lot of things have happened in my life, and I have not blogged about them.

It has gotten so that if I were to "catchup" here, I would need to post a 35-page post with 300 pictures, 100 code snippets, 50 rants and maybe 3 youtube videos.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "all a dream".

So... screw that and let's pretend that was [all a dream](http://www.ultimatedallas.com/episodeguide/dre).. all a dream: <http://www.ultimatedallas.com/episodeguide/dre>

and that you (if there are still any "you" out there) have not been neglected, and all is good.

So, what am I posting about? Oh, just a short rant.

I am saddened and pissed off by people who proudly say "I don't know anything about X", where X is math, knitting, astrophysics, football, medicine, pet care, hair-dressing, coffee brewing, politics, or dolphin-rearing (this doesn't apply to wine. It's perfectly ok not to care about wine, because most people who do care about wine are faking it).

There are tons of things I know nothing about. And you know what? Each one is a personal failure. Every time I see my mom knitting, and I simply cannot grasp how the hell it works, it pisses me off.

While it looks mind-crushingly boring, she's basically taking a string, make some knots using sticks, and producing a freaking sweater. How *cool* is that? I say pretty damn cool.

Whenever I see something that looks boring, or uninteresting, I always think, there is, somewhere, a person who has spent years of his life being the best at that. There is, somewhere, a person which is the world's authority on how to tie a string to make a sweater.

There is someone who is the non-plus-ultra of laying out coloured tiles so it looks good in the shape of a dolphin.

There is someone who is the best at using an incredibly inaccurate spraycan to draw on a wall.

There is someone who can look at a lot full of cars and tell you the logical order to remove them so they leave in minimum time.

There is someone who can hit a small ball, flying at 150km/h with a round stick, 1 out of every 3 times.

There is someone who can guess what word goes with almost every short definition.

There is someone who can tell you all regular spanish verbs that start with T.

There is someone who knows all about the history of journalism in Turkey (I actually walked in front of the Museum of Turkish Journalism, and it *pains* me that I couldn't go at the time).

There is someone who knows the dialog for every simpsons episode.

Every one of those things, and billions more, may be boring to me, but that is surely because *I* am failing at noticing the fun in it, it's a roughness of my soul, a failure of my perception, a flaw in my vision, which cripples me into saying they are all boring.

Boredom is not a sign of superiority, it's a sign of inferiority. Next time you see a bored person, someone that has nothing he wants to do, who looks condescendingly at those who are passionate about something, no matter what that may be, feel sad for him.

I have written about this in the past, with the general conclusion being "it's a pain in the ass".

So, now, here is how it's done.

1)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "devicenzo.py".

Start with a working PyQt application. In this example, I will use `devicenzo.py`.
`devicenzo.py`: <http://code.google.com/p/devicenzo/source/browse/trunk/devicenzo.py>

mostly because:

- a) It is a working PyQt application.
 - b) It uses a big chunk of PyQt
 - c) It's easy to test
- 2) Now you need a `setup.py`. Here's one that works, with extensive comments.

```
\ :c:`# We will be using py2exe to build the binaries.`\
\ :c:`# You may use other tools, but I know this one.`\

\ :kn:`from` \ :nn:`distutils.core` \ :kn:`import` \ :n:`setup` \
\ :kn:`import` \ :nn:`py2exe`

\ :c:`# Now you need to pass arguments to setup` \
\ :c:`# windows is a list of scripts that have their own UI and` \
\ :c:`# thus don't need to run in a console.` \

\ :n:`setup` \ :p:`(` \ :n:`windows` \ :o:`=` \ :p:`[` \ :s:`'devicenzo` \
\ :n:`options` \ :o:`=` \ :p:`{` \

\ :c:`# And now, configure py2exe by passing more options;` \
\ :s:`'py2exe'` \ :p:`:` \ :p:`{` \
```

```
\ :c:`# This is magic: if you don't add these, your .exe may`
\ :c:`# or may not work on older/newer versions of windows.`
```

```
    \ :s:`"dll_excludes"`` \ :p:`:``` \ :p:`[``
      \ :s:`"MSVCP90.dll"`` \ :p:``,```
      \ :s:`"MSWSOCK.dll"`` \ :p:``,```
      \ :s:`"mswsock.dll"`` \ :p:``,```
      \ :s:`"powrprof.dll"`` \ :p:``,```
      \ :p:``,``,```
```

```
\ :c:`# Py2exe will not figure out that you need these on its own.`
\ :c:`# You may need one, the other, or both.`
```

```
    \ :s:`'includes'`` \ :p:`:``` \ :p:`[``
      \ :s:`'sip'`` \ :p:``,```
      \ :s:`'PyQt4.QtNetwork'`` \ :p:``,```
      \ :p:``,``,```
```

```
\ :c:`# Optional: make one big exe with everything in it, or`
\ :c:`# a folder with many things in it. Your choice`
```

```
\ :c:`#           'bundle_files': 1,`
      \ :p:`)```
      \ :p:``,``,```
```

```
\ :c:`# Qt's dynamically loaded plugins and py2exe really don't`
\ :c:`# get along.`
```

```
\ :n:`data_files` \ :o:`='` \ :p:`[``
      \ :p:`(` \ :s:`'phonon_backend'`` \ :p:``,``` \ :p:`[``
        \ :s:`'C:\Python27\Lib\site-packages\PyQt4\plugins\phonon_b`
        \ :p:`)``,``,```
      \ :p:`(` \ :s:`'imageplugins'`` \ :p:``,``` \ :p:`[``
        \ :s:`'c:\Python27\lib\site-packages\PyQt4\plugins\imageformats`
        \ :s:`'c:\Python27\lib\site-packages\PyQt4\plugins\imageformats`
        \ :s:`'c:\Python27\lib\site-packages\PyQt4\plugins\imageformats`
        \ :p:`)``,``,```
      \ :p:`)``,``,```
```

```
\ :c:`# If you choose the bundle above, you may want to use this, too.`
\ :c:`#     zipfile=None,`
\ :p:`)```
```

- 1) Run `python setup.py py2exe` and get a dist folder full of binary goodness.

And that's it. Except of course, that's *not* it.

What this will do is create a binary set, either a folder full of things, or a single EXE file. And that's not enough. You have to consider at least the following:

- a) Put everything in resource files: images, qss files, icons, etc. Every file your app needs? Put it in a resource file and load it from there. That way

you don't have to care about them if you go the "one exe" road.

- b) Compile .ui files to .py (same reason)
- c) Figure out if you use Qt's plugins, and make them work. This includes:
 - using Phonon, using QtSQL, and using any image formats other than PNG.

After you have that, are you done? NO!

Your windows user will want an installer. I am not going to go into details, but I had a good time using BitRock's InstallBuilder for Qt. It's a nice tool, and it works. That's a lot in this field.

But is that all? NO!

You have to take care of the Visual Studio Runtime. My suggestion? Get a copy of the 1.1MB vc8redist_x86.exe (not the larger one, the 1.1MB one), and either tell people to install it manually, or add it to your installer. You are legally allowed (AFAIK) to redistribute that thing as a whole. But not what's in it (unless you have a VS license).

And we are done? NO!

Once you run your app "installed", if it ever prints anything to stderr, you will get either a dialog telling you it did, or worse (if you are in anything newer than XP), a dialog telling you it can't write to a log file, and the app will never work again.

This is because py2exe catches stderr and tries to save it on a logfile. Which it tries to create in the same folder as the binary. Which is usually not allowed because of permissions.

Solution? Your app should never write to stderr. Write an excepthook and catch that. And then remove stderr or replace it with a log file, or something. Just don't let py2exe do it, because the way py2exe does it is broken.

And is that it?

Well, basically yes. Of course you should get 4 or 5 different versions of windows to test it on, but you are pretty much free to ship your app as you wish. Oh, mind you, don't upload it to downloads.com because they will wrap your installer in a larger one that installs bloatware and crap.

So, there you go.

For the english readers: Sorry, spanish-only post!

En Argentina hay Internet. En la Internet hay un sitio que se llama Taringa. En Taringa hay links a libros escaneados.

Y no libros Creative Commons, ni de dominio público, libros publicados hace un rato, libros con autores que esperarían ganar algún mango con eso.

Y en Argentina hay leyes. Y una de esas leyes es la ley 11723 de propiedad intelectual. Y esa ley prohíbe copiar libros y repartir las copias, aunque no cobres por ello.

Entonces, en Argentina, hay una Taringa que tiene a sus dueños procesados por violar la ley 11723.

¿Está bien violar la ley? En general no. ¿En particular sí? Bueno, a veces, violar alguna ley con la que uno está en desacuerdo, puede estar bien. Por ejemplo, violar las leyes que segregaban a los negros en USA, estaba bien. Eso se llama desobediencia civil.

¿Entonces, lo que hacen en Taringa es desobediencia civil? Y que se yo. Ni idea. El tema es que uno no va y hace desobediencia civil esperando salir impune.

Uno lo haría porque *quiere* que lo pesquen, lo hace de forma pública y visible, de manera que al aplicársele el castigo, la represión o lo que fuera, se envíe un mensaje a la sociedad, para provocar el reemplazo de la norma injusta con una mejor, pero respetando el proceso democrático.

Veo muchos que se llenan la boca con "propagar la cultura" y con "el derecho inalienable a leer" y con "el antinatural e injusto monopolio del copyright". Mi respuesta para ellos es: Pfffffft. Con ruidito a pedo. Si pudiera salpicar un poquito de baba a través de la internet en su dirección mientras hago Pfffffft, lo haría.

¿Saben cómo se crea y se comparte cultura libre, nenes? Creando y compartiendo cultura, nenes. Si querés que haya libros libres, escribí un libro libre. Si querés que haya música libre, componé música libre. Si querés que se copie arte, hacé arte y ofrecé el derecho a copia.

Si tu idea de propagar cultura es poner un PDF de una revista "de informática" en rapidshare, sos un gil. Sos un gil porque 1) el 99% de las revistas de informática son una garompa en forma de revista, y 2) porque te creés un vivo. Y todos los que nos creemos vivos somos giles.

Al hacer eso, creás la sensación de que la revista es valiosa y especial cuando en realidad es un recalentado de cosas que ya pasaron hace un mes o dos. Si te creés un vivo por eso, sos un gil. No hay nada ahí que no esté afuera. Si hubiera algo ahí que no está afuera, entonces sabés qué? El tipo que lo puso ahí adentro se merece que le garpen.

Sí, ya sé, el modelo de negocio en base a la venta de copias es naif e injusto. Que la protección del copyright fomenta la creación es una excusa y un mito. Pfffffft. El solo hecho de que Disney sea tan grande que puede mover las leyes de EEUU para que Mickey nunca sea de dominio público debería darte la pista de que la protección del copyright ha sido buena para Disney. Y podrá gustarte o no lo que hacen (acordate: Pixar es parte de Disney), pero no podés decir que no crean cosas. En cualquier caso, podrás quejarte de que la protección del copyright funcionó **demasiado bien** y hay que debilitarla. Pero no podés decir que no fomentó la creación, por lo menos no sin mentir descaradamente.

Si realmente creés que repartir las cosas gratis es un mejor modelo de negocios que esa ingenuidad injusta de cobrar por las copias, DALE. Si empezás *vos*, **ahora** a aplicar ese modelo de negocios, mientras los demás están todavía con la perimida idea de cobrarle a la gente por lo que hacen... ¡Tenés una ventaja competitiva!

Ahora bien, si tu modelo de negocio es, en cambio, tomar lo que escribieron/cantaron/programaron otros, copiarlo y venderlo sin su permiso... bueno, digamos que no, no tenés ventajas competitivas porque eso lo puede hacer cualquier nabo que cree que va a salir impune.

Si en cambio creés en que está bueno compartir lo que hacés aunque no deje guita, entonces también es un buen momento para hacerlo.

Si realmente creés que la cultura libre es superior, usala y consumila en vez de consumir y usar la que no es libre. Si uno cree en la cultura libre, *no piratea*.

Si te llenás la boca hablando de cultura libre y después vas y te copiás el CD de Britney... sos un hipócrita que racionaliza para justificarse sus acciones ante sí mismo y sus pares. Te imaginás una opresión para sentirte un rebelde. Te hacés el piojoso para que te rasquen. Y no te lo respeto un corno.

So, this is 2012. Looks a lot like the previous model. On the other hand, there are going to be some changes around here. For starters, I am going to post. I am going to post every day in 2012. In this blog. Here. In 2012. 366 posts. At least.

Of course, this is wishful thinking, it's just a new year's resolution, and those are, like everything, doomed to fail, to corrupt themselves, to fall prey to rationalization,

latest victims of entropy and decay. Then again, so am I, and so are you, and it makes no sense to dwell in the sure failure to come, and instead I should focus on the current victory. This is the first day of 2012 and I am, indeed, posting today.

So, just to contradict myself, which I will do every time it may be more fun than being consistent with that moron I was in the past, let's talk about my past year.

Professionally, 2011 was a really big change, in that I stopped working on my own company and started (ok, that was in Dec 2009) working for Canonical. It has been really great in many ways, and not so great in others.

Great: I work with people I really like and respect. The work itself is full of interesting challenges.

Not Great: I have completely abandoned my free software projects, which are bitrotting.

Great: I have had the chance to visit three continents, and have awesome experiences.

Not Great: I have heard my kid cry on the phone about how he misses me. This year I may travel a bit less.

Great: Financial security for me and my family.

Not Great: If I had to grade my performance in a 1-10 scale, I *may* give myself a 6. I have lacked energy, and focus. I have been disorganized and lazy. I will try to improve.

Great: we shipped product.

Not Great: we could have shipped better product. I will do my part to make it better in 2012.

What else has been going on in my life in 2011? Well, my health is worse. I am fatter. I have been diagnosed with a hormonal imbalance which may account for some/most/all of it, and I will be starting treatment this month. This may also be to blame for part of my hypertension, and for my abysmal level of energy the last couple of years.

I have put my marriage under a big strain, which I hope I can revert in 2012, since I really love my wife, and want to make her happy.

I *will* get my mouth fixed this year. If you know me personally you may know what that means. I have awful teeth. I have never fixed them because of a perhaps understandable fear of having people put sharp things in my mouth. But I want to be able to smile to my family, so I will get it fixed. (BTW: you have not seen a honest, unthinking smile from me in about 15 years. Sucks, I know.)

How will I be able to do all these things? I don't know. I know I may not. I suspect I will not. I expect I will not. I will not. But I will try. I will do what I can, and take the fall for what I can't.

But right now, I am focusing on little bits. I am posting in this blog. 1 day down, 365 to go.

Almost noone likes to be tied down. That is surely a non-controversial statement. Of course, some people disagree (you freaks). But let's ignore them (freaks) for a few minutes, and consider the concept of tying and ties.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "what is or isn't a knot".

I have spent many an unpleasant minute trying to explain to laymen why there is a branch of mathematics that has a formal definition of [what is or isn't a knot](#).. what is

or isn't a knot: http://en.wikipedia.org/wiki/Knot_theory

. Usually that is met by the usual eye-rolling and comments of "you stupid math people" and "it's obvious" (but it isn't). If you don't know what a knot is, and what is a knot, then you don't know whether you are tied to something or not.

How can you know anything if you don't know what things are attached to you, and what you are attached to? Literally, you can't know yourself without knowing about knots, and how they bind you. Even eastern primitives who didn't know about the germ theory of disease recognized the importance of this, and preached the need to detach yourself, to cut the knots tying you to the material world.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "big city".

Consider the seminal 80's sitcom, Family Ties. It's all about the attachment between a boy and his money. And then the boy grows up, becomes a writer, goes to the big city.. big city: <http://www.imdb.com/title/tt0094799/>

and gets drunk and does a lot of drugs, but it is also about how he feels attached to his family... and about how the kid wears ties.

Is it a coincidence? I say it's impossible. Why is the clearest sign, the obvious indicator of belonging to grownup society a piece of string that you tie? Because otherwise it would fall off your neck? Yes. But sometimes an obvious symbolism is just an obvious symbolism. You tie your tie, and you tie yourself.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "16th century peasants".

Why don't women use ties? Well, because they have not yet reached that epoch in clothing. Ties are derived from Croatian Cravats and date all the way back to the 17th century. but women mostly still dress like [16th century peasants](http://www.abcgallery.com/R/rubens/rubens77.html).. 16th century peasants: <http://www.abcgallery.com/R/rubens/rubens77.html>

. If you look in renaissance pictures, all men have purses. Later, none of them do, because they have a great modern invention, called pockets. For some reason, women are stuck in the pre-pocket age of clothing.

Here's a simple exercise for heterosexually married people: count the pockets on women's garments and on men's garments. I (male) rarely can be found wearing fewer than 6 pockets. If I were to dress in modern clothes (suit), I would have over 10 pockets. Most women's wear (except jeans) has no pockets. Or (jeans) it's so ridiculously tight that you can't *use* the pockets. So women use purses. But if you use a purse, you are basically giving up a hand a certain part of the time. Would you give up a hand 10% of the time to make a fashion statement? I say if women were to make that decision consciously they wouldn't do it.

So why is that? Beats me. So, while men use ropes tied to their necks to show adherence to society, women are partly lamed for the same reason (not to mention the idea of buying shoes that are pretty but hurt a bit when walking). Goes to show that not all ties are visible. Following these conventions are invisible or only metaphorically visible ties. The later stoics (much smarter people than the previously mentioned eastern primitives: they had indoor plumbing) said things like this:

”Things not in our power include the body, property, reputation, office, and, in a word, everything which is not our doing. Things in our power are by nature free, unhindered, untrammelled; things not in our power are weak, servile, subject to hindrance, dependent on others.“

Now, how much fun is it to say that everything you own is *not* in your power. That the very power you possess is not in your power. But they also said that you had two things that were the most important: the will to get and the will to avoid. Those are the basic tools of human existence. If you have the will to avoid pain, you can not buy painful shoes. If you have the will to get pockets, you get pants with pockets. And you can decide what you want to get or avoid by just thinking. There is no secret.

Every once in a while, something awful happens. And then someone will say ”the people who do that should get the death penalty“. Invariably, that is a stupid argument.

Let’s start by disposing of the obvious low blow: ”if it happened to your son, you would ask for their death too“. To which the obvious answer is, of course I would. And I would be pushing for a stupid solution to the problem.

If something really, really bad happened to my son, I would be suffering, and in a state of violent emotion and distress. If I told you to jump off a bridge, would you? I guess not, because you would notice I am at the edge of madness and just told you that because I am feeling that way.

Well, the same is true about a victim’s relatives (or the victim himself) asking for a specific punishment: it’s a request born from pain and suffering. And as a society, we should not decide our actions based on the proposals of the ones almost insane with grief. Because we want to take the action that is better for society, not for a specific member of it. So, forget about that appeal to sentiment, because while completely understandable, it is stupid, because those grieving are allowed to be stupid.

Another failed argument for the death penalty is that it discourages future crime. This has been shown not to be the case, because similar countries or states with and without death penalty show no significant difference on crime rates.

I suspect this is because in many cases the criminal doesn’t intend to actually commit the part of the crime that brings the death sentence (murder when robbing? The criminal wanted to rob, not to murder) or he just would do it anyway (it’s not as if child rapists expect to have a nice time when captured. They just do it assuming they won’t get caught, or decide it’s worth it anyway).

And of course, the main reason why the death penalty is a bad idea is that it would be applied by public employees. Do you trust the government to decide how much to charge you in taxes? No, you do it yourself, with an accountant, and tell *them* how much to take. Do you expect the police to find a wallet you dropped on the street? Do you trust the government’s mail with really important packages? Are you confident about the government figuring out all environmental issues? Well, then why would you trust them to kill anyone they decide is a really, really bad guy?

The government is a great idea for things noone else wants to do (highways), or things noone else should be doing (law enforcement) but even in those cases you must allow for them providing a crappy service, and carefully use your input to limit what they can do.

The alternative is to assume that the government has a huge capability and competence but has just decided *not to show it*, which is perhaps the most amazing part of most conspiracy theories.

The US government can’t kill Castro, but can kill Kennedy. A government can’t keep the trains running, but can fake a moon landing. And so on, and so forth.

I once read a book where they described an interesting science: Conspiracy Theory Theory. It's the science of studying what conspiracy theories people believe in, to get insight into what they actually believe. If you are in favour of the death penalty, then you believe law enforcement doesn't make mistakes, or that killing a few random people every now and then (because they make mistakes) is not a problem. If the first, you are stupid, if the second, you are evil.

If it wasn't so pathetic, it may be funnier.

A while ago, a glacier collapsed. This is a periodical event, and a huge tourist attraction. Huge slabs of ice crashing down, and you can watch it from a safe distance.

Fun!

Now, here comes the reporting.

The [BBC](#):

A massive pond builds up behind the wall of snow before getting too heavy for the ice to hold and smashing down into the sea below.

Hmmm... well, that's a **lake**. But what the heck, the ocean is only a few hundred kilometers away. Over the Andes, crossing Chile.

[WISTV](#) (whatever that is):

No one hurt when portion of Argentinean glacier collapses

Noone has ever been hurt by this. Everyone knew it was collapsing, to be hurt you would have to get on a boat and cross a very very cold lake, too. And be the stupidest man on earth, since there were huge chunks of ice falling every few minutes since two days ago.

Hell, there's park rangers and it's forbidden to approach the glacier from the water!

Oh, and it's not an ice shelf.

But anyway, if you want to see at least the video, it's **really** cool :-). It's in this page, but I can't link it: [VIDEO](#)

One of the most vivid memories of my late childhood was when my father finally let me to stay at his table in the Gran Doria café, when it was still located in the dark bowels of a galería in Santa Fe's San Martín street.

I was maybe 12, and I had seen him sit there with a cortado while my mother went shopping with us, or while I went to one of those things kids go to (artistic expression classes? Puppetry workshop?) and it was such a mystery. It was like a three hour hole in my dad's life of which I had no information.

What would he do there? Who did he talk to? Did he read something? And always there, at the table when I came back was an empty small cortado cup.

I suspect that's when I started liking the idea of coffee. I was, of course, an inveterate hot chocolate drinker (El Quillá brand, unknown beyond that city, yet superior in my mind to any others), after a long, long time of drinking warm sweetened milk. And I know I had tried coffee before and hated it, but of course, sitting there, I said "un cortado, por favor". And boy was that thing awful. I did not drink coffee again for twenty years.

I did learn to like tea, or at least tea with milk, and learned, in college, to drink mate like a sponge. Bitter and strong as hell, the closest caffeine delivery mechanism to an IV drip, slow, weak and constant over hours. You have not really been awake until it's 5 AM, you are on your third thermos, and it feels like 2PM. It's like the wrong kind of pill in the Matrix.

But then I moved to Buenos Aires and I was alone. And drinking mate alone is like drinking Vodka alone, depressing and dirty, so I started going to cafés and ordering

lágrimas. A lágrima es like a backwards cortado. If you get a big cup and put a lágrima and a cortado in it you will get a decent café con leche. It's a pathetic beverage, only fit for the emotional wreck I was at the time.

But it's a gateway drink. And by 2002 I was drinking cortados. And by 2006 I had my own espresso machine and was some sort of caffeine Keith Richards, doing maybe 10 strong cups a day, buying expensive blends... and then I had to stop.

On January 1st 2008 I woke up at 4AM with intense chest pain. I thought I was having a heart attack. I walked to the hospital and it turned out to be gastritis. This happened again. And again. Not often, but once every year, then every six months, then every month, then four days in a row. And I had to give up coffee.

It was hell. I was asleep all day and awake all night, not having my crutch to modulate my sleep. I was grouchy, and annoying. I cheated. But then I stopped.

Sorry dad.

The title is a quote from Jurassic Park, if you were wondering. A girl, chased by dinosaurs, runs into a computer, notices it's Unix, says "That's Unix! I know Unix!"; and proceeds to hack something or other that lets her escape the previously mentioned dinosaurs.

Well, that's how life often feels for me, except with Math. And without dinosaurs. I am surrounded by trouble, something is completely broken in a way I can't quite get, things feel just slightly weird... and suddenly... that's math! I know math! (of course I *don't* know math, noone knows math. We all just know *some* math).

I may see an ad and notice the huge discount is just a cleverly disguised tiny discount. A snippet of news may reveal itself as complete nonsense after a cursory analysis. A forwarded mail may become slightly more annoying because of its obvious stupidity. Because of math.

Math is the tool I have to make sense of things. When the world gets confusing and scary, if it can be expressed in numbers, it calms me down. Afraid of the future? Let's consider the wonders of compound interest. Scared of death? Well, look at probabilities. Needing a little adventure? Well, there is game theory! Need a conversation starter? You can mention you once spent time *in class* figuring out the homeomorphism that explains the scene in Flashdance where the welder removes her bra through her sleeve.

Math tells me the world is not perfect. Math tells me the numbers I see on my computer are not really real numbers, and can't be trusted. Math tells me I can't cross all the bridges in Königsberg only once. Math tells me the number one is the most frequent number in today's paper. Math tells me Plato was right and there is a universe of pure ideas, and this is but a reflection.

On the other hand, Math tells me I can cut an orange into a few pieces, which reassemble into two, identical, oranges. So take what she says with a grain of salt.

I started this book with high hopes. After all, this was a book by a new-ish author that had won Hugo Award for Best Novel (2010), Nebula Award for Best Novel (2009), Locus Award for Best First Novel (2010), John W. Campbell Memorial Award (2010), Compton Crook Award (2010) Compton Crook Award (2010), Goodreads Choice Award Nominee for Science Fiction (2009). Impressive, uh?

Well, I don't know what they saw in it. Shallow characters, pervasive fatalism disguised as depth, improbable slang just because it sounds cool (windup girl? Really? That's how they will call genetically engineered humans in the future? Windups? Yeah, right).

Oh, and it's full of orientalism. And of things like implanting gene dogs into windups to make them loyal and subservient. And ghosts. Oh, crap, almost nothing in

this novel worked for me. Most pages had me slapping myself on the forehead.

I am not against a solid dose of weird. I like weird. I read Miéville, for crap's sake. But this is not weird, it's contrived.

And don't get me started on the end. The end is such an obvious analogy to Eden it almost made me puke. It's like the end of Battlestar Galactica, but set on the future. And with ladyboys.

I had a better time reading H. Rider Haggard's "Allan Quatermain", which I saw mentioned in the awesome series "The Victorian Hugos". Old and dated? Sure. But I would vote for it instead of "The Windup Girl" every day, and twice if they let me.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "this thread".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "this thread".

This comes from [this thread](http://comments.gmane.org/gmane.org.user-groups.python.argentina/516).. this thread: <http://comments.gmane.org/gmane.org.user-groups.python.argentina/516>

in the Python Argentina mailing list (which I strongly recommend if you read spanish).

I was the other day trying to do shell-script-like-things on python (as part of a monster setup.py) and I was annoyed that in shell it's easy to do this:

Or this:

Or this:

And on Python I had to do this, which is verbose and ugly:

```
\ :n:'cwd'\ \ :o:'='\ \ :n:'os'\ \ :o:'. '\ \ :n:'getcwd'\ \ :p:'() '\
\ :k:'try'\ \ :p:': '\
  \ :n:'os'\ \ :o:'. '\ \ :n:'chdir'\ \ :p:'('\ \ :s:'foo'\ \ :p:') '\
  \ :n:'os'\ \ :o:'. '\ \ :n:'system'\ \ :p:'('\ \ :s:'bar -baz'\ \ :p:
\ :k:'finally'\ \ :p:': '\
  \ :n:'os'\ \ :o:'. '\ \ :n:'chdir'\ \ :p:'('\ \ :n:'cwd'\ \ :p:') '\
```

This is what I wanted to have:

```
\ :k:'with'\ \ :n:'os'\ \ :o:'. '\ \ :n:'chdir'\ \ :p:'('\ \ :s:'foo'\ \
  \ :n:'os'\ \ :o:'. '\ \ :n:'system'\ \ :p:'('\ \ :s:'bar -baz'\ \ :p:
```

And of course, you can't do that. So, I asked, how do you implement that context manager? I got several answers.

1. That's available in Fabric:

```
\ :k:'with'\ \ :n:'cd'\ \ :p:'('\ \ :s:'foo'\ \ :p:') '\
  \ :n:'run'\ \ :p:'('\ \ :s:'bar'\ \ :p:') '\
```

2. It's not hard to do:

```
\ :k:'class'\ \ :nc:'DirContextM'\ \ :p:'('\ \ :nb:'object'\ \ :p:'\
\ :k:'def'\ \ :nf:'__init__'\ \ :p:'('\ \ :bp:'self'\ \ :p:','\
\ :bp:'self'\ \ :o:'. '\ \ :n:'new_dir'\ \ :o:'='\ \ \ :n:'ne
\ :bp:'self'\ \ :o:'. '\ \ :n:'old_dir'\ \ :o:'='\ \ \ :bp:'N

\ :k:'def'\ \ :nf:'__enter__'\ \ :p:'('\ \ :bp:'self'\ \ :p:'):\
\ :bp:'self'\ \ :o:'. '\ \ :n:'old_dir'\ \ :o:'='\ \ \ :n:'os
\ :n:'os'\ \ :o:'. '\ \ :n:'chdir'\ \ :p:'('\ \ :bp:'self'\ \

\ :k:'def'\ \ :nf:'__exit__'\ \ :p:'('\ \ :bp:'self'\ \ :p:','\
\ :n:'os'\ \ :o:'. '\ \ :n:'chdir'\ \ :p:'('\ \ :bp:'self'\ \
```

3. It's even easier to do:

```
\ :kn:'from'\ \ :nn:'contextlib'\ \ :kn:'import'\ \ \ :n:'contextma

\ :nd:'@contextmanager'\
\ :k:'def'\ \ :nf:'cd'\ \ :p:'('\ \ :n:'path'\ \ :p:'):\
\ :n:'old_dir'\ \ :o:'='\ \ \ :n:'os'\ \ :o:'. '\ \ :n:'getcwd'\
\ :n:'os'\ \ :o:'. '\ \ :n:'chdir'\ \ :p:'('\ \ :n:'path'\ \ :p:'):\
\ :k:'yield'\
\ :n:'os'\ \ :o:'. '\ \ :n:'chdir'\ \ :p:'('\ \ :n:'old_dir'\ \ \
```

4.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "path.py".

That's cool, so let's add it to path.py.. path.py: <https://github.com/dottedmag/path.py/pull/13/files>

5. Maybe check for exceptions

```
\ :nd:'@contextmanager'\
\ :k:'def'\ \ :nf:'cd'\ \ :p:'('\ \ :n:'path'\ \ :p:'):\
\ :n:'old_dir'\ \ :o:'='\ \ \ :n:'os'\ \ :o:'. '\ \ :n:'getcwd'\
\ :n:'os'\ \ :o:'. '\ \ :n:'chdir'\ \ :p:'('\ \ :n:'path'\ \ :p:'):\
\ :k:'try'\ \ :p:':'\
\ :k:'yield'\
\ :k:'finally'\ \ :p:':'\
\ :n:'os'\ \ :o:'. '\ \ :n:'chdir'\ \ :p:'('\ \ :n:'old_dir'\ \
```

All in all, I learned how to do context managers, about contextlib, about fabric and about path.py. Which is not bad for 15 minutes :-)

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "rst2pdf".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "rst2pdf".

Of all the corpses of my projects, there is one that I feel worse about, which is `rst2pdf`. `rst2pdf`: <http://rst2pdf.googlecode.com>

I feel bad about abandoning several, but `rst2pdf` was actually a useful tool, used by a bunch of people, and that it has never gathered enough momentum with other developers is sad.

So, I will pick it up. I will spend about 4 hours a week on it. The plan is to:

- 1) Gather some patches that are lingering on the issue tracker
- 2) Fix some simple-ish bugs
- 3) Make another release with 1) and 2)

And of course:

- 1) Not let it fall in disrepair again

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "hacked rst2pdf".

In the meantime, here is a nice thing I just heard about. Dimitri Christodoulou has `hacked rst2pdf`. `hacked rst2pdf`: <http://dimitri-christodoulou.blogspot.com/search/label/rst2pdf>

so that it can handle the `raw::html` directive.

This, dear friends is completely nuts, absolutely out of scope for any given docutils tool, and just too cool :-)

I will try to hijack his code (proper credit and so on), and incorporate it into `rst2pdf`.

And Dimitri, or anyone else who wants to do cool stuff with `rst2pdf`: let me know! I will give you commit rights immediately!

Interesting thing about writing a post every day: I didn't get blocked yet. Sure, this is only the 9th post, but considering my previous post-a-month cadence, and that I would just have nothing to write about, it's clear to me that writing leads to writing.

That is perhaps the most obvious thing that not everyone notices: the way to do things is to do things. The way to write free software is to write software that is free. To go to Alaska you have to go. To Alaska. To write, you have to write. To do music, you have to do music. To make bread, you have to make bread.

Will every piece of bread you make be good? Will everything you write be good? Will you get to Alaska? No. You will fail.

But if you don't do failed crap first, then there is no way to do anything decent later. I think it was Amadeo Carrizo (famous, old, goalkeeper) who said about a (not so famous, young, goalkeeper) "He's not bad, but he needs to be scored on a few hundred times more before he's good".

I have a long history of failure. I have a short story of successes. I am working on it.

It must be horrible to have said something smart, maybe even insightful, once, and be labeled for life as a visionary, or a pundit, or some other similar insult.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "on this article".

Consider Clay Shirky's comments [on this article](#).. on this article: <http://www.nytimes.com/2012/01/10/health/views>. It's absolute nonsense, based on random guessing.

I promise this: this blog will have no insight on anything as long as I write it. I couldn't handle the pressure.

Yes, you do. If you are a frequent reader of this blog, then you probably already know about binary numbers, hexadecimal numbers, and sundry non-decimal numbers. You know, the kind we nerds know about. The ones that make us confuse thanksgiving and christmas because `oct(31) == dec(25)`.

But how about normal people (or as I like to call them: people)? Well, they may look at you confusedly if you tell them that they use way more exotic things every day.

Let's start with the time. When you say "it's 10:30" well, that's a base-60 number.

If we add days, it gets harder, because days are base-24. So "2 days, 10 hours and 30 minutes" is just a difficult way to say $2*24*60 + 10*60 + 30$ minutes. It's a numerical system with *two different bases*.

Sure, it doesn't do the cutesy thing hex does of having extra symbols, like A meaning 10, but it's exactly that, except 20 is written "20" or "8PM".

And how about January 11th, at 5:20 PM? Well, that is also another way to express a number of minutes, in an even *more* complicated mixed-base system!

```
January = 0 months = 0 days = 0 hours = 0 minutes
11th = 11 days = 251 hours = 15060 minutes
5PM = 17 hours = 1020 minutes
20 = 20 minutes
```

```
Total: 16100 minutes
```

That way to express a date uses a mix of base 60, base 24, and base 365 (if we can, please, ignore leap years) or maybe base 60, base 24, base ~30 and base 12

I don't know if numerical systems with non-fixed bases have a *name* in mathematics, yet you use them, random non-math-person!

And you can even do arithmetic on them! Yes, you! You know what exact time it will be at "Jan 9th 2:10 + 12:15". You can even do multi-base arithmetic *in your head*.

And I have not mentioned seconds (base 60 again), years (multiple base 10 digits) and second fractions.

Yet, when hex and binary are explained to people in school, it's incredibly hard to make them "get it". And once they get *that* if you try to explain, say base-3 numbers, it's confusing again.

Corriendo en la lluvia con Tato, pasó lo inevitable. Pisé una baldosa floja y me empapé la otra pierna. Como dice el tango "igual que baldosa floja, salpico si alguien me pone el pie". Y porque tengo que pensar todo el día en inglés por el laburo, mi software cerebral de traducción instantánea se tildó.

¿Como miércoles le haría entender esa frase a un estadounidense? Es casi imposible. Allá nadie camina, menos abajo de la lluvia. Casi no hay veredas. Las veredas que hay no son de baldosas. Las baldosas seguramente no estarían flojas.

Para poder traducir algo, no alcanza con traducirlo, hay que traducirlo y que quede algo que habría dicho alguien en el idioma de destino. Para que signifique lo mismo,

tendría que estar hablando de algo que podría pasarle, por ejemplo, a un argentino y a un ruso, a un bosquimano y a un lapón. Sospecho que no es posible traducir en general. Que lo que vemos por ejemplo en este blog, que intenta, los demás días, ser bilingüe, es una colección de casos particulares más o menos afortunados.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “[lean acá si quieren](#)”.

Hace un tiempo empecé a traducir una novela de Cory Doctorow ([lean acá si quieren](#).. [lean acá si quieren](#): http://lateral.netmanagers.com.ar/static/For_the_Win.txt) y abandoné porque al releer lo que escribí, no parecía una novela de Doctorow, parecía otra cosa, una cosa peor. Y no vale la pena leer cosas peores.

Y de esa forma cuando programo el programa nunca es lo que yo quise, es una versión peor, escrita en un lenguaje extranjero, por un no-nativo. Programar es traicionar la visión para producir lo tangible. Escribir es más o menos lo mismo. Hablar es más o menos lo mismo. Ni siquiera mi voz que vos escuches es mi voz que yo escucho.

Vivimos cada uno encerrados en un iglú, tratando de charlar con mapuches que nos miran a través del hielo. A veces funciona. Saludos.

My dad was born in the province of Chaco, and migrated (very slowly) some 1500km to the south. The migration was slow enough that he managed to have two different local football teams to like.

The one he liked when I met him was Unión de Santa Fe, but the team of his childhood and youth was the remarkably obscure Chaco Forever. And of course, because it's what people do when their teams have no chance of ever winning anything, he liked one “big” team just to have some chance to celebrate (Boca Juniors, in his case).

Once, at the end of the 60s, he was driving north, towards Resistencia, the home of Chaco Forever, with my mom and my big brother, maybe a year old sitting in my mom's lap, no seatbelts. Those were the 60s and children survived because lead poisoning seems to be an antidote for second hand smoke and car crashes.

Then, his car stalls. That was hardly unusual. Our usual average speed on long trips was about 20km/h once you accounted for the stops to add water to the radiator, change flat tires, get a mechanic to see why the lights didn't work, and arriving at the wrong town. My dad liked cars, but cars hated him.

But before the car stalled, they had seen a light by the side of the road, up high. A light that seemed to follow them. And the car didn't start. That road in those times was lonely, and dark, and in bad maintenance. So stopping in the middle of it was a recipe for being killed by a truck.

A little later, the car starts, and the light appears again, and again it stops. Here, according to my mother, my dad got out of the car, and started shouting at the aliens to stop being idiots, that they were going to get him killed. After that, the light disappeared, and they continued traveling without further mechanical issues.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “[a ufo appeared over a football practice](#)”.

And a few days ago, this happened: [A UFO appeared over a football practice](#).. a ufo appeared over a football practice: http://www.visionovni.com.ar/modules/newbbex/viewtopic.php?topic_id=984&f

. The football team? Chaco Forever.

Well, just because I want to use KHTML in KR&N, I got the latest PyKDE. It says it is for KDE up to 3.0.3 ...

Well, with very minimal hacking (deleting two lines in KFontDialog's sip file) and a couple of shoves at the build.py, it seems to be compiling alright on KDE 3.1.

Oh, sure, it won't expose the new pieces of API in KDE 3.1, but who cares? It will work. I suppose I will post it somewhere as soon as it does.

Using idioms is a double-edged sword. One side lets you cut to the point. You say a phrase, and it's so loaded of meaning, it's a shortcut to what you mean. One side can cut you, if you just use it and you use it wrong.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "she wrote one today".

Let's consider Beatriz Sarlo, a respected argentine intelectual. She writes op-ed columns in a newspaper. [She wrote one today](http://t.co/kA8ZH6M6).. she wrote one today: <http://t.co/kA8ZH6M6>

. It ends:

Total, Boudou, sin bromita alguna, debe adecuarse a lo que le toque, obediendo el viejo refrán de que a un caballo regalado no hay que examinarlo para ver si viene completo.

Which I translate as:

In any case, Boudou, jokes aside, has to accept what he gets, following that old advice about not examining gift horses to see if they are whole.

One peculiar thing here is that we have almost the exact same saying in spanish and english. In english you don't look gift horses in the mouth. In spanish you don't check the teeth of gift horses. The problem here (if I may be so pedantic (yes you may (thanks other me! (my pleasure)))) is that Ms. Sarlo has no freaking clue of what that means.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here is google's first result".

Suppose you are buying a horse. You would check the mouth because you want to see if the horse is young or old (citation needed? [Here is google's first result](http://horses-arizona.com/pages/articles/teeth.html).. here is google's first result: <http://horses-arizona.com/pages/articles/teeth.html>

). That's important when you buy a horse. It's, on the other hand, really useless and rude when you are getting a horse as a gift:

Nice guy: Hey, here's a horse!

Rude moron: (looks in the mouth) No thanks!

That's why you don't look gift horses in the mouth. And that's also why you don't check amazon to see how much the book you got costs.

On the other hand, if you were a Tamarian, you could look at the horse's mouth and say "Temba, at rest".

Spain is a country. Córdoba is a place in Spain. In Córdoba there are churches. People go to churches. The people who attend church in Córdoba, Spain, are under the spiritual guidance of Demetrio Fernandez, bishop.

Just so we are clear: You are supposed to greet him by kissing his hand.
On the other hand, Demetrio Fernandez is an idiot who said this:

“The Minister for Family of the Papal Government, Cardinal Antonelli, told me a few days ago in Zaragoza that UNESCO has a program for the next 20 years to make half the world population homosexual. To do this they have distinct programs, and will continue to implant the ideology that is already present in our schools.”

I rest my case. Demetrio Fernandez is an idiot. Cardinal Antonelli is an idiot. And if you go to church in Córdoba, Spain, knowing about this, and kiss the idiot’s hand, you are an idiot.

Antonio María Delgado writes for the Miami New Herald. I would dare guess he’s supposed to know english, even if he writes in spanish.

Let’s backtrack a bit. The Economist publishes (as it has done for a while), something called the “Misery Index”, which attempts to capture the countries where people are feeling miserable, and rank them.

So, Mr. Delgado goes, reads that, sees Venezuela near the top, and I can imagine the idea lightbulb brightening to a dazzling 0.5 watts, his scream of “political angle!” and the rush to write this:

“Venezuela se ubicó el año pasado en el segundo puesto del ”Indice de Miseria“ elaborado por la revista británica The Economist...”

The problem, dear english reader, is that “miseria” doesn’t mean “misery”. It means “poverty”. In fact, it means “extreme poverty”. And thus, Mr. Delgado makes it seem as if the Economist is saying Venezuela is the second poorest country (in some sense).

While I personally dislike Chavez, this is either ignorant, or stupid. Your call.

- a) They all claim that evolution doesn’t work
- b) They all claim the world is less than 10000 years old

So you have now two choices:

- a) They believe they are saying the truth, which means they are morons.
- b) They believe they are lying, and only saying obviously false things because that will help them win, which means they are sleazy.

Have a fun future, US!

QTimer is a fairly simple class: you use it when you want something to happen “in a while” or “every once in a while”.

The first case is something like this:

```
\ :c: '# call f() in 3 seconds'\  
\ :n: 'QTimer'\ \ :o: '. '\ \ :n: 'singleShot'\ \ :p: '( '\ \ :mi: '3000'\ \ :p: ',
```

The second is this:

```
\ :c: '# Create a QTimer'\  
\ :n: 'timer'\ \ :o: '= '\ \ :n: 'QTimer'\ \ :p: '() '\  
\ :c: '# Connect it to f'\  
\ :n: 'timer'\ \ :o: '. '\ \ :n: 'timeout'\ \ :o: '. '\ \ :n: 'connect'\ \ :p: '( '\  
\ :c: '# Call f() every 5 seconds'\  
\ :n: 'timer'\ \ :o: '. '\ \ :n: 'start'\ \ :p: '( '\ \ :mi: '5000'\ \ :p: ') '\
```

Simple, right? Well, yes, but it has some tricks.

- 1) You have to keep a reference to “timer“

If you don't, it will get garbage-collected, and `f()` will never be called.

- 2) It may not call `f()` in 5 seconds.

It will call `f()` more or less 5 seconds *after you enter the event loop*. That may not be quickly after you start the timer at all!

- 3) You may get overlapping calls.

If `f()` takes long to finish and re-enters the event loop (for example, by calling `processEvents`) maybe the timer will timeout and call it again before it's finished. That's almost never a good thing.

So, you can do this:

```
\ :k:'def'\ \ :nf:'f'\ \ :p:'():'\
  \ :k:'try'\ \ :p:':'\
    \ :c:'# Do things'\
  \ :k:'finally'\ \ :p:':'\
    \ :n:'QTimer'\ \ :o:'. '\ \ :n:'singleShot'\ \ :p:'('\ \ :mi:'5000'\
\ :n:'f'\ \ :p:'() '\
```

What that snippet does, is, calls `f()` only once. But `f` itself schedules itself to run in 5 seconds. Since it does it in a `finally`, it will do so even if things break.

That means no overlapping calls. It also means it won't be called every 5 seconds, but 5 seconds plus whatever `f` takes to run. Also, no need to keep any reference to a `QTimer`.

Final tip: You can also use `QTimer` to do something “as soon as you are in the event loop”

```
\ :n:'QTimer'\ \ :o:'. '\ \ :n:'singleShot'\ \ :p:'('\ \ :mi:'0'\ \ :p:', '\
```

Hope it was useful!

You are in an open field west of a big white house with a boarded front door.

There is a small mailbox here.

>

We live in the golden age of text. People write more than ever before. People read more than ever before. Only a few short years ago, the preferred mechanism to contact other people was voice based “phone calls”. How quaint that appears to the modern person, that types messages through any of today's jungle of messaging systems.

Sure, we also take more pictures than ever before. And more video than ever before. Because we are, in general, in an information golden age. But people *expected* that.

Who expected, 20 years ago, that kids would prefer to type short messages to each other instead of having long phone calls? Who expected that people would want to *read* the messages they got, instead of listening to them? Or watching them?

Which is strange, since for as long as I have a memory, I have been reading that the younger folks can't read or write as well as the oldest generations (of which I am now part of). The dreaded lack of "written text comprehension", meaning kids simply did not understand what the heck they were reading. Sure, they could form the words in their minds, but the complex aspects (plot, etc) simply didn't catch.

But why are those kids who could not read writing and reading so much? Is that a paradox? Or is it just that they didn't care about what they were reading, and when, later in life, they decided to pay attention, they did *get it*?

Or maybe it is that understanding long plots is not what reading is about nowadays. That reading is about getting many small nuggets of data, and the whole correlation is done in our heads, instead of having it spelled out in long, comprehensive texts.

```
> open mailbox
```

```
Opening the mailbox reveals:  
A leaflet.
```

Maybe the problem with long texts and reading comprehension is that they are too specific. Once you explain everything, maybe it's boring, and people's mind wander off. Maybe you need to keep things short and open-ended. Maybe the reader wants to fill in the blanks.

```
> read leaflet
```

```
Taken.  
Welcome to Zork (originally Dungeon)!
```

```
Dungeon is a game of adventure, danger, and low cunning. In it  
you will explore some of the most amazing territory ever seen by  
mortal man. Hardened adventurers have run screaming from the  
terrors contained within.
```

So, maybe the right way to write in the 21st century is short and evocative, instead of clever and wordy. Maybe the gaming component of reading needs to be amped up, and the user will win imaginary badges whenever he gets some insight from what he is reading, like one of those whatevervilles that give you meaningless awards for meaningless tasks accomplished with meaningless effort.

```
> go west
```

```
You are in a forest, with trees in all directions around you.
```

Or maybe it's the other way around. Maybe reading (or writing) is becoming split into two different things. Maybe we are developing a high and a low reading. A high reading that is what traditionally was called reading, and a low reading that is short and functional and not all that interesting.

```
> go west
```

```
Forest
```

And maybe this means we will get things like China Miéville's novels, full of weird for weird's sake, and fun, and things like twitter, snippets full of whimsy and connection and wit (hey, I follow interesting people), and blogs full of disjointed miscelanea, and 9gag full of things that should not, in all honesty, be funny.

```
> reset
```

```
Starting over.
```

PS: <http://thcnet.net/zork/index.php>

This post makes no sense in english, so spanish only!

Drácula es un libro muy particular. Casi todo el mundo cree que sabe de qué se trata, pero en el 90% de los casos no es así. O sea, sí, saben que es de un vampiro, blabla.

Lo que no saben es nada del *libro*. Saben de las películas, del especial de Scooby Doo, de los chistes de vampiros, y cosas así, pero el libro en sí, no lo han leído.

¡Y es una lástima! Es un libro muy interesante. Para la época que se publicó, tiene un estilo dinámico y poco verborágico. Está lleno de acción, escenas memorables (no es raro que se hayan hecho tantas películas), personajes interesantes. ¡Y encima es un libro tecnófilo! No es una lectura forzada leer Drácula como una micro-expresión de la lucha entre la ciencia y la técnica positivista contra la cultura medieval reaccionaria, o cosas así.

Y entonces, cuando el otro día ví, en Work of Art (un reality), a unos diseñadores crear tapas para Drácula, se me ocurrió:

Editemos Drácula

Agarremos el original, que es de dominio público, hagamos una traducción moderna, hagamos ebooks, y quién te dice, una edición en papel. Regalémosle a la gente la posibilidad de leer una versión moderna de este libro buenísimo. Una traducción que no sea castiza, ni dé vergüenza hablando de “estofado con polvo de pimiento rojo” si no que diga goulash, o por lo menos “estofado con paprika”.

Yo tengo un poquito de cancha haciendo typesetting de libros. Seguro que se puede conseguir ilustrador/a/es copado/a/as/os para la tapa, títulos (¡o para ilustrar intercalado!)

Y... lo mejor es que:

- 1) Es un libro relativamente corto
- 2) Está escrito en muchas voces distintas. No estaría buenísimo que Mina Harker lo escriba una mujer? Que Van Helsing escriba totalmente distinto que Lucy?

Entonces: se necesitan muchas cosas, pero más que nada, se necesitan traductores. ¿Quién quiere salir en la tapa de este libro? ¿Quién quiere traer al auténtico Drácula, el vampiro en serio, un auténtico macho de los cárpatos, de vuelta a la vida? Anótense en los comentarios.

system-message

```
WARNING/2 in everything.txt, line 3  
Duplicate explicit target name: “de vicenzo”.
```

system-message

```
WARNING/2 in everything.txt, line 3  
Duplicate explicit target name: “de vicenzo”.
```

About a year ago, I wrote a small web browser, called [De Vicenzo](http://devicenzo.googlecode.com).. de vicenzo: <http://devicenzo.googlecode.com>

just for fun.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “previews when doing sphix docs”.

But hey, someone went and made it useful for something! Specifically, to provide previews when doing sphix docs.. previews when doing sphix docs: <http://pymolurus.blogspot.com/2012/01/document>
That's cool :)

Hello, my name is Roberto Alsina. You may know me from my appearances in “KDE Developers in the Stone Age” and “PyQt programmers gone wild”.

On the other hand, I am not:

- Roberto Alsina, who lives in Houston and is active in the Argentine expatriate community.
- Amado Roberto Alsina, paraguayan politician.
- Roberto Ariel Alsina (aka Roberstorm) who drives a motorcycle.
- Roberto Alsina, puertorican architect.
- Roberto Alsina, who lives in Villa Alemana and attended Colegio Buckingham.
- Roberto Alsina Ruibal, who I hope is wearing a wig.
- Roberto Gonzales Alsina, who lives in Canelones. I only *eat* canelones.
- The FEARED Roberto Nava Alsina who, while only 15 years old, is scoring goals in some amateur league somewhere.
- Roberto Lebron Alsina (I have slightly more hair)
- Roberto Antonio Gómez Alsina, activist against bullfighting

Hopefully, this will clear things up. Thanks for reading.

This has been a repeated discussion in the Python Argentina mailing list. Since it has not come up in a while, why not recap it, so the next time it happens people can just link here.

Some people for some reason do this:

```
\ :o: '>>>' \ :n: 'a' \ :o: '=' \ :mi: '2' \
\ :o: '>>>' \ :n: 'b' \ :o: '=' \ :mi: '2' \
\ :o: '>>>' \ :n: 'a' \ :o: '==' \ :n: 'b' \
\ :bp: 'True' \
\ :o: '>>>' \ :n: 'a' \ :ow: 'is' \ :n: 'b' \
\ :bp: 'True' \
```

And then, when they do this, they are surprised:

```
\ :o: '>>>' \ :n: 'a' \ :o: '=' \ :mi: '1000' \
\ :o: '>>>' \ :n: 'b' \ :o: '=' \ :mi: '1000' \
\ :o: '>>>' \ :n: 'a' \ :o: '==' \ :n: 'b' \
\ :bp: 'True' \
\ :o: '>>>' \ :n: 'a' \ :ow: 'is' \ :n: 'b' \
\ :bp: 'False' \
```

They are surprised because “2 is 2” makes more intuitive sense than “1000 is not 1000”. This could be attributed to an inclination towards platonism, but really, it’s because they don’t know what is is.

The is operator is (on CPython) simply a memory address comparison. If objects a and b are the same exact chunk of memory, then they “are” each other. Since python pre-creates a bunch of small integers, then every 2 you create is really not a new 2, but the same 2 of last time.

This works because of two things:

- 1) Integers are read-only objects. You can have as many variables “holding” the same 2, because they can’t break it.
- 2) In python, assignment is just aliasing. You are not making a copy of 2 when you do a = 2, you are just saying “a is another name for this 2 here”.

This is surprising for people coming from other languages, like, say, C or C++. In those languages, a variable int a will never use the same memory space as another variable int b because a and b are names for specific bytes of memory, and you can change the contents of those bytes. On C and C++, integers are a mutable type. This 2 is not that 2, unless you do it intentionally using pointers.

In fact, the way assignment works on Python also leads to other surprises, more interesting in real life. For example, look at this session:

```
\ :o: '>>>' \ :k: 'def' \ :nf: 'f' \ :p: '(' \ :n: 's' \ :o: '=' \ :s: ''
\ :o: '...' \ :n: 's' \ :o: '+=' \ :s: 'x'
\ :o: '...' \ :k: 'return' \ :n: 's'
\ :o: '...'
\ :o: '>>>' \ :n: 'f' \ :p: '()'
\ :s: 'x'
\ :o: '>>>' \ :n: 'f' \ :p: '()'
\ :s: 'x'
\ :o: '>>>' \ :n: 'f' \ :p: '()'
\ :s: 'x'
```

That is really not surprising. Now, let’s make a very small change:

```
\ :o: '>>>' \ :k: 'def' \ :nf: 'f' \ :p: '(' \ :n: 'l' \ :o: '=' \ :p: '['
\ :o: '...' \ :n: 'l' \ :o: '.' \ :n: 'append' \ :p: '(' \ :s: 'x'
\ :o: '...' \ :k: 'return' \ :n: 'l'
\ :o: '...'
\ :o: '>>>' \ :n: 'f' \ :p: '()'
\ :p: '[' \ :s: 'x' \ :p: ']'
\ :o: '>>>' \ :n: 'f' \ :p: '()'
\ :p: '[' \ :s: 'x' \ :p: ', ' \ :s: 'x' \ :p: ']'
\ :o: '>>>' \ :n: 'f' \ :p: '()'
\ :p: '[' \ :s: 'x' \ :p: ', ' \ :s: 'x' \ :p: ', ' \ :s: 'x' \ :p: ']'
```

And that is, for someone who has not seen it before, surprising. It happens because lists are a mutable type. The default argument is defined when the function is parsed, and every time you call f() you are using and returning the same l. Before, you were

also using always the same `s` but since strings are immutable, it never changed, and you were returning a new string each time.

You could check that I am telling you the truth, using `is`, of course. And BTW, this is not a problem just for lists. It's a problem for objects of every class you create yourself, unless you bother making it immutable somehow. So let's be careful with default arguments, ok?

But the main problem about finding the original `1000 is not 1000` thing surprising is that, in truth, it's uninteresting. Integers are fungible. You don't care if they are the same integer, you only really care that they are equal.

Testing for integer identity is like worrying, after you loan me \$1, about whether I return you a different or the same \$1 coin. It just doesn't matter. What you want is just a \$1 coin, or a 2, or a 1000.

Also, the result of `2 is 2` is implementation dependent. There is no reason, beyond an optimization, for that to be `True`.

Hoping this was clear, let me give you a last snippet:

```
\ :o: '>>>' \ :n: 'a' \ :o: '=' \ :nb: 'float' \ :p: '(' \ :s: 'NaN' \
\ :o: '>>>' \ :n: 'a' \ :ow: 'is' \ :n: 'a'
\ :bp: 'True'
\ :o: '>>>' \ :n: 'a' \ :o: '==' \ :n: 'a'
\ :bp: 'False'
```

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "at reddit".

UPDATE: lots of fun and interesting comments about this post [at reddit](http://www.reddit.com/r/Python/comments/p3owk/the_problem_is_is_is_it_not/).. at reddit:
http://www.reddit.com/r/Python/comments/p3owk/the_problem_is_is_is_it_not/

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "here".

and a small followup [here](http://lateral.netmanagers.com.ar/weblog/posts/BB981.html).. here: <http://lateral.netmanagers.com.ar/weblog/posts/BB981.html>

The `nth` module for iptables.

This nifty gadget lets you match the `nth` packet to a rule.

What for? Well, suppose you have two links, and have implemented split access as per the [LARTC](#)

According to the same LARTC, you can implement per-route load balancing by creating a multipath route.

That works well... as long as the traffic originates on the firewall itself (say, using a [Squid](#)).

If the traffic comes from a SNATd subnet, it breaks, because you SNAT (or MASQ) it to one of your external IPs, and then it's routed only through that link, for obvious reasons^[1].

You *can* route based on original source IP, so you can tell half the boxes to go left, and the other half to go right.

And then if the client boxes are used unevenly, your balancing sucks.

So, what's the solution? Match every second *state NEW* packet over each link.

Since iptables's MASQ or SNAT will make the *state ESTABLISHED,RELATED* packets follow the leader, each connection alternatively routes left or right.

While not 100% right (you can be unlucky and redirect all long connections on the same link), it is **much** better than the simple alternatives, and **much** simpler than the better alternatives.

But hey, nth is only on [patchomatic](#). And Red Hat's (Fedora's) kernel makes [patchomatic](#) go nuts.

So it's custom-kernel-compiling time, and I hate doing that. Really, this patch seems simple. Why is it not in?

Sorry: english only because my kbd is acting up an typing is a chore.

Here I am, writing in a place of evil so deep they charge you \$56 (that's pesos) for a tiny sandwich and a bottle of water: the Gianni & Vittorio café at Córdoba Airport.

It's like Mordor, except the water is kinda-sorta-slightly-cold, and I carry no rings. And there is very little lava. But the evil... the evil is dense, as it can only be at airports, Minas Morgul, and perhaps the choripán place near Lisandro de la Torre station (which I strongly recommend).

But why do I do it? Because I have already failed. I will not post everyday this year. I may not even be close. But I will post as much as I can. And I promise it won't be filler about why I did not do a real post.

So, onto the meat of this post we go.

It's inevitable that airports will have expensive and bad food. It will be expensive because there are a limited number of sellers, and getting to be one of them is expensive. Since they can charge whatever they want, they have no incentives to provide quality or service.

Plus, they own a captive customer base, since you are locked there, and there is no place to go outside the airport either.

That is classic government-mandated market distortion, with the airport management as the government, and you playing the role of you. This kind of clear examples are good because they show capitalism and free market advocates actually have a decent point when they remark on the defects of imperfect markets.

OTOH, I don't see any communists around here oppressing me, I see only happy capitalists taking my money.

PS: it seems there was a tornado here today.

¹

system-message

WARNING/2 in everything.txt, line 31566
Duplicate explicit target name: "1".

If you routed it through the other, your ISP would reject the packets, because the origin IP is foreign to him.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "the problem is is, is it not?".

Just a quick followup to [The problem is is, is it not?.. the problem is is, is it not?: http://lateral.netmanagers.com.ar/weblog/posts/BB979.html](http://lateral.netmanagers.com.ar/weblog/posts/BB979.html)

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "from reddit".

This is not mine, I got it from reddit.. from reddit: http://www.reddit.com/r/Python/comments/p3owk/the_problem

This should really not surprise you:

```
\ :o:'>>>' \ :n:'a' \ :o:'=' \ :p:'[\ \ :mi:'1' \ :p:', \ :mi:'2'
\ :o:'>>>' \ :n:'b' \ :o:'=' \ :p:'[\ \ :mi:'3' \ :p:', \ :mi:'4'
\ :o:'>>>' \ :n:'a' \ :ow:'is' \ :n:'b'
\ :bp:'False'
\ :o:'>>>' \ :n:'a' \ :o:'==' \ :n:'b'
\ :bp:'False'
\ :o:'>>>' \ :nb:'id' \ :p:'(\ \ :n:'a' \ :p:') \ :o:'==' \ :nb:'
\ :bp:'False'
```

After all, a and b are completely different things. However:

```
\ :o:'>>>' \ :p:'[\ \ :mi:'1' \ :p:', \ :mi:'2' \ :p:'] \ :ow:'is'
\ :bp:'False'
\ :o:'>>>' \ :p:'[\ \ :mi:'1' \ :p:', \ :mi:'2' \ :p:'] \ :o:'=='
\ :bp:'False'
\ :o:'>>>' \ :nb:'id' \ :p:'([\ \ :mi:'1' \ :p:', \ :mi:'2' \ :p:']
\ :bp:'True'
```

Turns out that using literals, one of those things is not like the others.

First, the explanation so you understand why this happens. When you don't have any more references to a piece of data, it will get garbage collected, the memory will be freed, so it can be reused for other things.

In the first case, I am keeping references to both lists in the variables a and b. That means the lists have to exist at all times, since I can always say `print a` and python has to know what's in it.

In the second case, I am using literals, which means there is no reference to the lists after they are used. When python evaluates `id([1,2]) == id([3,4])` it first evaluates the left side of the `==`. After that is done, there is no need to keep `[1,2]` available, so it's deleted. Then, when evaluating the right side, it creates `[3,4]`.

By pure chance, it will use the exact same place for it as it was using for `[1,2]`. So `id` will return the same value. This is just to remind you of a couple of things:

- 1) `a is b` is usually (but not always) the same as `id(a) == id(b)`
- 2) garbage collection can cause side effects you may not be expecting

Sorry, spanish only today!

Debo haber dado, en los últimos 15 años, unas 80 charlas. O tal vez 50. Qué se yo, no las ando contando. Desde hace unos cinco años a esta parte, me dicen que vienen saliendo buenas (con altibajos, por supuesto).

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “how to be a pycon speaker and do it well”.

Entonces, inspirado por [How to be a PyCon speaker and do it well..](http://readthedocs.org/docs/ref/en/latest/advice_for_pycon_speakers/index.html) how to be a pycon speaker and do it well: http://readthedocs.org/docs/ref/en/latest/advice_for_pycon_speakers/index.html

acá van una serie de consejitos para que los que no se animan a dar charlas se animen.

Este es el más importante: el público no muerde. En serio, lo peor que vas a escuchar es algo del nivel de “se ve que estaba nervioso” o “se perdió un poco en tal parte” o “el tema no me interesó mucho”. En la gran mayoría de los eventos que he estado, no hay ningún ánimo de agredir al orador. Nadie te va a tirar tomates. Y al día siguiente nadie se va a acordar si estabas nervioso.

Tenés que saber a quién le estás hablando. Hacéte un modelo de como es tu espectador ideal. Hacé una charla que le pueda interesar a ese modelo. Si tratás de hacer que le guste “a cualquiera” entonces no le va a gustar a nadie.

Pensá mucho el título. Si tu charla es para principiantes que quieran aprender flask, ponele de nombre algo como “Introducción para los que no saben nada nada nada de Flask pero quieren aprender un poquitito de como usar un microframework web que está buenísimo”. No le pongas “Flask: un microframework web”. Ok, no le pongas ninguno de los dos, ponele “Flask desde cero”. Yo le pondría “Flaska, no me claves, los puñales, por la espalda” pero eso es porque me gusta Bottle.

No pienses los slides. Pensá lo que querés decir. Una vez que lo sabés, partílo en, ponele, 10 pedacitos de 3 minutos, y hacé un slide para cada uno. Si no podés hacer un slide, hacé dos.

Los slides son el menú, no la comida. Los slides no son la charla. Si puedo entender la charla viendo los slides, entonces o tu charla está estirada, o tenés muchos slides. Asegurate que se lean. No importa tanto que sean lindos: **que se lean**. Regla: poné los slides en tu notebook, poné el brillo en mínimo, y miralo desde 3 metros. Si no lo ves, no se vé.

Cuanto ensayar, depende de cada uno. Yo no ensayo, pero es porque soy un tarado. Si ensayás demasiado sale medio sin espontaneidad, tal vez. Pero si no ensayás nada, a veces ni sale. O sea, ensayá hasta que te sientas cómodo, y ahí pará.

Hablá como hablás vos. No trates de hablar como otro, aunque te guste mucho mucho mucho como habla. No trates de ser gracioso, si sos gracioso va a salir natural. No trates de ser interesante, si la charla es interesante alcanza.

No hagas cosas interactivas a menos que estés segurísimo que no te vas a trabar.

No hables de cosas que no te interesen “porque son importantes” o “porque alguien debería hablar de eso”.

Hablá fuerte, aunque tengas micrófono. Cuando hablás fuerte hablás más claro y pausado.

Tenés que saber cuanto tiempo te queda. Todo el tiempo.

Tenés que tener dos o tres puntos para terminar la charla. Cuando se te va acabando el tiempo, la terminás en el que mejor te quede.

“Decíles lo que les vas a decir, decílo, y decíles que les dijiste”

Si no sabés, admitílo. Si guitarreás, decí “estoy guitarreando, pero...”. Si estás nervioso decí “me estoy meando”, si te perdiste tirá todo a la mierda y uníte al circo Rodas (o pedí un segundo para ordenarte).

Ponéle onda a la parte de preguntas y respuestas. No pongas cara de “no entendiste una goma de lo que dije”, porque capaz que es culpa tuya.

Rompé todas las recomendaciones que te dí si te parece que va a funcionar mejor. Por ejemplo, capaz que querés tener 100 slides y pasar uno cada 10 segundos. O no querés tener slides. O querés tener slides que hablan de otra cosa. O querés tener slides que pasen solos. O querés tener slides interactivos y actuar Míngo y Aníbal contra los fantasmas.

Y la más importante (si, ya sé que dije que la otra era lo más importante, no jodas): followup. Invitálos a charlar cuando te vean en el pasillo. Dejá tu mail, blog, twitter o lo que sea. Invitá una cerveza. La charla en sí sirve hasta ahí, es mejor si de la charla sacás un amigo.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “coffee and i”.

If you have read [Coffee and I.](http://lateral.netmanagers.com.ar/weblog/posts/BB960.html) coffee and i: <http://lateral.netmanagers.com.ar/weblog/posts/BB960.html>

... well, let’s say the 70s filter is proper, and that this was a happy evening for me.

Stories don’t repeat themselves if you don’t let them.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “this”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “this”.

I have a confession to make. For the last year or so, my main operating system has been Windows 7. Yes, I know, it may come as a shock to some, specially if you have read [this.](http://lateral.netmanagers.com.ar/stories/BBS45.html) this: <http://lateral.netmanagers.com.ar/stories/BBS45.html>

which is my post that had most hits in a day ever.

How did that happen? What happened to me? What was I thinking? It's a boring and uninteresting story.

I joined Canonical. My old notebook wouldn't cut it. My new one would not take Ubuntu without a fight. I said "hey, I will live in a VM!". The VM was deadly slow. I had to develop windows software (yes). Some stuff would not work right on the VM. And slowly, things just started piling up in the bare-metal OS, which was, yes Windows 7 Home Premium.

As a whole, Windows 7 is *not horrible*. Most things work well. What it is, is a desert for a developer. Sure, you can get a plant to grow there, but you have to put a lot of effort into it.

So, today I installed Kubuntu Oneiric (absolutely no problem now!), gathered all the data from the old notebook, the VM, the windows installation, deleted windows, and moved into Linux again, and made Windows the VM.

I missed it.

An alfajor is, in theory, a simple thing. It's a dessert sandwich. The argentinian alfajor is usually filled with dulce de leche, which means you just can't screw it up. Even a bad alfajor is going to be good.

Yes, there are some regional alfajores filled with other stuff. Don't pay any attention to those impostors. They are hipsters wearing fake glasses they don't need. What you want is dulce de leche.

But the filling is only half of a sandwich, and a third of an alfajor. There is also the things that surround, hold and contain the notoriously sticky dulce: cookies. Some variants have tried to improve on the cookie by either going soft (cake!) or hard (Milka Mousse), but the real deal is a soft-ish cookie, not too soft (so the alfajor doesn't desintegrate) and not too hard (so it doesn't feel like eating a hockey puck).

There is an impossible tradeoff, between dulce de leche and the cookie. Too much cookie, you are eating cookies. Too much dulce de leche, you may as well get a spoon and eat out of the jar (try it sometime). You want to balance, but the alfajor is walking food, it has to be edible by a 6-year old schoolkid while climbing a tree, it has to be solid enough, and not fall apart, and not cover him on melted frosting. The child has to be able to pass inspection of his hands after cleaning with just the wrapper and his own mouth.

The alfajor santafesino has adopted a minimax strategy, maximizing the amount of dulce de leche per amount of cookie, by using extra-thin "cookies" that are more like crackers, adding multiple layers, and making the whole alfajor thicker. I love it, but it is not for everyone.

The alfajor de maicena goes in a completely different direction, having a tasty cookie that can hardly hold any dulce de leche because it's too friable. Somehow that works wonderfully as well.

Then there is the covering. Chocolate, frosting, coconut, or nothing. Chocolate is bad in summer, melting and making you eat the alfajor from the wrapper as if it were a banana.

But if you have no idea what an alfajor is, which one should you have? I could point you to the most sublime alfajores, which would be life-changing experiences, but I will not. If you tried those, which you can only get in out of the way places, known only to initiates, where could you go from there? It's like your first date being with Sofía Vergara. That would be just leading you into a life of disappointment.

So get a chocolate Havanna. Get a Cachafaz. Get a Capitán del Espacio. Get a tube of Jorgitos. And when you have done your apprenticeship, when you are an alfajorman, when you are ready. Then you will know.

PS: Thanks to Juan Rodriguez Monti for the idea for this post.

Yes, that's my son drinking mate out of Donald Duck's skull. Caption?

I am very distracted when I walk down the street. Or rather, I am paying a lot of attention, but it's spread over a whole lot of different things.

Figure 13: Her name is *Faith Popcorn*. Seen on Mar del Plata.

My favourite, since I am a compulsive reader, is reading street signs. There is always something off about signs in a foreign country. They are either about things other countries don't care about, or are written in a completely different style.

Figure 14: Fixed hair braiding prices in Bahamas

And sometimes you run into things you just have never seen before. Those things can be found anywhere, and can be anything, since ... well, you have never seen them before.

Figure 15: And now, a hydrant wearing a sweater in Budapest.

It doesn't have to be something really strange, it may just be something you have not seen before by chance.

Figure 16: Street sweepers get OCD too. Seen on San Isidro.

Or maybe you just figure something out right there and then.

Figure 17: So that's why sugar cubes are better. Seen in Budapest.

Or ... you don't know what to say.

Figure 18: Seen at Tigre. I have no idea.

Or things you don't have where you come from.

Figure 19: Blimp! Seen in London.

Or they are just so *polite* to ask.

Or you don't understand at first.

Figure 20: How did that kid get there?

And then you do.

Figure 21: It's a trick fountain! Seen in Budapest.

Or maybe it's something you see every day, out of context.

Figure 22: A typical argentinian milanesa sandwich. Bought on the street in Budapest.

Figure 23: A DIA% supermarket, like the one near my home. In Istanbul.

And sometimes it's something you never suspected even existed, or how it could exist.

Figure 24: This is a chapter in a turkish book. It takes place at **my** wedding.

Or out of context.

Figure 25: Seen around the corner of my house.

Or alien.

Figure 26: Hotel towel, seen in Orlando, Florida.

Or

Figure 27: Yes, I did get a haircut. Seen in London.

Or

Figure 28: Seen in Junín.

Or

Figure 29: Ferry in Istanbul

Or

Figure 30: True TV Remote seen in a hotel in Avenida de Mayo, Buenos Aires, in 2004.

This, except for that TV remote, is just a small sample of what I have seen in the last 12 months. These have been a really *cool* 12 months.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "here".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "here".
```

Sorry, spanish only post. But you can listen to the song [here](http://tinysong.com/LNWF).. here: <http://tinysong.com/LNWF>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "paisano de hurlingham, de divididos".
```

Si tuviera 5 blogs y la energía para postear en todos, el cuarto sería “analizando demasiado la letra de canciones”. En homenaje a ese blog que nunca va a existir, este sería el primer post: [Paisano de Hurlingham, de Divididos.. paisano de hurlingham, de divididos: http://tinysong.com/LNWF](http://tinysong.com/LNWF)

(probablemente mi banda favorita).

Primero, la letra completa:

Paisano de Hurlingham
poda neblina
moneda o botón
ciego bilingüe
paso morales fue
sin la opalina
de Retiro a Pilar
busca el chanco al chabon.
Sapo explota en San Martín
los domingos a las diez
sable recto en la estación
berretin de mayor.
Canilla en el anden
gotea noticias
te grita el titular
mentiras sin picar.
Abejas con ombu
viajando en el panal
va la timba en el furgón.

Al parecer hay un cierto consenso (entre las tres personas a quienes les pregunté), de que esta canción es la descripción de un viaje en tren.

Las letras de Divididos no se caracterizan por ser interpretables linealmente, De hecho, sospecho que la mayoría son simplemente una serie de palabras una después de la otra porque “suenan bien juntas”.

El primer verso “Paisano de Hurlingham” es el título mismo de la canción, y, supongo, el protagonista de esta mínima odisea suburbana. “poda neblina” es interesante. No encuentro (gracias google) ninguna referencia a esa frase fuera de esta canción. Es posible que nadie jamás haya dicho “poda neblina” hasta que Mollo cantó esa estrofa.

Para que se hagan una idea de lo *raro* que es eso, hay dos referencias independientes a “navaja desierta”, que son dos palabras al azar sacadas del diccionario. ¿Entonces, qué es “poda neblina”? Bueno, si es muy temprano, hay neblina, y el paisano la atraviesa, la corta, la poda. Así que, tirando de los pelos (que lo vamos a hacer bastante), podemos suponer que nos ubica temporalmente en una madrugada neblinosa.

Más obvio es “moneda o botón”. Habla de hacer trampa, de pasar un botón donde debería haber una moneda. No se puede hacer eso con el cajero, ni con un vendedor, pero sí se puede en la limosna del “ciego bilingüe”.

“paso morales” es obvia. Es la calle Paso Morales, en Villa Tesei. De hecho esa calle corta la vía del tren que viene de Chacarita, lo que confirma que hablamos de un viaje por las vías.

Es difícil justificar “sin la opalina”. De hecho, no lo voy a intentar. “de Retiro a Pilar / busca el chanco al chabón” es tal vez la línea más obvia: es un guarda de tren que busca a un pasajero que no paga, y lo persigue desde Retiro a Pilar. El problema con esa obviedad es que:

- 1) El tren que hace Retiro/Pilar no pasa por Hurlingham
- 2) El tren que corta Paso Morales sale de Chacarita (y sí pasa por Hurlingham).

¿Es tal vez que “De Chacarita a Hurlingham” es imposible desde un punto de vista de métrica? Sería comprensible si así fuera.

“Sapo explota en San Martín / los domingos a las diez” es oscura. Además de que ninguna de las líneas de ferrocarril mencionadas pasa por San Martín. ¡Pero el recorrido Retiro/Pilar es de la línea San Martín! Si bien no logramos esclarecer qué sapo explota, porqué ni dónde, sí sabemos el cuándo. Esto fortalece la hipótesis de que el paisano por algún motivo está yendo a Pilar.

También es sanmartiniana la referencia al “sable recto en la estación”, por contraposición al famoso sable corvo del General. Que nunca jamás tuvo el grado de Mayor (ascendió de capitán a general), lo que complica encasillar “berretín de mayor”.

“Canilla en el andén / gotea noticias / te grita el titular / mentiras sin picar” es directa. Un canillita, un canilla, es un vendedor de diarios. Las canillas gotean, los canillitas gotean noticias. Gritan los titulares (aunque creo que ningún canillita grita los títulos desde 1947 o algo así). Mentiras sin picar, porque el papel no está picado, todavía, porque es un diario de hoy.

Y llegamos a la estrofa final, “Abejas con ombu / viajando en el panal / va la timba en el furgón.” Me resisto a dar una interpretación, mas allá de que las abejas en el panal van apretadas, y que ombú es una marca de papel para armar cigarrillos, que seguramente la gente de la banda ha usado en abundancia para drogarse, un ingrediente que sospecho importante en la escritura de sus letras.

¿Qué conclusión podemos sacar de este análisis? Bueno, yo, personalmente, preferiría no entender lo que dicen, que Divididos fuera una banda de rock eslovaco, y poder sentir la patada en la frente que es esta canción sin tratar de entender qué carajo es “sapo explota en san martín”. Pero eso es un problema mío.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “aquí”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “aquí”.

UPDATE: Algunas aclaraciones, datos extras, y un tema de rock checo, [aquí](http://lateral.netmanagers.com.ar/weblog/posts/BB988.html). aquí:
<http://lateral.netmanagers.com.ar/weblog/posts/BB988.html>
Again: spanish only!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “mi post de ayer”.

Después de [mi post de ayer](#).. mi post de ayer: <http://lateral.netmanagers.com.ar/weblog/posts/BB988.html>

acerca de la letra de “Paisano de Hurlingham” recibí un aluvión de correcciones y explicaciones, que enumero a continuación.

Opalina:

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “en flickr”.

Es una referencia a Opalinas Hurlingham, una fábrica abierta en 1948, y abandonada desde 1994. Hay una interesante colección de fotos de su interior [en flickr](#).. en flickr: <http://www.flickr.com/photos/52996436@N04/sets/72157625596663926/>

La empresa fué a la quiebra principalmente por un juicio pionero sobre daño ambiental: envenenaba las napas de la zona con arsénico.

La línea de tren: Es el ramal Retiro/Pilar del ferrocarril San Martín, que efectivamente pasa por Hurlingham. Mea culpa.

Abejas con ombú: Las abejas son trabajadoras. Ombú es la marca más conocida de ropa de trabajo en Argentina. Habla tal vez mal de mí que no se me ocurriera.

Sapo explota: Mi mamá hacía explotar sapos forzándolos a fumar, cuando era chica. No he oído otra referencia a sapos explosivos.

Sigue siendo un misterio el porqué el sapo explota los domingos a las 10.

Berretín de mayor: Jerga zonal acerca de ser un malandra.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “rock checo”.

Me acercan [rock checo](#).. rock checo: http://www.youtube.com/watch?v=CEI2_XEF_-0&feature=related, aunque no eslovaco.

Desde ya muchas gracias por sus aportes!

I have recently enabled tarpitting in one of my customer’s mail servers.

Tarpitting is adding a small delay after each recipient (after a certain number of them). The idea is that a message with a few recipients goes fast, a message with many goes slow.

So, it should make spammers less efficient.

Some mail administrators say tarpitting doesn’t work. That spammers, instead of sending a zillion mails over one connection, send a few over each of a zillion connections.

But a zillion connections are more expensive for the spammer! Or at least slower.

Well, I have proof that it **does** work. Sure, something like a per-ip limit on concurrent SMTP connections is a good complement, but even naïve tarpitting, all by itself, has a good effect.

Since I enabled it, peak message rate is down 60%, average is down 40%. Pretty good!

But a picture is worth 1000 words...

There is one area where Qt and Python (and in consequence PyQt) have major disagreements. That area is memory management.

While Qt has its own mechanisms to handle object allocation and disposal (the hierarchical QObject trees, smart pointers, etc.), PyQt runs on Python, so it has garbage collection.

Let's consider a simple example:

```
\ :kn:'from'\ \ :nn:'PyQt4'\ \ :kn:'import'\ \ :n:'QtCore'\
\ :k:'def'\ \ :nf:'finished'\ \ :p:'()':\
  \ :k:'print'\ \ :s:'"The process is done!"'\
  \ :c:'# Quit the app'\
  \ :n:'QtCore'\ \ :o:'.''\ \ :n:'QCoreApplication'\ \ :o:'.''\ \ :n:'insta

\ :k:'def'\ \ :nf:'launch_process'\ \ :p:'()':\
  \ :c:'# Do something asynchronously'\
  \ :n:'proc'\ \ :o:'='\ \ :n:'QtCore'\ \ :o:'.''\ \ :n:'QProcess'\ \ :p
  \ :n:'proc'\ \ :o:'.''\ \ :n:'start'\ \ :p:'('\ \ :s:'"/bin/sleep 3"'\ \
  \ :c:'# After it finishes, call finished'\
  \ :n:'proc'\ \ :o:'.''\ \ :n:'finished'\ \ :o:'.''\ \ :n:'connect'\ \ :p

\ :k:'def'\ \ :nf:'main'\ \ :p:'()':\
  \ :n:'app'\ \ :o:'='\ \ :n:'QtCore'\ \ :o:'.''\ \ :n:'QCoreApplication
  \ :c:'# Launch the process'\
  \ :n:'launch_process'\ \ :p:'()'\
  \ :n:'app'\ \ :o:'.''\ \ :n:'exec_'\ \ :p:'()'\

\ :n:'main'\ \ :p:'()'\
```

If you run this, this is what will happen:

```
QProcess: Destroyed while process is still running.
The process is done!
```

Plus, the script never ends. Fun! The problem is that `proc` is being deleted at the end of `launch_process` because there are no more references to it.

Here is a better way to do it:

```
\ :kn:'from'\ \ :nn:'PyQt4'\ \ :kn:'import'\ \ :n:'QtCore'\
\ :n:'processes'\ \ :o:'='\ \ :nb:'set'\ \ :p:'([])'\
\ :k:'def'\ \ :nf:'finished'\ \ :p:'()':\
```

```

    \ :k:'print' \ \ :s:'"The process is done!"' \
    \ :c:'# Quit the app' \
    \ :n:'QtCore' \ \ :o:'. ' \ \ :n:'QCoreApplication' \ \ :o:'. ' \ \ :n:'install' \ \ :o:'. ' \ \ :n:'main' \ \ :p:'()' \

\ :k:'def' \ \ :nf:'launch_process' \ \ :p:'()': \
  \ :c:'# Do something asynchronously' \
  \ :n:'proc' \ \ :o:'=' \ \ :n:'QtCore' \ \ :o:'. ' \ \ :n:'QProcess' \ \ :p:'()' \
  \ :n:'processes' \ \ :o:'. ' \ \ :n:'add' \ \ :p:'(' \ \ :n:'proc' \ \ :p:')' \
  \ :n:'proc' \ \ :o:'. ' \ \ :n:'start' \ \ :p:'(' \ \ :s:'"/bin/sleep 3"' \ \ :p:')' \
  \ :c:'# After it finishes, call finished' \
  \ :n:'proc' \ \ :o:'. ' \ \ :n:'finished' \ \ :o:'. ' \ \ :n:'connect' \ \ :p:'()' \

\ :k:'def' \ \ :nf:'main' \ \ :p:'()': \
  \ :n:'app' \ \ :o:'=' \ \ :n:'QtCore' \ \ :o:'. ' \ \ :n:'QCoreApplication' \ \ :p:'()' \
  \ :c:'# Launch the process' \
  \ :n:'launch_process' \ \ :p:'()' \
  \ :n:'app' \ \ :o:'. ' \ \ :n:'exec_' \ \ :p:'()' \

\ :n:'main' \ \ :p:'()' \

```

Here, we add a global `processes` set and add `proc` there so we always keep a reference to it. Now, the program works as intended. However, it still has an issue: we are leaking `QProcess` objects.

While in this case the leak is very short-lived, since we are ending the program right after the process ends, in a real program this is not a good idea.

So, we would need to add a way to remove `proc` from `processes` in `finished`. This is not as easy as it may seem. Here is an idea that will not work as you expect:

```

\ :k:'def' \ \ :nf:'launch_process' \ \ :p:'()': \
  \ :c:'# Do something asynchronously' \
  \ :n:'proc' \ \ :o:'=' \ \ :n:'QtCore' \ \ :o:'. ' \ \ :n:'QProcess' \ \ :p:'()' \
  \ :n:'processes' \ \ :o:'. ' \ \ :n:'add' \ \ :p:'(' \ \ :n:'proc' \ \ :p:')' \
  \ :n:'proc' \ \ :o:'. ' \ \ :n:'start' \ \ :p:'(' \ \ :s:'"/bin/sleep 3"' \ \ :p:')' \
  \ :c:'# Remove the process from the global set when done' \
  \ :n:'proc' \ \ :o:'. ' \ \ :n:'finished' \ \ :o:'. ' \ \ :n:'connect' \ \ :p:'()' \
  \ :c:'# After it finishes, call finished' \
  \ :n:'proc' \ \ :o:'. ' \ \ :n:'finished' \ \ :o:'. ' \ \ :n:'connect' \ \ :p:'()' \

```

In this version, we will still leak `proc`, even though `processes` is empty! Why? Because we are keeping a reference to `proc` in the `lambda`!

I don't really have a good answer for that that doesn't involve turning everything into members of a `QObject` and using `sender` to figure out what process is ending, or using `QSignalMapper`. That version is left as an exercise.

- Things you only like or believe because your mom said so.
- Tips for Time Travelers.
- Cute plants and their antics.
- 1001 ways to peel a cat.
- Things morticians say.

- Traveling for Time Tippers.
- Coins of the world: what do they taste like?
- Things found in people's noses.
- Surprise, that is not chicken!
- Time for Tip Travelers.
- World of Lint.

I love <http://goodreads.com> very much. It has measurably improved my life as a reader. I have read authors I wouldn't have read without it, books from those authors I would have ignored, and keeps track of what I read, am reading and will read.

What it has never been for me, is a social network. I would be about as happy with it if I knew noone else on the site, if it were just me and a bazillion strangers whose taste I can leech off.

Sure, I have a few friends there nowadays, but I hardly ever do anything "social" beyond accepting requests and posting reviews which I have no idea if someone reads.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "flickr".

I love Flickr.. flickr: <http://flickr.com>

where I put most of my pictures (soon: all of my pictures). It's cheap and I can upload an almost infinite amount of pics there, and I can share them with friends and family sometimes (by reposting them to facebook).

They were even kind enough to store the pictures I uploaded as a free user until I paid for the space to store them 5 years later.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "twitter".

I love Twitter.. twitter: <http://twitter.com>

because it's a place to post short things that don't deserve a blog post, to chatter with friends and not-so-friends, to know more people, and to waste some time every day.

One of those things is not like the others. One of those things I use for its social features, the others I use for other reasons, and I don't really care about them being social or not.

I think nowadays, for a social network to succeed, it has to cater to the antisocial, at least at first, when you know noone there. I don't go to Flickr to debate. I don't go to Goodreads to chat. I go there to put pictures and keep my books straight. And that's what kept me there long enough to meet people.

Some of you may know I have written about 30% of a book, called "Python No Muerde", available at <http://nomuerde.netmanagers.com.ar> (in spanish only). That book has stagnated for a long time.

On the other hand, I wrote a very popular series of posts, called PyQt by Example, which has (you guessed it) stagnated for a long time.

The main problem with the book was that I tried to cover way too much ground. When complete, it would be a 500 page book, and that would involve writing half a dozen example apps, some of them in areas I am no expert.

The main problem with the post series is that the example is lame (a TODO app!) and expanding it is boring.

So, what better way to fix both things at once, than to merge them!

I will leave Python No Muerde as it is, and will do a new book, called PyQt No Muerde. It will keep the tone and language of Python No Muerde, and will even share some chapters, but will focus on developing a PyQt app or two, instead of the much more ambitious goals of Python No Muerde. It will be about 200 pages.

I have acquired permission from my superiors (my wife) to work on this project a couple of hours a day, in the early morning. So, it may move forward, or it may not. This is, as usual, an experiment, not a promise.

I am in general allergic to HTML, specially when it comes to parsing it. However, every now and then something comes up and it's fun to keep the muscles stretched.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "really nice table".

So, consider the Ted Talks site. They have a [really nice table](http://www.ted.com/talks/quick-list).. really nice table: <http://www.ted.com/talks/quick-list>

with information about their talks, just in case you want to do something with them.

But how do you get that information? By scraping it. And what's an easy way to do it? By using Python and BeautifulSoup:

```
\ :kn:'from' \ \ :nn:'BeautifulSoup' \ \ :kn:'import' \ \ \ :n:'BeautifulSoup'
\ :kn:'import' \ \ \ :nn:'urllib'

\ :c:'# Read the whole page.'
\ :n:'data' \ \ :o:'=' \ \ \ :n:'urllib' \ \ :o: '.' \ \ \ :n:'urlopen' \ \ :p:'('
\ :c:'# Parse it'
\ :n:'soup' \ \ :o:'=' \ \ \ :n:'BeautifulSoup' \ \ :p:'(' \ \ \ :n:'data' \ \ :p:

\ :c:'# Find the table with the data'
\ :n:'table' \ \ :o:'=' \ \ \ :n:'soup' \ \ :o: '.' \ \ \ :n:'findAll' \ \ :p:'('
\ :c:'# Get the rows, skip the first one'
\ :n:'rows' \ \ :o:'=' \ \ \ :n:'table' \ \ :o: '.' \ \ \ :n:'findAll' \ \ :p:'('

\ :n:'items' \ \ :o:'=' \ \ \ :p:'[]'
\ :c:'# For each row, get the data'
\ :c:'# And store it somewhere'
\ :k:'for' \ \ \ :n:'row' \ \ \ :ow:'in' \ \ \ :n:'rows' \ \ \ :p: ':'
\ \ \ :n:'cells' \ \ \ :o:'=' \ \ \ :n:'row' \ \ \ :o: '.' \ \ \ :n:'findAll' \ \ \ :p:
\ \ \ :n:'item' \ \ \ :o:'=' \ \ \ :p:'{'}
```

```

\ :n: 'item' \ \ :p: '[' \ \ :s: ''date' \ \ :p: ']' \ \ \ :o: '= ' \ \ :n: 'cells
\ :n: 'item' \ \ :p: '[' \ \ :s: ''event' \ \ :p: ']' \ \ \ :o: '= ' \ \ :n: 'cell
\ :n: 'item' \ \ :p: '[' \ \ :s: ''title' \ \ :p: ']' \ \ \ :o: '= ' \ \ :n: 'cell
\ :n: 'item' \ \ :p: '[' \ \ :s: ''duration' \ \ :p: ']' \ \ \ :o: '= ' \ \ :n: 'c
\ :n: 'item' \ \ :p: '[' \ \ :s: ''links' \ \ :p: ']' \ \ \ :o: '= ' \ \ :p: '[' \ \
\ :n: 'items' \ \ \ :o: '.' \ \ \ :n: 'append' \ \ :p: '(' \ \ \ :n: 'item' \ \ :p: ')' \ \

```

And that's it! Surprisingly pain-free!
And she's still here.

Two nights ago, I was at a birthday party. In it, someone (name preserved because you don't know the guy) said something like "symbols are useful to rally people for a fight".

I answered "if I ever need to rally people to a fight, I would make sure to choose some uncool symbol (maybe a naked mole rat, rampant), so I at least know they are there because they understand the issue, and not because of cool marketing". Which is probably a pretty stupid thing to say, but:

- a) I was on my 3rd fernet
- b) I was trolling the other guy
- c) Fernet troll is drunk
- d) I kinda believe it

It's stupid because only accepting the help of true believers and not actively trying to make people believe are good ways to ensure you have a very motivated tiny minority on your side (Linux Year of The Desktop analogy ges here).

But I kinda believe it anyway, because being ugly and loved is warmer than being pretty and loved, because being difficult and appreciated is more valuable than being accessible and liked. But that only works if you are truly difficult, and not intentionally so, that's just being a poseur. And truly ugly, not Charlize-Theron-on-fat-makeup ugly.

So, to what point is any of us honestly annoying? That, friends, is the issue here. And if I ever have to lead others into something, I hope I do it as myself, and they come because of a reason that is theirs and not mine.

I am not here to convince people. I am here to know people.

So, you are writing a PyQt app, and you want it to support command line arguments. So you do something like this:

```

system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "a bunch of useful command line options already".

```

Or maybe even `QApplication([])`. Ok, you are doing it wrong. And this is wrong in most tutorials, too. Why? Because Qt (and thus PyQt) supports a bunch of useful command line options already: <http://developer.qt.nokia.com/doc/qt-4.8/qapplication.html#QApplication>

. So if you do it like in the first listing, and pass "--style=oxygen" or whatever, one of the following will happen.

- a) OptParser is going to tell you it's not a valid option and abort
- b) You will ignore the option and not do anything useful with it

c) You will have your own `-style` option and do *two* things with it

All three outcomes are less than ideal.

The right way to do this is:

This way, you give PyQt a chance to process the options it recognizes, and, then, you get to handle the rest, because `app.arguments()` has all Qt options removed.

The bad side of this is, you will make `--help` slightly slower, since it will have to build a `QApplication` to do nothing, and you will have undocumented options. Solution for both problems left as an exercise.

Sorry, spanish-only post, because I am *so* not translating this.

UPDATE: Resulta que (al contrario de lo que digo abajo) PABLOOO no es pythonista7 si no un amigo que lo ayuda para que no pierda el año en la facultad.

Siempre en las listas de tecnología alguien te tira algo como el viejo “Cómo hacer preguntas inteligentes” y cosas así. Es útil porque, asombrosamente, hay muchos que no son capaces (o no les sale) de pedir ayuda de manera que *permita* que te ayuden.

Los demás no tienen idea de qué estás hablando. Entonces necesitás explicar desde cero. Si tirás un bodeque de código de 400 líneas y decís “no anda, arréglenlo”, nadie se va a molestar siquiera en leerlo.

Todos están ayudando gratis. Tenés que recompensarlos de alguna manera, aunque sea haciéndoles sentir que el problema es interesante, o que están ayudando a alguien que lo aprecia y los escucha.

Tenés que darle pelota a lo que te dicen, si no, es frustrante. Si es frustrante, y no te pagan, al carajo.

No podés pedir “hagan mi trabajo” o “hagan mi tarea” o “hagan este práctico”. Eso lo tenés que hacer **vos**. Como mucho te va a ayudar alguien con alguna cosa que te trabes, como te ayudaría tu profesor/compañero de trabajo.

No tenés que enojarte si no obtenés lo que querés. Puede ser que sea porque es muy difícil y los demás tampoco lo saben, o porque justo está todo el mundo ocupado, o porque sos un hinchapelotas que nadie quiere ayudar. Casi seguro que no es que están todos *cogiendo secretarias*.

Sin más preámbulo, les presento a pablitoooooo*, el pythonista número 7, comunista revolucionario, recuperador de su plusvalía por el mecanismo de pedir prestada la de los demás. No se han cambiado los nombres porque nadie ahí es inocente ;-)

Nota: PABLO* y pythonista7 son la misma persona, ok?

```
[20:45:47] <PABLOOO> holassssssssssss
[20:46:06] <PABLOOO> gentes...tengo q presentar mañana este juego
[20:46:09] <PABLOOO> http://pastebin.lugmen.org.ar/7304
[20:46:11] <lalita> PABLOOO: [#3225] pastebin.lugmen.org.ar - El pastebin m
[20:46:25] <PABLOOO> no va ni pa tras ni pa adelante
[20:46:37] <PABLOOO> me podran dar una mano....plizzzzzzzzzzzz
[20:52:56] <PABLOOO> holassssssssss
[20:53:03] <PABLOOO> hay alguien????
[20:53:53] <alecu> PABLOOO, hola
```

[20:54:06] <PABLOOO> holasss...como va???
 [20:54:12] <alecu> PABLOOO, lo lamento, pero mi religión no me permite hacer
 [20:54:22] <PABLOOO> nuuuuuuuuu
 [20:54:30] <PABLOOO> y si cambias de religion???
 [20:54:52] <alecu> PABLOOO, y si cambiás la actitud? :-)
 [20:55:04] <alecu> PABLOOO, si preguntás por cosas específicas, seguro que
 [20:55:04] <PABLOOO> ...no pego 1...
 [20:55:27] <PABLOOO> lo puse aca a las 3 o 4 de la tarde...me digeron una p
 [20:55:33] <PABLOOO> no saque nada
 [20:55:43] <alecu> PABLOOO, empezá chiquito. Es decir, andá arreglando de a
 [20:57:24] <PABLOOO> tengo ese while de..... no deja cargar al otro while.
 [20:57:47] <alecu> PABLOOO, aha.
 [20:57:47] <PABLOOO> alguna sugerencia...
 [20:57:49] <alecu> PABLOOO, No entendí nada de lo que dijiste recién.
 [20:57:59] <PABLOOO> el while...
 [20:58:26] <alecu> PABLOOO, hay 13 whiles en tu código
 [20:58:27] <PABLOOO> me quede ahi...tengo un juego ...tengo q ponerle vidas
 [20:58:58] <alecu> PABLOOO, en que curso tenés que entregarlo?
 [20:59:22] <PABLOOO> fundamentos de programacion unaj
 [20:59:35] <alecu> PABLOOO, en que universidad o carrera?
 [20:59:43] <PABLOOO> sacamelo y me acuerdo de vos con mi profe
 [20:59:47] <PABLOOO> XDDDDDD
 [21:00:16] <alecu> PABLOOO, si yo fuera tu profe no podría aprobarte. :-)
 [21:00:34] <ralcina> PABLOOO: decí una cosa sola que no anda, lo mas clara
 [21:00:37] <PABLOOO> naaaaaaaaaa....pero tengo la intencionnn
 [21:00:44] <ralcina> PABLOOO: pero no podes esperar que te hagan los trabaja
 [21:01:01] <PABLOOO> pero mañana es la final...o fui
 [21:02:02] <PABLOOO> el problema es la def reglas...juego corria bien
 [21:02:47] <ralcina> PABLOOO: esa funcion tiene casi 200 líneas.
 [21:02:53] <PABLOOO> en def reglas...yo quiero meterle la opcion de q me ca
 [21:03:18] <PABLOOO> dame 1 minuto y lo pienso bien
 [21:03:19] <ralcina> PABLOOO: no tengo idea ni de que juego se supone que e
 [21:03:32] <PABLOOO> ok...toy en eso
 [21:04:12] <alecu> ok, me llaman a cenar...
 [21:04:20] <alecu> PABLOOO, después contanos a la lista como te fue.
 [21:04:45] <PABLOOO> okassssss
 [21:05:47] <rbistolfi> PABLOOO: trata de identificar bien el problema, no s
 [21:18:52] <rbistolfi> Hau yottabit
 [21:25:35] <rbistolfi> PABLOOO: che, esta dificil de leer ese codigo :[
 [21:25:53] <PABLOOO> jajajajajaja....yo soy asi
 [21:26:11] <rbistolfi> igual fijate que "fin" esta definido afuera de la fu
 [21:26:24] <rbistolfi> y despues la definis denuvo, pero en el scope local
 [21:26:29] <PABLOOO> creo q lo complique mucho...y esto en la recta final..
 [21:26:41] <rbistolfi> entendes la diferencia entre local y global?
 [21:26:56] <PABLOOO>
 [21:27:00] <rbistolfi> tambien fijate que los loops tienen que tener una co
 [21:27:41] <rbistolfi> no encuentro donde "f" se utiliza luego para salir o
 [21:27:52] <rbistolfi> o no entiendo lo que quieres hacer :)
 [21:27:56] <PABLOOO> me anote para algoritmo...y no se q ...capaz q en un p
 [21:28:04] <PABLOOO> por q hoy...

[21:29:14] <rbistolfi> PABLOOO: en la linea 181
[21:29:39] <PABLOOO> pera
[21:29:42] <rbistolfi> vos decis que vas a iterar hasta que f sea True
[21:30:38] <rbistolfi> bueno, me parece que f nunca se convierte en True
[21:30:38] <rbistolfi> despues usas "fin"
[21:30:38] <rbistolfi> asi que me parece te confundiste con "f" y "fin"
[21:32:26] <rbistolfi> yottabit: so, cuando venis a Argentina?
[21:34:14] <PABLOOO> f con fin
[21:34:18] <PABLOOO> toy en eso
[21:53:31] <pythonista7> lo podes terminar,es un trabajo muy important y lo
[21:53:50] <rbistolfi> pythonista7: ?
[21:56:30] <pythonista7> <http://pastebin.lugmen.org.ar/7304>
[21:56:31] <lalita> [#3225] <http://pastebin.lugmen.org.ar/7304> : pastebin.l
[21:56:39] <yottabit> *toser*
[21:56:54] <pythonista7> este es el ejercicio,en realidad un jueguito simpl
[21:57:46] <rbistolfi> yottabit: ya somos dos :P
[21:58:45] <pythonista7> ponganse media pila,lo sacan asi nomas o no???
[21:58:59] <rbistolfi> pythonista7: no
[21:59:02] <rbistolfi> sorry
[22:00:30] <rbistolfi> no creo que tengas suerte aca, pidiendo que te hagan
[22:01:31] <rbistolfi> mira
[22:02:11] <pythonista7> y no
[22:02:14] <pythonista7> creo q no
[22:02:21] <pythonista7> si se rascan a cuatro manos
[22:02:44] <pythonista7> o estan cogiendo con alguna secretaria.....:P
[22:02:58] <rbistolfi> si es tan importante, porque no estudias y lo haces
[22:04:00] <pythonista7> lo estudie,y lo se,solo kier una opinion
[22:04:06] <pythonista7> pero se ve q es al pedo
[22:04:21] <rbistolfi> ya te dije unas cosas que estaban mal
[22:04:33] <rbistolfi> lo del "fin" y "f"
[22:04:36] <pythonista7> x gent como esta el pais esta asi,cero solidaridad
[22:04:38] <rbistolfi> lo arreglaste?
[22:04:48] <pythonista7> deja q busco x otro lado
[22:05:01] <pythonista7> no hay drama,fuck!!!
[22:05:05] <rbistolfi> gracias
[22:06:00] <rbistolfi> le estoy contando a la Vane sobre este pibe, y no me
[22:06:05] <TiN> juaz cero solidaridad xD
[22:10:49] <gatox> ahhh bue

1 hora mas tarde...

[23:36:19] <PABLOOO> holassssssssssss
[23:36:24] <PABLOOO> yo otra vezzzzzzzzzz
[23:36:31] <PABLOOO> hay alguien?????
[23:39:09] <PABLOOO> holassssssssssss
[23:39:17] <PABLOOO> queda alguien...
[23:39:23] <PABLOOO> todos durmiendo
[23:39:27] <PABLOOO> zzzzzzzzzzzz
[23:43:47] <-- PABLOOO (ba8695e7@gateway/web/freenode/ip.186.134.149.231) h
[23:44:25] --> PABLOOO (ba8695e7@gateway/web/freenode/ip.186.134.149.231) h

```
[23:44:35] <PABLOO0> holaaaaaaaaaaaaassssssss  
[23:44:46] <PABLOO0> hay alguien????  
[23:46:47] <PABLOO0> holasssssssss
```

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “very far from each other”.

When we were not married, Rosario and I lived [very far from each other](http://g.co/maps/fkxcu).. very far from each other: <http://g.co/maps/fkxcu>

which meant that when we met, we stuck together for 24, or 36 hours. Our 3rd date lasted almost 3 days.

A few days ago, we decided to have one of those, even though we were with Tato for a part of it.

So, we went from home to the Museo Participativo de Ciencias, in Recoleta, a cool kids science museum, full of cool stuff.

First we went to Buenos Aires Design, place full of cool and very overpriced designer stuff. Bought two pens.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “*very* far away from us”.

Since Georgina, a friend of ours who lives [*very* far away from us](http://g.co/maps/psb4t).. [*very* far away from us](http://g.co/maps/psb4t): <http://g.co/maps/psb4t>

works at the Centro Cultural Recoleta, we decided to kidnap her for a cup of coffee

In the meantime, Tato acquired two balloons and started making inappropriate things with them:

But that was just a moment. Then... **I’m an elephant!!!!**

We entered the Centro Cultural Recoleta, which is a very cool place...

Tato and us performed a bunch of experiments. He even got to play with a Vandergraaf generator!!!!

Then we wandered the scary areas...

When we got out, it was getting dark...

And Tato discovered the sad truth about candied apples: they are crap...

We left tato with his aunties Laura and Agustina and started the grownup part of the date. First, we saw Anonymous protesting scientology (and a trollface!)

We had a picada at a place called Babieca. Great mortadela! I saw an example of rampant capitalism...

Then our bank heist was foiled by a teller not giving us the “small denomination, unmarked, not-sequential” bills we wanted...

Turns out Rosario had never been to the Ateneo Grand Splendid, the prettiest bookstore I have ever seen. So we went there to see some books and get a soda or something...

The roof is pretty special...

This is a restored theater, and the bar is in the stage...

And you can see the ancient lights board (luckily: disabled)...

So, we walked on Corrientes avenue and saw this (and yes, you can fake 7 seconds of tango by following the steps, but it's very hard for the lady)...

Then there was a guy, walking in an elephant's trunk thong, while his friends (who had shorn him and covered him in flour) whistled and shouted: a "despedida de soltero", ritual hazing for a guy getting married...

He ended talking with a 7-foot-tall transvestite in red sequins. The transvestite shouting "you are a healthy boy!"...

So, after a nice evening at a borrowed apartment, next morning we got Tato back! A hobbitsian 3-hour breakfast ensued...

Then taxi, train home, and we got back almost exactly 26 hours after we left. Everyone had fun, and thanks to Agustina and Laura (or Laura and Agustina), because without the mellis, this would have been impossible.

In a whim, I checked out kbindings/dcopython from KDE's CVS.

I see the README: dcopython is broken

Then I said to myself: maybe I can fix it. And you know what? It seems to be **not** broken! :-)

At least for simple data types, that is.

dcopython lets your python program become a DCOP server or client.

A DCOP server is capable of being controlled by KDE's kdcop, and is a very simple way to make your application externally scriptable.

A DCOP client is something that contacts a DCOP server, so that means you can control and script KDE applications (or other DCOP servers) from python scripts.

The neatest thing here is that this stuff doesn't require Qt!

I intend to use it to make some of my apps externally scriptable without PyKDE.

It should be obvious that it is not possible to have a job, a girlfriend, a hobby and a blog.

So, I have been neglecting the blog ;-)

Expect more updates now, since I am scaling down the job thing a bit.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "posting about how a sony clie would be a much nicer ebook reader than my palm" m100.: <http://lateral.netmanagers.com.ar/weblog/posts/P199.html>

In a day like this, in 2004, I was [posting about how a Sony Clie would be a much nicer ebook reader than my palm](http://lateral.netmanagers.com.ar/weblog/posts/P199.html) m100.: <http://lateral.netmanagers.com.ar/weblog/posts/P199.html>

Really. This:

Figure 31: A sony Clie SJ-30

Was, indeed much better than this:

Figure 32: A Palm m100

And I was **happy** that I could get the Clie for only 100 dollars after trading my m100. And it had a battery life of a whole **2 days**. Nowadays, for the same 100 dollars, I got a Kindle touch, with a battery life of, like, forever.

OTOH, at night I read on an android phone, which is about the same size and shape as the Clie and has even shorter battery life (much better screen of course).

So, the more things change, the more they stay the same, I guess.

In 2006 I republished an essay about the Lord Of The Rings and sysadmins. It's as painful as it sounds.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "here".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "here".
```

You can read it [here](http://lateral.netmanagers.com.ar/stories/41.html).. here: <http://lateral.netmanagers.com.ar/stories/41.html>
Spanish-only post, sorry!

Hace como dos años escribí un pedazo de un libro. De vez en cuando lo miro con cariño y pienso que estaría bueno terminarlo, o redefinirlo y cerrarlo, y cosas así. Hasta hice un plan que nunca pude poner en práctica porque la vida te lleva a hacer cosas distintas.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "iwi-131 programación de computadores - 1er semestre 2012".
```

Resulta que están usando mi humilde cacho de libro como material de estudio en la materia [IWI-131 Programación de computadores - 1er Semestre 2012](http://progra.usm.cl/index.html).. [iwi-131 programación de computadores - 1er semestre 2012](http://progra.usm.cl/index.html): <http://progra.usm.cl/index.html>

de la

```
system-message
WARNING/2 in everything.txt, line 32732
Block quote ends without a blank line; unexpected unindent. backrefs:
```

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “universidad técnica federico santa maría”.

[Universidad Técnica Federico Santa María.. universidad técnica federico santa maría:](http://www.usm.cl/)
<http://www.usm.cl/>

en Chile.

Por un lado, me pone contento. Por otro lado me pone nervioso que un libro que cita esto sea material de estudio para chicos de 18:

Hasta que cumple veinticinco, todo hombre piensa cada tanto que dadas las circunstancias correctas podría ser el más jodido del mundo. Si me mudara a un monasterio de artes marciales en China y estudiara duro por diez años. Si mi familia fuera masacrada por traficantes colombianos y jurara venganza. Si tuviera una enfermedad fatal, me quedara un año de vida y lo dedicara a acabar con el crimen. Si tan sólo abandonara todo y dedicara mi vida a ser jodido.

—Neal Stephenson (Snow Crash)

Por otro lado más, me da ganas de terminarlo. Por el último lado (con lo cual lo que tengo es una sensación cuadrangular), pienso: que raro sería que te den un libro en la facu que está todo incompleto. Es como hacerse fan de *The Event* y nunca enterarse que era el famoso *Event*, como haber visto *Twin Peaks* y nunca sacarse de encima la dudas de qué era toda esa bizarreada. Como seguir todavía esperando que Mel Brooks haga la segunda parte de la historia del mundo para poder entender “*Jews in Space*”.

Ahora podría no terminarlo como decisión artística!

Y por supuesto:

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “nikola”.

Alva is almost the opposite of [Nikola.. nikola: http://nikola.ralsina.com.ar](http://nikola.ralsina.com.ar)

. If Nikola is about making static sites, Alva is a dynamic site. However, as Hegel suggests, from the thesis and the antithesis comes the synthesis.

So, Alva is about dynamically creating static sites. If you want to have Nikola in your server instead of in your own computer, and have the convenience of an online tool, that’s the niche Alva tries to fill.

So, you would install Alva, and use it like any other web-based blogging tool. Yet, behind the scenes, you would have Nikola, and all the performance and security benefits of static sites.

And maybe someday, I (or someone) will put up a multi-user version of Alva, and you will be able to get hosted blogs, knowing all the data is yours and you can leave anytime and do your own thing.

This is very very early stages. So early it does not work yet. But here’s a teaser:

‘ <http://lateral.netmanagers.com.ar/galleries/random/nikola-admin3.png> ’ _There is no firm timeframe for this, it depends on a ton of other stuff and may not even happen.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “blog.txt theme”.

If you want a very minimalistic theme for a Nikola-based site, I just did a quick and dirty port of Scott Wallick’s [blog.txt theme](#).. blog.txt theme: <http://wordpress.org/extend/themes/blogtxt>

If there is anything nice here, he did it. If there is something wrong or broken, I did it instead.

I did it basically to see if it was possible to port wordpress themes to Nikola. And it is, but it involves reading php files and loosely reinterpreting them into Mako templates.

While the port is far from perfect, it’s a reasonable starting point for someone who is really interested in it.

Here is how it looks:

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “here is the download”.

‘<<http://lateral.netmanagers.com.ar/galleries/random/blogtxt.png>>‘_And [here is the download](#).. here is the download: `/static/nikola-blogtxt-theme.zip`

. To use it, just unzip it in your themes folder, set THEME to “blogtxt” in your `dodo.py` file, and rebuild the site.

At least the CSS files are easily adapted.

This theme is under a LGPL license (see included `license.txt`), enjoy!

Every day there is a new post decrying Apple’s tasteless use of skeuomorphism (you know, making calculator programs look like calculators and note-taking apps look like notepads?).

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “i said that in 2-thousand-freaking-four”.

I totally agree that skeuomorphic apps are ugly and stupid. [I said that in 2-thousand-freaking-four](#).. i said that in 2-thousand-freaking-four: <http://lateral.netmanagers.com.ar/stories/33.html>

. But just looking at the latest abomination (it seems to be a sound recorder that looks like a ree-to-reel, of all things) and sneering is worse, because that means you don’t have any ideas of where design comes from, and I say this being a person with as much taste as a walrus.

Design comes from people. There is a grander design behind that specific design, which you could call a guideline, or a philosophy, or in some cases a zeitgeist. For 50 years, there has existed a consensus about cleanliness of design being a good thing. It started in some specific niches while others went in other directions (car fins!) and later each area of design has moved, like a pendulum, towards cleanliness or “specialness”.

Once you go “clean”, and everyone goes “clean” there is very little you can do to make your product distinctive, and a tension is created to make it less clean and more “special”.

Google’s entry page used to be absolutely clean. A place to enter text, and two buttons. Now it has a menu with 11+ items, 3 buttons, and an icon. Apple’s OS9 was ascetic, and now OSX is a sea of bouncy colorful things shouting at you.

The skeuomorphism and other indications of overdesign, of complication, in apple's apps is not unintentional, it's an intentional attempt at making the applications special, appealing, and distinctive. It *is* ugly and awful, but it is so *intentionally*, because the very concepts of ugliness and awfulness are just a vague consensus among the users, and Apple surely felt confidence that users, accustomed to Apple's role as kings of taste, would *change* their taste to fit. And as far as I can see that is **exactly** what has happened.

Users are not the ones complaining about Apple's design style, *other designers* are complaining. That signals, to me, a disconnect between the taste of designers and the taste of users. And honestly, the taste of designers is only of vague academic interest to companies trying to sell product.

Apple's hardware stays minimalistic because they have successfully branded it. If you see a squarish slab of black glass with a button, you think iPad or iPhone depending on size, not "generic minimalistic touch device". On software, that did not work. There was nothing interesting or innovative, or distinctive in minimalistic design for applications.

So they started with colorful gumdrops, moved onto brushed metal, and then into fake stitched leather, because they are trying to find something that can be as successfully and powerfully branded as "silvery slim wedge with black keys" is now.

Designers apparently seem to believe there is certain specific "cleanliness" that is the hallmark of "good" design, and that ripped paper and other skeuomorphic affectations are signs of bad taste. That is silly and ahistoric. Cleanliness is just a fashion, reel-to-reel digital recorders are an attempt at *creating* a taste. It's ambitious, and respectable.

On the other hand, it is ugly as hell.

You know how I know we live in a unicode world now? Of the 300+ unicode arrows, all display correctly by default on Windows, Mac and Ubuntu, except maybe ~25.

← → ↑ ↓ ⇔

A few years ago, many apps and fonts were confused by simple letters with diacritics like á. So, yes, real progress happens.

Don't know how to handle this target <target dupnames="[u'nikola']" ids="[i'd1']" refuri="<http://nikola.ralsina.com.ar>"/> ..

```
system-message  
ERROR/3 in everything.txt, line 32885  
Unexpected indentation. backrefs:
```

title: A Year With My Kindle

I got my first Kindle a year ago. I quickly replaced it with my current and second kindle, a Kindle Touch.

So, how well has it worked? Pretty damn well. I am a fairly heavy user, I think, and the Kindle has traveled quite a bit, in bags, suitcases and cargo pockets. The only care I take is to use a leather cover when outside the house.

I have read, according to goodreads.com, somewhat over 17000 pages in this year, in 61 books. That's a lot of pages. And if you look at the gadget now, it still looks brand new. No scratches, everything operates correctly, even the battery still holds the charge fine even if it's down to about two weeks per charge instead of almost three.

I still miss the older kindle's page-turning buttons. Using a touchscreen to turn pages is idiotic. but hey, it works, and I can still do it one-handed (yay for huge-hand boy here!)

The only things I don't quite like are the same ones as when I bought it.

- The page has too little contrast when not ideally lighted.
- You can't read in the dark.

Since the new paperwhite fixes both of those, I am getting one. I have already sold this one, and the difference is not a lot, so it's a very cheap upgrade.

Quite happy about Amazon's ability to not suck at giving me goods in exchange for money, too! It's rare that I want a book and it's not out there in Kindle format (still waiting for Evan Dara's Easy Chain!)

So, no unexpected issues, has brought a lot of fun, was cheap... that's the definition of gadget paradise to me.

Last saturday I attended the 2nd Buenos Aires Chili Cookoff. Lots of people, lots of great food. I was with my kid and he can't eat anything that spicy, but hey, there were cookies :-)

Figure 33: How crowded was it? About three times this crowd was there.

I had little experience with Chili, because it's not exactly a common dish around here, so, surprised by the variety, and most were quite nice, and despite exhaustive warnings by the cooks, none was unedibly spicy (the dreaded Zombie Chili which promised to melt my mouth? Kinda sweet and bland.)

Favourite? Lafitte's revenge, which really was a sort of bean-free bourguignon. Also, the smoked veggie chili was quite awesome. And the cookies were killers!

Next year, my wife is talking about cooking for it. I have had her chili, and I trust she can do better than 80% of the contestants this year. Plus I get to beta test her recipes, so win/win for me.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "nikola's".
```

One of the easiest ways to personalize how your site looks is using color and typography. While Nikola's.. nikola's: <http://nikola.ralsina.com.ar>

“site” theme

```
system-message
WARNING/2 in everything.txt, line 32962
Block quote ends without a blank line; unexpected unindent. backrefs:
```

is **intended** to be rather bland and neutral, it doesn't *have* to be that way, and it's easy to change.

To do these changes, you don't even need to know any CSS, HTML, or programming!

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "customize page".
```

Here's the trick: Nikola is using Twitter Bootstrap to handle its styling. And they provide a handy web form to create a customized version, just for you, at their [customize page](http://twitter.github.com/bootstrap/download.html).. customize page: <http://twitter.github.com/bootstrap/download.html>

So, if you want auvergne navigation bars and avocado backgrounds, with courier fonts all over the place, that's where you do it. Just change the value in the right variable to whatever color you want.

Once you have your `bootstrap.zip`, go to your site's folder, and create `themes/mytheme/assets` and unzip it in there, so that you have `themes/mytheme/assets/css`, `themes/mytheme/assets/js`, etc.

Create a file called `themes/mytheme/parent` containing the word `site`.

Then edit your `dodo.py` (or `conf.py` if you are using the git master) and change the `THEME` option "mytheme".

Rebuild your site, and voilà, all your customizations are now in place.

This awfulness, for example, was done by setting just three variables (`bodyBackground`, `textColor`, and `sansFontfamily`):

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "this webcomic".
```

This is a response to a response to [this webcomic](http://theoatmeal.com/comics/religion).. this webcomic: <http://theoatmeal.com/comics/religion> titled

system-message

WARNING/2 in `everything.txt`, line 33002

Block quote ends without a blank line; unexpected unindent. backrefs:

“How to suck at your religion”. While Oatmeal’s comic is crass and paints things in broad terms, *it’s a freaking webcomic*. So it’s supposed to do that. But the response is so full of phallacies (and lacking in webcomic-ness) that it may deserve a response.

I have promised not to be a troll (anymore) so I will try to answer in a sensible manner.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “the article i am replying to”.

Here’s [the article I am replying to](http://catholicdefense.blogspot.com.ar/2012/07/a-ca).. the article i am replying to: <http://catholicdefense.blogspot.com.ar/2012/07/a-ca>

go read it

system-message

WARNING/2 in `everything.txt`, line 33011

Block quote ends without a blank line; unexpected unindent. backrefs:

if you want. I will not reply to all of it, but will instead cherrypick a couple of paragraphs.

In response to the “forcing dogma” panel:

So... religion is fine, unless you actually believe in it? Should parents not pass their political, ethical or moral views on to their children as well? What parts of parenting would be left if parents were to avoid passing their views on to their kids? The irony here is that silence is itself a statement. Avoiding any mention of God to your kids sends as clear a message as talking about God: specifically, it tells your kids that God’s existence is either untrue, unknown, or unimportant. Because if you knew Him to exist, surely you’d share that knowledge, right?

Let’s start from the top: you don’t know god exists. You have *faith* that he exists, but you don’t know it for a fact. If you knew for a fact that he exists, you could not possibly have faith because faith *excludes* certainty. As your bible says, faith is “the substance of things hoped for, the evidence of things not seen.”

So, do I tell my son god doesn’t exist? Nope. I tell him I think he doesn’t exist, and that I have never seen or heard of any reliable evidence or datum that points towards his existence, but also that some people do believe he does exist. I told him that because I feel that’s a honest answer. If your honest answer is “god exists”, then bully for you, but from the point of view of a non-believer you are telling your son a lie, or at best a half-truth. And if you really don’t know he exists *for a fact* then you are just lying.

Now, are you sayin that you know god exists factually? Based on what? That’s the usual slippery slope for this argument. The religious are the ones making statements of fact based on tradition. To the rest of us, they just seem to be playing loose with what “fact” means, or what “god” means or what “know” means.

So, no, don’t avoid mentions of god, just avoid lying to your kids if you can.

This next section is probably the worst, because it's just an incoherent argument. A kid asks, "Dad, what happens to us after we die?" The author compares providing the Christian answer to this question with correcting your kid for having green as a favorite color. What?? That just isn't a coherent argument. In what world are those two ideas parallel, or even comparable?

According to the webcomic, good parenting is to pretend to be agnostic, and say that "no one really knows for sure." Of course, if the Resurrection is true, that claim is false. So to be a good parent, you apparently have to deny the Resurrection and embrace agnosticism, treating beliefs about the afterlife as mere matters of personal preference like having a favorite color. This is just... stupid. There's just no other way of describing it. Imagine if we treated everything that way. "Dad, what's 3 x 3?" "No one really knows for sure. What do YOU think 3 x 3 is?"

So, comparing life after death with color preference is stupid and incoherent, but comparing it the christian belief of resurrection with basic arithmethics is a-ok? That must have taken some effort to write with a straight face, I'm sure.

So, let's go slowly on this one. Beliefs about the afterlife are, like most other beliefs, probably not a personal preference, but just something you have, because of, in most cases, indoctrination early in life, peer pressure, and just because you live in a society where that belief is normal and approved of.

But what is it your belief in the afterlife is *not*?

- It's not inherent to "you". If you were born in another place or time, you would probably believe something else.
- It's not undisputed. Because there exists a majority of people who don't believe the same thing, either by details or entirely.
- It's not unique. Because other religions have had similar resurrection beliefs.
- It's not reliable. Even if we were to accept everything the bible says as true that would not mean we know what will happen to you or to me after we die. We would have a testimony about what happened in a few days in the afterlife of a specific person, at a point in the past, as told to someone by someone. Is that the same as *knowing* what will happen? No it's not.

Let's compare that to 3x3 as the author attempted:

- If I was a chinese in the 12th century: 3x3 is 9.
- There is no group of people that believes 3x3 is 8 or 10.
- There has not been in the past any real disagreement about the value of 3x3. We have not achieved that result via a gradual improvement.
- We rely on 3x3 being 9 every day in our lives. If you drive a car, use a phone, or zip your pants, you are agreeing 3x3 is 9.
- We don't expect 3x3 not to be 9 in the future.

Notice any differences? Yes, me too.

Personally, I consider your faith in god more akin my liking Queen (the band, not the ruler). I was exposed to Queen at the right time, it was approved by my peers, and I like it. On the other hand, I understand that Queen is not everyone's cup of tea, and I don't claim Queen to be the "right" band.

The whole "if the Resurrection is true, that claim is false" line of thought is not logical. If my cat had wings, then the claim that winged cats are awesome is false. But my cat doesn't have wings. Does it make the winged cats less or more awesome that he doesn't? It's not that it's not right, it's that it's not even wrong.

Also, Oatmeal, shame on you about Galileo, really, look it up ;-)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "cloud atlas".

Finished reading [Cloud Atlas](http://www.goodreads.com/book/show/49628.Cloud_Atlas).. cloud atlas: http://www.goodreads.com/book/show/49628.Cloud_Atlas, gave it 5 stars. Here's a quick review:

I am not going to explain this book. It's enough, I think, to say I loved it, and that it's strange, and that it's a bit of a mystery.

Imagining a universe in which all the contents of the book could be real at the same time in a way that would allow all the pieces to be written as they are and yet, be, somehow, not the novel they are, but a found artifact, is both depressing and elusive.

At the end, I felt something I can only describe as retrospective hope, the feeling that things were supposed to end up better, but that even as terribly as they did end, were it not by that earlier hope, they would have been more grim.

The control the author has over his own style is impressive. This book feels written by half a dozen completely different writers.

Some quotes (which may only make sense once you read the book):

"The sun was deaf'nin' so high up, yay, it roared an' time streamed from it."

"In the first set, each solo is interrupted by its successor: in the second, each interruption is recontinued, in order. Revolutionary or gimmicky? Shan't know until it's finished, and by then it'll be too late"

"What wouldn't I give now for a never-changing map of the ever-constant ineffable? To possess, as it were, an atlas of clouds."

And then he told me "Roberto's a known troll" and I was like "really?" he seems like a nice guy!

—Not saying who said it

So, that happened a while ago in a dinner. And I am that troll. I must confess it, so here it is. I have a long, *long*, **long** history of trolling. Perhaps I could rationalize it into just being argumentative, or just enjoying a good argument, but no, really, I derive pleasure from seeing people react to what I say.

That could maybe be in some way constructive, I could just do it to make people see things from a different angle, and in my good moments I do that. But in others I just find someone I dislike, or someone who says something that rubs me the wrong way, and I ... well, it's not pretty.

What I usually do is up the ante. I escalate. I say outrageous things. Usually just slightly more outrageous than what the other guy says, so I can still rationalize it as "he started it", or "he deserves it" but honestly, deep inside, I know it's not true. I know I

am being nasty because I like it. And that sucks. It sucks that I have this mean streak in me, and I can't get rid of it. It sucks that I have *fun* insulting the other poor bastard.

I could even say that most of the people I troll are trolls themselves. But that's not really an excuse. If they deserved contempt for their trolling, then I am just as deserving for mine. I am sure they, also, have their reasons and those reasons sound perfectly rational to themselves.

I could say that I often troll those who spread poisonous speech. That I troll those who are spreading misinformation and evil ideas. And yes, that is true, but again, I could reply without trolling. I could reply rationally, and just point out where they are saying nonsense, where they are spreading stupidity, and avoid the flair, avoid the gratuitous insult.

I could say that I troll those who bother me personally out of their own free will. So yes, telling a Jehovah's Witness that I will be at the door as soon as I wash my hands of the blood sacrifice I make to Satan is funny, but why is it any better than just telling them I am not interested.

Oh, sure, I could say that telling someone who insulted me "opinions are like assholes, and so are you" is somewhat clever, but I could just say "you are wrong, rude, and undeserving of my attention".

I sure could say that trolling someone who is rude, insulting, obnoxious and disruptive may teach him (it's always a he, isn't he?) a lesson, but it's just not true. They just see those responses as challenges.

I could say that feeding the other troll is just harmless fun, but it's not true because I know it hurts people, and creates an obnoxious environment. I could say that trolling those who deserve to be trolled is fair. But it's not true because I am noone's judge.

So, if I have trolled you, accept this post as an apology. Not a "sorry if I offended you" apology, but a straight apology. I am sorry I trolled you. I will try to stop.

I will not stop saying and writing what I think, and I will not abstain from replying to other people's opinions, but I will do a honest try to do it in a not-trolling manner, of offering my opinion as it is, instead of as a performance to make the other one feel humiliated or hurt or undeserving.

Because, and here is my only piece of excuse: I hate bullies. I was sent to school two years too early. So that you have an idea how that can affect someone, I am of average height, but until I was 13 I was convinced I was very short. I managed to become a decent swimmer, but until I was 12 or so, I was convinced I sucked at all sports and games. I was increasingly bullied since I first ever met my "peers" until I was able to kick the crap out of them. I became a tough kid. I could take a beating like you can't imagine, because I was too small to hit back, and the only way not to be a total loser was to keep getting up and going to be beaten again, while making sarcastic comments.

And it turned out that those sarcastic comments worked. I got my ass kicked, but the next day that kid remembered how I was getting back on my feet and joking about his mother. And that, of course, led to further bullying. When I was about 12, I was so extreme, I scared everyone so much, they *stopped*. I was scary. I talked like a freaking psycho. I probably was close to being one.

And the lesson I learned was that if you make fun of people, they got hurt enough to want to avoid it later. So, I was, in fact, a bully myself. And thus, I became that which I hated. I have had better times and worst times, I have been more in control and not so much. And it has taken me a long time to figure this out. I don't like what I became. So I will stop. And I am sorry.

Just gentle ribbing and self-deprecation from now on. Promise.

Este sábado cumplo 41 (no, no hago fiesta, no es que no te invité), y creo que llegó el momento de decidir que voy a ser cuando sea grande.

Mi problema principal para decidir un tema tan crucial antes de hoy es que me interesan muchas cosas. Y las que no me interesan... sospecho que si le pusiera un poco de ganas me interesarían.

Pero peor, tengo esa convicción interna, totalmente injustificada de que si me pongo, puedo más o menos hacer casi cualquier cosa que no involucre una aptitud física específica.

O sea, no, nunca voy a ser power forward en la NBA, ni triatlonista, ni equilibrista, ni ninja. Pero... sospecho que podría ser un guitarrista mediocre, o un poeta medio pelo, un escritor adecuado, o un razonable peluquero, por no decir un competente paleontólogo, un interesante director de cine, un actor promedio, un interesante analista político, o un buen taxista.

Esa sospecha es infundada dado mi desinterés en la política, poesía, detalles de cómo se toca la guitarra, y no saber manejar, entre otras cosas.

Así que durante los próximos diez años, mientras me voy haciendo grande, me voy a concentrar en cosas que ya sé hacer, como... epa. Lo charlamos el año que viene.

These are the slides (in spanish) of my "Developing apps with Ubuntu One" from last saturday at UbuconLA. They filmed it, so there may be a video someday!

El viernes que viene me voy a París. Me llevo a mi mujer, a mi nene, y a mi vieja. Me llevo a mi mujer porque nunca tuvimos luna de miel, entonces cualquier excusa es buena, me llevo a mi nene porque tardé mucho, y me llevo a mi vieja porque se lo debo.

Mi mamá tiene 78 años y cuando estudiaba, pupila, en un colegio de monjas, estudiaba francés. Y cuando estudiás francés semipresa, las lecciones sobre París, el libro con fotos de la torre Eiffel... se me hace que deben haber sido material de los sueños de esa piba de quince. Yo sé que mi vieja sueña con este viaje hace más de sesenta años.

También lo soñaba mi viejo que siempre contaba de cuando tenía que escribir una redacción (en francés obviamente) sobre el puerto de Le Havre y como no la había hecho la inventó al vuelo, hizo como que la leía y le pusieron un diez igual, en su secundario en Resistencia, pero mi viejo se enfermó y se murió.

Entonces ahora que puedo, porque con la tarjeta la saco en muchas cuotas, y tengo un descuento, y el pasaje mío lo paga un cliente, y Tato paga la mitad, y alquilar un departamento allá sale lo mismo que en Mar del Plata (o lo mismo que una carpa en La Perla), y tengo un ahorrito, hace un tiempo le dije a mi vieja que sacara el pasaporte, que se venía conmigo.

Y que no, que estoy grande, que qué me vas a llevar, para qué, qué voy a hacer allá, pero el ojo bueno se le iluminaba, porque mi vieja por adentro es todavía esa alumna pupila del colegio en Santa Fe, que tenía que estudiar en silencio sus lecciones de francés, sobre esa tierra lejana, exótica de gente con boina.

Y la plata no compra la felicidad, pero esta plata, justo esta plata, me deja darle de regalo del día de la madre un viaje con el que sueña hace mi vida y media, ir a Montmartre, sentarnos en alguna parte, pedir dos cafés, morfarnos dos croissants, darle un beso a mi Tato, abrazar a mi mujer, a mi viejita, saludar a mi viejo, y sé que se me va a plantar un lagrimón, si se me está piantando ahora, mientras escribo, cómo no se me va a plantar allá.

Y me la llevo para allá. Y vamos a ser felices.

Ha habido una noticia que ha sacudido el ánimo de los argentinos en las últimas semanas, provocando hasta repercusiones globales. Estoy hablando, por supuesto, de la suba de precio de la yerba mate.

Sí, la yerba se fué al carajo. Y por supuesto, cuando un precio sube es culpa del gobierno. Y cuando no sube es malo porque el gobierno oprime a los productores. Y en ambos casos, es este gobierno prepotente, omnipotente, omnisciente y omnímodo el culpable de la suba o no suba.

Bueno, no creo. Habiendo conocido en mis épocas de trabajador de la UNL y/o estudiante a bastante gente que hoy en día tiene cargos políticos de importancia, les puedo decir que en su mayoría son una banda de pelotudos, que están en política por su incapacidad de mantener un laburo honesto y/o productivo. Y los otros son radicales, que es peor.

Sospecho que cuando subió la yerba, Moreno dijo “eh? en serio?”, y que se puso a ver como mierda hacía para que bajara antes que le hinchen las pelotas.

Sospecho que el aumento afectó a casi ninguno de los que se quejaron porque subió por 15 días y después bajó, y cuánta yerba consumís en 15 días? O sea, que te costó? \$30? \$50? Yo justo había comprado 2 kilos de yerba “La Vuelta” en Coto a \$4.50 el kilo, así que me sentí por unos días como si fuera Warren Buffett.

Se acuerdan cuando el morrón se había ido de \$5 a \$18 el kilo y estaba todo el mundo haciendo boycott al morrón rojo? Bueno, resulta que a veces los productos, por lo que puta fuere, se van a la mierda en precio y después bajan. O no bajan y no los consume nadie, y chau. Y sí, el gobierno es el que maneja (hasta ahí) la economía, y sí, a quién le vas a echar la culpa si no? Uy, no hay licuadoras! Yegua montonera, renuncie! Uy, no hay yerba pero no amo a mi mujer entonces me gustaría que hubiera yerba! Ojo, que yo sí la amo. A tu mujer no, a la mía. En fin, esas cosas.

No soy una persona que tenga grandes ilusiones políticas, en el sentido de que sospecho que ser gobierno no te da tanto poder como uno cree, no te hace competente, y si sos un siome con poder sos sobre todo un siome y que ni se levantan a la mañana los poderosos con la idea de cagarnos ni con la idea de salvarnos. Sospecho que se levantan a la mañana con ganas de seguir siendo poderosos, nomás, y que si cagarnos los ayuda en esa misión, alegremente nos cagarán, y si no no.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “tao-pai-pai”.

Pero veo mucha gente que cuando le preguntás como le va te dice que nunca estuvo mejor, pero cuando le preguntás que opina del gobierno dice que son peores que una combinación de Menem, Videla, y Tao-Pai-Pai.. tao-pai-pai: http://es.dragonball.wikia.com/wiki/Tao_Pai_Pai_vs_Tens

Esa gente no se da cuenta de la tensión entre su idea de un gobierno omnipotente que está intentando continuamente cagarlos, y es responsable de la variación del precio de los palmitos, unos ineficientes, corruptos, una lacra que los persigue pero ellos logran superar gracias a, seguramente, su inefable capacidad, su incorruptibilidad, su agilidad para escaparse.

Como puede ese gobierno omnipotente y omnisculpable, omniobstructor de sueños de clase media ser derrotado por ellos? Porqué les está yendo bien si están siendo perseguidos por esas bestias feroces sedientas de sangre de ciudadano de Nuñez/Belgrano/Palermo? Es explicable únicamente por la inmensa capacidad del susodicho Nuñezeño/Belgranense/Palermitaño?

Me imagino que, o no es tan omnipotente, o no los persigue tanto, y no son tan capaces, y si el gobierno fuera otro que hiciera otras cosas capaz que les iría un poco mejor, o un poco peor, o más o menos igual. Sospecho que no es determinante el

gobierno en como vive uno. Sospecho que es mucho más determinante uno, pero que el clima ayuda. Sospecho que lo que mata es la humedad, a menos que uno esté nadando, que es el deporte más completo.

Sospecho que ser hincha de un equipo de fútbol hoy en día es preocuparse por el color de la remera de gente que no conozco y corre atrás de una pelota, pero sospecho que ser parte de un partido político es parecido, pero estoy seguro que defender a una empresa es veinte veces más tonto. Sospecho que si le ofrecieran X dólares a Repsol, nos tiraba YPF por el marote, y que si le ofrecían X/2 es un ultraje, por lo que putas son, lo que falta es discutir precio.

Sospecho que estamos hablando de plata nomás.

Sospecho que van a decir “este Roberto, siempre ranteando”, sospecho que es cierto.

Un abrazo.

Siguiendo con mi inútil cruzada para acosar pavos a través del sarcasmo, hoy le toca al ínclito Claudio María Domínguez: pseudo gurú.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “periodista y conductor”.

Para el que no lo conozca, el niño Claudio María (“periodista y conductor”.. “periodista y conductor”: http://es.wikipedia.org/wiki/Claudio_Mar%C3%ADa_Dom%C3%ADnguez) ganó Odol Pregunta, allá por los años 70, contestando sobre mitología griega. Escribió el libro “El milagro de los sanadores filipinos” (lamentablemente no subtítulo “cómo hacerse rico con hígados de pollo”), y es un claro exponente de esa gente de fé indiscriminada.

El cree en todo. Cree en Jesús y en Buda, en la reencarnación y en el cielo, en los filipinos que te operan con los dedos y en el Dalai Lama, en Sai Baba y en los unicornios que cagan arcoiris.

Lo que a mí no me queda claro es qué pecado cometió en sus vidas anteriores este señor para tener esa voz. Porque esa voz es mal karma, esa voz es el castigo porque no quedaban mas escarabajos peloteros libres para reencarnarte. Esa voz es un arma de destrucción masiva.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “etapas del pase de plano”.

En esta entrega voy a analizar su opus intitulado [Etapas del Pase de Plano](http://espiritualidaddiaria.infobae.com/etapas-del-pase-de-plano/).. etapas del pase de plano: <http://espiritualidaddiaria.infobae.com/etapas-del-pase-de-plano/>

Lamentablemente (sospecho que hoy voy a usar mucho esa palabra) no se trata de un análisis del “pase plano” de los trucos de cartas, usado por el mítico René Lavand.

La sabiduría del budismo tibetano explica todo sobre el momento del pase de plano

Ajá. Todo todo. Explicado. Por la sabiduría. Del budismo tibetano. Hablemos un momento del budismo tibetano. En particular, de los monjes budistas tibetanos, que eran los miembros de una teocracia medieval en el siglo 20. Que se puede decir de los

sabios budistas tibetanos, excepto que si eran tan sabios es extraño que su país fuera tan atrasado, inhóspito y deprimente.

Las tres etapas en el momento del pase de plano físico

¡No tení título negro!

La sabiduría milenaria del budismo tibetano, explica que en el pase de plano se comienza a ver con el tercer ojo, es decir que se percibe la totalidad en 360 grados. Cesa la visión física, la motricidad corporal y empieza el vuelo del alma que se va desprendiendo y elevando.

Me encanta esa figura retórica de “X explica que cuando Y, pasa Z”. Ejemplos: “Yo te explico que si me das \$500, sos mejor tipo”. “El señor explica que si jodés con la barra de platense, los hinchas del calamar te bajan el comedor”.

Ambos paralelos son mucho más comprobables que lo que menciona el niño Carlos María. Pero bueno, ponéle que la sabiduría milenaria del budismo tibetano dice eso. ¿Sabés qué cultura tiene una sabiduría milenaria? La tuya. Si sos católico, tenés un libro sagrado con partes muy anteriores a la colonización del Tibet.

Por otro lado, sabés desde cuando gobernaban en Tibet los Dalai Lama? Desde el siglo XVI. Son mas nuevos que la monarquía española (que tremendos chorros son también, como todo monarca). Esa “milenaria sabiduría del budismo tibetano” data del siglo X. O sea, es milenaria *apenitas*. El budismo en general data del siglo V a.c. o sea 2500 años, o sea menos de la tercera parte que el judaísmo. Así que sí, es milenaria, pero eso no es ni siquiera raro, y no es particularmente antigua.

Por otro lado: ¿por qué corno va a ser mejor que sea milenaria? ¿Acaso deberíamos guardar nuestro bazo en un jarro porque la milenaria sabiduría egipcia dice que sin eso te vas a quedar sin bazo en el mas allá?

La partida depende de cómo se ha vivido y acá viene la primera disquisición. La manera en la que uno ha vivido, es como uno parte del cuerpo.

Ponéle. ¿Es como un tercer pezón? Ah, no es “como uno parte del cuerpo”, no “como una parte del cuerpo”. A menos que uno sea una señora, supongo. Ahí sí es “como una parte del cuerpo”. Pero igual es “la manera en que una (o uno) parte del cuerpo”, no que es como si fuera un pedazo del mismo. Del mismo cuerpo. No, no tengo el mismo cuerpo que esa señora. Dejá nomás.

¿Cómo sería eso? si tuviste una vida noble, con percepciones espirituales, si te pudiste desidentificar del cuerpo creyendo que eras mucho más, la partida empieza a ser ilimitada.

Evidentemente el niño Dominguez se cree mucho más, así que esto le viene bárbaro. Cuando dice “la partida” se refiere a algo como “la partida de ajedrez” o al hecho de irse? Si es lo segundo, ¿qué quiere decir “empieza a ser ilimitada”? Nada. Será lo primero, entonces..

Si viviste una vida muy prisionera de inseguridades, de apegos a los objetos y a las personas y mantuviste tu mente en el “me pica, me duele, me llama, no me llama, en el pobre de mi y la víctima recurrente” la partida se torna muy confusa.

Leo ese entrecomillado y me duele el ojo derecho. No creo que sea una coincidencia, creo que el problema es que empieza con un “en el” afuera de las comillas y después mete otro adentro, con lo que queda sumamente feo. Una partida confusa, será algo así como un Ferro/Platense. Cero a cero, nos recuerda Baglietto (aunque en realidad rara vez Ferro y Platense empataron cero a cero, búsqüenlo).

Si tuvieramos que resumir enseñanzas completas del libro Tibetano de la vida y de la muerte, podríamos referirnos a tres etapas claves para entender la partida del cuerpo y poder manejar esos conceptos durante años antes de tener que transitar ese momento.

Bueno, si tuviera que resumir la guerra y la paz, de Tolstoi, podría hablar de dos etapas claves para entenderla: antes y después de Austerlitz. ¿Te queda clarísimo, no? Así podés manejar esos conceptos durante años antes de tener que transitar el momento de leerlo.

Primera gran etapa:

A ver...

Lo primero que sucede es la desidentificación del cuerpo, el alma ve su cuerpo inerte.

Ajá.

Si tuviste mucho apego a ese cuerpo, el alma quiere volver pero el cordón de plata (que une el alma al cuerpo) una vez cortado energéticamente, hace que el alma se desprenda por completo de lo físico.

¿Como se corta energéticamente un cordón de plata? ¿hay un laser? ¡Decíme que es con un laser!

El alma en ese instante sabe que es mucho más que un cuerpo.

Bueno, sí, supongo que si estuviera flotando sobre mi cuerpo sabría eso. O estoy muy drogado, en un telo, y los espejos del techo están buenísimoooooos.

La mayoría de aquellos que tuvieron una muerte clínica, y retornaron, cuentan que no querían regresar en lo más mínimo y la sensación de paz era indescriptible, que mientras veían todo desde una cierta altura, no podían creer la belleza y la luz que percibían de ellos mismos al estar fuera del cuerpo.

Nótese que habla de muerte clínica, no de muerte biológica. La muerte biológica es irreversible, con lo cual nadie te puede contar que se siente. La muerte biológica, supongo, es cuando se corta el cordón ese, y no se puede volver a unir. ¿Pero si cuando se corta no se puede volver, qué testimonio hay de que se corte? ¿Como sabemos lo que pasa? Bueno, no sabemos, y Dominguez nos lo va a explicar.

Pero igual, no, ni ahí. De hecho la mayoría de los que tuvieron una muerte clínica y retornaron dicen que no vieron una miércoles, porque estaban desmayados. De los que dicen que vieron algo (una “Near Death Experience” como le dicen los nortños), que son alrededor del 18% (según el estudio de Van Lommel) y no una mayoría como dice este pejerito.

Veamos los datos de Van Lommel: de 344 pacientes que sufrieron muerte clínica por ataque cardíaco, 62 reportaron ver algo. De esos 62, 31 dijeron que sabían que

estaba muertos, 15 afirmaron tener una experiencia extracorpórea, 16 recuerdan un túnel, y el 56% afirmo experimentar sensaciones agradables, mientras que ninguno reportó molestia o miedo. Supongo que fue algo tipo “¿eh? ¿me morí? ¡Y bué!”.

Así que como suele suceder, apenas se menciona una cosa que es medianamente confirmable, resulta que es un invento, o simplemente ignorancia. Esperable.

Relatan incluso que mientras los médicos intentaban reanimarlos, ellos exclamaban con intensidad que no lo hicieran, para permanecer en esa situación idílica. Cuando se les pregunta, por qué volvieron a este plano más denso, suelen coincidir los testimonios, en que seres de una energía espiritual bellísima, les hicieron notar las materias que quedaban por resolver, y que podían elegir no volver al cuerpo en esta ocasión pero no podían evitar una nueva encarnación, y que sería mucho mas valiente retornar por unos años a ese cuerpo que los había acompañado en esta vida, cesar con karmas acumulados, y ser un testimonio para tantos de la percepción que se tiene cuando uno se aleja de la materia densa. Ser ejemplos coherentes de la vivencia fuera del cuerpo, como una experiencia mucho más sublime que cuanto momento gozoso uno pudiera haber experimentado estando encarnado.

De nuevo, solo el 56% afirmó experimentar emociones positivas. Ninguno de esos 344 afirmó ver esos seres que menciona Dominguez, mucho menos tener un diálogo, mucho menos que hubiera un patrón común en descripciones de esos seres y/o esos diálogos.

Dado que las historias de esos 344 no son coherentes, eso de “Ser ejemplos coherentes de la vivencia blah blah” es una pavada.

El cielo, es saber que no tenemos nada que ver con la parte química y ósea.
Saber que somos un espíritu sutil conectado al todo.

Ajá. Si no tenemos nada que ver con la parte “química y ósea” (que admito es una poética aunque confusa paparruchada), que por favor el señor Dominguez se abstenga del consumo de carbono y calcio.

Las almas más densas vuelven a los ambientes que han conocido y fagocitan energías de igual densidad. Los tibetanos lo llaman el bardo astral.

¡Que barderos estos tibetanos! Estimado autor, creo que la palabra fagocitar no significa lo que usted cree que significa. En particular no significa “consumir” o “alimentarse de”

Aclaro, ya que el señor no lo aclara, que “bardo astral” no es romper butacas en un teatro de calle Corrientes. De nada.

No hay que temer, porque nunca el desencarnado puede entrar en el ámbito del encarnado, por eso cuando alguien desencarna, envíalo a luz rápidamente, decíle que es luz y que avance a planos más elevados.

¿Pero, pero, cómo le digo? ¿Tengo que andar todo el tiempo diciendo “andá a la luz” por las dudas si hay alguno perdido por acá? ¿No sería eso cruel con las polillas? ¿No me va a agarrar a bastonazos un ciego por insensible? Si le digo a mi prima política “Sos luz”, me va a decir, “no, soy Sol, Luz es la morocha!” (true story).

Resumiendo: todo lo comprobable es falso. Todo lo no comprobable es verso. Sigamos.

Segunda gran etapa:

A ver...

Luego de desidentificarte con el cuerpo, la segunda gran etapa es desidentificarte de tu mente. Todo es mente. Hay una mente baja, una mente superior y una mente trascendente que es la del ser, la consciencia.

Pero si todo es mente, y me desidentifico de la mente, me desidentifico de todo, ¿no?

La mente baja sería leer esto en estado de adormecimiento,

Debo confesar que estoy empezando a adormecerme.

la mente superior es ser consciente que estás leyendo esto y saberte conectado con todo lo que te rodea.

¡Ok!

Una mente trascendente sería saber que excedemos los límites del cuerpo y que toda la creación extraordinaria con la que me estoy conectando no existiría si no existiese una consciencia que no tiene nombre ni forma que permea y genera todo.

Yo le digo Cacho. Cacho, el que permea y genera todo.

Desidentificarte de la mente significa que las creencias de tu mente van a aparecer.

Ajá. ¿Porqué significa eso? Porque así, leyendoló, medio que significa otra cosa.

Cuando alguien dice: “Vino Jesús, Sai Baba, la abuela” esa es una creación de tu mente que vibra en la sintonía afín de los seres espirituales a los que has amado en vida.

O sea, yo amo a un ser espiritual y mi mente va a crear algo, porque mi mente vibra en sintonía con ese ser espiritual que amo. ¿O la que vibra en esa sintonía es la creación? (ese Sai Baba, ponéle). Exijo una aclaración.

Es obvio que una persona que adoró a Jesús, vea a Jesús.

Bueno, no, obvio, lo que se dice obvio, no es.

Por eso se dice que esas energías de un altísimo nivel vibratorio, son las que aparecen para contener el pase de plano, el pase de etapas. Los guías y maestros, que amamos en vida, están en ese instante, para acompañarnos.

Se dice que “se dice” es lo que uno dice cuando quiere que algo parezca tener una autoridad imaginada por el lector en vez de poner pelotas y decir *quién* dice. Sospecho que lo dice Claudio María Domínguez, doble ganador de Odol Pregunta.

Como cuando una voz suave nos va despertando del sueño nocturno, de igual manera los guías nos ayudan en la transición.

A mí me despierta Celine Dion. Porque *odio* a Celine Dion.

Los maestros espirituales nos dicen que siempre volvemos a estar con nuestros seres queridos. Al estar desencarnados estamos conectados vibratoriamente entre todos.

Interesante concepto. Claro que asume una exacta simetría del querer. ¿Y si yo quiero a alguien y esa persona secretamente me odia? ¿Va a estar conmigo, pero yo no con él? ¿Y porqué voy a pasar la post-vida con una mentira?

La experiencia de partida del cuerpo en esta segunda instancia, es tan extraordinaria o tan temible como lo creas.

Dado que carecemos por completo aún de los testimonios que falsifica el autor acerca de la primera etapa, estamos ante nuestra vieja amiga, la afirmación contundente. Va a ser así. Porque va a ser así, te digo. Ah, y si es lindo, sí, y si es feo, sí. Porque lo que digo es tan fífi que no importa lo que te pase sigue siendo cierto. Porque estoy tipeando cosas que no significan una mierda.

Esta es la etapa de la experiencia inexorable de lo que creés que va a suceder. Quien cree que merece el cielo, vive situaciones llenas de luz, quien cree que merece el infierno, vive imagenes aterradoras como creación de su mente.

O sea, los psicópatas que carecen de conciencia de sus maldades: al cielo. Los honestos, conscientes de sus fallas y flaquezas: al infierno. Sos un moco como agente de viajes, Dominguez. Por lo menos otras teorías acerca de la post-vida tratan de dar justicia, vos das un tenedor libre.

Pero aún los cielos e infiernos deben ser eventualmente abandonados cuando la mente logra trascender aquello que creyó que sucedería y se funde finalmente en la verdad última y suprema.

Porque tenemos montones de testimonios de gente que fué al infierno y después salió, y se fundió en la verdad última y suprema, que evita, dicho sea de paso, explica abajo, la reencarnación. Ah no. de hecho, lo que está describiendo es una perfecta barrera para bloquear la información acerca de ese evento. Si eso pasa, es *claramente imposible que lo sepamos*. Con lo que no solo es improbable que sea cierto, si no que aún si lo fuera, estarías mintiendo, Dominguez.

¿Qué pasa con los ateos? También van directo a su sistema de creencias, si no han creído en nada, encuentran la nada más absoluta, y después de un tiempo de bucear en la nada, su mente aspira a algo más y pide que exista algo, y en ese instante logran avanzar a una etapa superior. Esa experiencia dura hasta que en un momento de desesperación después de estancarse en la nada, dice: ¿no puede haber algo más? Y ante ese cuestionamiento, está apto para el pase de plano. Alguien que está en un tormento mental, lo que llaman infierno, en un punto pide que alguien lo rescate y va a estar apto para el pase de plano, pero nadie deja de experimentar el deseo de su creencia más fuerte.

No funcionaría. Si los ateos creen en la nada (que no, que no es así, que creemos que no hay nada, que no es lo mismo), al estar ese ateo ahí, en la nada, tendría un momento en que diría, pará un minuto, esto no es la nada, porque estoy yo. Cogito ergo sum, que le dicen. Te puede llevar tipo 30 segundos.

Resumen: un delirio inconsistente e inmoral.

Tercera gran etapa: La más sublime, ya que es el estado de éxtasis del alma, cuando ya se desecharon las creencias limitantes de la mente y se abandonan las imágenes recientes para avanzar.

¡Suena groso!

En esta etapa, el alma se funde en su verdadera esencia.

Ajá. Las cosas inmateriales no se funden.

Sai baba me explicaba que en ese instante la luz no es como el sol sino como millones de soles y que no es una luz cegadora, es una luz que envuelve al alma y la lleva a la comprensión de que esa luz es lo que ella es, lo que siempre fue.

¿Y como sabía eso Sai Baba?

Se permanece en ese estado y se sigue la expansión ilimitada de la conciencia si ya no hay karma y no hay que seguir aprobando materias. El alma observa con ecuanimidad en esta etapa lo que hizo en vida mientras estaba en el cuerpo. Ve si le quedan materias sin aprobar, como ser la materia del amor incondicional, el perdón, la alegría de vivir, trascender la carencia, aprender la solidaridad y a salirse de los apegos que lastiman, avanzar en la comprensión de la propia divinidad, etc.

¿Me das el listado de materias? ¿Hay correlativas? ¿Se puede rendir libre?

El alma comprende claramente lo que le causó a otras partes de sí misma. Se da cuenta que siempre fue ella misma pero en otras formas, por eso todo lo que hizo en ese momento lo experimenta, porque siempre fue ella.

¿Lo qué?

Y por ley de causa y efecto, lo revive. Si quedaron materias, desanda el túnel y va creando las circunstancias afines a su nueva encarnación, va eligiendo el cuerpo en el que va nacer, el sexo, los padres, las situaciones a vivir con las personas adecuadas que le van a espejar aquello que tiene que aprender y decide cómo va a ser el primer septenio y el entorno.

Esta es buenísima. Para los que no lo saben, está diciendo que el alma, al reencarnarse, elige en qué condiciones va a vivir su primeros siete años. Esa es tal vez la enseñanza más perversa de todas las mencionadas hoy.

¿Que te moriste de hambre a los tres meses? Vos lo elegiste antes de nacer. No es culpa del Dalai Lama, que es apenas el señor feudal de la zona. ¿Que sos pobre? Culpa tuya. ¿Que estás desnutrido? Culpa tuya. Nadie tiene ningún derecho inalienable, nadie tiene ninguna obligación. Todo castigo será asignado por el alma en la próxima reencarnación, todo sufrimiento es merecido porque es ese castigo. Y sos castigado por algo que no sabés qué es, que nadie puede saberlo, porque fué en otra vida. Pero jodéte, comé gusanos y calláte.

Es repugnante. Todo el edificio teórico y moral de la rueda kármica es asqueroso, primitivo, tendiente a perpetuar el status quo de los privilegiados, a mantener señores pelados viviendo de limosnas y sintiéndose bien consigo mismos.

No, Claudio María Domínguez, lo que describís no es paz, no es luz, es horrible. Lo que describís es un infierno que ni siquiera tiene una aspiración de justicia.

Todos aquellos que “compran” esta idea de “espiritualidad” deberían pensar, un poco, en qué están metiéndose. Y vos, Dominguez, seguirás currando, porque siempre hay crédulos, siempre hay gente desesperada dispuesta a darte diez pesos por explicar que hay un filipino mágico que te saca el cancer con los dedos, porque esa es la clase de basura que sos.

Poné cara de bueno, hablá con esa voz de bobo superado. Sos una lacra. Y vos lo sabés.

Sorry, spanish only!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “1”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “1”.

Hace mucho tiempo hice unos artículos que llamé Discutiendo con Cualquiera (1..
1: <http://lateral.netmanagers.com.ar/weblog/posts/BB898.html>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “2”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “2”.

, 2.. 2: <http://lateral.netmanagers.com.ar/weblog/posts/BB899.html>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “3”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “3”.

, 3.. 3: <http://lateral.netmanagers.com.ar/weblog/posts/BB903.html>

) donde me tomaba mucha más molestia de la necesaria para responderle a un cambio.

Bueno, vamos a probar de nuevo. Me linkearon en facebook a esta hermosa página llamada “¿Evolución o Creación?” en <http://www.tubreveespacio.com/evolucion.html> escrita por Edgar Martínez, y voy a tratar de responder como corresponde. Con sarcasmo. Porque el sarcasmo es la forma más baja del ingenio, como dijo alguien, pero por lo menos *es* ingenio. No como otras cosas.

Ya arranca ambicioso:

Recientes investigaciones derrumban por completo la teoría de la evolución y le dan nueva credibilidad a la creación de la vida por una inteligencia superior ¿Para Ripley's? Adéntrate en este fascinante artículo

¿Evolución o Creación?

Años atrás la teoría de la creación de todo ser viviente por Dios no se ponía a discusión, pero surge Charles Darwin y, con su teoría de la evolución, parecía que explicaba por completo la existencia de la vida en la tierra.

Este... no? Para empezar, la teoría de la evolución no explica la existencia de la vida en la tierra. Eso, llamado el problema de la abiogénesis, está fuera de su rango. Tampoco explica las mareas, ni por qué brilla el sol, ni cómo puede ser que nadie confiese que le gustan los caramelos media hora. Explica otras cosas.

Por otro lado, ha habido desde la antigüedad gente que no dice que a los seres vivientes los creó dios. Por ejemplo, los hindúes. Los babilonios. Los romanos. Los griegos. Los aztecas. Los mayas. El 100% de la población hace 7000 años. Etc.

Pero Charles no contaba con los avances científicos de nuestros días.

Lo raro sería que hubiera contado con ellos. Imagino a Darwin, en Argentina, a punto de ser picado por una vinchuca, pensando, “¡La pucha, si tan solo contara con un insecticida ambiental eficiente!”

La fantástica historia que se relata en el capítulo bíblico de Génesis, de la creación del mundo en 6 días y en la que Dios creó a la mujer de la costilla del primer hombre, Adán, sonaban mas bien a leyenda e invención del hombre, parecía uno mas de los innumerables mitos de la creación del mundo que han existido en diversas culturas antiguas.

100% de acuerdo. Lo sigue pareciendo.

Y entonces surge la teoría de la evolución, de Charles Darwin.

Es importante entender el momento histórico en el que se acomoda fértilmente la teoría de la evolución de Charles Darwin.

En aquella época, la iglesia católica ejercía un poder tiránico.

Darwin no vivía en un país católico. Vivía en Inglaterra, hogar de la iglesia anglicana. Que son protestantes. Que no reconocen la autoridad del papa. Que llamaban a los católicos cosas como “mariólatras”.

La educación, las únicas universidades, pertenecían a la Iglesia. Imponía sus puntos de vista, sin abrirse a las expresiones de otras ideas. Y al no ocuparse en las cosas de Dios, sino en cuestiones de riquezas y poder, habían hecho perder la fe de las masas.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “test acts”.

Darwin estudió en Cambridge. Cambridge tuvo su primer capellán católico en 1896, cuando Darwin tenía ya 14 años de muerto. De hecho, Cambridge permitió el ingreso de católicos recién en 1871 (cuando Darwin tenía ya 62 años) cuando se derogaron los [Test Acts](http://en.wikipedia.org/wiki/Test_Act). test acts: http://en.wikipedia.org/wiki/Test_Act

. De hecho, la congregación para la propagación de la fé de la iglesia católica, había proclamado que asistir a Cambridge era probable causa de pecado mortal dada la atmósfera de liberalismo y escepticismo reinantes.

Es evidente que el autor no tiene la más puta idea acerca de la historia de las universidades en Inglaterra, o la vida de Darwin

Pues bien, llega Charles Darwin con su teoría... ¡la abrazaron! Explicaba que los animales se habían originado a sí mismos, y que las especies de animales se habían originado a partir de unas pocas, por medio del proceso de la selección natural.

Ponéle.

Si un animal necesitaba adquirir la habilidad de imitar el color de su habitat para sobrevivir y lo lograba, eso era una característica que le daba ventaja sobre los demás, y era una característica que de forma natural se heredaba a su progenie.

Ponéle.

La sencilla lógica de la teoría había dado en el clavo, las masas habían encontrado en esta teoría la forma de expresar su descontento hacia la iglesia católica.

O... de explicar el mundo que las rodeaba. Que es mucho más interesante.

La iglesia católica había olvidado por completo su esencia: conocer la Biblia.

Ponéle.

Demonstraron conocer nada de lo que hablaban, y en vez de defender con argumentos la creación del mundo por Dios, los líderes de la iglesia empezaron ataques personales a Charles Darwin, diciéndole a Darwin que él era descendiente de un mono.

No, el que le dijo eso no era un líder de la iglesia católica. Era el obispo Wilberforce, de la iglesia anglicana. Y no se lo dijo a Darwin, se lo dijo a Huxley. Específicamente, le preguntó si “descendía de un mono de parte de padre o de madre”, que hay que admitir, es por lo menos gracioso.

El pueblo se dio cuenta que la iglesia católica no defendía coherentemente su punto de vista.

La belleza y sencillez de la teoría darwiniana consiguió el apoyo de la comunidad científica y el pueblo. La creación por Dios era dogma de fe, la darwiniana, lógica.

Ponéle.

Y a su vez, la comunidad científica se cegó. Empezaron a ver todo bajo el prisma evolutivo. Porque eso es lo que deseaban ver.

Ni ahí. Pero bué... ponéle.

El pueblo, cegado antes por la teoría religiosa, ahora creía dogmática y ciegamente en la darwiniana. No razonaban. Eterno hábito de no pensar y ser seguidor. Los pocos hombres que piensan, empezaron a cuestionar e investigar la teoría darwiniana.

¿“El pueblo”, “ahora”? ¿Ahora cuándo? Y desde ya tenemos el doble discurso. Antes aceptaban el creacionismo porque nadie lo discutía, ahora es que no razonan. Y el típico gambito elitista de “los pocos hombres que piensan”. Nótese la horrenda concordancia de tiempos verbales.

Te vas a sorprender de lo que todos aceptan como verdad, que te han enseñado tus profesores de la escuela y que lees a cada rato en diversos artículos. Y que no son ciertos. Agárrate de la silla.

Me agarro.

Una coincidencia sorprendente entre la teoría de Darwin y la Biblia de la creación de la vida, es que el orden de etapas que describe la Biblia que fue creada la vida (origen de los océanos, creación de la tierra, aparecimiento de la vida marina, surgimiento de aves y mamíferos) es la misma secuencia de las divisiones principales del tiempo geológico, como las describe la ciencia.

Las pelotas. Las pelotas deben haber sido creadas junto con los mamíferos supongo. pero igual: concordancia las pelotas.

Esto lo afirma el geólogo Wallace Pratt. Aunque usted... no lo crea.

Ajá. Vamos a hacer algo que nunca hace este tipo de gente, y citar a Wallace Pratt:

“If I as a geologist were called upon to explain briefly our modern ideas of the origin of the earth and the development of life on it to a simple, pastoral people, such as the tribes to whom the Book of Genesis was addressed, I could hardly do better than follow rather closely much of the language of the first chapter of Genesis.”

—Wallace Pratt

Nótese que dice “si le tuviera que explicar a simples pastores”. No somos simples pastores. Somos gente moderna, que puede llegar de Egipto a Israel en 40 minutos, no 40 años.

Y es la única historia antigua del surgimiento del mundo que cuadra con la científica.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “el mito de creación de los indios pawnee”.

Las pelotas, como venía diciendo, son aproximadamente creadas en el tercer día en el libro del Génesis. Sin embargo, lo que dice este señor es simplemente erróneo. Esa explicación no cuadra con la explicación científica, o por lo menos no mejor que, por ejemplo, [el mito de creación de los indios pawnee](http://www.bluewolfspirit.com/blog/2004/12/the_pawnee_crea.html). el mito de creación de los indios pawnee: http://www.bluewolfspirit.com/blog/2004/12/the_pawnee_crea.html

que comienza la creación

system-message

WARNING/2 in `everything.txt`, line 33856

Block quote ends without a blank line; unexpected unindent. backrefs:

con un acto de parlamentarismo.

Darwin también decía que los individuos más capaces de cada especie son los que sobrevivían, y que las características que hacían que éstos individuos sobrevivieran se transmitían de generación a generación.

Y de mejora en mejora... ¡se producían nuevas especies!

Una teoría hermosamente lógica y sencilla.

Pero en los últimos 100 años la ciencia descubrió el código genético (ADN) cuyas funciones anulan por completo esa teoría.

Este.... no?

La función del código genético es perpetuar las características fisiológicas del animal, permitiendo una variación en cuanto al tamaño y color, pero no permite cambios tan grandes que se creen nuevas especies.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "blaquito".

Ajá. Veamos. Yo tengo un código genético. Mi gato [blaquito](http://picplz.com/user/ralsina/pic/0b2fm/).. blaquito: <http://picplz.com/user/ralsina/pic/0b2fm/> tiene un código genético. Pero el código genético no

system-message

WARNING/2 in `everything.txt`, line 33873

Block quote ends without a blank line; unexpected unindent. backrefs:

permite cambios tan grandes como aquellos entre especies. Ajá.

Pondré un ejemplo práctico de cómo funciona el código genético.

Es como si un león tuviera un programa de computación (código genético) llamado león, y que hace que un león tome las características que le conocemos en cuanto a tamaño, forma, instintos, etc.

No. El ADN no es como "un programa de computación". No es ni remotamente parecido.

Y es un programa protegido contra borrado y escritura ¿Este programa (código genético) puede sufrir errores tales que permita la creación de nuevas especies? No.

Ah, nuestra vieja amiga, la prueba por afirmación contundente. ¿Porqué no puede? Porque **NO**.

Como un programa de computación que le entra algún virus o que accidentalmente el programador borra un archivo, y eso hace que el programa funcione de manera diferente, el código genético también puede recibir daños.

Se le conocen como mutaciones.

Soy un programador. No tenés idea.

Los científicos intentaron modificar este programa de computadoras de la naturaleza (código genético) para probar que sí se podían generar nuevas especies.

Hicieron un experimento con moscas del vinagre, induciendo mutaciones del código genético con rayos x ¿Resultados? Surgieron moscas en su gran mayoría con características inferiores a las promedio.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "leerlo acá".

¿Sabés qué? También hicieron otras cosas. Como por ejemplo, criar bacterias que pueden nutrirse de cosas que antes no podían. Podés [leerlo acá](http://myxo.css.msu.edu/lenski/pdf/2008,%20PNAS,%20Blount%20et%20al.pdf). leerlo acá: <http://myxo.css.msu.edu/lenski/pdf/2008,%20PNAS,%20Blount%20et%20al.pdf>

pero claro, es un poco más complicado que tu versión.

Cuando se cruzaron las moscas mutantes, sorpresivamente después de algunas generaciones, surgieron moscas normales.

Las pelotas! Ah, no, cierto, las moscas no tienen pelotas.

El genetista Richard Goldsmith se desesperó, porque los cambios originados en las moscas, aún si se juntaran todos, eran tan pequeños que no crearían una nueva especie.

No Goldsmith. Goldschmidt. Autor de obras tales como "The genetic background of chemically induced phenocopies in *Drosophila*". Así que tiráme una referencia. Dáale, una solita.

La revista investigación y ciencia informa que los daños al código genético inducen una respuesta de emergencia del código genético, y éste genera automáticamente enzimas que reparan el daño.

¿Adónde informa eso?

Es como si un programa de computadoras le fueran borradas algunas líneas de programa, que harían que no funcionara bien.

Pero como si ese programa automáticamente lo detectara y pusiera las líneas borradas por sí mismo, y en unos días ya funcionara como su programador originalmente hizo el programa ¡Fantástico! ¿No? pues aunque los programas de computadoras actualmente no hacen eso, el código genético sí lo hace.

Ajá. Bueno, pero no. En ambos casos. Hay programas con redundancia donde un daño al mismo no impide su funcionamiento. Es posible dañar el código genético más allá de lo reparable.

El código genético, al ser una especie de programa de computadoras maravillosamente creado, a prueba de modificaciones y autoreparable, es el principal obstáculo para la teoría de la evolución.

Las pelotas, si bien fueron creadas recién en el tercer día (o tal vez el cuarto?) son una maravillosa fuente de material genético (literalmente).

Los científicos ahora dicen que debió haber existido un código genético primitivo que permitía cambios, pero que después evolucionó y ya no los permitió. A la fecha, no han podido comprobarlo.

Estos famosos “los científicos” y sus dichos.

¡Pero si hay en los museos los descendientes del hombre! ¡Pero si la ciencia dice que descendemos del mono y está científicamente comprobado! ¡Pero si existieron los hombres de Java, Neanderthal, Tepexpan y Cro-Magnon!

¡Claro!

Te explico. Respecto a que descendemos del mono, se hizo un experimento con “Ko Ko”, una gorila hembra, para enseñarle a hablar y demostrar que el hombre pudo haber evolucionado a partir del mono.

¿Eh? Sí, se le enseñó lenguaje de señas a una gorila. ¿Qué tiene que ver eso con la evolución del hombre y con el sabor del kinoto?

El experimento explica que la gorila fue capaz de expresar sentimientos y emociones a través de un sistema de comunicación parecido al que usan los sordomudos.

Pero en el libro *¿Cómo nos comunicamos?* de editorial Alhambra, las personas del experimento concluyen que Ko Ko jamás podría hablar con el lenguaje de los humanos, puesto que el aparato de fonación natural de los monos no permite la articulación de palabras tal como los humanos hacemos.

Por eso le enseñaron *lenguaje de señas*. Porque es una *gorila*. Y los gorilas *no hablan*.

En pocas palabras, si un mono no pudo desarrollar el lenguaje humano con todos los medios disponibles para estimularlo, por su aparato de fonación...
¡Menos lo iban a hacer los monos de la antigüedad en estado salvaje!
¿Cómo te quedó el ojo?

Bien gracias. Apliquemos el mismo razonamiento a un tema más sencillo. Esta sociedad no ha conseguido enseñar a razonar al autor de ese párrafo a pesar de contar con todos los medios disponibles para estimularlo. Ergo, el autor es un mono tí. ¿Como te quedó el ojo?

De los hombre de la antigüedad, te voy a hacer la siguiente observación:
Si el hombre hubiera evolucionado, en la misma zona geográfica estaría la prueba.

Ajá.

Estaría el hombre de Neanderthal en la misma zona geográfica, que el de Cro-Magnon y el de Java, puesto que un hombre era la “mejora” del otro, producto del mismo ambiente natural, según la teoría de la evolución.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “especiación alopátrica”.

No. Le presento al señor la teoría de [especiación alopátrica](http://en.wikipedia.org/wiki/Allopatric_speciation).. especiación alopátrica:
http://en.wikipedia.org/wiki/Allopatric_speciation

. Claro, es un poquito más complicada de lo que dice la biblia.

Pues te informo que el hombre de Neanderthal solo se ha encontrado...¡En Alemania! El de Cro-Magnon...¡Solo en Francia! El de Java...¡Solo en Indonesia!. El de Pekín...¡Solo en China! El de Tepexpan...¡Solo en México!

Le presento al señor las cuevas de Châtelperron, donde fueron hallados fósiles de Cromagnon y Neanderthal. Y esta linda lista de donde se han encontrado fósiles de Homo Neanderthalensis:

- Uzbekistán
- Ucrania
- Siria
- Rusia
- Rumania
- Portugal
- Israel
- Irak
- Irán
- Italia
- Inglaterra
- Francia
- España
- Eslovenia
- Croacia
- Chequia
- Bélgica

- Alemania

Ouch.

Te repito, si hubiera existido evolución, la prueba debería de estar en la misma zona geográfica.

Listo, entonces ya está, está probado y abandonás?

Observa este hecho la próxima vez que vayas a un museo de historia natural.

Estos hombres simplemente eran variaciones dentro de la misma especie humana.

Actualmente muchos científicos afirman que estos hombres de la antigüedad no difieren en nada de los de ahora.

En el caso del hombre de Cro-magnon, sí, es parecido el cambio.

¿Sabes que? Yo creo en Dios, lo veo como el ingeniero de computación y arquitecto perfecto.

Si un arquitecto construye casas en la playa, elegirá materiales ligeros. Si construye en ciudad, elegirá materiales diferentes por el clima. Pero utiliza las mismas estructuras.

¿Eh? Ponéle.

De la misma forma, Dios usa las mismas estructuras óseas, pero adapta el volumen y forma de acuerdo a las necesidades de su creación.

Las alas de las aves están hechas con huesos huecos, que son mas ligeros y facilitan la función de volar.

Las aletas de las ballenas, cubren una mayor área y facilita la función de nadar.

¿Has observado la increíble sabiduría de la naturaleza? Los camaleones adaptan su color instintivamente al color de su medio ambiente, hay abejas que son engañadas por flores con forma similar a la de dichos insectos, y todo esto para que las flores sean polinizadas...¡en fin!

En fin... ¿qué?

La asombrosa armonía y diseño de la naturaleza lo intentó explicar la teoría darwiniana de la selección natural.

Sí.

Pero desde el momento en que ya te expliqué que todos los animales y plantas tienen un código genético, preprogramado, y ese código es como un programa de computadoras que no se puede modificar, y ya vienen especificadas por default las características de cada animal y planta...

Claro, como ya me explicaste está todo claro... no.

¡Dios es también el mejor programador! ¿Qué te pareció el sensacional diseño del código genético, a prueba de modificaciones y autorenovable?

Me encanta tener material genético. ¡Menos mal que la iglesia católica descifró el funcionamiento de los genes leyendo la biblia!

Si el camaleón puede imitar el color del árbol...¡Es porque está en el código genético del camaleón! ¡Y es porque Dios lo hizo así!

¡Y si el Chori Domínguez tiene dos pies izquierdos es porque está en su código genético! ¡Porque dios lo hizo así! Por favor, tratemos de mantener un mínimo de seriedad. Cada vez que el autor dice algo que es posible comprobar, termina siendo una pelotudez. Afirmar estentóreamente pelotudeces sólo logra hacer ruidos pelotudos. Porque las pelotas las creó dios. Acuérdense.

¡En fin! Si te interesa el tema, pregunta a tus profesores, lee, investiga...
¡Y después me dices si estás de acuerdo conmigo o no!

No estoy de acuerdo. Todo lo que escribiste es una cadena de falsedades, tonterías y afirmaciones vacuas.

Hasta la próxima.

¡Nos vemos!

Colaboración de Edgar Martínez México

De nada.

I am doing a second theme for Nikola, and what better way to test it than migrating this site to it!

In the process, I stopped using custom templates here, organized how a theme is done, how translations work, and other stuff.

Comments about broken / untranslated / missing stuff are much appreciated!

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "nikola".

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "nikola".

I just switched this blog to a new [Nikola.. nikola: http://nikola.ralsina.com.ar](http://nikola.ralsina.com.ar)

theme, that

system-message

WARNING/2 in `everything.txt`, line 34080

Block quote ends without a blank line; unexpected unindent. backrefs:

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "doc".

I am calling Readable. Because it tries to be readable. It's loosely based on the looks of a WP theme called [doc](http://wordpress.org/extend/themes/doc).. doc: <http://wordpress.org/extend/themes/doc>, but it shares no code.

If you are seeing this through a planet, this is how it looks:

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "don't trust the bitch in apartment 23".

First, a bit of advice: If you have not seen [Don't trust the bitch in apartment 23](http://www.imdb.com/title/tt1819509/).. don't trust the bitch in apartment 23: <http://www.imdb.com/title/tt1819509/> go ahead and do it, it's very funny.

The hero of the story is a bright young woman (played by an actress that looks like Zooey Deschanel's smart, evil cousin) who has discovered a unsatisfied need in the real estate market: roommates who will take your deposit, make your life hell, convince you to leave, yet commit no crimes vile enough to warrant police's attention.

Having found that need, she, like any true human, fulfills it in heroic fashion, thus taking what's rightfully hers by virtue of her ingenuity and willpower.

Her latest roommate is attracted to her magnetic personality and strong élan, and displays some interesting wit of her own when she reacts to our hero's scheme by stealing and selling all her belongings, which leads to funny hijinks.

Since this is a utopic comedy, state's interference in the affairs of our heroes is inexistant, allowing them to exercise sovereign rights over their own bodies, like when they give alcohol to a thirteen year old, or deal in drugs, which as we all know, are victimless crimes, played for laughs (and really, the chinese pills bit *is* funny).

I have only seen the pilot, but I look forward to the protagonists' starting a steel company, or maybe a railroad corporation, or maybe joining John Galt.

I give this show 4 Rands out of five.

I had proposed a talk for PyCon Argentina called "Driving 3 Nails with a Shoe". I know, the title is silly, but the idea was showing how to do things using the wrong tool, intentionally. Why? Because:

- 1) It makes you think different
- 2) It's fun

The bad side is, of course, that this talk's contents have to be a secret, or else the fun is spoiled for everyone. Since the review process for PyConAr talks is public, there was no way to explain what this was about.

And since that means the reviewers basically have to take my word for this being a good thing to have at a conference, which is unfair, I deleted the proposal. The good (maybe) news is that now everyone will see what those ideas I had were about. And here is nail number 1: Writing a spreadsheet using `doit`.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “a famous recipe by raymond hettinger”.

This is not my first “spreadsheet”. It all started a long, long time ago with a famous recipe by Raymond Hettinger.. a famous recipe by raymond hettinger: <http://code.activestate.com/recipes/355045-spreadsheet/>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “and again”.

which I used again and again and again (I may even be missing some post there).

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola”.

Since I have been using [doit](#) for [Nikola](#) I am impressed by the power it gives you. In short, doit lets you create tasks, and those tasks can depend on other tasks, and operate on data, and provide results for other tasks, etc.

See where this is going?

So, here’s the code, with explanations:

`cells` is our spreadsheet. You can put anything there, just always use “cell-name=formula” format, and the formula must be valid Python, ok?

```
\ :kn: 'from' \ \ :nn: 'tokenize' \ \ :kn: 'import' \ \ :n: 'generate_tokens' \
\n: 'cells' \ \ :o: '=' \ \ :p: '[' \ \ :s: '"A1=A3+A2"' \ \ :p: ',' \ \ :s: '"A2'
\n: 'values' \ \ :o: '=' \ \ :p: '{}'
```

`task_calculate` creates a task for each cell, called `calculate:CELLNAME`. The “action” to be performed by that task is evaluating the formula. But in order to do that successfully, we need to know what *other* cells have to be evaluated first!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “calculated dependencies”.

This is implemented using doit’s [calculated dependencies](#).. calculated dependencies: <http://python-doit.sourceforge.net/dependencies.html#calculated-dependencies> by asking doit to run the task “get_dep:FORMULA” for this cell’s formula.

```
\ :k: 'def' \ \ :nf: 'evaluate' \ \ :p: '(' \ \ :n: 'name' \ \ :p: ',' \ \ :n: 'form
\n: 'value' \ \ :o: '=' \ \ :nb: 'eval' \ \ :p: '(' \ \ :n: 'formula' \ \ :p:
\n: 'values' \ \ :p: '[' \ \ :n: 'name' \ \ :p: ']' \ \ :o: '=' \ \ :n: 'value
\n: 'print' \ \ :s: '" ' \ \ :si: '%s' \ \ :s: ' = ' \ \ :si: '%s' \ \ :s: '" '\
\n: 'def' \ \ :nf: 'task_calculate' \ \ :p: '():'
```

```

\ :k: 'for' \ \ :n: 'cell' \ \ :ow: 'in' \ \ :n: 'cells' \ \ :p: ': ' \
  \ :n: 'name' \ \ :p: ', ' \ \ :n: 'formula' \ \ :o: '=' \ \ :n: 'cell' \ \ :
  \ :k: 'yield' \ \ :p: '{ ' \
    \ :s: 'name' \ \ :p: ': ' \ \ :n: 'name' \ \ :p: ', ' \
    \ :s: 'calc_dep' \ \ :p: ': ' \ \ :p: '[' \ \ :s: 'get_dep: ' \ \ :s:
    \ :s: 'actions' \ \ :p: ': ' \ \ :p: '[( ' \ \ :n: 'evaluate' \ \ :p:
  \ :p: '}' \

```

For example, in our test sheet, A1 depends on A3 and A2 but those depend on no other cells. To figure this out, I will use the tokenize module, and just remember what things are “names”. More sophisticated approaches exist.

The `task_get_dep` function is a `doit` task that will create a task called “`get_dep:CELLNAME`” for every cell name in `cells`.

What `get_dep` returns is a list of `doit` tasks. For our A1 cell, that would be [`calculate:A2`”, `calculate:A3`”] meaning that to calculate A1 you need to perform those tasks first.

```

\ :k: 'def' \ \ :nf: 'get_dep' \ \ :p: '(\ ' \ \ :n: 'formula' \ \ :p: ')': \
  \ :sd: '""' \ \ :o: '=' \ \ :p: '{ ' \
  \ :n: 'deps' \ \ :o: '=' \ \ :p: '{ ' \
  \ :k: 'try' \ \ :p: ': ' \
    \ :k: 'for' \ \ :n: 'token' \ \ :ow: 'in' \ \ :n: 'generate_tokens' \ \ :
      \ :k: 'if' \ \ :n: 'token' \ \ :p: '[' \ \ :mi: '0' \ \ :p: ']' \ \ :o:
        \ :n: 'deps' \ \ :p: '[' \ \ :n: 'token' \ \ :p: '[' \ \ :mi: '1' \ \ :
  \ :k: 'except' \ \ :ne: 'IndexError' \ \ :p: ': ' \
    \ :c: '# It's ok' \
    \ :k: 'pass' \
  \ :k: 'return' \ \ :p: '{ ' \
    \ :s: 'result_dep' \ \ :p: ': ' \ \ :p: '[' \ \ :s: 'calculate: ' \ \ :s:
  \ :p: '}' \

\ :k: 'def' \ \ :nf: 'task_get_dep' \ \ :p: '(): ' \
  \ :k: 'for' \ \ :n: 'cell' \ \ :ow: 'in' \ \ :n: 'cells' \ \ :p: ': ' \
    \ :n: 'name' \ \ :p: ', ' \ \ :n: 'formula' \ \ :o: '=' \ \ :n: 'cell' \ \ :
    \ :k: 'yield' \ \ :p: '{ ' \
      \ :s: 'name' \ \ :p: ': ' \ \ :n: 'formula' \ \ :p: ', ' \
      \ :s: 'actions' \ \ :p: ': ' \ \ :p: '[( ' \ \ :n: 'get_dep' \ \ :p: ',
    \ :p: '}' \

```

system-message
WARNING/2 in everything.txt, line 3
 Duplicate explicit target name: “here”.

system-message
WARNING/2 in everything.txt, line 3
 Duplicate explicit target name: “here”.

And that’s it. Let’s see it in action. You can get your own copy [here](http://lateral.netmanagers.com.ar/static/dodo.py).. here: <http://lateral.netmanagers.com.ar/static/dodo.py> and try it out by installing `doit`, editing `cells` and then running it like this:

```

ralsina@perdido:~/dosheet$ doit -v2 calculate:A3
. get_dep:4
{}
. calculate:A3
A3 = 4
ralsina@perdido:~/dosheet$ doit -v2 calculate:A2
. get_dep:2
{}
. calculate:A2
A2 = 2
ralsina@perdido:~/dosheet$ doit -v2 calculate:A1
. get_dep:A3+A2
{'A3': None, 'A2': None}
. get_dep:4
{}
. calculate:A3
A3 = 4
. get_dep:2
{}
. calculate:A2
A2 = 2
. calculate:A1
A1 = 6

```

As you can see, it always does the minimum amount of effort to calculate the desired result. If you are so inclined, there are some things that could be improved, and I am leaving as exercise for the reader, for example:

1)

```

system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "uptodate".

```

Use `uptodate.. uptodate: http://python-doit.sourceforge.net/dependencies.html#uptodate` to avoid recalculating

```

system-message
WARNING/2 in everything.txt, line 34277
Block quote ends without a blank line; unexpected unindent. backrefs:

```

dependencies.

2)

```

system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "computed values".

```

Get rid of the global values and use `doit's computed values.. computed values: http://python-doit.sourceforge.net/dependencies.html#saving-computed-values`

instead.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "the full listing".

Here is [the full listing](http://lateral.netmanagers.com.ar/listings/dodo.py.html).. the full listing: <http://lateral.netmanagers.com.ar/listings/dodo.py.html>, enjoy!

Since I have a small kid, I know stuff other people don't. Specifically, I know that a surveillance state is forthcoming and that noone will care. Noone that matters that is. Because they all will have learned about it when they are three. So, by 2035, the expectation of privacy will be: None.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "special agent oso".

There is a classic south american leftist book called "To Read Donald Duck" that explains, from a marxist-theorist point of view, how Donal Duck forges the public conscience. If Dorfman saw one episode of [Special Agent Oso](http://disney.go.com/disneyjunior/special-agent-oso).. special agent oso: <http://disney.go.com/disneyjunior/special-agent-oso>

he would have an aneurysm. Let me summarise every episode:

- 1) Adorable Kid in any country. The show specifically shows what country it is, and the characters have localized features, names and dresses.
- 2) Adorable Kid has a problem.
- 3) His predicament is filmed by a robot ladybug, which reports via a satellite to an unnamed organization.
- 4) The mysterious boss called "Mr. Dos" (By the way, the chief of the argentinian secret service is called "Señor Cinco". Just saying!) assigns the mission to one of his "agents", usually our main character, Oso.
- 5) Oso uses advanced technology to find Adorable Kid, and with the guidance of his "Paw Pilot" (how dated is that?), teaches Kid the three easy steps to solve his predicament.
- 6) Oso learns a valuable lesson for his apparently endless Special Agent training program.

This is wrong at so many levels it's hard to keep track of them, but let's try anyway.

- There is an unnamed organization that has the resources to know when every kid can't tie his shoes, and send an agent to help.
- Every ladybug may be a robotic satellite-capable surveillance device.
- This organization will send agents to get in contact with kids in any country without any adult supervision.
- They have unlimited resources, including space stations, artificially intelligent bird-robocopters.

- Their agents are not only good, kind and helpful, they are adorable *stuffed animals*.

The message is so blunt that it's not even mildly hidden to require marxist analysis, this cartoon says, loud and clear, that unnamed organizations look at everything you do, but it's for your own good, and when those organizations enter your life, it's only to help you and protect you, and in the process, these virtuous groups become even more virtuous.

Having lived in latin america in the 70s and 80s, I can say: bugger, we wished for that to be so! In reality, these things usually hire very few stuffed animals, and quite a bunch of plain old animals.

The constant surveillance is not even thought about, it's just assumed to be there, there is no consideration that kids deserve, need or even *have* privacy or a privacy expectation, the ladybugs routinely film the kids in their homes or even bedrooms, and send the images to a satellite for automated monitoring. Jeremy Bentham lacked the imagination and technical resources to imagine this, so he had to put his prisoners in a circle, to be watched by mere human guards.

So, what can we do? Probably nothing. I fully expect my kid to grow up with no expectation of privacy, and no concept of doing things outside the purview of a government, officially or unofficially.

Is that evil? Maybe, but it will be their normal. Just like we don't expect to have silence, or private electronic communications unless we take specific measures (you all know that, right?), and we expect all our online actions to be tracked by someone (you do expect that, right?)

My hope is a world of hypocrites, who have a public facade and a secret life. I can only hope my son will become Batman.

Just saw a post in Google+ today by Amanda Blain, where she shows a picture of a book saying this:

Never give up on anybody. Miracles happen every day.

—Someone

Of course, no, they don't happen every day. They hardly ever happen or never happen, depending on how you define miracle. Things that happen every day are not miracles, they are common. It's cheapening the word.

I did reply, though:

If god is almighty, pray for the convergence of the -1^n series, and see how it works. There is no god but math.

—Me

Which is quite a troll, even for my standards, but hard to argue with, I expect. This post is a more serious response to that original quote. And my response is, sometimes,

you need to give up. Further, sometimes, not giving up is stupid, painful, dangerous and selfish.

Stupid People don't really change all that much. They do it very slowly, when they do. And you are not (usually) the other person's guardian. There comes a point in people's lives where helping them hurts them. Or worse, hurts the one helping. Consider an abusive partner. Why should you not give up? Why allow him/her to hurt you further, in hope for a future change? That is just stupid and...

Painful Because you are being hurt, either in body or in mind. And being hurt is, of course, bad. So why enable it? Why allow someone to cause you pain, just for his own sake? What are you *telling* that person? That you feel good about helping those who hurt you? That is...

Dangerous Because he may believe you. You are teaching that person that you feel good helping those who hurt you, and that makes you a better person, so he may just try to help you be *much* better by hurting you further. And really, if that's how you feel, you are being ...

Selfish Because helping someone just to feel self-righteous and pious and good is a scam. Help because you make *the other* feel good, not because it feels good for you. I see people trapped in abusive relationships, almost screaming "See how good I am! See the pain I take for love!" which is both selfish and insane.

So, give up. Because the path of the righteous man is beset on all sides by the inequities of the selfish and the tyranny of evil men. Blessed is he who, in the name of charity and good will, shepherds the weak through the valley of darkness, for he is truly his brother's keeper and the finder of lost children. But enough is enough. And stupid is bad.

The other day I was told by a friend of mine "you let your kid eat Chicken Nuggets at McDonalds? Do you know what they are made of? Amonia!" This is old stuff, but really, it's never too late to snicker at people. Let me show you a picture:

Figure 34: What a chicken nugget is made of. Maybe.

Surely, it can't be good to eat suspicious pink thing made of unidentifiable chicken bits, right? Sorry, I have no better way to say it, but, that's stupid. What that is, is a way to get every little bit of protein out of a dead chicken. And that is *good*. Because if we are going to kill a chicken, the least we can do is eat all of it. And this thing does such a good job even fussy five year olds *like* eating the unidentifiable chicken bits.

This is both ecologically sound and economically wise, but *smart*. If native americans did this, we would forever show it as an example of their moral superiority back when we had not killed them with chickenpox, along with their crafty woodcarving.

So, with that out of the way, let's look at another picture:

Figure 35: What Soylent Green will be made of.

If I lived in that future, and was told “Soylent Green is made of people! people!” (Should I have used a spoiler alert for a movie released when I was 2? In 1973?), I would say “cool, that guy always had good taste”. In the movie, it’s not like someone is kidnapping guys and feeding them to the soylentizer. It’s a voluntary affair. People say “I’m old, this life sucks, make me into green food”. Then, he’s euthanized and, later, processed and eaten. That’s not shocking, that is logical.

If I ever die (which I intend not to), you are all authorized to turn me into pink goo, dye me green, and turn me into special St Patrick’s day chicken McNuggets.

Bon appetit.

Sorry, spanish only, since it’s a video of me speaking in spanish ;-)

Finalmente, gracias a Juan Rodriguez Monti, video de mi charla “Haciendo trampa: Trucos para programar menos” de PyCon Argentina 2011.

Y los slides:

Lots of noise recently about Jeff Atwood’s post about why you should not learn to code. I am here now telling you you *should* learn to code. But only after you learn a few other things.

You should learn to speak. You should learn to write. You should learn to listen. You should learn to read. You should learn to express yourself.

Richard Feynman once described his problem solving algorithm as follows:

1. Write down the problem
2. Think real hard
3. Write down the solution

Most of us cannot do that because we are not Richard Feynman and thus, sadly, cannot keep all the solution in our head in step 2, so we need to iterate a few times, thinking (not as hard as he could) and writing down a bit of the solution on each loop.

And while we who code are unusually proud of our ability to write down solutions in such a clear and unforgiving way that even a computer can follow them, it’s ten, maybe a hundred times more useful to know how to write it down, or say it, in such a way that a human being can understand it.

Explanations fit for computers are bad for humans and viceversa. Humans accept much more compact, ambiguous, and expressive code. You can transfer high level concepts or design to humans much easier than to computers, but algorithms to computers much easier than to humans.

I have a distrust of people who are able to communicate to computers easier than with fellow humans, a suspicion that they simply have a hole in their skillset, which they could easily fix if they saw it as essential.

And it is an essential skill. Programmers not only run on coffee and sugar and sushi and doritos, they run on happiness. They have a finite endowment of happiness and they spend it continuously, like drunken sailors. They perform an activity where jokingly they measure productivity on curses per hour, a lonely endeavour that isolates them (us) from other humans, from family and friends.

If a developer cannot communicate he isolates. When he isolates he can't cooperate, he cannot delegate. He can't give ideas to others, he can't receive them, he can't share.

And since lots of our communication is via email, and chat, and bug reports, and blogs, it's better if he can write. A developer who cannot write is at a serious disadvantage. A developer who cannot write to express an idea cannot explain, he doesn't make his fellows better. He's a knowledge black hole, where information goes to die behind the event horizon of his skull.

So, learn to write. Learn to speak. Learn to read and listen. Then learn to code.

This is the procedure I use to prepare for speaking at conferences. It may work for others, it may not. It may be good, it may be bad.

1. Be asked to speak
2. Be unsure about it for a few days
3. Be asked if you are going to move your ass and do it already
4. Think title
5. Send proposal at deadline +/- 48 hours
6. Spend the next week or two thinking of cancelling
7. Figure out what the title means
8. Spend the next N-1 weeks until the conference with a constant dialog in your head about what you should talk about
9. Turn that constant unorganized dialog into a little story
10. Write down the story headings in slides
11. Attend the conference
12. Sit in the back while you polish slides
13. Start talking
14. Show slides
15. Stop talking

I added a very minor feature to the site. Up here ^ you should be able to see a link that says "reSt". If you click on it, it will show you the "source code" for the page.

I did this for a few reasons:

- 1) Because a comment seemed to suggest it ;-)
- 2) Because it seems like a nice thing to do. Since I so like reSt, I would like others to use it, too. And showing how easy it is to write using it, is cool.

- 3) It's the "free software-y" thing to do. I am providing you the preferred way to modify my posts.
- 4) It was ridiculously easy to add.

Also, if you see something missing, or something you would like to have on the site, please comment, I will try to add it.

Suppose you wanted to have free TV. Suppose by that you mean series, not movies or news (both of which you can get in other ways).

Also, suppose you want to have it on your actual TV. Also, that you don't mind breaking IP laws. Well, this is how I would do it, if that was the case.

First: get hardware.

You need a general computational device with TV connectors and lots of storage. Like, say, a Mele A1000 or A2000 running android, and a SATA disk. This will cost you around \$150 or so.

The Mele has HDMI, VGA and composite, so it can connect to pretty much anything that can display images. You could even use a 64GB SD card instead of a SATA drive if you want to go cheaper.

Second: choose some series.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "showrss".
```

You *could* go to [ShowRSS](http://showrss.karmorra.info/).. showrss: <http://showrss.karmorra.info/>

create an

```
system-message
WARNING/2 in everything.txt, line 34633
Block quote ends without a blank line; unexpected unindent. backrefs:
```

account and just choose series you like. ShowRSS would then provide you with a nice RSS feed updated as your shows are uploaded.

You could even choose between SD and HD versions.

Third: get the episodes.

You would need a BitTorrent client that supports this:

- 1) Android.
- 2) Choosing the destination folder (uTorrent for android doesn't)
- 3) RSS Feeds with autodownload

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "adownloader".
```

For example: [aDownloader](https://play.google.com/store/apps/details?id=com.AndroidA.DroiDow).. adownloader: <https://play.google.com/store/apps/details?id=com.AndroidA.DroiDow>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “can fix it”.

If you are using the Mele and the SATA drive, there is a bug in aDownloader where it can't write to the SATA disk. You can fix it.. can fix it: <http://forum.xda-developers.com/showthread.php?p=2162822>

if

system-message

WARNING/2 in `everything.txt`, line 34656
Block quote ends without a blank line; unexpected unindent. backrefs:

you have root and some patience.

Fourth: get a good video player.

You want one that:

- 1) Has HW decoding
- 2) Has SW audio decoding (a mele-specific issue)
- 3) Supports many video formats
- 4) Supports subtitles, if you need them

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “mx player”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “mx player”.

For example: **MX Player**.. mx player: <https://play.google.com/store/apps/details?id=com.mxtech.videoplayer.ad&t=2>

Fifth: get subtitles.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “subloader”.

If you ever watch something that is not in your native language, you also want something that automatically gets subtitles for your videos. For example **Subloader**.. subloader: https://play.google.com/store/apps/details?id=com.sregg.android.subloader&feature=search_result#?t=W2

How well would this work? Well, it would take roughly 5 minutes to download a show in SD or 15 minutes to download it in HD on a 10Mbps internet link.

It would let you play them in your own time.

It would let you see shows before they air in your country, and even if they will never air there.

It is also illegal as all hell, so don't do it.

Thanks to the cooperation of **Humitos** who gave me his wordpress backup, I did some improvements in the wordpress.com import feature of **Nikola, my static website/blog generator**

So, if you were to try to use `nikola_wordpress_importer` from master now, it would:

1. Not crash ;-)
2. Download attachments
3. Fix links to attachments so they work on the new site

However, I am now unsure of what exactly is in wordpress.com's export XML file. The posts themselves are in this form:

```
Muchas gracias Nico por hacer el video este. Grosos, quedó muy bueno.
```

```
[youtube=http://www.youtube.com/watch?hl=es&v=882qxARXa6c]
```

Two things jump to me:

1. That's not HTML
2. WTF is that youtube thing?

I am having some success processing it as markdown, since that handles the paragraph breaks and some other stuff. Maybe the youtube embedding is done with a markdown extension?

Anyone knows?

It has been said that if noone hates you, you are doing it wrong. I must be doing it quite wrong because there seem to be a lot of people who like me, for some reason. For example, today I was in a dark mood all afternoon. I wrote aggressive stuff on twitter and IRC.

And what do I get? People asking me what was wrong and offering help. It's unfair. Unfair to them because they should not be subject to the moods of any asshole (including me). I can only claim, in the way of excuses, that my wife no longer works next to me all day.

If she were there, and saw me acting like that, she would have slapped me back into sensible behaviour by reminding me that I am a lucky bastard. She doesn't even have to say anything, she just has to be there. So, she arrived, I noticed I am a lucky bastard, and I am all mellow now.

So, if I acted like a bastard with you today, or any other day, sorry dude, I am a bit of a bastard. But I am a lucky one, and when I don't forget that, I am, I think, bearable.

I found a camera today at home, and then it hit me: this was the camera we took to Istanbul, dropped on the floor, never worked again, and I never found after we came back! And it still had the SD card in it!

So, here are the pictures (not even filtered), so family can see them.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: `“a naked singularity“` by `sergio de la pava”`.

A while ago I finished reading “A naked singularity” by Sergio de la Pava.. “a naked singularity” by sergio de la pava: <http://www.goodreads.com/book/show/6643543-a-naked-singularity>

If you have not read it, stop reading this now, take a week off and read it first. You may love me for telling you that, or you may hate me with the intensity of a thousand suns, but I doubt you will find it in you to say “meh” after you do.

I will try not to go into plot details, even though it’s perhaps impossible to spoil this book. If I told you how it ends, it would not make any difference. If I told you about the chimp, or about the Casio Carousel, it may not matter. Or maybe it was a monkey, and the Carousel is done by Sony. Which it is, in the book. By Sony, not a monkey. I think.

Remember that kid in school that could tell the best jokes, and he did those funny voices? De la Pava *writes* in funny voices. And he breaks your heart with funny voices. His book doesn’t meander, it goes straight and quick, determined and in a hurry right out to the middle of nowhere, then keeps going, goes offroad, keeps going, runs through a few walls, and comes back here, by just going and going. It has a motor, it has a rocket engine, it never blinks.

This book is ostensibly about a lawyer straying his path. It’s probably written about something else. I have this strange feeling that most characters are imaginary, even though they talk and feel very real to me, who am writing this and thus am supposed to be more real than they are. It’s hallucinatory gritty realism. It’s poetic and technical. It’s bullshit as an art form.

Which is, really, what attracted me to it. The dialog, the monologues, are to regular bullshit like fine dinners are to hot dogs. Sergio de la Pava may be the finest bullshit-giver in this blue planet of ours (take it from one dabbler in the art), and we are all lucky to have him writing.

UPDATE: How could I forget to mention that this book, this novel, has some of the best box writing since Mailer? It’s an incredible feat, and I just forgot about it because there is so much stuff in it.

And yes, it has swords in it.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola”.

Since some people really don’t like Mako, and prefer Jinja2, I decided to make the template engine modular in Nikola.. nikola: <http://nikola.ralsina.com.ar>
my static site/blog generator.

All things considered, there is very little difference on the API side:

system-message

WARNING/2 in `everything.txt`, line 34815

Explicit markup ends without a blank line; unexpected unindent. backrefs:

I even ported the default theme to Jinja! And not all that much is different on the template side, either:

system-message

WARNING/2 in `everything.txt`, line 34820

Explicit markup ends without a blank line; unexpected unindent. backrefs:

But hey, to each his own. Other template engines are probably easy to plug, too.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “[here in github](#)”.

Branch up for review [here in github](#).. here in github: <https://github.com/ralsina/nikola/pull/6> and yes, anyone reading this who knows Jinja2 or python is welcome to review it.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: “[john carter](#)”.

Anoche ví [John Carter](#)... john carter.: <http://www.imdb.com/title/tt0401729/>

Recuerdo vagamente haber

system-message

WARNING/2 in `everything.txt`, line 34849

Block quote ends without a blank line; unexpected unindent. backrefs:

leído el libro en que está basado, “Princess of Mars” cuando tenía unos 8 años y era socio de la Biblioteca Mariano Moreno, sujeto a una dieta estricta de Hardy Boys, Bomba el chico de la selva, y Burroughs.

Me sorprende mucho que haya sido el fracaso económico que fue. Es divertida, está bien hecha, es un poco antigua en el sentido de que cuenta una historia de manera directa, sin vueltas. Hay que darle un premio especial al director porque ¡Las escenas de acción *se entienden*! Siempre sabes quién está haciendo qué cosa, y la relación espacial de la gente en las escenas tiene sentido.

Claro, la historia pasa rápido, hay montones de personajes secundarios que no se desarrollan porque no alcanza el tiempo, y es una pena que probablemente nunca veamos la segunda parte.

Understanding what goes on inside my head is not easy for me. I am separated from the thing I am examining by subjectivity, and suffer a comprehensive observer effect that clobbers simple reporting and disables my comprehension.

So, it's rare that I have a moment where I *feel* one way or another in a clear manner. To me "How are you?" is confusing. I tend to hesitate for ten seconds while scrambling for the standard response, like the terminator scrolling through response trees. Usually my state is, I feel exactly like myself. I feel the way I feel, and I just have not put that in words, scales, and comparisons, much less *one* word.

Half the time it's easier for me to know how others feel. I look at my wife and I *know*. I look at my son and I can probably tell you if he got a good lunch at school today, and whether he won his last rock paper scissors match.

I can do that because I can see them. I can see their faces, and I know how they look, how they change, how they react, I know Juan does this thing with his lip when he's frustrated, I know Rosario puts her sweater backwards if she's distracted.

I have to get my self-status indirectly. I woke up early and rested. I look forward to working, or to doing something in the weekend. I attack a task with intensity, I avoid an encounter, I forget to start music, I abandon projects, I reach out to people, I can't come up with ideas, I make up excuses, I make a quick joke.

I have to wonder where that comes from, then. Who is the me doing those things I examine to decide how I feel? I feel like my head is a town and I sit in a cafe, in a street table, and listen to the passers-by, gauging the mood.

Sometimes, just sometimes, I wish I was simpler, and straightforward. I wish I could do things without thinking so much. I wish I could react normally without intermediating myself in my own thoughts.

Of course maybe everyone does the same things. Maybe everyone is the same. Even if not the same, strange in the same way, just like things can be all different and part of a class, different in degrees and the same in essence.

I am fine, thanks. And you?

Spanish only!

Ayer Jorge Lanata, que siempre me cayó bien, y a quién siempre acompañé en sus aventuras editoriales (¡Compré su libro de relatos! ¡Extraño Crítica! ¡Me gustaba Silvina Chediek!) dijo una pelotudez bastante importante.

Lo chicanearon con aplicarle no sé que multa por fumar en un estudio de televisión, lo que está prohibido por ley. Entonces hizo un berrinche y dijo "voy a dejar de fumar cuando ustedes dejen de robar", seguido de explicaciones de cómo él no le hace mal a nadie, etc etc.

Supongamos que no le hace mal a nadie. Supongamos que "ellos" roban y no paran. ¿Y? ¿Qué pasa? No pasa nada, excepto que Lanata se da el gusto de no tener que parar de fumar dos horas. No hace ninguna diferencia excepto para él. Y mientras tanto seguiría rompiendo la ley, que es tan ley como la que prohíbe robar, ya que estamos.

Decir esa boludez que dijo Lanata es como hacer una huelga de morfi, o sea lo opuesto de una huelga de hambre. Es prometer no prenderse fuego hasta que los yanquis se vayan de Vietnam, es prometer comerse un chori todos los días hasta que el Tibet sea libre, es comer ensalada de cetáceo hasta que se salven las ballenas.

Si uno se va a poner en adalid de la ética (cosa que yo trato de no hacer porque es un laburo jodido, justamente), no puede dar pie, no puede dejarla picando, porque si la deja servida se la mandan a guardar. Lanata, la dejás servida. Largá el pucho. Pensá un rato. Después criticá.

Figure 36: The Lenovo mini wireless keyboard and trackball

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "dx".
```

Just got this today from DX.. dx: <http://dx.com>

and it's awesome.

1. I got it in a termosealed bag with a note from the Argentinian mail company saying "sorry it's broken, we got it that way".
2. The box looked like a rhino had seen it was on fire and tried to put it out.
3. It's spotless and works perfectly.

So, what is it? A RF (not bluetooth) mini-keyboard + trackball.

Does it work? Brilliantly. You take the dongle, plug it into a USB port, and it works, no pairing, no config, no drivers, no nothing.

But does it work on Linux? Yep, everything, from the volume keys, to the "player" keys, to the weird orange key that apparently launches nautilus. All the multimedia keys work, with OSD feedback, and they do the obvious thing on clementine.

Are the keys nice? They are a lot like an expensive QWERTY slider phone's. No travel, but nice feedback, and the layout is reasonable. It's mostly a thumb-keyboard, and the shape works nicely for that.

And the trackball? Works with the thumb. I woul like a "press-the-trackball" to click, and strangely it's a bit noisy! But not bad at all.

What else? Light, has a compartment to store the dongle, seems rather robust. Comes with batteries.

So far, really happy with it, will use it for my living room computer/mediathing. This sign (the one on the left) is on the Education Council of the place where I live.

There is, really, no such thing as art. There are only artists.

—Gombrich

When I first read it I was shocked by the sheer stupidity of the thing, but hey, let's think it over a bit. I suppose that saying someone is an artist, meaning a person that creates art, and at the same time claim that art, as such, does not exist, sounded like a good idea at the time. It's the kind of counter-intuitive slogan that makes people say "oh, deep, dude" between bong hits.

It's deeply stupid literally, in that it would, of course, make artists people who create something that doesn't exist, which puts them at the level of aspiring unicorn wranglers and theologians in the futility of their chosen profession. But in a way, it *is* true. Because "art" is not a thing, it has no material existence, although it has many examples.

So, pieces of art exist, but art as a whole doesn't. Art would be the platonic idea from which sculptures, paintings and novels are but a reflection. Which is a metric ton of bullshit but is at least somewhat defensible. And of course whether something is art or not is completely subjective, so art is like a club where objects enter or not based on opinion, and that club exists only in each person's head.

But in any case, that doesn't make it not existing, just because something only exists in your head that doesn't mean it doesn't *exist*. It means it exists in your head, no matter how silly that sounds. So, art does exist, in the same way that imagination exists, or memories exist, or thought exists. Which again goes to show Gombrich was just crafting a silly slogan.

So, we have established that existence is not exactly a black and white thing, because there is at least material existence, and things that don't exist materially? Wrong, because your brain processes are material too. Thought, imagination, and art, all exist materially, in your brain. We just don't have the instruments to measure them, or report on them, except via that unreliable thing called people who just will not shut up about all those things they perceive in their brains. So, art exists, materially.

I know that may sound slightly strange, but what *doesn't* exist? Do the dragons of Pern exist? Yes, they exist, you can buy "The Dragons of Pern" in Amazon. And yes, specific dragons exist as well, because if they did not, how could we know they are carnivorous, oviparous and warm-blooded? Sure, they don't exist materially in the form of dragons, but I know that because I *remember* them. How could I remember something that doesn't exist? I remember them because I read about them. I have a memory of their description. There is a description of Pernese dragons as written by Anne McCaffrey, and that's not as good as actual, touchable, warm dragons, but it's the next best thing.

If I mention pink elephants, I am bringing them into existence, not as elephants, but as a description of an elephant, pink. It's an attenuated existence, but is the same one Japan has for me, who have never been there and must make do with notoriously unreliable testimony about their exotic temples, bizarre habits and enormous fire-breathing atomic lizards.

Obviously this is not what people mean, in daily usage, when they say existence, since it would include things people are happy to say *do not* exist, like Pernese dragons, pink elephants and Gojira. So, in a twisted way, maybe Gombrich is right. Except of course that existence is not a democratic decision. So maybe just people are wrong to say Gojira doesn't exist.

Or maybe existence is not a useful property for things. Maybe what we should use is reality. Because while Gojira exists, he is not real, in the sense that his material form is not that of a huge Tokio-stomping man in a rubber suit. Even though there used to be a guy, dressed on a rubber suit, stomping on a "Tokio", but I digress.

So, art is not real, maybe? Well, no, because noone is claiming art has a physical nature distinct from that representation it has in our heads. Noone claims art is yellow, smaller than a teapot, and covered in purple hair. Art is an abstract concept. And abstract concepts exist.

So Gombrich was full of shit.

Sorry, spanish Only, because it's my son speaking ;-)

Empecemos con un par de citas:

No está científicamente demostrado que el alma no se transmita en un trasplante de órganos.

—Mariló Montero

Sabías que cada siete minutos, en este país, nace un negro sin alma?

—BB King

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "dijo esa pelotudez".

Mariló Montero es una persona que trabaja en televisión, y [dijo esa pelotudez.. dijo esa pelotudez](http://alt1040.com/2012/10/marilo-montero-alma): <http://alt1040.com/2012/10/marilo-montero-alma>

. Podría hacer un post explicando educadamente que el alma no se aloja en las córneas, ni en el bazo, pero vamos directo a lo importante: el alma no existe.

El alma, todos los que creen que tenemos te lo van a decir, es inmaterial. Las cosas inmateriales no existen. El alma es útil como metáfora, es útil como atajo para describir un conjunto de cosas relacionadas con la personalidad y la conciencia, pero el alma en sí no es una cosa.

Y no, las no-cosas no viven en el intestino, así que no, no se transmiten.

¿Si no, cuando te extirpan las amígdalas, te volvés un poco más desalmado?

¿Cuando cagás, mandás tu alma a Berazategui?

¿Cuando te cortás el pelo, te acortás el espíritu?

¿Si me transplantan algo de Mariló Montero, me vuelvo idiota?

Creo que lo de la señora Montero es simplemente un intento de modernizar alguna superstición medieval, y siendo ella española, es posible suponer que es algo católico. Supongo que el salto de una religión a un chorizo lleno de alma de vaca es breve.

Last year I was in Budapest for UDS-O. Not many pictures since I had no camera other than a very bad phone.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "here they are".

But [here they are](http://ralsina.com.ar/galleries/budapest/).. here they are: <http://ralsina.com.ar/galleries/budapest/>
Sorry, spanish only.

“¿Argentino?” me preguntó la camarera del Fay Fay. Pero claro, cenando solo, en Carmelo, no te conocen, calculo que un 80% de probabilidades de que seas argentino. En fin. Si no te gusta llamar la atención te conviene ir a Montevideo, o por lo menos a Colonia. Pero la cachola es más barata, y para lo que tenés que hacer Carmelo alcanza. Y son las once de la noche, y el hotel no tiene agua caliente, y el único lugar abierto es el Fay Fay, y el carrito de chacho, y está fresco para comer parado en la calle.

Pero bueno, uno está donde está y hace lo que hace, y eso te preguntan siempre, qué estás haciendo. Y se te ocurre inventar que si tomás las banconos de todos tus amigos, venís acá, sacás pesos uruguayos del cajero, comprás dólares en la casa de cambio, los llevás a Argentina, los cambiás por euros, los traés acá, y los volvés a cambiar por dólares eso te deja un 7.8% de ganancia y es legal, pero sería complicado y no es cierto, y no querés explicar, entonces tenés que pensar algo, y tirás que venís porque sos un viajero en el tiempo.

Que somos todos, porque vamos para el futuro a un día por día, pero que tenés una teoría, que acá no es el mismo momento que allá. Que se llaman igual, en todos lados se llama 25 de mayo del 2012, pero que a pesar de eso, acá se siente como si fuera, ponele, 22 de junio de 1987, pero con wifi.

Si, Carmelo es la década del 80 pero con wifi, esa es la excusa, y no, si no hubiera wifi no vendría, por una cuestión de laburo, ¿viste? y empezás a mencionar las cosas obvias, como que esos sauces llorones en el río son iguales a los que había en la vuelta del pirata cuando eras chico, y que hay bosta en la calle, y que todas las veredas tienen esa capa de tierra que vos no ves desde 1991, y los carteles pintados a mano en las vidrieras.

Pero por supuesto, la evidencia más clara de que en Uruguay no es el siglo 21 es Natalia Oreiro. Porque la Oreiro no sólo está muy buena, si no que es, claramente, una mina nacida en 1950. Alcanza verle la cara, no es una cara de ahora, es una cara de hace 30, 40, 50 años, por eso llama la atención, porque minas lindas hay millones, pero ella es linda como sería tu vieja cuando era joven, si tu vieja era linda, como tu vieja debería haber sido de linda.

Es que las chicas jóvenes son siempre lindas. El chiste es ver cuarentonas lindas. En un lugar en que las adolescentes son hermosas pero las de cuarenta no, es que algo falla, es que algo les pasó en esos veinte años que les sacó lo lindas, ¿no?

Por eso, para saber cual país tiene las minas más lindas no mires adolescentes, no mires modelos, ni actrices, mirá maestras de primaria, ponele, de entre 35 y 50. Si son lindas, es que ese país vale la pena, porque agarró lindas adolescentes, las puso en un laburo jodido e ingrato, pero así y todo no las arruinó, quiere decir que el resto de las cosas que les pasaron no fueron tan malas. Si parecen de 20 sería una cagada, tienen que parecer de 35, de 40, de 50, pero con una sonrisa, con una actitud, de mina linda.

Entonces te decía que eso, que vengo por la máquina del tiempo, y sí, quiero volver a los 90 porque eran más largos, los 90. Fijate, el gobierno de Mendez duró tipo lo mismo que Nestor más Cristina, pero decíme que no pareció el doble, el triple... sería que yo era más rápido en los 90, entonces ahora que soy más lento el mundo se me puso rápido, y se me van los años así, de golpe, entonces si vengo acá, en el área de influencia de la hipotética máquina del tiempo de Natalia, entonces los días se me hacen más largos, y funciona, te juro, llevo 4 horas acá, pero se sienten como 4 días, 4 semanas, claro, 4 días de éstos, no 4 días de los 90.

Pero bueno, sí, de paseo. Un canadiense y una coca light.

The next time someone suggests you take a dietary supplement, or a plant or herb extract, or tells you to eat, drink, apply on your skin, inhale, or do anything with the argument that it's natural, remember this:

NATURE HATES YOU

Almost everything that comes from nature will kill you. All animals would prefer that you die, all plants are toxic, and it's only through a long evolutionary process that you are *barely* capable of digesting any vegetables.

There is a reason why we take aspirin instead of chewing bark, and why we cook meat before we eat it. Freaking potatoes are toxic if eaten raw.

Give parsley to a parrot, give chocolate to a dog, give celery to a cat, give hemlock to you: DEAD. Because all around you is trying to kill you. All the time.

Nature is not a mother, nature is actively hostile to you and will be happy to make your life as hard, brief and painful as possible. Man has changed that. Saying "oh, but this is natural" is an indictment, a warning sign.

Raw foodists, paleo diet advocates, herbal-supplement-chuggers: you are all the beneficiaries of a long history of knowing what is semi-safe to eat because everything else will **kill you**.

I will not eat anything nature gives me until it has been carefully tested by men and science. Because science is good and nature is evil.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "nikola".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola”.

This is a new feature in Nikola.. nikola: <http://nikola.ralsina.com.ar>, my static website and blog generator.

It’s aimed at showing *pieces* of files in the posts, and giving you a link to the full file, does reasonable syntax highlight thanks to pygments, and the syntax for embedding in your posts is not excessively awful, so I can use it for some of my tutorial / book projects.

Here’s an example:

```
.. listing:: md.py python
   :start-at: def compile_html
   :end-before: output =
```

This is not merged into master yet, but should be soon.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola static site/blog generator”.

I implemented a first draft of a wordpress.com importer for my [Nikola static site/blog generator](http://nikola.ralsina.com.ar).. nikola static site/blog generator: <http://nikola.ralsina.com.ar>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “this blog”.

It’s a work in progress but it works well enough that it can turn [this blog](#) into [this blog](#)

What works:

- Posts
- Pages
- Drafts
- Tags
- Personal information

What doesn’t work:

- Comments
- Links across posts/pages are left untouched
- The output is pretty generic

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “master”.

This needs testing, so feel free to get `master.. master`: <http://github.com/ralsina/nikola>

and

```
system-message
WARNING/2 in everything.txt, line 35310
Block quote ends without a blank line; unexpected unindent. backrefs:
```

try it. Assuming `foo.xml` is your data from `wordpress.com`:

```
nikola_import_wordpress foo.xml
cd new_site
doit && doit serve
```

And point your browser to <http://localhost:8000> to see the result :-)

Minor tweaking of the generated `conf.py` is generally needed for improved results.

A simple yet powerful and flexible static website and blog generator, based on `doit`, `mako`, `docutils` and `bootstrap`.

I built this to power this very site you are reading, but decided it may be useful to others. The main goals of Nikola are:

- Small codebase: because I don't want to maintain a big thing for my blog
- Fast page generation: Adding a post should not take more than 5 seconds to build.
- Static output: Deployment using `rsync` is smooth.
- Flexible page generation: you can decide where everything goes in the final site.

```
system-message
• WARNING/2 in everything.txt, line 3
  Duplicate explicit target name: "mako".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "mako".
```

Powerful templates: Uses `Mako`.. `mako`: <http://www.makotemplates.org/>

```
system-message
• WARNING/2 in everything.txt, line 3
  Duplicate explicit target name: "docutils".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "docutils".
```

Clean markup for posts: Uses `Docutils`.. `docutils`: <http://docutils.sourceforge.net/>

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “doit”.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “doit”.

Don't do stupid builds: Uses `doit`.. `doit`: <http://python-doit.sourceforge.net/>

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “bootstrap”.

Clean HTML output by default: Uses `bootstrap`.. `bootstrap`: <http://twitter.github.com/bootstrap/>

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “disqus”.

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “disqus”.

Comments out of the box: Uses `Disqus`.. `disqus`: <http://disqus.com/>

- Tags, with their own RSS feeds
- Easy way to do a blog
- Static pages outside the blog
- Multilingual blog support (my own blog is english + spanish)

I think this initial version achieves all of those goals, but of course, it can be improved. Feedback is very welcome!

Nikola's home page is currently <http://nikola-generator.googlecode.com>

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “this very site”.

Version 1.2 of Nikola, my static site generator and the software behind [this very site](http://lateral.netmanagers.com.ar).. [this very site](http://lateral.netmanagers.com.ar): <http://lateral.netmanagers.com.ar>, is out!

Why build static sites? Because they are light in resources, they are future-proof, because it's easy, because they are safe, and because you avoid lockin.

New Features:

- Image gallery (just drop pics on a folder)
- Built-in webserver for previews (`doit -a serve`)
- Helper commands to create new posts (`doit -a new_post`)
- Google Sitemap support
- A Handbook!
- Full demo site included
- Support for automatic deployment (`doit -a deploy`)
- Client-side redirections

And of course the *old* features:

- Write your posts in reStructured text
- Clean, customizable page design (via bootstrap)
- Comments via Disqus
- Support any analytics you want
- Build blogs with tags, feeds, feeds for your tags, indexes, and more
- Works like a simple CMS for things outside your blog
- Clean customizable templates using Mako
- Pure python, and not a lot of it (about 600 lines)
- Smart builds (doit only rebuilds changed pages)
- Easy to extend and improve
- Code displayed with syntax highlighting

Right now Nikola does literally everything I need, so if you try it and need something else... it's a good time to ask!

More info at <http://nikola-generator.googlecode.com>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "nikola".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "nikola".

By popular request, Nikola.. nikola: <http://nikola.ralsina.com.ar>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “github”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “github”.

now has its source code at [GitHub](https://github.com/ralsina/nikola).. github: <https://github.com/ralsina/nikola>

Also, if you tried version 2.1 and it failed, try 2.1.1, because I forgot to add a couple of files in one of the themes in 2.1.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “official nikola blog”.

Released version 2.1 of Nikola, my static blog/site generator. More information at the [Official Nikola Blog](http://nikola.ralsina.com.ar/blog/version-21-released.html).. official nikola blog: <http://nikola.ralsina.com.ar/blog/version-21-released.html>

in the

system-message

WARNING/2 in `everything.txt`, line 35440
Block quote ends without a blank line; unexpected unindent. backrefs:

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “official nikola site”.

[Official Nikola Site](http://nikola.ralsina.com.ar).. official nikola site: <http://nikola.ralsina.com.ar>

which are, of course, done with Nikola :-)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola”.

Version 3.0.1 of [Nikola](http://nikola.ralsina.com.ar).. nikola: <http://nikola.ralsina.com.ar>

my static site generator

system-message

WARNING/2 in `everything.txt`, line 35448

Block quote ends without a blank line; unexpected unindent. backrefs:

is ready for initial user testing.

I have merged a bunch of patches from Kay Hayen and Kadefor:

- `disqus_developer` is gone
- `addthis` buttons are optional (controlled by a new option)
- You can have more than one static files folder to be merged into output.
- Image galleries support uppercase extensions.
- Templates can not link to CSS that is empty/missing
- Better Google Sitemaps
- Preserve (some) metadata when copying files, like timestamps
- Don't overwrite existing posts with `doit new_post`
- Generate valid URLs from unicode titles

So, if you want to try it out:

<https://github.com/ralsina/nikola/zipball/3.0.1>

I will make it a real release if nothing bad shows up before sunday or monday.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "nikola".

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "nikola".

I just released version 3 of my static site generator, **Nikola**.. nikola: <http://nikola.ralsina.com.ar>

It's a *major* release, there is hardly any code from the previous version that was not moved, prodded, broken or fixed!

The main features of Nikola:

- Blogs, with tags, feeds, archives, comments, etc.
- Themable
- Fast builds, thanks to `doit`
- Flexible
- Small codebase (programmers can understand all of Nikola in a few hours)
- `reStructuredText` and `Markdown` as input languages
- Easy image galleries (just drop files in a folder!)

- Syntax highlighting for almost any programming language or markup
- Multilingual sites
- Doesn't reinvent wheels, leverages existing tools.

Changes for this release (not exhaustive!):

- New optional template argument for “nikola init”
- New “install_theme” task
- Optional address option for the “serve” task
- Better looking codeblocks
- Russian translation
- Use markdown/reSt compiler based on post extension
- Don't fail when there are no posts/stories/galleries/tags
- Use configuration options as dependencies
- Use more relative links for easier site relocation
- Syntax highlight for markdown
- Better multicore builds (make the -n 2 or -n 4 options work)
- Configurable output folder
- Don't fail on posts with periods in the name
- Different page names for different languages
- Recognize (some) Mako template dependencies
- Is now a more “normal” python package.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “nikola my static site/blog generator”.
```

Version 4.0.2 of [Nikola my static site/blog generator](http://nikola.ralsina.com.ar).. nikola my static site/blog generator: <http://nikola.ralsina.com.ar>

is ready

```
system-message
WARNING/2 in everything.txt, line 35545
Block quote ends without a blank line; unexpected unindent. backrefs:
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “here”.
```

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “here”.

for download. More details [here](#).. here: <http://nikola.ralsina.com.ar/blog/nikola-402-released.html>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “release 4 of nikola”.

I have just uploaded [release 4 of Nikola](#).. release 4 of nikola: <http://nikola.ralsina.com.ar>
my static blog/site generator. Here are some of the changes in

system-message

WARNING/2 in `everything.txt`, line 35553
Block quote ends without a blank line; unexpected unindent. backrefs:

this version:

Features

- Previous/Next post links
- Teaser support
- Support posts with HTML “sources”
- Site checking script (`nikola_check`)
- Maximum image size in galleries
- Image descriptions in galleries
- Image exclusion in galleries
- Special “draft” tag
- Pretty code listings (“code galleries”)
- Page descriptions
- Easy theme tuning via Bootswatch
- Support for WebAssets bundles
- “Filters” for powerful file post-processing

Bugfixes

- Improved HTML output
- Support multiple time formats in post metadata
- Slugify tag names for URLs
- Archive path and filename configurable
- Galleries sorted by date (supports EXIF)
- Rotate gallery thumbnails (EXIF)
- Tag feeds in tag pages
- Colorbox support in restructured text figures
- Fix for content displaying too wide
- Changelog

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola-discuss group”.

As usual, you can discuss bugs or make feature requests at the [nikola-discuss group](#).
nikola-discuss group: <http://groups.google.com/group/nikola-discuss>
and I love to know of sites using Nikola.
Hope you like it!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola”.

One of the things I hacked at during this PyCamp is trying to figure out a *niceworkflow* for Nikola.. nikola: <http://nikola.ralsina.com.ar>

, something that will enable users that are not so technical, to use it.

One first step is Nikola-as-a-service, which is meant for technical users anyway but lays down the infrastructure for this to work semi-nicely.

In the video below, you will see me do this:

- Go to GitHub
- Take a starter’s blog I provided, and do a clone
- Go to the nikola-as-a-service site, and login (via twitter)
- Create a site using my fork’s repo URL

- Get a “webhook” URL, and add it to my fork’s admin as a post-commit hook
- Edit a file in github’s web UI and commit it (you can of course just push from any github client)
- Automatically, the site nikola-as-a-service publishes gets updated.

Please don’t try to use this service yet because:

- 1) It’s running in a \$4.50/month server
- 2) It’s the same server my own blog uses
- 3) I will turn it off, delete everything, etc. every once in a while
- 4) I am editing the code on the server, so no guarantees it will not just *stop working*.

So, here’s the video:

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “nikola”.
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “nikola”.
```

Two upcoming features for the next release of Nikola.. nikola: <http://nikola.ralsina.com.ar>, my static site generator, due sometime in August.

Filters

Filters let you postprocess your output. Think of it like instagram for websites, but useful. You can configure per file extension a series of python functions or shell commands, which will be applied in place to the output file.

For example, suppose you want to apply yui-compressor to your CSS and JS files:

```
\ :n: 'FILTERS' \ \ :o: '=' \ \ \ :p: '{' \ \
  \ :s: '".css"' \ \ \ :p: ':' \ \ \ :p: '[' \ \ \ :n: 'filters' \ \ \ :o: '.' \ \ \ :n: 'yui'
  \ :s: '".js"' \ \ \ :p: ':' \ \ \ :p: '[' \ \ \ :n: 'filters' \ \ \ :o: '.' \ \ \ :n: 'yui'
\ :p: '}' \ \
```

There, `filters.yui_compressor` is a simple wrapper around the command so that it applies in-place to the output files.

If you use strings there (untested), they are taken as commands. The “%s” will be replaced by the filename, the usual crazy shell quoting rules apply:

```
\ :n: 'FILTERS' \ \ \ :o: '=' \ \ \ :p: '{' \ \
  \ :s: '".jpg"' \ \ \ :p: ':' \ \ \ :p: '[' \ \ \ :s: 'jpegoptim ' \ \ \ :si: '%s' \ \ \
  \ :s: '".png"' \ \ \ :p: ':' \ \ \ :p: '[' \ \ \ :s: 'pngoptim ' \ \ \ :si: '%s' \ \ \
\ :p: '}' \ \
```

Keep in mind that the filters modify the *output* of Nikola, not the input, so your images, CSS, and JS files will not be touched in any way. And of course changing the filters applied to a file will force a rebuild, so you can experiment freely.

Bundles

Having many separate CSS or JS files is usually a nono for performance reasons because each one may involve a separate HTTP transaction. The solution is to “bundle” those files in a single, larger file.

The reason not to do that is that usually it means having a huge, uncomfortable thing to handle. So Nikola tries to give you the best of both worlds, by letting you have separate files, and bundling them (or not) on build.

There is a new option, `USE_BUNDLES` that defaults to `False`, and there are some changes in the theme templates so that it uses the bundled version when needed.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “webassets”.
```

This was only possible thanks to [Webassets](http://webassets.readthedocs.org/).. webassets: <http://webassets.readthedocs.org/>. However, if you don’t have Webassets installed, or you don’t enable `USE_BUNDLES`, this should cause no changes in the output.

Conclusion

These new features will allow Nikola users to improve their site’s performance with minimal tweaking, which is always a good thing.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “nikola”.
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “nikola”.
```

This friday is the beginning of PyCamp, four days of python hacking without distraction or pause. And I want to code a lot. My main target is features for [Nikola](http://nikola.ralsina.com.ar).. nikola: <http://nikola.ralsina.com.ar>

my static blog generator.

If you are attending PyCamp (or even if you are not), you are welcome to join me in implementing these in a marathon of kickass coding starting this friday and lasting all weekend.

I have a few ideas in my head, but I want *more*. These are the ones I have, please add more in the comments if you have any:

Code Gallery Like image galleries but for code. Put code in a folder and it will be beautifully displayed. With the addition of a “listings” docutils directive, it will make showing code in detail and in context easy and powerful, and make Nikola more attractive to programmer-bloggers.

Gallery Polishing Image galleries are implemented and work, but they could use a lot of polish. From making them more network-efficient, to image RSS feeds, recursive galleries, gallery metadata, image texts, and much more.

File Pipelines Want to minimize your CSS? Tidy your HTML? pngcrush your images? apply HTML transformations? Other things I can't imagine?

File pipelines would bring the power of the unix shell to a site generator, letting you connect lego-like filters, some provided, some from the community, into a powerful machinery.

Online Editing (Alva) While static site generators have lots of benefits, they have one significant downside: you edit the files in your own device. A online editor for Nikola lets you edit them through a web interface for blogging-from-anywhere goodness.

Nikola Hosting (Shoreham) Why not create a service where the user feeds posts to a server and then the server publishes them? The feeding can be via a DVCS, or a file sync service, or via online editors, and the output is published automatically or at the push of a button.

Drafts I don't do drafts. I type and that's it. But others prefer more cautious and sane approaches. So, how should drafts work? While the feature may be easy to implement, it's a good beginner programmer's task, where you have to think more about what you want to achieve and providing a good user experience than about just banging code.

So, is there something you saw in another static blog generator and Nikola lacks? Any cool ideas and want a friendly codebase to hack them on? Do you have any crazy ideas noone would touch with a ten-foot-pole but you think would be awesome to have? Well, now's a good time to talk about it!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "nikola's design goals".

A fast-loading site is a good thing. Your site's speed matters. Which is why creating fast sites is one of [Nikola's design goals](http://nikola.ralsina.com.ar).. nikola's design goals: <http://nikola.ralsina.com.ar>

Sure, it's not meant to create e-commerce sites, but if you are doing just a simple corporate site, or a personal blog, why would you not make it as fast as possible anyway?

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "one data point, from kay hayen".

So, here's [one data point, from Kay Hayen](http://www.nuitka.net/posts/nikola-sp).. one data point, from kay hayen: <http://www.nuitka.net/posts/nikola-sp>

Figure 37: Guess when Kay switched to Nikola.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "this blog".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "this blog".
```

But you don't have to just believe that graph. Here's Yahoo's YSlow report on [this blog](http://lateral.netmanagers.com.ar).. this blog: <http://lateral.netmanagers.com.ar>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "$5 vps".
```

, which is hosted in a \$5 VPS.. \$5 vps: <http://burst.net/linvps.shtml>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "quassel".
```

(as are other sites, BTW. The *same* VPS. As is a bunch of services. And my [Quassel](http://quassel-irc.org/).. quassel: <http://quassel-irc.org/>
core)

It scores **94** on that test. That's 94 out of 100 possible points in the "Small Site or Blog" ruleset.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "apache".
```

How much server tuning was needed? I added one line to the [Apache](http://httpd.apache.org).. apache: <http://httpd.apache.org>

config:

```
AddOutputFilterByType DEFLATE text/html text/plain text/xml text/css
```

That makes it compress those files before sending it to the user and improved the third item from F to A.

So, when I say Nikola makes your site fast, don't take my word for it, just check it out.

I managed to do some minor work today on Nikola, the static website generator used to generate ... well, this static website.

- Implemented tags (including per-tag RSS feeds)
- Simplified templates
- Separated code and configuration.

The last one was the trickiest. And as a teaser, here is the *full* configuration file to create this site, except HTML bits for analytics, google custom search and whatever that would make no sense on other sites. I hope it's somewhat clear.

```
\ :c:`# -*- coding: utf-8 -*-`

\ :c:`# post_pages contains (wildcard, destination, template) tuples.`
\ :c:`#`
\ :c:`# The wildcard is used to generate a list of reSt source files (whatever)`
\ :c:`# That fragment must have an associated metadata file (whatever/thing)`
\ :c:`# and optionally translated files (example for spanish, with code "es")`
\ :c:`#     whatever/thing.txt.es and whatever/thing.meta.es`
\ :c:`#`
\ :c:`# From those files, a set of HTML fragment files will be generated:`
\ :c:`# whatever/thing.html (and maybe whatever/thing.html.es)`
\ :c:`#`
\ :c:`# These files are combined with the template to produce rendered`
\ :c:`# pages, which will be placed at`
\ :c:`# output / TRANSLATIONS[lang] / destination / pagename.html`
\ :c:`#`
\ :c:`# where "pagename" is specified in the metadata file.`
\ :c:`#`

\ :n:`post_pages` \ :o:`=` \ :p:`(`
    \ :p:`(` \ :s:`"posts/*.txt"` \ :p:``,` \ :s:`"weblog/posts"` \ :p:``,`
    \ :p:`(` \ :s:`"stories/*.txt"` \ :p:``,` \ :s:`"stories"` \ :p:``,`
\ :p:`)`

\ :c:`# What is the default language?`

\ :n:`DEFAULT_LANG` \ :o:`=` \ :s:`"en"`

\ :c:`# What languages do you have?`
\ :c:`# If a specific post is not translated to a language, then the version`
\ :c:`# in the default language will be shown instead.`
\ :c:`# The format is {"translationcode" : "path/to/translation"}`
\ :c:`# the path will be used as a prefix for the generated pages location`
```

```

\ :n: 'TRANSLATIONS' \ \ :o: '=' \ \ :p: '{' \
  \ :s: '"en"' \ \ :p: ':' \ \ :s: '"' \ \ :p: ', ' \
  \ :s: '"es"' \ \ :p: ':' \ \ :s: '"tr/es"' \ \ :p: ', ' \
  \ :p: '}' \

\ :c: '# Data about this site' \
\ :n: 'BLOG_TITLE' \ \ :o: '=' \ \ :s: '"Lateral Opinion"' \
\ :n: 'BLOG_URL' \ \ :o: '=' \ \ :s: '"http://lateral.netmanagers.com.ar"' \
\ :n: 'BLOG_EMAIL' \ \ :o: '=' \ \ :s: '"ralsina@kde.org"' \
\ :n: 'BLOG_DESCRIPTION' \ \ :o: '=' \ \ :s: '"I write free software. I have a
  \ :s: '"everything. I write quickly. A weblog was inevitable."' \

\ :c: '# Paths for different autogenerated bits. These are combined with the
\ :c: '# paths.' \

\ :c: '# Final locations are:' \
\ :c: '# output / TRANSLATION[lang] / TAG_PATH / index.html (list of tags)' \
\ :c: '# output / TRANSLATION[lang] / TAG_PATH / tag.html (list of posts for
\ :c: '# output / TRANSLATION[lang] / TAG_PATH / tag.xml (RSS feed for a tag
\ :n: 'TAG_PATH' \ \ :o: '=' \ \ :s: '"categories"' \
\ :c: '# Final location is output / TRANSLATION[lang] / INDEX_PATH / index-
\ :n: 'INDEX_PATH' \ \ :o: '=' \ \ :s: '"weblog"' \
\ :c: '# Final locations for the archives are:' \
\ :c: '# output / TRANSLATION[lang] / ARCHIVE_PATH / archive.html' \
\ :c: '# output / TRANSLATION[lang] / ARCHIVE_PATH / YEAR / index.html' \
\ :n: 'ARCHIVE_PATH' \ \ :o: '=' \ \ :s: '"weblog"' \
\ :c: '# Final locations are:' \
\ :c: '# output / TRANSLATION[lang] / RSS_PATH / rss.xml' \
\ :n: 'RSS_PATH' \ \ :o: '=' \ \ :s: '"weblog"' \

\ :c: '# A HTML fragment describing the license, for the sidebar.' \
\ :n: 'LICENSE' \ \ :o: '=' \ \ :s: '""'
  <a rel="license" href="http://creativecommons.org/licenses/by-nc-sa/2.5
  </a>
"" \

\ :c: '# A search form to search this site, for the sidebar. Has to be a <li
\ :c: '# for the default template (base.tmpl).' \
\ :n: 'SEARCH_FORM' \ \ :o: '=' \ \ :s: '""'
  <!-- google custom search -->
  <!-- End of google custom search -->
"" \

\ :c: '# Google analytics or whatever else you use. Added to the bottom of <
\ :c: '# in the default template (base.tmpl).' \
\ :n: 'ANALYTICS' \ \ :o: '=' \ \ :s: '""'
  <!-- Start of StatCounter Code -->
  <!-- End of StatCounter Code -->

```

```

        <!-- Start of Google Analytics -->
        <!-- End of Google Analytics -->
    """"\

\ :c:`# Put in global_context things you want available on all your templats
\ :c:`# It can be anything, data, functions, modules, etc.`\
\ :n:`GLOBAL_CONTEXT`\ \ :o:`='\ \ :p:`{'\
\ :s:`analytics`\ \ :p:`:\ \ \ :n:`ANALYTICS`\ \ :p:`',`\
\ :s:`blog_title`\ \ :p:`:\ \ \ :n:`BLOG_TITLE`\ \ :p:`',`\
\ :s:`blog_url`\ \ :p:`:\ \ \ :n:`BLOG_URL`\ \ :p:`',`\
\ :s:`translations`\ \ :p:`:\ \ \ :n:`TRANSLATIONS`\ \ :p:`',`\
\ :s:`license`\ \ :p:`:\ \ \ :n:`LICENSE`\ \ :p:`',`\
\ :s:`search_form`\ \ :p:`:\ \ \ :n:`SEARCH_FORM`\ \ :p:`',`\
\ :c:`# Locale-dependent links`\
\ :s:`archives_link`\ \ :p:`:\ \ \ :p:`{'\
\ :s:`es`\ \ :p:`:\ \ \ :s:`<a href="/tr/es/weblog/archive.html`
\ :s:`en`\ \ :p:`:\ \ \ :s:`<a href="/weblog/archive.html">Archiv
\ :p:`'},`\
\ :s:`tags_link`\ \ :p:`:\ \ \ :p:`{'\
\ :s:`es`\ \ :p:`:\ \ \ :s:`<a href="/tr/es/categories/index.htm
\ :s:`en`\ \ :p:`:\ \ \ :s:`<a href="/categories/index.html">Tag
\ :p:`'},`\
\ :p:`'}'\

\ :nb:`execfile`\ \ :p:`(' \ \ :s:`"nikola/nikola.py"\ \ :p:`')`

```

Yes, I know that's not a real word.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "nikola".

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "nikola".

So, the git master of Nikola.. nikola: <http://nikola.ralsina.com.ar>

now has plugins.

system-message

WARNING/2 in everything.txt, line 35969
Block quote ends without a blank line; unexpected unindent. backrefs:

In fact, not only does it have plugins, but is basically *made* of plugins.

You see, the code used to mostly be inside a class (called Nikola), and ... it had grown. That class had grown up to around 2000 lines of code. Which is completely ridiculous.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "yapsy".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "yapsy".
```

So, using Yapsy.. yapsy: <http://yapsy.sourceforge.net>

I turned the code inside

```
system-message
WARNING/2 in everything.txt, line 35978
Block quote ends without a blank line; unexpected unindent. backrefs:
```

out: almost all the code that was in that monster class was moved into plugins and the class turned into a smart plugin loader.

This has brought several advantages:

- Now you can extend Nikola for your own purposes. Just create a plugin folder in your site, and put the extra functionality there.
- Support for whole categories of things is now modular. Want to support a different markup? A different template system? Just write a plugin.
- Want to add completely new functions to Nikola? Like, say, a way to import data from another blog tool, or something like planet software? Well, just add a "command plugin".

What has changed for the user?

- You can delete `dodo.py` from your site.
- Use `nikola build` instead of `doit`
- Use `nikola serve` instead of `doit serve`

Some minor things still need doing in this area (notable, fixing the docs), but the main refactoring is there.

Short and sweet:

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "nikola".
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "nikola".
```

In Nikola.. nikola: <http://nikola.ralsina.com.ar>

my static site/blog generator,

system-message

WARNING/2 in `everything.txt`, line 36010
Block quote ends without a blank line; unexpected unindent. backrefs:

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "check it out".

the Tag pages used to just be a big list of titles linking to the posts in the tag. Now they can (optionally) look just like the blog does, but filtered. [Check it out...](http://lateral.netmanagers.com.ar/categories/python.html) check it out.: <http://lateral.netmanagers.com.ar/categories/python.html>

Minor tweaks still needed (like, mentioning the tag somewhere in the title), but the basic functionality (filtering, pagination, layout, etc) is complete.

To enable it in your site, just use master from git and enable `TAG_PAGES_ARE_INDEXES = True` in your `conf.py`.
English only!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "nikola".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "nikola".

I have not stopped working on Nikola.. nikola: <http://nikola.ralsina.com.ar>, my static site generator. Here are the plans:

- 1) Finish the theme installer (so you can get a theme from the site easily)
- 2) Implement a theme gallery on the site (same purpose)
- 3) Fix a couple of bugs
- 4) Update manual
- 5) Polish a few theme bits
- 6) Release version 3.x (new major number because it requires manual migration)

After that, I will push on projects Shoreham (hosted sites) and Smiljan (planet generator) and make them more public. Shoreham will become a real web app for those who don't want to have their own server. For free, hopefully!

Once I have that, I have no further feature ideas, really. So I need more people to start using it, and that means I have to start announcing it more.

So, stay tuned for version 3.x sometime next week.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "rst2pdf".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "rst2pdf".

Post-Nikola, I will do a `rst2pdf..rst2pdf`: <http://rst2pdf.googlecode.com>

release, and then will get back to work on a book.

I did some work today to get Nikola properly packaged. This involves some minor changes on the workflow for site authors. I am not 100% sure I have it right yet, so here is a short video showing how it works right now in the `packaging` branch I am doing.

The new thing is the `nikola init foldername` command, the rest is all old stuff. Basically, you stop having a full copy of Nikola for each site and everything is in a centralized location.

You can still do your own themes by putting them in `themes/themenameand` add new tasks, files, etc. The configuration `dodo.py` is unchanged except for the "magic bit" which is slightly different.

So, not really invasive, easy to migrate to, and enables much easier updates in the future, as long as we don't break any important stuff in a non-compatible way.

Here is the video:

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "nikola".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "nikola".

I have been *flooded* by awesome patches for `Nikola..nikola`: <http://nikola.ralsina.com.ar>

by different contributors, so there will be

system-message

WARNING/2 in `everything.txt`, line 36085
Block quote ends without a blank line; unexpected unindent. backrefs:

a new release very soon. In the meantime, see what happens if someone with design skills does a theme for it:

If I learned one important thing in college (and I like to think I do because otherwise I wasted a lot of time there) that important thing is how to measure things.

You may think that you don't need to go to college to learn that, and you are right, but the interesting bit, if I may sound like a social studies major for a few seconds, is how arbitrary measurements are. They are the one bit where all that "reality is a social construct" insanity is kinda true.

Consider the distance between two places. ¿How far is my house from my mother's?

Well, google says it's 447km away. But you already know that's not true! If I were to go by, say, helicopter, or unstoppable tank, I may be able to take a somewhat shorter path. Were I to use the Underminer's tunneling machine, I could take an even shorter path.

Figure 38: The Underminer follows the shorter, red line.

But all that is mostly changing paths, so you may still feel some confidence that you know how far my mom's house really is, except that we just may not be able to take a specific path, but the distance is a "real" thing.

But geographical, geometrical distance is only one way to measure. There are other metrics, and they may be more or less valid. For example, my mother's house is 6 hours away by bus.

It's also U\$S 60 away by bus, U\$S 30 by car. Unless I take my son with me, in which case it's U\$S 120 by bus, but still U\$S 30 by car.

And if I really really want to go there, it's an impulsive decision away, and if I don't really want to, there is a whole lot of convincing between here and there.

And if I were as poor as I once was, then maybe it's infinitely far away because there is no way to get there from here.

When someone says "the world is smaller now" that's not metaphor, that's maths.

When I grow up, I want a pink car. Girls like pink.

—Tato (my son, age 4 at the time)

There has been a lot of talk lately about codes of conduct in conferences. I don't have answers to much, but I do have a lot of questions in my head, and some things seem to come to my mind because of that, so I will do a little head dumping, and let's see if clarity appears.

So. The main thing seems to be that the proposed codes of conduct aim at making events inclusive, and more diverse, and welcoming to people who may have felt unwelcome in the past. That these groups involve *women* should be a call of attention. Women? Women are half the world, and apparently we have been excluding them, whether intentionally or not.

So, in principle, if adopting a code of conduct helps that, I am all for it. Same about gays, lesbians, transsexuals, etc. They are not 50% of the world, but they are about 10%

of it, so it's a very large amount of people, and adding them to our groups is another easy optimization.

However, it concerns me a bit that these codes of conduct contain language like this:

Harassment includes offensive verbal comments related to gender, sexual orientation, disability, physical appearance, body size, race, religion, sexual images in public spaces, deliberate intimidation, stalking, following, unauthorized or inappropriate photography or recording, sustained disruption of talks or other events, inappropriate physical contact, and unwelcome sexual attention.

Some are completely obvious, people who disrupt the speakers at conferences are to be thrown out, intimidation is abominable, stalking is creepy, etc. But "offensive verbal comments related to *everything*"?

And here, I have to make a small detour and talk about street signs. When I visited London, one thing that called my attention was the language in street signs. Here's an example:

Figure 39: In other countries it would say "No dogs", wouldn't it?

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "punishable by jail".
```

So, why do they have signs like this? It's probably because they are very polite people. Also, it may be because offensive signs are [punishable by jail...](http://t.co/K3wzfu2J) punishable by jail.: <http://t.co/K3wzfu2J>

I have a completely unfounded suspicion that the politeness came first, and then came the right not to be offended, which ended encoded into law, and now you just *have* to be polite, or else.

I am quite loudly atheist, yet I only mention it outside this blog if someone mentions *his* religion. So, for example, if someone says "god asks less and yet he forgives", I may say "that's because he doesn't exist", as a joke. I am now assuming that saying that in a conference with that kind of code of conduct is going to be considered offensive behaviour. And let's say I am ok with that, I *can* still go for a few hours without offending people. Have not tried it lately, but I am sure I can do it.

And I know that at a private event, like a conference, there is no inherent freedom of speech, because there is the right of admission, and I can just be kicked out without any explanation, and I am also fine with that, because I reserve the right to kick people out of my own home, too.

So, sure, let's keep religion out of it. It has no place in a technical conference one way or another, and in any case, I will wear my invisible pink unicorn shirt as an outward sign of my atheism (it looks just like a gray v-neck t-shirt).

And I am totally fine about not mocking or harassing people because of their gender or sexual preferences. I am old and provincial enough that when two men start kissing next to me, I feel awkward. Luckily, I am enough of an adult that I just think to myself, “dude, you are a provincial prude” and look the other way. After all, I have seen people take exception to me kissing my *wife* in public, so, live and let live, whatever. I like women, my wife likes men, so I can understand you liking either.

On the other hand, I understand that the mere existence and presence of some people can be offensive to others. I know people who would rather stand for 2 hours than sit next to a transsexual. Or would rather get off the bus instead of being there. And I am enough of an old, provincial prude that I understand them. So, offending is not the thing here, because if offending is the thing, then the mere presence of someone can offend others, and that’s the exact opposite of what we want. We want them to either not be offended, or be offended and get over it, or be offended and *not care*.

So, handing out invitations to threesomes to people in hallways is a bit too much (I never invite people to threesomes before the fourth date, it is *gauche*). Hitting on people in bars at night is probably not too bad, unless it’s a constant thing that ruins the night for someone (what do I know, I have never hit on someone or been hit on in a bar. Except by other men. Just my luck!) in which case I expect a group of nice people to form a protective ring around the poor person who is just too attractive? (again, what do I know, I have never been attractive).

The thing we want is politeness. We want to be nice to each other. We want everyone to be as nice as they possibly can to as many people as they can. Specially, we want everyone to be extremely nice to the people they like the least. Because with people you get along with, you can do crazy stuff you can’t do with others.

On the other hand, I suspect there is something else here I am missing. Because tolerance and respect is just not my thing. I am all for proselitizing and disrespect, for creative annoyance and pushing people outside their comfort zones. But I try not to do it personally, I try to throw things to the crowd and see what they do with them.

I mean, I have been photographed without my consent. I have even had my shirt scanned without asking permission (ok, I admit having a QR code in a shirt is sort of an implicit agreement), I have been called names, but I know that, in the words of a sci-fi writer, I live life in the low difficulty setting, because I am a rather healthy white heterosexual male born in middle class with a job, so again, I don’t quite know what it’s like to be insanely attractive, or gay, or insanely attractive to gays, or anything. I am not *harassable*. My face protects me. I know others don’t have such powerful defenses.

So, while that kind of language does fill me with trepidation, and makes me wonder what kind of community I have been living in, oblivious to all these things I read about lately, I will accept those codes and try to follow them. I have never intentionally broken them, even before they existed (I did once take an inappropriate picture, it was a joke, I only showed it to one person, and I deleted it, and I really am sorry and would not do it again, ok?)

So, I hope to see a lot of people I don’t know in the next free software events I attend. Hopefully I will not offend any of them in a bad way. I will not be too brash. I will try to be inclusive. I will try to be *nice*. But remember. If I am *very, very nice* to you, it may be because I can’t stand you. You’re welcome.

I have written a few days ago about some issues with SSL certificates.

This post will describe the strangest thing I have done or work purposes in the last year, year and a half.

We get reports that users of Ubuntu Precise Pangolin, in Turkey, are getting a SSL certificate error. I suspect it may be because there is a government firewall there (the “Halal Internet”) So, we try debugging it with them. And we get nowhere yet. So, I had the idea of trying to get a turkish IP address and debug from there.

I started looking for a VPS, couldnot find any, then Rodney Dawes found a Turkish VPN provider. Which is weird in that a million Turks are using VPNs to get *outof* the Turkish censoring firewall, but hey, I will do things for the users.

So, I get that, I make it work, and then I debug. No error. So, how do I know if I *am* behind the Turkish firewall? I tried to open a porn site, expecting it to fail. But it did not.

So, oficialmente I saw like 10 seconds of porn while working, for work purposes.

BTW: the correct way to know is to try to access <http://www.rollingstone.com>

Sorry again, spanish only, because it’s another video of me speaking in spanish (this time with a spaniard)

Si querés tener muchos usuarios de tu programa, es fundamental que el universo de “gente que puede usar el programa” sea lo más grande posible. El mejor camino para ello es hacer que el sistema operativo que usa/sufre tu potencial usuario esté soportado. Hacer eso no es increíblemente difícil, pero siempre hay alguna cosa rara. La conferencia tuvo lugar en la ciudad de Junín (Buenos Aires, Argentina) en Septiembre de 2011, y fué organizada por PyAr (Python Argentina).

Video:

Slides:

Finishing Day 2, here’s the update.

- Woke up 10 time because David Litvak doesn’t know how to turn off his alarm. Now hating “Good Day Sunshine” by The Beatles. Sorry Ringo!
- Learned about Celery/Flask/Juggernaut. Tomorrow will try to hack on the Nikola server idea using that. Hugo Ruscitti already did a lot of work!
- Helped Martín Gaitán do Nikola gallery improvements (looking much nicer now!)

system-message

- **WARNING/2** in `everything.txt`, line 3
[Duplicate explicit target name: “code listings for nikola”](#).

Implemented [code listings for Nikola](#).. code listings for nikola: <http://lateral.netmanagers.com.ar/weblog/posts/n>

- Had a huge spider walk up my leg.
- Had asado for dinner!

- Played with remote controlled cars.
- Saw a guy juggle while riding a monocycle
- Played in the ping pong tournament (record 1-1, out in the group phase)
- Fixed a bunch of bugs that popped while implementing stuff.

Tomorrow: will hack all day in the nikola hosting idea.
So, day 3 and next-to-last of PyCamp is done.

- Great day, sunny, not all that cold
- Empanadas at lunch, pizza for dinner, cake for tea. Feeling kinda spoiled today.
- Lots of hacking at Nikola-as-a-service (details below)
- PyAr meeting by a huge, somewhat scary bonfire earlier tonight

So: Nikola-as-a-service is an idea where you can keep your blog somewhere, and this service will get the data, and publish a nice site for you.

Here's the current workflow, which is just one of a dozen that can be implemented because this thing is quite simple:

1) The authenticator

Currently it has twitter authentication. You never need to create an account, just login with some service you already have accounts with. Anything with OAuth will work.

2) The data provider

Currently, github. Soon, Ubuntu One. Later, who knows. A data provider is something from where we can grab data, and that can notify us (automatically or by having the user click on a button) when we should get that data and rebuild the site.

3) The renderer

I am doing it with Nikola, of course :-)

4) The infrastructure

Jobs using Redis and Celery, server app using Flask, rendering using Nikola

5) How does it work?

You go to github, clone a barebones blog. Do your modifications. Go to nikola-as-a-service, and login via something. Then you give Nikola your github repo's URL, and you get a webhook URL. Go back to github, and configure the webhook.

From that moment on, every time you push to github, your blog is updated :-)

In the future: every time you save to Ubuntu One, your blog is updated. In the further future: Every time you X to Y, your blog is updated.

It's going to be cool :-)

Going to sleep, so here's a quick rundown of my 1st day at Pycamp:

- Woke up at 6AM, and after taking trains, buses and automobiles, arrived around noon. Place is nice, wheather is cold. Beds are bunks.
- Presented ideas, voted working slots
- Implemented a Nikola feature to get themes from <http://bootswatch.com>
- Implemented first draft of pipelines for post-processing generated files
- Martín Gaitán is working on improving the image galleries
- Hugo Ruscitti is doing some celery+flask magic to create a Nikola hosting service
- Had fun with lots of other things and talking with a lot of people.
- Played some table tennis (won 1, lost 1)
- Got a firefox t-shirt
- Got a Ninja-IDE mug that changes colour with temperature.

Now, exhausted, going to bed.

So you couldn't make it to PyCamp? This is the place. Sorry, I can't post the awesome, so it's just a picture :-)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "an isolated place".

This friday marks the beginning of the 2012 PyCamp. What's PyCamp? It's a lot of python programmers (this year, about 50) gathering in [an isolated place](#).. an isolated place: <http://goo.gl/maps/LsnD>

for a long weekend, with nothing

system-message

WARNING/2 in `everything.txt`, line 36483
Block quote ends without a blank line; unexpected unindent. backrefs:

to do except code. We will have our meals catered, there are no TVs, there is hardly anything within walking distance, and it's going to be very cold.

So, it's going to be **awesome**. It's a rare chance for me to spend a few days hacking at my own personal projects, uninterrupted by more important things like family, work, cooking, or socializing except between nerds.

Sure, there is the occasional monocycle riding, or juggling lesson, or shooting practice, but really, three or four solid days of hacking.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "nikola".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola”.

I intend to work on projects related to Nikola.. nikola: <http://nikola.ralsina.com.ar>
my static site generator, so if you are interested in that and going to pycamp, I want to talk with you.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “in the nikola-discuss group”.

And if you are interested but *not* going to pycamp, there is no reason not to join in a virtual sprint. We’ll have internet. There is IRC. I will have *time*. It’s a weekend! Please share any interesting ideas you have about static site generators [in the nikola-discuss group](https://groups.google.com/forum/#!forum/nikola-discuss)..
in the nikola-discuss group: <https://groups.google.com/forum/#!forum/nikola-discuss>
and we’ll see how much can get implemented or at least started.

This saturday I attended Pyday Luján. I don’t know if other places have such things as PyDays, so here’s a brief summary:

- 1) It is a day-long event
- 2) It usually takes place in a university
- 3) It has one or two tracks
- 4) It’s about python
- 5) It’s fun

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “universidad nacional de luján”.

This time, it took place at the [Universidad Nacional de Luján](http://www.unlu.edu.ar/).. [universidad nacional de luján](http://www.unlu.edu.ar/): <http://www.unlu.edu.ar/>

which is a biiiiit too far from my home:

Luckily Facundo Batista was going, and he’s a driver, not a pedestrian, and took me there.

I was to speak about “Doing one thing and doing it right”. These are the boring slides:

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “here”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “here”.

If you don't see them up there, go [here](#).. here: <http://speakerdeck.com/u/ralsina/p/hacer-una-cosa-pero-hacerla-bien>

I really, really wanted to record this. But I failed. So, combining that with my reluctance about repeating myself, it means this thing will only ever be seen by those who were there last saturday.

Great fun was had, an unwise number of choripanes was eaten, and lots of interesting conversations occurred.

Hopefully soonish there will be some pictures, too (although I did not take any). So, thanks to Yamila and the rest of the Luján organizers!

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "pyday luján".

I will be speaking at [PyDay Luján](#).. pyday luján: <http://www.pyday.com.ar/lujan2012> next saturday May 5th.

The title is "Doing one thing and doing it well" and it's *noteven* fully haped in my head yet, which is usually a **goodsign**.

So, come join me and a bunch of smarter people talking about programming for a whole day, for free. It will be fun!

Mentioning this blog gets you candy. Just saying.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "reddit".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "reddit".

There is a large thread in [reddit](#).. reddit: http://www.reddit.com/r/Python/comments/vwy6z/prefer_text_config_files about using Python as configuration file format. I want to be clear about that:

DON'T DO THAT, UNLESS YOU HAVE A VERY GOOD REASON.

If you need to ask if it's a good idea, then you don't have a good reason. If you are sure that you have a good reason, then **maybe** you have a good reason.

There are many reasons for it, but I will explore just two, then offer a suggestion.

Python is read-only, and configuring is not programming.

Sure, it's *easy* to use python as a config file. You just import the thing, and there you go, all the data is there. But now your configuration syntax is a general purpose language that can do things like pop up dialogs when you parse it.

Your config can now depend on the whole internet, the hardware, the weather, and interactive input. Powerful? Sure. Good idea? Maybe, sometimes. But your application is now not able to configure itself.

If your application wants to store any kind of setting, it won't be able to. So most interactive, desktop apps, just should not use python for this, ever.

But what about non-interactive tools? Well, using python means that other tools can't write to the config file, either, which makes the tool less powerful. The power to have tools use tools is one of the cornerstones of modern computing, and you just

cut your app off that ecosystem. Depending on what language the tool uses it may not even be able to *parse* your config file.

And what happens when someone is told “use this config fragment to achieve X”? Well, odds are, if the recipient has done anything that takes advantage of using python as a config format, then the fragment will not work. It would be like doing copy/paste from random code in github into your own program and expecting it to work.

So, you can’t write to it from the app, you can’t get configuration tips from the internet, you can’t use other tools to modify config files, and other tools have a hard time parsing your files.

Also, it means that to handle the general case of configuring your app, you need a programmer. That is almost certainly overkill. Very few apps need that kind of thing. If your app can only be configured by programmers, you may have *failed* at making a good app (exceptions exist).

And what’s the advice? Well, the advice is “don’t do that” and the corollary is “configure using data, not code”. use INIs, or XML, or YAML, or JSON, or plain text files, or whatever. But not code.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “nikola”.

PS: My latest project, Nikola.. nikola: <http://ralsina.com.ar>

uses python as

system-message

WARNING/2 in `everything.txt`, line 36631
Block quote ends without a blank line; unexpected unindent. backrefs:

a configuration language. I thought I had a good reason. I didn’t.

My team has been working on porting some PyQt stuff to Mac OSX, and we have run into several Qt bugs, sadly. Here are two, and the workarounds we found.

Native dialogs are broken.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “qfiledialog.getexistingdirectory”.

Using `QFileDialog.getExistingDirectory..` `qfiledialog.getexistingdirectory`: <http://qt-project.org/doc/qt-4.8/qfiledia>
we noticed the following symptoms:

- If you do nothing, the dialog went away on its own after about 20 seconds.
- After you used it once, it may pop up and disappear immediately. Or not.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "dontusenativedialog option."

Solution: use the **DontUseNativeDialog option...** dontusenativedialog option.: <http://qt-project.org/doc/qt-4.8/qfile>

Widgets in QTreeWidgetItem don't scroll.

When you use Widgets inside the items of a QTreeWidgetItem (which I know, is not a common case, but hey, it happens), the widgets don't scroll with the items.

Solution: use the `-graphicssystem raster` options. You can even inject them into `argv` if the platform is darwin.

If you have actual, paper books and want to catalog their info quickly, this book-data.py script may be handy:

```
\ :kn:'import' \ \ :nn:'sys' \
\ :kn:'import' \ \ :nn:'time' \
\ :kn:'import' \ \ :nn:'gdata.books.service' \
\ :kn:'import' \ \ :nn:'json' \

\ :k:'def' \ \ :nf:'get_book_info' \ \ :p:'(\ \ :n:'isbn' \ \ :p:'):' \
\ :k:'print' \ \ :s:'"Looking for ISBN:"' \ \ :p:', \ \ :n:'isbn' \
\ :n:'google_books' \ \ :o:'=' \ \ :n:'gdata' \ \ :o:'.' \ \ :n:'books' \
\ :n:'result' \ \ :o:'=' \ \ :n:'google_books' \ \ :o:'.' \ \ :n:'search' \
\ :n:'data' \ \ :o:'=' \ \ :p:'[\ \ :n:'x' \ \ :o:'.' \ \ :n:'to_dict' \
\ :k:'if' \ \ :ow:'not' \ \ :n:'data' \ \ :p:': \
\ :k:'print' \ \ :s:'"No results"' \
\ :k:'return' \
\ :n:'title' \ \ :o:'=' \ \ :n:'data' \ \ :p:'[\ \ :mi:'0' \ \ :p:'][\ \
\ :k:'with' \ \ :nb:'open' \ \ :p:'(\ \ :n:'title' \ \ :o:'+' \ \ :s:'.' \
\ :n:'f' \ \ :o:'.' \ \ :n:'write' \ \ :p:'(\ \ :n:'json' \ \ :o:'.' \
\ :k:'print' \ \ :s:'"Guardada info de ' \ \ :si:'%s' \ \ :s:'" en ' \ \

\ :k:'if' \ \ :n:'__name__' \ \ :o:'==' \ \ :s:'"__main__"' \ \ :p:': \
\ :k:'while' \ \ :bp:'True' \ \ :p:': \
\ :n:'isbn' \ \ :o:'=' \ \ :n:'sys' \ \ :o:'.' \ \ :n:'stdin' \ \ :o:
\ :k:'if' \ \ :n:'isbn' \ \ :p:': \
\ :n:'get_book_info' \ \ :p:'(\ \ :n:'isbn' \ \ :p:')' \
\ :n:'time' \ \ :o:'.' \ \ :n:'sleep' \ \ :p:'(\ \ :mi:'1' \ \ :p:')' \
```

What does it do? It reads ISBN numbers from standard input and saves the book's info in a title.json file for later processing and formatting.

system-message

WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "jsonwidget".

If you want to edit that information, you can just do it or you can try doing a little script using `jsonwidget..` jsonwidget: <http://jsonwidget.org>

like this:

```
python -c 'import jsonwidget; jsonwidget.run_editor("abook.json", schemafile
```

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "this file."

Where `abook.json` is a file generated by the previous script and `gbooks.schema` is [this file](http://pastebin.ubuntu.com/1089975/)... this file.: <http://pastebin.ubuntu.com/1089975/>

Oh, and if your books have barcodes, you can just do:

```
zbarcam --raw | python bookdata.py
```

Show your computer your books and let it do the rest :-)

PS: I would love if someone gathered all this and made a nice personal book cataloging thing.

Just released `rst2pdf` version 0.92. `Rst2pdf` is a tool to convert restructured text to PDF without requiring multi-hundred-megabytes of software. It supports styling, multiple page layouts, font embedding, vector images, and much more.

Also, now it has a real website: <http://rst2pdf.ralsina.com.ar>

Try it out and let me know of the obvious, horrible bug I forgot to fix (happens in most `rst2pdf` releases ;-)

Sorry, spanish only!

El problema es que el español es una lengua viva, y como todo aquel que ha vivido tanto, se esta poniendo senil.

—Perrito

En Twitter me acaba de decir el amigo Perrito esa frase redonda, cerradita, impecable, cual culo de muñeca Yoli-Bell. Y tiene razón. Por eso yo no escribo en español, ni en su cuñado sectario el castellano, yo escribo en argentino.

Y eso no quiere decir escribir "que hacé cabeza, a cuanto la birra" ni "el alazán se retoba" sino simplemente escribir mas o menos como sueno en mi cabeza, lo que no es tan fácil como parece, pero si no lo pienso mucho me sale. Si lo pienso, sale otra cosa, que será mejor o será peor, pero es otra cosa.

Entonces, venía diciendo que yo escribo en argentino, porque el español está viejo. Porque el concepto mismo de "así se escribe el español", "así se habla en español" es viejo. Es la idea de que hay una mesa en España adonde se sienta una cantidad de gente (¡alguno que me cae bien, incluso!) que decide que de un día para otro decir `cederrón` está bien y antes estaba mal. `Cederrón`, oíme, por el codo derecho de Kali (el del medio), es una señal de que el concepto está mal.

Que haya una mesa adonde te dicen que ahora al que nace en Austria hay que decirle austriaco sin acento, es un insulto, es como si en tu oficina un día te ponen un cartel que dice que el edulcorante se abre del lado de la "E" de Equal Sweet, y que si lo abris del lado de la "t" está mal y sos un burro.

Es legislar la nada, es decretar lo inútil. Es la confesión de parte que me releva de probar que son inútiles de toda inutilidad. Es decir "¡Tenemos mucho tiempo al pedo, hagamos que sólo no lleve nunca acento!".

Repudien eso. Repudien la autoridad autoasignada de vetustos catedráticos de la metrópolis en quiebra, y hablen en argentino. Y si les dicen que hablan mal, mándenlos a la concha del pato, que está bien dicho en perfecto argentino. Y si no sos argentino, bueno, no importa, no te cobramos los derechos.

While walking along the river before dawn I laid down on a bench and looked up, and saw the tree, clear and green against the orange clouds in the night sky, and thought, hey, that looks cool, and tried to take a picture.

The screen in my camera stayed obstinately black. I changed settings, moved ISOs, touched on different places trying to convince it to focus and set aperture for the darkest or the lightest areas of what I knew to be there.

And it remained black. And suddenly, I had a dissenting opinion, that there was not a clear green tree there, and that the sky was not full of orange clouds, but that it was all black, starless and empty, empty of tree, of cloud.

I placed my hand above the camera, hoping to catch a glimmer of it, and still, the display was a square of darkness separating my fingers from my arm, as empty as before, mocking me featureless.

Why was it so black, if I could see clearly. If there were lampposts giving light, and I could see clearly, and there *was* a tree. I knew the camera worked. What was I doing, by the river, at 4AM, on a tuesday, laying on a bench, looking up, with a camera?

You expect your senses to work. You expect to perceive what is there, and not perceive what is not. You expect to see reality, to not see irreality, to listen to things, to not listen to unthings, to touch truth, to smell shit.

What would happen if you had two sets of senses, two visions, and they disagreed, and you were not sure which one to trust, which one is *right*, which one is *true*? What would happen if the camera was right and my eyes were wrong, and I was actually not seeing, but imagining, and the truth was empty, and the tree was not there, and the sky was black.

Then I enabled flash, and the ugly picture convinced me to, someday, get a better camera, and never forget to take my gastritis medicine when going for trips on isolated locations.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "sherlock".

Often movies or TV series come in pairs. These days I watched two series that are obviously related, *Sherlock*.. *sherlock*: <http://www.imdb.com/title/tt1475582/>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "elementary".

and *Elementary*.. *elementary*: <http://www.imdb.com/title/tt2191671/>

and there is even a movie series by guy Ritchie (which I actually like!) but let's talk TV.

I am not going to be original in saying *Sherlock* is the superior show. But why is it?

Well, I think it mostly comes to one being done by people who have read the books, and the other by people who heard about them.

For example, that evil word "Elementary". It's not in the books. It's in the movies, though. So, if you focus on second-hand sources, it makes sense to use it, but if you care about the original materials it makes sense to carefully avoid it.

There's also the problem of Elementary's Holmes looking like a hobo. Holmes was fastidiously neat. He was a slob about his lodgings, but he always kept himself clean and well dressed.

Or let's consider addiction. Yes, in the books Holmes shoots cocaine and does morphine. Thing is, those things were not even illegal at the time. Cocaine was a *cough medicine*. So, transposing that into nicotine addiction makes sense, specially since Holmes was also a very heavy smoker even for victorian standards. Turning it into a drug habit that forces Holmes into rehab (rehab!) doesn't. Also, "this is a three patch problem"? Have to chucle at that, dude.

The Watsons also are quite different. I quite like Lucy Liu's deadpan delivery of everything, but Watson is not supposed to be a damaged person that nurtures. He's a thrill seeker, a badass character that is only mild-mannered when compared to his company. Again, Sherlock walks closer to the books there, while Elementary tries to shoehorn some weird personal-growth subplot.

Yes, Watson is the one that brings out the human side of Holmes, but he does that not by being all soft and cuddly, he does it by being a hard headed bastard who stands up to him. He's a true friend, and friends don't take shit from friends, at least not without giving shit back. In Sherlock he does that, and clearly Holmes respects him. In Elementary, Watson is tolerated, and treated like a pet.

Winks. Both series try to make reference, more or less oblique, to the source material. Again, it feels like Elementary is working from second hand references. If I could find you the "Holmes intentionally avoids learning things of no immediate relevance" bits in both, the Elementary one was a groaning exposition, including physical demonstration of how water displaces oil. In Sherlock? Well, it's an argument. Incredulity on one side, quiriness on the other, funny dialog.

Because that's what Elementary isn't. it's just not fun. And a Holmes that's not fun, is a broken Holmes.

At Canonical's Online Services we can do cool stuff on fridays. We do cool stuff all week, actually, but on fridays we can do crazier cool stuff.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "nikola".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "nikola".

So, today, I ripped off a great service offered by <http://calepin.co> and implemented a prototype blog-through-Ubuntu-One web application. Of course, it's powered by Nikola.. nikola: <http://nikola.ralsina.com.ar>

The code is absolute nonsense, and it needs to be looked at by someone who understands Django, OAuth, OpenID, and programming in general better than I do, but hey, it does work (for a very loose definition of "work").

It's called Shoreham and no, you can't have it yet.

As a teaser, here's a video. With a pony.

In the near future I will do a better post about this explaining the code, etc.

I have been watching Galactica (the newer one) on and off for a few months. And there is one small thing that drives me nuts every time I see it. It throws me off the story, and completely breaks the world-building that's going on.

Octogonal paper.

Galactica paperwork/BSGPaperwork001.jpg Galactica paperwork/BSGPaperwork001.jpg

Figure 40: This. Makes. No. Sense. Nice font, though.

Why? Because paper is square for functional and manufacturing reasons. You take a spool of paper, you cut it, you end up with rectangular pieces.

Are you manufacturing linen paper? Then you need to build frames to do it, and making octogonal frames is much harder.

To make octogonal paper sheets you need to cut every one of them from square sheets. That is stupid.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "octogonal *tractor paper*.".

Oh, it gets worse. In the BSG universe, they have [octogonal *tractor paper*... octogonal *tractor paper*.: http://archive.propworx.com/1002/1424](http://archive.propworx.com/1002/1424)

What sort of idiot came up with that? How can that even *work*!

So, a fun series but every time I see a piece of paper I want to scream.

I maintain a couple of small "planet" sites. If you are not familiar with planets, they are sites that aggregate RSS/Atom feeds for a group of people related somehow. It makes for a nice, single, thematic feed.

Recently, when changing them from one server to another, everything broke. Old posts were new, feeds that had not been updated in 2 years were always with all its posts on top... a disaster.

I could have gone to the old server, and started debugging why rawdog was doing that, or switch to planet, or look for other software, or use an online aggregator.

Instead, I started thinking... I had written a few RSS aggregators in the past... Feedparser is again under active development... rawdog and planet seem to be pretty much abandoned... how *hard* could it be to implement the minimal planet software?

Well, not all that hard, that's how hard it was. Like it took me 4 hours, and was not even difficult.

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "i already have one".

One reason why this was easier than what planet and rawdog achieved is that I am not doing a static site generator, because [I already have one.. i already have one: http://nikola.ralsina.coma.r](http://nikola.ralsina.coma.r)

so all I need this program (I called it Smiljan) to do is:

- Parse a list of feeds and store it in a database if needed.
- Download those feeds (respecting etag and modified-since).
- Parse those feeds looking for entries (feedparser does that).
- Load those entries (or rather, a tiny subset of their data) in the database.
- Use the entries to generate a set of files to feed Nikola
- Use nikola to generate and deploy the site.

So, here is the final result: <http://planeta.python.org.ar> which still needs theming and a lot of other stuff, but works.

I implemented Smiljan as 3 doit tasks, which makes it very easy to integrate with Nikola (if you know Nikola: add “from smiljan import *” in your dodo.py and a feeds file with the feed list in rawdog format) and voilá, running this updates the planet:

```
system-message
```

```
WARNING/2 in everything.txt, line 36987  
Inline emphasis start-string without end-string.
```

```
doit load_feeds update_feeds generate_posts render_site deploy
```

Here is the code for smiljan.py, currently at the “gross hack that kinda works” stage.
Enjoy!

```
\ :c: '# -*- coding: utf-8 -*-'\  
\ :kn: 'import '\ \ :nn: 'codecs'\  
\ :kn: 'import '\ \ :nn: 'datetime'\  
\ :kn: 'import '\ \ :nn: 'glob'\  
\ :kn: 'import '\ \ :nn: 'os'\  
\ :kn: 'import '\ \ :nn: 'sys'\  
  
\ :kn: 'from '\ \ :nn: 'doit.tools'\ \ :kn: 'import '\ \ :n: 'timeout'\  
\ :kn: 'import '\ \ :nn: 'feedparser'\  
\ :kn: 'import '\ \ :nn: 'peewee'\  
  
\ :k: 'class'\ \ :nc: 'Feed'\ \ :p: '('\ \ :n: 'peewee'\ \ :o: '.'\ \ :n: 'Model'  
    \ :n: 'name'\ \ :o: '='\ \ :n: 'peewee'\ \ :o: '.'\ \ :n: 'CharField'\ \ :  
    \ :n: 'url'\ \ :o: '='\ \ :n: 'peewee'\ \ :o: '.'\ \ :n: 'CharField'\ \ :p  
    \ :n: 'last_status'\ \ :o: '='\ \ :n: 'peewee'\ \ :o: '.'\ \ :n: 'CharField'  
    \ :n: 'etag'\ \ :o: '='\ \ :n: 'peewee'\ \ :o: '.'\ \ :n: 'CharField'\ \ :  
    \ :n: 'last_modified'\ \ :o: '='\ \ :n: 'peewee'\ \ :o: '.'\ \ :n: 'DateTi  
  
\ :k: 'class'\ \ :nc: 'Entry'\ \ :p: '('\ \ :n: 'peewee'\ \ :o: '.'\ \ :n: 'Mode  
    \ :n: 'date'\ \ :o: '='\ \ :n: 'peewee'\ \ :o: '.'\ \ :n: 'DateTimeField'\  
    \ :n: 'feed'\ \ :o: '='\ \ :n: 'peewee'\ \ :o: '.'\ \ :n: 'ForeignKeyField'  
    \ :n: 'content'\ \ :o: '='\ \ :n: 'peewee'\ \ :o: '.'\ \ :n: 'TextField'\
```

```

    \ :n:'link'\ \ :o:'='\ \ :n:'peewee'\ \ :o:'.'\ \ :n:'CharField'\ \ :
    \ :n:'title'\ \ :o:'='\ \ :n:'peewee'\ \ :o:'.'\ \ :n:'CharField'\ \ :
    \ :n:'guid'\ \ :o:'='\ \ :n:'peewee'\ \ :o:'.'\ \ :n:'CharField'\ \ :

\ :n:'Feed'\ \ :o:'.'\ \ :n:'create_table'\ \ :p:'('\ \ :n:'fail_silently'
\ :n:'Entry'\ \ :o:'.'\ \ :n:'create_table'\ \ :p:'('\ \ :n:'fail_silently'

\ :k:'def'\ \ :nf:'task_load_feeds'\ \ :p:'():'\
  \ :n:'feeds'\ \ :o:'='\ \ :p:'[]'\
  \ :n:'feed'\ \ :o:'='\ \ :n:'name'\ \ :o:'='\ \ :bp:'None'\
  \ :k:'for'\ \ :n:'line'\ \ :ow:'in'\ \ :nb:'open'\ \ :p:'('\ \ :s:''\
    \ :n:'line'\ \ :o:'='\ \ :n:'line'\ \ :o:'.'\ \ :n:'strip'\ \ :p:''\
    \ :k:'if'\ \ :n:'line'\ \ :o:'.'\ \ :n:'startswith'\ \ :p:'('\ \ :
      \ :n:'feed'\ \ :o:'='\ \ :n:'line'\ \ :o:'.'\ \ :n:'split'\ \ :
    \ :k:'if'\ \ :n:'line'\ \ :o:'.'\ \ :n:'startswith'\ \ :p:'('\ \ :
      \ :n:'name'\ \ :o:'='\ \ :s:' '\ \ :o:'.'\ \ :n:'join'\ \ :
    \ :k:'if'\ \ :n:'feed'\ \ :ow:'and'\ \ :n:'name'\ \ :p:':'\
      \ :n:'feeds'\ \ :o:'.'\ \ :n:'append'\ \ :p:'([\ \ :n:'feed'\
      \ :n:'feed'\ \ :o:'='\ \ :n:'name'\ \ :o:'='\ \ :bp:'None'\

  \ :k:'def'\ \ :nf:'add_feed'\ \ :p:'('\ \ :n:'name'\ \ :p:','\ \ :n:
    \ :n:'f'\ \ :o:'='\ \ :n:'Feed'\ \ :o:'.'\ \ :n:'create'\ \ :p:''\
    \ :n:'name'\ \ :o:'='\ \ :n:'name'\ \ :p:','\
    \ :n:'url'\ \ :o:'='\ \ :n:'url'\ \ :p:','\
    \ :n:'etag'\ \ :o:'='\ \ :s:'caca'\ \ :p:','\
    \ :n:'last_modified'\ \ :o:'='\ \ :n:'datetime'\ \ :o:'.'\ \ :n:
    \ :p:')'\
    \ :n:'f'\ \ :o:'.'\ \ :n:'save'\ \ :p:'()'\

  \ :k:'def'\ \ :nf:'update_feed_url'\ \ :p:'('\ \ :n:'feed'\ \ :p:','\
    \ :n:'feed'\ \ :o:'.'\ \ :n:'url'\ \ :o:'='\ \ :n:'url'\
    \ :n:'feed'\ \ :o:'.'\ \ :n:'save'\ \ :p:'()'\

  \ :k:'for'\ \ :n:'feed'\ \ :p:','\ \ :n:'name'\ \ :ow:'in'\ \ :n:'f'
    \ :n:'f'\ \ :o:'='\ \ :n:'Feed'\ \ :o:'.'\ \ :n:'select'\ \ :p:''\
    \ :k:'if'\ \ :ow:'not'\ \ :nb:'list'\ \ :p:'('\ \ :n:'f'\ \ :p:')'\
      \ :k:'yield'\ \ :p:'{'\
        \ :s:'name'\ \ :p:':'\ \ :n:'name'\ \ :p:','\
        \ :s:'actions'\ \ :p:':'\ \ :p:'('\ \ :n:'add_feed'\ \ :
        \ :s:'file_dep'\ \ :p:':'\ \ :p:'[\ \ :s:'feeds'\ \ :
      \ :p:'}'\
      \ :k:'elif'\ \ :nb:'list'\ \ :p:'('\ \ :n:'f'\ \ :p:')[\ \ :mi:'C
        \ :k:'yield'\ \ :p:'{'\
          \ :s:'name'\ \ :p:':'\ \ :s:'updating'\ \ :o:'+'\ \ :
          \ :s:'actions'\ \ :p:':'\ \ :p:'('\ \ :n:'update_feed_u
        \ :p:'}'\

\ :k:'def'\ \ :nf:'task_update_feeds'\ \ :p:'():'\
  \ :k:'def'\ \ :nf:'update_feed'\ \ :p:'('\ \ :n:'feed'\ \ :p:'):\

```

```

\ :n:'modified'\ \ :o:'='\ \ :n:'feed'\ \ :o:'. '\ \ :n:'last_modifi
\ :n:'etag'\ \ :o:'='\ \ :n:'feed'\ \ :o:'. '\ \ :n:'etag'\
\ :n:'parsed'\ \ :o:'='\ \ :n:'feedparser'\ \ :o:'. '\ \ :n:'parse
\ :n:'etag'\ \ :o:'='\ \ :n:'etag'\ \ :p:', '\
\ :n:'modified'\ \ :o:'='\ \ :n:'modified'\
\ :p:') '\
\ :k:'try'\ \ :p:': '\
\ :n:'feed'\ \ :o:'. '\ \ :n:'last_status'\ \ :o:'='\ \ :nb:'s
\ :k:'except'\ \ :p:': '\ \ :c:'# Probably a timeout'\
\ :c:'# TODO: log failure'\
\ :k:'return'\
\ :k:'if'\ \ :n:'parsed'\ \ :o:'. '\ \ :n:'feed'\ \ :o:'. '\ \ :n:'g
\ :k:'print'\ \ :n:'parsed'\ \ :o:'. '\ \ :n:'feed'\ \ :o:'. '\
\ :k:'else'\ \ :p:': '\
\ :k:'print'\ \ :n:'feed'\ \ :o:'. '\ \ :n:'url'\
\ :n:'feed'\ \ :o:'. '\ \ :n:'etag'\ \ :o:'='\ \ :n:'parsed'\ \ :o:
\ :n:'modified'\ \ :o:'='\ \ :nb:'tuple'\ \ :p:'(\ \ :n:'parsed'
\ :k:'print'\ \ :s:'"=====>"\ \ :p:', '\ \ :n:'modified'\
\ :n:'modified'\ \ :o:'='\ \ :n:'datetime'\ \ :o:'. '\ \ :n:'datet
\ :n:'feed'\ \ :o:'. '\ \ :n:'last_modified'\ \ :o:'='\ \ :n:'modi
\ :n:'feed'\ \ :o:'. '\ \ :n:'save'\ \ :p:'() '\
\ :c:'# No point in adding items from missinfg feeds'\
\ :k:'if'\ \ :n:'parsed'\ \ :o:'. '\ \ :n:'status'\ \ :o:'>'\ \ :
\ :c:'# TODO log failure'\
\ :k:'return'\
\ :k:'for'\ \ :n:'entry_data'\ \ :ow:'in'\ \ :n:'parsed'\ \ :o:
\ :k:'print'\ \ :s:'"====="
\ :n:'date'\ \ :o:'='\ \ :n:'entry_data'\ \ :o:'. '\ \ :n:'get
\ :k:'if'\ \ :n:'date'\ \ :ow:'is'\ \ :bp:'None'\ \ :p:': '\
\ :n:'date'\ \ :o:'='\ \ :n:'entry_data'\ \ :o:'. '\ \ :n:
\ :k:'if'\ \ :n:'date'\ \ :ow:'is'\ \ :bp:'None'\ \ :p:': '\
\ :k:'print'\ \ :s:'"Can't parse date from:"\
\ :k:'print'\ \ :n:'entry_data'\
\ :k:'return'\ \ :bp:'False'\
\ :n:'date'\ \ :o:'='\ \ :n:'datetime'\ \ :o:'. '\ \ :n:'datet
\ :n:'title'\ \ :o:'='\ \ :s:'"\ \ :si:'%s'\ \ :s:': '\ \ :s:
\ :n:'content'\ \ :o:'='\ \ :n:'entry_data'\ \ :o:'. '\ \ :n:
\ :n:'entry_data'\ \ :o:'. '\ \ :n:'get'\ \ :p:'(\ \ :s:
\ :n:'guid'\ \ :o:'='\ \ :n:'entry_data'\ \ :o:'. '\ \ :n:'get
\ :n:'link'\ \ :o:'='\ \ :n:'entry_data'\ \ :o:'. '\ \ :n:'lin
\ :k:'print'\ \ :nb:'repr'\ \ :p:'([\ \ :n:'date'\ \ :p:', '\
\ :n:'entry'\ \ :o:'='\ \ :n:'Entry'\ \ :o:'. '\ \ :n:'get_or_
\ :n:'date'\ \ :o:'='\ \ :n:'date'\ \ :p:', '\
\ :n:'title'\ \ :o:'='\ \ :n:'title'\ \ :p:', '\
\ :n:'content'\ \ :o:'='\ \ :n:'content'\ \ :p:', '\
\ :n:'guid'\ \ :o:'='\ \ :n:'guid'\ \ :p:', '\
\ :n:'feed'\ \ :o:'='\ \ :n:'feed'\ \ :p:', '\
\ :n:'link'\ \ :o:'='\ \ :n:'link'\ \ :p:', '\
\ :p:') '\
\ :n:'entry'\ \ :o:'. '\ \ :n:'save'\ \ :p:'() '\

```

```

    \ :k:'for' \ \ :n:'feed' \ \ :ow:'in' \ \ :n:'Feed' \ \ :o:'. ' \ \ :n:'se
      \ :k:'yield' \ \ :p:'{' \ \
        \ :s:''name' \ \ \ :p:': ' \ \ \ :n:'feed' \ \ \ :o:'. ' \ \ \ :n:'name' \ \
        \ :s:''actions' \ \ \ :p:': ' \ \ \ :p:'[( \ \ \ :n:'update_feed' \ \ \ :
        \ :s:''uptodate' \ \ \ :p:': ' \ \ \ :p:'[ \ \ \ :n:'timeout' \ \ \ :p:':
        \ :p:']}' \ \

\ :k:'def' \ \ \ :nf:'task_generate_posts' \ \ \ :p:'():' \ \

  \ :k:'def' \ \ \ :nf:'generate_post' \ \ \ :p:'( \ \ \ :n:'entry' \ \ \ :p:'): ' \ \
    \ :n:'meta_path' \ \ \ :o:'=' \ \ \ :n:'os' \ \ \ :o:'. ' \ \ \ :n:'path' \ \ \ :
    \ :n:'post_path' \ \ \ :o:'=' \ \ \ :n:'os' \ \ \ :o:'. ' \ \ \ :n:'path' \ \ \ :
    \ :k:'with' \ \ \ :n:'codex' \ \ \ :o:'. ' \ \ \ :n:'open' \ \ \ :p:'( \ \ \ :n:
      \ :n:'fd' \ \ \ :o:'. ' \ \ \ :n:'write' \ \ \ :p:'( \ \ \ :s:'u' \ \ \ :si:
      \ :n:'fd' \ \ \ :o:'. ' \ \ \ :n:'write' \ \ \ :p:'( \ \ \ :s:'u' \ \ \ :si:
      \ :n:'fd' \ \ \ :o:'. ' \ \ \ :n:'write' \ \ \ :p:'( \ \ \ :s:'u' \ \ \ :si:
      \ :n:'fd' \ \ \ :o:'. ' \ \ \ :n:'write' \ \ \ :p:'( \ \ \ :s:'u' \ \ \ :se:
      \ :n:'fd' \ \ \ :o:'. ' \ \ \ :n:'write' \ \ \ :p:'( \ \ \ :s:'u' \ \ \ :si:
    \ :k:'with' \ \ \ :n:'codex' \ \ \ :o:'. ' \ \ \ :n:'open' \ \ \ :p:'( \ \ \ :n:
      \ :n:'fd' \ \ \ :o:'. ' \ \ \ :n:'write' \ \ \ :p:'( \ \ \ :s:'u'.. raw:: h
      \ :n:'content' \ \ \ :o:'=' \ \ \ :n:'entry' \ \ \ :o:'. ' \ \ \ :n:'conte
      \ :k:'if' \ \ \ :ow:'not' \ \ \ :n:'content' \ \ \ :p:': ' \ \
        \ :n:'content' \ \ \ :o:'=' \ \ \ :s:'u'Sin contenido' \ \
      \ :k:'for' \ \ \ :n:'line' \ \ \ :ow:'in' \ \ \ :n:'content' \ \ \ :o:'. '
        \ :n:'fd' \ \ \ :o:'. ' \ \ \ :n:'write' \ \ \ :p:'( \ \ \ :s:'u' \ \ \ :
  \ :k:'for' \ \ \ :n:'entry' \ \ \ :ow:'in' \ \ \ :n:'Entry' \ \ \ :o:'. ' \ \ \ :n:
    \ :k:'yield' \ \ \ :p:'{' \ \
      \ :s:''name' \ \ \ :p:': ' \ \ \ :n:'entry' \ \ \ :o:'. ' \ \ \ :n:'id' \ \ \
      \ :s:''actions' \ \ \ :p:': ' \ \ \ :p:'[( \ \ \ :n:'generate_post' \ \ \
      \ :p:']}' \ \

```

We all hear all the time that less is more. That simple is better, that complex is worse, that options are evil, that defaults are what matter.

And yes, that is about 90% true. Except when it is false, which I know because I bought a coat a few weeks ago.

This is a rather nice coat, and if you saw it without much care you would miss one of its best features: it has two pockets on each side.

Let's think about why we want pockets in the sides of coats:

- 1) To put our hands when it's cold. Since this is a cold weather coat, that's important. In moderate climates like this one, gloves are more trouble than they are worth, and just sticking hands in pockets is enough.
- 2) To put stuff that is not hands in them: keys, phones, money, candy, etc.

For the first use case, we want the pockets to be shallow angled, so that the hand goes in naturally, almost horizontally. Also, we want the access to be unobstructed, so no zippers, which also scratch the wrists.

For the second use case, we want things not to fall off. So we want either a vertical pocket (perhaps with a flap) or a zipper. Zippers suck because you can forget to zip them, and things fall off. Vertical pockets are awful to put your hands in.

So, my jacket has two pockets on each side, one with a zipper, one without. One for hands, one for things. Since it's a thick coat you don't see it unless you know what you are looking for, and it's trivial to use: everything goes in the zipped one, except my hand. I can even check the contents of the zipped pocket without getting my hands out of *their* pockets.

This is one case where more is more, complex is better, options are awesome, and defaults don't matter. Now, if you find a place in software where that's the case, that's an opportunity.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "ubucon la".

I will be speaking at **Ubucon LA**.. ubucon la: <http://ubuconla.org/>
this next saturday

system-message

WARNING/2 in `everything.txt`, line 37199
Block quote ends without a blank line; unexpected unindent. backrefs:

June 2nd. The title is "Developing apps for Ubuntu One" and I expect it will be interesting, awesome, entertaining and full of *amazing* special effects.

Either that, or a regular talk. Hope to see lots of Ubuntu Users there!

Today is the 5th birthday of the most awesome kid I know. So, here he is, on a rollercoaster, a few days ago.

As he would say: it has been GREAT! and INCREDIBLE!

Happy birthday Tatito.

Tato had today his first day at football school. The other day he was crying because all his friends went there and he didn't, so, what the heck, he goes now too.

He doesn't have too much of a clue yet, but he tries hard ;-)

Before I gave up on becoming an educated man, I studied math. And to this day it pisses me that noone has figured out how to make math interesting to the math-averse. Here's a small attempt.

Let's consider the following scenario, based completely on things I know, not things I *do* ;-)

Suppose that in a city called San Isidro, there is a house. Houses in San Isidro pay a municipal tax, in exchange for the service of garbage collection, street sweeping, tree trimming, and street lighting.

It's a very small tax, but let's say it's \$100 a month because it's a nice, easy to handle number.

Also, San Isidro is in a country called Argentina. In that country there are several laws that affect the home owners:

- 1) You can't sell a house if you owe any taxes.
- 2) The owner has a 1% chance of wanting to sell the house each month.
- 3) Debts expire after 5 years.
- 4) If you are sued and you lose, you pay the other guy's lawyer fees.
- 5) Lawyer fees are capped to 25% of the money being disputed.
- 6) Lawyers are reluctant to help you sue someone if they get very little money (defined as less than \$2000)
- 7) If sued by the city for owed taxes, the owner always loses.
- 8) Unpaid taxes accrue 2% compound interest monthly. So, if you don't pay your \$100, you will owe \$102, then \$104.04, \$106.0128 etc.

With all those elements in place, let's play a game called "Tax Golf"!

The game is played by an indetermined number of players called owners and one special player called city.

The game is played to 100 "months" or until all property has been sold.

The goal of the game, for the owners, is to pay as little money as they can. The score is calculated like this: amount of money you paid divided by the time you owned the house.

The owner with the lowest score is the winner.

The goal of the game for the city, is to get as much money as he can. He's not competing against the players, for him it's a sort of solitaire where he competes against his past performance.

This, my friends is math. Math is a tool that helps you (among other things) do the right thing in this sort of complicated, arbitrary, real life scenario.

So, what's a good strategy for a owner, and for the city?

In a second post next wednesday, I will give some answers to those questions, using python.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "fede".

Everyone has that box, full of mystery USB cables, broken headphones, a mechanical mouse, printer cable and more USB cables. EVERYONE.

—Fede.. fede: <https://twitter.com/#!/fedee/status/203136297043378176>

I answered "I threw it away last friday". But that was a lie. I actually put everything in a bag, and had not thrown it away yet.

So here it is:

And now it's really gone. Including:

- Mini CDs with copies of the USB mass storage driver for windows (5)
- Parallel printer cable (2)
- Business cards of people I don't remember (12)
- Keys to a house that's not mine (1)
- Cigarette holder (1)
- Paper to do cigarettes (1 pack, "El Ombú" brand) (No, I don't smoke)
- Hard drives (3, total a bit below 4GB)
- CD reader (not writer) (1)
- RS-232 Serial cables (2)
- Mechanical keyboards (1, sticky)
- Artificial sweetener (4 packages)
- Mechanical mice (2)
- Leather case for Palm Pilot (1)
- Battery for a notebook I don't have anymore (1)
- AA batteries (3)
- IDE/PATA cables (2)
- Unrecognizable rubberstamp (1)
- Blockbuster credential (1)

And there was one thing I did not throw away. That's for some other day.

Because I am going to be traveling for three weeks, I was thinking today about packing. Last year, I spent five weeks on the road, and I packed a cabin-size suitcase, plus a messenger bag. Didn't feel like I had forgotten anything, so I guess I didn't.

That was spring and this will be late fall, so I will add a sweater and a better jacket.

So, it seems the amount of things I need to take with me on a trip doesn't correlate with the duration of the trip beyond a certain point. It seems that all the material assets I need to live fit in a smallish suitcase (plus a bag). I moved to Buenos Aires twelve years ago with a suitcase, a TV and a cat.

So, ignoring details such as family and sentimental value (which, for me, material things have none), suppose everything I own burned down in a fire. How much money would I need to miss nothing I lost?

Assuming I am moved into a bare apartment with the usual things in it (no furniture, but has a stove and a fridge), here's my list:

Clothes 7 t-shirts, 3 pants, 2 shorter pants, 2 pairs of sneakers, 2 sweaters, a coat, misc. About \$400 or so?

Computer I am considering upgrading the one I have for a \$800 one, so surely no more than that.

Phone I have a \$90 phone, but let's replace my kindle too, so a nicer one: \$250

Furniture A table, a few chairs, a mattress (I had to replace my bed recently, mattress in the floor for a week was surprisingly nice!). Perhaps \$500?

Random Cutlery, dishes, a pot, a frying pan, towels, bedsheets, coffee mug, toiletry items, blanket, extension cords, etc. \$1000?

Nothing else comes to mind, so, it seems I need a bit over \$2000, let's say \$2500 to be on the safe side. It's interesting to imagine I could move to, say, Costa Rica, by just having a job there, \$2500 in my pocket and no baggage.

This site has been lateral.pycs.net, lateral.fibertel.com.ar, lateral.blogspot.com, and has been, for several years, lateral.netmanagers.com.ar.

Well, I am slowly going to deprecate that URL, and the new URL will be <http://ralsina.com.ar> where you already can find it since a few months ago.

It will involve some work moving comments around and such, but nothing much should change, all old links should remain valid, and all comments should stay attached to the right post.

I will contact the various planets that aggregate it, but since the feed will remain constant thanks to feedburner, I expect not to lose anyone in transition.

I spent a long weeked at PyCamp. It was awesome. How could spending 4 days with 50 geeks hacking things and programming python not be awesome. But I was alone, because really, my wife (a lawyer) and son (too young to code yet) would have been bored to tears.

And I know something similar happened to a bunch of others there. Let's face it, we geeks are getting older. While there is still an influx of young people, we old ones are refusing to go away, and we are now married, and have kids, and leaving for days is unfair to our families.

So, I wish there was an event where I could go and do this, while my wife could go as well, and my son too, and there would be something for everyone. Maybe we would not hack all day long, but just half a day. Maybe we would hack more games and then beta test them on the kids.

One good thing about kids is that once you cross a certain threshold, in a controlled environment, the need for adult monitoring *decreases*. In m experience, 1 kid needs 1 adult, 2 kids need 1.5, and 5 kids need .5 adults.

The board games at night would be inclusive (my wife likes role games, for example), the juggling classes could include kids, people could work on artwork... I don't know, do things that are not exclusively hacking. I love hacking but it's a smaller part of my life than it was in the past.

So, anyone has any good ideas? Know of such a place or event?

Three years ago, I started a series of long posts called "PyQt by Example". It reached five posts before I abandoned for a series of reasons that don't matter anymore. That series is coming back starting next week, rewritten, improved and extended.

It will do so in a new site, and the "old" posts will be retired to an archive page. Why? Well, the technologies used in some of them are obsolete or don't quite work nowadays. So, the new versions will be the preferred ones.

And while I am not promising anything, I have enough written to make this something quite longer, more nicely layouted, more interesting and make it cover more

ground. BUT, while doing some checks on the traffic statistics for the old posts, some things popped out.

This was very popular About 60% of my site's traffic goes to those five posts. Out of about 1200 posts over 12 years, **60%** of the viewers go to the **0.4%** of the pages. That is a lot.

It's a long tail The traffic has not decreased in three years. If anything, it has *increased*

```
system-message
ERROR/3 in everything.txt, line 37461
Figure caption must be a paragraph or empty comment.

.. figure:: https://p.twimg.com/Aw0MHhoCAAAXmro.png:large
   :width: 80%

   A long and *tall*\ tail.

backrefs:
```

So, all this means there is a desire for PyQt documentation that is not satisfied. I am not surprised: PyQt is great, and the recommended book is not free, so there is bound to be a lot of demand.

And, here's the not-so-rosy bit: I had unobtrusive, relevant, out-of-the-way-but-visible ads in those pages for more than two years. Of the 70000 unique visitors, not even one clicked on an ad. Don't worry, I was not expecting to get money out of them (although I would love to some day collect a \$100 check instead of having google hold my money for me ad eternum).

But really? Not even *one* ad click? In more than two years, thousands of people? I have to wonder if I just attract cheap people ;-)

As I have mentioned in half a dozen posts already, I spent the last weekend at PyCamp 2012. But what I have not written about is what exactly it was, and why anyone would want to attend one, or maybe organize one. So that's what this post is about.

PyCamp was organized by PyAr, the Python Argentina community. PyAr is a very special bunch of people, who are completely amateur, and do everything for love and fun. Since PyAr is a very special group of people, the things PyAr causes, inspires or creates are special as well.

So, since a few years ago, what happens is someone finds a place with bunk beds, a large room, perhaps somewhat isolated, that provides meals, and is cheap (it's as hard as it sounds) and rents it for a long weekend. Then everyone is invited to chip in for the rent money.

This year, 4-days, all inclusive, costed roughly U\$S 100. Sure, it's not exactly luxury accomodations, but it does what it has to do, which is give us shelter and protect us from wild animals.

Thus, you end up with a few dozen nerds with computers, one of them is great at setting up wireless (Joac!), one is the MC (Alecú!), one helps around (Facundo!) one is the liaison with the location (Pindonga!) and so on, the work is spread around, and we have *time* and *company* to hack.

So, on the first morning, everyone proposes what he would like to work on. Those proposals are voted by the public, and those with more votes are assigned slots (5 a day), where they will be the main focus of attention.

So, what happens if your proposal is not voted? Well, you either find a proposal you like, and join it, or you just do your thing. Because this is not a democracy, this is anarchy, the votes are just a way for everyone to know what people will be doing, and to find places to fit in if you want (BTW, there is a situation in LeGuin's *The Dispossessed* which is so much like this, it's scary).

After that, you just do what you want. You can put your headset on, and code, or mingle and chat, or join a group, or do a bit of everything. Since meals are catered, you don't have to worry about breaks. When the meal is ready, everyone breaks at the same time and socializes in communitary tables.

Does all this sound as strange to you as it does to me? A bunch of grown professionals acting like hippies. Well, it feels strange too, but that doesn't mean it doesn't feel great. It even works great. Once you see what the others are doing, things you wouldn't expect start looking like fun (Celery!?! Juggernaut! Android!) and the sheer excitement of people telling you "look, I did this!" is infectious, and exhilarating.

Also, RC cars, kinect hacking, android hacking, electric guitar hacking, juggling, monocycle lessons, a firepit, alcohol, coffee, mate, boardgames, cardgames, music, jokes, adrenaline, huge spiders, asado, cold, vim, ninja, ping pong, robot spaceships, people you see only twice a year if that, questions, not knowing the answers, figuring things out on the run, getting help in that thing you have been stuck for weeks, having the piece someone else has been stuck on for weeks, feeling like some sort of bearded buddha and a total ignoramus in 5 minutes...

And at least I, at least this year, had a very productive weekend. I got help from a bunch of people in things I was daunted by, I felt like an active programmer instead of a suit, which is always nice, since I don't *own* a suit, and had a great time. Laughed a lot. Made a couple new friends. Saw a bunch of old ones. Helped a few people.

So, I would like other people to have as great a time as I had. Of course coming to Argentina is probably not a great idea. It's an expensive trip, if you don't speak spanish you will miss a lot, and if PyCamp gets too big it may stop being fun at all.

But why not do something similar? Doesn't have to be about Python, you can do it about making stuff, about programming in general, whatever. Just get a somewhat comfortable, somewhat isolated place with a reasonable catering and get your 50 nearest geeks there, and have a ton of fun.

You may get something useful done, too.

I was a sysadmin for a long time. I did that for money, so I never really wanted to spend time doing the same thing in my own time, which lead to a severe case of cobbler's children walking barefoot in my private server.

So, today at lunch, I decided to clean up my garbage. So this is what I ended up with, which is the minimal server that is good enough to be generally useful for me.

Hosting

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "burst.net".
```

This is a cheap VPS provided by the nice folks at burst.net. burst.net: <http://burst.net> who are not giving me anything to speak nice things about their service. However, I will do it anyway:

- Crazy cheap (\$5.50 but I have a 20% discount for life)
- Good amount of monthly bandwidth
- Lots of disk space
- Good uptime
- Fast network
- Very cheap
- Decent performance

Distribution

I had CentOS 5 installed, and it stays. If burst ever starts offering Ubuntu Precise, I may switch. Or, since this works, I may not.

What's good about CentOS? It's stable and boring.

What's bad about CentOS? It's too boring. Lots of cool stuff just isn't packaged.

Web Server

I need to serve a bunch of domains, but I have a peculiarity: they are all static sites. I want:

- Low resource usage
- Decent performance (that mostly involves supporting ranges and content negotiation)
- Stable
- Support directory indexes
- Easy configuration
- Virtual domains by name

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “gatling”.

Almost any server works well for this. Even Apache, except for the easy configuration bit. I ended up with [gatling](http://www.fefe.de/gatling/).. gatling: <http://www.fefe.de/gatling/>

because it fits those

system-message

WARNING/2 in `everything.txt`, line 37615
Block quote ends without a blank line; unexpected unindent. backrefs:

criteria fairly well.

- It uses about 1.4MB of RAM , which is always nice in a VPS

system-message

- **WARNING/2** in `everything.txt`, line 3
Duplicate explicit target name: “pretty fast”.

It’s [pretty fast](http://www.ferdianto.com/2009/09/05/webserver-benchmark-gatling-vs-nginx).. pretty fast: <http://www.ferdianto.com/2009/09/05/webserver-benchmark-gatling-vs-nginx>

- Has not crashed in 2 hours?
- Supports indexes
- Here’s the configuration: “-c /srv/www -P 2M -d -v -p 80 -F -S” (yes, there is no config file at all)
- Virtual domains are just folders and symlinks inside /srv/www which is the easiest possible way to do it.
- It supports reverse proxying for when I want to try a python web app I am working on.

Mail Server

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “ssmtp”.

No, I don’t want a mail server. I have gmail and/or a real mail server for that. I want to get the mails from cron. For this, I used [ssmtp](https://wiki.archlinux.org/index.php/SSMTP).. ssmtp: <https://wiki.archlinux.org/index.php/SSMTP> and an extra gmail account. It works, and here’s the whole config:

```
root=roberto.alsina@gmail.com
mailhub=smtp.gmail.com:587
UseTLS=YES
UseSTARTTLS=YES
AuthMethod=LOGIN
AuthUser=roberto.alsina.3@gmail.com
AuthPass=notputtingthetrueoneheredude
```

The best I can say about this configuration is that it works, and doesn't involve running a daemon.

Misc

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "openvpn".
```

For when I need to be in two places at the same time: [OpenVPN](http://openvpn.net/).. openvpn: <http://openvpn.net/>

rules, and

```
system-message
WARNING/2 in everything.txt, line 37662
Block quote ends without a blank line; unexpected unindent. backrefs:
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "quassel core".
```

there is no argument. I have a squid running occasionally, and there is a [Quassel core](http://quassel-irc.org/).. quassel core: <http://quassel-irc.org/>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "mosh".
```

for IRC stuff. I installed [mosh](http://mosh.mit.edu).. mosh: <http://mosh.mit.edu>

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "rsync".
```

to make ssh less painful, [rsync](http://en.wikipedia.org/wiki/Rsync).. rsync: <http://en.wikipedia.org/wiki/Rsync>

handles file deployment

```
system-message
WARNING/2 in everything.txt, line 37671
Block quote ends without a blank line; unexpected unindent. backrefs:
```

and backup storage, cron schedules stuff, and that's it.

Status

Plenty of free RAM and CPU (yes, that's the full process list):

```
[root@burst1 ~]# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1   2156    664 ?        Ss   22:01   0:00 init [3]
root      1135  0.0  0.1   2260    576 ?        S<s  22:01   0:00 /sbin/udev
root      1518  0.0  0.1   1812    572 ?        Ss   22:01   0:00 syslogd -m
root      1594  0.0  0.1   7240   1032 ?        Ss   22:01   0:00 /usr/sbin/
root      1602  0.0  0.2   4492   1112 ?        Ss   22:01   0:00 crond
root      1630  0.0  0.1   5684    716 ?        Ss   22:01   0:00 /usr/sbin/
root      1631  0.0  0.0   5684    444 ?        S    22:01   0:00 /usr/sbin/
root      1636  0.0  0.2   3852   1372 ?        S    22:01   0:01 /opt/diet/
root      1677  0.0  0.2   4284   1232 ?        Ss   22:02   0:00 SCREEN /ro
root      1678  0.0  2.1  36688  11148 pts/0    Ssl+ 22:02   0:03 /root/quas
root      3228  1.0  0.7  12916   4196 ?        Ss   23:28   0:13 mosh-serve
root      3229  0.0  0.3   3848   1588 pts/2    Ss   23:28   0:00 -bash
root      3275  0.0  0.1   2532    908 pts/2    R+   23:48   0:00 ps aux

[root@burst1 ~]# w
 23:49:03 up  1:47,  1 user,  load average: 0.00, 0.01, 0.00
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
root      pts/2    186.153.52.253  23:28   0.00s  0.01s  0.00s w

[root@burst1 ~]# free
              total        used        free      shared    buffers     cached
Mem:           524800         49100       475700          0           0           0
-/+ buffers/cache:         49100       475700
Swap:              0              0              0
```

All things considered, fairly happy with the result.

system-message

```
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "marcial candioti and ituzaiingó".
```

Walking to school, I always passed by a house full of bullet holes. That was, if memory serves, the corner of **Marcial Candioti and Ituzaiingó**.. marcial candioti and ituzaiingó: <http://g.co/maps/5v2nr>

. I never knew why that house was full of bullet holes. The window shades were always down, and noone ever explained it to me. It was not a long walk, just 4 blocks, from home to to the Mariano Moreno school.

I was curious about that house, since I was 7, and curiosity is the natural state of seven-year-olds everywhere, and there were not many houses like that (number of houses with bullet holes I've seen since: **0**). Of note is that this was 1977, an interesting year where there were about 600 bombings in Argentina, and there was a rampant dictatorship killing and kidnapping seemingly at random.

And I was walking to school, alone. And that was not strange for me. I had been moving around in the city by myself for *years*. I used to go from home to art classes by city bus when I was 5. Yes, I was taking city buses, alone, into the city downtown, in Argentina, while a dictatorship kicked out a president. So I could attend puppetry workshop.

My mother was a school principal in the worst slum imaginable. I was 4 and was under the watchful eye of the school cook and keepers. Which means that if they had to go buy something, I went with them, I played in the yards, I got lice every day, and parasites at least once.

My father used to make my brother sleep when he was a baby by putting him on the right seat of his car and driving through the night. And I don't mean "putting him in a seat" as in a safety seat, he was swaddled in a blanket, placed on a seat, no seatbelts, while my sleepy father drove through the night.

I used to spend the summers, when I was about 8, in a place where the expected activities were starting fires, building bows, pissing competitions, pinning crickets into boards, climbing 30 feet tall trees, eating random fruits from suspect bushes, fishing moribund bats from the swimming pool and trying to nurse them to health in pea cans, poking wasp nests, hunting toads, escaping to wander sand roads without telling anyone. One whole summer, we could see through the bus window a dead horse rotting in a ditch a few minutes before arriving. We washed our utensils with dirt. We drank well water from hoses. We carved wood using knives. We dugged pits using shovels.

When I was about twelve, I roamed the Buenos Aires downtown at night, bought theater tickets, attended the shows alone, rode the subway, got lost a lot, visited gaming arcades, ate pizza standing up, disappeared from morning to midnight.

I was remembering these things last night, talking with my wife about my son, who just turned five a few weeks ago, and I walk to school every morning. His school is roughly 3 blocks from home, in a nice neighborhood. The sole idea of him walking to school alone is unthinkable. How old should he be before he can walk to school? How old must he be before he can cross the street to buy candy. How old before he needs his own money, before he has a key to the house, before he can go to the park to play alone, before he can visit a friend down the street alone, before he can go to the toilet in public places alone.

I have no idea. The experience of having been 5 in 1976 doesn't help me know what being 5 in 2012 is like, I am at a loss, I am a foreigner from a different place, where our children fought wolves with their bare hands. I am not prepared for this. Either that or my parents were completely insane.

It sucks because it's *expensive*. Or at least it's expensive for how lame the hardware provided is.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "mele a1000 box".

The other day I got a [Mele a1000 box](http://www.pcworld.com/businesscenter/article/254097/70).. mele a1000 box: <http://www.pcworld.com/businesscenter/article/254097/70> for which I will do a full review soon, I hope. But I really need to get the word out: this thing kicks Raspberry PI in the ass so hard it's not even funny.

For starters, it costs twice as much. Which may seem bad, but trust me, even at twice the cost, it's cheaper. Because if you are the kind of person for whom the \$35 makes a difference, then you are also probably someone who doesn't have a HDMI-capable TV or monitor. And the Mele works with HDMI, but also with RCA connectors and VGA, which means pretty much any TV or monitor manufactured in the last 20 years will work in a pinch. You can repurpose ancient monitors (1024x768 CRTs from 1996? **WORKS FINE**) that are available everywhere.

Also, it comes with Wifi, which the Pi doesn't. And of course Ethernet, too.

It comes with 4GB of internal storage and 512MB of RAM. The Pi? None and 256mb. See a pattern here? You pay twice as much and it comes with more than twice as much hardware in it.

Also, it has *two* USB connectors (1 in the Pi model A, 2 in the Pi model B).

One important feature: it comes in a *box*. A nice, sturdy, small box, that can be nicely placed so that it doesn't break apart.

The CPU is several times faster and several generations more modern. The GPU is slightly less powerful, but you won't notice, the only thing the Pi can do with that GPU is decode video really quick, for anything remotely interesting the CPU is a bottleneck.

And both are proprietary, but at least someone is working on open Mali400 drivers.

The Mele is useful out of the box, you can just install Ubuntu on it, or use Android, which is open source. It's fully rooted, you don't need to do anything strange to install other operating systems in it (just like the Pi!) except that since the CPU is so much better, it can just do stuff easier.

Did I mention that while the Pi has only a SD card for storage while the Mele has 4GB of flash, *and* a SD slot, *and* an eSata slot? And that using that it could be a nice small file server?

And the kicker: it's not a charity. It's \$70 because you are paying a business to create the things, and doesn't depend on getting severely discounted chips from Broadcom, it just is a good product at a good price. And that's just *awesome* because it means in 2 years we will have something about as powerful as the Mele at the Pi's pricepoint, and something much more powerful at the Mele's price, and that's sustainable, because the people who are building it and designing it, and selling it, and shipping it, are making a living out of it, while kicking the ass of the charity. And that's *good*. And if you want 100000 of these, you order them, and you get them, instead of having to pre-buy, and wait months, and see them ship in tiny batches.

So, buy this, or something like it.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "free shipping".

Also: \$92.60 with [free shipping](http://dx.com/s/mele+a1000).. free shipping: <http://dx.com/s/mele+a1000>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "steady state theory".

The [Steady State Theory](http://en.wikipedia.org/wiki/Steady_State_theory).. steady state theory: http://en.wikipedia.org/wiki/Steady_State_theory

says, in rough

system-message

WARNING/2 in `everything.txt`, line 37830
Block quote ends without a blank line; unexpected unindent. backrefs:

terms that the universe has been and will be more or less as it currently is, because there are parallel processes of creation and destruction.

So, galaxies exhaust but then there are new galaxies, and the thing, as a whole, remains unchanged, in a way. Sure, it's not the same galaxy, and nothing that was in the old galaxy remains, but if you avoid specifics, things are the same.

I feel the same thing happens in the free software universe. The two forces are reaction and features.

Reaction is the creative force. Most, if not all, free software is reactive. It itches, I scratch, scratching is reactive. There are many examples:

- There is no free desktop! Let's create KDE!
- KDE is not the right kind of free, sort of! Let's create GNOME!
- KDE and GNOME are too heavy, bloated and whatever! Let's create XFCE!
- XFCE is not as lean and mean as before! Let's create LXDE!
- LXDE is (we'll find something) let's create WHATEVER-DE!

This even happens within a project:

- Hey, we created KDE!
- KDE 1.x has no object model and GNOME does, let's do KDE 2.0 on CORBA!
- Hey, that sucked, let's do KDE 2.0 on DCOP and KParts!
- Well, that's old and crufty, let's do KDE 3.0 cleaner!
- KDE 3.x looks old, let's throw all the UI away and do KDE 4!

And I am sure those familiar with any software product that has lived long enough to go through major rewrites and upheaval can do similar lists.

However, sometimes, the complains just don't go away.

- Netscape 4.x is slow and crufty! We rewrote it as Mozilla!
- Mozilla (SeaMonkey) is slow and crufty! We rewrote it as Firefox!
- Firefox is slow and crufty! We wrote Chrome using webkit!

And guess what people say about Chrome? It's slow.

So, the lesson there seems to me that writing a lightweight, generally useful, web browser is impossible. Why? Because of features.

As projects age, they grow features. Like the strange ear and nose hair men start growing in their 30s, features are a fact of the lifecycle. And with features come code, because that's how you do features.

And code is a liability, as (I hope) you all know. The more code you have, the more expensive it is to add things, and to be swift about improving your application. Most *successful* projects die, or grow senescent, hobbled by the weight of their features.

So how does a project stay young? I can think of a few ways.

It may have a benign (or evil, for that matter) dictator, with the right amount of hostility towards features (Linux). It may be so exquisitely modular that features don't couple with each other (emacs). It may reinvent itself every 5 years and throw everything away (KDE). It may have a very clear focus on one feature and a culture around it (Bacula).

And for each of those mechanisms, there are incountable examples of projects with too annoying dictators, projects overengineered to absurdity, stalled rewrites that never release and absolute focus on a feature noone cared about.

Sorry, the universe is a tough place.

Figure 41: Like it?

My niece goes to school. She has homework. Every now and then, her mom and greatmom will not be able to help her. I am sort of the math helper of last resort in the family. So, the other day I got this question:

What is the next number on this series:

2, 12, 16, 17, 18, 19

I just could not come up with an answer at the time. I have been told the answer is 200. Why? Because it's a list of numbers starting with "d".

Oh, you may say two doesn't start with "d" but hey, this was in spanish, so it does. The problem is, this is just not mathematics, at all.

It's about a completely incidental feature of the representation of a number in spanish, which is a completely unmathematical thing. Makes me wonder why they are making students think about this. Anyone knows?

Spanish only, since it's a post about how to buy stuff in China and having it delivered to Argentina :-)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "dx.com".

Hice mi compra más grande a [DX.com](http://dx.com).. dx.com: <http://dx.com>

desde que empecé con

system-message

WARNING/2 in `everything.txt`, line 37964
Block quote ends without a blank line; unexpected unindent. backrefs:

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "mele a1000".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "mele a1000".

esto: un **Mele A1000**.. mele a1000: <http://dx.com/p/mele-1080p-android-2-3-internet-tv-set-top-box-w-wifi-optic>

Y ya casi, casi, casi lo tengo en mis manos, pero se quedó en la aduana. Eso normalmente sería una muy mala noticia, ya que significaría que me tengo que tomar medio día del trabajo, ir a la aduana, esperar varias horas, pagar y retirarlo.

Pero ayer me llegó un telegrama que lo convirtió en una noticia, ni buena ni mala. El telegrama me dice que como llegó un paquete del exterior de un "remitente no informado" (?), tengo las siguientes opciones:

- 1) Pagar en el correo (cualquier sucursal) los derechos de aduana, y me va a llegar a casa en los próximos cinco días.
- 2) Llamar por teléfono, decir en qué sucursal lo quiero retirar, y pagar cuando lo retiro.
- 3) No hacer nada y que o lo devuelven o será "procesado como abandono".

Además, que si quiero usar mi franquicia anual para pagar menos, por favor avise.

Es *genial*. Es *honesto*. Hasta me pusieron el monto correcto a pagar (50% del valor, al cambio oficial), que no sé como hicieron.

En fin, que de ahora en más, comprar cosas de China parece que es más fácil que nunca. Y si compraste algo y querés contar cómo te fué, o querés leer sobre esas cosas que vamos pidiendo... bueno, <http://loquemellegodechina.blogspot.com>

It's not that I need a CDN in any way, since the traffic for this site is little and the way the site is built is light, but hey, it's free, easy to setup and easy to leave when I feel like it. And I expect to have significantly higher traffic eventually after I finish some not-so-secret projects.

What's CloudFlare's service? They take over your DNS, then put a reverse proxy between your site and the clients. That reverse proxy then uses a CDN to serve you the pages from a conveniently located server, and can rewrite the HTML/JS/CSS in some ways to make it faster/safer/nicer.

It also supposedly will protect my site from different kinds of attack (the only one that could possibly affect me was DOS attack, but thanks anyway ;-)

Also, they offer a platform so apps can provide services for me, like intruder detection, analytics, and others, which is a very cool idea.

So, I created an account at cloudflare.com and configured it so that <http://ralsina.com.ar> (which is this exact same site except for wrong comment counts) is served via cloudflare, and lateral.netmanagers.com.ar is served directly.

What I've seen so far:

- Setup is very simple
- It works, even setting up experimental features
- It does seem very slightly faster, but that's not a surprise since the tiny server the site runs on has good connectivity and ample unused resources.

- It does do a good job of automatically optimizing some things in ways that are generally accepted as a good idea (in other words, my pingdom and YSlow numbers moved up)

So: no pain, maybe some gain. I will probably move all sites into it tonight.
 Sorry, spanish only.

costa pobre.jpg costa pobre.jpg

Figure 42: Miren la banda.

No tengo muchos amigos. Tengo tres o cuatro, ponéle. Seguro no más de 10. Tengo muchos conocidos, tengo mucha gente que me cae bien, habrá alguno que le caigo bien, habrá otros que me conocen. He tenido amigos que nunca ví, tengo amigos que nunca veo, tengo amigos que nunca voy a ver porque tuvieron la mala idea de morirse antes que los vea.

Tengo amigos que quiero mucho, tengo amigos que me cagan de risa, tengo amigos y tengo amigas, tengo alguna hermana postiza que la quiero como si fuera mi hígado, tengo mi esposa que es más amiga, tengo amigos que quisiera ver más.

Hay amigos que no me conocen, pero si alguien te regala un libro que te cambia la vida, o una canción que te levanta a la mañana, o un programa de radio que te hace olvidar todas las noches la bosta que fué el día, esos son amigos míos, aunque yo no sea amigo suyo.

Hay gente que no conozco que me ha agradecido alguna cosa que hice y supongo que yo seré amigo suyo aunque no sean amigos míos. A ellos les digo que son amigos míos también.

No tengo amigos de la infancia, no tengo amigos de la adolescencia, tengo algún amigo de mis veintipico, tengo más de mis treintas, todavía no tengo uno de los cuarentaypico. Tengo amigos en otros países, tengo amigos acá al lado. Tengo pocos pero son variados. Tengo pocos pero son los mejores.

Y cuando me muera, si me hacen la gauchada de no morirse antes ellos, ojalá tenga una corona que diga: Tus Amigos. Y se la destapen y se la tomen, los putos.

As I hope you know, if you get a string of bytes, and want the text in it, and that text may be non-ascii, what you need to do is decode the string using the correct encoding name:

```
\ :s: '\ \ :gp: '>>> '\ \ :s: 'á' '\ \ :o: '. '\ \ :n: 'decode' \ \ :p: '(' \ \ :s:
\ :go: 'u' \xe1' '\
```

However, there is a gotcha there. You have to be *absolutely sure* that the thing you are decoding is a string of bytes, and not a unicode object. Because unicode objects also have a decode method but it's an incredibly useless one, whose only purpose in life is causing this peculiar error:

```

\ :s:'' \ :gp:'>>> '\ \ :s:'u'á'\ \ :o:'.'\ \ :n:'decode'\ \ :p:'('\ \ :s
\ :gt:'Traceback (most recent call last):
'\ \ :err:'File "<stdin>", line 1, in <module>'\
\ :go:'File "/usr/lib/python2.7/encodings/utf_8.py", line 16, in decode
    return codecs.utf_8_decode(input, errors, True)
UnicodeEncodeError: 'ascii' codec can't encode character u'\xel'
in position 0: ordinal not in range(128)'\

```

Why peculiar? Because it's an **Encode** error. Caused by calling **decode**. You see, on unicode objects, decode does something like this:

The user wants a unicode object. He *has* a unicode object. By definition, there is no such thing as a way to utf-8-decode a unicode object. It just makes **NO SENSE**. It's like asking for a way to comb a fish, or climb a lake.

What it should return is `self`! Also, it's annoying as all hell in that the only way to avoid it is to check for type, which is totally unpythonic.

Or even better, let's just not have a decode method on unicode objects, which I think is the case in python 3, and I know we will never get on python 2.

So, be aware of it, and good luck!

Because of varied reasons, I have spent a little time in my life looking at SSL certificates. I have spent time creating, signing, configuring, buying, deploying and validating them. I have hardly ever seen something like Valicert, though.

So, a user has a SSL error, and there appears a certificate that has nothing to do with the site he's supposed to be accessing. It's marked as invalid (nevermind) and is from a CA I have never heard of, called Valicert.

It seems to be a valid CA, its certificates mostly validate, etc, but something is fishy (besides the fact that there is a freaking Valicert certificate where there should not be one and I have no idea why).

```

system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "valicert.com".

```

It seems Valicert is or was owned by GoDaddy, which should already be a problem, but it gets worse. The URL for the CA? valicert.com.. valicert.com: <http://valicert.com>

which is not

```

system-message
WARNING/2 in everything.txt, line 38125
Block quote ends without a blank line; unexpected unindent. backrefs:

```

```

system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: "www.valicert.com".

```

configured as a site. Then I try www.valicert.com.. www.valicert.com: <http://www.valicert.com>

which

system-message

WARNING/2 in `everything.txt`, line 38128

Block quote ends without a blank line; unexpected unindent. backrefs:

does exist but is not about a CA but about AxWay, a “Business Interaction Networks company”.

It contains nuggets like “What can our cloud-based community management solution do for your bottom line?” and “Is your file transfer system visibility-impaired?” and “Our award-winning products, solutions and services enable the business-critical transactions required to accelerate performance within and among enterprises – while providing management, security and governance on interactions throughout business networks.”

And then I had the (bad) idea to try <https://www.valicert.com> ... a picture should be enough:

These bozos, this company that dares offer file transfer whatnots and has “award winning products”... has a self-signed certificate, that expired in 2010, for localhost freaking localdomain in their public webserver.

So, I am guessing Valicert doesn’t exist anymore, godaddy kept the CA alive until all certs expire and for some reason AxWay is a bunch of incompetents who bought the domain (but why???) and really, trusting CAs is getting harder each day.

If you see this, you may notice some changes in the site.

So, here is a short explanation:

- I changed the software and the templates for this blog.
- Yes, it’s a work in progress.
- The new software is called Nikola.
- Yes, it’s pretty cool.

Why change? Are you kidding? My previous blog-generator (Son of BartleBlog) was not in good shape. The archives only covered 2000-2010, the “previous posts” links were a lottery, and the spanish version of the site was missing whole sections.

So, what's Nikola? Nikola is a static website generator. One thing about this site is that it is, and has always been, just HTML. Every “dynamic” thing you see in it, like comments, is a third party service. This site is just a bunch of HTML files sitting in a folder.

So, how does Nikola work?

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “restructured text”.
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “restructured text”.
```

Nikola takes a folder full of txt files written in [restructured text](#).. restructured text: <http://docutils.sourceforge.net>, and generates HTML *fragments*.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “mako templates”.
```

Those fragments plus some light metadata (title, tags, desired output filename, external links to sources) and Some [Mako Templates](#).. mako templates: <http://www.makotemplates.org> create HTML pages.

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “bootstrap”.
```

```
system-message
WARNING/2 in everything.txt, line 3
Duplicate explicit target name: “bootstrap”.
```

Those HTML pages use [bootstrap](#).. bootstrap: <http://twitter.github.com/bootstrap/> to not

```
system-message
WARNING/2 in everything.txt, line 38195
Block quote ends without a blank line; unexpected unindent. backrefs:
```

look completely broken (hey, I never claimed to be a designer).

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "doit".

system-message

WARNING/2 in `everything.txt`, line 3

Duplicate explicit target name: "doit".

To make sure I don't do useless work, `doit..` `doit:` <http://doit.sourceforge.net> makes sure only the required files are recreated.

Why not use <whatever>? Because, for diverse reasons, I wanted to keep the exact URLs I have been using:

- If I move a page, keeping the Disqus comments attached gets tricky
- Some people may have bookmarked them

Also, I wanted:

- Mako templates (because I like Mako)
- Restructured text (Because I have over 1000 posts written in it)
- Python (so I could hack it)
- Easy to hack (currently Nikola is under 600 LOC, and is almost feature complete)
- Support for a multilingual blog like this one.

And of course:

- It sounded like a fun, short project. I had the suspicion that with a bit of glue, existing tools did 90% of the work. Looks like I was right, since I wrote it in a few days.

Are you going to maintain it? Sure, since I am using it.

Is it useful for other people? Probably not right now, because it makes a ton of assumptions for my site. I need to clean it up a bit before it's really nice.

Can other people use it? Of course. It will be available somewhere soon.

Missing features? No tags yet. Some other minor missing things.

It starts with these two aliens:

And a lawyer called Nick Carter, who is not *this* Nick Carter:

If I were to describe the plot, it would make me sound insane, which is a good thing. So, I will just let the book trailer do the work:

Have you ever read Douglas Adams and wished the plot started making some sense? Have you ever read Terry Pratchett and wished there was more than one ex-cruciatingly stretched joke per book?[#id3]

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “year zero”.

Well, if you have, I recommend you give [Year Zero](http://www.goodreads.com/book/show/12953520-y).. year zero: <http://www.goodreads.com/book/show/12953520-y>
a try. It’s hilarious, it has a plot of sorts, and has at least three different jokes in it.
A working knowledge of lame 80s (and 70s) music helps but is not horribly necessary.
So, I give this five stars[#id4]_ and recommend it to every one.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “here”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “here”.

This is a short story, in spanish, so if you want to read it, click [here](http://lateral.netmanagers.com.ar/tr/es/weblog/posts/yendo-para-pycon.html).. here:
<http://lateral.netmanagers.com.ar/tr/es/weblog/posts/yendo-para-pycon.html>

This may cause some palpitations in some friends of mine who laugh at me for using kwrite, but it really is not. Any time you spend configuring, choosing, adjusting, tweaking, changing, improving, patching or getting used to your editor is time invested, for which you need to show a benefit, or else it’s time wasted.

19

system-message

WARNING/2 in `everything.txt`, line 38261
Duplicate explicit target name: “id3”.

And yes, I know that’s practically the point of Terry Pratchett’s style.

20

system-message

WARNING/2 in `everything.txt`, line 38263
Duplicate explicit target name: “id4”.

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “a naked singularity”.

It’s the second 5-star book for me this year, after [A Naked Singularity](http://www.goodreads.com/book/show/6643543-a-naked-singularity).. a naked singularity:
<http://www.goodreads.com/book/show/6643543-a-naked-singularity>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: “the mongoliad”.

I didn’t even give those to [The Mongoliad](http://www.goodreads.com/book/show/12853147-the-mongoliad).. the mongoliad:
<http://www.goodreads.com/book/show/12853147-the-mongoliad>
even though I am a hopeless Neal Stephenson fanboy.

Let's look at SLOC, which while discredited as a measure of programmer's productivity, surely does work as a measure of how much a programmer *types*, right?

Well, estimates of total code production across the lifetime of a product vary (just take the SLOC of the product, divide by men/days spent), but they are usually something between 10 and 100 SLOC per programmer per day. Let's be generous and say 200.

So, 200 lines in eight hours. That's roughly one line every two minutes, and the average line of code is about 45 characters. Since I assume you are a competent typist (if you are not, shame on you!), it takes less than 20 seconds to type that.

So, typing, which is what you often tweak in your editor, takes less than 15% of your time. And how much faster can it get? Can you get a line written in 10 seconds? Then you just saved 8% of your day. And really, does your editor save you *half* the typing time?

How much time do you lose having your eyes wonder over the sidebars, buttons, toolbars, etc?

So while yes, typing faster and more efficiently is an optimization, it may also be premature, in that, what the **hell** are we doing the other 80% of the time? Isn't there something we can do to make that huge chunk of time more efficient instead of the smaller chunk?

Well, I think we spent most of that time doing three things:

- 1) Reading code
- 2) Thinking about what code to write
- 3) Fixing what we wrote in that other 20%

The first is easy: we need better code *readers* not *editors*. It's a pity that the main interface we get for looking at code is an editor, with its constant lure towards just *changing* stuff. I think there is a lost opportunity there somewhere, for an app where you can look at the code in original or interesting ways, so that you understand the code better.

The second is harder, because it's personal. I walk. If you see me walking while my editor is open, I am thinking. After I think, I write. Your mileage may vary.

The third is by far the hardest of the three. For example, autocomplete helps there, because you won't mistype things, which is interesting, but more powerful approaches exist, like constant running of tests suites while you edit. Every time you leave a line, trigger the affected parts of the suite.

That's much harder than it sounds, since it means your tools need to correlate your test suite to your code very tightly, so that you will see you are breaking stuff the *second* you break it, not minutes later.

Also, it should enable you to jump to the test with a keystroke, so that you can fix those tests if you are changing behaviour in your code. And of course it will mean you need tests ;-)

Which brings me to a pet peeve of mine, that editors still treat the file as the unit of work, which makes no sense at all. You never want to edit a file, you want to edit a function, or a class, or a method, or a constant but never a file. Knowing this was the sheer *genius* of ancient Visual Basic, which was completely ignored by all the snobs looking down at it.

So, instead of tweaking your editor, get me a tool that does what I need please. I have been waiting for it since VB 1.0. And a sandwich.

warning

This post is 99% lies, but I want to hear the arguments against it. If I tell you now it doesn't count as a real lie, I have learned from financial press ;-)

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "reddit".

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "reddit".

UPDATE: Interesting discussions in `reddit`.. reddit: http://www.reddit.com/r/programming/comments/wnkrr/your_

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "hacker news".

and `hacker news`.. hacker news: <http://hackerne.ws/item?id=4251109>

system-message

WARNING/2 in `everything.txt`, line 3
Duplicate explicit target name: "i am trying!".

Since I decided to stop being a troll (**I am trying!**.. i am trying!: <http://t.co/RHhfBUMg>) I noticed that people take what I write waaaaay too seriously. This post is a gentle reminder that I have posted more about silly stuff than about serious stuff. **MUCH** more.

So, if you ever find yourself thinking "Hey, Roberto seems to be making an interesting point", first think about the twin bananas. If what I wrote still looks interesting, proceed.

Docutils System Messages

system-message

ERROR/3 in `everything.txt`, line 18881
Too many autonumbered footnote references: only 0 corresponding footnotes available. backrefs:

system-message

ERROR/3 in `everything.txt`, line 1804
Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 1808

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 1822

Duplicate target name, cannot be used as a unique reference: "3".

system-message

ERROR/3 in everything.txt, line 1828

Duplicate target name, cannot be used as a unique reference: "4".

system-message

ERROR/3 in everything.txt, line 2463

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 2538

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 2538

Duplicate target name, cannot be used as a unique reference: "3".

system-message

ERROR/3 in everything.txt, line 2571

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 2888

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 2901

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 2929

Duplicate target name, cannot be used as a unique reference: "3".

system-message

ERROR/3 in everything.txt, line 2942

Duplicate target name, cannot be used as a unique reference: "4".

system-message

ERROR/3 in everything.txt, line 2978

Duplicate target name, cannot be used as a unique reference: "5".

system-message

ERROR/3 in everything.txt, line 6862

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 6864

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 6874

Duplicate target name, cannot be used as a unique reference: "3".

system-message

ERROR/3 in everything.txt, line 6876

Duplicate target name, cannot be used as a unique reference: "4".

system-message

ERROR/3 in everything.txt, line 6880

Duplicate target name, cannot be used as a unique reference: "5".

system-message

ERROR/3 in everything.txt, line 6886

Duplicate target name, cannot be used as a unique reference: "6".

system-message

ERROR/3 in everything.txt, line 6894

Duplicate target name, cannot be used as a unique reference: "7".

system-message

ERROR/3 in everything.txt, line 13828

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 13856

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 14111

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 14228

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 14601

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 14601

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 15180

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 15261

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 15261

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 15288

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 15310

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 15396

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 15406

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 15420

Duplicate target name, cannot be used as a unique reference: "4".

system-message

ERROR/3 in everything.txt, line 15464

Duplicate target name, cannot be used as a unique reference: "3".

system-message

ERROR/3 in everything.txt, line 15494

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 15496

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 16112

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 16558

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 16564

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 16586

Duplicate target name, cannot be used as a unique reference: "3".

system-message

ERROR/3 in everything.txt, line 16959

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 16988

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 16988

Duplicate target name, cannot be used as a unique reference: "3".

system-message

ERROR/3 in everything.txt, line 16992

Duplicate target name, cannot be used as a unique reference: "4".

system-message

ERROR/3 in everything.txt, line 17044

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 17046

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 17727

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 18881

Duplicate target name, cannot be used as a unique reference: "id3".

system-message

ERROR/3 in everything.txt, line 18889

Duplicate target name, cannot be used as a unique reference: "id4".

system-message

ERROR/3 in everything.txt, line 19022

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 20920

Duplicate target name, cannot be used as a unique reference: "5".

system-message

ERROR/3 in everything.txt, line 20935

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 20955

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 20959

Duplicate target name, cannot be used as a unique reference: "3".

system-message

ERROR/3 in everything.txt, line 20963

Duplicate target name, cannot be used as a unique reference: "4".

system-message

ERROR/3 in everything.txt, line 21017

Duplicate target name, cannot be used as a unique reference: "id3".

system-message

ERROR/3 in everything.txt, line 21017

Duplicate target name, cannot be used as a unique reference: "id4".

system-message

ERROR/3 in everything.txt, line 22336

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 23242

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 23378

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 23380

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 23420

Duplicate target name, cannot be used as a unique reference: "3".

system-message

ERROR/3 in everything.txt, line 23926

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 23934

Duplicate target name, cannot be used as a unique reference: "4".

system-message

ERROR/3 in everything.txt, line 23962

Duplicate target name, cannot be used as a unique reference: "5".

system-message

ERROR/3 in everything.txt, line 23972

Duplicate target name, cannot be used as a unique reference: "7".

system-message

ERROR/3 in everything.txt, line 23976

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 23978

Duplicate target name, cannot be used as a unique reference: "3".

system-message

ERROR/3 in everything.txt, line 23984

Duplicate target name, cannot be used as a unique reference: "6".

system-message

ERROR/3 in everything.txt, line 24027

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 26194

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 26317

Duplicate target name, cannot be used as a unique reference: "id3".

system-message

ERROR/3 in everything.txt, line 26331

Duplicate target name, cannot be used as a unique reference: "id4".

system-message

ERROR/3 in everything.txt, line 28362

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 28372

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 28378

Duplicate target name, cannot be used as a unique reference: "3".

system-message

ERROR/3 in everything.txt, line 28382

Duplicate target name, cannot be used as a unique reference: "4".

system-message

ERROR/3 in everything.txt, line 29870

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 29922

Duplicate target name, cannot be used as a unique reference: "2".

system-message

ERROR/3 in everything.txt, line 31550

Duplicate target name, cannot be used as a unique reference: "1".

system-message

ERROR/3 in everything.txt, line 38248

Duplicate target name, cannot be used as a unique reference: "id3".

system-message

ERROR/3 in everything.txt, line 38259

Duplicate target name, cannot be used as a unique reference: "id4".