



Python en GNOME

Alvaro Lopez Ortega
alvaro@alobbs.com

Que es Python

- Alto nivel
- Interpretado
- Orientado a objetos
- Tipado dinamico
- Tipos de alto nivel
- Bytecode

Que es Python (II)

- Desarrollo raaaaapido de aplicaciones
- Muy productivo (dias/persona)
- Libre
- Interprete interactivo
- Extensible (C / C++ / ..)
- 9 a~os de Python

Que es Python (III)

□ Gran portabilidad:

○ Unix

- ▷ Linux
- ▷ (Free/Open/Net)BSD
- ▷ Sun Solaris
- ▷ SGI IRIX
- ▷ DEC Unix, Ultrix
- ▷ AIX
- ▷ HP-UX
- ▷ SCO Unix
- ▷ NeXT
- ▷ Minix

○ MS-DOS, Windows (3.11, 95, 98, NT, 2000, CE)

○ Macintosh

○ OS/2

○ PalmOS

○ y otros: BeOS, Amiga, QNX, VMS, Psion Epoc, Cray T3E, RISC OS y VxWorks

Pequen~a historia

- ABC
- Guido Van Rossum en 1990
- CNRI (2.0.1 compatible con GPL)
- Ahora propiedad de PSF

El lenguaje

- El movimiento se aprende andando ;-)

```
import smtplib
```

```
con = smtplib.SMTP('localhost')
```

```
con.sendmail('yo@micasa.es',
```

```
            'gnome-desarrollo@es.gnome.org',
```

```
            'Hola a todos :-)')
```

```
con.quit()
```

Muy alto nivel
Poco tiempo de desarrollo

El lenguaje

- Paquetes, Modulos, Clases, Funciones..
- Polimorfismo y herencia multiple
- Excepciones
- Sintaxis "corta" por bloques

Comparacion (Python)

```
f = open('/tmp/borrar', 'wb')
for i in xrange(1000000):
    f.write(str(i))
f.close()
```

Comparacion (Java) ;-G

```
import java.io.*;
public class ComparacionPython {
    public static void main(String[] args) {
        try {
            File f = new File("/tmp/borrar");
            PrintWriter ps = new PrintWriter
                (new OutputStreamWriter (new
                    FileOutputStream(f)));
            for (int i = 0; i < 1000000; i++) {
                ps.print(String.valueOf(i));
            }
            ps.close();
        } catch(IOException ioe) {
            ioe.printStackTrace();
        }
    }
}
```

Codigo

- Metodos especiales en las clases:
 - `__init__`
 - `__del__`
 - `__add__`, `__sub__`, `__mod__`, ...

- Prog. declarativa + funciones lambda
 - `impares = filter (lambda x: x%2, lista)`

 - `suma_total = reduce (lambda x,y: x+y, lista)`

 - `cubos = map (lambda x: x**2, lista)`

Codigo (II)

- Clases == Tipos (Python 2.2)

```
class defaultdict(dict):
    def __init__(self, default=None):
        dict.__init__(self)
        self.default = default

    def __getitem__(self, key):
        try:
            return dict.__getitem__(self, key)
        except KeyError:
            return self.default
```

Codigo (ejemplo real): Gbot

- 224 lineas de codigo

- Op
- Fortune
- Noticias de BP y news.google.com
 - asincrono
 - multithread
- Mas funcionalidades
 - 'seen'
 - 'join/part'
 - 'sex'

gbot.py

GNOME desde Python

- Es posible escribir aplicaciones GNOME desde Python (James Henstridge)
 - PyGTK
 - PyGNOME

- Binding de GNOME 2.0

- GObject

- Glade

- Bonobo

Ejemplo GnomeAbout

```
from gtk import *
from gnome.ui import *

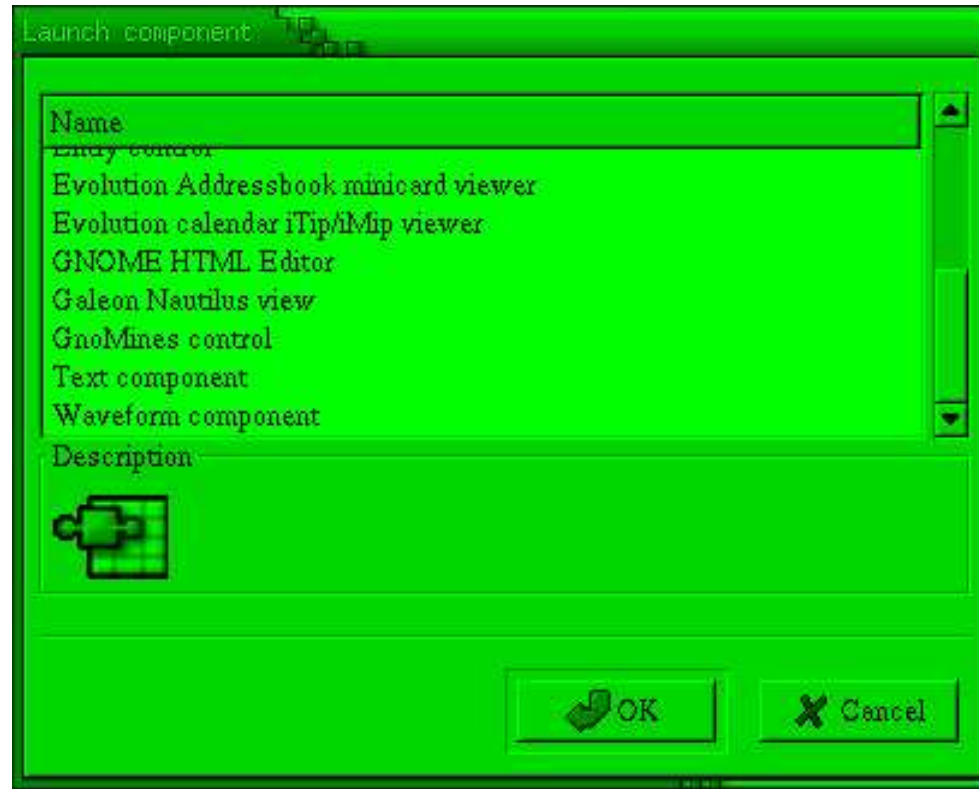
about = GnomeAbout('Titulo', 'Version',
                   'Copyright',
                   ['Autor1', 'Autor2', '...'],
                   'Comentarios')

about.connect('destroy', mainquit)
about.show()
mainloop()
```



Bonobo

- Uso de componentes Bonobo en una aplicacion Python



Bonobo (II)

- Componentes Bonobo escritos en Python

```
import bonobo
```

```
import gtk
```

```
def control_factory (factory, object_id):
```

```
    label = gtk.GtkLabel ("Sample Control")
```

```
    label.show()
```

```
    return bonobo.BonoboControl (label)
```

```
bonobo.BonoboGenericFactory (
```

```
    "OAFIID:GNOME_SampleControlFactory",
```

```
    control_factory)
```

```
bonobo.main ()
```

Glade

```
import gtk, gtk.glade
```

```
xml = gtk.glade.XML("mifichero.glade")
```

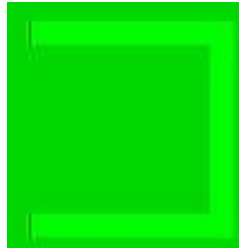
```
xml.signal_autoconnect(locals())
```

```
gtk.main()
```



GNOME Canvas

```
canvas = GnomeCanvas()  
canvas.set_size (300, 300)  
canvas.root().add ('line',  
                    points = (10,10, 90,10,  
                              90,90, 10,90),  
                    width_pixels = 10,  
                    fill_color = 'blue')  
canvas.show()
```



Aplicaciones con extensiones en Python

- Guppi3
- Dia (plugging de scripting)
- Gnumeric (soporte en plugin)
- XChat
- Eighty
- Blade ;-)

Extension de aplicaciones

- Caso mas interesante

```
#include <Python.h>
```

```
int main (int argc, char **argv)
{
    Py_Initialize();
    PyRun_SimpleString( ..... );
    Py_Finalize();

    return 0;
}
```

Extension de aplicaciones (II)

Dos sentidos

Binding

○ Python -> C

○ funcion C -> Python

Extension

○ C -> Python

○ funcion Python -> C

```
PyObject *pEjemplo;  
pEjemplo = PyString_FromString("Hola");  
Py_DECREF(PyObject);
```

Mas informacion

<http://www.python.org>

▷ /doc/current

comp.lang.pyhon

<http://www.gnome.org/~james>

<http://www.es.gnome.org>

<http://listas.aditel.org/listinfo.py/python-es>