



# Python

## Primeros Pasos

Juan José Iglesias González - [bille@gpul.org](mailto:bille@gpul.org)

Adrián Pérez de Castro - [moebius@gpul.org](mailto:moebius@gpul.org)



# Guión

- [ Spam & Eggs
- [ Why Python?
- [ Let's be friends!





# Spam & Eggs

De cómo ciertos personajes influyen en uno de los desarrollos más veteranos del software libre





# Spam & Eggs

Autor: Guido Van Rossum

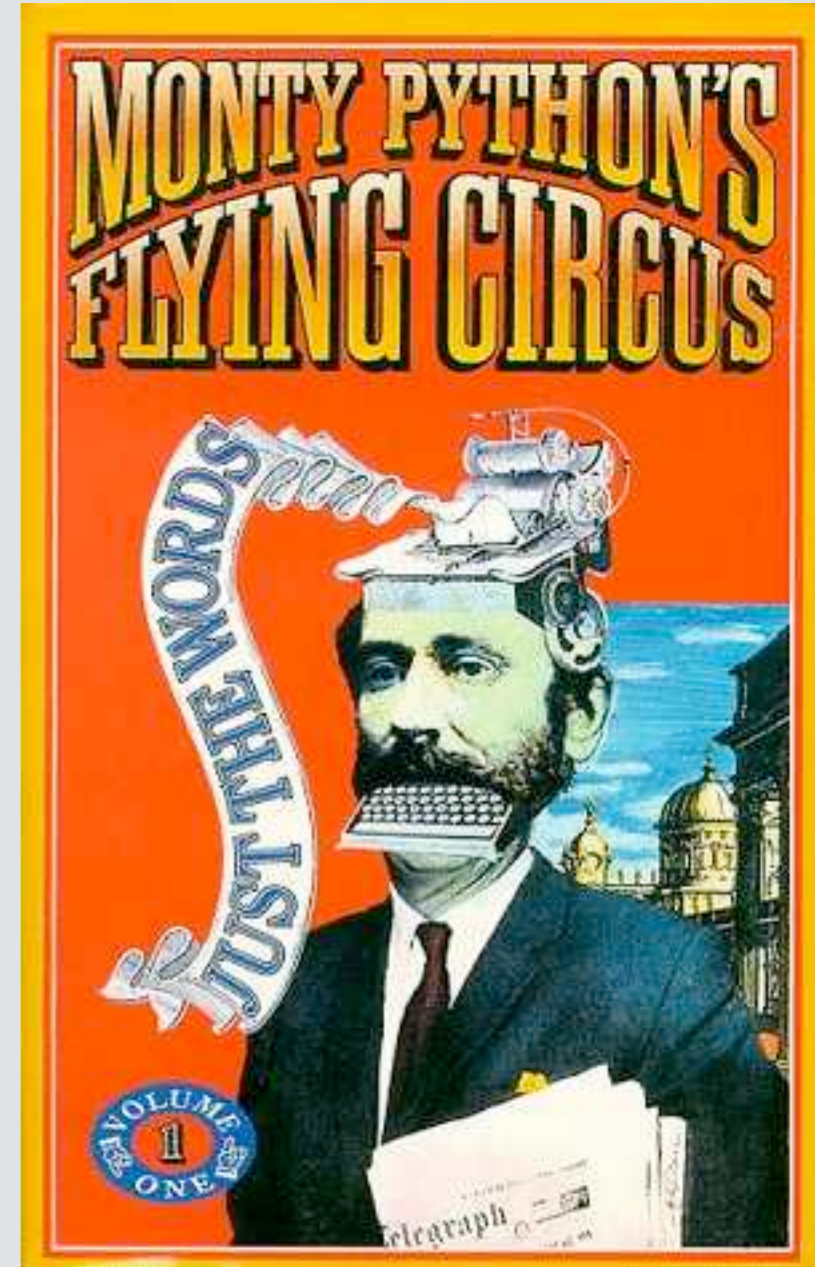
- Instituto Nacional de Investigación Matemática e Informática, Noruega
- Mediados de los 80
- Lenguaje ABC: enseñar programación a no informáticos
- ABC no tuvo repercusión





# Spam & Eggs

- Amoeba + Python = Python
- Lenguaje de scripting para Amoeba
- Basado en ABC
  - Nueva sintaxis de bloques
  - Tipos predefinidos
- Python es mucho más **fácil de extender** de forma modular





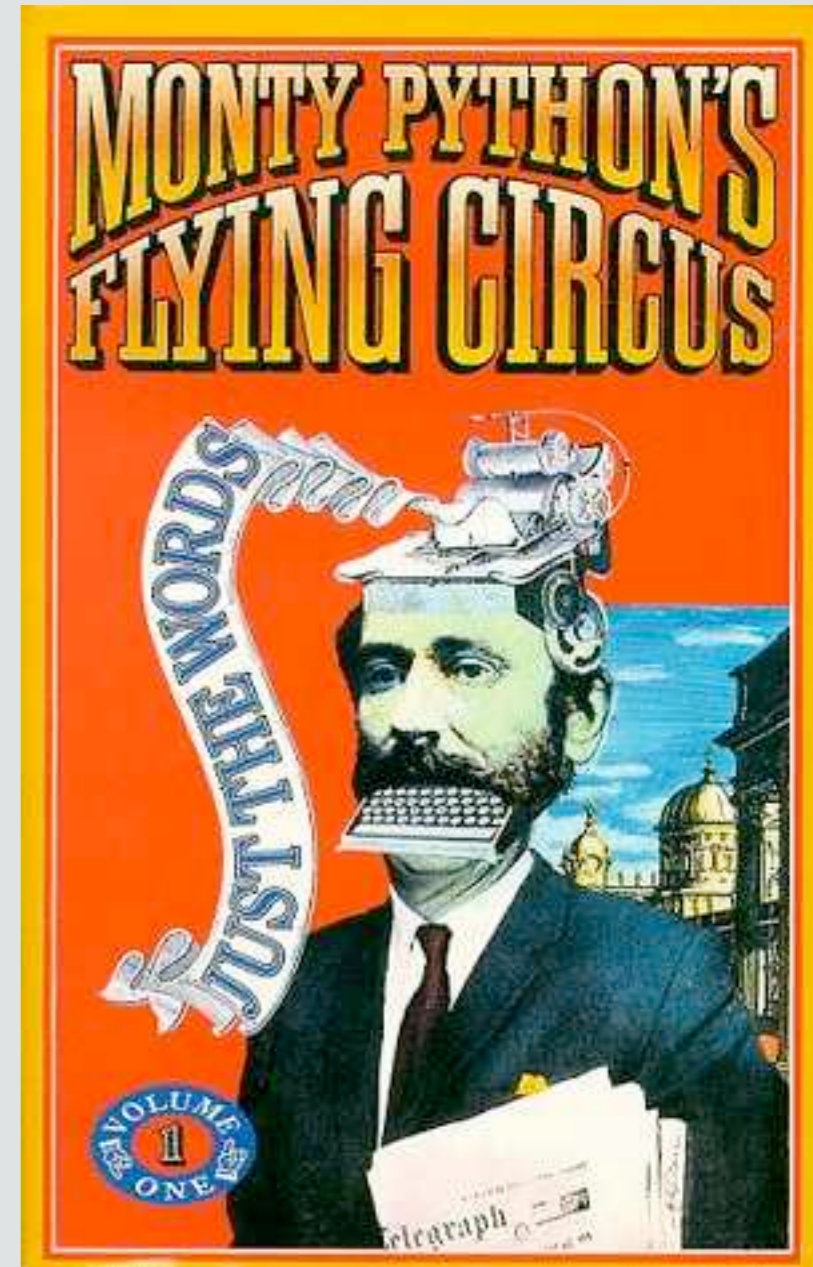
# Spam & Eggs

Sistema Operativo Linux

V Jornadas

## Filosofía

- Python debe ser divertido
- Su nombre proviene de la serie de televisión «Monty Python's Flying Circus»
- Son habituales referencias al genial grupo humorístico inglés al hablar de Python 8^)
- Más zen: `import this`





# Why Python?

Razones, hechos y ejemplos de por qué puede ser conveniente tener un reptil a mano



# Why Python? – Hoooola!!

Un ejemplo clásico

```
print "Hello world!"
```

clásico

Otro ejemplo clásico

módulo

```
import sys
```

variable

```
name = sys.argv
```

identidad

```
if len(name) == 2:
```

indexado

```
    print "Hello " + name[1]
```

```
else:
```

concat

```
    print "Hello random chorbo"
```

bloque







# Why Python? – Tipos

Todo son objetos – Incluso los que vienen de serie

método

```
lst = ["spam", "and"]  
lst.append(3)  
lst.append("eggs")
```

lista

iterador

```
for item in lst:  
    print item,
```

intro

```
# We also have "while"  
i = 0  
while i < len(lst):
```

polim.

indent.

```
    print lst[i],  
    i += 1      # But not i++!!
```





# Why Python? – Tipos

Todo son objetos – Incluso los que vienen de serie

hash

```
d = { "eggs": 3, "milk": 1 }  
d["cornbeef"] = "some"
```

hetero

```
print "Supermarket List:"  
for item in d:  
    print "_" * 3, d[item], item
```

cast

```
d = dict(  
    [(x, x+1) for x in range(5)]  
)  
print repr(d)
```

mult.

compres.

pretty

rango





# Why Python? – Lambda!

## «Pitonización» funcional

slice?

fun!

```
l = range(5, 15, 2) # Woahh!!  
m = map(lambda x: x+1, l)  
print repr(l)
```

BTBF

## «Pitonización» recursiva

WAE

```
def fact(n):  
    if n == 0: return 1  
    else: return n * fact(n-1)  
print fact(5)
```

call





# Why Python? – 00

## Tratando reptiles como objetos

clase

```
class Reptile:  
    def __init__(self, name='Pitty', **a):  
        self._name = name  
        self._attr = a
```

defecto

atrib.

hash

método

```
    def y(self, name):  
        return self._attr.get(name)  
    def x(self): return self._name
```

obj.

```
p = Reptile('Python', age=5, eyes='blue')  
q = Reptile('Boa', age='3', length=2)
```

```
print p.x(), p.y('age'), p.y('eyes')  
print q.x(), q.y('age'), q.y('eyes')  
p._name = 'Mutant-Python'  
print p.x()
```

None

oculto?





# Let's be friends!

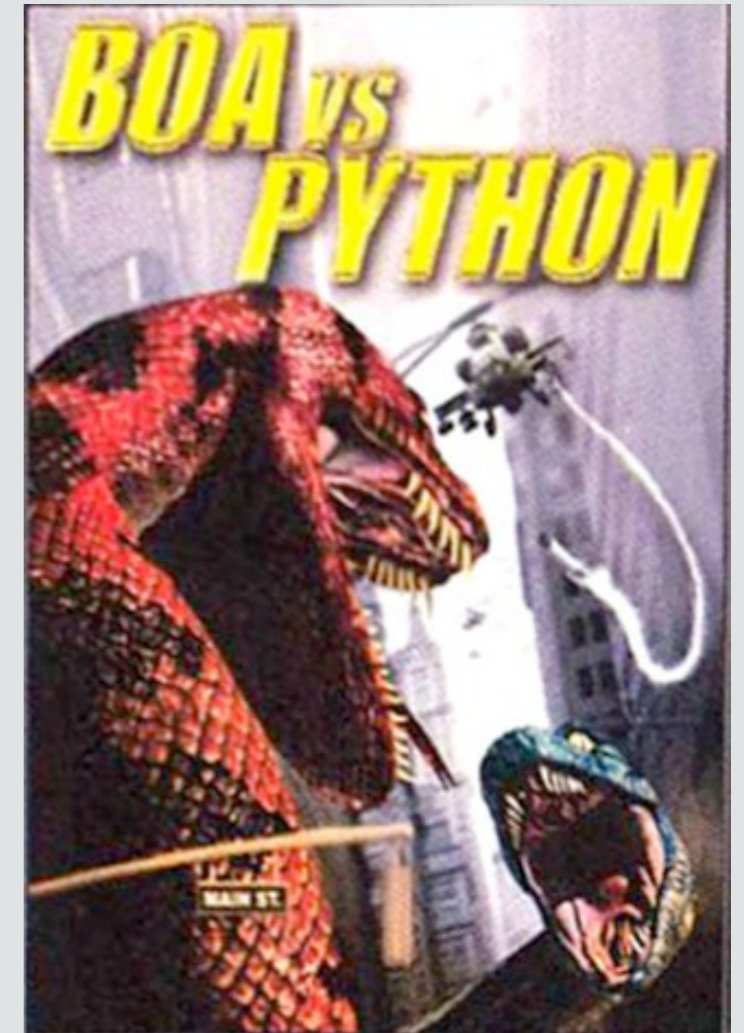
¿Miedo a estar sólo? Python tiene todas las respuestas



# Let's be friends! – Usos

## En empresas

- NASA
- Yahoo, Google, Infoseek
- Industrial Light & Magic, Pixar
- Red Hat
- Microsoft
- Blade

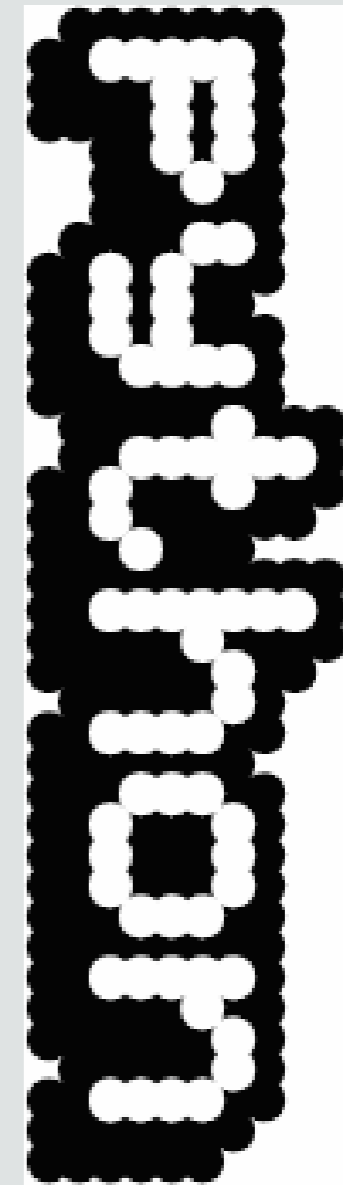




# Let's be friends! – Tech

## Módulos incorporados

- Más de 200: REs, compresión, serialización y persistencia automáticas, servicios de red, threads, codificación de archivos, bases de datos, XML, ncurses, Tk, audio, internacionalización, SGML, interfaz con el sistema operativo, números aleatorios, XDR, ...
- Además: módulos de terceros

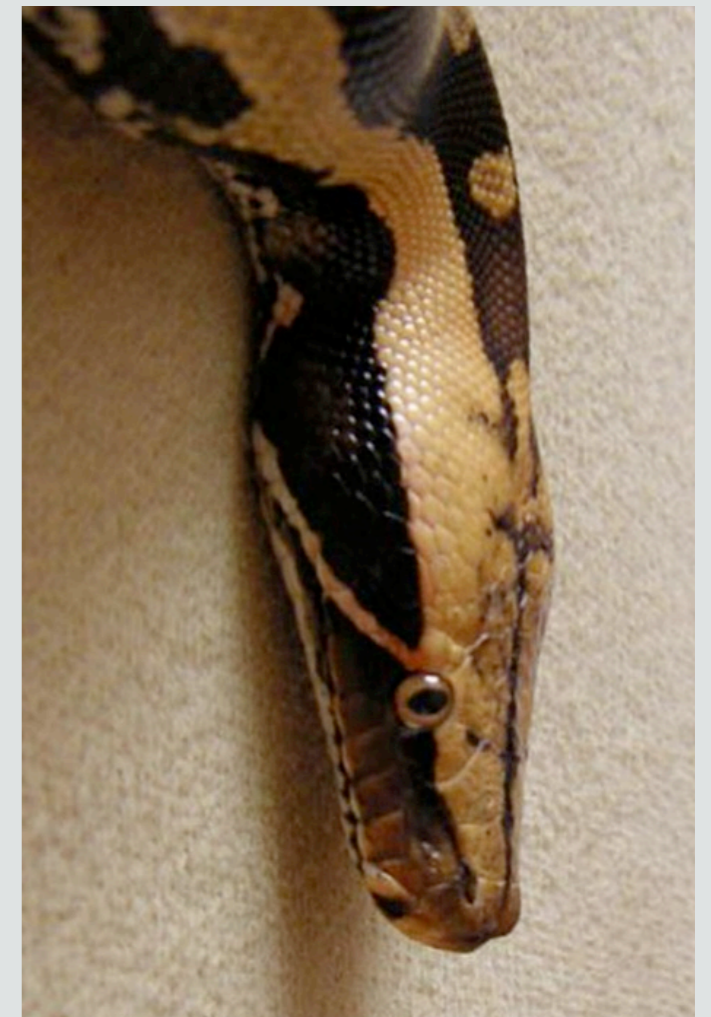




# Let's be friends! – GUIs

## Desarrollo de GUIs

- Qt & KDE, GTK+ & Gnome
- AWT & Swing (con Jython)
- FOX, wxWindows, Tk, Cocoa (con PyObjC), Win32, Windows.Forms (a través de .NET)...
- Más en la charla de programación de GUIs con Python ;-)







# Let's be friends! – XREFs

## Servicios web

— XML-RPC, SOAP, servidor/cliente HTTP, mod\_python, **Zope**...

## Interoperabilidad

— .NET, Jython (Java™), SWIG (C & C++), PyObjC (Cocoa), Corba...

## Aplicaciones cliente/servidor

— BitTorrent, Nicotine, PyMusique

## Juegos

— PyGame, PyDance





# Gracias, Bryan

Sistema Operativo Linux

V Jornadas

## DOCTOR FUN

6 Apr 2000



Copyright © 2000 David Farley, d-farley@metabolab.unc.edu  
<http://metabolab.unc.edu/Dave/drfun.html>

This cartoon is made available on the Internet for personal viewing only. Opinions expressed herein are solely those of the author.

