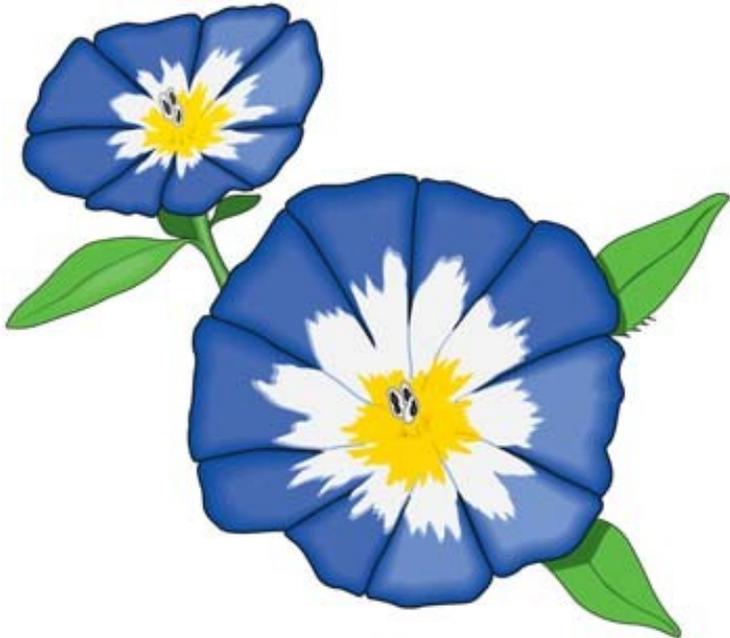


Python: Programación de Verdad



Jorge Luis Hernández C
@lesthack

lesthack@gmail.com
<http://lesthack.com.mx>

Python: Programación para Machos

Jorge Luis Hernández C
@lesthack

lesthack@gmail.com
<http://lesthack.com.mx>



Mitos sobre la Programación:

- ***La programación es una de las habilidades técnicas más difíciles.***
 - *Programar es como hablar un lenguaje nuevo, solo que mucho mas limitado y reglamentado.*
- ***Lo hacen chicos jóvenes socialmente ineptos y en total aislamiento, casi siempre de noche***

¿Quien se identifico...?



Mitos sobre la Programación:

- ***La programación es algo que no ayuda a la gente.***
 - *Vale, puedes resolver modelos matemáticos a mano si tu quieres.*
- ***Los lenguajes interpretados no son eficientes***
 - La eficiencia la de el programador.

Que SI es y que NO es esta platica

- Que no es:
 - No es un Curso rápido de Python.
 - Ni mucho menos un taller de este.
 - No es una excusa para que faltes a tus clases.
- Que si es:
 - Una oportunidad de aprender un lenguaje nuevo.
 - Una simple charla de cuates que les relata mi experiencia con python.
 - Una posible inspiración de aprendizaje y emprendimiento en proyectos nuevos.

¿ Por que es interesante aprender Python ?

- Es un lenguaje muy sencillo.
- Quieres facilitarte la vida.
- Por que se lleva bien con todos.
- Por que quieres impresionar al profesor.
- Por que cuando quieras pasarle la tarea a alguien, este no le entienda absolutamente nada.

Ya enserio, ¿de que va esto?

- Lenguaje Maduro (~19 años)
 - Diseño elegante y robusto
 - Evolucionaria
- Fácil de Aprender
 - Se lee como pseudo-código
 - Sintaxis legible y expresiva

Ya enserio, ¿de que va esto?

- Portable y Multiplataforma
 - Windows
 - Mac
 - Unix
 - Linux

Características Sustanciales

- Multiparadigma
 - Objetos
 - Procedural
 - Modular
 - Aspectos

Características Sustanciales

- Tipado Dinámico y de Conteo de Referencias
- Interpretado
- Interactivo
- Es Software Libre y Open Source
- Viene con las baterías incluidas

Propiedades

- Compilación a Bytecode
 - Implícita
 - Automática
- Tipos de datos de alto nivel
 - Numéricos sin límites
- ¿ Arreglos ? Naaah !!
 - Listas
 - Diccionarios
 - Tuplas
 - Conjuntos

¿De quién fue la idea?

- Python fue publicado en 1991 por Guido Van Rossum.
 - Conocido como el **Benevolente Dictador Vitalicio**
 - Actualmente la versión estable es 2.7
 - Reciente publicación de la versión 3.2
 - Publicada bajo la licencia Python Software Foundation License compatible con GPL.

facebook



Name:

Guido van Rossum

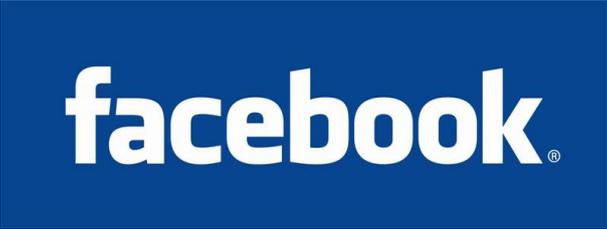
Email:

guido@python.org

Status:

Blog: Python in the Scientific World --..

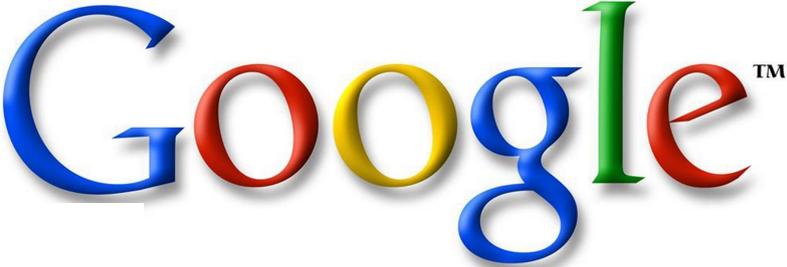
Pero pues, ¿Quién usa Python?



facebook®



YAHOO!®



Google™



DREAMWORKS
ANIMATION SKG



CANONICAL



NASA

Ok.

pero queremos ver
acción

El típico ejemplo baboso de “Hola mundo”

Python:

```
print “hola mundo”
```

Java:

```
public class Main{  
    public static void main(String args[]){  
        System.out.println("Hola mundo");  
    }  
}
```

Aprendiendo Python Básico de jalón

- Entradas/Salidas
- Tipos Datos
- Estructuras
- Funciones

Aprendiendo Python Básico de jalón

- Entradas/Salidas

```
#!/usr/bin/env python
```

```
entrada = raw_input("Dame un número: ")  
print "Tu número es el: %d " % entrada
```

Aprendiendo Python Básico de jalón

- Tipos de Datos
 - Recuerda, su tipado es dinámico

```
numero = 10  
print numero, type(numero)
```

```
numero = "10"  
print numero, type(numero)
```

```
cadena = 'El codigo ascii de la F es: %d' % ord("F")  
print cadena, type(cadena)
```

```
flotante = 12312.12  
print flotante, type(flotante)
```

```
10 <type 'int'>  
10 <type 'str'>  
El codigo ascii de la F es: 70 <type 'str'>  
12312.12 <type 'float'>
```

Aprendiendo Python Básico de jalón

- Tipos de Datos
 - Recuerda, todo es un objeto

```
print "cadena separada por espacios".split(" ")
```

```
['cadena', 'separada', 'por', 'espacios']
```

```
tupla = (1,2,5,9)
```

```
print tupla
```

```
(1, 2, 5, 9)
```

```
diccionario = {"Manzana":"Roja",  
               "Limon":"Verde",  
               "Mango":"Amarillo"}
```

```
print diccionario["Manzana"]
```

```
del diccionario["Manzana"]
```

```
print diccionario
```

```
Roja
```

```
{'Limon': 'Verde', 'Mango': 'Amarillo'}
```

Aprendiendo Python Básico de jalón

- Tipos de Datos
 - Recuerda, todo es un objeto

```
lista = ["debian","ubuntu","fedora"]
print lista
lista.append("archlinux")
print lista
lista.remove("ubuntu")
print lista

print lista.pop()

print lista
```

['debian', 'ubuntu', 'fedora']

['debian', 'ubuntu', 'fedora', 'archlinux']

['debian', 'fedora', 'archlinux']

archlinux

['debian', 'fedora']

Aprendiendo Python Básico de jalón

- Las estructuras y la elegancia de la indentación

If
else

```
edad = 18
if edad >= 18:
    print "Usted es mayor de edad"
else:
    print "Usted no es mayor de edad"
```

Apreniendo Python Básico de jalón

- Las estructuras y la elegancia de la indentación

for

```
for i in range(65,91):  
    print chr(i),
```

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Aprendiendo Python Básico de jalón

- Las estructuras y la elegancia de la indentación

while

```
q = "Aceptas que el hombre es mejor que la mujer (Si/No) ? "  
  
while( raw_input(q) != "Si" ):  
    print "Vamos, intentalo de nuevo."  
  
print "Excelente, vas por buen camino :D"
```

Aprendiendo Python Básico de jalón

- Funciones

```
def esPalindroma(cadena):  
    cadena = cadena.replace(" ", "")  
    if cadena == cadena[::-1]:  
        return True  
    return False  
  
print esPalindroma("anita lava la tina de su mama")
```

True

Probando Limites

```
import sys
```

```
n = sys.maxint
```

```
print n
```

```
print n+1
```

```
print n+100
```

```
print n*1000
```

```
2147483647
```

```
2147483648
```

```
2147483747
```

```
2147483647000
```

```
public class Main{  
    public static void main(String args[]){  
        int n = Integer.MAX_VALUE;
```

```
        System.out.println(n);
```

```
        System.out.println(n+1);
```

```
        System.out.println(n+100);
```

```
        System.out.println(n*1000);
```

```
    }
```

```
}
```

```
2147483647
```

```
-2147483648
```

```
-2147483549
```

```
-1000
```

Zen de Python

Tim Peters

- Hermoso es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Complejo es mejor que complicado.
- Plano es mejor que anidado.
- Disperso es mejor que denso.

Zen de Python

Tim Peters

- La legibilidad cuenta.
- Los casos especiales no son suficientemente especiales como para romper las reglas.
- Aunque lo pragmático gana a la pureza.
- Los errores nunca deberían dejarse pasar silenciosamente.
- A menos que se silencien explícitamente.

```
>> import this
```

La legibilidad Cuenta

- Los programas deben escribirse para que los lean las personas, y sólo de forma circunstancial para que los ejecuten las máquinas.

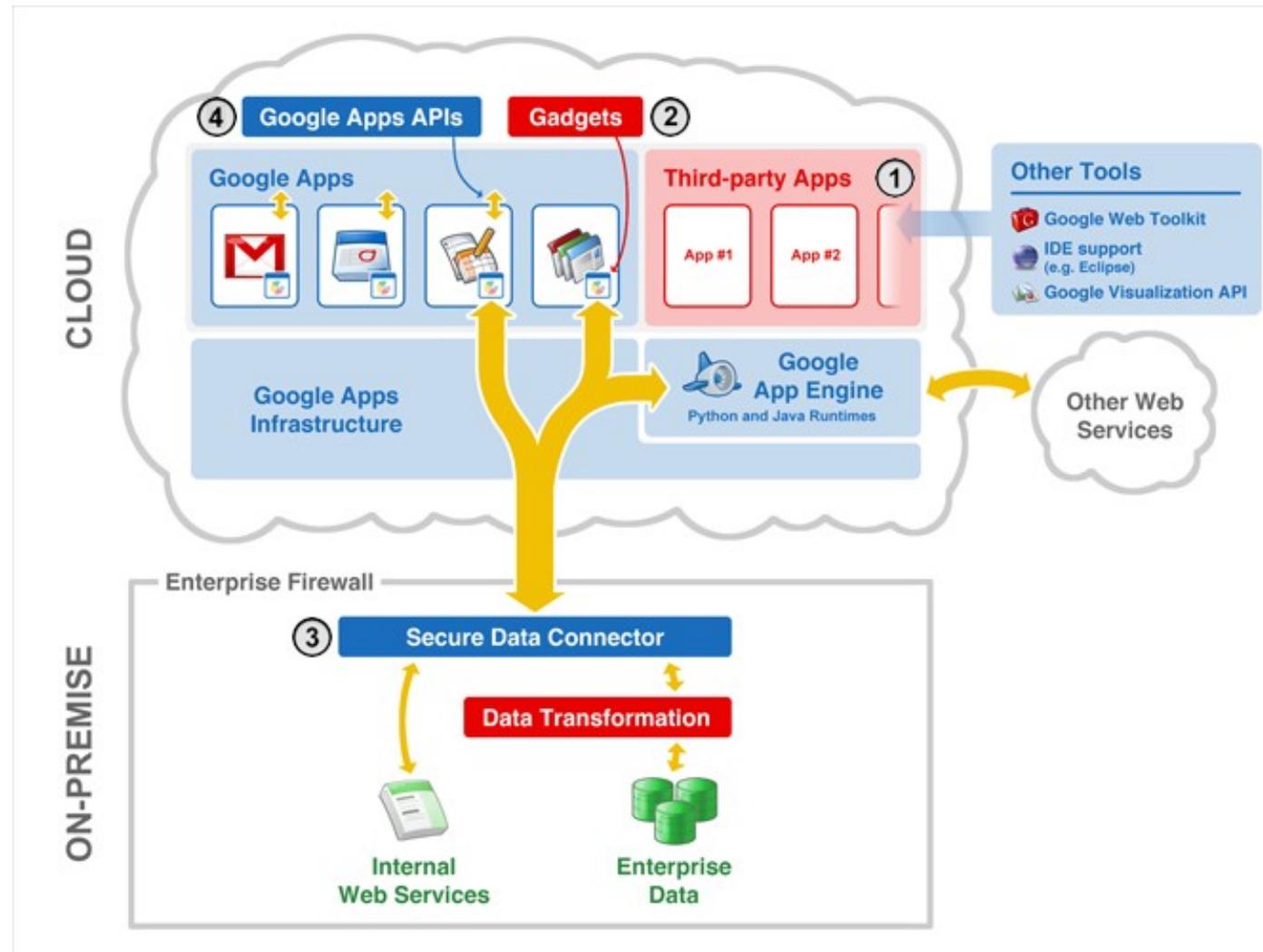
Abelson y Sussman, Estructura e Interpretación de Programas de Computadora

¿Como y donde utilizarlo?

- Desktop:
 - Soporte librerías gráficas
 - Gtk (+1, +2, +3)
 - Qt
 - WX
 - Win32
 - Database
- Games:
 - PyGame
- Web:
 - Webpy
 - Zope
 - Plone
 - Django
- Otras
 - PIL (procesamiento imagenes)
 - Sympy (matemática simbólica)
 - Numpy (matemática alto performance)

Python y Google en la Web

Google
App Engine



Un ejemplo con PyGame

```
import sys, os, pygame, time
pygame.init()
```

```
size = width, height = 320, 240
speed = [1, 1]
black = 0, 0, 0
```

```
screen = pygame.display.set_mode(size)
```

```
ball = pygame.image.load("ball.png")
ballrect = ball.get_rect()
```

```
while 1:
    time.sleep(0.01)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
    ballrect = ballrect.move(speed)
    if ballrect.left < 0 or ballrect.right > width:
        speed[0] = -speed[0]
    if ballrect.top < 0 or ballrect.bottom > height:
        speed[1] = -speed[1]
    screen.fill(black)
    screen.blit(ball, ballrect)
    pygame.display.flip()
```



Scanner de Puertos

```
import nmap

nm = nmap.PortScanner()

nm.scan(hosts='192.168.1.0/24',arguments='-p 22-443')

for host in nm.all_hosts():
    if nm[host].state() == "up":
        print "Ip:", host
        print "\tName:", nm[host].hostname()
        for protocol in nm[host].all_protocols():
            ports = nm[host][protocol].keys()
            for port in ports:
                print "\t\t", port, nm[host][protocol][port]['state']
```

Ip: 172.20.100.52

Name:

Ip: 172.20.100.1	384 filtered
Name:	133 filtered
Ip: 172.20.100.2	392 filtered
Name:	402 filtered
Ip: 172.20.100.253	152 filtered
Name:	281 filtered
Ip: 172.20.100.254	154 filtered
Name:	288 filtered
23 open	417 filtered
Ip: 172.20.100.41	418 filtered
Name:	167 filtered
80 open	431 filtered
111 open	306 filtered
Ip: 172.20.100.43	314 filtered
Name:	65 filtered
80 open	78 filtered
22 open	120 filtered
	416 filtered
	229 filtered
	112 filtered
	372 filtered
	376 filtered
	255 filtered

Ip: 172.20.100.57

Name:

Gracias por su atención

Jorge Luis Hernández C
@lesthack

lesthack@gmail.com
<http://lesthack.com.mx>

Debian México
<http://blog.debian.mx>

Planeta Python
<http://pythonmexico.org/>