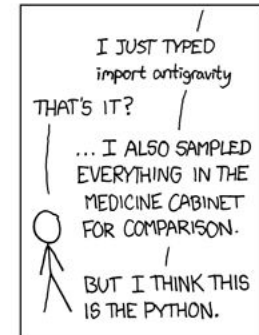
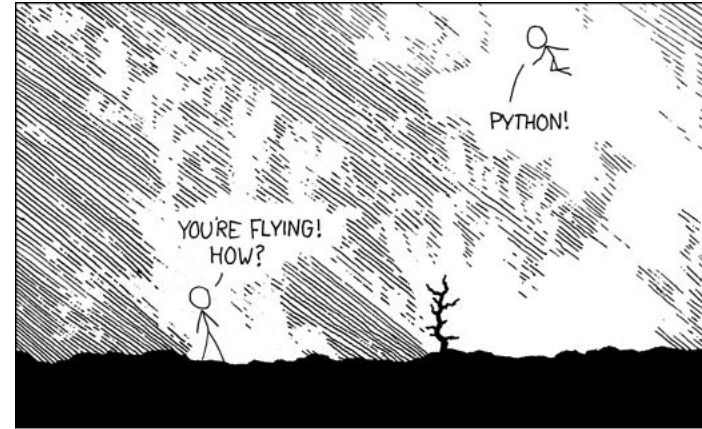




Presentado por Cusol

Características

- Lenguaje de alto nivel
- Multiparadigma
 - ◆ POO
 - ◆ Programación Imperativa
 - ◆ Programación Funcional
- Multiplataforma
- Dinámicamente tipado
- Interpretado



Un poco de Historia

Presentado en 1991 por **Guido Van Rossum**, en honor a Monty Python.

Liberado con Python Software Foundation License, compatible con la Licencia pública general de GNU a partir de la versión 2.1.1

¡Es de todos y para todos!

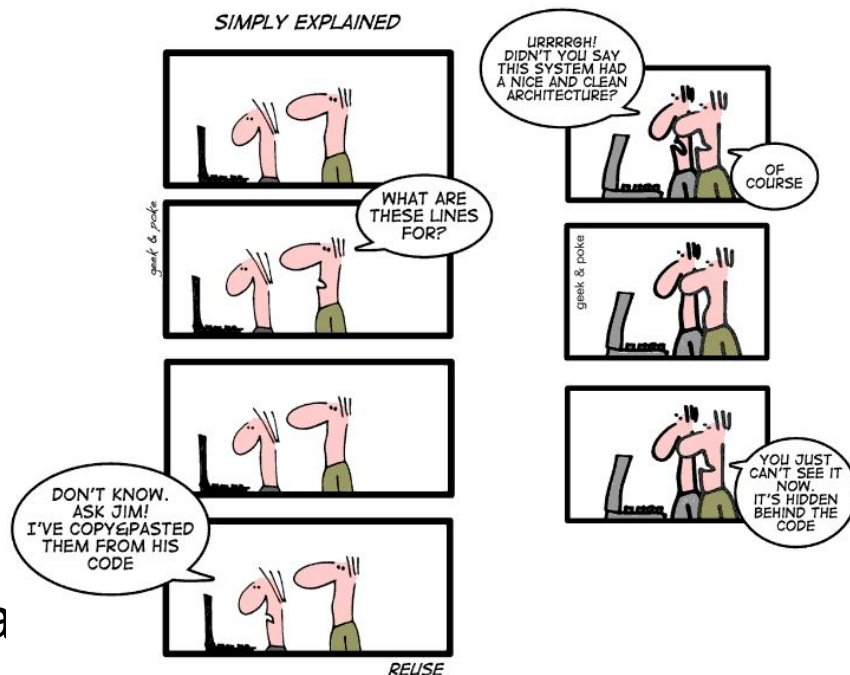


...Y filosofía

La filosofía de python es:

- Bello es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo..

Python enfatiza la importancia de la productividad y legibilidad del código.



¿Para qué sirve?

- Scripting
- Programación en la web (Django)
- Aplicaciones stand-alone

No es bueno para...

Programación de bajo nivel

Usan Python...

NETFLIX



Google



django



Instagram

You Tube

Tipos de Datos y Operadores

Entero: a = 2

Float: b = 2.0

Cadenas: s = "Hola Mundo!"

Boolean: vacio = False

Listas: lista = [12, "Camila", False]

Tuplas: (1, 2, "w", ":)")

Diccionarios:

dias = {"Lunes": 1, "Martes": 2}

Lógicos

and, or, not

Aritméticos

+, -, *, /, //, %, **

De comparación

==, <, >, <=, >=, !=

Es dinámicamente tipado

Estructuras

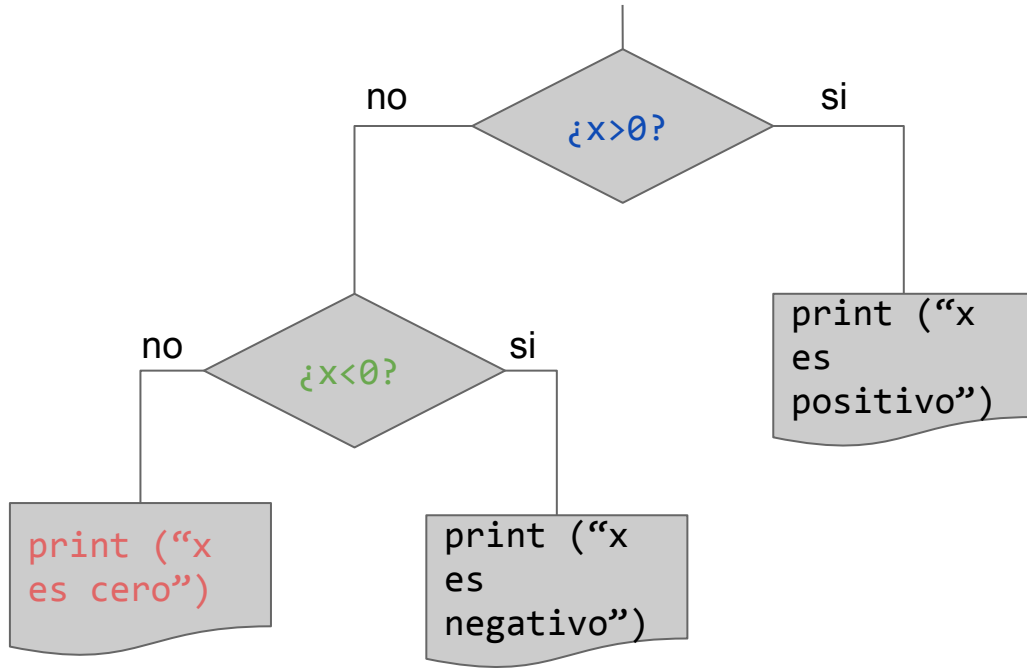
- De decisión

```
if condicion1:
    instruccion1
elif condicion2:
    instruccion2
...
elif condicionN:
    instruccionN
else:
    instruccion
```

Ejemplo:

```
if x > 0:
    print "Positivo"
elif x < 0:
    print "Negativo"
else:
    print "Zero"
```


Ejemplo:



```
if x > 0:
    print "Positivo"
elif x < 0:
    print "Negativo"
else:
    print "Zero"
```

Estructuras: For y While

- **For**

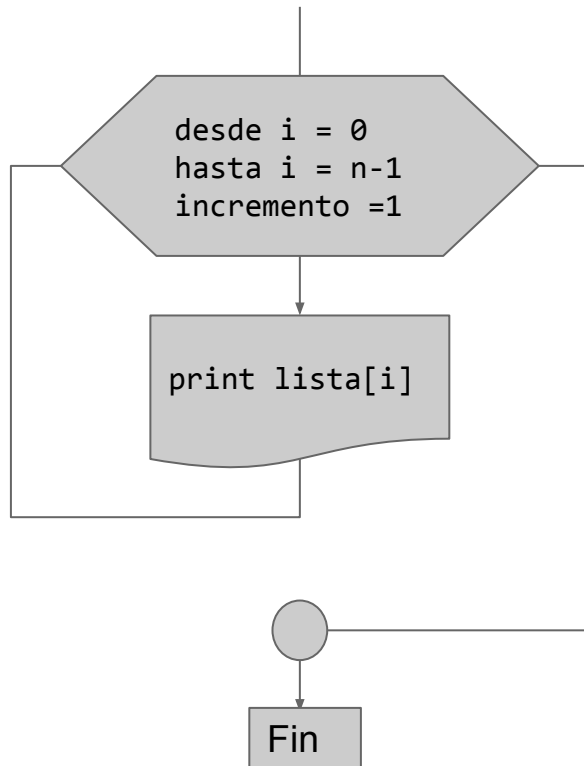
```
lista = [2, 3, 1, 4, 67, 21, 42, 3]  
n = len(lista)
```

```
for x in lista:  
    print x
```

```
for i in range(n):  
    print lista[i]
```

Puedes acceder a caracteres de cadenas por índices

```
a = "Hello world!"  
for x in a:  
    print a
```

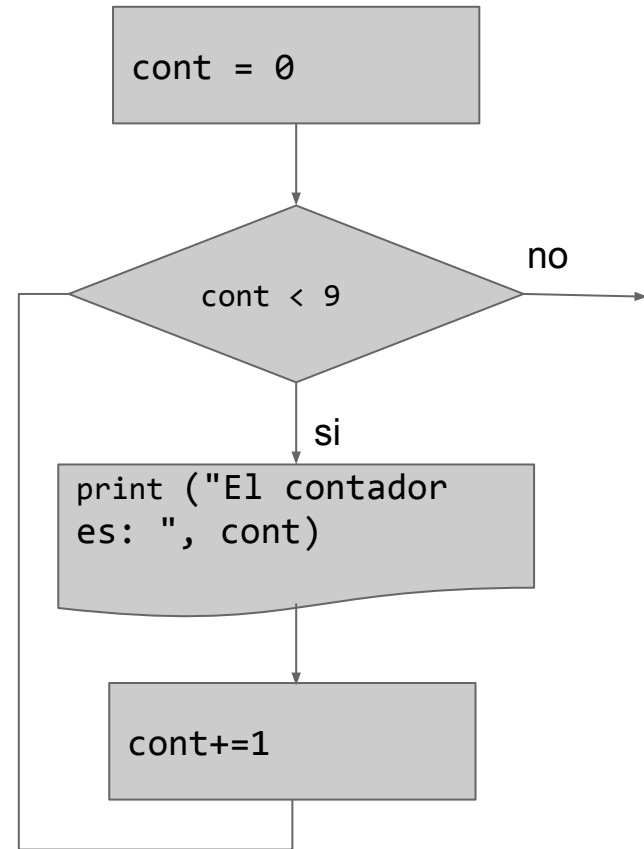


- **While**

`while` condición:
 instrucciones

Ejemplo:

```
cont = 0
while (cont < 9):
    print("El contador es: ", cont)
    cont += 1
# esto es igual a cont = cont + 1
```



Algunas funciones básicas

`print()`

`input()`

`f = open("archivo.txt", opcion)`

opciones: `w, r, a, r+`

`f.read()`

`f.write()`

`list.append()`

`len()`

`string.format()`

`type()`

`int()`

`float()`

`str()`

Importando módulos

...evita reinventar la rueda, los módulos o librerías de otros nos ahorran tiempo.

Usaremos **math** como ejemplo, este módulo tiene funciones matemáticas para números reales.

¿Cómo lo uso?

Al comienzo de tu código.

```
import math  
math.sqrt(2)
```

```
from math import *    → Evita esto!  
sqrt(2)
```

Si sólo necesitas una función en específico importa sólo esa:

```
from math import sqrt  
sqrt(2)
```

Crea tus funciones!

Las funciones nos permiten guardar código para reusarlo.

Ejemplo:

Con `def` “defines” una función

Una función puede recibir parámetros (en este caso x)

```
def imprimir(x):  
    print(x)
```

Llamando la función
`imprimir(“Hola mundo!”)`
Resultado: Hola mundo!

```
def cuadrado(x):  
    return x**2
```

Dónde aprender más?

Libros

Think Python. Allen B. Downey

Python para Todos. Raúl González Duque

En la Web

<https://www.python.org/> Documentación

<http://www.codecademy.com> Curso interactivo

<http://www.reddit.com/r/Python/>

Taller

Crear una función `get_addres` que tome una dirección y extraiga los números de coordenada (calle, carrera).

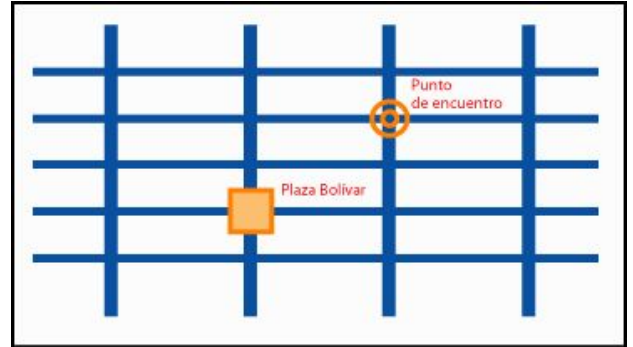
Resultado esperado: La dirección será una cadena con el formato “Calle 56 # 20-22” De la cual sólo nos interesa 56 y 20. El resultado deberá ser una lista [“56”, “22”]

Ayuda: Usa la función `split()` de python que devuelve una lista con las palabras separadas según el caracter que le pongas.

```
a= "Hello world"
```

```
a.split() Resultado: ['hello', 'world']
```

```
a.split("o") Resultado: ['hell', ' w', 'rld']
```



Taller

Crear una función `get_distance` que tome dos direcciones y calcule la distancia mínima entre estas, usa la fórmula de la distancia euclidiana.

Ayuda:

- 1) Puedes usar la función anteriormente creada para encontrar las coordenadas!
- 2) para transformar una cadena en número usa `int()`. Ej: `int("2")`

Taller: Usando NumPy y Matplotlib

Veamos cómo podemos usar Python para computación científica con diversas librerías.

```
#Importamos Librerías##
import numpy as np
import matplotlib.pyplot as plt

#Creamos la nube de puntos
x = np.array([0, 1, 2, 3])
y = np.array([1, 1.5, 1.8, 2.6])

#Construimos una matriz
A = np.vstack([x, np.ones(len(x))])
A = A.T #Hacemos su traspuesta
```

```
#Obtenemos parametros de la recta
m, c = np.linalg.lstsq(A, y)[0]
print ("La recta obtenida es:  y = " + str(m) + "x + " + str(c))
```

```
plt.plot(x, y, 'o', label='Nube puntos', markersize=15)
plt.plot(x, m*x + c, 'r', label='Regresion Lineal')
plt.legend()
plt.show()
```

```
raw_input("Enter para salir")
```