



Python para el cálculo científico

Daniel Lubián Arenillas

12 de febrero de 2018



Presentando Python

Librerías importantes

Programación orientada a objetos

Sintaxis básica de Python

Sentencias de control

Numpy: el `ndarray`

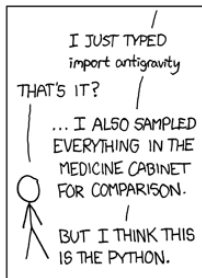
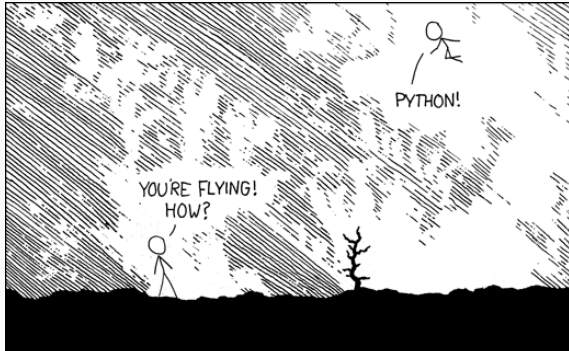
Scipy

Matplotlib.pyplot

Presentando Python



- Creado por Guido van Rossum en 1991.
- Lenguaje de propósito general:
 - Cálculo científico
 - Desarrollo web
 - Administración de sistemas
 - GUIs
 - Inteligencia artificial
 - Todo es posible
- Lenguaje multiparadigma, permite programación estructurada, orientada a objetos, funcional,...
- Lenguaje interpretado, no compilado.





- Dos versiones: 2.7 y **3.6**
- Libre, abierto y gratuito, con una comunidad enorme → todo a golpe de Google.
- Mil y una librerías abiertas y gratuitas.
- Rápido y fácil de escribir, puede ser lento de ejecutar.

Para escribirlo: Spyder, cuadernos Jupyter, Pycharm, VS Code, Atom, Geany, Notepad++...



IP[y]:
IPython

Numpy: cálculo numérico

Scipy: cálculo científico

Matplotlib: graficado

Pandas: análisis de datos

IPython: consola interactiva



Un **objeto** tiene **métodos** (“funciones”) y **atributos** (“variables”) que lo constituyen.

- `A.shape`
- `z.conjugate()`
- `v.reshape((3,4))`

Todo Python trabaja con objetos¹

¹(hasta donde yo sé)

Sintaxis básica de Python



Entero	<code>integer</code>	<code>1</code>
Con coma flotante	<code>float</code>	<code>1.</code>
Complejo	<code>complex</code>	<code>1. + 2j</code>
Booleano	<code>boolean</code>	<code>True</code>



Cadenas $\rightarrow s = \text{"perro"}$

Listas $\rightarrow l = [1, \text{'perro'}, \text{True}]$

Tuplas $\rightarrow t = (1, \text{'perro'}, \text{True})$

Diccionarios $\rightarrow d = \{ \text{'1': 2, 'el': 'perro'} \}$

Ojo: los índices van de 0 a $n-1$, como en C:

- $s[0] = \text{'p'}$
- $l[-1] = \text{True}$
- $t[3]$ no existe
- $d[\text{'el'}] = \text{'dog'}$



Condición: if-elif-else

```
1 if 1 in x:  
2     haz esto  
3 elif 2 < 7 and 2 > 1:  
4     haz esto otro  
5 else:  
6     o esto
```



Bucle: for

```
1 x = [2, "muse"]
2 n = 2
3
4 for i in range(0, n):
5     for x_i in x:
6         print(i, x_i)
7
8 # SALIDA:
9 # 0 2
10 # 0 muse
11 # 1 2
12 # 1 muse
```



List comprehensions

```
1 x = [i**2 for i in range(6)]  
2 print(x)  
3  
4 ## SALIDA  
5 # [0, 1, 4, 9, 16, 25]
```

Numpy: el ndarray

Scipy

Matplotlib.pyplot
