

Seminario 1: Ciencia de Datos con Python

Daniel Molina Cabrera

Inteligencia de Negocio, UGR



¿Qué es Python?



¿Qué es Python?



¿Qué es Python?



Es un lenguaje de programación

- Es un lenguaje para programar todo tipo de aplicaciones.
- Se diseñó para que fuese **fácil de usar** y **divertido**.

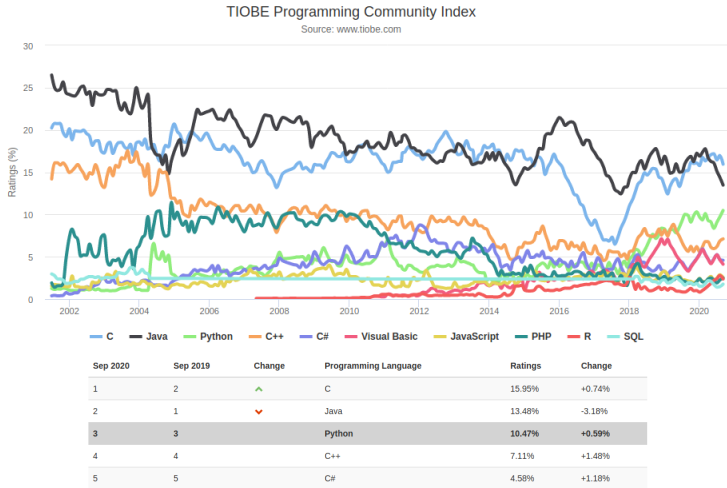
¿Quién lo inventó?

python



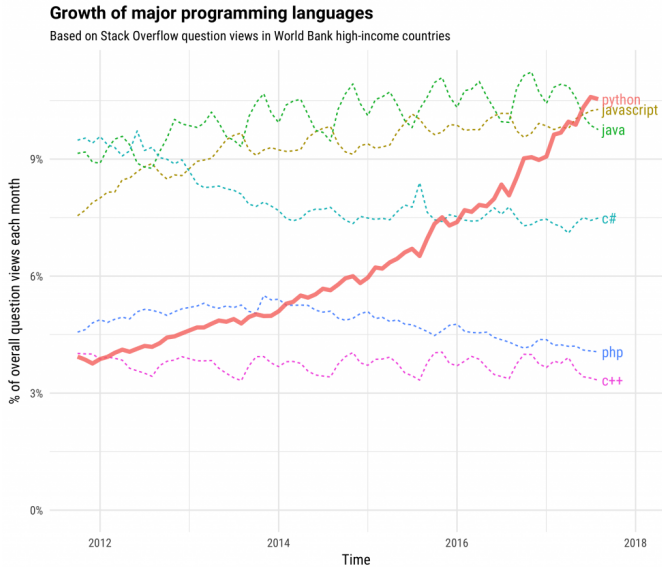
- Inventado por Guido Van Rossum.
- Creado en 1989 en vacaciones de navidad.
- Pensado para enseñar programación a niños.
- Muy bien aceptado por la comunidad.
- No dependiente del autor: dilema del autobús.
- Influyente: Ruby, ...

¿Por qué Python? Es popular



Es popular

Tendencia



Ciencias de Datos con Python



Muchas librerías científicas en Python, lenguaje de la ciencia.

Machine Learning y Deep Learning con Python

Librería más popular de Machine Learning



Librerías más populares de Deep Learning



PyTorch



TensorFlow

Es muy sencillo

C/C++

```
#include <iostream>

int main(void) {
    std::cout <<"Hola a todos desde C++" <<std::endl;
}
```

Java

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

Python

```
print("Hola a todos desde Python\n")
```

Es legible

Uso natural del listas (y diccionarios)

```
list = ["fruta", "cereales", "berenjena"]  
  
for item in list:  
    print(item)
```

Ejemplo: Implementar programa grep

```
from sys import argv  
  
def main(fname, word):  
    with open(fname, "r") as file:  
  
        for line in file:  
            if word in line:  
                print(line)
```

¿Qué necesito?



Requisitos

- Un ordenador.
- Da igual el SO: Windows, Linux, MacOS.
- Instalaremos Python3 usando Anaconda (opcional si ya está instalado).

Hacerlo en casa

Ocupa espacio, copiarlo/instalarlo con tiempo.

Instalando Python



Instalación

Disponible en

<https://docs.anaconda.com/anaconda/install/>

Descargar la versión adecuada

Anaconda Installers

Windows 🪟

Python 3.8

64-Bit Graphical Installer (466 MB)

32-Bit Graphical Installer (397 MB)

MacOS 🍏

Python 3.8

64-Bit Graphical Installer (462 MB)

64-Bit Command Line Installer (454 MB)

Linux 🐧

Python 3.8

64-Bit (x86) Installer (350 MB)

64-Bit (Power8 and Power9) Installer (290 MB)

Instalando en Windows

Anaconda3 5.1.0 (64-bit) Setup



Choose Install Location

Choose the folder in which to install Anaconda3 5.1.0 (64-bit).

Setup will install Anaconda3 5.1.0 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Users\anaconda\Anaconda3

Browse...

Space required: 2.5GB

Space available: 13.6GB

Anaconda, Inc.

< Back

Next >

Cancel

Instalando en Windows

Anaconda3 5.1.0 (64-bit) Setup



ANACONDA

Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

☐ Add Anaconda to my PATH environment variable

Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☒ Register Anaconda as my default Python 3.6

This will allow other programs, such as Python Tools for Visual Studio, PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.6 on the system.

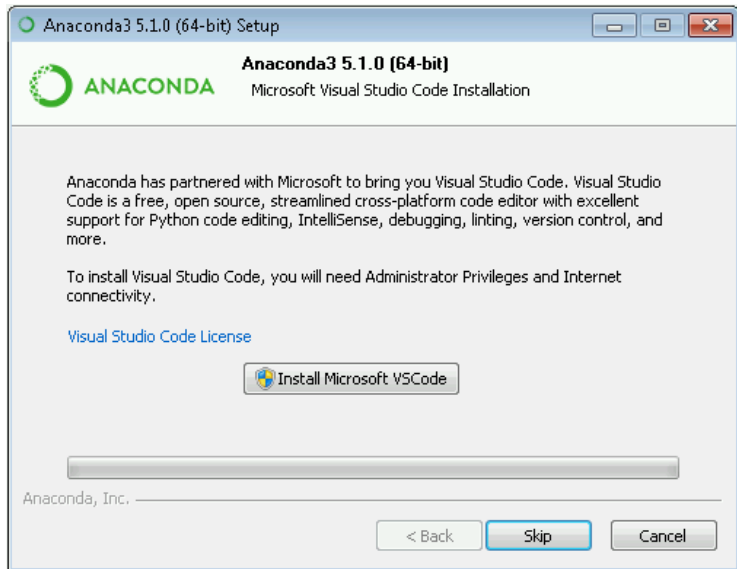
Anaconda, Inc.

< Back

Install

Cancel

Instalando en Windows



Instalando en Windows

Anaconda3 5.1.0 (64-bit) Setup



Thanks for installing Anaconda3!

Anaconda is the most popular Python data science platform.

Share your notebooks, packages, projects and environments on Anaconda Cloud!

☒ Learn more about Anaconda Cloud

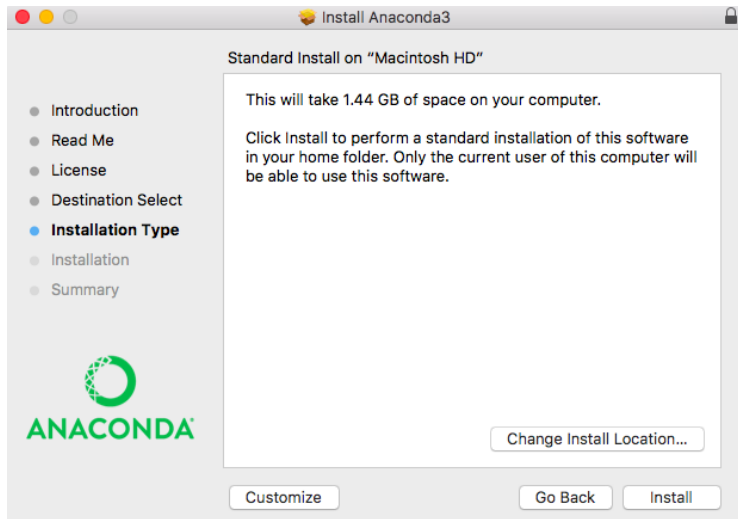
☒ Learn how to get started with Anaconda

< Back

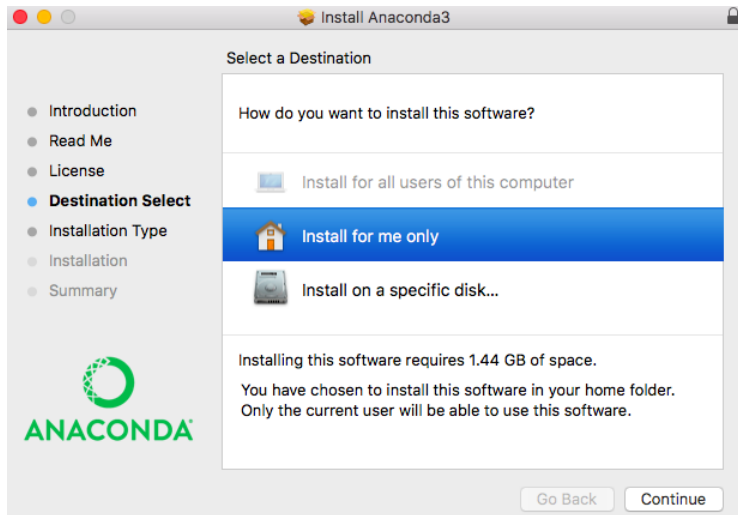
Finish

Cancel

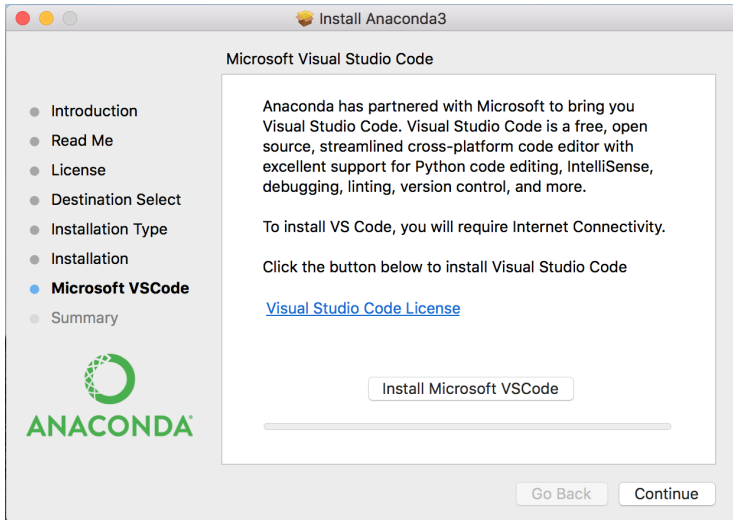
Instalando en MacOS



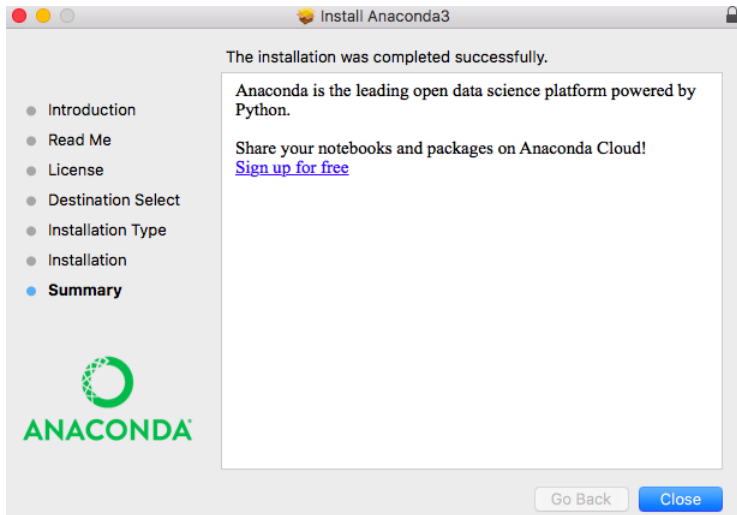
Instalando en MacOS



Instalando en MacOS



Instalando en MacOS



Instalando en Linux

Usando Anaconda

```
bash ~/Downloads/Anaconda3-5.1.0-Linux-x86_64.sh
```

Desde el sistema de paquetes

```
sudo apt install python3  
python3 -m pip install --upgrade pip  
python3 -m pip install jupyter
```

Entornos

Formato interactivo

`python` Línea de forma interactiva.

`ipython/jupyter` interfaz con *esteroides* (autocompletado, ...).

`ipython/jupyter notebook` Interfaz web.

Notebook

- Entorno desde el navegador.
- Fácil para pruebas rápidas (usaremos los primeros días).
- Formato de ficheros `.ipyb` aceptado por Github.

Editores Específicos de Python

`Tonny` Editor para aprendizaje.

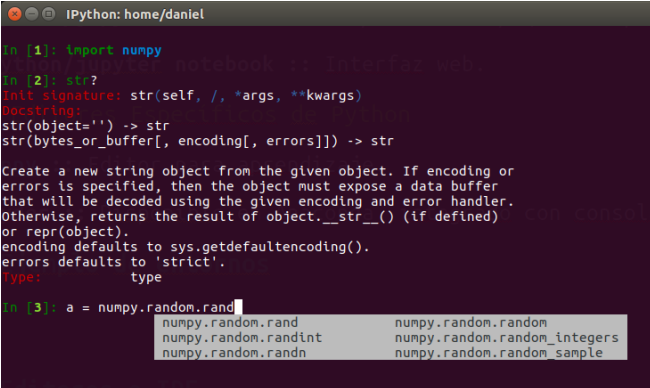
`Spyder` Disponible en Anaconda, integrado con consola.

Ejemplo de entornos (Python por defecto)

```
daniel@ubuntu:~/Descargas/opt/screen$ python3
Python 3.6.1 [Anaconda custom (64-bit)] (default, May 11 2017, 13:09:58)
[CC 4.4.7 201311 (Red Hat 4.4.7-1)] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import numpy
>>> a = numpy.random.rand(30)
>>> for x in a:
...     print(x)
...
0.499233852424
0.960378153162
0.17979869278
0.421647289513
0.869444245156
0.073945665169
0.788487696445
0.797646443367
0.101940844059
0.474467458971
0.417601137896
0.40539583659
0.285597840242
0.454929511648
0.151935443775
0.259959550369
0.692344940073
0.384080715125
0.882240175938
0.702248028962
0.0994603536405
0.330414214792
0.262422047151
0.859998337479
0.811760093835
0.170895118696
0.551435218441
0.346895972828
0.441369069151
0.341333841852
>>> print(a)
[ 0.49923385  0.96037815  0.17979869  0.4216473   0.86944425  0.07394567
  0.7884877   0.79764644  0.10194684  0.47446746  0.41760114  0.40539584
  0.28559784  0.45492951  0.15193544  0.25995955  0.69234494  0.38408072
  0.88224018  0.70224803  0.09946035  0.33041421  0.26242205  0.85999834
  0.81176009  0.17089512  0.55143522  0.34689597  0.44136907  0.34133384]
```

Figure: consola por defecto de python

Ejemplo de entornos (IPython/Jupyter)



```
IPython: home/daniel

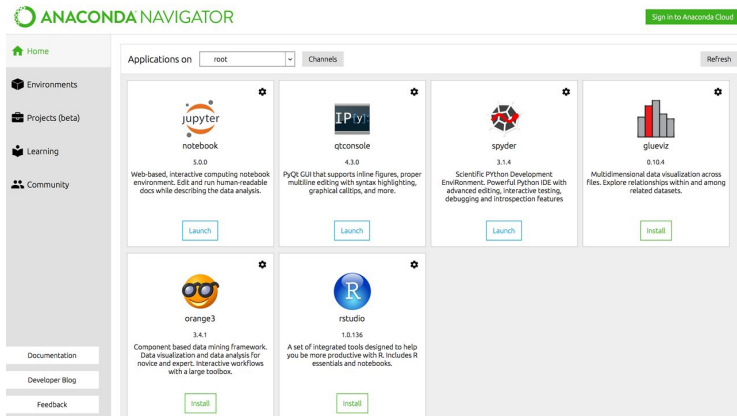
In [1]: import numpy
In [2]: str?
Init signature: str(self, /, *args, **kwargs)
Docstring:
str(object='') -> str
str(bytes_or_buffer[, encoding[, errors]]) -> str

Create a new string object from the given object. If encoding or
errors is specified, then the object must expose a data buffer
that will be decoded using the given encoding and error handler.
Otherwise, returns the result of object.__str__() (if defined)
or repr(object).
encoding defaults to sys.getdefaultencoding().
errors defaults to 'strict'.
Type:
    type

In [3]: a = numpy.random.rand
         numpy.random.rand      numpy.random.random
         numpy.random.randint   numpy.random.random_integers
         numpy.random.randn     numpy.random.random_sample
```

Figure: consola de ipython/jupyter

Ejemplo de entornos (IPython/Jupyter notebook)



Ejemplo de entornos (IPython/Jupyter notebook)

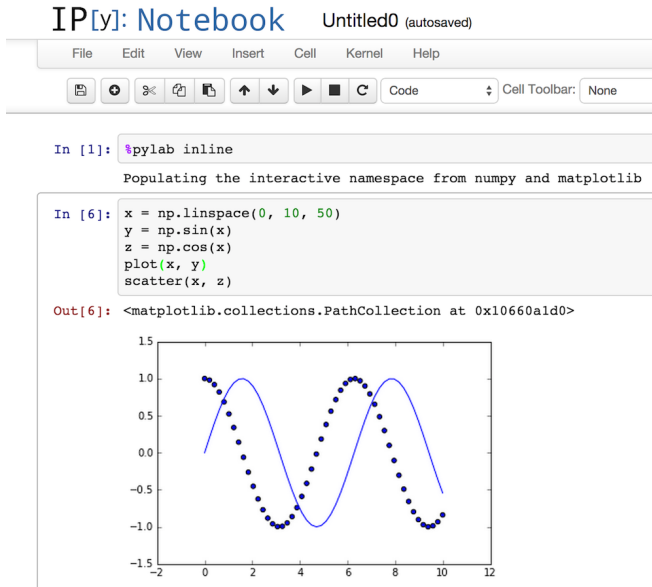
Command Prompt - jupyter notebook

```
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Ajit>jupyter notebook
[I 16:35:15.317 NotebookApp] writing notebook server cookie secret to C:\Users\Ajit\AppData\Local\Temp\jupyter_notebook_cookie_secret
[I 16:35:15.092 NotebookApp] JupyterLab alpha preview extension loaded from C:\Users\Ajit\AppData\Local\Temp\jupyterlab
JupyterLab v0.27.0
Known labextensions:
[I 16:35:15.099 NotebookApp] Running the core application with no additional extensions
[I 16:35:15.208 NotebookApp] Serving notebooks from local directory: C:\Users\Ajit\AppData\Local\Temp\jupyterlab
[I 16:35:15.208 NotebookApp] 0 active kernels
[I 16:35:15.209 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=499305c7f8f6aec5a564aadd0be303f39df0edd740ffb761
[I 16:35:15.209 NotebookApp] Use Control-C to stop this server and shut down all kernels
[C 16:35:15.216 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
    http://localhost:8888/?token=499305c7f8f6aec5a564aadd0be303f39df0edd740ffb761
[I 16:35:15.658 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

Ejemplo de entornos (IPython/Jupyter notebook)



Uso de Notebook

Dividido en celdas

Trozo de código que se puede ejecutar, y muestra el resultado.

Interactivo

Muy útil para explorar datos, resultados, ...

Permite documentar

Existen celdas notebook para explicar los pasos.

Uso de Notebook

Dividido en celdas

Trozo de código que se puede ejecutar, y muestra el resultado.

Interactivo

Muy útil para explorar datos, resultados, ...

Permite documentar

Existen celdas notebook para explicar los pasos.

Muy recomendable para este tipo de uso

¿Y ahora qué?

- 1 Instalar el software.
- 2 Activar Notebook.
- 3 Cargar notebook del tutorial.
- 4 Ir probando.
- 5 Hacer los ejercicios pedidos.

Aprendizaje

¿Y ahora qué?

- 1 Instalar el software.
- 2 Activar Notebook.
- 3 Cargar notebook del tutorial.
- 4 Ir probando.
- 5 Hacer los ejercicios pedidos.

Pero antes dar unos pocos consejos (de mi experiencia) que sorprenden un poco.

Sobre sintaxis

Formato

Los bloques empiezan con “:” y luego se tabulan, terminan cuando termina dicha tabulación

```
def main(value):  
    if value < 0:  
        value *= -1  
  
    for i in range(value):  
        print(f"Iteración {i}")  
  
    return value  
  
main(-4)
```

Estilo de programación

- Puede usarse con funciones o clases (como C++).
- Por simplicidad basta con funciones, no haremos clases.
- Usaremos librerías que siguen notación OO.

```
from pathlib import Path

def print_files(dir):
    "Imprime ficheros del directorio"
    assert dir.is_dir()

    for file in dir.iterdir():
        if ".jpg" in file.name:
            print(file)

def main():
    "Programa principal"
    dir = Path(".")
    assert dir.exists()
    print_files(dir)

main()
```

Consejos de programación

Intentad usar el lenguaje de forma sencilla.

No programar estilo C++

```
lista = ["fruta", "cereales", "berenjena"]
```

```
for i in range(0, len(lista)):
    print(lista[i])
```

Consejos de programación

Intentad usar el lenguaje de forma sencilla.

No programar estilo C++

```
lista = ["fruta", "cereales", "berenjena"]
```

```
for i in range(0, len(lista)):
    print(lista[i])
```

Usar la sencillez que ofrece

```
list = ["fruta", "cereales", "berenjena"]
```

```
for item in list:
    print(item)
```

¿Y ahora qué?

- 1 Instalar el software.
- 2 Activar Notebook.
- 3 Cargar notebook del tutorial.
- 4 Ir probando.
- 5 Hacer los ejercicios pedidos.

Recursos útiles

- [Tutorial sobre Jupyter](#)
- [Dive into Python 3](#) (Excelente tutorial)
- [Think Python: How to Think Like a Computer Scientist](#) (Nivel básico, no supone ningún conocimiento de programación)