

Sesión 1

Ecosistema Python

Javier Cózar
Curso: Introducción a Python
Almería, 2 Abril 2019

























- Introducción
- Instalación y entornos
- Iniciación en Python

- Fácil, intuitivo y potente
- Agilidad en desarrollo
- Interpretado y portable
- Libre (Python Software Foundation)
- Propósito general (data science, web, ui, sysadmin, embedded)



*Guido Van Rossum
(Benevolent
Dictator for Life)*

Fuente: IEEE Spectrum

Language Rank	Types	Spectrum Ranking
1. Python	  	100.0
2. C++	  	99.7
3. Java	  	97.5
4. C	  	96.7
5. C#	  	89.4
6. PHP		84.9
7. R		82.9
8. JavaScript	 	82.6
9. Go	 	76.4
10. Assembly		74.1

PYTHON2 (2.7)



¡Deprecado (2020)!

<https://pythonclock.org>

PYTHON3 (3.5, 3.6, 3.7, 3.8)

- Comprobar
dependencias
- Comprobar
retrocompatibilidad



INSTALACIÓN STANDALONE

- <https://www.python.org>
- Soporte oficial
- Para usuarios iniciados/técnicos
- Flexible
- Instalación básica



ANACONDA

- <https://www.anaconda.com>
- Privada, distribución enterprise
- Cómoda para usuarios no técnicos
- Gestionada
- Instalación stack científico

- Lenguaje interpretado: se ejecuta en un intérprete
- Modo batch (scripts o programas)

```
python myscript.py
```

- Modo interactivo

```
Python 3.6.2 (default, Jul 17 2017, 16:44:45)  
[GCC 4.2.1 Compatible Apple LLVM 8.1.0 (clang-802.0.42)] on  
darwin  
Type "help", "copyright", "credits" or "license" for more  
information.  
>>> print("Hello")
```

Instalación e Intérprete python



- Python divide su funcionalidad en **paquetes** y **módulos**
- La instalación de python incluye funcionalidad básica
 - **Extensible** mediante la instalación de estos elementos
- **PyPi** un repositorio con más de 173.650 proyectos
- **pip** es un paquete para **gestionar** estas extensiones

<https://bootstrap.pypa.io/get-pip.py>

URL del script de descarga del paquete pip

- Es un paquete pensado para ejecutarse desde el terminal, no desde el intérprete de python
- Para ejecutar paquetes, se usa `python -m paquete <argumentos>`

Para instalar un paquete

```
python3 -m pip install requests
```

```
python3 -m pip install requests==2.21.0
```

Para ver la lista de paquetes instalados

```
python3 -m pip list
```

Para desinstalar un paquete

```
python3 -m pip uninstall requests
```

- La instalación consiste en la **descarga** del código fuente en una **carpeta** de tu sistema
- ¿Cómo gestionamos las **versiones**? ¿Las **dependencias** de todos nuestros proyectos en **una misma carpeta**?

¿Qué versión del paquete requests instalamos?



- Los **entornos virtuales** permiten instalar las dependencias en carpetas **separadas por proyectos**

SISTEMAS UNIX

WINDOWS

Creación de un nuevo entorno virtual

```
python3 -m venv venv
```

```
python3 -m venv venv
```

Activación del entorno virtual

```
source venv/bin/activate
```

```
venv\Scripts\activate
```

Desactivar el entorno virtual

```
deactivate
```

```
deactivate
```

- Otra ventaja es la **reproducibilidad** de los entornos

Especificar las dependencias en un fichero requirements.txt

```
requests==2.21.0  
jupyter==1.0.0  
...
```

Instalar las dependencias especificadas en el fichero

```
python -m pip install -r requirements.txt
```

Crear el fichero requirements con las dependencias de tu venv

```
python -m pip freeze > requirements.txt
```


Gestión de entornos virtuales





- Entorno de ejecución **interactivo** y **visual**
- Permite combinar **código**, **documentación** y **gráficos**
- Instalación

```
pip install jupyter
```

- Ejecución (abre el navegador web por defecto)

```
python -m jupyter notebook
```


Entorno jupyter notebook

Introducción a python

