

# Introducción a Python

Motivación, características.

# Contenidos

- ♦ Introducción
- ♦ Jupyter/Anaconda
- ♦ Conceptos básicos

# Introducción

- ♦ Desarrollado por Guido van Rossum en 1991.
- ♦ Muy utilizado en la investigación y en la enseñanza.
- ♦ Hay dos versiones:
  - Python 2 en el 2000.
  - Python 3 en el 2008.
  - No son completamente compatibles entre si.
  - Los materiales están probados con Python 3.6 y 3.7

# Introducción

Python es:

- ♦ Multiparadigma: orientado a objetos, imperativo, funcional.
- ♦ Multiplataforma.
- ♦ Lenguaje interpretado.
  - Al ejecutar el comando “python” sin argumentos se lanza el intérprete. El intérprete recibe una expresión, la ejecuta y muestra el resultado. Usando como argumento el nombre de un fichero se ejecuta el script. El código fuente (scripts) en ficheros con extensión **.py**.
- ♦ Distintos entornos de desarrollo. Se usará Jupyter Notebook en las clases, (el alumno puede usar Eclipse, PyCharm, JupyterLab o lo que prefiera).

# Jupyter/Anaconda

Jupyter es un entorno de desarrollo que utiliza notebooks.

Un notebook es un proyecto donde tenemos:

- ♦ El código de nuestra aplicación
- ♦ Los resultados que produce nuestro código.
- ♦ La documentación, donde podemos explicar aspectos del código o de la aplicación con texto o imágenes.

Tienen **pros** y **contras**:

- ♦ Pros: Rápido para prototipar y explicar código.
- ♦ Contras: No es una herramienta para desarrollar grandes proyectos. Dificulta el trabajo con muchas clases, las herramientas para control de versiones, pruebas, refactors etc no están tan avanzadas. No está pensado para sustituir a Eclipse o Pycharm.

# Jupyter/Anaconda

En resumen:

- ♦ Bueno para prototipos y para explicar el código y los resultados (docencia, investigación, hobbies).
- ♦ No tan apropiado para proyectos grandes como muchas personas y muchas clases, debido a que las herramientas típicas de otros IDEs no están tan avanzadas.
- ♦ En un TFG o en el trabajo se pueden hacer prototipos con Jupyter. Pero no se debería hacer el proyecto principal.

# Jupyter/Anaconda

En Python al igual que otros lenguajes hay librerías. Estas librerías a veces hay que bajarlas e instalarlas.

Anaconda es una distribución de Python que contiene Jupyter y decenas de librerías para todo tipo de problemas.

Usar Anaconda y Jupyter:

- ♦ (Opción 1) Instalarlo.
- ♦ (Opción 2, sin instalar) <https://jupyter.org/try>
- ♦ (Opción 3, sin instalar) Google Colab

<https://colab.research.google.com/notebooks/welcome.ipynb>

# Jupyter/Anaconda

- ♦ Vamos a esta dirección  
<https://www.anaconda.com/download/>
- ♦ Descargamos la última versión para python 3 del sistema operativo que utilicemos (Windows, Linux, Mac) 64 o 32 bits según sea nuestro ordenador.
- ♦ En windows hacer doble click en el ejecutable, en linux ejecutar `bash Anaconda3-4.4.0-Linux-x86_64.sh` (o el nombre de la nueva versión)
- ♦ **Windows:** Se recomienda no alterar el PATH. Hacer “Inicio” → “Anaconda Navigator”
- ♦ **Linux/Mac:** Para modificar el PATH abrir el archivo `.bashrc` o `.bash_profile` (en la carpeta de usuario) y añadir:
  - **`export PATH="/<path to anaconda>/bin:$PATH"`.**



# Jupyter/Anaconda

- ♦ Una vez instalado, para utilizar un notebook, abrimos un terminal y escribimos:
  - **jupyter notebook**

# • Conceptos básicos

- ♦ Abrir el notebook de la primera clase, leer las explicaciones y evaluar el contenido de las celdas
  - Ejecuta Jupyter Notebook App (ver sección anterior).
  - En el Notebook Dashboard navegar hasta encontrar el fichero correspondiente y clicarlo.
    - ★ Si se usa el servicio web habría que usar “upload” para subir el fichero de .ipynb