



Curso de Python

Mentoría del Curso

- **Gabriela Arévalo (Ph.D)**
- Docencia Universitaria de 20 años de experiencia.
- Desarrollo de Investigación
- Gestión Universitaria desde 2016

Dinámica de Clases

- *Introducción al tema (vía meet)*
- *Lectura de apuntes del / de los temas con ejercicios de inducción*
- *Ejercicios a desarrollar (pueden consultar en sus grupos)*
- *Canal de dudas será Discord*
- *Comparación de Ejercicios desarrollados en su grupo*
- *Comparación de Soluciones con la de la Mentoría*
- *“Informe” escrito de las Diferencias*
- *Subir al Aula Virtual el archivo con sus soluciones y las diferencias detectadas. Los ejercicios deben poder ejecutarse.*
- *Resumen de la última media hora de clase (si es posible)*

Python como lenguaje

- Está diseñado para ser fácil de leer y simple de utilizar
- Es un lenguaje de alto nivel
- Gramática sencilla, clara y muy legible
- Tipado dinámico y fuerte
- Interpretado
- Orientado a Objetos:
 - Overloading de constructores, herencia múltiple, encapsulamiento, interfaces y polimorfismo
- Open source
- Librería estándar muy amplia
- Versátil: aplicaciones de escritorio, aplicaciones de servidor, aplicaciones web



Paradigmas de programación

- Programación Imperativa
- Programación Funcional
- Programación orientada a objetos

Tipos de Ejecución

- Interpretado en la consola de Python
- Interpretado en fichero
- Compilado a bytecode
- Compilado a ejecutable del sistema

Interpretado en la consola de Python

- Se ejecuta cada instrucción que introduce el usuario de manera interactiva.

```
> python
>>> name = "Gabi"
>>> print("Hola ", name)
Hola Gabi
```

Interpretado en Fichero (1/2)

- Se leen y se ejecutan una a una todas las instrucciones del fichero.

```
# Fichero hola.py  
name = "Alf"  
print("Hola ", name)
```

```
> python hola.py  
Hola Gabi
```


Interpretado en Fichero (2/2)

- También se puede hacer el fichero ejecutable indicando en la primera línea la ruta hasta el intérprete de Python.

```
#!/usr/bin/python3
name = "Gabi"
print("Hola", name)
```

```
> chmod +x hola.py
> ./hola.py
Hola Gabi
```

Compilado a bytecode

```
# Fichero hola.py  
name = "Gabi"  
print("Hola " + name)
```

```
> python -O -m py_compile hola.py  
> python __pycache__/hola.cpython-  
37.pyc  
Hola Gabi
```

Compilado a ejecutable del sistema

Hay distintos paquetes que permiten compilar a un ejecutable del sistema operativo usado, por ejemplo **pyinstaller**.

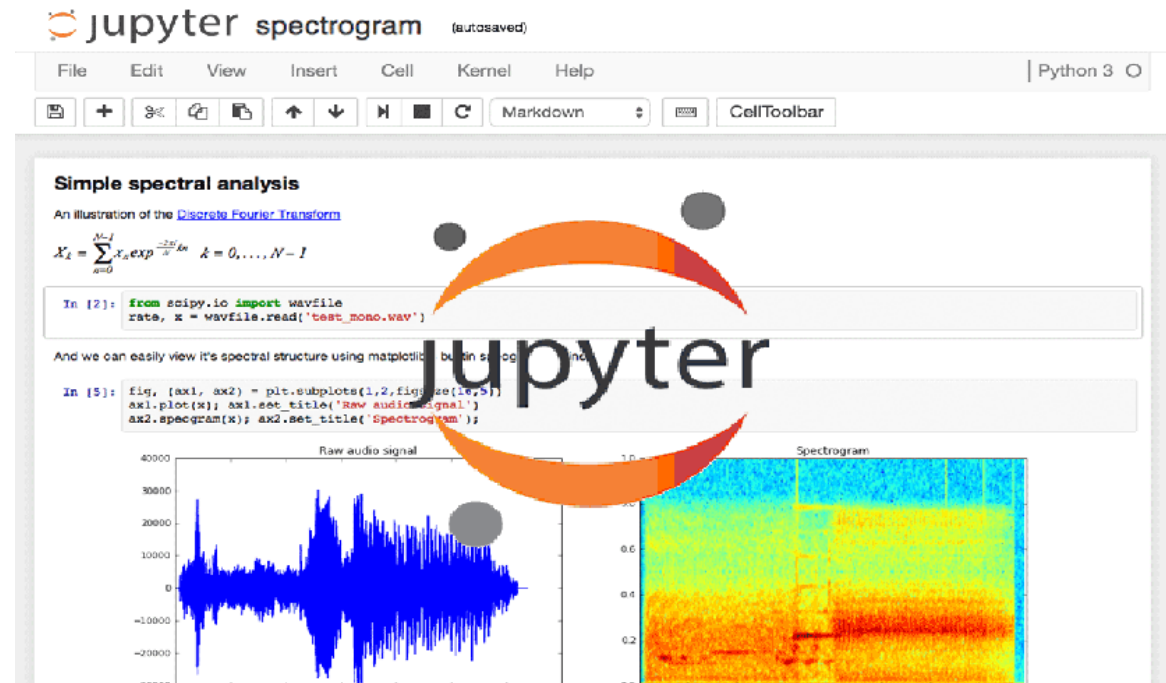
```
> conda install pyinstaller  
> pyinstaller hola.py  
> ./dist/hola/hola  
Hola Gabi
```

Herramientas para Python

- Jupyter Notebook
- Colaboratory
- *IDE*

Jupyter Notebook

- **Jupyter Notebook** es una aplicación web que permite crear y compartir documentos que contienen código fuente, ecuaciones, visualizaciones y texto explicativo. Entre sus usos está la limpieza y transformación de datos, la simulación numérica, el modelado estadístico, el aprendizaje automático y mucho más.



Instalación de Python y Jupyter

- En Windows se sugiere utilizar el entorno de conda:

<https://anaconda.org/anaconda/python>

- En Linux el intérprete del lenguaje viene instalado por default.

Para correr jupyter correr este comando:
`"jupyter-notebook"`

Luego abrir el navegador en
["http://localhost:8888/"](http://localhost:8888/)

Instalación de Paquetes

- En Windows se utilizará el gestor de paquetes de conda.
- En linux: `pip3 install jupyter`

Collaboratory

Es un servicio de google al que podemos acceder desde un navegador y que ejecuta las notebooks en la nube.

- Se puede acceder desde <https://colab.research.google.com/>
- O desde el google drive:
nuevo > más > conectar aplicaciones

