

Contenido creado por:

Isabel Maniega

La importancia de estar en las clases

- Este contenido ha sido incluido como resumen de las primeras horas del curso de Data Science
- Al principio todo es siempre complejo, entonces te pedimos un poquito de paciencia
- **Te recomendamos asistir a las clases..**
- De acuerdo, sabemos que **lleva mucho tiempo, pero es la mejor forma de entender ciertas cosas**
- La mayoría de dudas que puedas tener, serán (probablemente) cosas que se han comentado varias veces.

Gracias por adelantado por la atención!

Comentaremos que es lo primero que solemos hacer en Python..

y en este caso concreto en Jupyter Notebook es importar nuestras dependencias..

Importamos nuestras dependencias

```
In [1]: # ejemplo:

# pypi pandas (pip install pandas) en la cmd

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# NOTA:
# Deben importarse arriba del todo del archivo.
# (por lo menos cuando "terminemos" nuestro proyecto deben irse para arriba)
```

-----Primeros pasos-----

- -1- ¿Por qué necesitamos programar?
- -2- ¿Por qué Python?
- -3- Pros/Contras de Python
- -4- Tipos de proyectos que podemos hacer con Python: AI, IoT, Quantum Computing, Web, etc.
- -5- Python programado en el Bloc de notas
- -6- Python 3.8 versión estable (Marzo 2022)
- -7- Saber qué python tenemos: python --version
- -8- Editores de código: ATOM / VS Code / Jupyter Notebook / Anaconda / Colab / PyCharm / etc.
- -9- Entornos Virtuales (pip install virtualenv) "source env/bin/activate" (Linux)
- -10- Estructuras de Datos: Explicación con Pizarra (index para Arrays, listas, etc.)

-----Introducción-----

-1- Explicación general de Jupyter

- Explicación general de Jupyter.
 - Guardar archivo
 - Crear/quitar celdas
 - Ejecutar 1 celda/todas a la vez
 - Code/Markdown
 - etc

todo esto fue comentado en la primera clase.

y a día de hoy, con mucho esfuerzo, ya tienes esto dominado!

-2- ¿Cuándo usar Jupyter en vez de ATOM / VS Code / PyCharm, etc ?

Posibles casos de uso:

- Necesito hacer muchas gráficas y extraer información
- Quiero ir ejecutando código poco a poco para ir viendo el desarrollo del proyecto
- Temas de Inteligencia Artificial, aunque a veces es muy útil **Google Colab**

Ya vas entendiendo seguramente esto que comentamos..

Google Colab nos queda pendiente, por el momento

-3- ¿Existen Formas de hacer Aplicaciones Web embebidas en Jupyter?

- Existen muchos Frameworks a día de hoy que permiten hacer cosas dentro de este entorno,

no solamente WebApps

Ahora ya te suenan cosas como FastAPI, Tkinter, etc. pronto lo veremos además

-4- Extensión del archivo

- Observando arriba: .ipynb
- Normalmente en Python .py
- En Aplicaciones GUI con Python, idealmente: .pyw (no importante de momento)

Por el momento con archivos de Jupyter, ya lo tenéis dominado

-5- Recomendaciones Iniciales

- -1- Guardar el archivo cada poco, para evitar perder información
- -2- Restaurar el Kernel y ejecutar todas las celdas una vez hagamos bastantes cambios.

OJO: El valor de las variables puede verse afectado durante el proceso

Kernel --> Restart & Run All

- -3-

Entender el Algoritmo PREVIO a programarlo.

- -4-

Poner muchos comentarios a nuestro código

- -5-

Hacer un código ordenado

1º para entenderlo nosotros

2º para que sea legible por otros programadores

-6- Ojo con el valor de las variables

```
In [2]: x = 2
        print(x)

2
```

```
In [3]: y = 4
        y

Out[3]: 4
```

```
In [4]: x=5
        print(x)

5
```

-7- Conocer nuestro directorio de trabajo

```
In [5]: pwd

Out[5]: 'C:\\Users\\hti_b\\Desktop\\Temas_Revisados '
```

-8- x=1000 Vs x==1000

- x=1000 en programación no es igual que en el Algebra que todos conocemos
- Uno es asignación y el otro sirve para comparación en bucle if por ejemplo

-9- Instalación de Librerías

- pip install
- ejemplo: pip install pandas
- (NO siempre es así pero en la mayoría de casos se cumple)
- Es posible hacerlo en el propio Jupyter, pero idealmente ir a la CMD

Botón de Buscar (cmd)

3 posibles formas:

- pip install (cmd)
- pip install (Jupyter)
- !pip install (Jupyter si no funciona sin el "!")

```
In [6]: # !pip install pandas
```

- pypi: recomendación general

Lo ideal es copiar el nombre de la librería e ir a esa web

- Jupyter Vs Anaconda -> Jupyter
 - En Anaconda tenéis muchas librerías preinstaladas.
 - (Algunas veces) da Anaconda problema de incompatibilidad con algunos proyectos.
 - Idealmente: pip install notebook

-10- Importamos nuestras dependencias (mejor arriba)

```
In [7]: # ejemplo:

# pypi pandas (pip install pandas)

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# NOTA:
# Deben importarse arriba del todo del archivo.
# (por lo menos cuando "terminemos" nuestro proyecto deben irse para arriba)
```

Hasta aquí unas primeras recomendaciones, introducción

Pronto aprenderemos algunas cosas más de Python en este entorno

Esperamos que tu conclusión al leer esto sea..

- YA HE APRENDIDO MUCHAS COSAS!
- Aun me queda camino por recorrer además!

Isabel Maniega