

Módulo 1. Introducción a la ciencia de datos

Jupyter notebooks

Javier Cózar

Máster en Ciencia de Datos e Ingeniería de Datos en la Nube





IPython

- Python fue diseñado como un lenguaje para **scripting**
- La interfaz básica adolecía de varios problemas
- El paquete **IPython** surgió para solventarlos (*Interactive Python*)
 - Gestión de bloques, no de líneas
 - Visualización de gráficos (en nuevas ventanas, i.e. usando [Qt](#))
 - Ejecutar comandos de bash dentro del intérprete (usando la exclamación)

```
python -m pip install IPython
```

```
python -m IPython
```



Jupyter notebook

- Como evolución a IPython surgió [Jupyter](#)
- Nos permite interactuar con un intérprete de Python a través de una página web en el navegador
- Una libreta se divide en celdas, donde podemos escribir código en Python y ejecutar cada una cuando queramos
- IDE cómodo e interactivo (también muestra gráficos), perfecto para **Data Scientist**
- Para otro tipo de proyectos, existen IDEs muy potentes como [PyCharm](#)



Jupyter notebook

- La evolución de jupyter notebook es **jupyterlab**
- Se instala como un paquete más

```
python -m pip install jupyterlab
```

- Sigue un modelo **cliente - servidor**
 - El intérprete expone un servidor web
 - Accedemos desde cualquier navegador a través de la red

```
python -m jupyterlab
```



Jupyter notebook

- Una libreta tiene asociado un intérprete de Python, llamado **Kernel**
- Las libretas se dividen en celdas, que pueden contener código en **Python** o texto en [Markdown](#)
- Al trabajar con una libreta nos podemos encontrar en dos modos diferentes:
 - Modo comandos: no aparece el cursor en la celda. Se cambia a este modo pulsando escape, o clicando en la parte izquierda de la celda
 - Modo edición: aparece el cursor en la celda. Se cambia a este modo pulsando enter, o clicando en la parte central de la celda
- En función del modo podremos usar [shortcuts](#)

JUPYTERLAB

Entorno de desarrollo

