

Fundamentos Python 1

— Sintaxis Básica

Las herramientas del científico de datos

Juan Manuel Moreno — jmmoreno@profesores.imf.com



ÍNDICE

1. Objetivos unidad 1
2. El lenguaje de programación Python
3. Entorno de programación — Jupyter Notebook
4. Anexo — Operadores Python
5. Casos prácticos

01

Objetivos unidad 1

1.– Objetivos Unidad 1

- Conocer los principales fundamentos de Python.
- Saber instalar Jupyter Notebook, la herramienta que vamos a utilizar para trabajar con Python.
- Realizar desarrollos básicos en Python a través de Jupyter Notebook
- Saber cómo declarar, procesar y distinguir distintos tipos de variables.
- Conocer y manejar las sentencias condicionales If – Else.
- Conocer el funcionamiento de los bucles en Python, for y while.
- Trabajar con las principales estructuras de datos en Python: Tuplas, listas y diccionarios de datos.
- Comprender cómo modularizar los programas a través de funciones.
- Resolver problemas de diferente dificultad.



02

El lenguaje de programación Python

2.- El lenguaje de programación Python

2.1.- ¿Qué es Python?

- Python, fue creado en la década de los 90 por Guido Van Rossum, es Orientado a Objetos (POO), se trabajan con estructuras de datos de alto nivel con una sintaxis simple, una de las grandes ventajas de programar en Python es la curva de aprendizaje, es decir, en poco tiempo, podemos estar realizando grandes desarrollos software, en este caso, orientados al tratamiento de datos, dado el auge de ciertas tendencias como la Inteligencia Artificial y la Ciencia de Datos, han catapultado a Python como uno de los lenguajes más populares, más demandados y más utilizados del mundo.





Fuente:

<https://hackr.io/blog/best-programming-languages-to-learn>

2.- El lenguaje de programación Python

2.2.- Características

Orientado a Objetos

- Se realizan abstracciones del mundo real.

Multiplataforma

- Disponible en los principales SO.

Tipado dinámico

- No es necesario especificar el tipo de variable en el momento de creación.

Lenguaje interpretado

- Necesitamos el intérprete de Python para ejecutar cualquier programa.



03

Entorno de programación — Jupyter Notebook

3.– Entorno de programación — Jupyter Notebook

3.1.– Características de Jupyter Notebook

- Entorno de programación desde nuestro navegador (el que tengamos por defecto)
- En un notebook podemos incluir código, documentación, texto, tablas, imágenes, gráficas, etc. Es decir, funciona de una forma similar a un cuaderno en el que vamos incluyendo los apuntes de una lección.
- Muy extendido en el mundo de la investigación gracias a su facilidad para mostrar resultados.
- Cada vez más extendido en otros lenguajes.
- Tiene autocompletar.
- La ayuda de las funciones se despliega en línea.
- Utilizaremos Jupyter notebook desde una distribución Anaconda.



3.2.– Instalación

- Accedemos a: <https://www.anaconda.com/products/individual> pulsamos en Download

Individual Edition

Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Download

3.2.– Instalación

- Seleccionamos el instalador adecuado para nuestro Sistema Operativo, automáticamente comenzará la descarga.

Anaconda Installers

Windows 

Python 3.8

64-Bit Graphical Installer (457 MB)


32-Bit Graphical Installer (403 MB)

MacOS 

Python 3.8

64-Bit Graphical Installer (435 MB)

64-Bit Command Line Installer (428 MB)

Linux 

Python 3.8

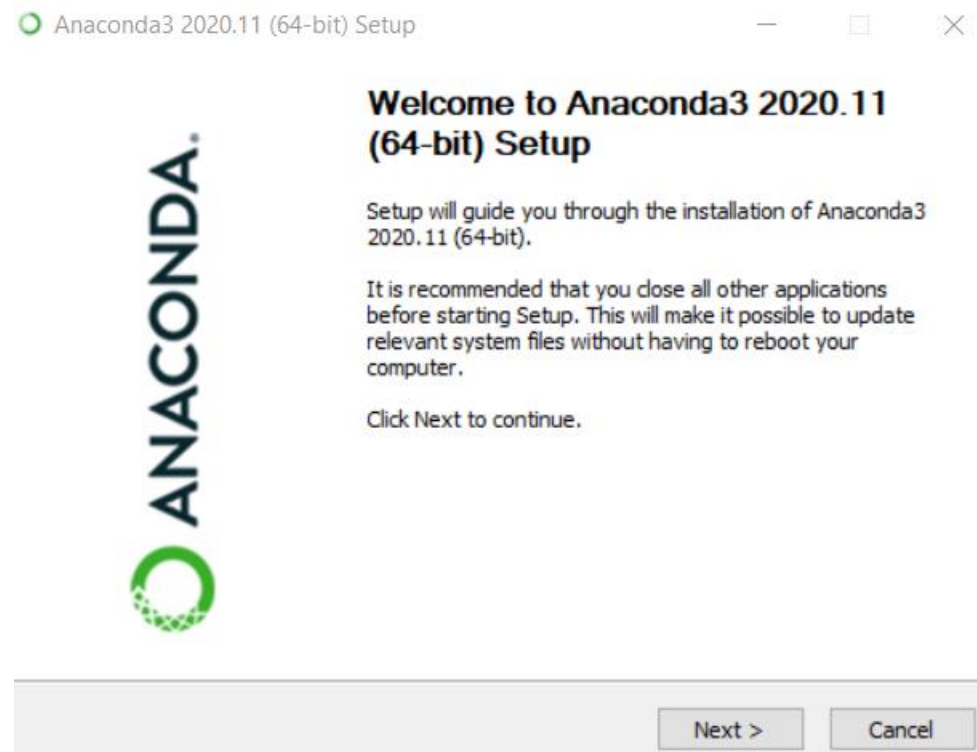
64-Bit (x86) Installer (529 MB)

64-Bit (Power8 and Power9) Installer (279 MB)

3.– Entorno de programación — Jupyter Notebook

3.2.– Instalación

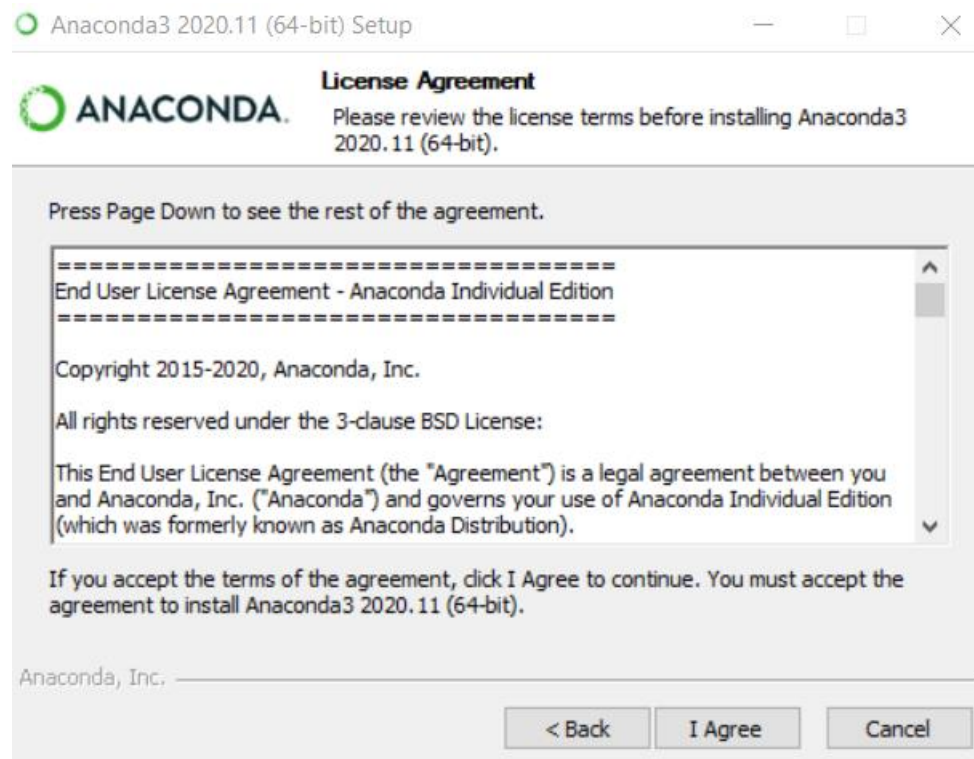
- Ejecutamos el archivo descargado. Pulsamos en Next



3.- Entorno de programación — Jupyter Notebook

3.2.- Instalación

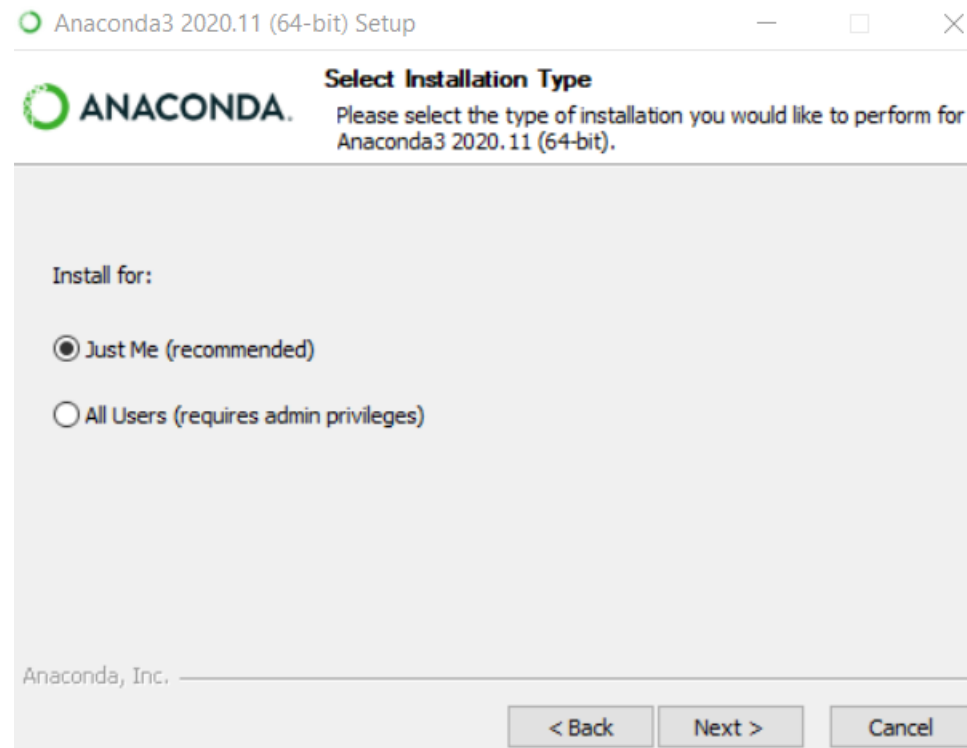
- Aceptamos el acuerdo de licencia



3.– Entorno de programación — Jupyter Notebook

3.2.– Instalación

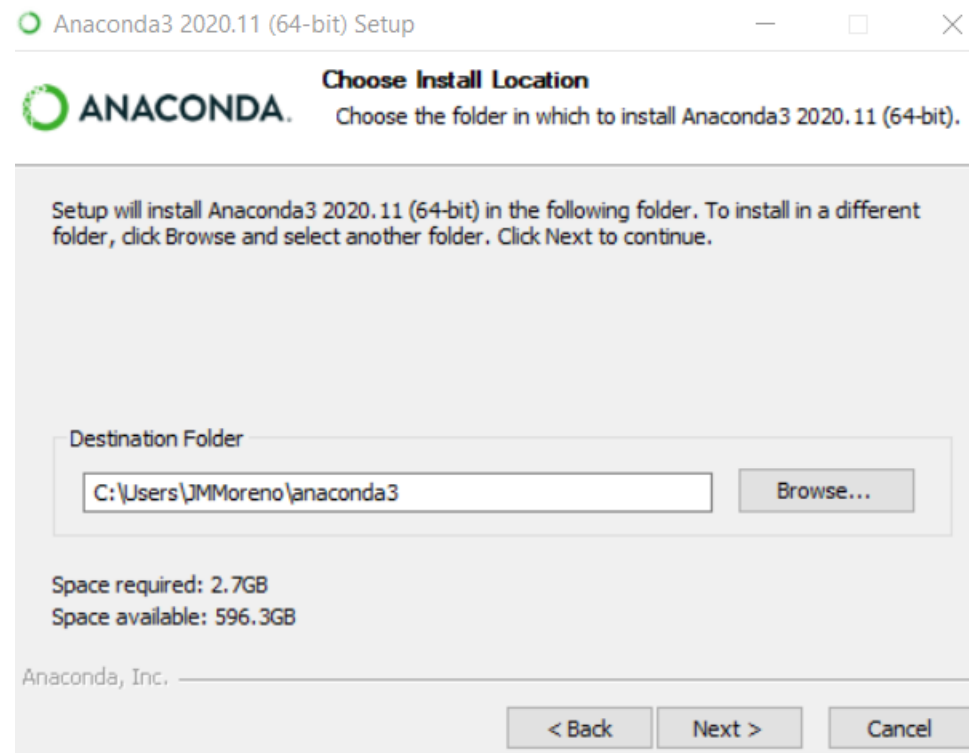
- Seleccionamos Just for me



3.– Entorno de programación — Jupyter Notebook

3.2.– Instalación

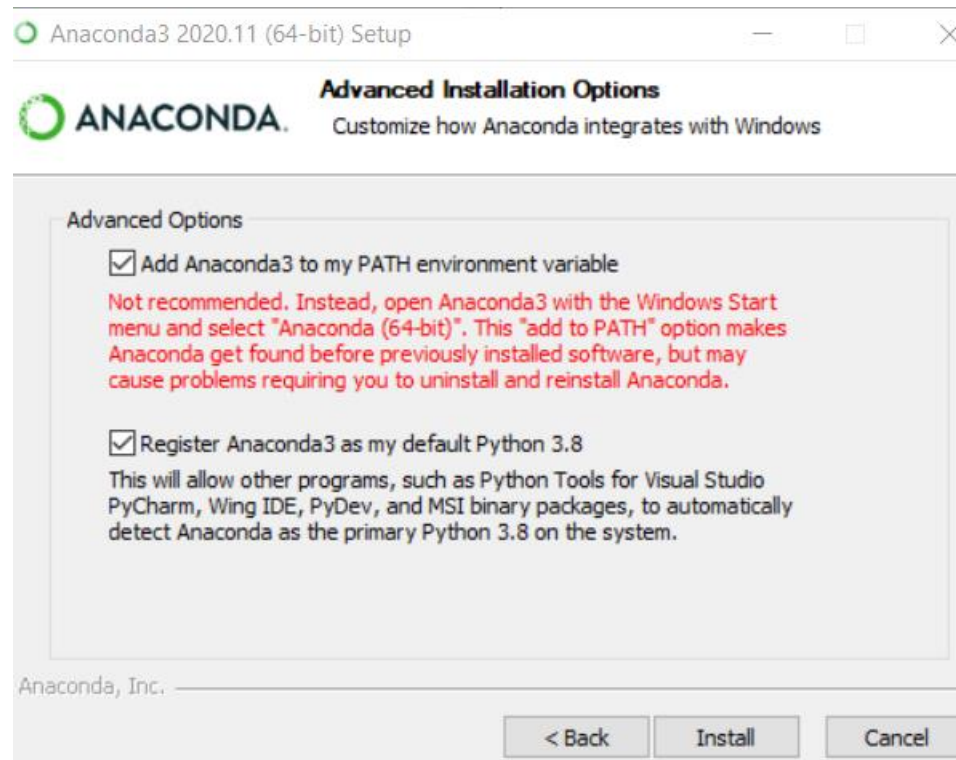
- Se recomienda dejar por defecto la ruta de instalación



3.- Entorno de programación — Jupyter Notebook

3.2.- Instalación

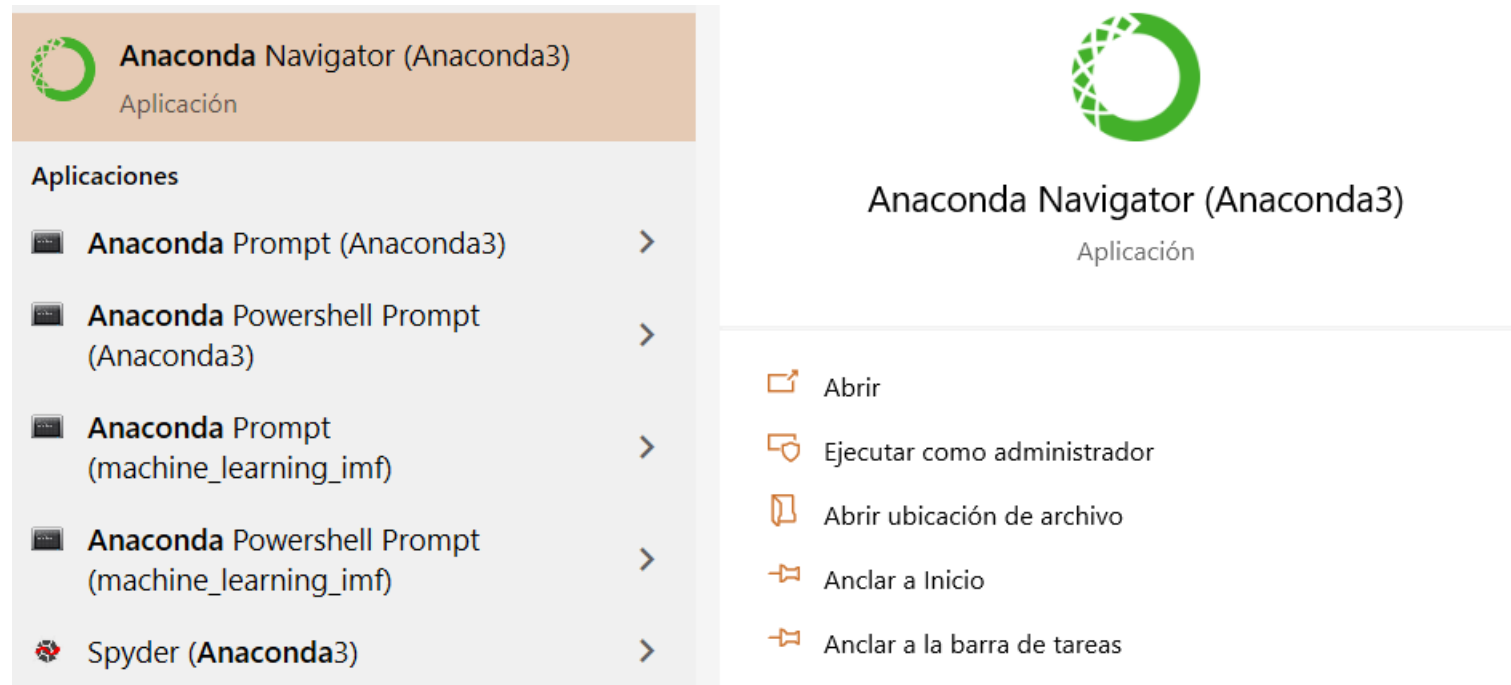
- Importante agregar el Path de Python y Anaconda como variables de entorno y pulsamos en Install



3.- Entorno de programación — Jupyter Notebook

3.2.- Instalación

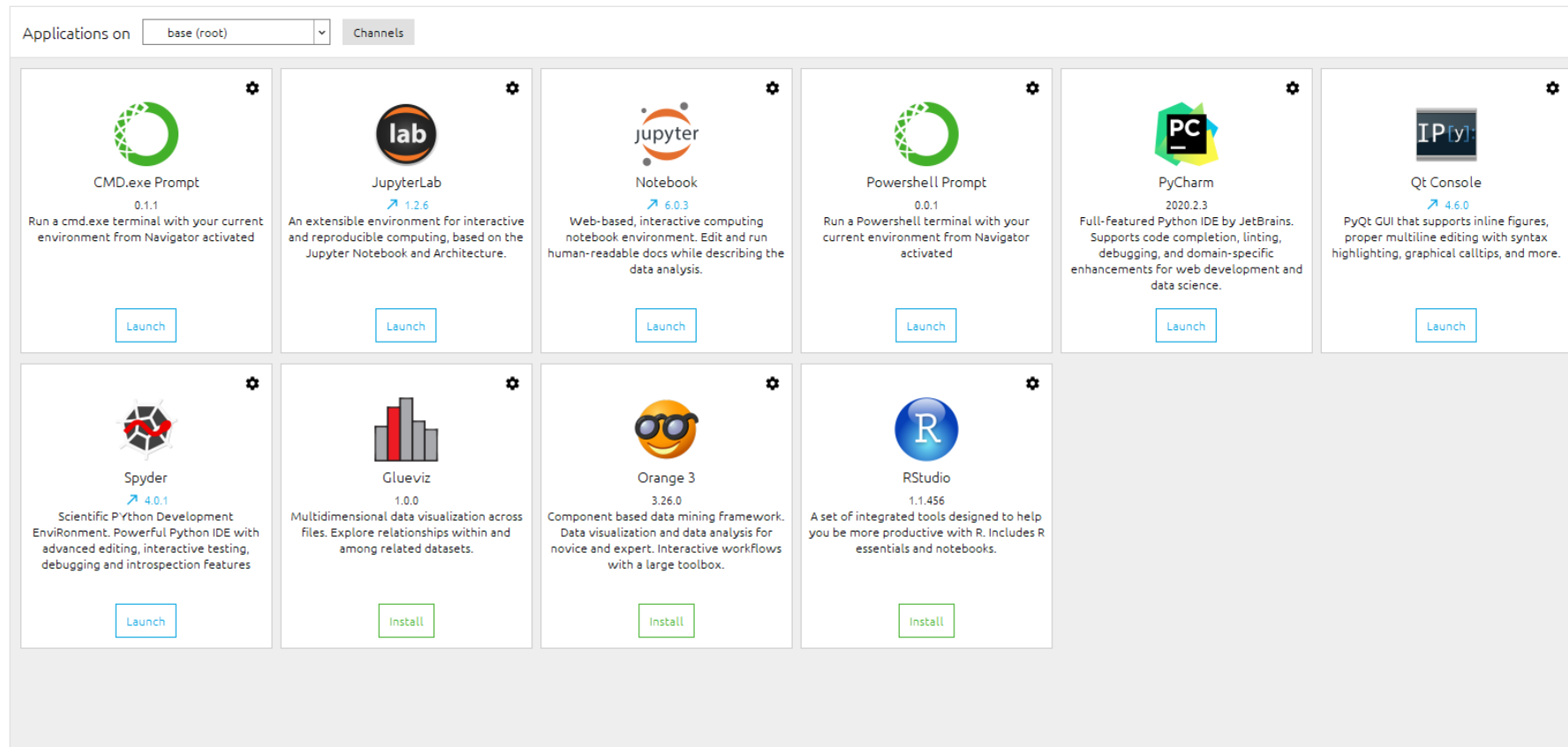
- Una vez finalizado el proceso de instalación, buscamos y lanzamos la aplicación Anaconda Navigator



3.- Entorno de programación — Jupyter Notebook

3.3.- Anaconda Navigator


- Aparecerá un gestor de aplicaciones, lanzamos Notebook



3.– Entorno de programación — Jupyter Notebook

3.3.– Anaconda Navigator

- Aparecerá en nuestro navegador Jupyter Notebook



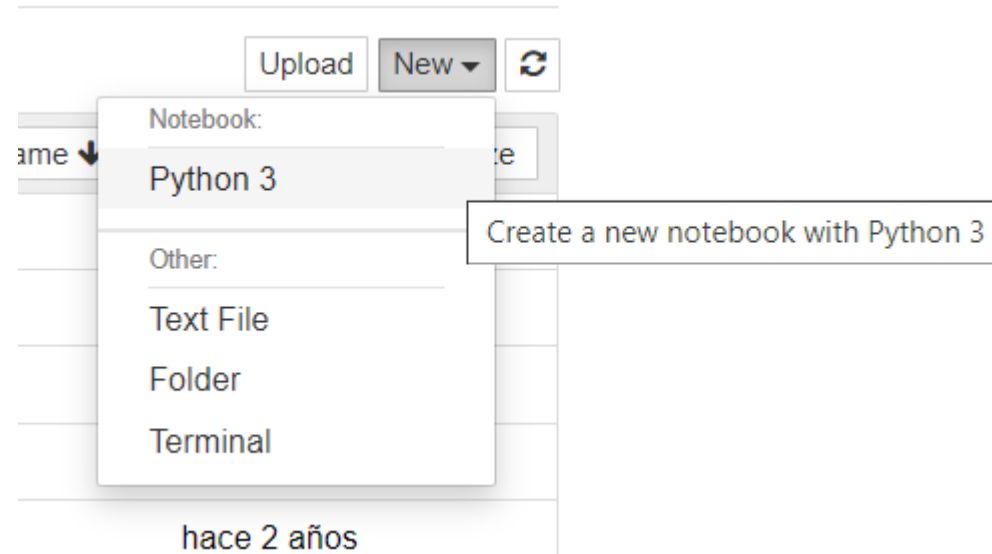
The screenshot displays the Jupyter Notebook web interface. At the top, the 'jupyter' logo is on the left, and 'Quit' and 'Logout' buttons are on the right. Below the logo, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a message: 'Select items to perform actions on them.' To the right of this message are 'Upload', 'New', and a refresh icon. The main area is a file browser showing a list of files and folders. The table has columns for 'Name', 'Last Modified', and 'File size'. The files listed are:

Name	Last Modified	File size
0		
3D Objects	hace 9 días	
Anaconda3	hace 21 minutos	
Contacts	hace 9 días	
Desktop	hace 21 minutos	
Documents	hace 2 horas	
Downloads	hace 25 minutos	
Favorites	hace 9 días	
Links	hace 9 días	
Microsoft	hace 6 meses	
Music	hace 9 días	
OneDrive	hace 22 días	
Pictures	hace 9 días	
Saved Games	hace 9 días	
scikit_learn_data	hace 5 meses	
Searches	hace 9 días	
Videos	hace 9 días	
VirtualBox VMs	hace 12 días	

3.– Entorno de programación — Jupyter Notebook

3.4.– Jupyter Notebook

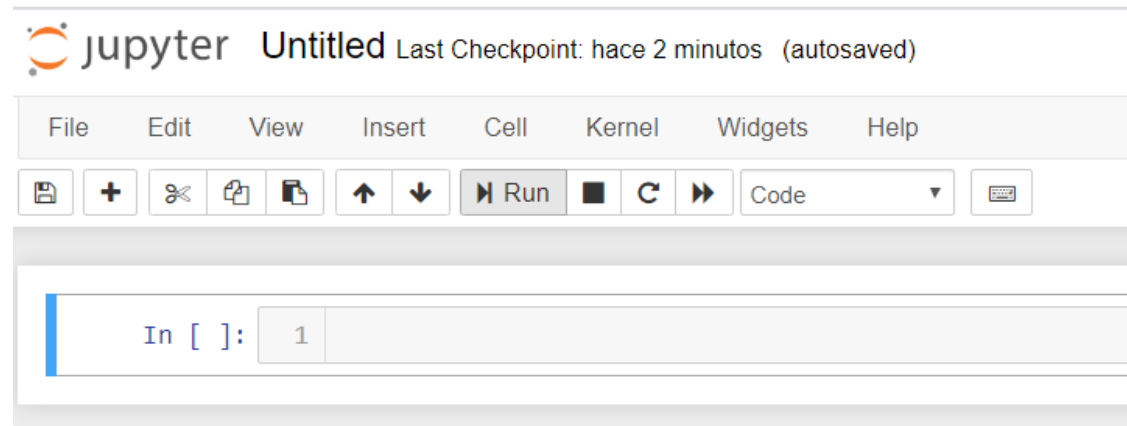
- Para lanzar un nuevo notebook pulsamos sobre la pestaña New y seleccionamos Python 3



3.– Entorno de programación — Jupyter Notebook

3.4.– Jupyter Notebook

- Aparecerá la siguiente vista con un notebook vacío, como podemos ver por defecto se llamará 'Untitled'

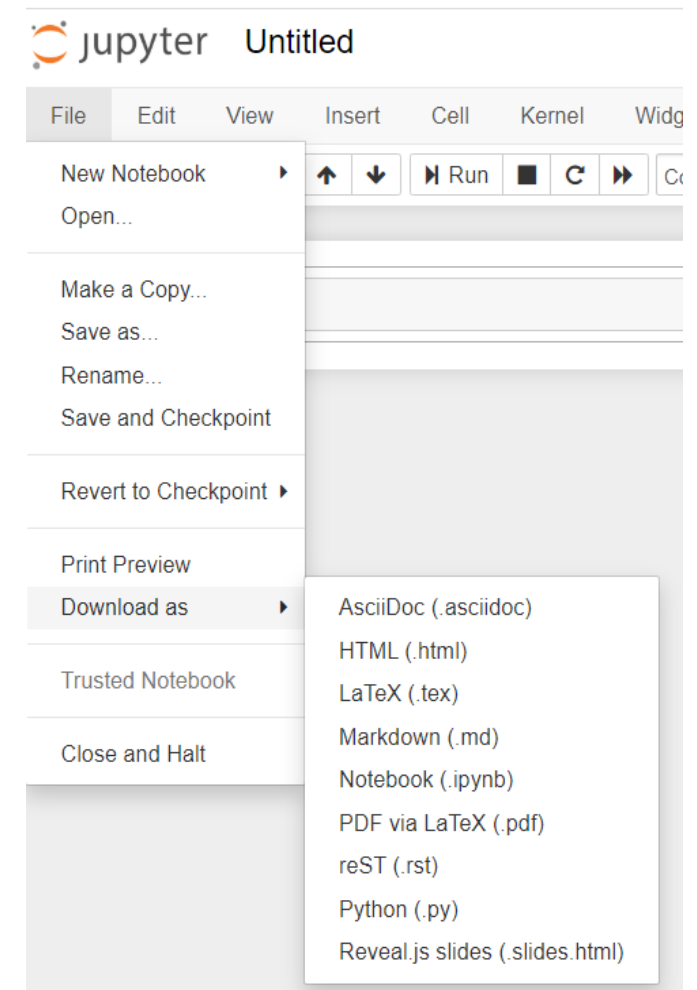


3.– Entorno de programación — Jupyter Notebook

3.4.– Jupyter Notebook

• Dentro de esta sección podemos crear un nuevo notebook, renombrarlo, hacer una copia, guardar un checkpoint del mismo, imprimirlo y descárgalo, las opciones que más utilizaremos serán:

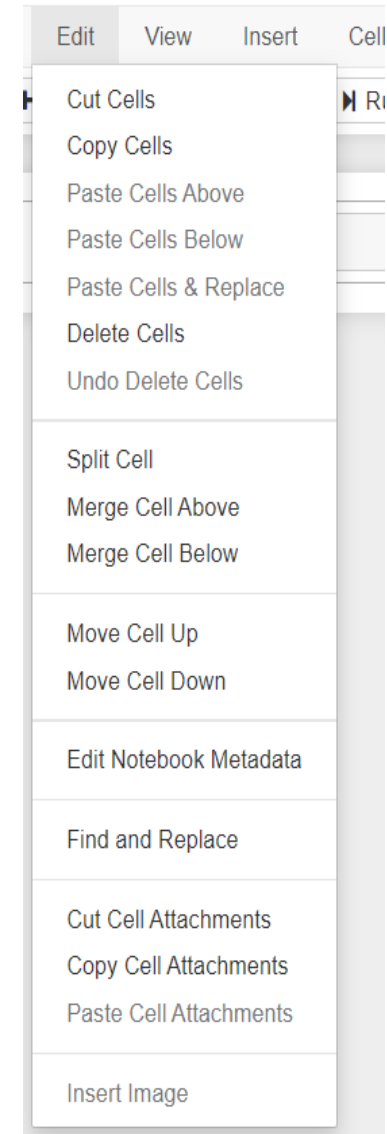
- Notebook
- PDF
- HTML
- Python



3.– Entorno de programación — Jupyter Notebook

3.4.– Jupyter Notebook — Vistas Edit

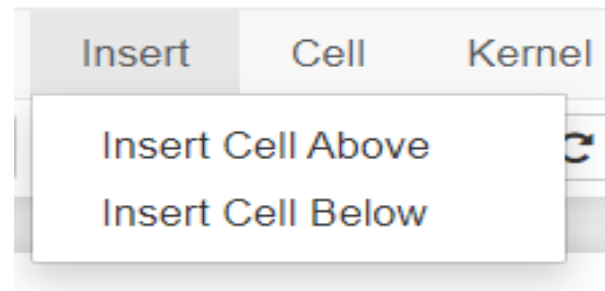
- Desde esta sección podremos básicamente gestionar las celdas de código o anotaciones ya sea, copiando, cortando o borrando celdas como operaciones más básicas y, posteriormente, también tendremos opciones más avanzadas en edición de celdas como dividir en dos celdas una celda o juntar dos celdas en una. Además, podremos mover las celdas de posición hacia arriba o hacia abajo, esto es muy recomendable para ir organizando nuestro código y que quede de una forma más legible.



3.– Entorno de programación — Jupyter Notebook

3.4.– Jupyter Notebook — Vistas Insert

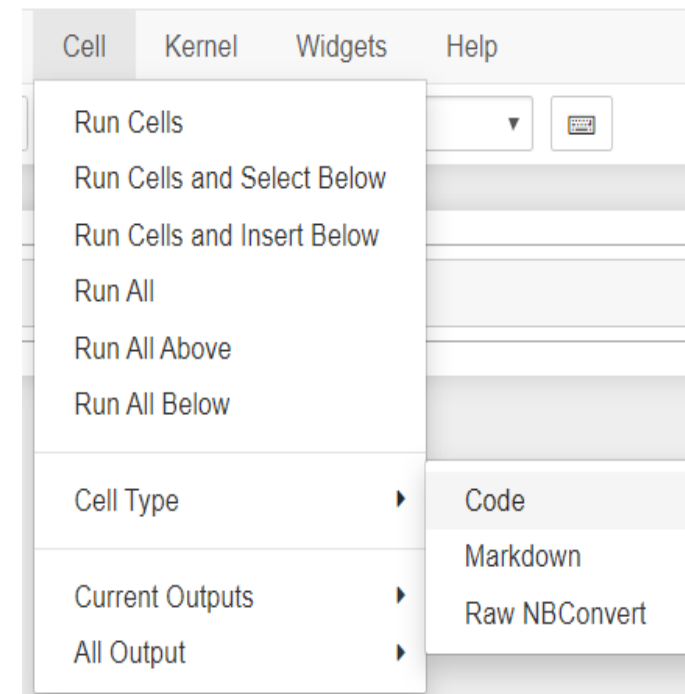
- Esta sección es la encargada de poder insertar nuevas celdas ya sea a continuación de la celda en donde tengamos el cursor, o después de la celda en donde tengamos el cursor



3.– Entorno de programación — Jupyter Notebook

3.4.– Jupyter Notebook — Vistas Cell

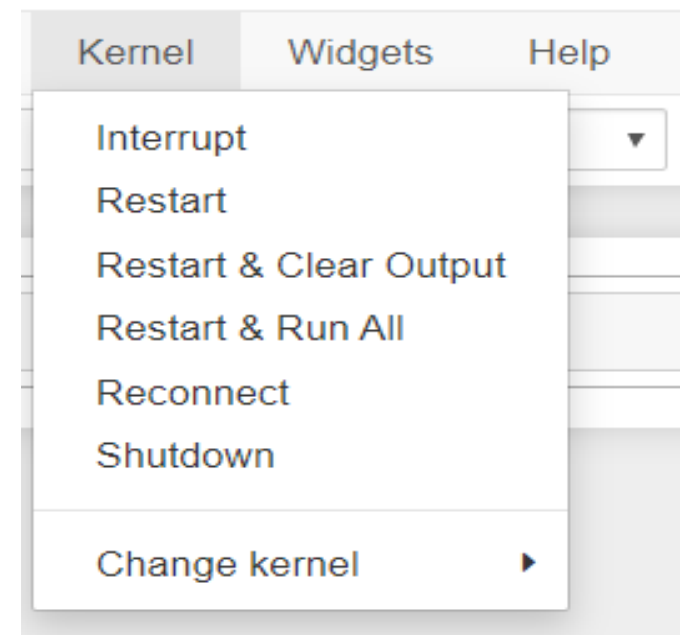
- Esta parte se encarga de ejecutar las celdas de código, pudiendo elegir entre una sola celda, ejecutar desde el cursor hasta el final, ejecutar desde el principio hasta el cursor o ejecutar todas las celdas.
- Es también muy importante hacer énfasis en los tipos de celda que podemos encontrarnos en un notebook:
- Code: Celdas con código fuente Python.
- Markdown: Celdas con anotaciones estilo markdown. (muy aconsejable para documentar nuestros notebooks).
- Raw NBConvert: Celdas en crudo, la ejecución de estas celdas no produce ninguna salida. (Por lo general, es un formato poco utilizado).



3.– Entorno de programación — Jupyter Notebook

3.4.– Jupyter Notebook — Vistas Kernel

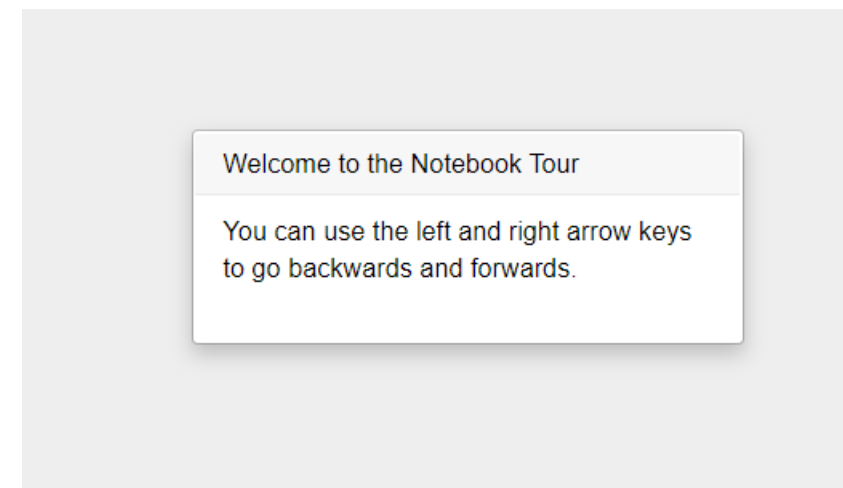
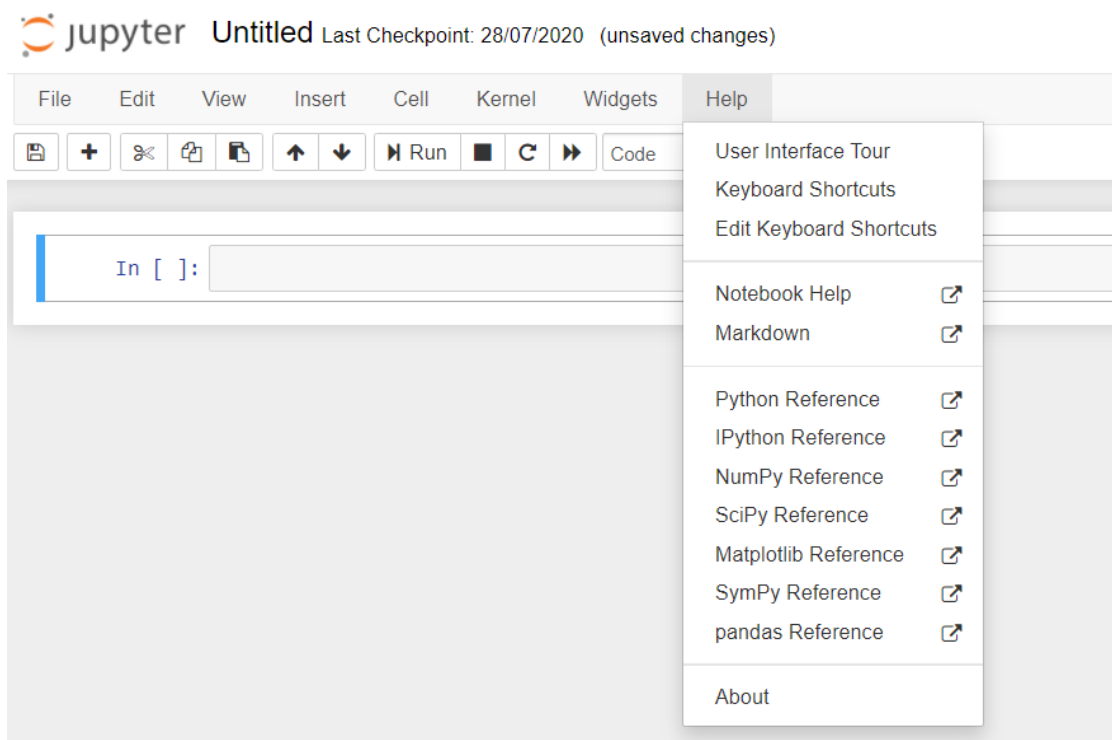
- Esta es una de las secciones más importantes ya que se encarga de mantener activo el kernel de Python, a través de esta sección, podremos pararlo, reiniciarlo, reiniciar y limpiar los output de las celdas, reiniciar y ejecutar todas las celdas, reconectar y apagar el kernel.
- Siempre que se interrumpa la actividad del kernel, todas las variables que estuviera almacenadas en memoria se borrarán.



3.– Entorno de programación — Jupyter Notebook

3.4.– Jupyter Notebook — User Interface Tour

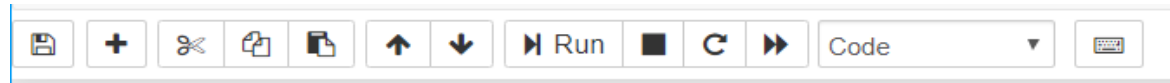
- De forma automática, se puede ver un pequeño tour por la interfaz de Jupyter notebook, desde la pestaña Help – User Interface Tour



3.– Entorno de programación — Jupyter Notebook

3.4.– Jupyter Notebook — Botonera principal

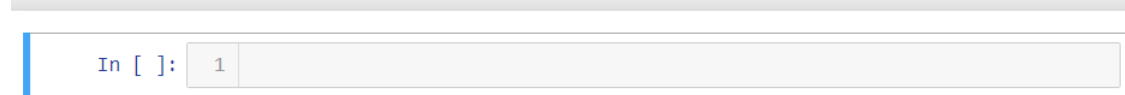
- En ella podremos encontrar los botones principales para poder gestionar el notebook y sus celdas como, guardar el notebook, añadir, cortar, copiar y pegar celdas, moverlas de posición, así como ejecutar una celda del notebook y parar la ejecución de una celda. Al igual que en la sección 'Cell', podremos definir el tipo de celda que queremos implementar a través del menú desplegable.



3.– Entorno de programación — Jupyter Notebook

3.4.– Jupyter Notebook — Celda o chunk

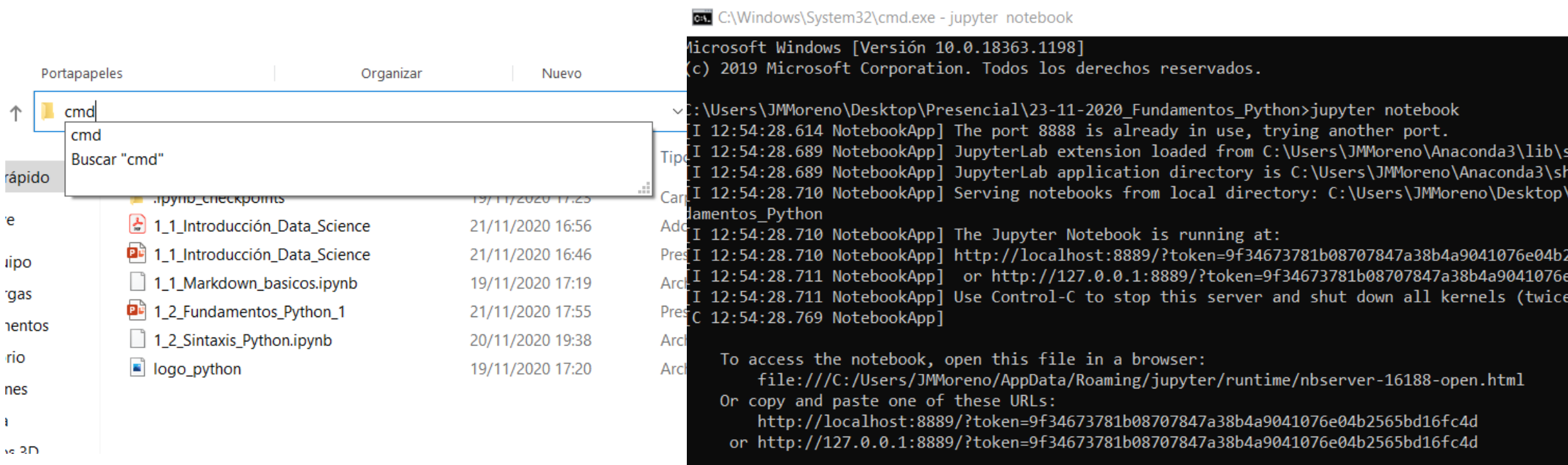
- Dentro de una celda escribiremos el código fuente que queramos implementar en Python, comentarios o anotaciones Markdown. A medida que vayamos ejecutando celdas, dentro del apartado In [], podremos ver el seguimiento del código ya que dentro de este apartado se guarda el orden en el que se han ido ejecutando las celdas.



3.- Entorno de programación — Jupyter Notebook

3.4.- Jupyter Notebook — Lanzar notebooks

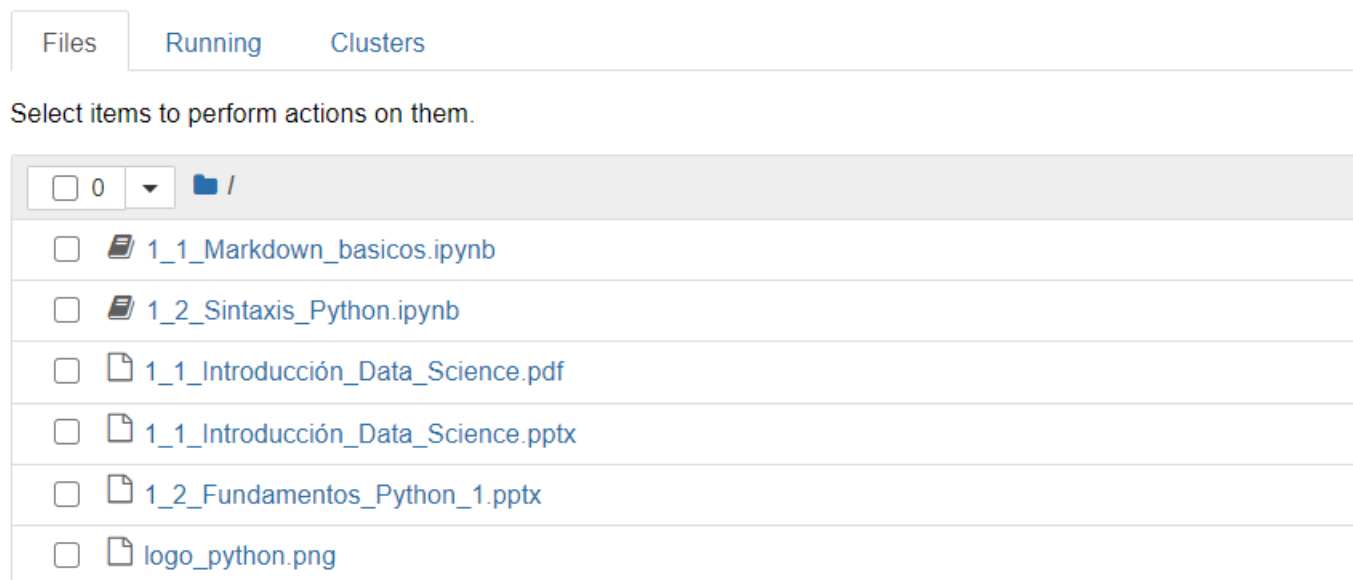
- Desde un directorio en el que tengamos notebooks que vayamos a editar o, simplemente el directorio en el que tengamos datos con los que queremos trabajar o, el contenido de una lección, lanzamos una consola de comandos.



3.– Entorno de programación — Jupyter Notebook

3.4.– Jupyter Notebook — Lanzar notebooks

- Tendremos Jupyter Notebook abierto directamente desde el directorio sobre el que hemos lanzado Jupyter, ahorrándonos tener que pasar por Navigator.



Seguimiento práctico del contenido

A partir de aquí, vamos a seguir la introducción de Markdown y la sintaxis básica de Python desde los siguientes notebooks:

[1_1_Markdown_basicos.ipynb](#)

[1_2_Sintaxis_Python.ipynb](#)

04 Anexo operadores Python

4.1 Aritméticos

Operadores aritméticos	
Operación	Código
Suma	$a + b$
Resta	$a - b$
Multiplicación	$a * b$
División real	a / b
Resto o módulo	$a \% b$
Exponencial	$a ** b$
División entera	$a // b$

4.2 Asignación

Operadores de asignación		
Operación	Código	Equivalencia
Asignación	$a = b$	
Suma	$a += b$	$a = a + b$
Resta	$a -= b$	$a = a - b$
Multiplicación	$a *= b$	$a = a * b$
División real	$a /= b$	$a = a / b$
Resto o módulo	$a %= b$	$a = a \% b$
Exponencial	$a **= b$	$a = a ** b$
División entera	$a //= b$	$a = a // b$

4.3 Comparación

Operadores de comparación	
Comparador	Código
Mayor que	$x > y$
Menor que	$x < y$
Mayor o igual que, menor o igual que	$x \geq y$, $x \leq y$
Igual que	$x == y$
Distinto que	$x != y$
and	$x > y$ and $x < z$
or	$x > y$ or $x < z$
not	$\text{not}(x > y \text{ and } x < z)$, $\text{not}(x > y \text{ or } x < z)$

4.4 Identidad

Operadores de identidad	
Comparador	Código
is	x is y
is not	x is not y

4.5 Identidad

Operadores de pertenencia	
Comparador	Código
in	'a' in 'hola'
not in	'a' not in 'hola'

IMF

Smart Education