

# JUPYTER

## NOTEBOOK



# Presentación

- **Juan Ignacio Rodríguez de León**
- **Ingeniero en Informática por la Universidad de Las Palma de Gran Canaria**
- **Uso Python desde la versión 1.4**
- **Actualmente trabajo como analista en la unidad de informática del Parlamento de Canarias**
- **@jileon en twitter**
- **euribates@gmail.com**





# Objetivos del curso

- **Aprende a crear y a usar notebooks de Jupyter**
- **Algunos conceptos y tecnologías que trataremos:**
  - **Python**
  - **Markdown**
  - **Numpy**
  - **Scipy**
  - **Matplot**
  - **Sympy**
  - **Julia**
  - **R**
  - **Estadísticas**
  - **Método de Montecarlo**
  - **Mathjax**
  - **Github**

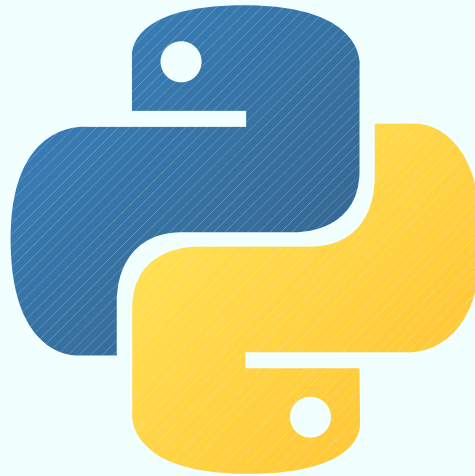
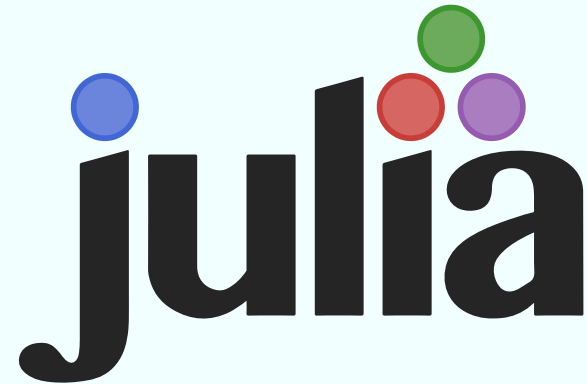


# Qué es Jupyter notebook

- **Entorno interactivo, basado en la web, que permite generar documentos enriquecidos que pueden contener código, texto, fórmulas matemáticas, dibujos, contenidos multimedia...**
- **Los notebooks pueden ser convertidos a otros formatos de archivos como HTML, presentaciones de diapositivas, LaTeX, PDF, Python, etc...**



Jupyter = Python + Julia + R



# Ejercicio 1 – Arrancar Jupyter

```
jupyter --version
```

```
jupyter notebook
```



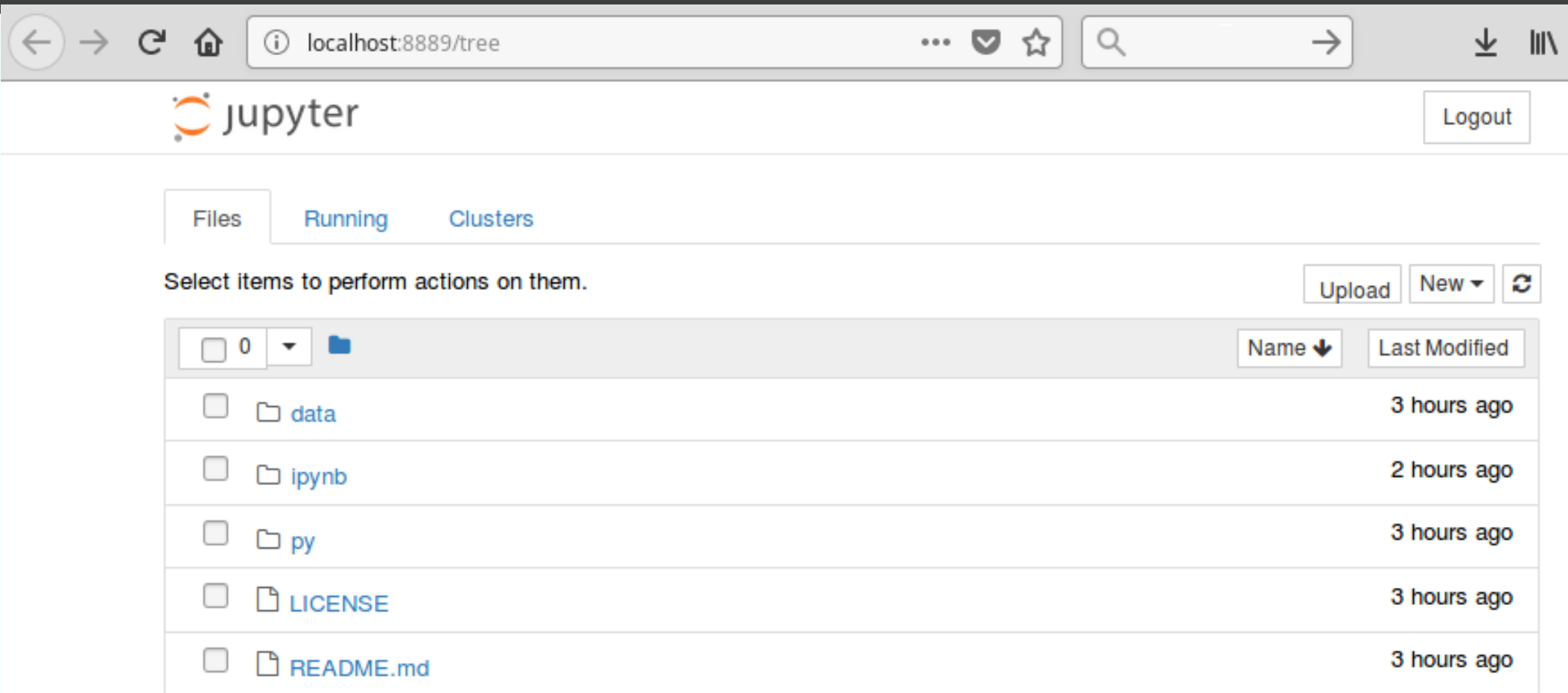
# Ejercicio 1 – Arrancar Jupyter

```
jupyter --version
```

```
jupyter notebook
```



# Primera mirada a Jupyter



The screenshot shows the Jupyter web interface in a browser. The address bar indicates the URL is `localhost:8889/tree`. The Jupyter logo is in the top left, and a "Logout" button is in the top right. Below the logo, there are tabs for "Files", "Running", and "Clusters", with "Files" being the active tab. A message says "Select items to perform actions on them." To the right of this message are buttons for "Upload", "New" (with a dropdown arrow), and a refresh icon. Below this is a table of files and folders. The table has two columns: "Name" (with a downward arrow) and "Last Modified". The table contains five rows: a folder named "data" (3 hours ago), a folder named "ipynb" (2 hours ago), a folder named "py" (3 hours ago), a file named "LICENSE" (3 hours ago), and a file named "README.md" (3 hours ago). Each row has a checkbox on the left for selection.

localhost:8889/tree

jupyter Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↕ ↻

<input type="checkbox"/> 0 ▾	Name ↓	Last Modified
<input type="checkbox"/> data		3 hours ago
<input type="checkbox"/> ipynb		2 hours ago
<input type="checkbox"/> py		3 hours ago
<input type="checkbox"/> LICENSE		3 hours ago
<input type="checkbox"/> README.md		3 hours ago



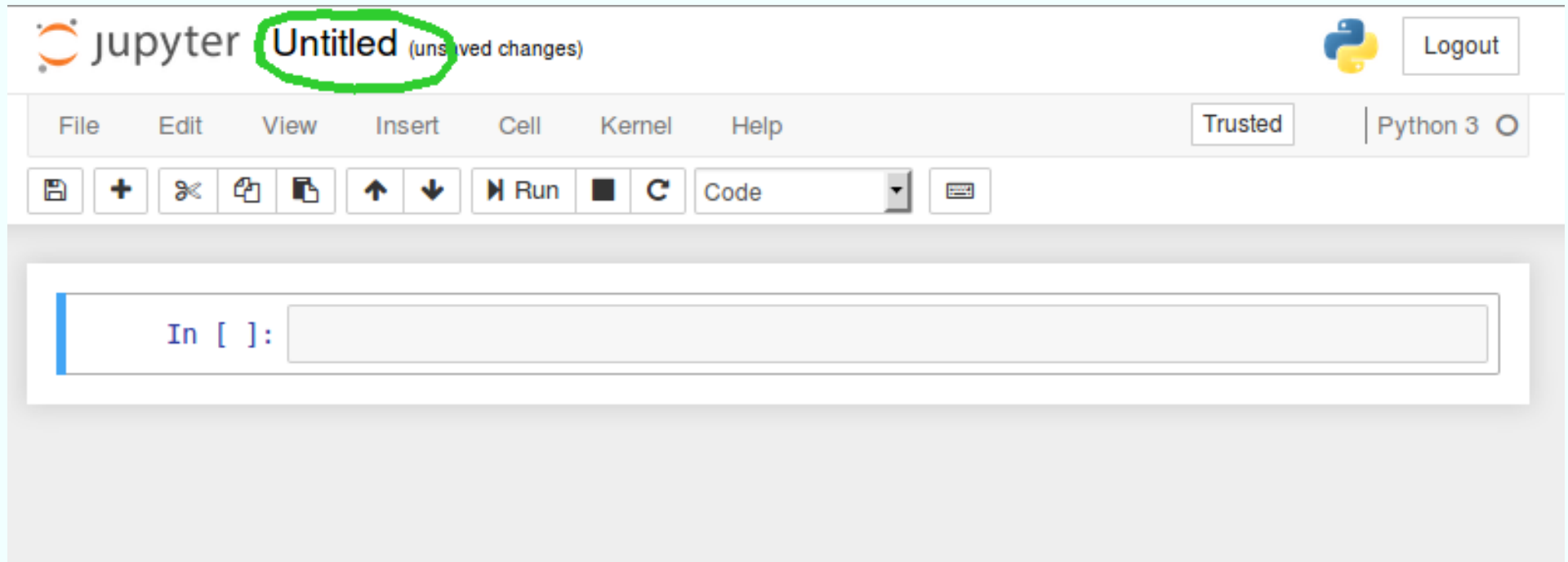


# Ejercicio 2 – Nuestro primer notebook

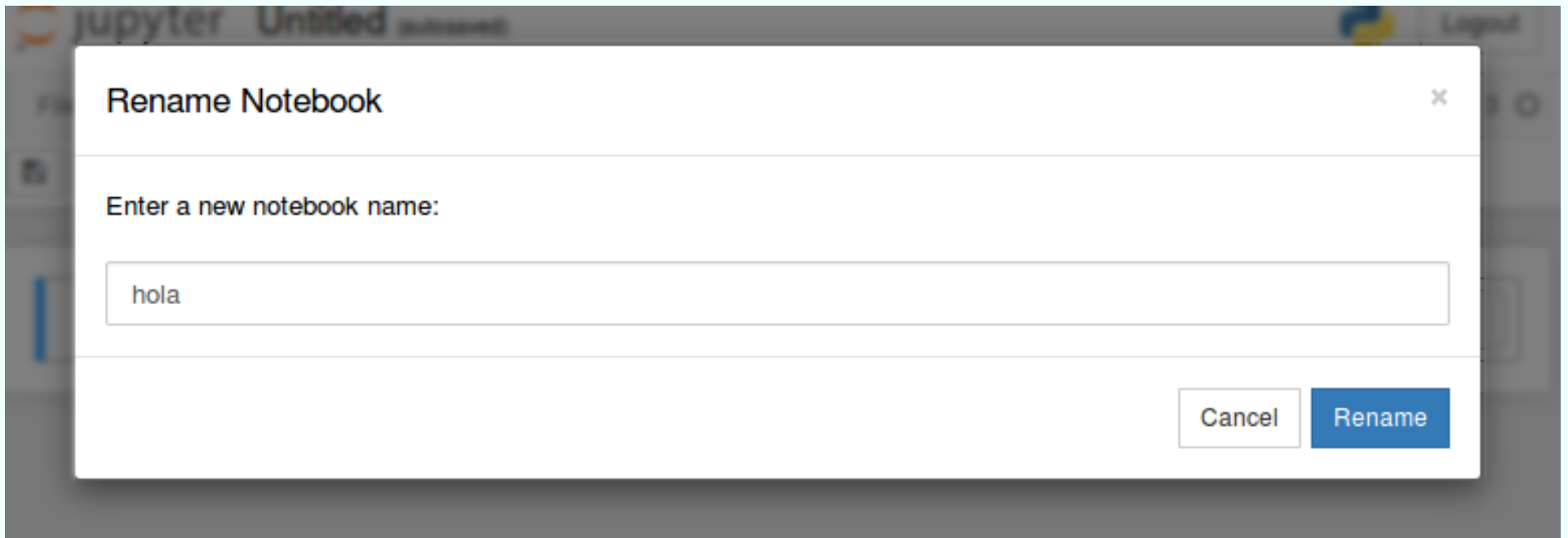
- **Seleccionar en el botón new → Python 3**
- **Nos abre una nueva pestaña**
- **El notebook no tienen nombre, cambiarlo por hola**
- **Salvar el notebook**
- **Comprobar que el nuevo notebook aparece en el dashboard (La pestaña original)**



# Asignar un nombre



# Indicar el nuevo nombre



A screenshot of the Jupyter Notebook interface showing a 'Rename Notebook' dialog box. The dialog box is white with a title bar that says 'Rename Notebook' and a close button (X) in the top right corner. Below the title bar, there is a text input field with the placeholder text 'Enter a new notebook name:'. The input field contains the text 'hola'. At the bottom right of the dialog box, there are two buttons: 'Cancel' and 'Rename'.

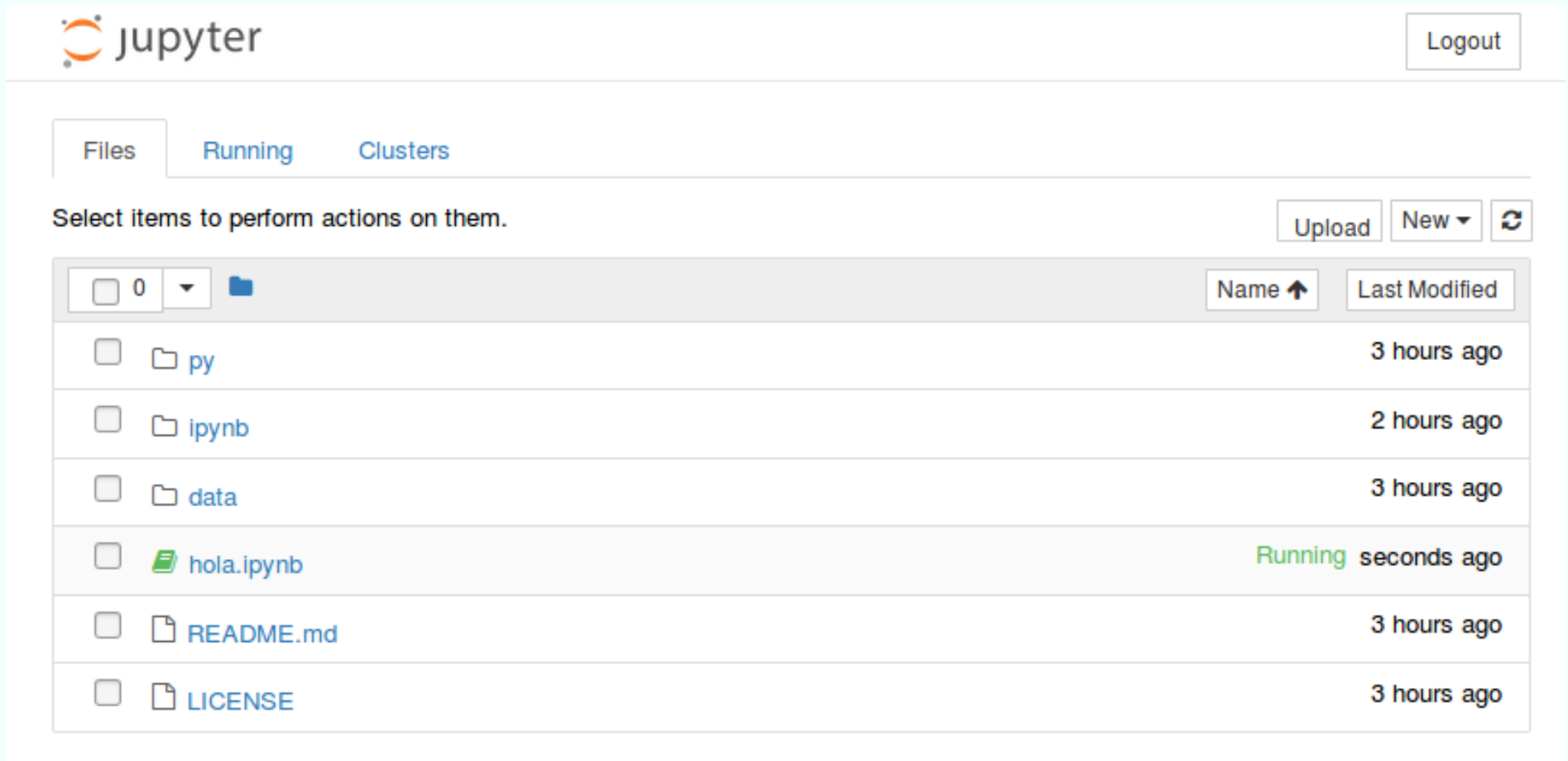
Rename Notebook

Enter a new notebook name:

hola

Cancel Rename

# Volvemos al dashboard



The image shows the Jupyter dashboard interface. At the top left is the Jupyter logo, and at the top right is a 'Logout' button. Below the logo are three tabs: 'Files', 'Running' (which is active), and 'Clusters'. A message 'Select items to perform actions on them.' is displayed. To the right of this message are buttons for 'Upload', 'New' (with a dropdown arrow), and a refresh icon. Below this is a table of running notebooks and files. The table has two columns: 'Name' (with an upward arrow icon) and 'Last Modified'. The first row shows a folder icon, a checkbox, the number '0', a dropdown arrow, and a folder icon. The subsequent rows show folders named 'py', 'ipynb', and 'data', and files named 'hola.ipynb' (with a green notebook icon), 'README.md' (with a document icon), and 'LICENSE' (with a document icon). The 'Last Modified' column shows '3 hours ago' for the folders and files, and 'Running seconds ago' for the 'hola.ipynb' notebook.

jupyter Logout

Files Running Clusters

Select items to perform actions on them. Upload New ▾ ↻

<input type="checkbox"/> 0 ▾		Name ↑	Last Modified
<input type="checkbox"/>	📁	py	3 hours ago
<input type="checkbox"/>	📁	ipynb	2 hours ago
<input type="checkbox"/>	📁	data	3 hours ago
<input type="checkbox"/>	📄	hola.ipynb	Running seconds ago
<input type="checkbox"/>	📄	README.md	3 hours ago
<input type="checkbox"/>	📄	LICENSE	3 hours ago

# Ejercicio 3 – Hola, mundo

- Es tradición al empezar con un nuevo lenguaje de programación hacer el programa más sencillo que haga algo
- En este caso, solo imprimir el texto “¡Hola, mundo!”
- Más información en Wikipedia
  - [https://es.wikipedia.org/wiki/Hola\\_mundo](https://es.wikipedia.org/wiki/Hola_mundo)



# Cosas que hemos aprendido

- En python se usa la función `print()` para escribir texto
- El texto debe estar delimitado por comillas
- Los notebooks son una lista de celdas
- Por ahora, solo hemos visto celdas que ejecutan código python





# Pyhon – Tipos de datos

- **Enteros**
- **De coma flotantes o reales**
- **Lógicos o Booleanos**
- **Cadenas de texto**



# Ejercicio 4 – Calcula $2^{124}$

- Los enteros en Python no tienen en principio límite de tamaño, excepto el marcado por la memoria
- Internamente se usan enteros de 64 bits, pero si la cantidad es mayor, se pasa a una clase especial sin límite, Long, de forma transparente



# Que hemos aprendido

- Los enteros aceptan todas las operaciones básicas que podríamos esperar: suma, resta, división, multiplicación, etc...
- El operador potencia o exponenciación es **\*\***
- No hay límite a la resolución de un entero
- El último resultado evaluado en la celda se mostrara, aun sin hacerle un print



# Números en coma flotante

- Hay que incluir el punto decimal para que el interprete de Python lo reconozca como tal
- Como en casi todos los lenguajes, se usa el estandar IEEE para aritmética en coma flotante IEEE 754
- Esta representación no puede almacenar con precisión absoluta algunos números



# Ejercicio 5 – Números reales

- Crear una variable entera y una variable en coma flotante o entera
- Llamar a la función `type` para ver el tipo de cada variable
- Asignar a una variable una suma compuesta de 10 veces el número 0.1
- Pista

```
a = 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1  
+ 0.1 + 0.1 + 0.1 + 0.1
```





# Qué hemos aprendido

- Hay que usar el punto decimal para que la variable se entienda como real o de coma flotante
- La representación en coma flotante no es perfecta
- Las variables se asignan con el carácter =
- Las variables son fuertemente tipadas (más sobre eso más adelante)
- Los nombres de variables en python distinguen mayúsculas y minúsculas



# Variables booleaas

- Solo dos valores posibles, True y False
- Operadores logicos and, or, not
- Las comparaciones producen resultados booleanos
- El operador de igualdad es ==
- El operador de diferente es !=
- Internamente, True se representa con 1 y False con 0



# Ejercicio 6 : El operador módulo

- Sabiendo que el operador módulo es %
- ¿Cómo podemos saber si el número 35184103522305 es divisible de forma exacta por 131071?



# Ejercicio 6 : El operador módulo

- Sabiendo que el operador módulo es %
- ¿Cómo podemos saber si el número 35184103522305 es divisible de forma exacta por 131071?

```
35184103522305 % 131071 == 0
```

# Cadenas de texto

- **Se tienen que delimitar con comillas, pero podemos usar 4 formas diferente:**
  - **Comillas simples**
  - **Comillas dobles**
  - **Triples comillas simples**
  - **Triples comillas dobles**
- **Los caracteres especiales se “escapan” con \**  
**(Por ejemplo, \n significa “salto de línea”)**



# Operaciones con cadenas de texto

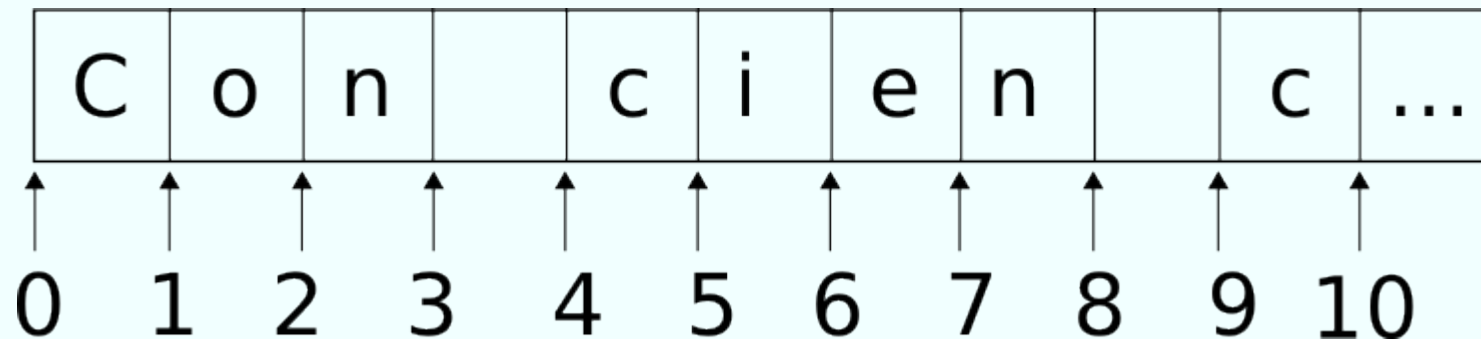
- Con la función `len` obtenemos su longitud
- Podemos acceder a su contenido mediante índices
- El primer carácter tiene índice 0
- El índice no es la posición, sino "El número de caracteres que hay antes del que me interesa"
- Podemos usar números negativos: El -1 es el último





# Rodajas o slices

- Podemos acceder a secciones con la sintaxis [índice inferior:índice superior]
- Podemos omitir el inferior, el superior o ambos
- Los escasos se aceptan con indulgencia



# Ejemplo de slices

```
>>> s = 'Con cien cañones por banda,'
>>> s[0:3] # primeros tres caracteres
'Con'
>>> s[:8] # primeros ocho caracteres
'Con cien'
>>> s[8:] # todo, excepto primeros 8 carac.
' cañones por banda,'
>>> s[4:8]
'cien'
>>> s[-6:]
'banda,'
```

# Estructuras de control: for

- La sentencia for permite recorrer una secuencia
- Las únicas secuencias que conocemos por ahora son las strings
- La forma es:

```
for <item> in <secuencia>:  
    # Hacemos cosas con item  
    print(item)
```

# Ejercicio 7: Imprimir caracteres

```
for c in 'Abracadabra':  
    print(c)
```

# Qué hemos aprendido

- La sentencia `for` itera sobre una secuencia
- La indentación de código en python no es opcional: tiene significado
- Todo lo que esta indentado debajo del `for` forma el bloque que se ejecuta repetidamente
- No se usa ningún índice. La variable del bucle es directamente la que corresponde a la secuencia



# ¿Y si necesito el índice?

- La función `range()` devuelve una secuencia de números, cuyos límites dependen de los parámetros suministrados:
  - `range(n)` devuelve el rango `[0..n-1]`
  - `range(a, b)` devuelve el rango `[a..(b-1)]`
  - `range(a, b, c)` devuelve el rango `[a..(b-1)]` pero en incrementos de `c`



# Ejercicio 8: Uso de range y slices

**Hacer un programa que genere la salida:**

```
A  
Ab  
Abr  
Abra  
Abrac  
Abraca  
Abracad  
Abracada  
Abracadab  
Abracadabr  
Abracadabra
```

# Ploting (gráficas)

- La forma más usada es matplotlib
- Usar el magic `%matplotlib inline` para que las gráficas se inserten en el notebook

