

# Módulo de adaptación

## Master en Business Intelligence y Big Data

**PROFESOR/A**  
Antonio Sarasa Cabezuelo

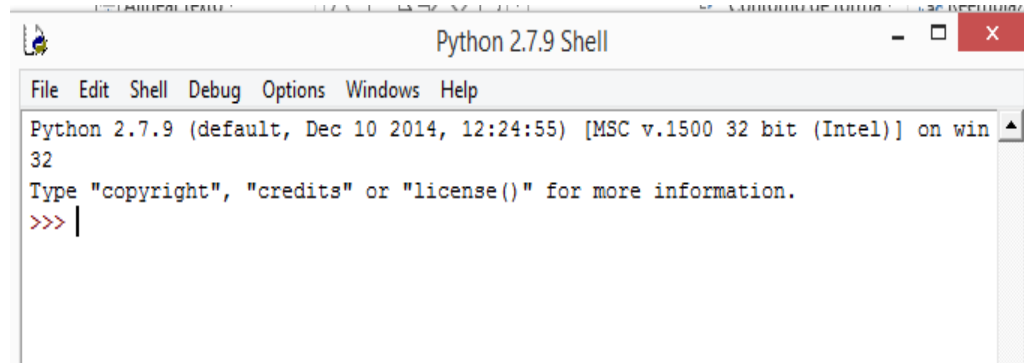
# Entornos de desarrollo para Python

# Entornos de desarrollo para Python

- Existen una gran variedad de entornos de desarrollo para trabajar con Python, de los cuales se van a comentar:
  - Idle
  - Anaconda
  - DataJoy

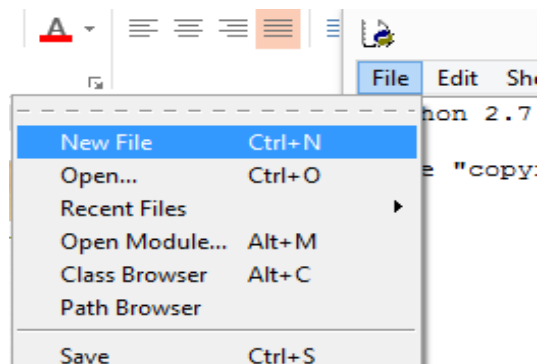
# Idle

- Es el entorno de desarrollo que se instala con Python.
- Cuando se ejecuta aparece una pantalla en la que se tiene un shell que interpreta las instrucciones de Python.



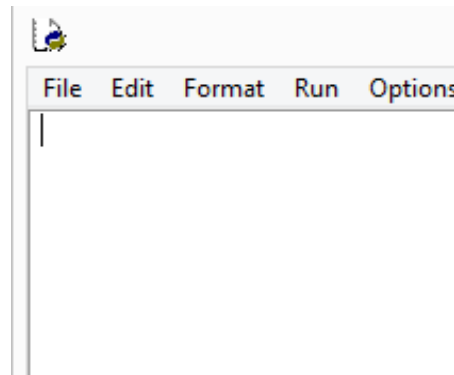
# Idle

- El shell dispone en la parte superior un conjunto de opciones de edición, depuración,...
- Sin embargo si se quieren crear programas que puedan ser almacenados en memoria se debe utilizar la opción “New File” del menú “File”



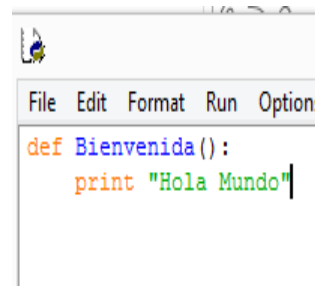
# Idle

- Cuando se pulsa sobre “New File” aparece una nueva pantalla que servirá para editar el programa en Python.



# Idle

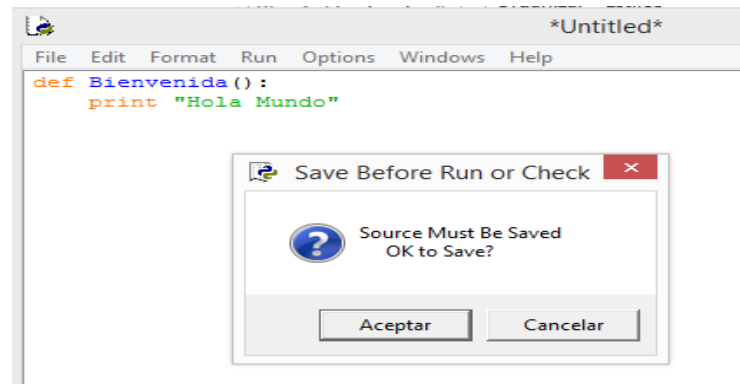
- En la pantalla de edición el programador escribe el programa que posteriormente quiere ejecutar. El editor dispone de coloreado de palabras reservadas, así como de ayuda contextual(cuando se escribe una instrucción se muestra información de la misma).



```
File Edit Format Run Options
def Bienvenida():
    print "Hola Mundo"
```

# Idle

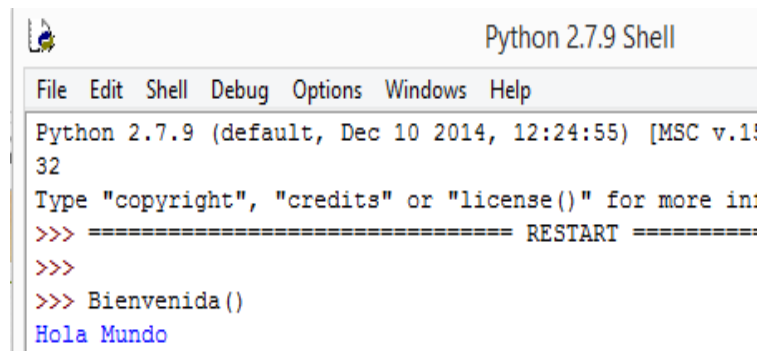
- Una vez que se ha escrito el programa se puede ejecutar. Para ello se pulsa el menú “Run” y sobre la opción “Run Module”. En ese instante el editor pide que guardemos el archivo en disco.





# Idle

- Una vez almacenado, el editor devuelve el control a la pantalla del shell, desde la cual se puede ejecutar el programa.



```
Python 2.7.9 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.132
Type "copyright", "credits" or "license()" for more in:
>>> ===== RESTART =====
>>>
>>> Bienvenida()
Hola Mundo
```

# Anaconda

- Es un potente entorno de desarrollo para Python que puede descargarse de:

<http://continuum.io/downloads>

- Existen versiones de Anaconda para los principales sistemas operativos. En las transparencias se ilustrarán para un entorno Windows.

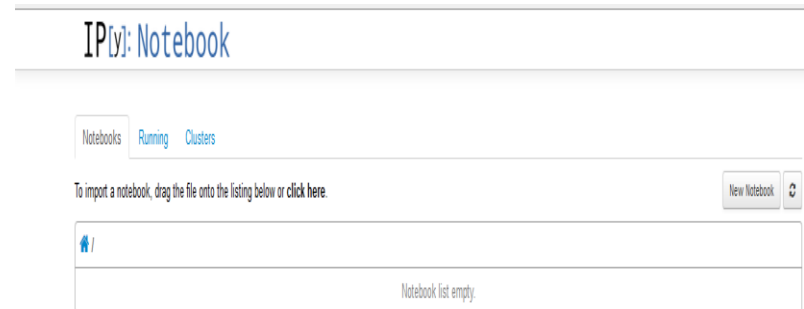
# Anaconda

- La instalación de Anaconda permite acceder entre otras a las siguientes herramientas:
  - IPython Notebook.
  - Spyder

# IPython Notebook

- Es una herramienta web de desarrollo a la que se puede acceder pulsando sobre el programa IPython(Py 2.7) Notebook, tras lo cual aparece en el navegador. En caso de no ejecutarse el navegador se puede encontrar en la dirección <http://localhost/8888>

```
IPython (Py 2.7) Notebook
2015-09-22 22:30:50.828 [NotebookApp] Using existing profile dir: u'C:\Users\asarasa\ipython\profile_default'
2015-09-22 22:30:50.877 [NotebookApp] Using MathJax from CDN: https://cdn.mathjax.org/mathjax/latest/MathJax.js
2015-09-22 22:30:51.464 [NotebookApp] Serving notebooks from local directory: C:\Users\asarasa\Documents\IPython Notebooks
2015-09-22 22:30:51.467 [NotebookApp] 0 active kernels
2015-09-22 22:30:51.467 [NotebookApp] The IPython Notebook is running at: http://localhost:8888/
2015-09-22 22:30:51.469 [NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

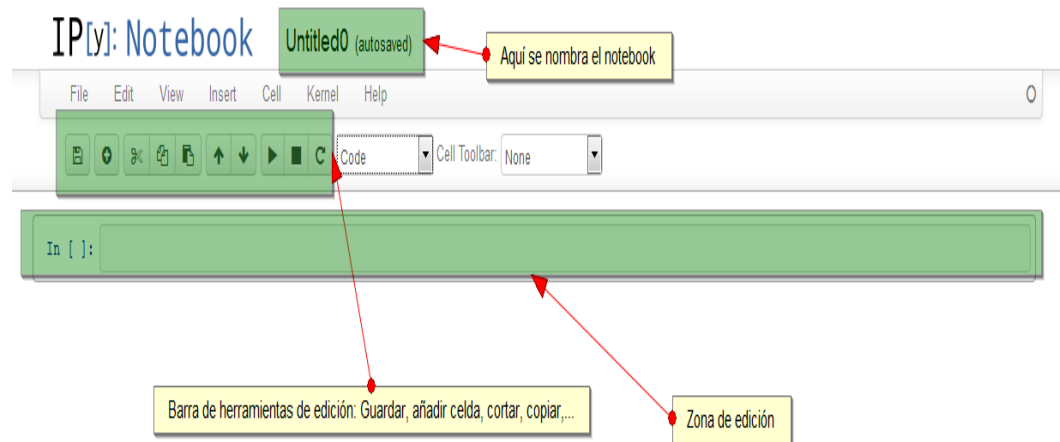


# IPython Notebook

- La principal función de esta herramienta es la creación de “notebooks”, documentos en los que aparecen trozos de código que pueden estar documentados, así como ejecuciones del código.
- Un notebook puede ser almacenado, exportado a varios formatos e intercambiado.

# IPython Notebook

- Para crear un nuevo “notebook” se pulsa sobre “New Notebook”, y aparece la interface de edición del notebook.



# IPython Notebook

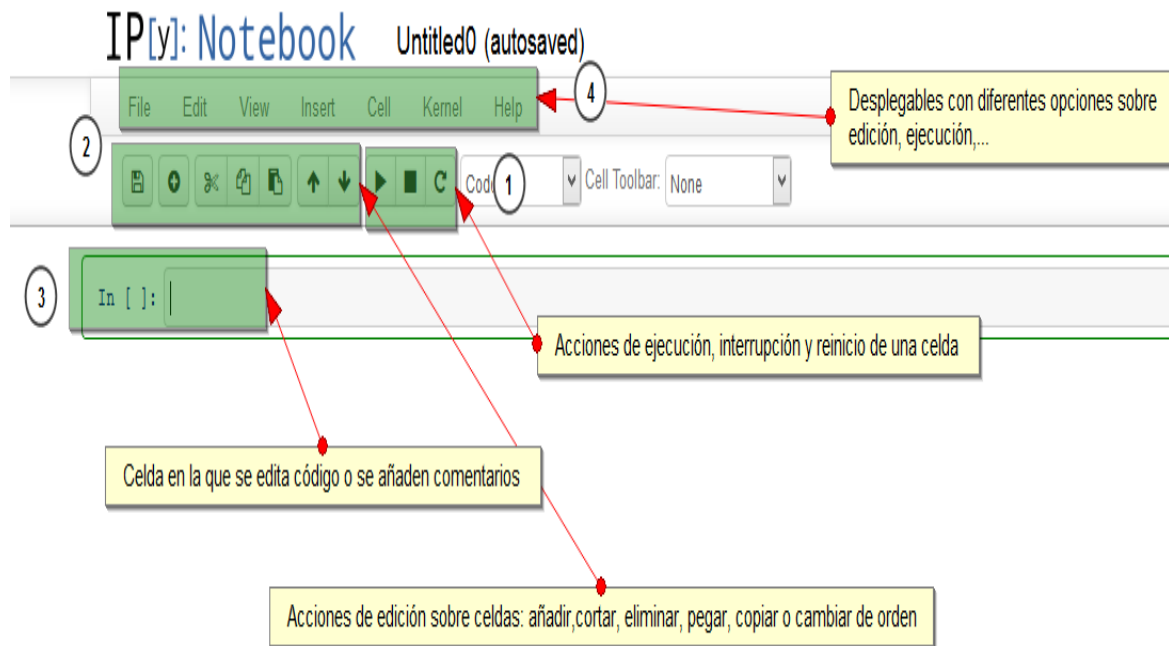
- La unidad de edición de un notebook es la celda. En ella se introduce el código en Python o bien la información que documenta dicho código.
- Para ejecutar el código que se introduce en una celda es necesario pulsar sobre la flecha de la barra de herramientas, y para parar la ejecución sobre el cuadrado.

# IPython Notebook

- Además sobre las celdas se pueden realizar determinadas acciones de actualización:
  - Cortar celdas
  - Pegar celdas.
  - Eliminar celdas.
  - Cambiar el orden de las celdas.



# IPython Notebook



# IPython Notebook

- En el notebook se diferencian diferentes tipos de celdas, según el contenido.
- Junto a las opciones de ejecución de una celda, aparece un desplegable con los tipos de celdas:
  - Code: Código en python.
  - Markdown: Permite escribir texto formateado con el objetivo de documentar usando el lenguaje de marcas Markdown.
  - Raw Text: Igual que el anterior permite escribir texto sin formato.
  - Heading: Permite embeber código html.

# IPython Notebook

- El flujo normal de edición de una celda consiste en:
  - Elegir el tipo de celda. Por defecto son de tipo “Code”. Dependiendo el tipo de celda, IPython lo interpreta de diferente manera.
  - Una vez introducido el código o texto en la celda, se debe ejecutar. Para ello se pulsa sobre el icono en forma de flecha.
  - A continuación se genera una nueva celda para editar.

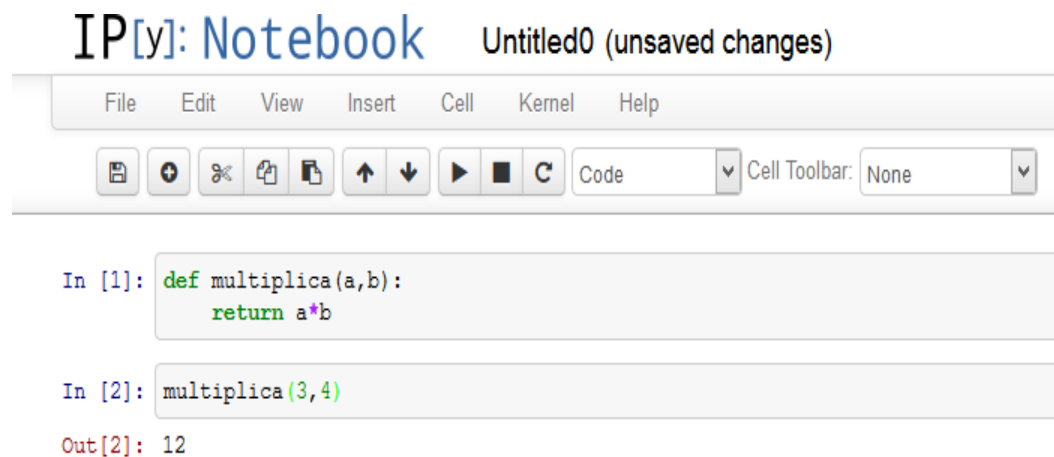
# IPython Notebook

- Por ejemplo si se quiere crear una celda que contenga la función:

```
def multiplica(a,b):  
    return a*b
```

- Y a continuación se quiere multiplicar 3 por 4:  
multiplica(3,4)

# IPython Notebook



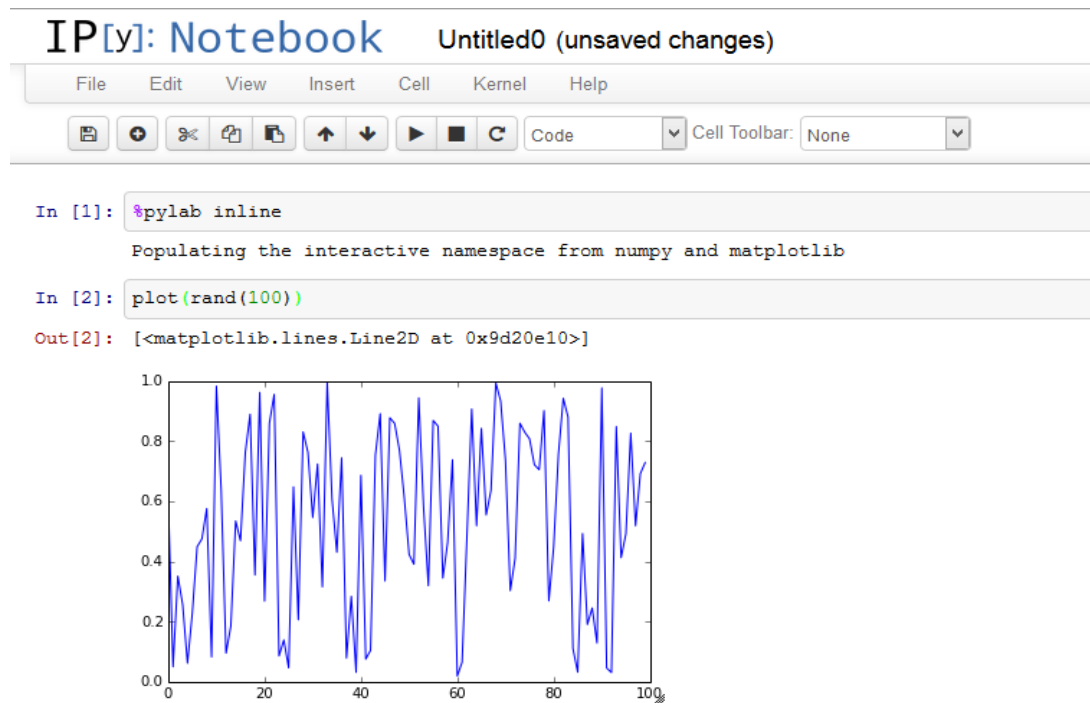
The screenshot displays the IPython Notebook interface. At the top, the title bar reads "IP[y]: Notebook" and "Untitled0 (unsaved changes)". Below this is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, and Help. A toolbar follows, containing icons for saving, opening, closing, and running cells, as well as a dropdown menu for the cell type (currently set to "Code") and a "Cell Toolbar" dropdown (set to "None"). The main area shows two input cells. The first cell, labeled "In [1]:", contains a Python function definition: `def multiplica(a,b):` followed by an indented `return a*b`. The second cell, labeled "In [2]:", contains the function call `multiplica(3,4)`. Below the second cell, the output is displayed as "Out[2]: 12".

```
In [1]: def multiplica(a,b):  
        return a*b  
  
In [2]: multiplica(3,4)  
Out[2]: 12
```

# IPython Notebook

- Una de las operaciones típicas con el notebook de IPython consiste es la representación gráfica de datos mediante matplotlib. Para ello se puede hacer de varias formas:
  - Opción 1. Usando solo pylab
    - Se introduce en la celda: `%pylab inline`.
    - En las siguientes líneas se puede llamar a las funciones de representación gráfica

# IPython Notebook

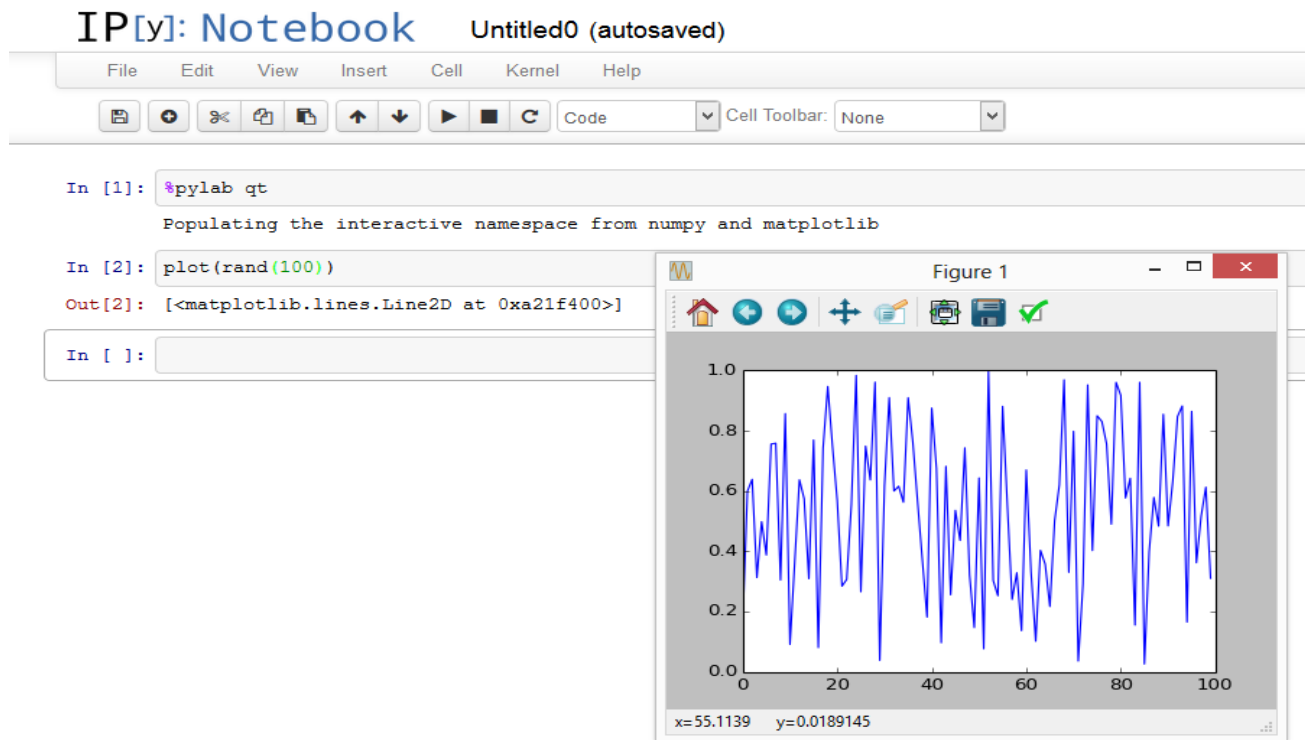


# IPython Notebook

- Opción 2. Usando las librerías QT
  - Se introduce en la celda: `%pylab qt.`
  - En las siguientes líneas se puede llamar a las funciones de representación gráfica



# IPython Notebook



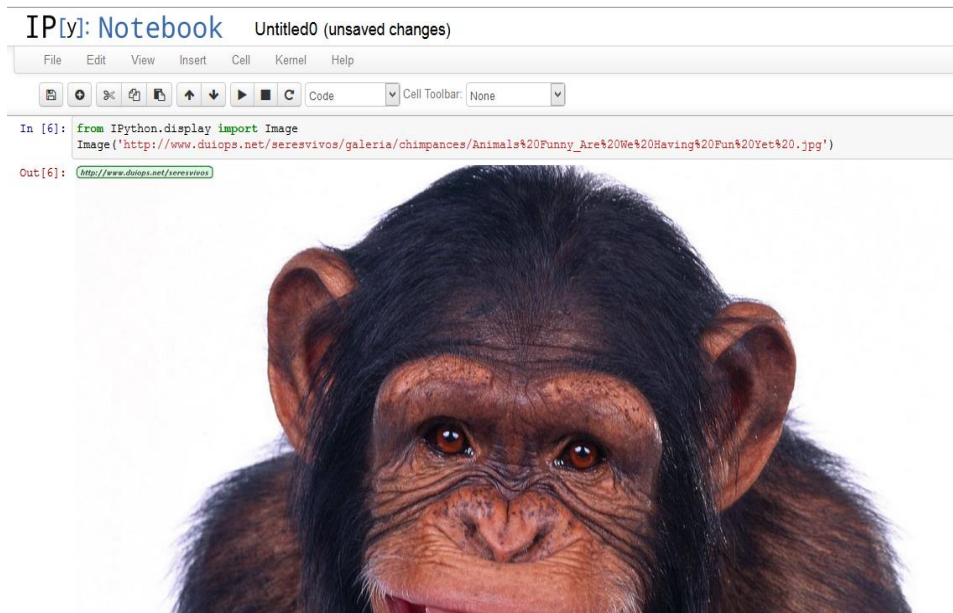
# IPython Notebook

- Otras utilidades:
  - Insertar videos de Youtube.



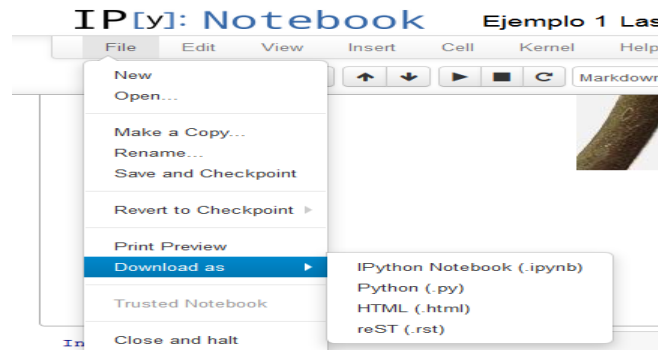
# IPython Notebook

- Insertar imagenes.



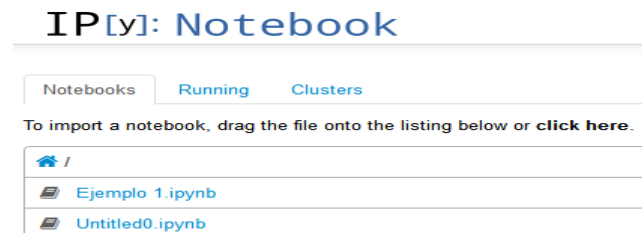
# IPython Notebook

- Una opción muy interesante de IPython es la posibilidad de importar y exportar los notebooks. De esta forma se facilita compartir y reutilizarlos.
- Para exportar un notebook, basta tenerlo abierto, y usar el menú File.



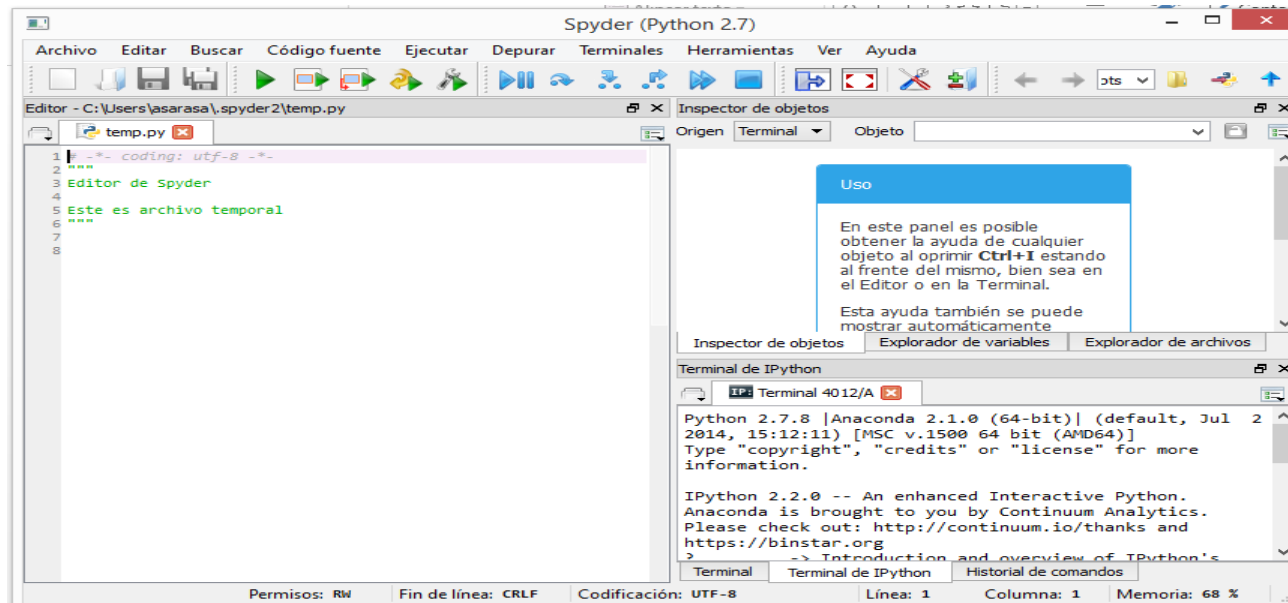
# IPython Notebook

- Se selecciona Download as, y dentro de ese desplegable se elige la opción de exportación. En general se elegirá IPython Notebook(.ipnb).
- Para importar un notebook, en la página principal de IPython, dónde aparecen listados todos los notebooks, basta arrastrar el notebook que se quiere importar.



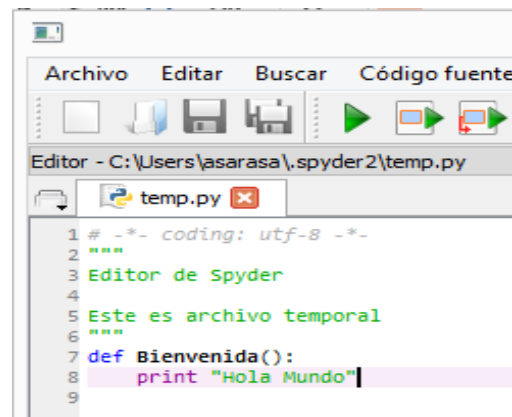
# Spyder

- Es un entorno de desarrollo para Python basado en IPython. Se accede pulsando sobre el programa Spyder.



# Spyder

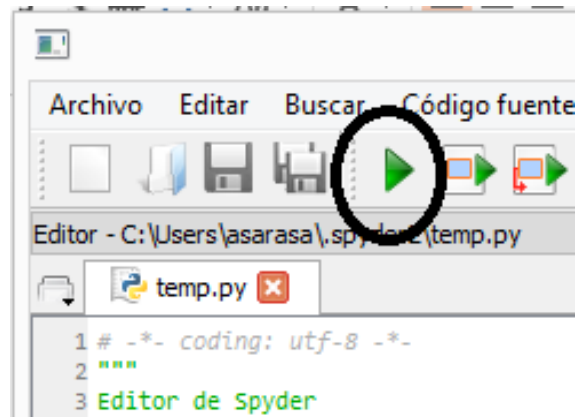
- Dispone de una gran variedad de herramientas de edición, depuración,...
- El uso básico del entorno consiste en:
  - Escribir el programa en Python en la ventana izquierda



```
1 # -*- coding: utf-8 -*-
2 """
3 Editor de Spyder
4
5 Este es archivo temporal
6 """
7 def Bienvenida():
8     print "Hola Mundo"
9
```

# Spyder

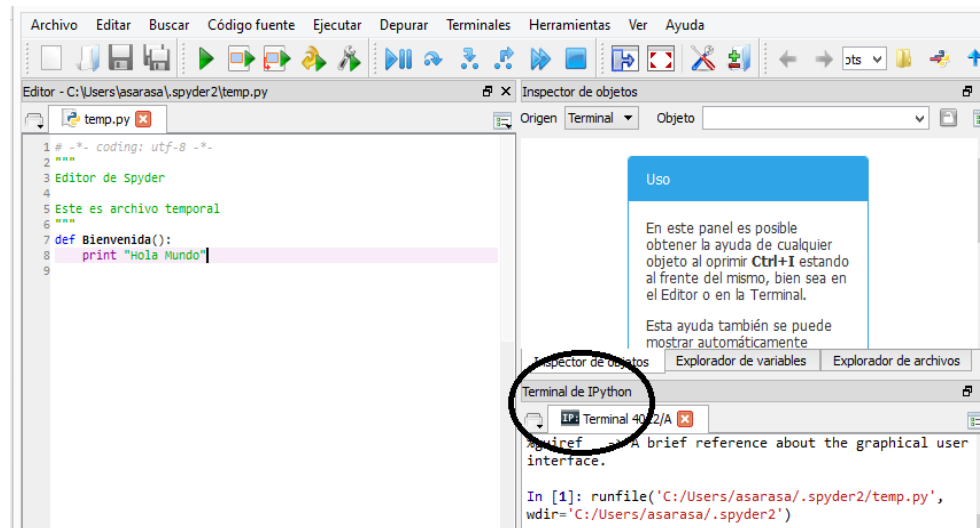
- A continuación se pulsa sobre el botón que representa una flecha verde





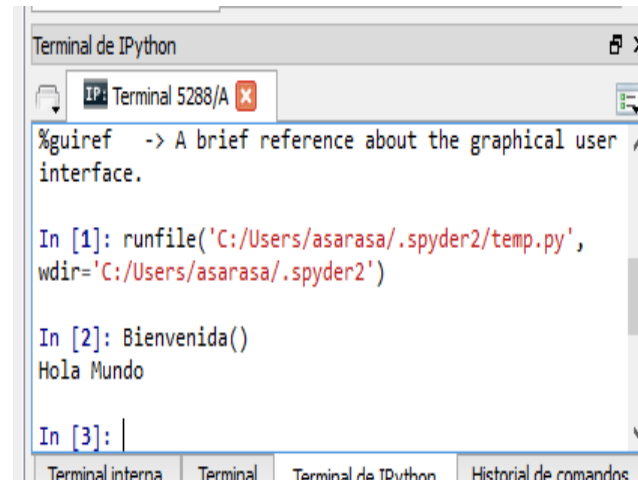
# Spyder

- Al pulsar sobre el botón verde se interpreta el programa en Python, y ya es posible ejecutarlo en la ventana inferior derecha, la cual se trata de un terminal de IPython.



# Spyder

- En el terminal IPython ejecutamos el programa



The screenshot shows a terminal window titled "Terminal de IPython". It contains the following text:

```
%gui ref -> A brief reference about the graphical user interface.
```

```
In [1]: runfile('C:/Users/asarasa/.spyder2/temp.py',  
wdir='C:/Users/asarasa/.spyder2')
```

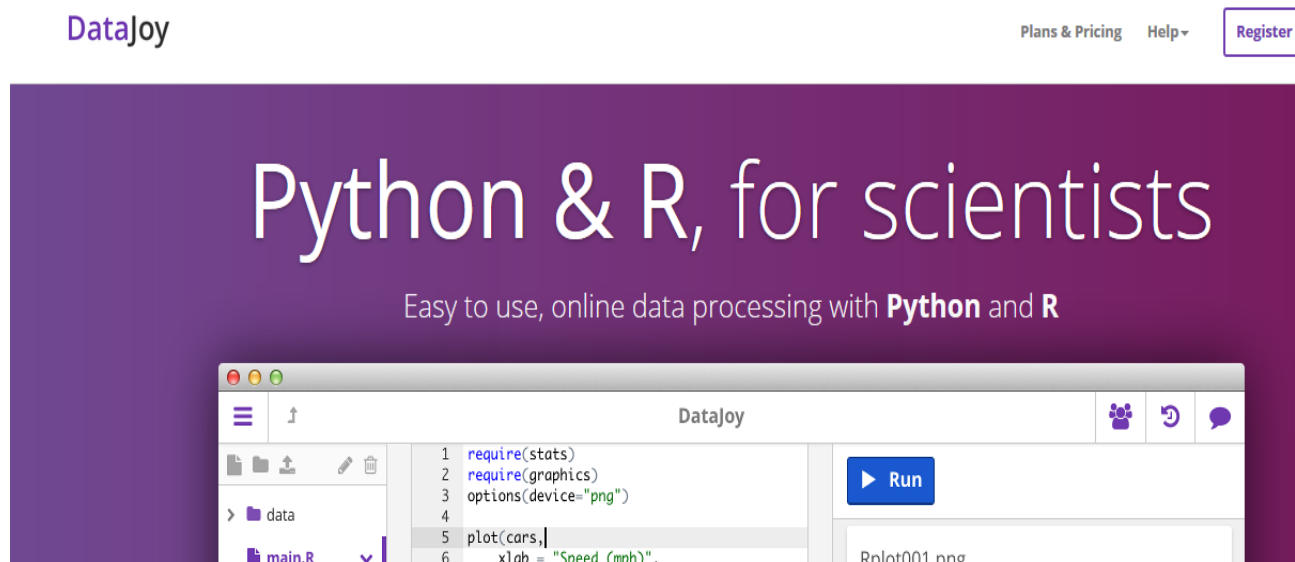
```
In [2]: Bienvenida()  
Hola Mundo
```

```
In [3]:
```

At the bottom of the window, there are four tabs: "Terminal interna", "Terminal", "Terminal de IPython", and "Historial de comandos".

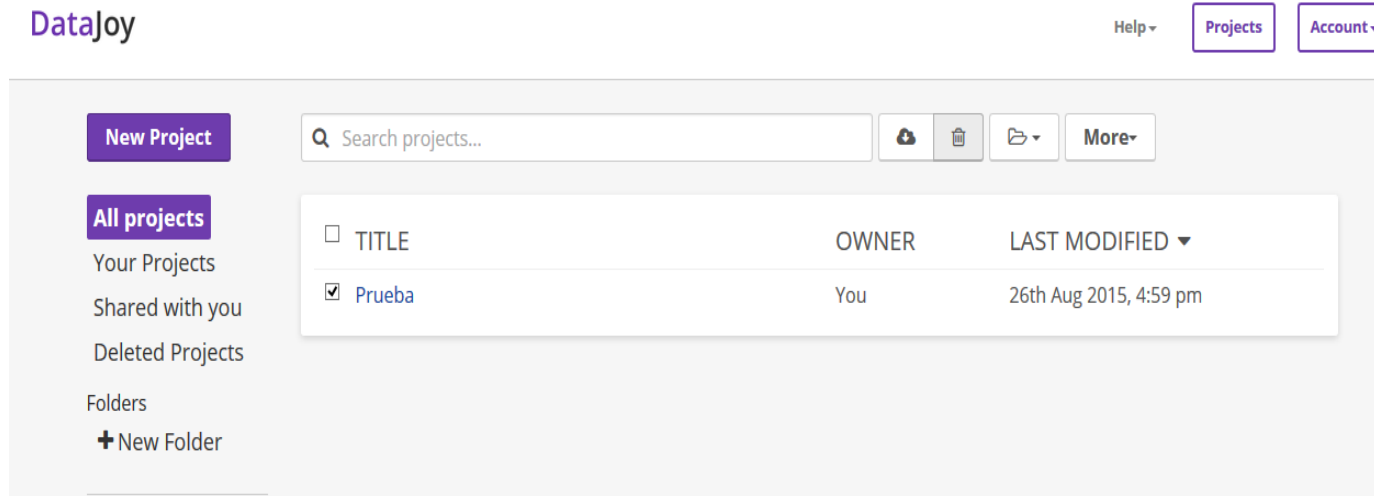
# DataJoy

- Es un entorno de desarrollo gratuito online que se puede encontrar en la siguiente página:  
<https://www.getdatajoy.com/>



# DataJoy

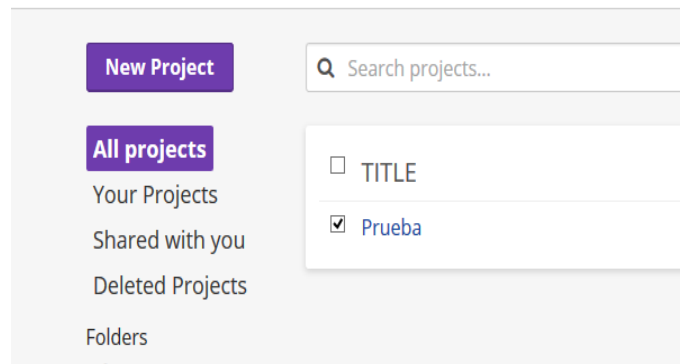
- Hay que registrarse para poder usarlo.
- Una vez dentro del sistema, aparece una pantalla de gestión de proyectos.



# DataJoy

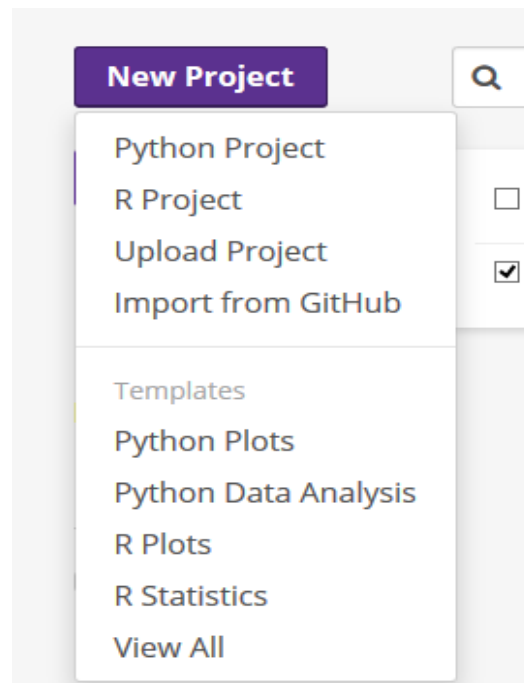
- En dicha pantalla existen diferentes opciones: crear carpetas, bajarse proyectos, eliminar, hacer copias, renombrar,...
- Para crearse un nuevo proyecto se pulsa sobre “New Project”

DataJoy



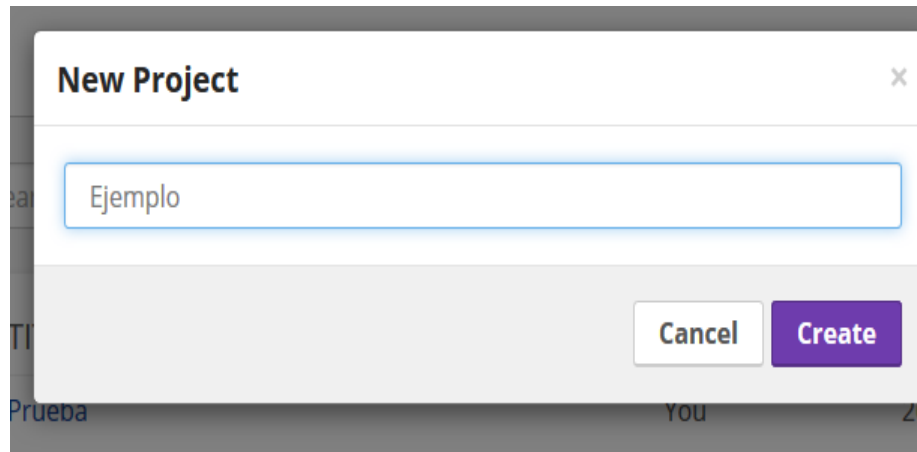
# DataJoy

- Al pulsar sobre “New Project” aparecen las distintas posibilidades



# DataJoy

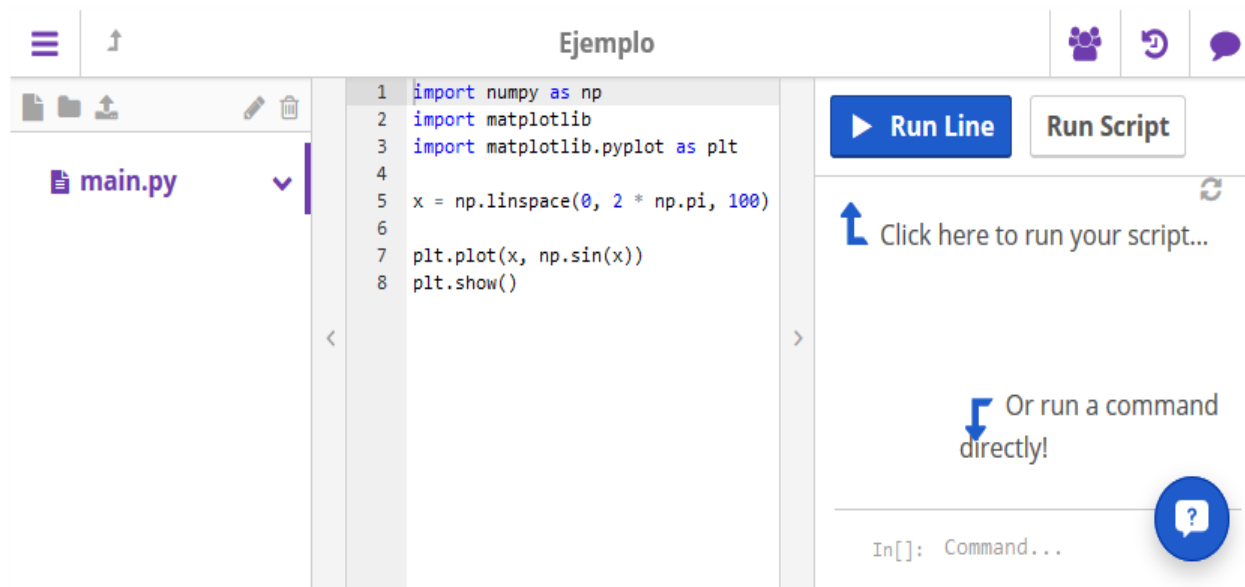
- Se selecciona la opción “Python Project”, y pide un nombre para el mismo.



The screenshot shows a 'New Project' dialog box. The title bar reads 'New Project' with a close button (X) on the right. Below the title bar is a text input field containing the word 'Ejemplo'. At the bottom right of the dialog are two buttons: 'Cancel' and 'Create'.

# DataJoy

- Al crearse el proyecto aparece una pantalla de edición



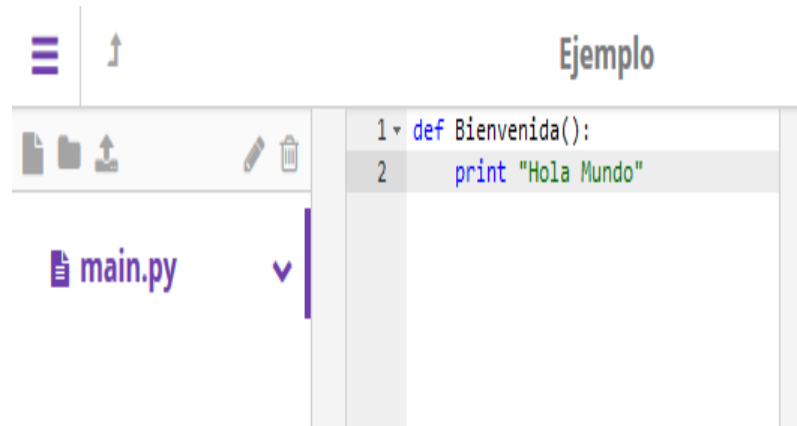


# DataJoy

- En la pantalla de edición aparece un código de ejemplo que se puede eliminar, el nombre por defecto del programa en Python puede ser renombrado, y hay diferentes opciones de ejecución.

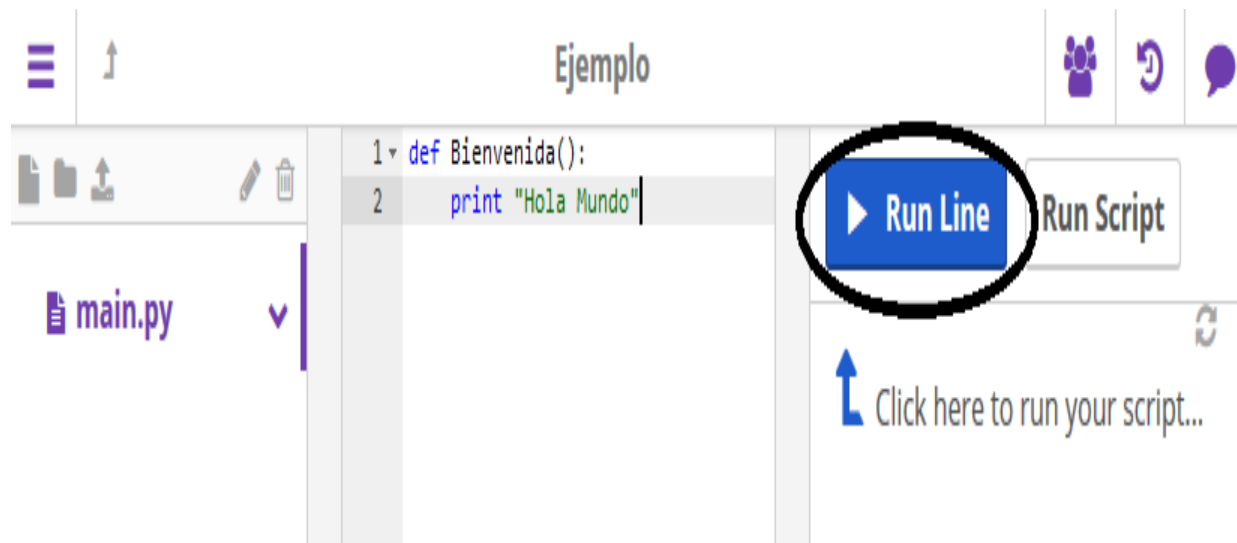
# DataJoy

- Si se quiere escribir un programa en Python, entonces en el marco central de la ventana escribimos el código.



# DataJoy

- A continuación se pulsa sobre el botón “Run Line”



# DataJoy

- En el marco derecho de la ventana aparece la ejecución del programa.



The screenshot displays the DataJoy IDE interface. On the left, a code editor titled 'Ejemplo' contains the following Python code:

```
1 def Bienvenida():  
2     print "Hola Mundo"
```

On the right, there is a panel for running code. It features a blue 'Run Line' button and a white 'Run Script' button. Below these buttons, the execution output is shown:

```
In[1]: print "Hola Mundo"  
  
Out[1]: Hola Mundo
```