

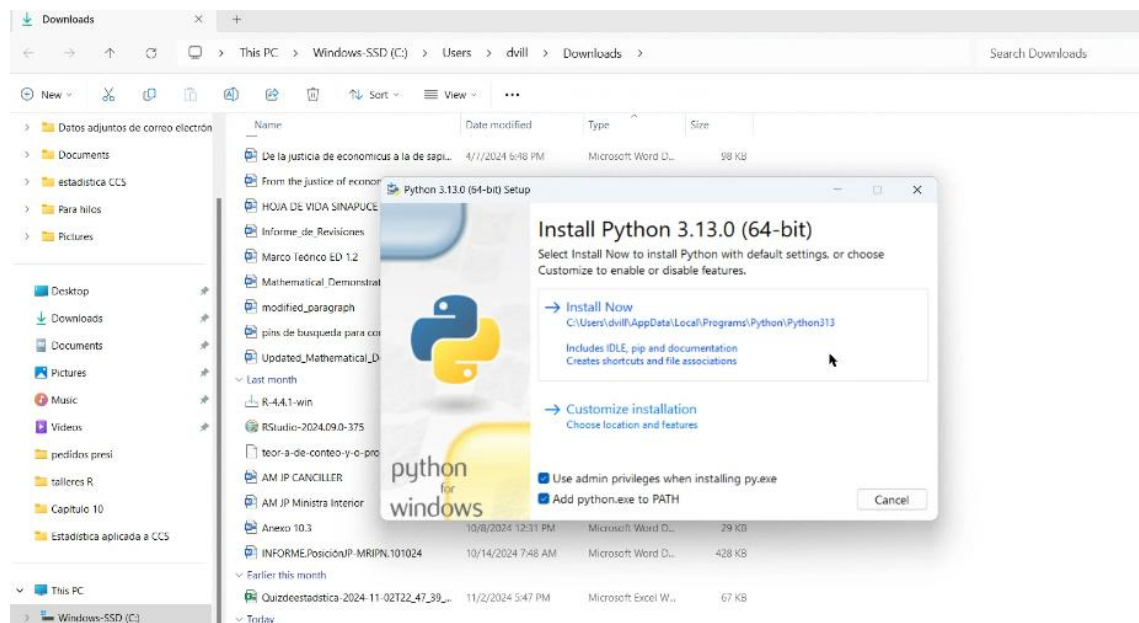
## Creación de entorno de Anaconda para VSC (versión 2024.11.16)

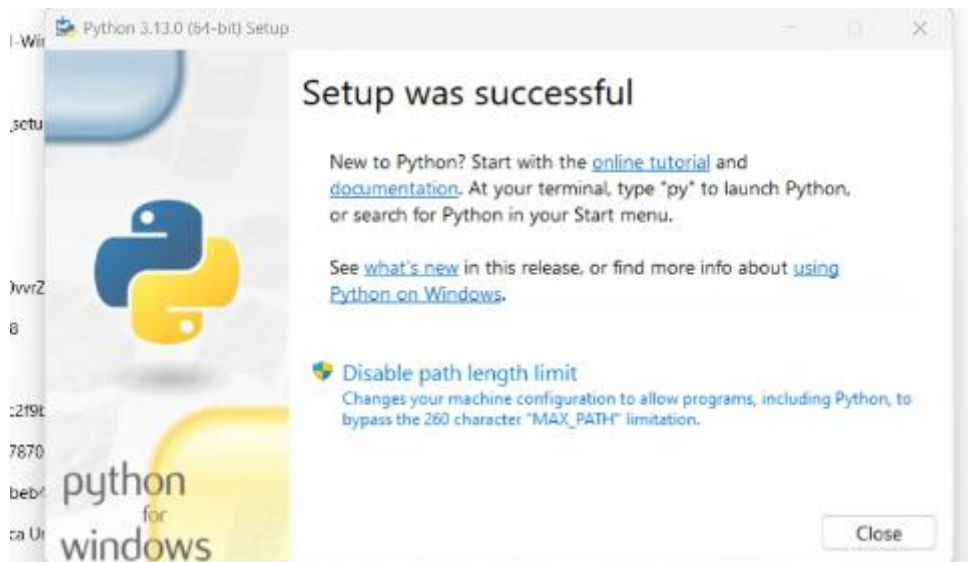
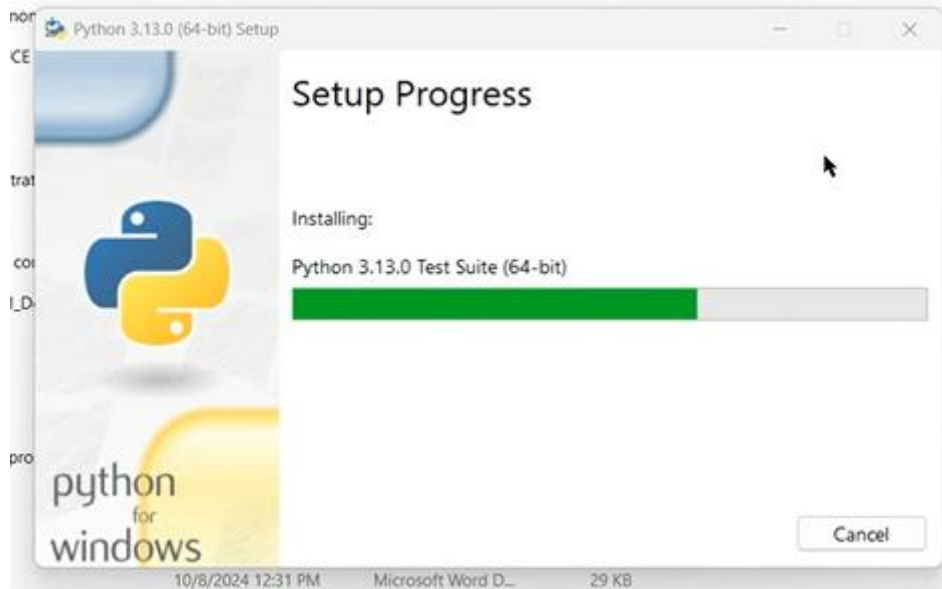
### 0. Instalar Python, Anaconda (Asegurarse de añadir Path en los dos programas)

- <https://www.python.org/downloads/>



Luego de seleccionar las opciones marcadas poner Install Now





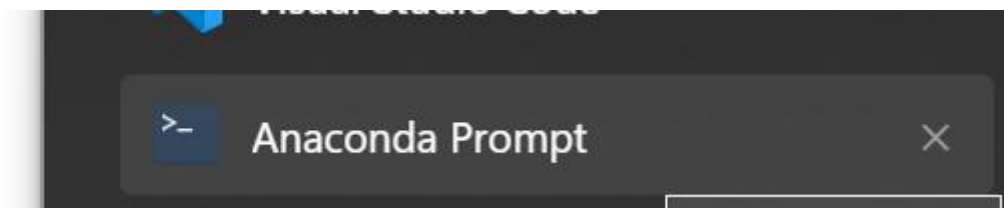
Al buscar aparece instalado



- <https://anaconda.org/> (crearse una cuenta y descargar)

## 0. Crear entorno en Anaconda

Abrimos Anaconda Prompt



Y ejecutamos el siguiente comando, con el cual le asignamos un nombre al ambiente en este caso “mate”, también podemos especificar una versión específica de Python que necesitamos de ser necesario ahí deberíamos correr (`conda create --name mate python=3.10`) por ejemplo, pero vamos a correr simplemente:

```
conda create --name mate python=3.10
```

```
Anaconda Prompt
(base) C:\Users\dvill>conda create --name mate python=3.10|
```

Luego damos en "y" y la tecla Enter para confirmar

```
Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate mate
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

Con esto nos indica también cómo activar o desactivar el ambiente.

```
(mate) C:\Users\econo>python --version
Python 3.12.7
```

## 1. Activar el entorno

En Anaconda Prompt corremos el siguiente código, para activar el entorno que llamé "mate" y damos Enter.

conda activate mate

```
(base) C:\Users\econo>conda activate mate
```

```
(mate) C:\Users\econo>|
```

Se puede ver ahora que ya no estoy en "base" sino en mi entorno "mate".

Podemos verificar la versión de Python que estamos usando en nuestro ambiente:

con el comando:

python --version

```
(mate) C:\Users\dvill>python --version
Python 3.10.15
```

Cuando termine de trabajar en el ambiente creado y quiero crear o entrar en otro puedo desactivarlo (aquí se puede ver como pasamos de “mate” a “base” al desactivarlo)

```
(mate) C:\Users\econo>conda deactivate

(base) C:\Users\econo>|
```

Podemos ver los entornos que tenemos con: “conda env list”

```
(mate) C:\Users\econo>conda env list
# conda environments:
#
base                  C:\ProgramData\anaconda3
mate                  * C:\Users\econo\.conda\envs\mate
                     C:\anaconda3
```

Si se desea borrar un ambiente creado salimos del ambiente creado y vamos al ambiente base y luego borramos

conda deactivate

conda remove --name mate --all

```
(mate) C:\Users\dvill>conda deactivate

(base) C:\Users\dvill> conda remove --name mate --all

Remove all packages in environment C:\Users\dvill\.conda\envs\mate:

## Package Plan ##

  environment location: C:\Users\dvill\.conda\envs\mate

The following packages will be REMOVED:

  anyio-4.6.2-py310haa95532_0
  argon2-cffi-21.3.0-pyhd3eb1b0_0
```

Y presionamos en “y”

```
Proceed ([y]/n)? y
```

```
Proceed ([y]/n)? y
```

```
Everything found within the
s and any non-conda files, w
(y/[n])? y

(base) C:\Users\dvill>
```

Si vuelvo a verificar los ambientes se puede ver que ya no existe el ambiente mate

```
(base) C:\Users\dvill>conda env list
# conda environments:
#
base * C:\ProgramData\anaconda3
```

Ahora vamos a crear el ambiente mega

conda create --name mega python=3.10

```
(base) C:\Users\dvill>conda create --name mega python=3.10
```

```
Proceed ([y]/n)? y
```

Y activo el nuevo ambiente

```
(base) C:\Users\dvill>conda activate mega
(mega) C:\Users\dvill>
```

## 2. Instalar las librerías necesarias en mi entorno

Correr este código en Anaconda Prompt e ir confirmando con “y”

conda install pandas numpy matplotlib jupyter ipykernel sympy  
ipywidgets nodejs

```
(mega) C:\Users\dvill>conda install pandas numpy matplotlib jupyter ipykernel sympy ipywidgets nodejs
```

```
zstd pkgs/
Proceed ([y]/n)? |
```

Y se empiezan a descargar las librerías necesarias:

```

Downloading and Extracting Packages:
mkl-2023.1.0           | 155.9 MB | #1
qt-main-5.15.2         | 59.4 MB  | ##5
icu-73.1               | 29.5 MB  | ####
libclang13-14.0.6     | 22.5 MB  | #####6
python-3.12.7          | 16.4 MB  | #####8
pandas-2.2.2           | 14.3 MB  |
openssl-3.0.15         | 7.8 MB   |
matplotlib-base-3.9.0 | 7.8 MB   |
numpy-base-1.26.4     | 6.6 MB   |
pyqt-5.15.10          | 3.7 MB   |
tk-8.6.14              | 3.5 MB   |

```

```

Proceed ([y]/n)? y

```

```

Downloading and Extracting Packages:

```

```

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

```

Por favor no poner esto

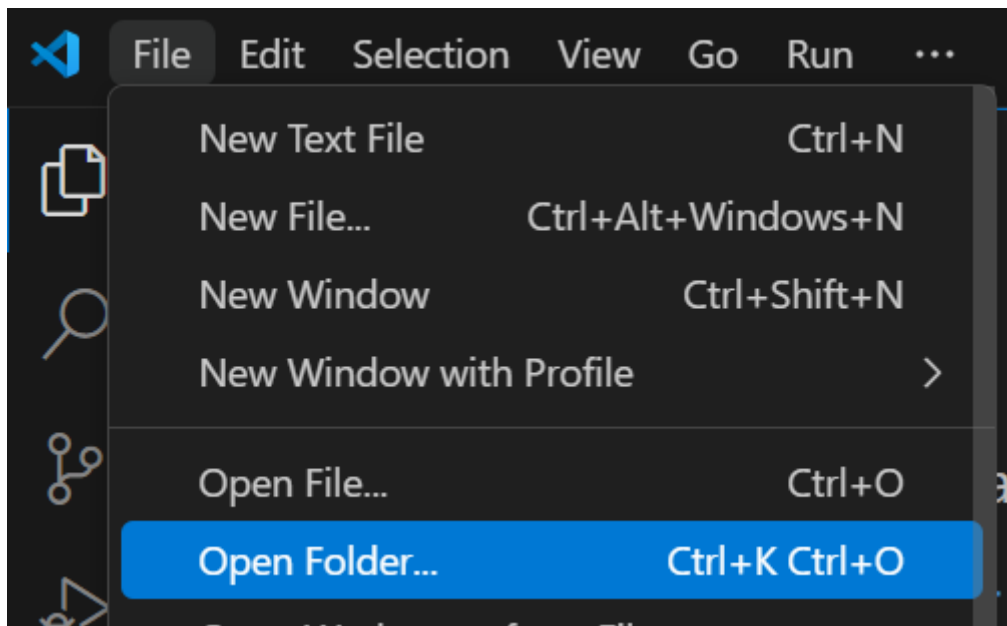
```

(mega) C:\Users\dvill>conda destroy the world|

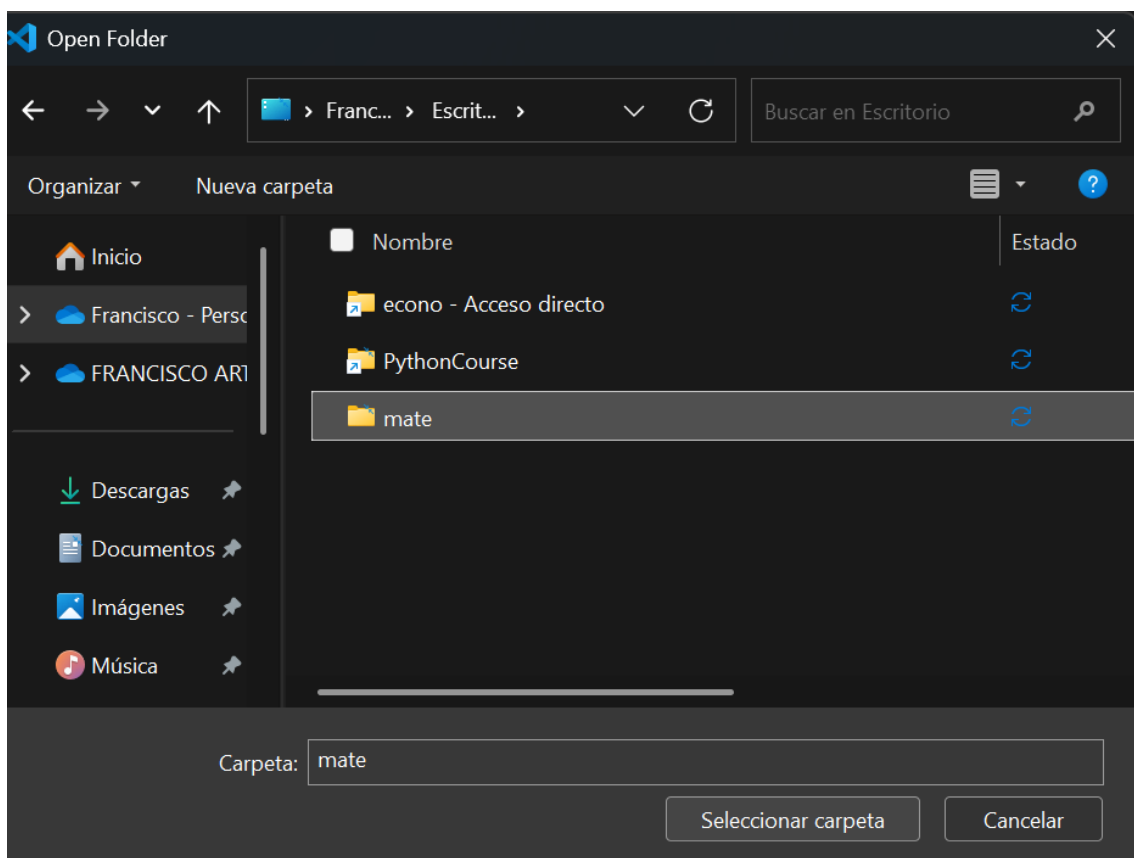
```

### 3. Abrir un directorio en VSC y utilizar el ambiente creado

Abrimos VSC y abrimos una carpeta donde guardaremos los archivos a utilizar:



En mi caso he creado una carpeta llamada “mate” en mi escritorio y selecciono esta carpeta

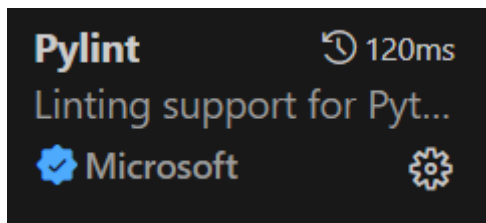
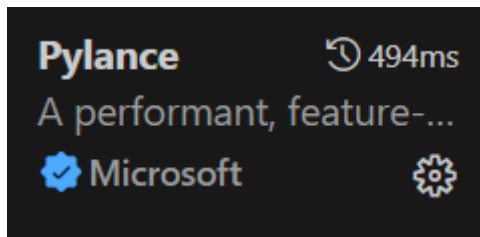
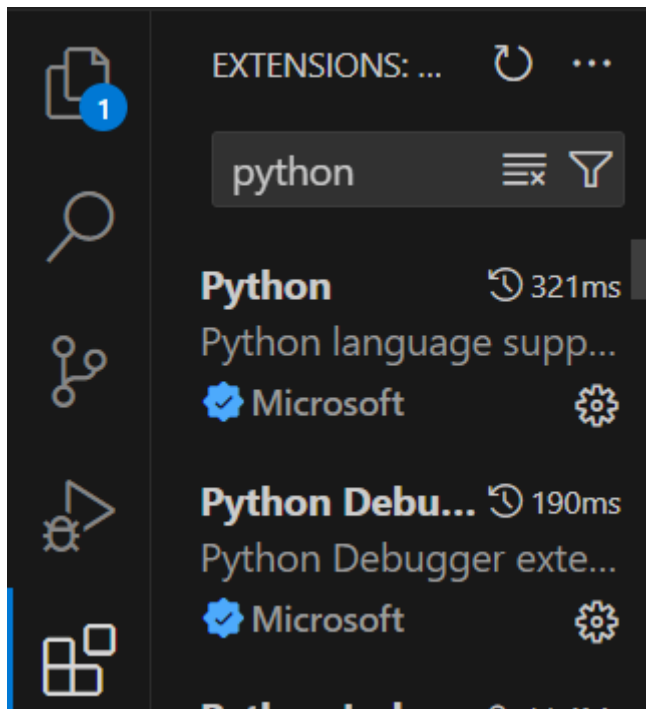


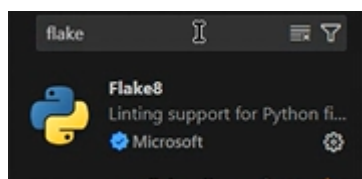
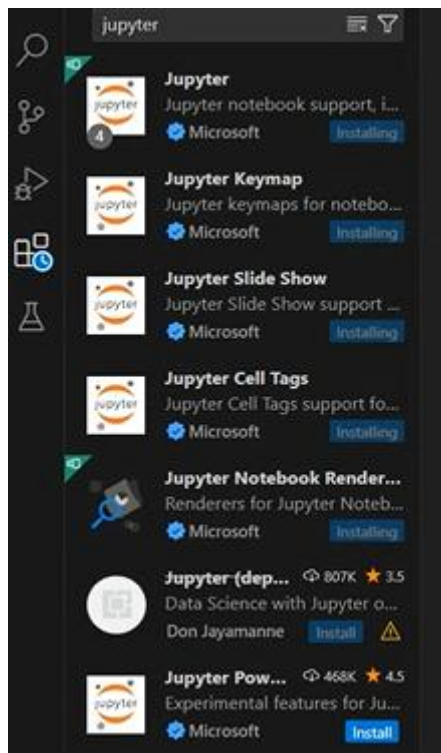
#### 4. Instalar extensiones de VSC para trabajar con Python y Jupyter Notebooks



Al lado izquierdo seleccionar e instalar las extensiones de Python y Jupyter Notebooks,

dando click en el botón de los tres cuadrados

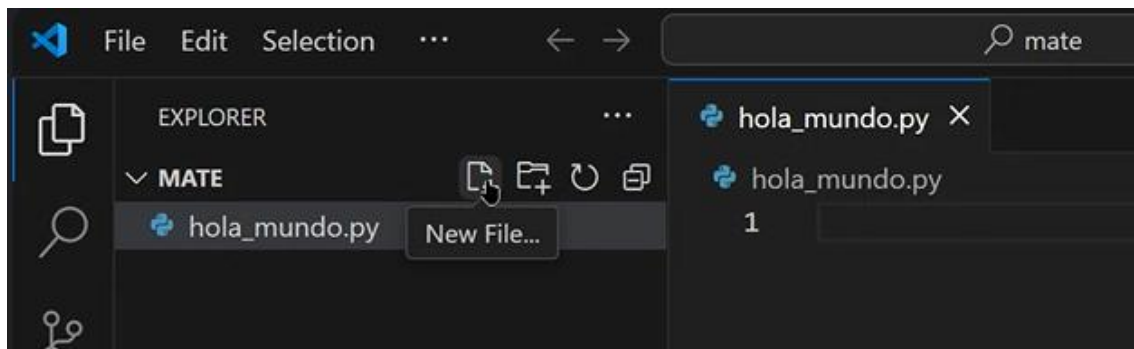




Otras extensiones automáticamente aparecerán y hay que instalarlas

Activar el mejorar el formato al guardar los archivos.

Creamos un nuevo archivo hola\_mundo.py



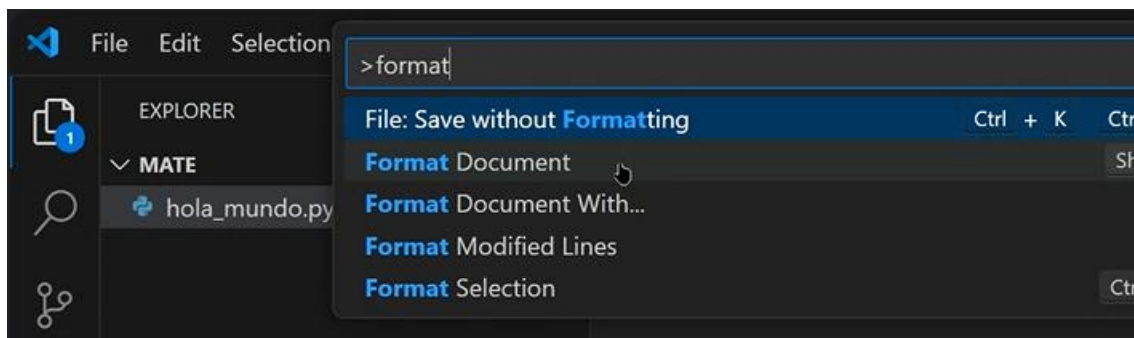
Y ponemos por ejemplo

```
a=3
```

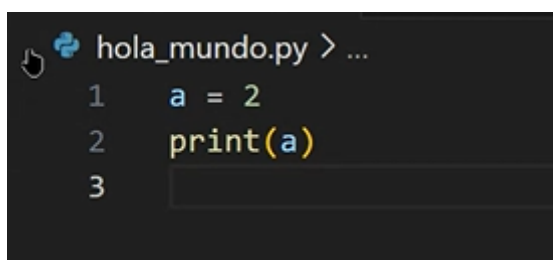
```
print(a)
```

y para que se habilite el cambio de formato Shift + Ctrl + P (o Shift + Command + P en macOS)

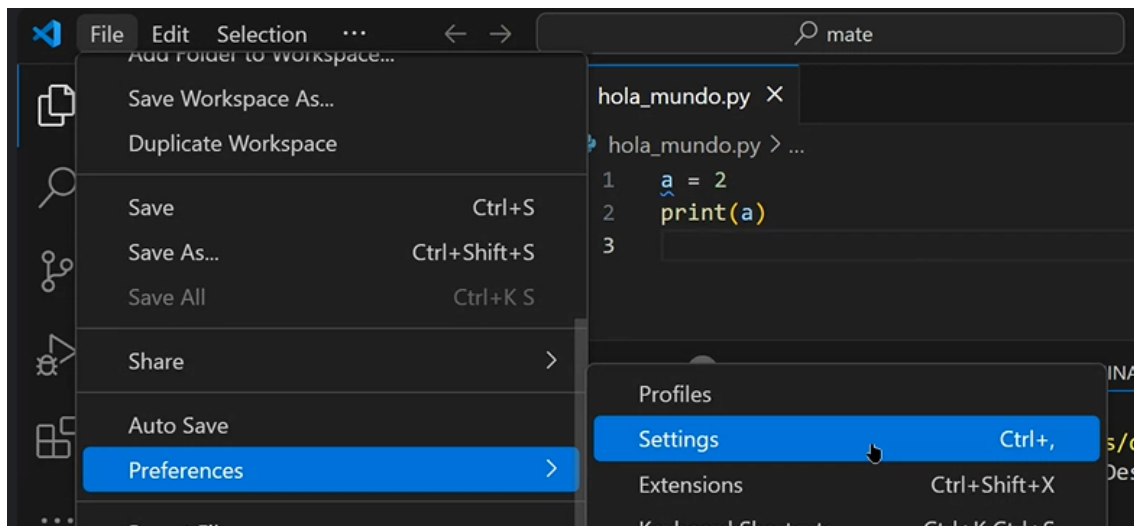
escribimos Format Document



Y se mejora el formato del código

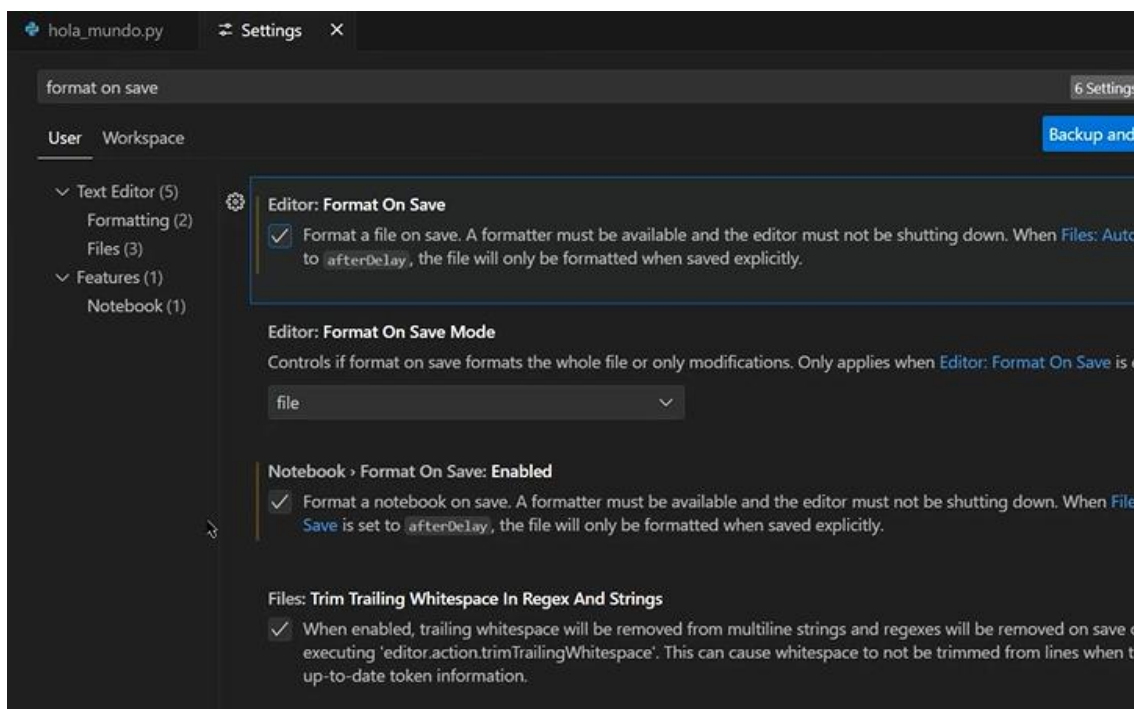


Para que se habilite el mejorar formato de código al guardar

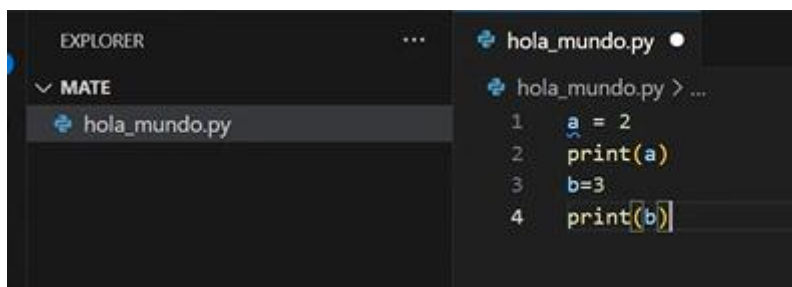


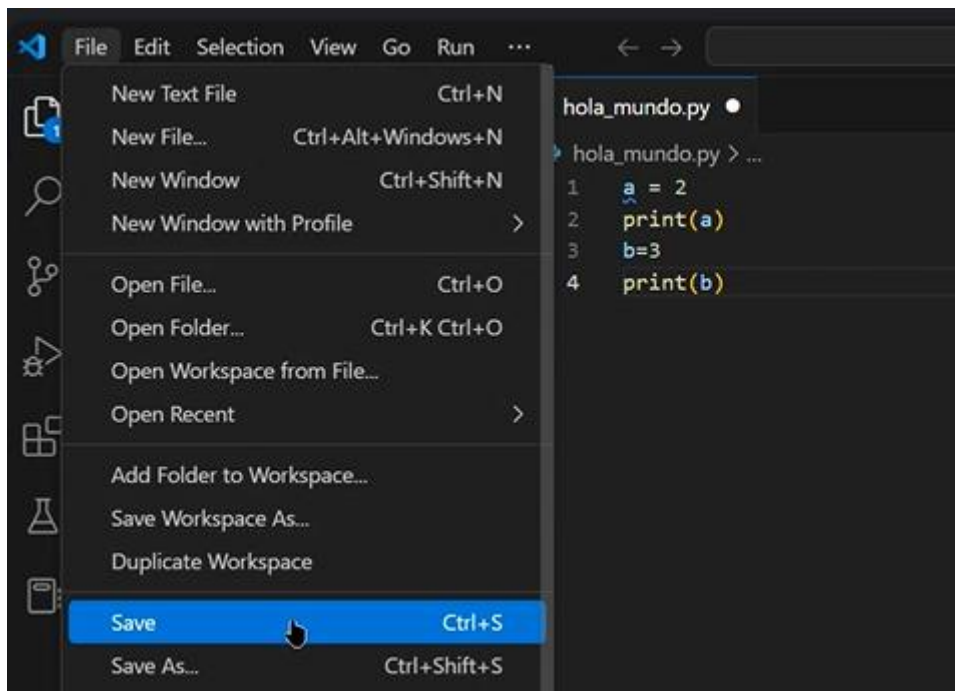
## Buscas

1. En la barra de búsqueda, escribe "formatOnSave".
2. Activa la opción **Editor: Format On Save**.

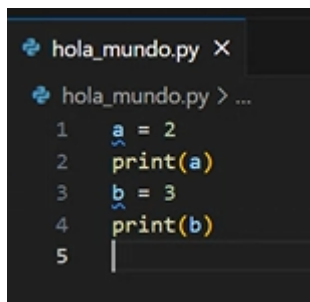


Así al guardar Ctrl+S

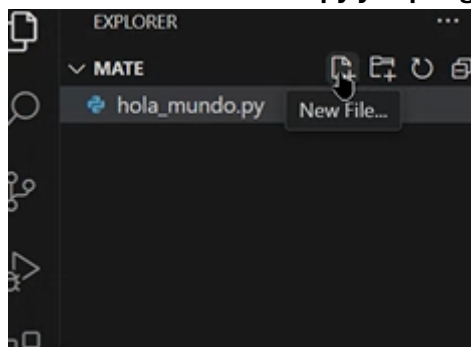


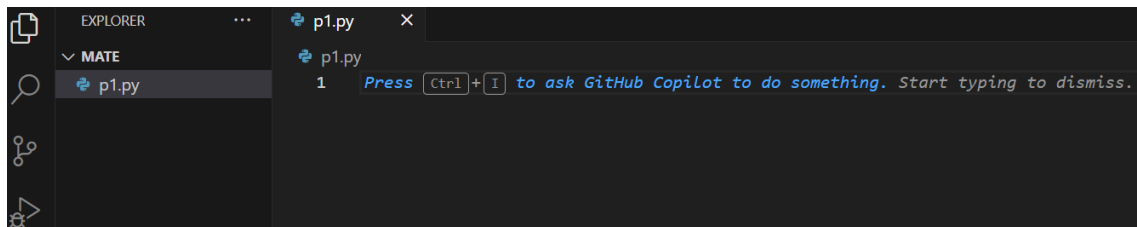
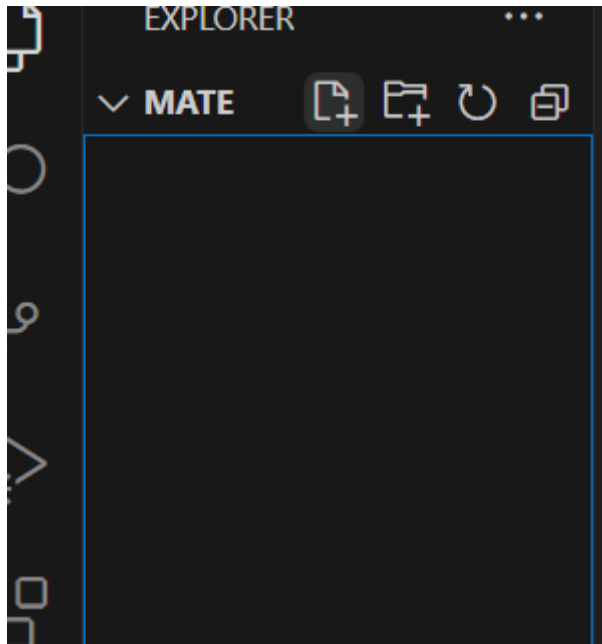


Se mejora el código

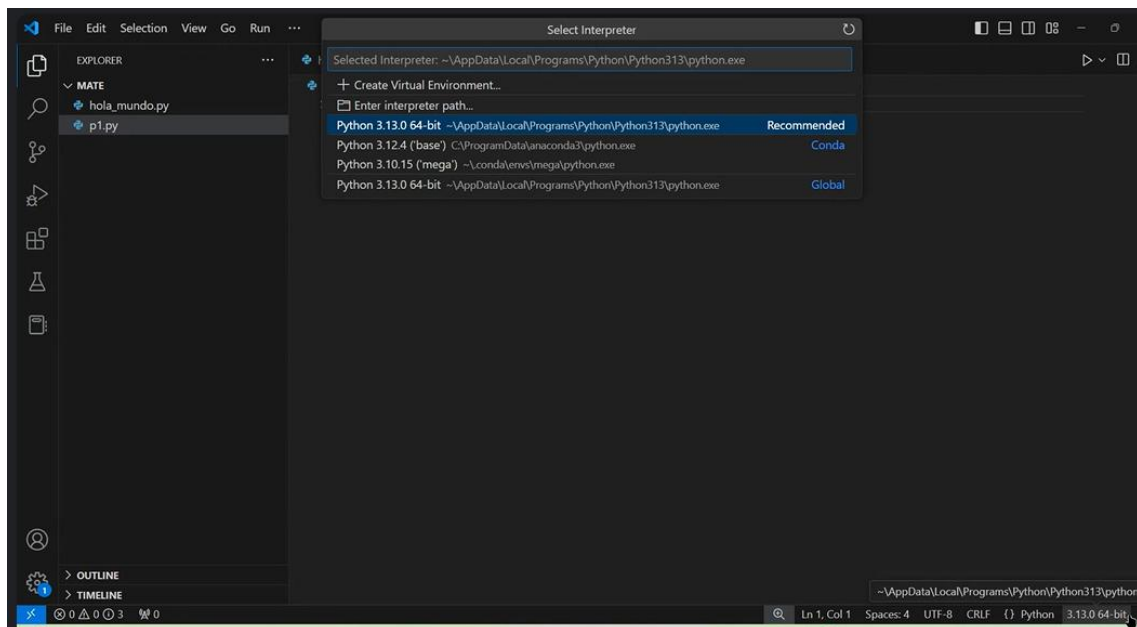


5. Creo un nuevo archivo .py y le pongo un nombre por ejemplo p1.py

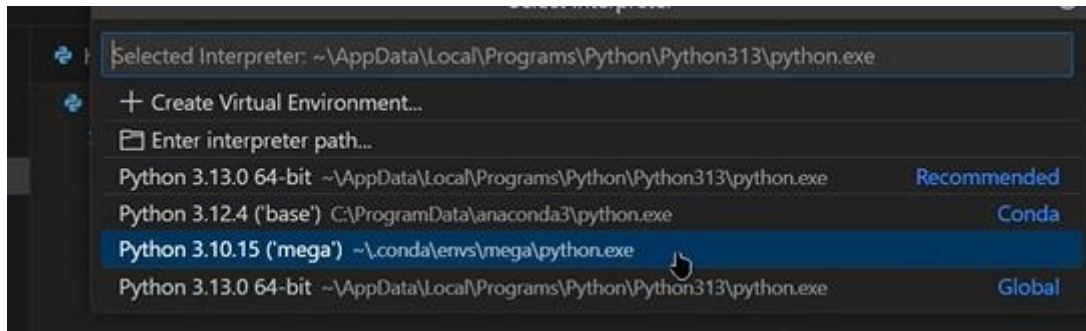




7. Escogemos el ambiente credo “mega” en la parte inferior derecha damos click



Y la seleccionamos:



## 6. Copiamos el código y lo ejecutamos, dando click en el botón de Play



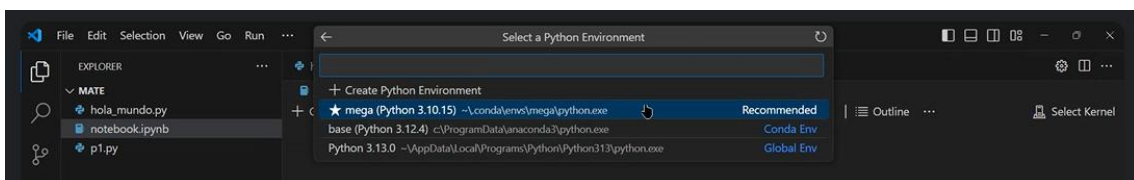
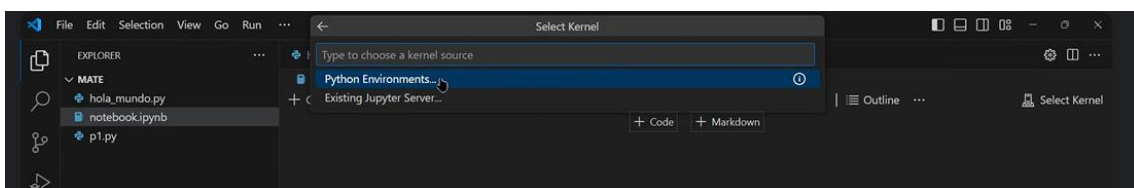
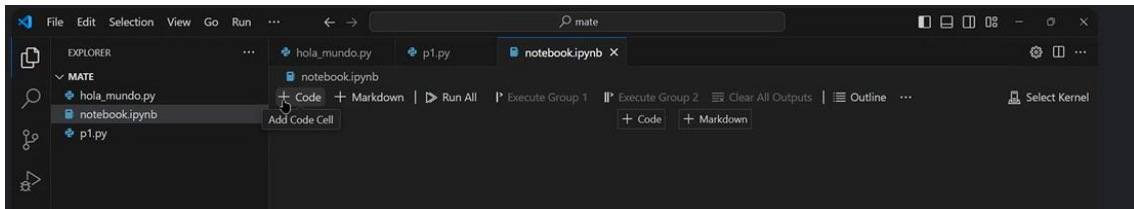
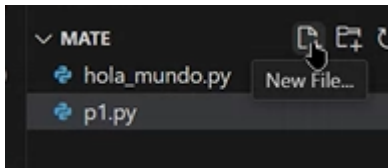
Hogar 0	Hogar 1	Firma 0	Firma 1	total
w[0,0]	w[1,1]	-w[0,0]	-w[1,1]	w[0,0]+w[1,1]-w[0,0]-w[1,1]
g[0,0]	g[1,1]	-g[0,0]	-g[1,1]	g[0,0]+g[1,1]-g[0,0]-g[1,1]
c[0,0]	c[1,0]	-c[0,0]	-c[1,0]	c[0,0]+c[1,0]-c[0,0]-c[1,0]
c[0,1]	c[1,1]	-c[0,1]	-c[1,1]	c[0,1]+c[1,1]-c[0,1]-c[1,1]
)+g[0,0]-c[0,0]-c[0,1]				0

```

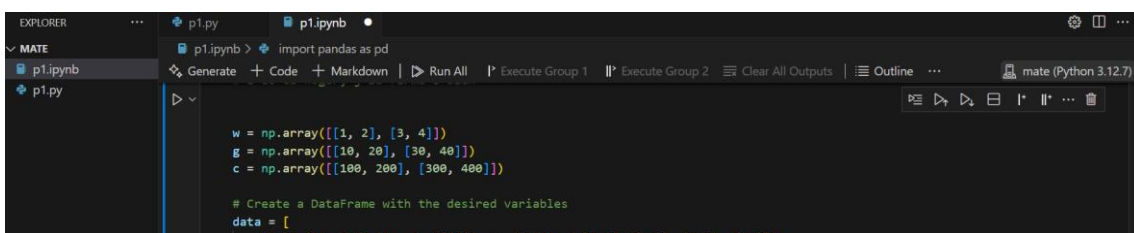
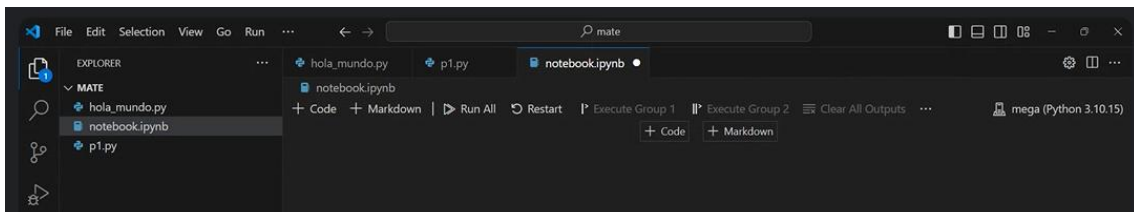
p1.py
p1.py > ...
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 # crear variables de salarios w(i,j), ganancias g(i,j), consumos c(i,j)
6 # i es el hogar, j la firma o bien
7
8
9 w = np.array([[1, 2], [3, 4]])
10 g = np.array([[10, 20], [30, 40]])
11 c = np.array([[100, 200], [300, 400]])
12
13 # Create a DataFrame with the desired variables
14 data = [
15     ['w[0,0]', 'w[1,1]', '-w[0,0]', '-w[1,1]', 'w[0,0]+w[1,1]-w[0,0]-w[1,1]'],
16     ['g[0,0]', 'g[1,1]', '-g[0,0]', '-g[1,1]', 'g[0,0]+g[1,1]-g[0,0]-g[1,1]'],
17     ['c[0,0]', 'c[1,0]', '-c[0,0]', '-c[1,0]', 'c[0,0]+c[1,0]-c[0,0]-c[1,0]'],
18     ['c[0,1]', 'c[1,1]', '-c[0,1]', '-c[1,1]', 'c[0,1]+c[1,1]-c[0,1]-c[1,1]'],
19 ]

```

- También se puede crear un .ipynb (Jupyter Notebook) y verificar que esté activado el entorno en la parte superior derecha



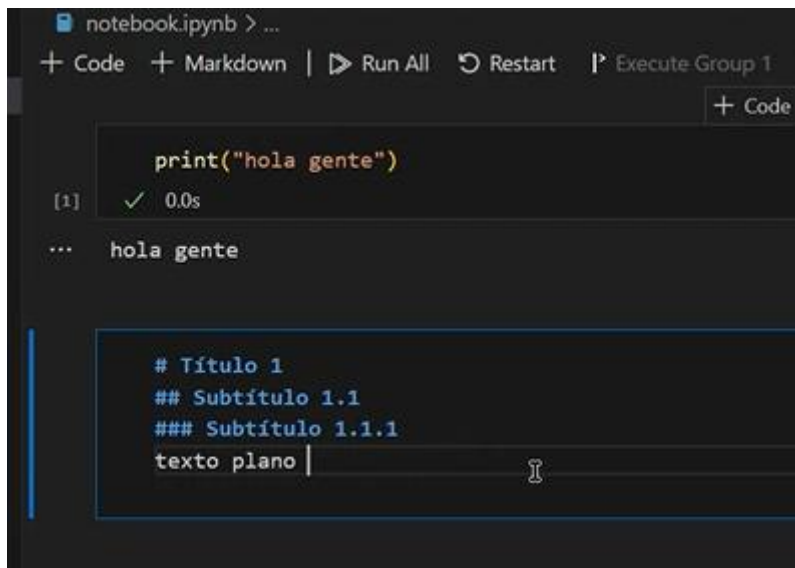
Aparece en la parte superior derecha el ambiente “mega”



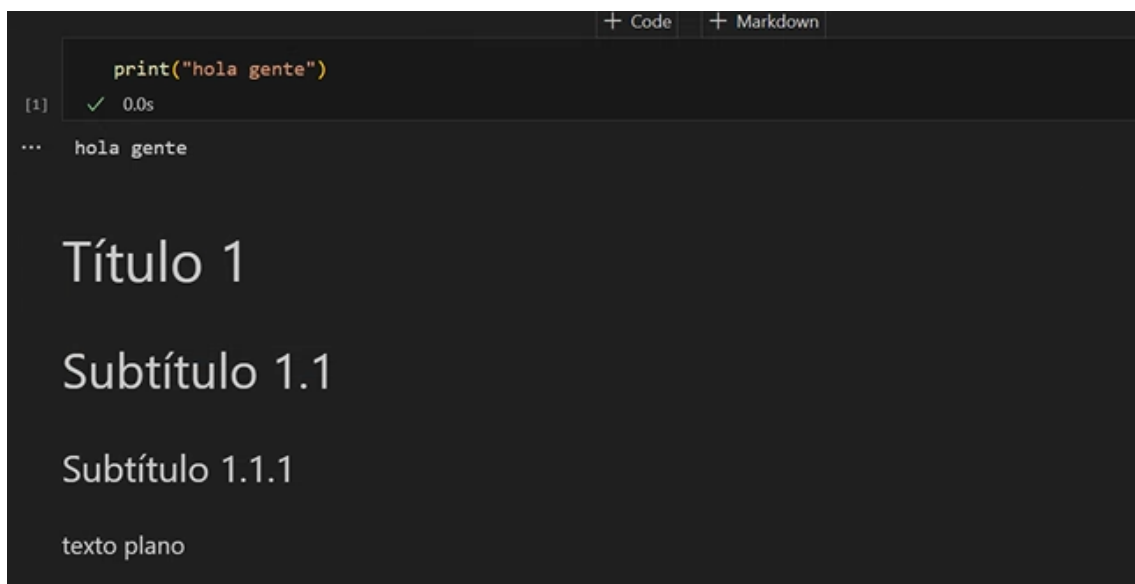
	Hogar 0	Hogar 1	Firma 0	Firma 1	total
salarios	w[0,0]	w[1,1]	-w[0,0]	-w[1,1]	w[0,0]+w[1,1]-w[0,0]-w[1,1]
ganancias	g[0,0]	g[1,1]	-g[0,0]	-g[1,1]	g[0,0]+g[1,1]-g[0,0]-g[1,1]
consumo b0	c[0,0]	c[1,0]	-c[0,0]	-c[1,0]	c[0,0]+c[1,0]-c[0,0]-c[1,0]
consumo b1	c[0,1]	c[1,1]	-c[0,1]	-c[1,1]	c[0,1]+c[1,1]-c[0,1]-c[1,1]
total	w[0,0]+g[0,0]-c[0,0]-c[0,1]	w[1,1]+g[1,1]-c[1,0]-c[1,1]	-w[0,0]+-g[0,0]-c[0,0]-c[0,1]	-w[1,1]+-g[1,1]-c[1,0]-c[1,1]	0



En Jupyter notebook puedes combinar Markdown y Código



The screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with buttons for '+ Code', '+ Markdown', 'Run All', 'Restart', and 'Execute Group 1'. Below the toolbar, there's a code cell with the text `print("hola gente")`. The output of this cell is shown as `[1] ✓ 0.0s` followed by `hola gente`. Below the code cell, there's a markdown cell containing the following text: `# Título 1`, `## Subtítulo 1.1`, `### Subtítulo 1.1.1`, and `texto plano`. The cursor is positioned at the end of the `texto plano` line.



The screenshot shows the rendered output of the Jupyter Notebook. The code cell's output is `hola gente`. Below it, the markdown cell's content is rendered as a large heading 

# Título 1

, followed by a subheading 

## Subtítulo 1.1

, then a sub-subheading 

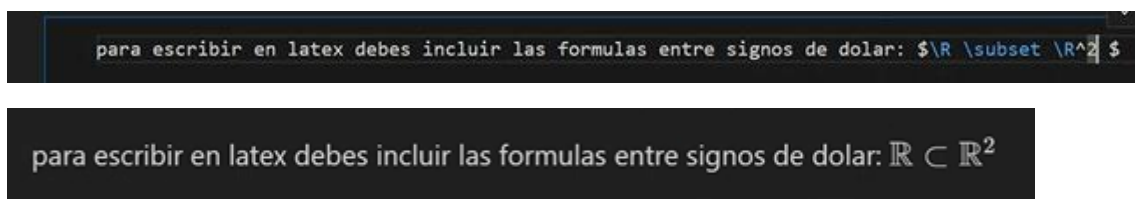
### Subtítulo 1.1.1

, and finally the text 

texto plano

.

También puedes escribir Latex debes usar los símbolos de \$\$



The screenshot shows a Jupyter Notebook interface. The code cell contains the text: `para escribir en latex debes incluir las formulas entre signos de dolar:  $\mathbb{R} \subset \mathbb{R}^2$` . The output of this cell is the rendered LaTeX expression:  $\mathbb{R} \subset \mathbb{R}^2$ .