

You will Learn how to install and prepare the python environment using Vscode or Jupyter Notebook.

### 1.2.1. Installing Python (latest stable release - 3.13/ or newer)

- Open your web browser and go to the official Python downloads page: <https://www.python.org/downloads/>
- Scroll down and locate Python 3.13.7 in the list of available downloads.
- Click on the version, which takes you to its download page.

Looking for a specific release?  
Python releases by version number:

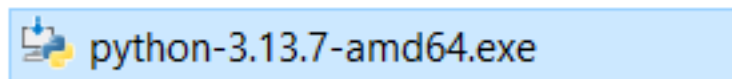
Release version	Release date	Click for more	
Python 3.13.7	Aug. 14, 2025	<a href="#">Download</a>	<a href="#">Release Notes</a>
Python 3.13.6	Aug. 6, 2025	<a href="#">Download</a>	<a href="#">Release Notes</a>
Python 3.13.5	June 11, 2025	<a href="#">Download</a>	<a href="#">Release Notes</a>
Python 3.11.13	June 3, 2025	<a href="#">Download</a>	<a href="#">Release Notes</a>

- Under "Files" on the download page, scroll down and choose the appropriate installer for your system:
  - For Windows 64-bit, click the Windows installer (64-bit) link (usually an .exe file), choose the 32-bit installer if needed.

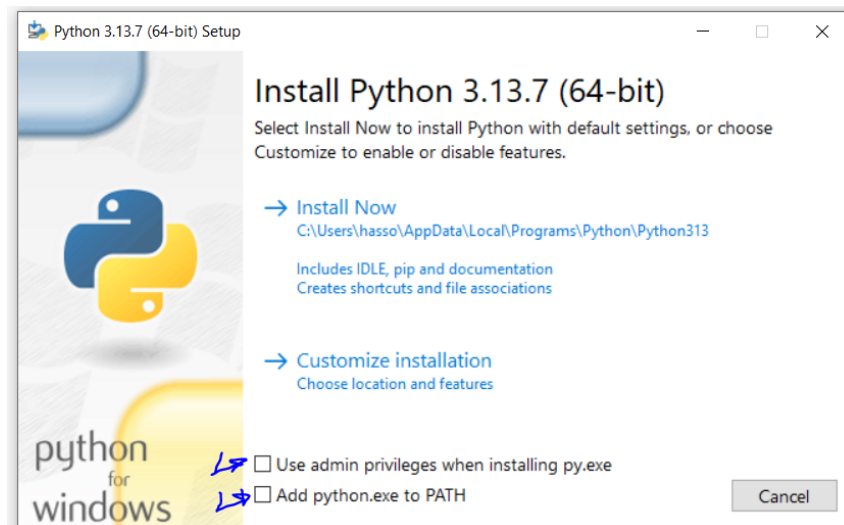
**Files**

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore	SBOM
<a href="#">Gzipped source tarball</a>	Source release		138c2e19c835ead10499571e0d4cf189	28.0 MB	SIG	.sigstore	SPDX
<a href="#">XZ compressed source tarball</a>	Source release		256cdb3bbf45cdce7499e52ba6c36ea3	21.7 MB	SIG	.sigstore	SPDX
<a href="#">macOS 64-bit universal2 installer</a>	macOS	for macOS 10.13 and later	ac0421b04eef155f4daab0b023cf3956	67.8 MB	SIG	.sigstore	
<a href="#">Windows installer (64-bit)</a>	Windows	Recommended	1da92e43c79f3d1539dd23a3c14bf3f0	27.5 MB	SIG	.sigstore	SPDX
<a href="#">Windows installer (32-bit)</a>	Windows		f501c1b321c82412ed330ec5604cac39	26.2 MB	SIG	.sigstore	SPDX

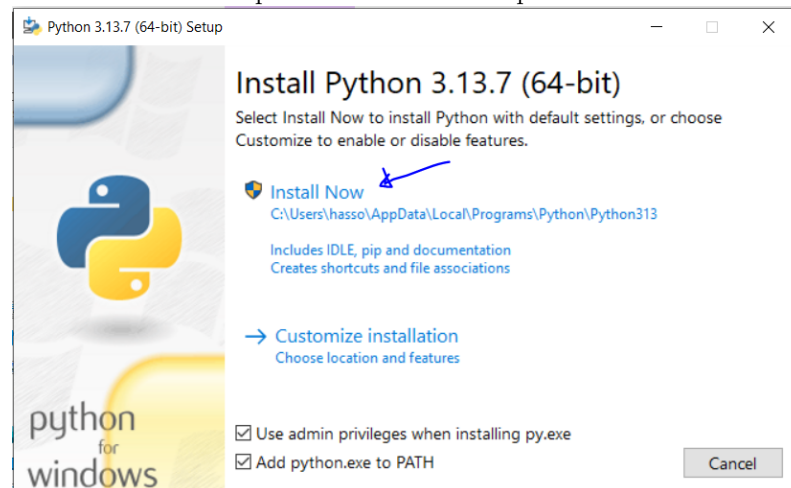
- Once downloaded, run the installer file.
  - It should be in the Download directory.



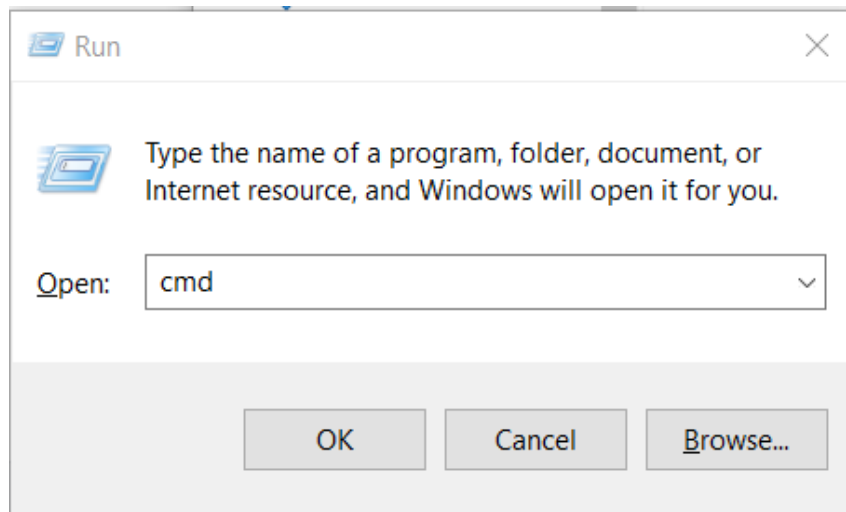
- In the installer window:
  - Important: Check the box Add Python to PATH at the bottom.



- Click Install Now to proceed with default options.



- Wait for the installation to complete, then click Close.
- Verify the installation:
  - Open Command Prompt (on keyboard press: Win + R, type cmd, and press Enter) or search tool bar for cmd or powershell.



- Type **python --version** and press Enter.

```
PS C:\Users\hasso> python --version
```

- You should see the installed Python version, such as

```
Python 3.13.7
```

For macOS and Linux, similar steps apply where you download the appropriate installer or use package managers like brew (macOS) or apt (Linux).

This simple step-by-step ensures Python 3.13.x is correctly installed with PATH setup to allow running Python from any command line window.

References:

- Official Python download page: <https://www.python.org/downloads/>
- Installation guide on RealPython: <https://realpython.com/installing-python/>
- Windows specific guide by DigitalOcean: <https://www.digitalocean.com/community/tutorials/install-python-windows-10>

---

### 1.2.2. Installing and setting up Jupyter Notebook

There is multiple ways to work with python as mentioned before. In this section you will be introduced on how to use Jupyter Notebook and make a python environment to work with.

- Installing Jupyter Notebook outside VS Code
  - Install Jupyter Notebook using pip `pip install notebook`.

*Note: If this did not work then use `python -m pip install notebook`.*

```
Collecting notebook
  Downloading notebook-7.4.5-py3-none-any.whl.metadata (10 kB)
Collecting jupyter-server<3,>=2.4.0 (from notebook)
```

You should see a similar output wait till the installation finishes. It will automatically install any missing files.

- Launching Jupyter Notebook from terminal `jupyter notebook`

```
PS C:\Users\hasso> jupyter notebook
```

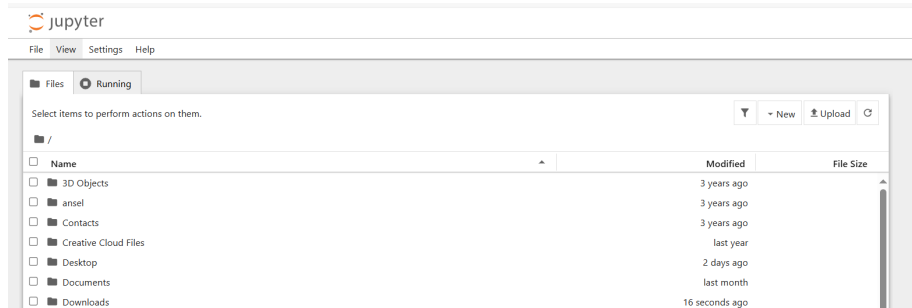
This starts a local server on your machine using the local ip and you will have to access it through a browser (for example chrome).

- Paste similar generated link (`http://localhost:8888/?token=...`) to a browser.

```
PS C:\Users\hasso> jupyter notebook
[2025-09-17 10:02:36.515 ServerApp] Extension package jupyter_lsp took 0.1744s to import
[2025-09-17 10:02:36.727 ServerApp] Extension package jupyter_server_terminals took 0.2216s to import
[2025-09-17 10:02:37.677 ServerApp] jupyter_lsp | extension was successfully linked.
[2025-09-17 10:02:37.885 ServerApp] jupyter_server_terminals | extension was successfully linked.
[2025-09-17 10:02:37.893 ServerApp] jupyterlab | extension was successfully linked.
[2025-09-17 10:02:37.181 ServerApp] notebook | extension was successfully linked.
[2025-09-17 10:02:38.042 ServerApp] notebook_shim | extension was successfully linked.
[2025-09-17 10:02:38.101 ServerApp] notebook_shim | extension was successfully loaded.
[2025-09-17 10:02:38.104 ServerApp] jupyter_lsp | extension was successfully loaded.
[2025-09-17 10:02:38.105 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[2025-09-17 10:02:38.109 LabApp] JupyterLab extension loaded from C:\Users\hasso\AppData\Local\Programs\Python\Python3
13\Lib\site-packages\jupyterlab
[2025-09-17 10:02:38.109 LabApp] JupyterLab application directory is C:\Users\hasso\AppData\Local\Programs\Python\Pyth
on313\share\jupyter\lab
[2025-09-17 10:02:38.118 LabApp] Extension Manager is 'pypi'.
[2025-09-17 10:02:38.653 ServerApp] jupyterlab | extension was successfully loaded.
[2025-09-17 10:02:38.659 ServerApp] notebook | extension was successfully loaded.
[2025-09-17 10:02:38.663 ServerApp] Serving notebooks from local directory: C:\Users\hasso
[2025-09-17 10:02:38.663 ServerApp] Jupyter Server 2.17.0 is running at:
[2025-09-17 10:02:38.664 ServerApp] http://localhost:8888/tree?token=788765c020c7c830afceda2c5ba77776cd8a13f2464f8a5e
[2025-09-17 10:02:38.664 ServerApp] http://127.0.0.1:8888/tree?token=788765c020c7c830afceda2c5ba77776cd8a13f2464f8
a5e
[2025-09-17 10:02:38.664 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirm
ation).
[2025-09-17 10:02:38.885 ServerApp]

To access the server, open this file in a browser:
file:///C:/Users/hasso/AppData/Roaming/jupyter/runtime/jpserver-17724-open.html
Or copy and paste one of these URLs:
http://localhost:8888/tree?token=788765c020c7c830afceda2c5ba77776cd8a13f2464f8a5e
http://127.0.0.1:8888/tree?token=788765c020c7c830afceda2c5ba77776cd8a13f2464f8a5e
Skipped non-installed server(s): basedpyright, bash-language-server, dockerfile-la
nguage-server-nodejs, javascript-typescript-languageserver, jedi-language-server, julia-language-server, pyrefly, pyright, p
ython-language-server, python-lsp-server, r-languageserver, sql-language-server, texlab, typescript-language-server, uni
fied-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, yam
l-language-server
```

- In the browser you will have the following



What you see is the main directory of your computer.

You should go to **Downloads** so that you can access the downloaded workshop from there it is called **Python\_for\_Data\_Science-master**.

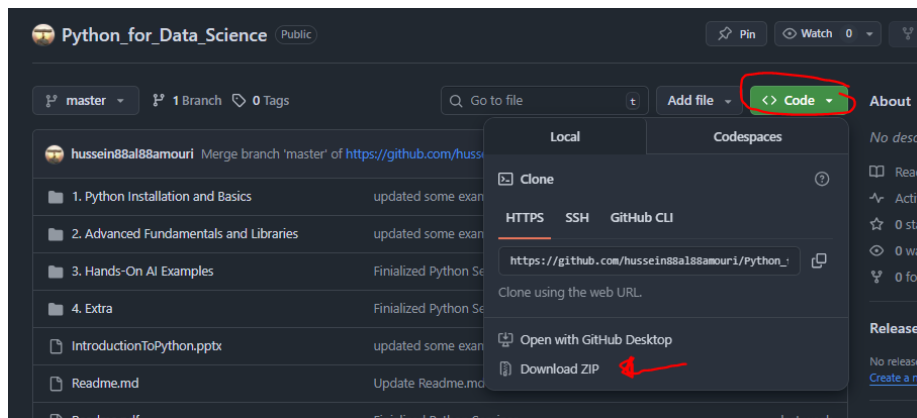
This setup tutorial covers local IDE (VS Code) with Python and Jupyter support, and cloud-based notebook usage with Google Colab for versatile Python programming and AI workflows.

References:

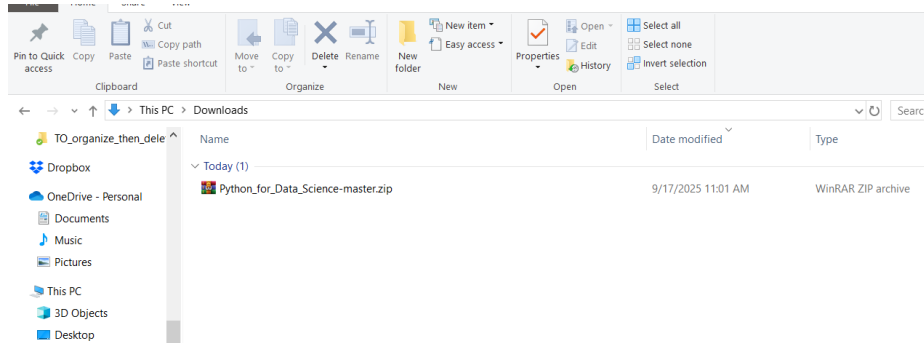
- Official VS Code Python docs: <https://code.visualstudio.com/docs/python/python-tutorial>
- Google Colab Welcome: <https://colab.research.google.com/notebooks/intro.ipynb>
- YouTube tutorial examples for VS Code Python setup

### To download the workshop

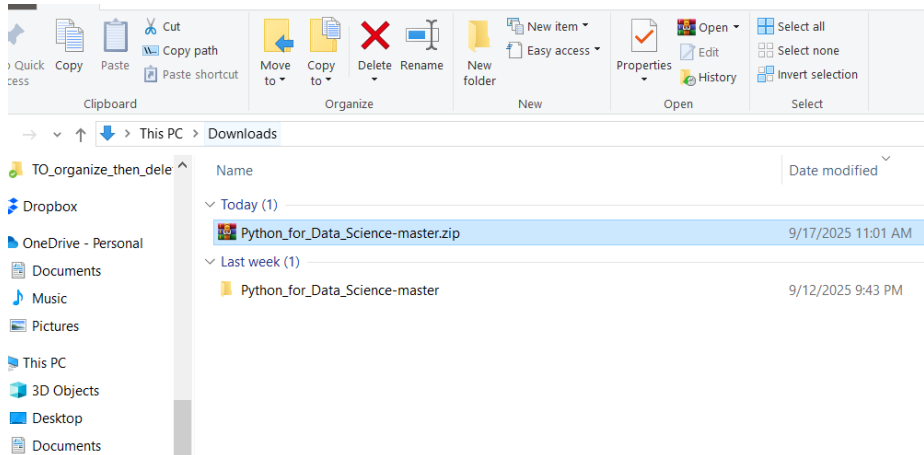
- Go to the following link: [https://github.com/hussein88al88amouri/Python\\_for\\_Data\\_Science](https://github.com/hussein88al88amouri/Python_for_Data_Science)
- You should press on the green button as in the following figure, and then download the zipped file.



You should have a downloaded zipped file.



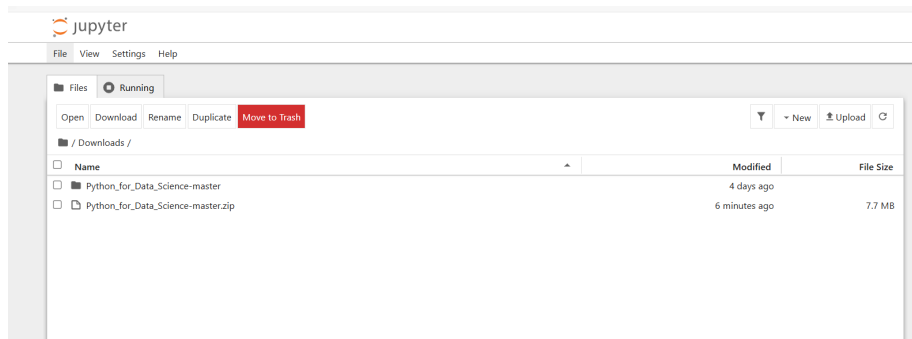
- Note you downloaded a zipped file, which means it is compressed and hence you should uncompress it. Right click on the zipped folder and choose **Extract here**. After extraction you should have the following:



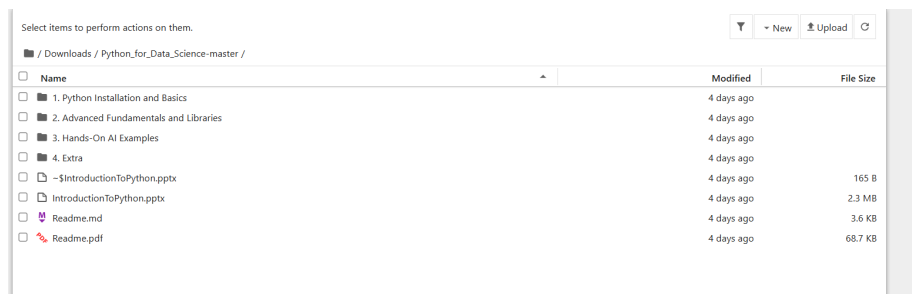
**Access the workshop from Jupyter Notebook** Now after we installed python, jupyter notebook, and the workshop you can directly access the material from the the notebook.

Go the browser where you had the notebook launched before. ***If it gives you an error launch it again following the steps in 1.2.2. Launching jupyter notebook.***

- Back to the notebook, you will see the extracted folder similar to the following



- Now you can access the workshop

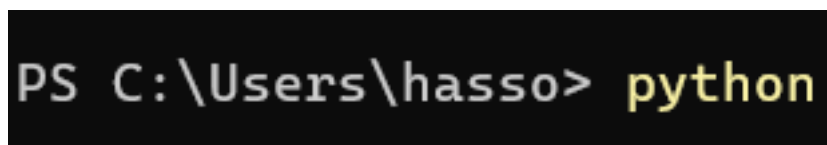


**During the workshop you will learn how to interact and work with jupyter notebook.** You are welcomed to try and play with the material yourself.

### 1.2.3. Introduction to interactive coding

Python provides an interactive mode, commonly referred to as a Python shell or REPL (Read-Eval-Print Loop), which allows you to execute Python code line by line and see the immediate results. This mode is useful for testing code snippets, experimenting with language features, and performing quick calculations.

**Launching the Python Shell** To launch the Python shell, open a command prompt or terminal and enter the python command.



This will start the Python interpreter and display a prompt indicating that you can enter Python code.

```
PS C:\Users\hasso> python
Python 3.13.7 (tags/v3.13.7:bceelc3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

Depending on your system, you may need to use `python3` command to launch the Python 3 interpreter.

**Note:** `>>>` indicates that you are working in interactive mode.

**Executing Python statements interactively** Once the Python shell is launched, you can start entering Python statements and expressions, and the interpreter will execute them immediately. For example, you can try entering simple mathematical expressions:

```
>>> 2 + 3 # Output: 5 (Anything after # is considered a comment no need to write it down).
>>> 10 / 2 # Output: 5.0
```

You can also define variables, create functions, and execute more complex code:

```
>>> x = 5
>>> y = 2 * x + 3
>>> y # Output: 13

>>> def square(n):
...     return n ** 2
...
>>> square(4)
```

**Exiting the Python Shell** To exit the Python shell, you can use the `exit()` or `quit()` function or press the appropriate key combination for your operating system (such as `Ctrl+Z` or `Ctrl+D`).

```
PS C:\Users\hasso> python
Python 3.13.7 (tags/v3.13.7:bceelc3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 2 + 3
5
>>> exit()
PS C:\Users\hasso> |
```

By running Python in interactive mode, you have a convenient way to test code, experiment with different ideas, and get immediate feedback. It's a valuable tool for learning and exploring Python's features and capabilities

---

#### 1.2.4. Running Python code: scripts vs notebooks

- Python scripts are saved as text files with the `.py` extension while notebooks in `.ipynb`.



- The scripts can be executed using various methods while notebooks are interactive and need to be used with either of the methods mentioned in previous section.
- Running Python scripts allows you to execute a series of Python statements or functions stored in a file.
- To create a Python script, you can use any text editor or integrated development environment (IDE) of your choice.
- Once you have created your Python script, you can execute it from the command line by invoking the Python interpreter followed by the name of the script file.

- Open a command prompt or terminal, navigate to the directory where your script is located, and use the following command:

```
PS C:\Users\UserName> python my_script.py
```

The Python interpreter will read the contents of the script file, execute the statements sequentially, and display the output or perform any desired actions.

## EXTRA:

**Python virtual environments** Python provide a concept which is virtual environments this allows you to create an isolated python installed with all requirements in a specific directory unaffected by any change that someone else can do the main python installation.

This is done by using **venv** or **anaconda**:

- Quick intro to virtualization using venv
  - You can use venv following the command below

```
PS C:\Users\hasso> python -m venv
usage: venv [-h] [--system-site-packages] [--symlinks | --copies] [--clear] [--upgrade] [--without-pip]
            [--prompt PROMPT] [--upgrade-deps] [--without-scm-ignore-files]
            ENV_DIR [ENV_DIR ...]
```

- To create an environment you should specify the directory name which will be the name of your environment as well

```
PS C:\Users\hasso> python -m venv test_env
```

This will create the test\_venv in the current directory, in this case it is C:\Users\hasso> (since it is on my machine, it should be different for you).

- To access the new environment simply do the following

- \* If you are using powershell

```
PS C:\Users\hasso> .\test_env\Scripts\Activate.ps1
(test_env) PS C:\Users\hasso> |
```

You can see now that you are working inside the environment.

\* If you are using cmd simply use `.\test_env\Scripts\Activate.bat`

– To deactivate the environment write in the terminal `deactivate`

- Using Anaconda:

– You have to install anaconda on your machine. <https://www.anaconda.com/download#windows>

– You can follow this link for the installation. <https://www.geeksforgeeks.org/installation-guide/how-to-install-anaconda-on-windows/>

– Check that conda is installed by writing the following in the cmd

```
conda -V
```

– Update the conda environment

```
conda update conda
```

– Set up the virtual environment

```
conda create -n test_env python
```

Note that you can specify the python environment but you have what version does anaconda support by checking their website.

– Activate the environment using

```
conda activate test_env
```

– To deactivate

```
conda deactivate
```

## Installing and using VS Code as Python IDE

- Installing and Setting up VS Code for Python

– Download and install Visual Studio Code (VS Code) from <https://code.visualstudio.com/>

– Open VS Code. Go to the Extensions view by clicking the square icon on the sidebar or press `Ctrl+Shift+X`.

– Search for the Python extension by Microsoft and click Install. This provides Python language support, debugging, IntelliSense, linting, and more.

– In VS Code, open a folder as your workspace via `File > Open Folder`.

– Create a new Python file with `.py` extension (e.g., `hello.py`).

– Select your Python interpreter by opening the Command Palette (`Ctrl+Shift+P`), typing `Python: Select Interpreter`, and picking the installed Python version.

– You can now write and run Python code in VS Code by clicking the green Run Python File button or running code snippets in the terminal.

- Setting up Jupyter Notebook in VS Code
  - Within VS Code, install the Jupyter extension (search and install in Extensions view).
  - Open or create .ipynb files (Jupyter notebooks) directly inside VS Code.
  - Run cells interactively, visualize outputs, and leverage notebook features without leaving the editor.

### Using google colab

- Using Google Colab for Python/Jupyter Notebooks (cloud alternative)
  - Visit <https://colab.research.google.com/>
  - Sign in with a Google account.
  - Create a new notebook, which runs Python code on Google's cloud servers without local installation.
  - Upload or create notebooks, run code cells interactively, and share notebooks online easily.

**Jupyter LAB** In addition to jupyter notebook, there is an alternative which is much powerfull that has file explorer to access your workspace and it also allows to run different interactive shells, notebooks, command prompts.

You can Install Jupyter Lab

```
```bash
pip install jupyterlab
```
```

To run use:

```
```bash
jupyter lab
```
```

If this does not work try using:

```
```bash
pip3 install jupyterlab
```
```

```
```bash
python3 -m jupyterlab
```
```