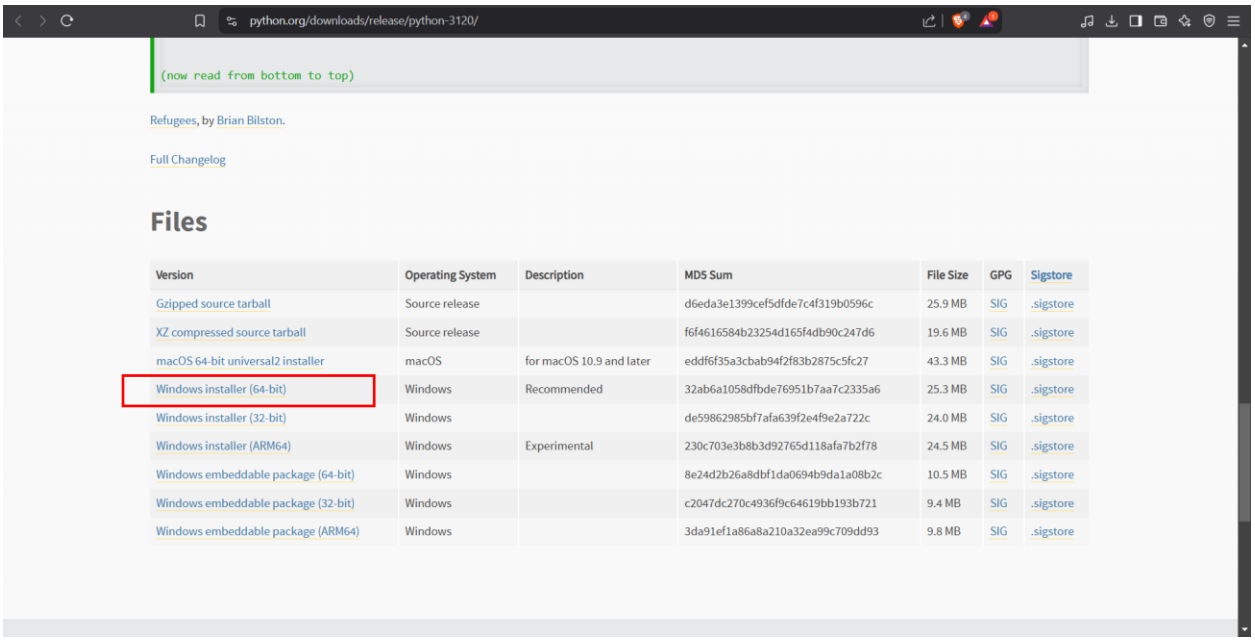
SPARK: Python Workshop Setup Guide

Step 1: Download Python 3.12

1. Go to the official Python website: https://www.python.org/downloads/release/python-3120/
2. Scroll down and under "Files," download **Windows installer (64-bit)**.



Step 2: Install Python

1. Run the downloaded installer (`python-3.12.X-amd64.exe`).
2. **Check the box** for **"Add Python to PATH"** (important).
3. Click **"Customize installation"** (optional but recommended).
   - Ensure "pip" and "IDLE" are selected.
   - Keep the default installation directory or change it as needed.
4. Click **"Install"** and wait for the process to complete.
5. Once installed, click **"Close"**.

Step 3: Verify Installation

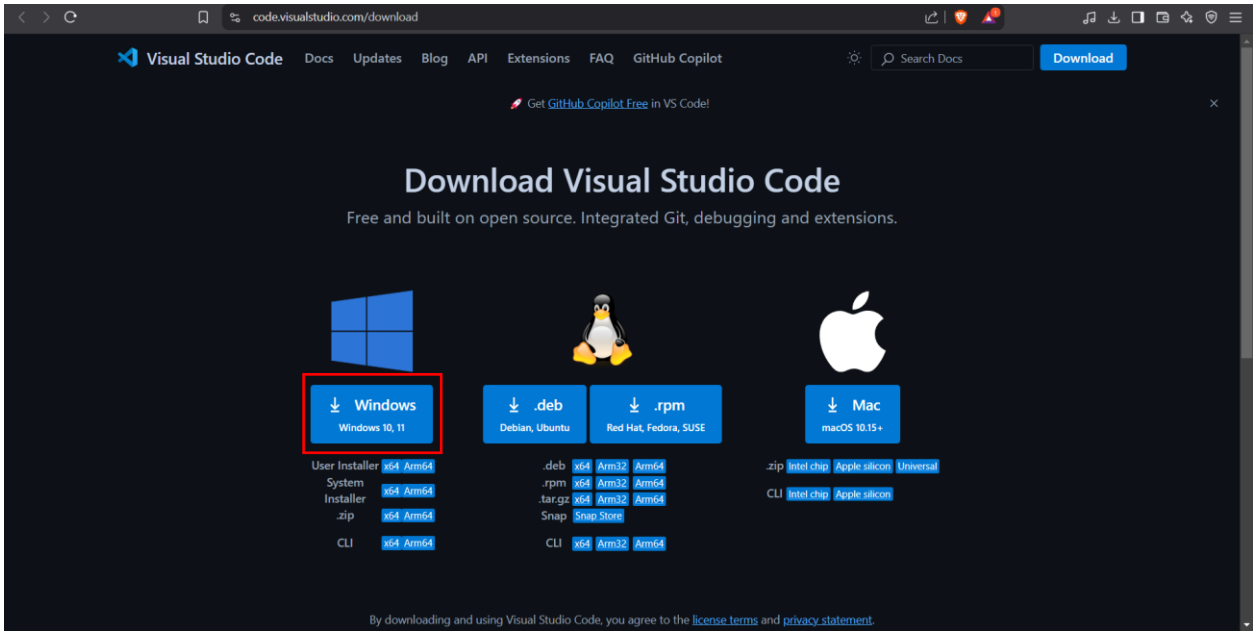
1. Open **Command Prompt** (`Win + R` → type `cmd` → press **Enter**).
2. Type the following command and press **Enter**:

```
python --version
```

   - If installed correctly, it should display: `Python 3.12.X`.

Step 4: Download and Install VS Code

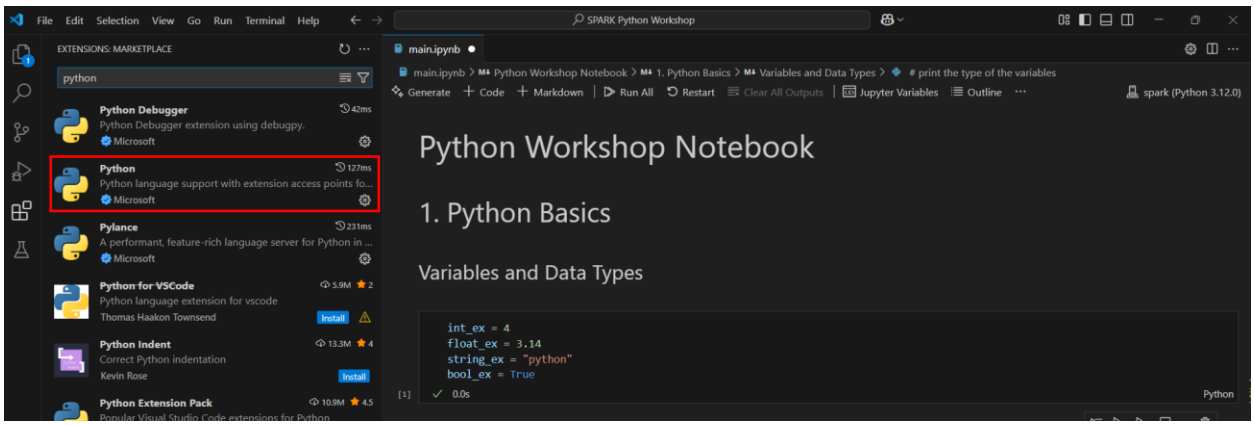
1. Go to the official Visual Studio Code website: https://code.visualstudio.com/download
2. Download the Windows installer for your system (64-bit or 32-bit).



3. Run the downloaded installer (VSCodeSetup-x64-xxx.exe).
4. Follow the installation prompts:
   - Accept the license agreement.
   - Choose the installation folder or use the default.
   - Select additional tasks (e.g., create a desktop icon, add to PATH, etc.).
5. Click "Install" and wait for the process to complete.
6. Once installation is complete, click "Finish" to open VS Code.

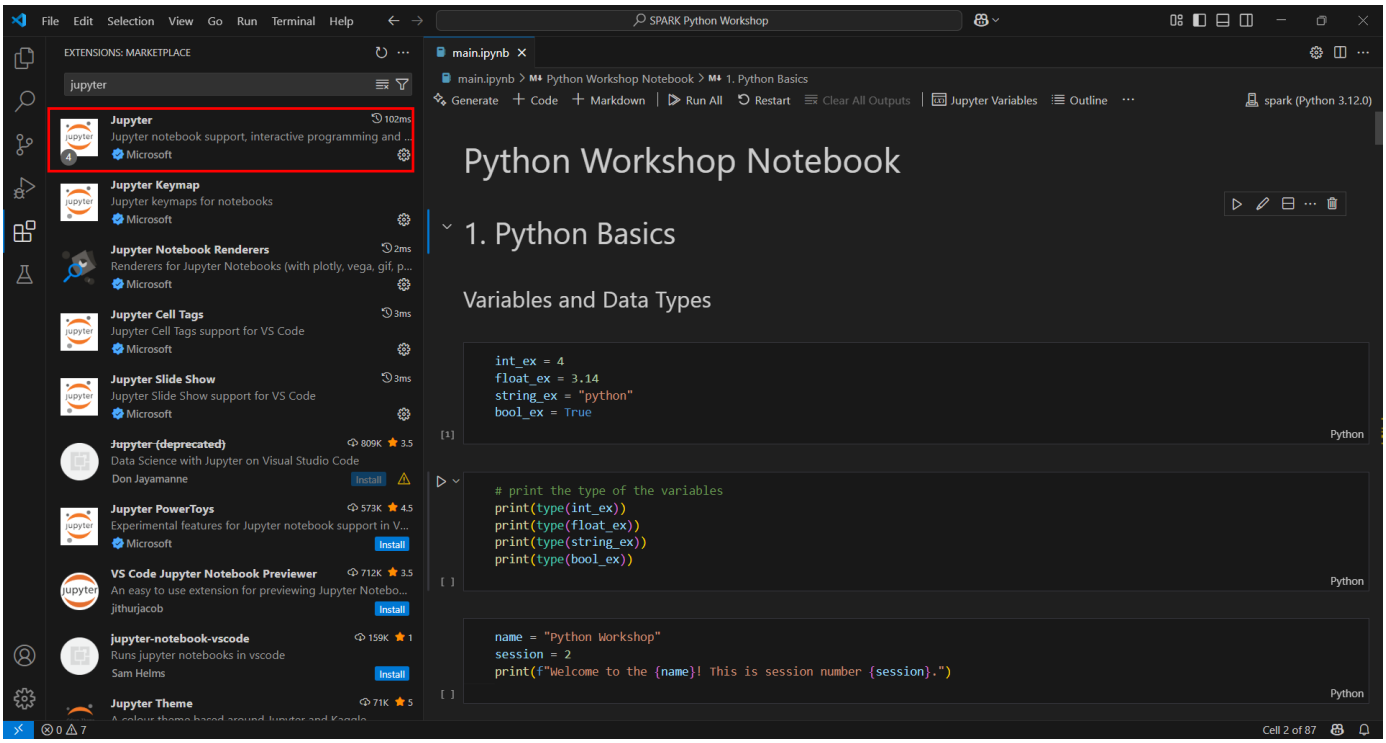
Step 5: Install Python Extension for VS Code

1. Open VS Code.
2. Click on the Extensions icon on the left sidebar (or press Ctrl+Shift+X).
3. In the search bar, type Python.
4. Install the Python extension by Microsoft (it should be the first result).
   - Click the Install button.



Step 6: Install Jupyter Extension for VS Code

1. Open VS Code.
2. Click on the Extensions icon on the left sidebar (or press Ctrl+Shift+X).
3. In the search bar, type Jupyter.
4. Install the Jupyter extension by Microsoft (it should be the first result).
    - Click the Install button.

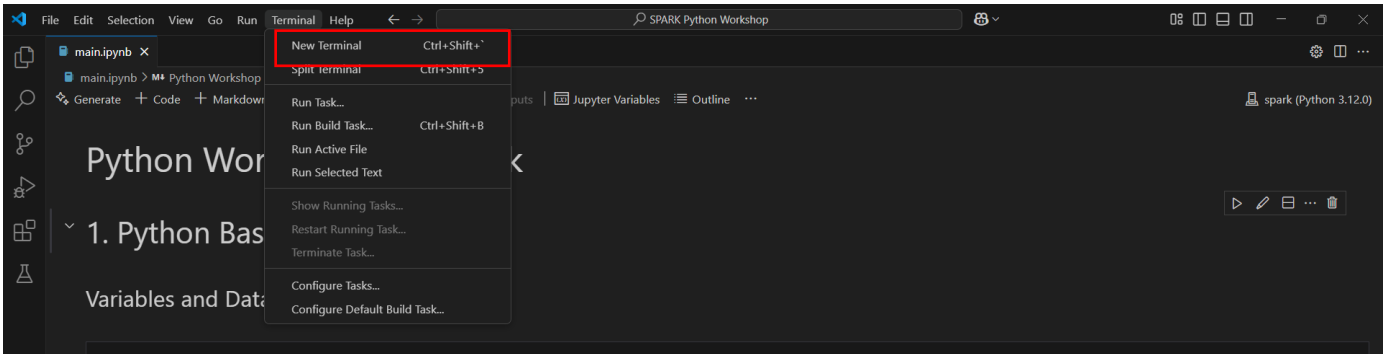


Step 7: Open the Project Folder in VS Code

1. Download this repository from GitHub **or** extract the ZIP file you have:
    - Click here to access the Git repository.
2. Once downloaded, extract the ZIP file (if applicable) to your local machine.
3. Open the extracted repo folder in VS Code:
    - You can click on File → Open Folder and select the folder you want to work with.

Step 8: Create and Activate the Virtual Environment (Optional but recommended)

1. Open the Terminal in VS Code:
    - Click on Terminal → New Terminal from the menu, or use the shortcut Ctrl + ` (backtick).



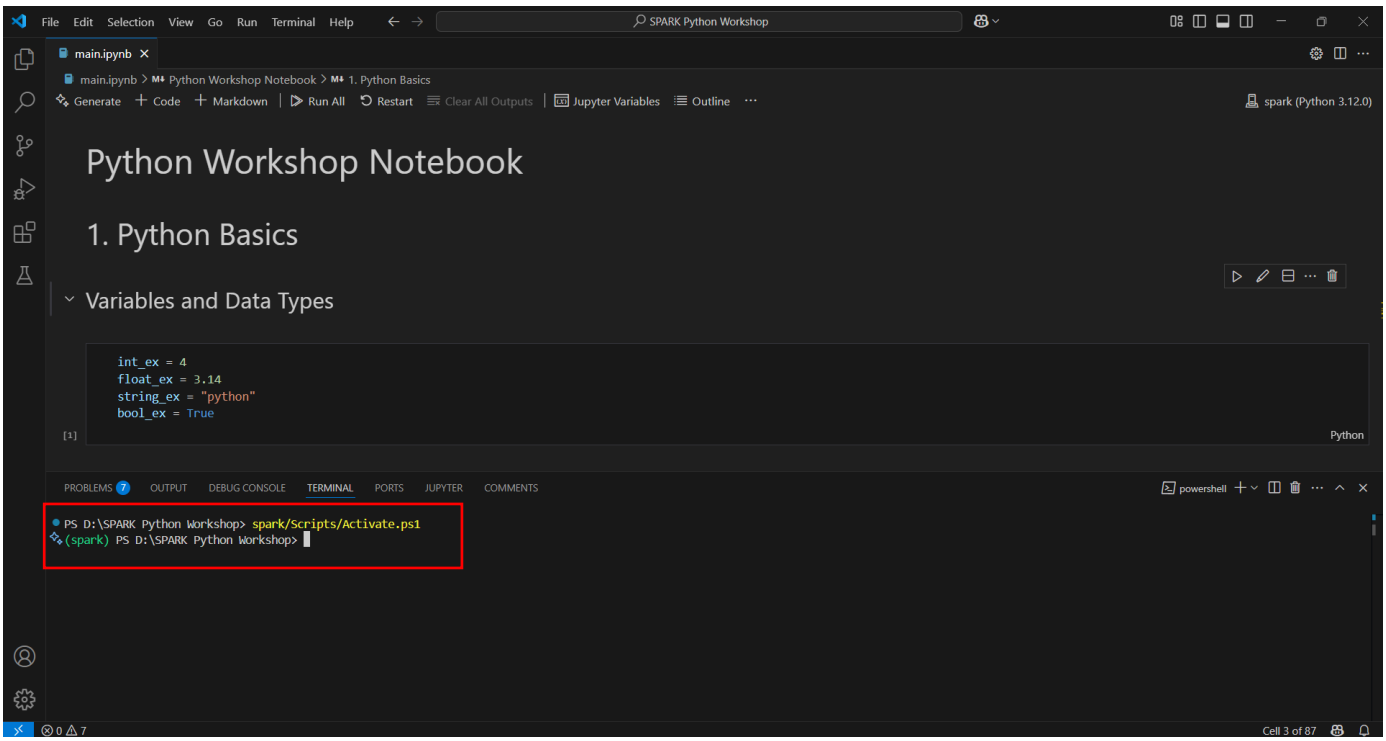
2. Create a virtual environment: (Here, the name of the virtual env is ‘spark’. You can change it preferably.)

```
python -m venv spark
```

3. Activate the Virtual Environment:
    - In the terminal, activate your virtual environment (spark) by running:

      `spark\Scripts\Activate.ps1` (If PS)

    - After activation, you should see (spark) appear at the beginning of the command line, indicating the virtual environment is active.



Step 9: Install Jupyter Kernel

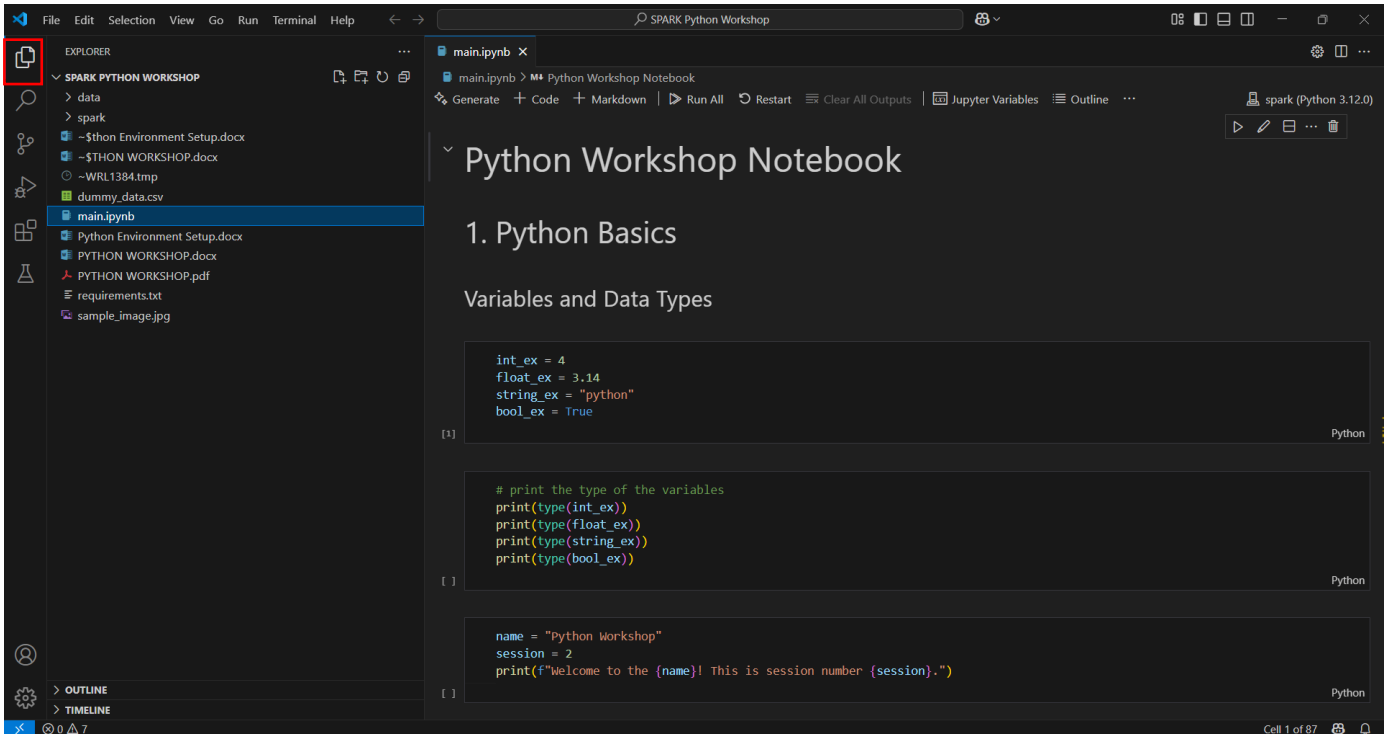
1. While the virtual environment is active, install the Jupyter kernel for that environment:

   `pip install ipykernel`

   `pip install -r requirements.txt` (Assuming you are in the given project folder and if you want to install the all dependencies for the session. I will be mentioned this again anyway.)

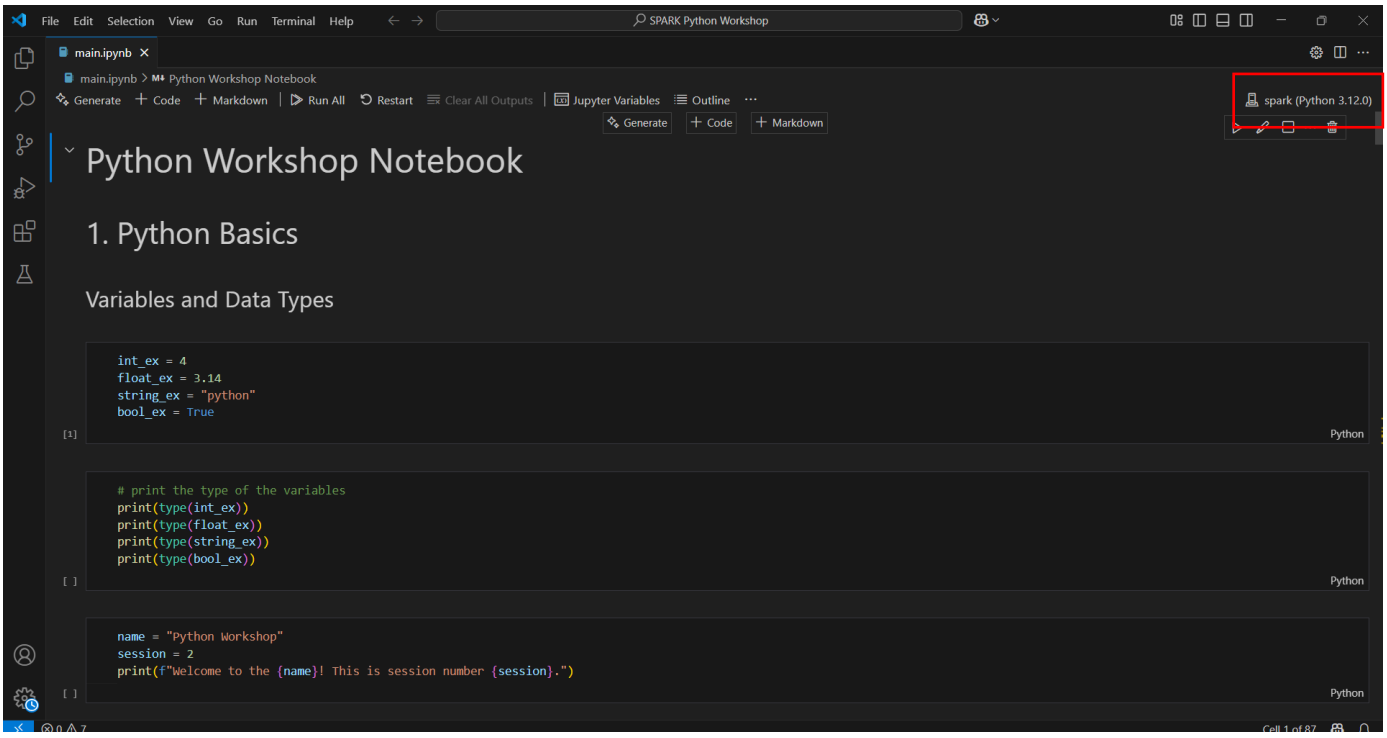
Step 10: Open the Jupyter Notebook

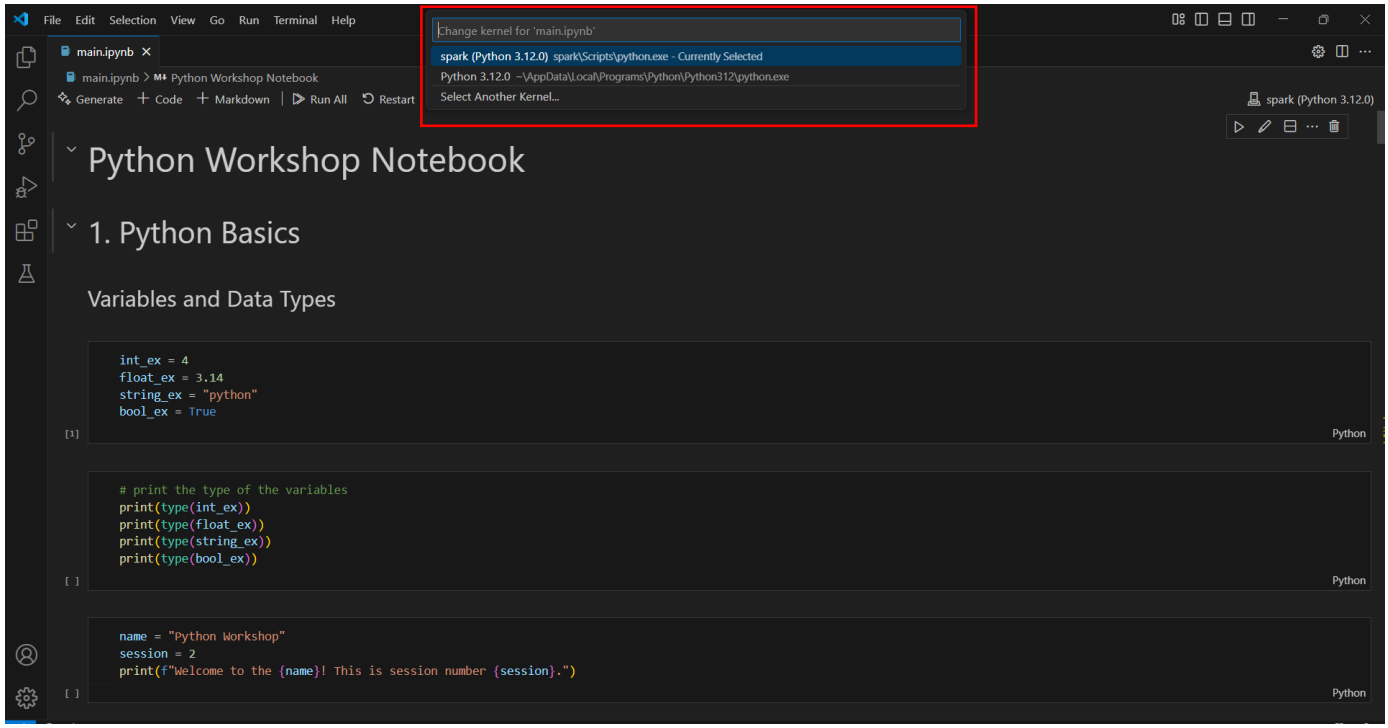
1. With your project folder open in VS Code, click on the .ipynb file for the Jupyter notebook to open it.



Step 11: Select the Kernel

1. In the open notebook, click on the kernel name in the top-right corner of the notebook.





2. A list of available Python kernels will appear. Select the kernel corresponding to your virtual environment (e.g., spark or the specific environment you created).
    - If your environment isn't listed, ensure it's activated and the ipykernel package is installed.

Step 12: Start Working in the Jupyter Notebook

1. With the kernel set to your virtual environment, you can now start writing and running Python code in your notebook, using the environment's libraries and dependencies.