

VS Code Setup Guide for Python Development

What is VS Code?

- Visual Studio Code (VS Code) is a **lightweight but powerful source code editor** developed by Microsoft.
- VS Code supports **debugging, embedded Git control, syntax highlighting, intelligent code completion**.

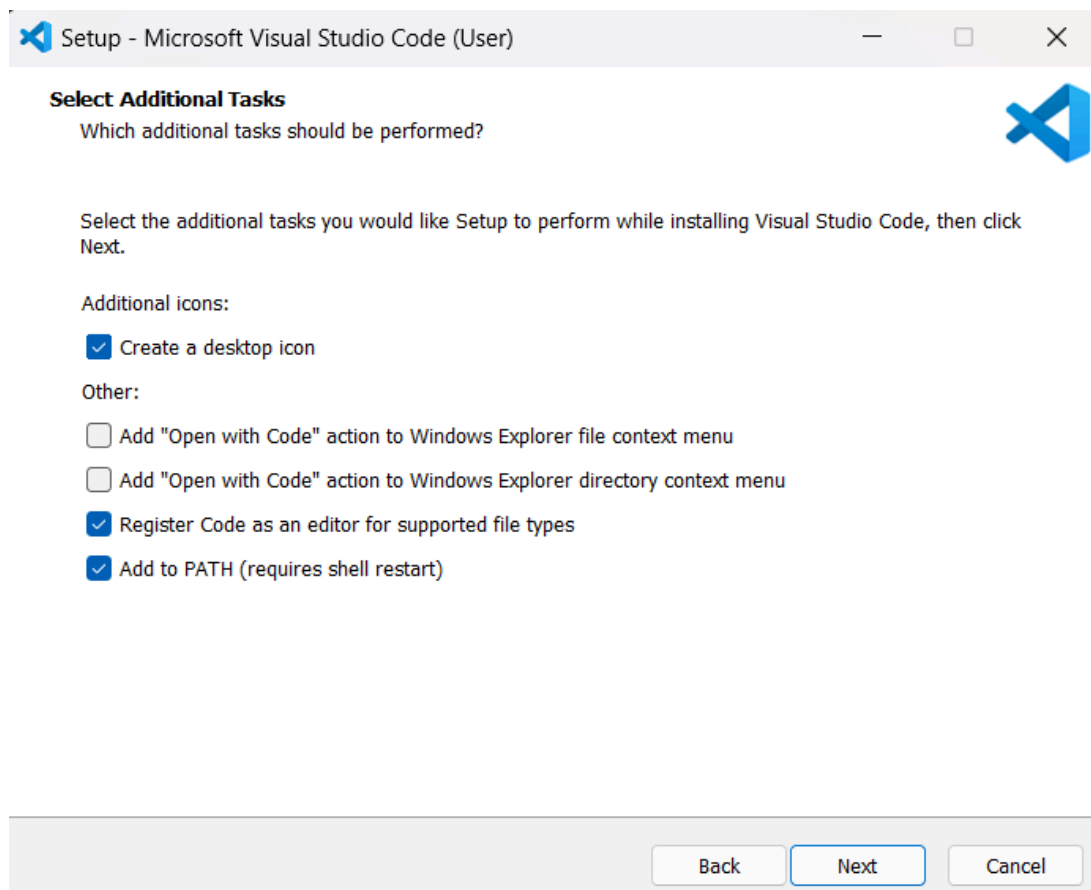
Why Use VS Code for Python?

- **Built-in Terminal:** You can directly run Python scripts, manage virtual environments, and execute Git commands from within VS Code's integrated terminal.
- **Debugging Capabilities:** VS Code provides powerful debugging tools for Python, helping you find and fix errors quickly.
- **Git Integration:** VS Code has built-in Git support, simplifying version control and collaboration.
- **Jupyter Notebook Support:** You can run and edit Jupyter notebooks directly in VS Code, which is great for data science and interactive programming.
- **Extensive Extension Support:** VS Code has a rich ecosystem of extensions, especially for Python, making development more efficient and enjoyable.

How to Install VS Code on Windows

Follow these steps to install VS Code on your Windows system:

1. **Download VS Code:** Go to the official VS Code website <https://code.visualstudio.com/> and download the Windows installer.
2. **Run the Installer:** Execute the downloaded `.exe` file (it should be in the Download folder).
3. **Accept the License Agreement:** Read and accept the license agreement.
4. **Choose Installation Location:** Select the installation location or use the default.
5. **Select Start Menu Folder (optional):** Choose a Start Menu folder or leave the default.
6. **Select Additional Tasks (optional):** Check the boxes for creating a desktop icon and adding VS Code to your PATH environment variable. This is very important.



7. **Select "Create a desktop icon"** optional
8. **Install:** Click "Install" to begin the installation process.

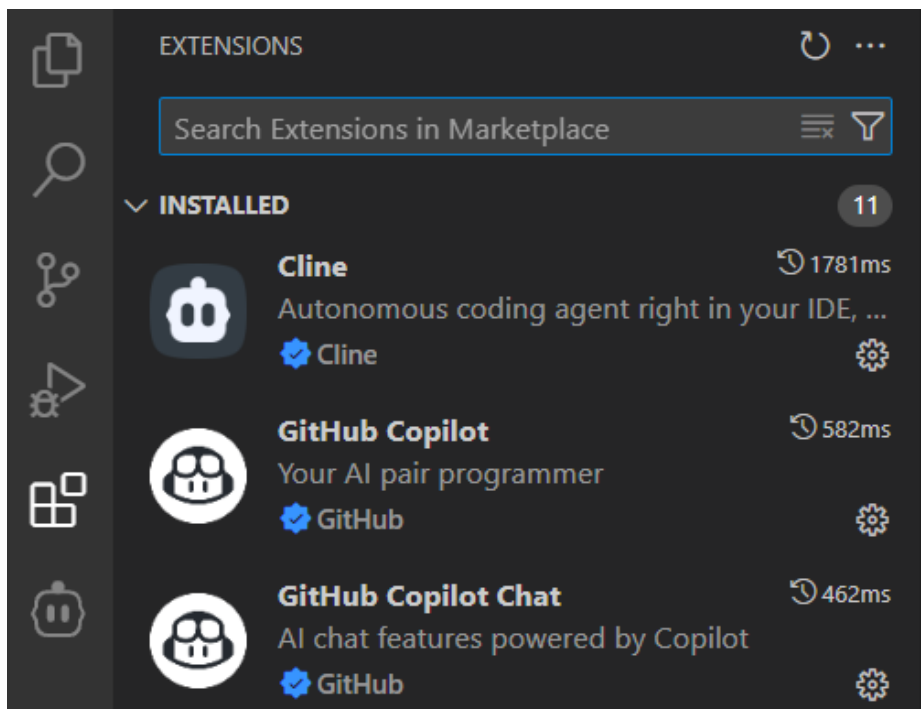
9. **Launch VS Code:** Once installed, launch VS Code.

Essential Configurations

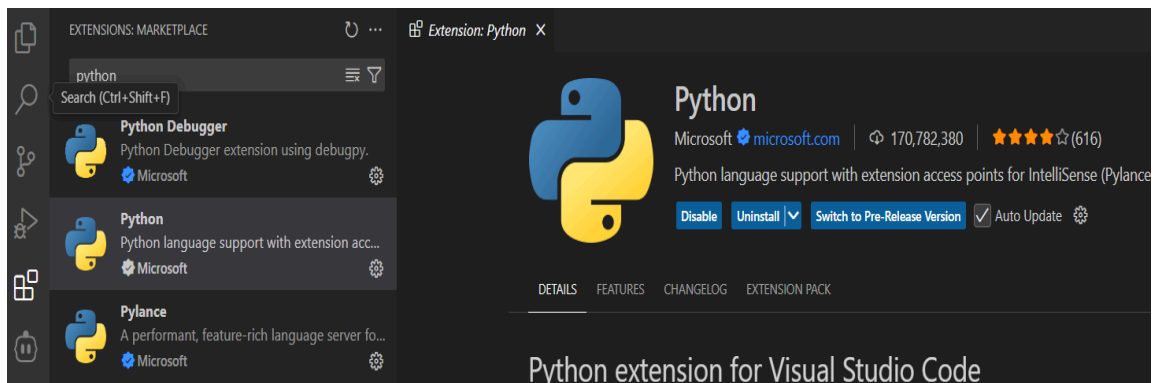
Installing the Python Extension

To enhance Python development in VS Code, install the official Python extension:

1. Open VS Code.
2. Click on the Extensions icon in the Activity Bar on the side (or press `Ctrl+Shift+X`).

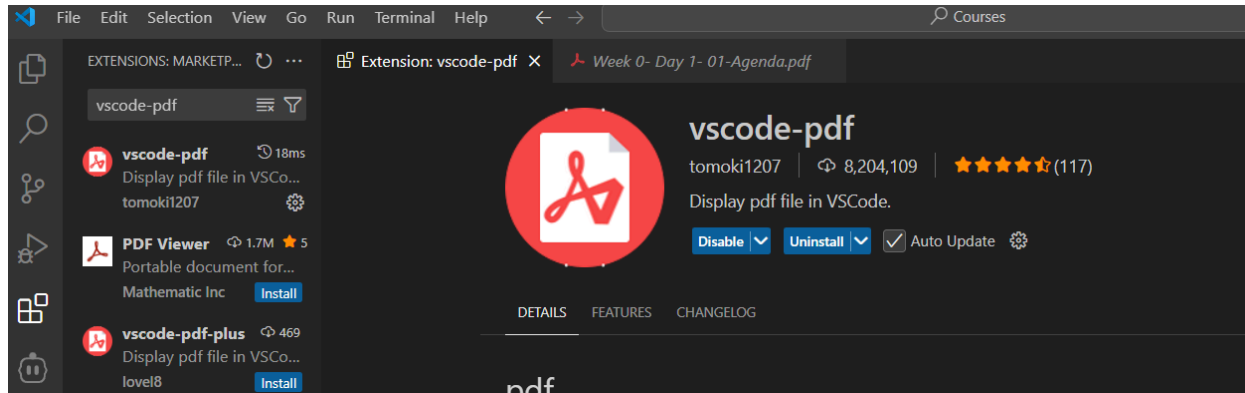


3. Search for "Python" by Microsoft.
4. Mostly, it should be pre-installed, if not Click "Install" for the extension.



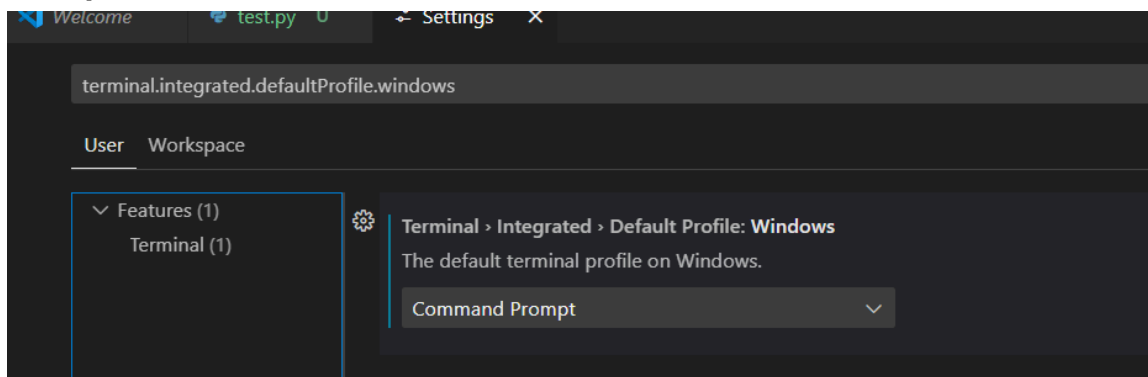
Installing VScode-pdf:

VScode can open other file types like images with png and jpeg formats, and csv file. To open PDF file we need "vscode-pdf" extension

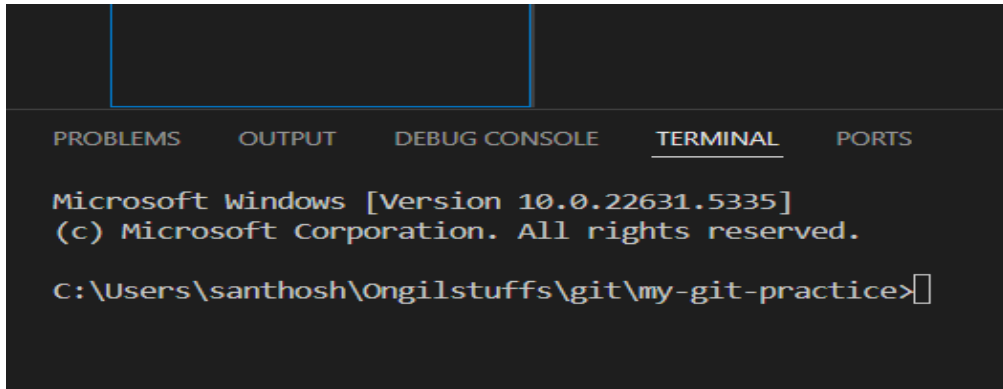


Using the Terminal (cmd)

1. Go to **File > Preferences > Settings**.
2. In the Settings search bar at the top, type the following:
"terminal.integrated.defaultProfile.windows"
3. You should see a setting named **Terminal > Integrated: Default Profile: Windows**. Click on the dropdown menu associated, select **"Command Prompt"**.



4. Go to **Terminal > New Terminal** in the top menu.
5. You should now see a **cmd.exe** terminal open within your VS Code window.



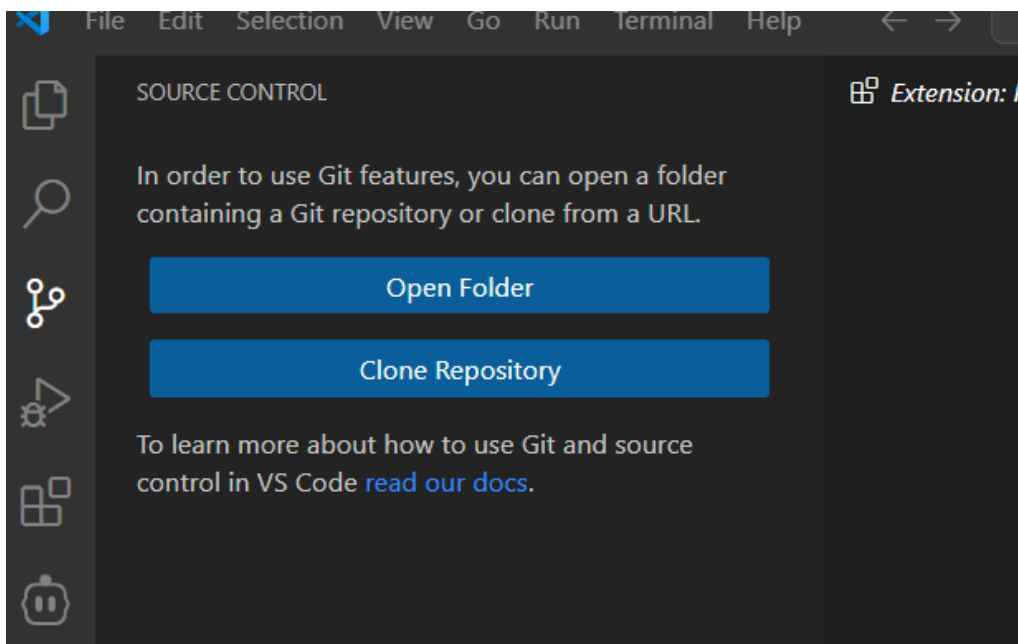
```
Microsoft Windows [Version 10.0.22631.5335]
(c) Microsoft Corporation. All rights reserved.

C:\Users\santhosh\Ongilstuff\git\my-git-practice>
```

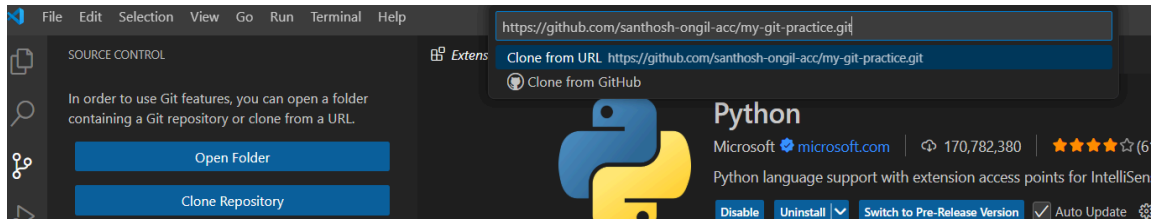
Git Configuration

VS Code has built-in Git integration:

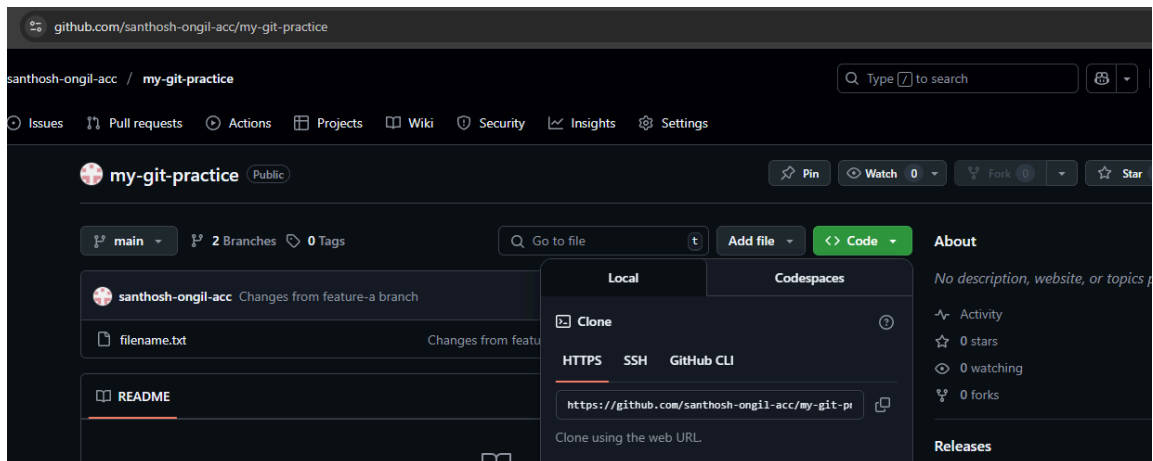
1. Make sure Git is installed on your system. If not, download and install it from git-scm.com. (Installation steps in Week0-Day1-Git installation.pdf)
2. Open VS Code (start with a new window) and go to the Source Control view (Ctrl+Shift+G).



3. You can initialize a Git repository or clone an existing one from this view.
4. Click "Clone Repository", and it will ask you to enter the url of your repository. Provide your repository URL, and select "Clone from URL"



Note: You can find your repository in github website and copy as below



5. A File explorer will open and select a folder to clone the repository.
6. VS Code provides an interface for committing, pushing, pulling, and managing branches.

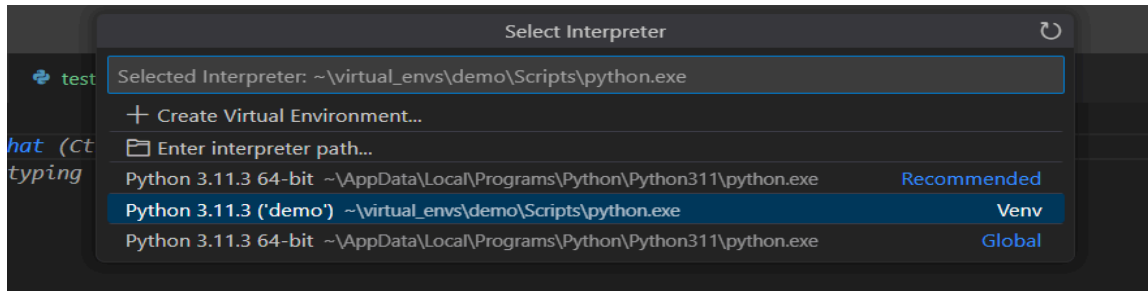
Python Interpreter Configuration and Virtual Environment Mapping

It's a good practice to use virtual environments for Python projects.

1. Open a Python file in VS Code or create a new one **"python_test.py"**.
2. VS Code will usually detect your Python interpreter automatically. If not, a message will appear to select an interpreter. (Hint: Python version will be displayed like 3.11.3)



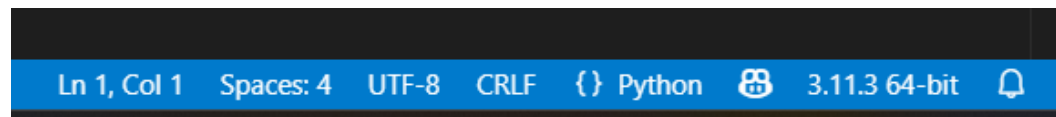
3. Click on the interpreter displayed in the bottom left of the status bar.
4. Select your desired Python interpreter from the list or select the existing virtual environment or create a new virtual environment.



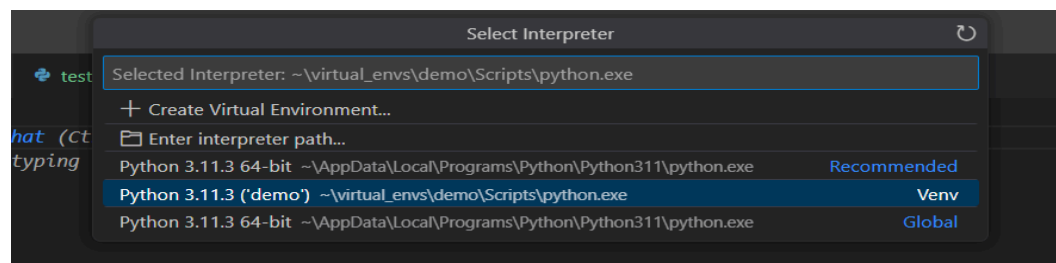
5. Please refer "Week 0- Day 1- 05-virtual environment setup.pdf" file to create a virtual environment

Note: If you not able to find the virtual environment already created, please follow below steps

1. Select python interpreter displayed in the bottom left of the status bar. (Hint: Python version will be displayed like 3.11.3)



2. Select "Enter interpreter path"

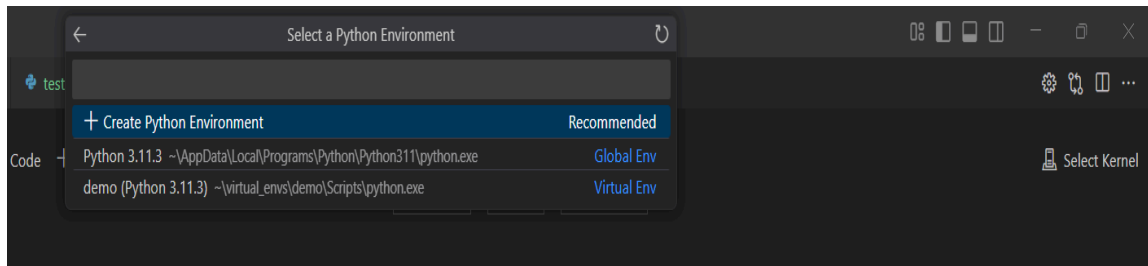


3. Browse to the file explorer and go to the virtual environment you created and select "python.exe" file under "Scripts" folder
Example: C:\Users\santhosh\virtual_envs\demo\Scripts\python.exe

Jupyter Notebook in VS Code Extension

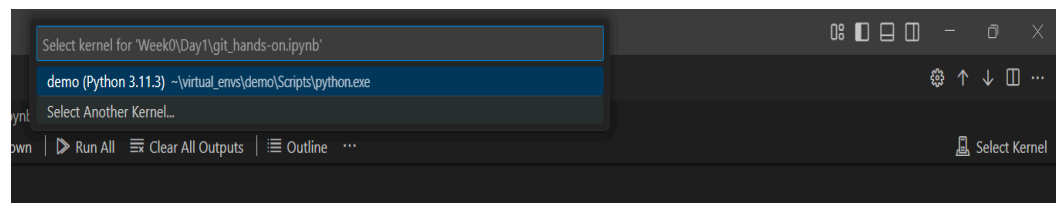
The Python extension allows you to work with Jupyter Notebooks (.ipynb files) directly within VS Code:

1. Open an existing **.ipynb** file or create a new one **"jupyter_test.ipynb"**.
2. VS Code will automatically recognize it as a Jupyter Notebook.
3. You can run cells, edit code, and view output directly in VS Code.
4. Select the virtual environment created as kernel in top right corner

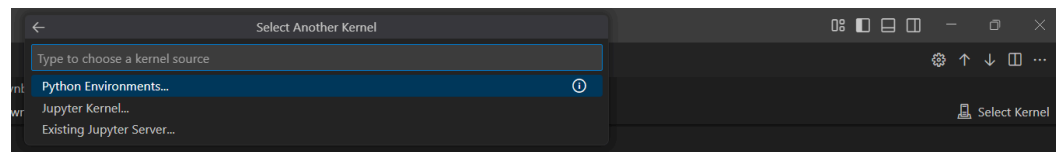


Note: If you are not able to find the virtual environment already created, please follow below steps.

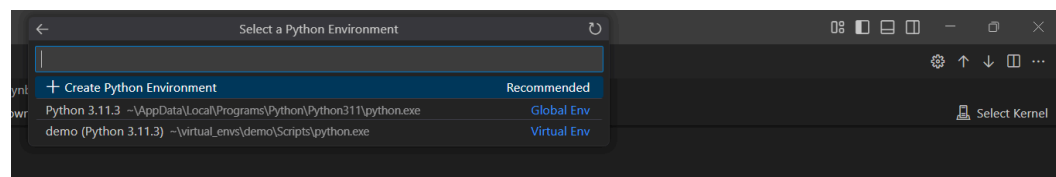
1. Click "select Kernel" in the top right corner of the vs code.
2. Click "Select Another Kernel".



3. Click "Python Environments"



4. List of available virtual environments will be shown, select your preferred virtual environment.

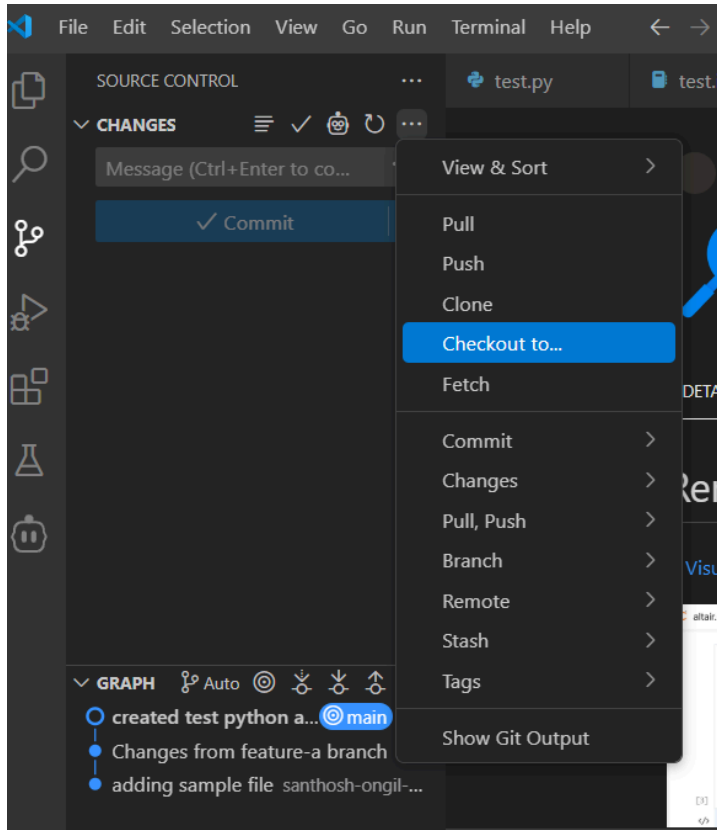


5. If you are not able to find your preferred virtual environment, please close and open vs code and try again.

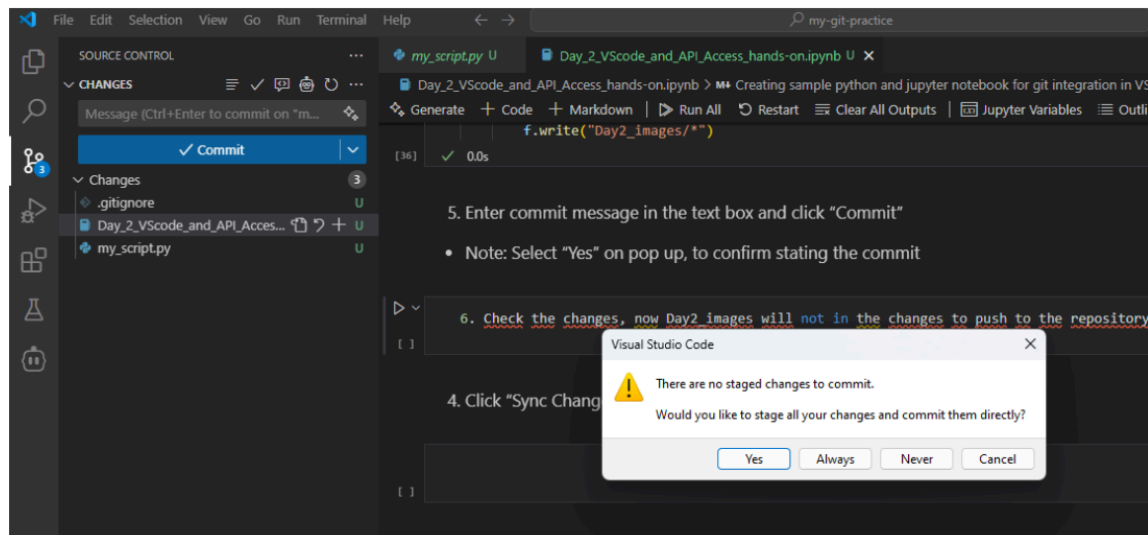
Git Commit, Push, Pull execution:

Let's use **python_test.py** and **jupyter_test.ipynb** files and push to the repository

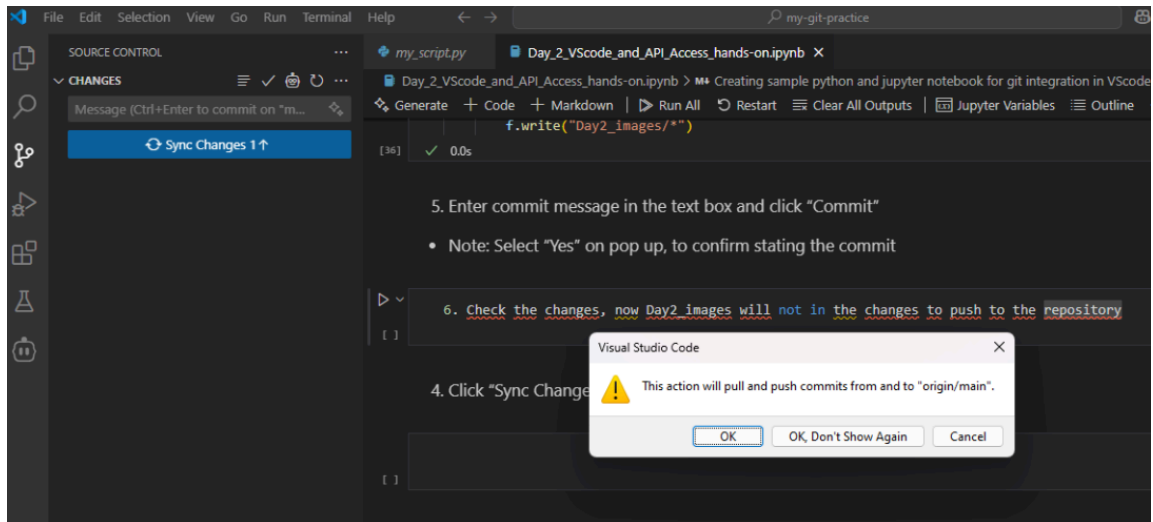
1. We can perform git commands like pull, push, commit, clone etc. select "more Action" or "..." Under the "Source control" tab.



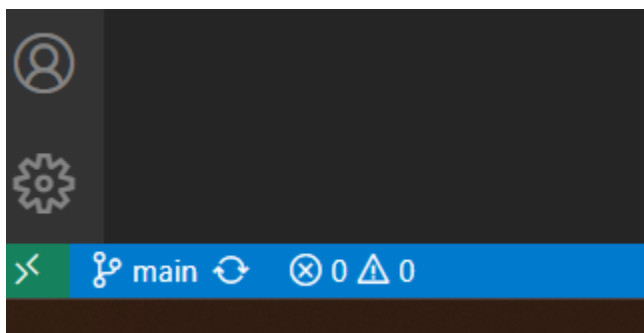
2. To push change, Goto "Source control" tab. We can see changes we made.
 3. Enter the commit message in the text box and click "Commit".
- Note: Select "Yes" on pop up, to confirm stating the commit



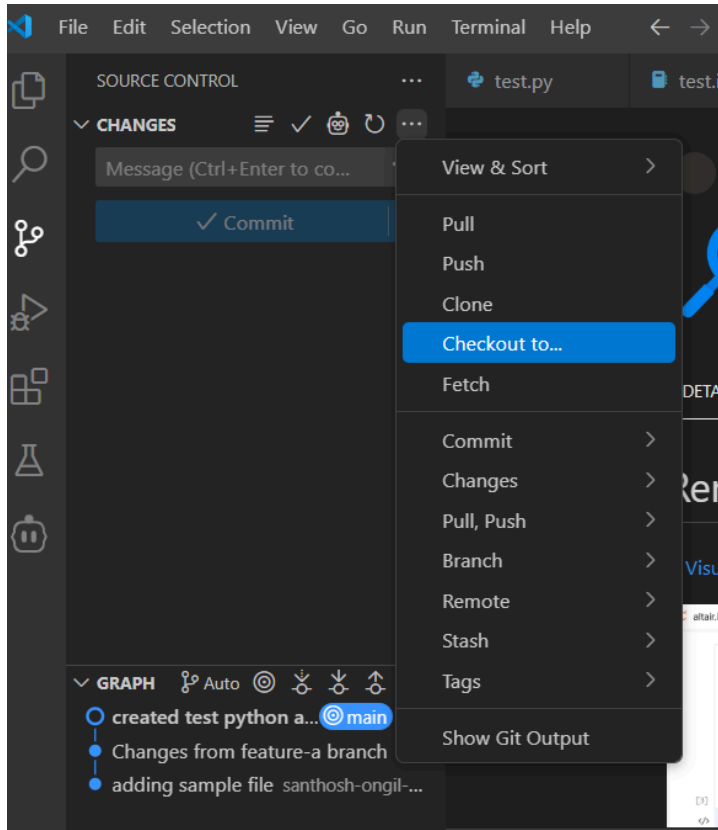
4. Click "Sync Changes" to push you local changes to the repository



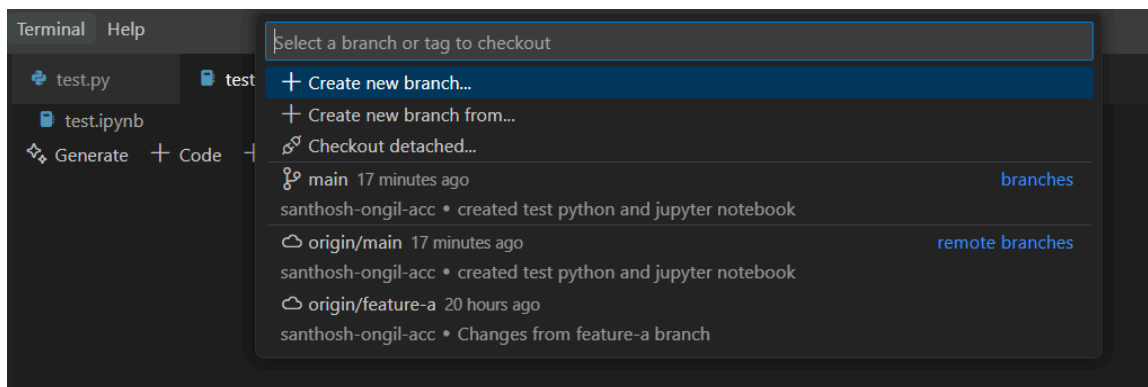
5. To check out to other branch or create new branch or list all branches from repo
6. Select branch in left bottom, below to Setting and Accounts Icon



7. Or select "more Action" or "..." Under the "Source control" tab.



8. Options will show to switch to another branch, create a new branch, and list available branches



Cline Extension for vscode

- Autonomous coding agent right in your IDE, capable of creating/editing files, running commands, using the browser, and more with your permission
- Cline supports API providers like OpenRouter, Anthropic, OpenAI, Google Gemini, AWS Bedrock, Azure, GCP Vertex, and Cerebras.

- Cline is a valuable extension for AI development in VS Code. Here's how to install it:

Installing the Cline Extension

Cline is a valuable extension for AI development in VS Code. Here's how to install it:

1. Open VS Code.
2. Click on the Extensions icon in the Activity Bar (or press Ctrl+Shift+X).
3. Search for "Cline."
4. Click "Install" for the Cline extension.
5. Follow any additional setup instructions provided by the extension after installation. This may involve configuring API keys or authentication if required.

OpenRouter

What is OpenRouter?

OpenRouter is an open-source platform that acts as a unified API for various large language models (LLMs). It allows developers to access multiple LLMs through a single API endpoint.

Why use OpenRouter?

- **Access to Multiple Models:** OpenRouter provides access to a wide range of language models from different providers, giving flexibility and choice.
- **Load Balancing and Fallback:** It can intelligently route requests to different models based on availability, performance, or cost, ensuring reliability.
- **Unified API:** Developers only need to learn one API to interact with multiple language models, simplifying integration.
- **Open Source:** Being open source, it offers transparency and community-driven development.

How does OpenRouter work?

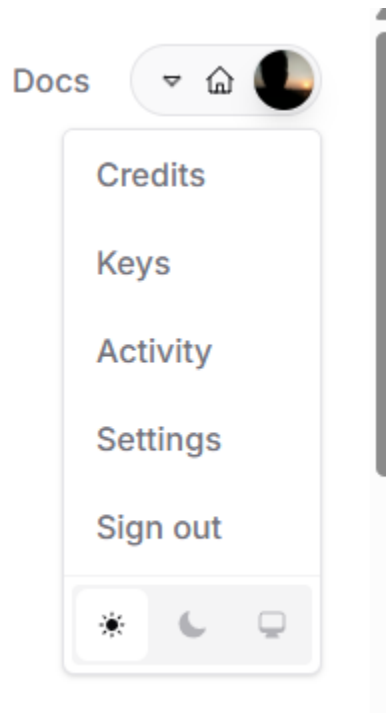
OpenRouter acts as a proxy between your application and the various underlying language models.

1. Your application makes an API request to OpenRouter's endpoint.
2. OpenRouter receives the request and decides which language model should handle it based on configurations or availability.
3. The request is forwarded to the chosen language model's API.
4. The response from the language model is sent back to OpenRouter.
5. OpenRouter forwards the response back to your application.

This abstraction layer allows for seamless integration with different LLMs without needing to change your application's code significantly.

Get an OpenRouter API key:


- Go to the [OpenRouter](#) website.
- Sign up for an account using your email or preferred method.
- Once logged in, navigate to the API Keys section.



- Generate a new API key. Copy and securely store this key, as it will be needed to access OpenRouter's services.

API Keys

Create API Key

Manage your API keys to access all models from OpenRouter 

vs_code_api

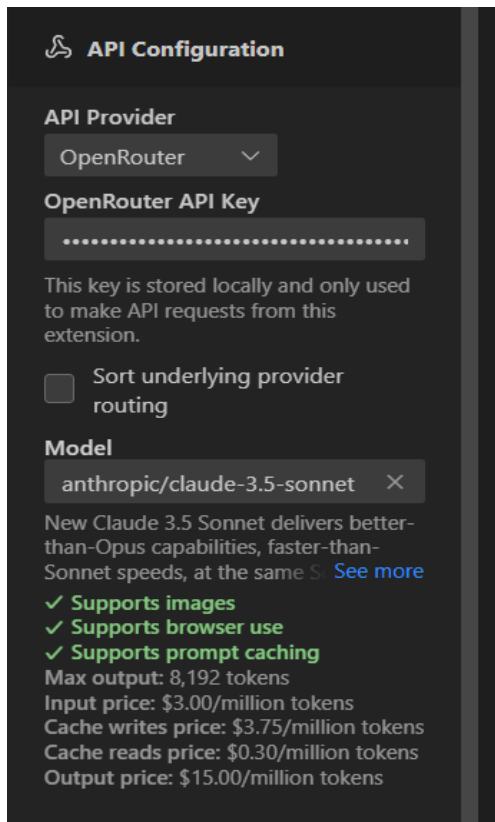
\$0.000 used • Unlimited

sk-or-v1-15b ... ad5

⋮

Cline setup

- Select "cline" extension in the options on the left side.
- Select "API provider" as "OpenRouter" and paste the API key obtained from the OpenRouter site.
- Select the model based on the requirements.



API Configuration

API Provider

OpenRouter ▼

OpenRouter API Key

.....

This key is stored locally and only used to make API requests from this extension.

☐ Sort underlying provider routing

Model

anthropic/claude-3.5-sonnet ✕

New Claude 3.5 Sonnet delivers better-than-Opus capabilities, faster-than-Sonnet speeds, at the same \$ [See more](#)

- ✓ Supports images
- ✓ Supports browser use
- ✓ Supports prompt caching

Max output: 8,192 tokens

Input price: \$3.00/million tokens

Cache writes price: \$3.75/million tokens

Cache reads price: \$0.30/million tokens

Output price: \$15.00/million tokens