

# Settings of Software Development Environment for Python projects





- Software Download and Install
- Anaconda
  - Create and activate a virtual environment
  - Install libraries
- Git & GitHub
  - Create a repository on GitHub
  - Pull and Push request
- Virtual Studio Code (VSCode)
  - Create a new workspace / new file
  - Choose appropriate kernel / virtual environment
  - Run Python / Jupyter notebook files
  - Connect to GitHub
  - Clone from / Submit to GitHub repository

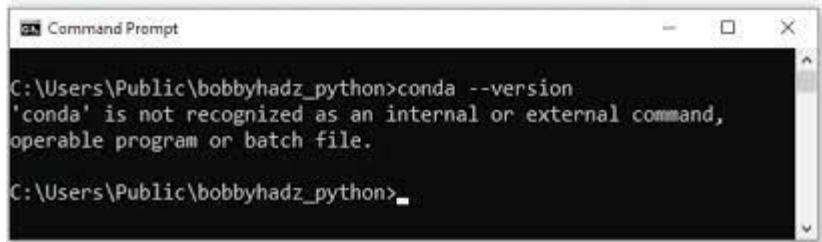
# Content



# Prerequisites: Download and Install

Software	Download link	Install Video (win11)	Install Video (Mac)
Anaconda	<a href="#">Download Now   Anaconda</a>	<a href="#">How to Install Anaconda and Jupyter Notebook on Windows 11 (2023) (youtube.com)</a>	<a href="#">Install Anaconda Python, Jupyter Notebook And Spyder on Mac / MacOS (2024) (youtube.com)</a>
Git	<a href="#">Git - Downloads (git-scm.com)</a>	<a href="#">How to Download &amp; Install Git on Windows 11 - Git Installation (youtube.com)</a>	Git has been installed into Mac by default
VS Code	<a href="#">Download Visual Studio Code - Mac, Linux, Windows</a>	<a href="#">bash - Change the default terminal in Visual Studio Code - Stack Overflow</a>	<a href="#">How to Install Visual Studio Code on Mac   Install VSCode on macOS (2024) (youtube.com)</a>

# Environment variables setting (windows)

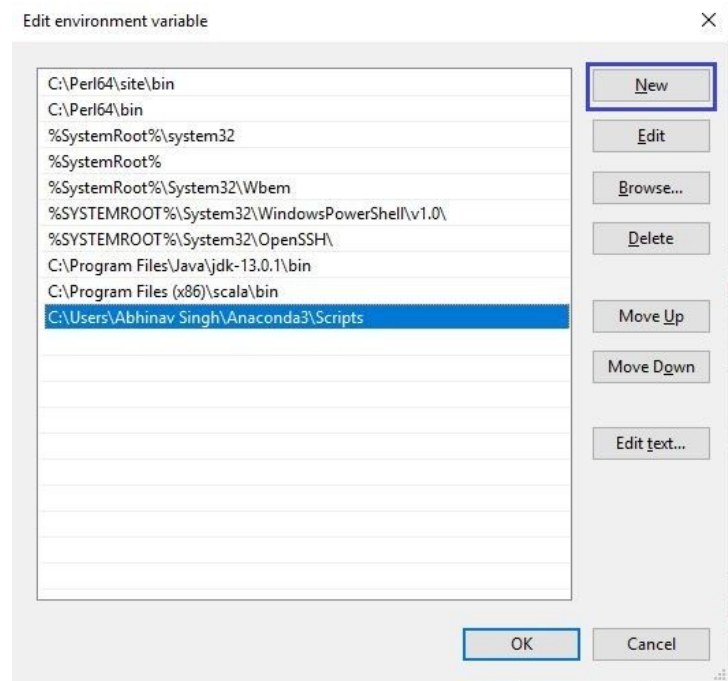


```
Command Prompt

C:\Users\Public\bobbyhadz_python>conda --version
'conda' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Public\bobbyhadz_python>
```

Add conda to environment variable: [How to Fix Conda Not Recognized as Internal or External Command](#)  
[Add Conda to Environment Variable - YouTube](#)





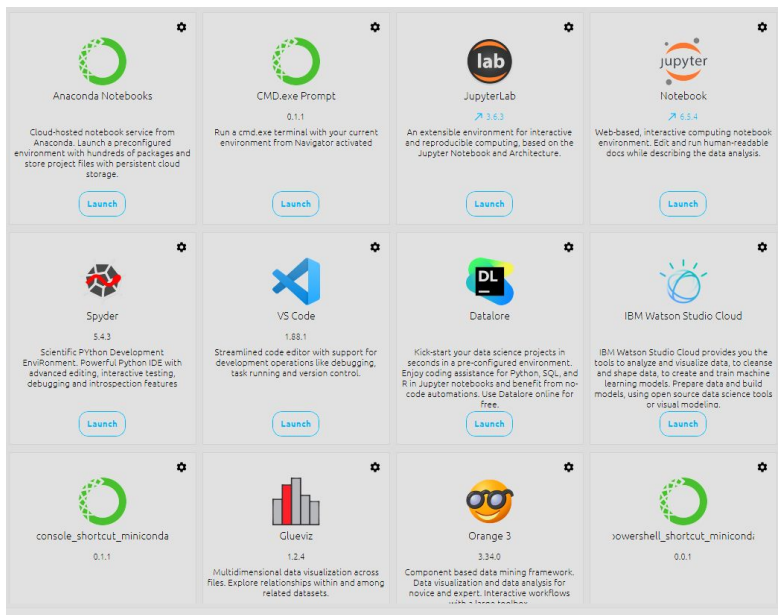
# Content

- Software Download and Install
- Anaconda
  - Create and activate a virtual environment
  - Install libraries
- Git & GitHub
  - Create a repository on GitHub
  - Pull and Push request
- Virtual Studio Code (VSCode)
  - Create a new workspace / new file
  - Choose appropriate kernel / virtual environment
  - Run Python / Jupyter notebook files
  - Connect to GitHub
  - Clone from / Submit to GitHub repository



# Anaconda

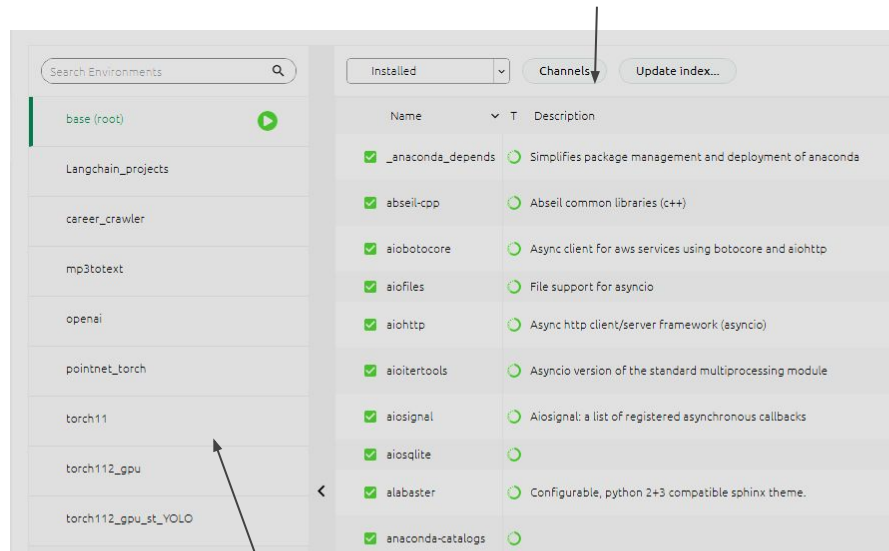
## Integrated Development Environments (IDEs)



A grid of 12 IDEs available in Anaconda, each with a logo, name, version, description, and a 'Launch' button.

IDE	Version	Description
Anaconda Notebooks		Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.
CMD.exe Prompt	0.1.1	Run a cmd.exe terminal with your current environment from Navigator activated
JupyterLab	3.6.3	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Jupyter Notebook	6.5.4	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Spyder	5.4.3	Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
VS Code	1.88.1	Streamlined code editor with support for development operations like debugging, task running and version control.
Datalore		Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Datalore online for free.
IBM Watson Studio Cloud		IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.
console_shortcut_miniconda	0.1.1	
Glueviz	1.2.4	Multidimensional data visualization across files. Explore relationships within and among related datasets.
Orange 3	3.34.0	Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a flow toolbox.
xowshell_shortcut_miniconda	0.0.1	

## Packages Management



Screenshot of the Anaconda Packages Management interface. The left sidebar shows a list of environments, and the right pane shows the installed packages for the selected environment.

**Search Environments:** [Search Bar]

**Installed:** [Dropdown Menu]

**Channels:** [Dropdown Menu]

**Update index...** [Button]

Name	✓	T	Description
base (root)			
Langchain_projects			
career_crawler			
mp3totext			
openai			
pointnet_torch			
torch11			
torch112_gpu			
torch112_gpu_st_YOLO			
_anaconda_depends	✓		Simplifies package management and deployment of anaconda
abseil-cpp	✓		Abseil common libraries (c++)
aiobotocore	✓		Async client for aws services using boto3 and aiohttp
aiofiles	✓		File support for asyncio
aiohttp	✓		Async http client/server framework (asyncio)
aioitertools	✓		Asyncio version of the standard multiprocessing module
aiosignal	✓		Aiosignal: a list of registered asynchronous callbacks
aiosqlite	✓		
alabaster	✓		Configurable, python 2+3 compatible sphinx theme.
anaconda-catalogs	✓		

## Virtual Environment Management



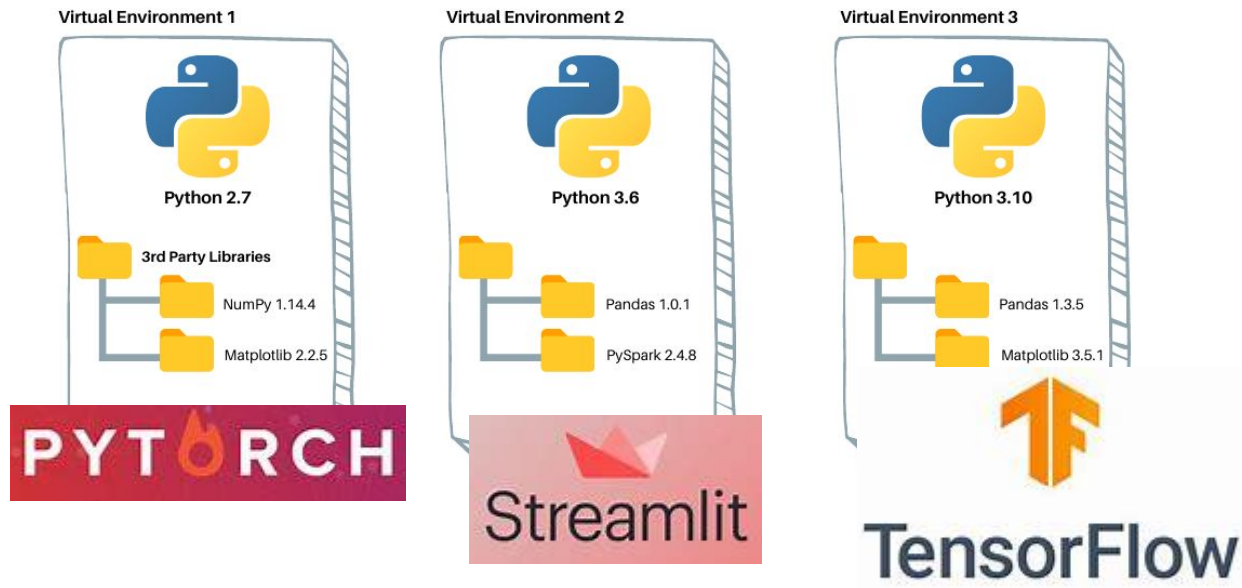
# Virtual Environment

Everything about Pandas





# Virtual Environment







# Anaconda commands

# To check the anaconda was installed successfully

```
conda -version
```

# To create a new virtual environment

```
conda create - - name your_virtual_environment python=3.10 # optional: you can specify the python version
```

# To show the virtual environments list

```
conda env list
```

# To activate the virtual environment

```
conda activate your_virtual_environment
```

# Deactivate from an active environment

```
conda deactivate
```



# Content

- Software Download and Install
- Anaconda
  - Create and activate a virtual environment
  - Install libraries
- Git & GitHub
  - Create a repository on GitHub
  - Pull and Push request
- Virtual Studio Code (VSCode)
  - Create a new workspace / new file
  - Choose appropriate kernel / virtual environment
  - Run Python / Jupyter notebook files
  - Connect to GitHub
  - Clone from / Submit to GitHub repository




# Create a new repository

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*  
 stemgene / Repository name \*  
your\_repository  
your\_repository is available.

Input your repository name

Great repository names are short and memorable. Need inspiration? How about [glowing-octo-barnacle](#)?

Description (optional)

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Click “Add a README file”

Add .gitignore

.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: **None**

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set [main](#) as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

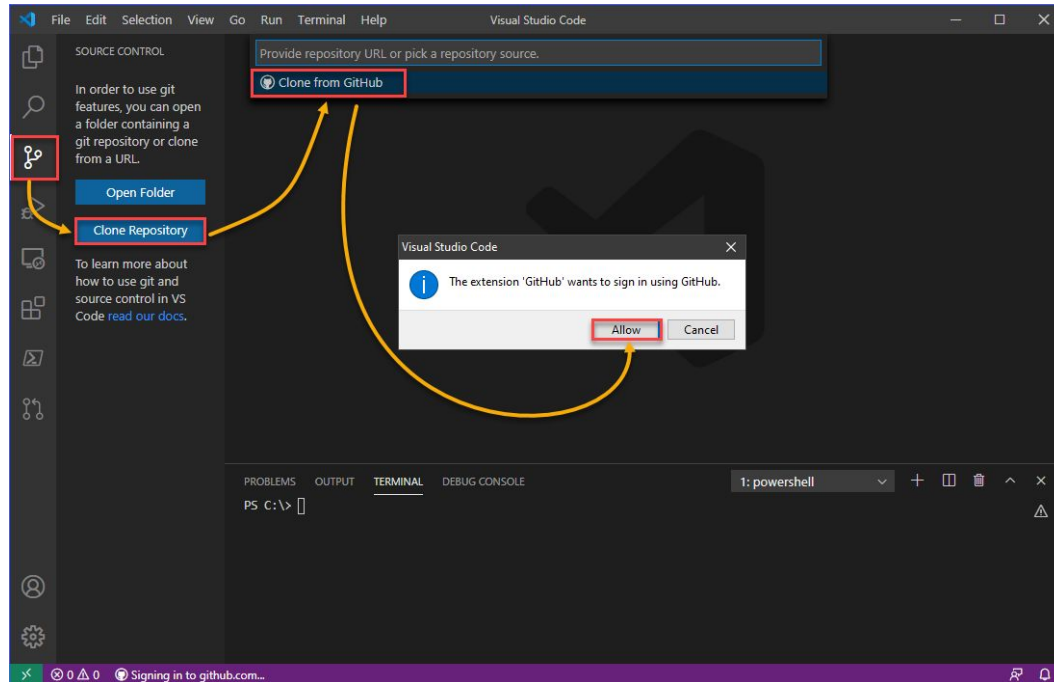
Create repository



# Content

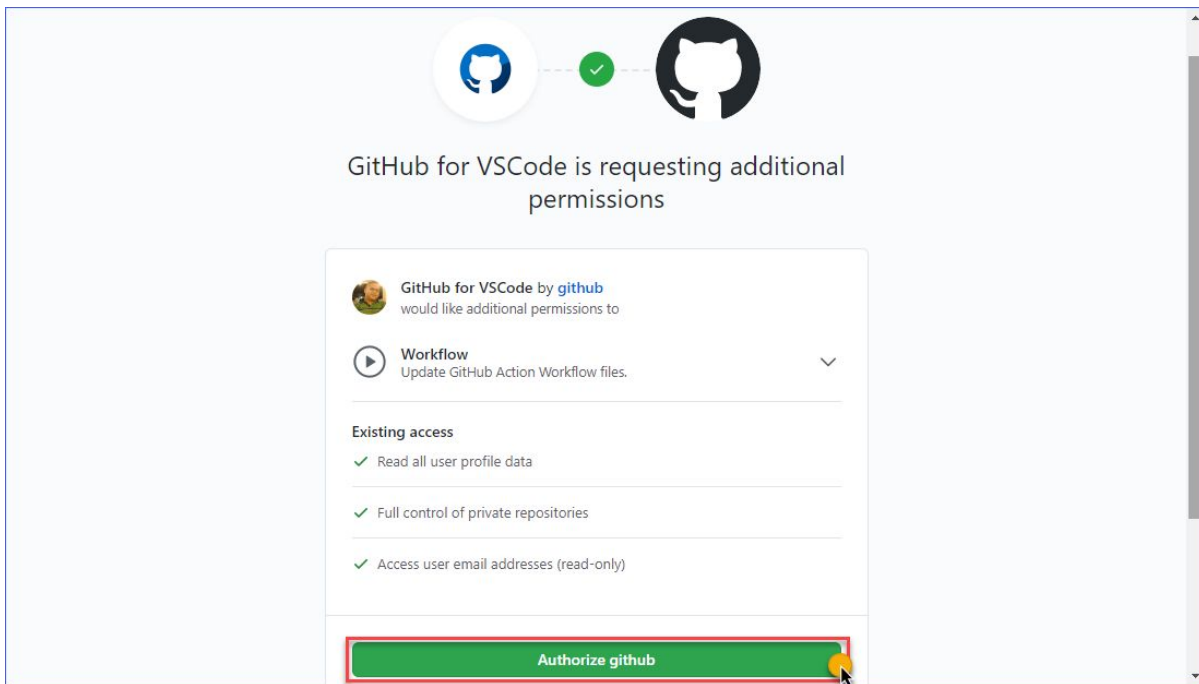
- Software Download and Install
- Anaconda
  - Create and activate a virtual environment
  - Install libraries
- Git & GitHub
  - Create a repository on GitHub
  - Pull and Push request
- Virtual Studio Code (VSCode)
  - Create a new workspace / new file
  - Choose appropriate kernel / virtual environment
  - Run Python / Jupyter notebook files
  - Connect to GitHub
  - Clone from / Submit to GitHub repository

# Clone repository to local

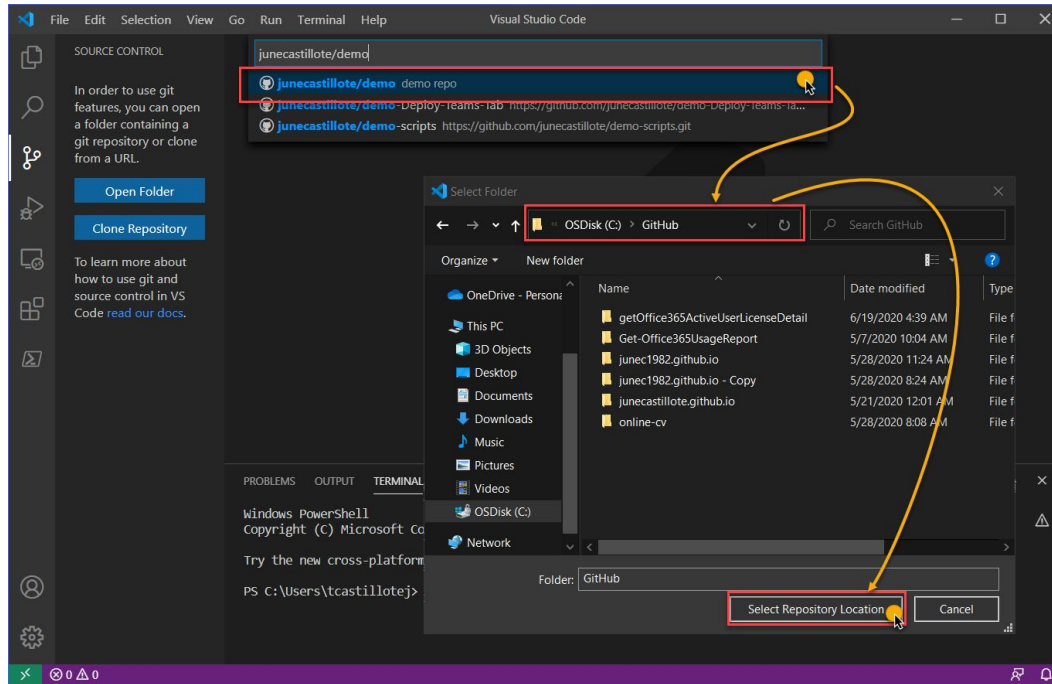




# Connect VS Code with GitHub



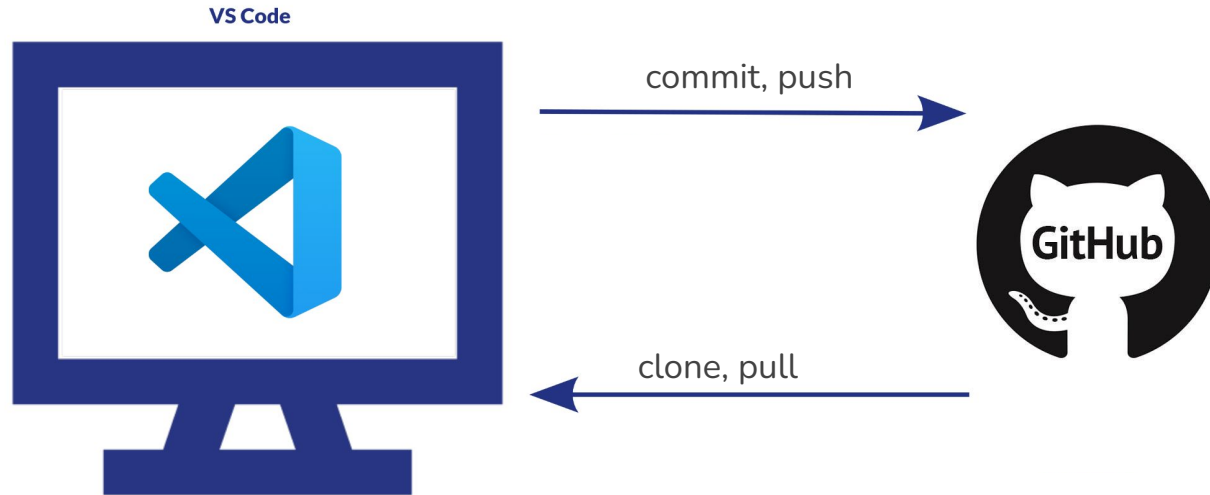
# Find a local folder to restore repository



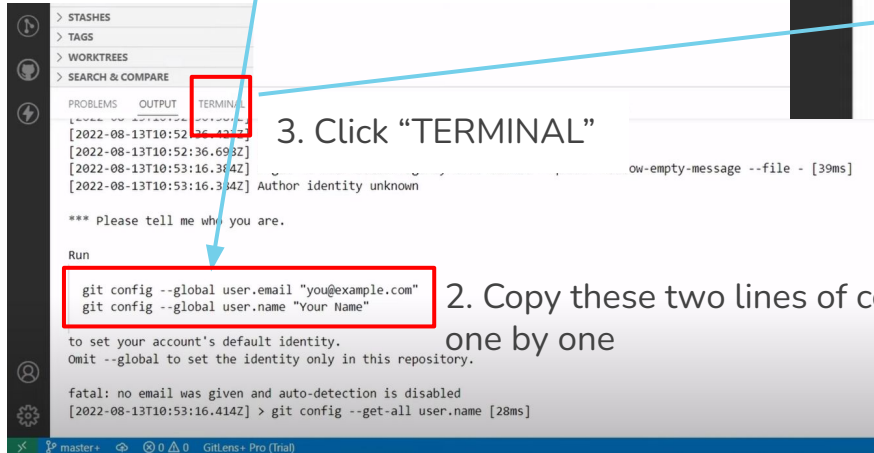
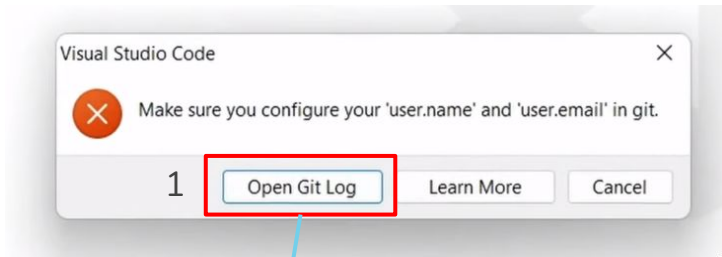




# Connect VS Code with GitHub

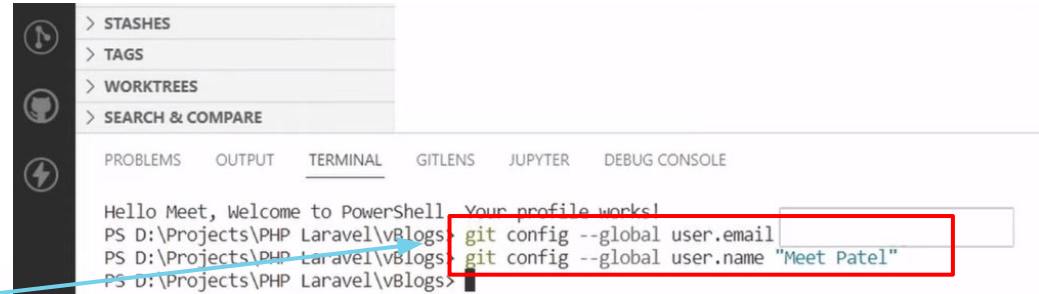


# Fix the 'user.name' / 'user.email' issue



3. Click "TERMINAL"

2. Copy these two lines of code one by one



4. Input your GitHub user.name and user.email



# Happy Coding

