Quick Start Guide: VS Code, Python, and Jupyter

Course Computing Setup

Goal

By the end of this guide, you will be able to:

- Write and run Python code
- Use Jupyter notebooks
- Make plots and analyze data

All work in this course will be done **inside Visual Studio Code (VS Code)** using a Python environment named `geo-env`.

Step 1: Install Visual Studio Code

Download and install VS Code from:

`https://code.visualstudio.com`

Use the default installation options.

Windows users: If the installer asks whether to *Add VS Code to PATH*, check that box.

Step 2: Install Python (Official Version)

Install Python from:

`https://www.python.org`

Important for Windows

During installation:

- Check **“Add Python to PATH”**
- Use default installation options

Verify Installation (Mac and Windows)

Open a terminal:

- Mac: Applications → Utilities → Terminal
- Windows: Command Prompt or PowerShell

Run:

```
python --version
pip --version
```

If `python` does not work on Windows, try:

```
py --version
```

Step 3: Install VS Code Extensions

Open VS Code, go to the Extensions tab, and install:

- **Python** (Microsoft)
- **Jupyter** (Microsoft)

These allow Python scripts and Jupyter notebooks to run inside VS Code.

Step 4: Create a Course Folder (Finder / File Explorer)

You do **not** need the terminal for this step.

Mac

1. Open **Finder**
2. Click **Documents**
3. Right-click → New Folder
4. Name the folder:

```
earth_history_code
```

5. Open VS Code
6. File → Open Folder
7. Select `earth_history_code`

Windows

1. Open **File Explorer**
2. Click **Documents**
3. Right-click → New → Folder
4. Name the folder:

```
earth_history_code
```

5. Open VS Code
6. File → Open Folder
7. Select `earth_history_code`

Important: All terminals, environments, scripts, and notebooks must be created inside this folder.

Step 5: Create the Python Environment (geo-env)

Open the VS Code terminal:

Terminal → New Terminal

Mac

```
python -m venv geo-env
source geo-env/bin/activate
python -m pip install --upgrade pip
```

Windows (PowerShell)

```
python -m venv geo-env
.\geo-env\Scripts\Activate.ps1
python -m pip install --upgrade pip
```

If activation fails, run once:

```
Set-ExecutionPolicy -Scope CurrentUser RemoteSigned
```

Then try activating again.

Windows (Command Prompt)

```
python -m venv geo-env
geo-env\Scripts\activate.bat
python -m pip install --upgrade pip
```

Check That It Worked

```
python --version
```

The Python path should include geo-env.

Step 6: Install Required Packages

Inside the earth_history_code folder, create a file named:

```
requirements.txt
```

Paste the following:

```
numpy
scipy
pandas
matplotlib
```

```
jupyter
ipykernel

xarray
netcdf4
h5py

scikit-learn

scikit-image
pillow

tqdm

ruff
black
pytest
```

Install the packages:

```
pip install -r requirements.txt
```

Step 7: Register geo-env as a Jupyter Kernel

```
python -m ipykernel install --user --name geo-env --display-name "Python (
   geo-env)"
```

Step 8: Select the Python Interpreter in VS Code

1. Press **Ctrl+Shift+P** (Windows) or **Cmd+Shift+P** (Mac)
2. Run **Python: Select Interpreter**
3. Choose the interpreter inside `geo-env`

Step 9: Test with a Jupyter Notebook

In VSCode, go to select File, then New File from the dropdown menu. Then create a new file named:

```
test.ipynb
```

In the top-right of the notebook, select:

`Python (geo-env)`

Run:

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 10, 200)
plt.plot(x, np.sin(x))
plt.title("geo-env works")
plt.show()
```

If you see a plot, your setup is complete.

Common Issues

Kernel Not Showing Up

```
python -m pip install ipykernel
python -m ipykernel install --user --name geo-env --display-name "Python (
   geo-env)"
```

Restart VS Code.

VS Code Using the Wrong Python

Run **Python: Select Interpreter** again and choose `geo-env`.

Permission Errors (Mac)

```
python -m pip install --upgrade pip setuptools wheel
```

You Are Done

If something breaks later:

- Restart VS Code
- Activate `geo-env`
- Re-select the interpreter