

Programming Techniques

Student Setup Guide

Python 3, Anaconda, VS Code, Jupyter Notebooks and Google Colab

This guide is intended for students with no previous programming experience. Follow the steps in order and stop at each **Verification checkpoint**. If the verification works, continue to the next section.

Important

For this course you only need **Python 3 and the Python Standard Library**. Anaconda is allowed and convenient, but no external scientific Python packages are required for the course itself.

1. What you need

By the end of this guide you should have:

- Python 3 installed locally;
- either a standard Python installation or Anaconda;
- Visual Studio Code installed;
- the Microsoft **Python** and **Jupyter** extensions installed in VS Code;
- a working local `.py` file;
- a working local Jupyter Notebook `.ipynb`;
- access to Google Colab;
- a dedicated Google Drive folder for the course.

1.1. Choose one Python installation route

	Route A: Anaconda	Route B: Standard Python
Best for	Beginners who want Python, environment management, and Jupyter together	Students who prefer a smaller, standard Python installation
Includes Python	Yes	Yes
Includes Jupyter	Normally yes	Install separately with <code>pip</code>
Recommended for this course	Yes, easiest path	Also fully supported

Important

Choose **one main route**. Installing both Anaconda and another Python distribution is possible, but beginners can easily select the wrong interpreter in VS Code. If you already have Python installed, verify it first before installing another copy.

2. Step 1 – Check whether Python is already installed

Open a terminal:

- **Windows:** Start menu → search for *Terminal*, *PowerShell*, or *Command Prompt*.

- **macOS:** Applications → Utilities → Terminal.
- **Linux:** open your usual Terminal application.

Try the appropriate command:

```
# Windows
python --version
py --version

# macOS / Linux
python3 --version
```

If you see something similar to `Python 3.x.y`, Python 3 is already available.

Verification checkpoint

You should see a version beginning with `Python 3`. If the command is not found, continue with one of the installation routes below.

3. Step 2A – Install Python with Anaconda

Use this section if you choose **Route A**.

3.1. Windows

1. Go to the official Anaconda website and download the current **Anaconda Distribution** installer for Windows.
2. Run the downloaded installer.
3. For a personal computer, the default per-user installation is normally appropriate.
4. Keep the default installation folder unless you have a specific reason to change it.
5. Complete the installation.
6. Open **Anaconda Prompt** from the Start menu.
7. Run:

```
conda --version
python --version
```

3.2. macOS

1. Go to the official Anaconda website and download the installer for macOS.
2. Choose the installer matching your Mac architecture when prompted (Apple Silicon or Intel).
3. Run the installer and follow the default installation procedure.
4. Open Terminal.
5. Run:

```
conda --version
python --version
```

If `conda` is not found immediately, close and reopen Terminal once, then try again.

3.3. Linux

1. Go to the official Anaconda website and download the Linux installer for your architecture.
2. Open Terminal and move to the folder containing the downloaded `.sh` installer, usually `Downloads`.
3. Run the installer with Bash, replacing the file name with the one you downloaded:

```
cd ~/Downloads
bash Anaconda3-<version>-Linux-<architecture>.sh
```

4. Read and accept the license when requested.
5. The default installation location is normally fine.
6. Allow the installer to initialize Conda if asked.
7. Close and reopen Terminal.
8. Verify:

```
conda --version
python --version
```

Verification checkpoint

Both commands should return a version number. If they do, your Anaconda installation is ready.

4. Step 2B – Install standard Python 3 without Anaconda

Use this section if you choose **Route B**.

4.1. Windows

Current Python releases for Windows use the official **Python Install Manager**.

1. Go to python.org/downloads.
2. Download and install the Python Install Manager offered for Windows.
3. When installation is complete, open a **new** Terminal or PowerShell window.
4. Run:

```
python --version
py --version
```

If no Python runtime has yet been installed, the manager can install the latest stable Python when first needed. You can also inspect installed runtimes with:

```
py list
```

Verification checkpoint

At least `python -version` or `py -version` should show Python 3.

4.2. macOS

1. Go to python.org/downloads.
2. Download the current macOS .pkg installer for Python 3.
3. Double-click the downloaded package.
4. Follow the standard installation procedure and keep the default options.
5. After installation, open the folder `/Applications/Python 3.x/`.
6. Run `Install Certificates.command` once if it is provided by the installer.
7. Open a **new** Terminal window and verify:

```
python3 --version
python3 -m pip --version
```

Important

Do not delete or modify the Python installation supplied internally by macOS or Apple developer tools. Install your course Python alongside it and use the Python installed from python.org.

4.3. Linux

Most Linux distributions already provide Python 3. First try:

```
python3 --version
```

If Python is not installed, use your distribution's package manager. For example:

Ubuntu / Debian-based systems

```
sudo apt update
sudo apt install python3 python3-pip python3-venv
```

Fedora-based systems

```
sudo dnf install python3 python3-pip
```

Then verify:

```
python3 --version
python3 -m pip --version
```

Verification checkpoint

You should see Python 3 and a working pip installation.

5. Step 3 – Test Python before installing the IDE

Create a folder for the course, for example:

- Windows: `Documents\ProgrammingTechniques`
- macOS/Linux: `~/Documents/ProgrammingTechniques`

Create a text file named `hello.py` containing:

```
print("Hello, Programming Techniques!")
```

Run it from a terminal.

Windows

```
python hello.py
```

macOS/Linux

```
python3 hello.py
```

If you installed Anaconda, `python hello.py` will normally work from an Anaconda-enabled terminal.

Verification checkpoint

The terminal should print exactly:
Hello, Programming Techniques!

6. Step 4 – Install Visual Studio Code

1. Go to code.visualstudio.com/download.
2. Download the installer for your operating system.
3. Install VS Code using the default options.

Platform notes:

- **Windows:** the User Installer is appropriate for most students.
- **macOS:** open the `.dmg` file and drag Visual Studio Code into **Applications**.
- **Linux:** use the package appropriate for your distribution (for example `.deb`, `.rpm`, or Snap).

Open VS Code after installation.

7. Step 5 – Install the VS Code extensions

In VS Code:

1. Click the **Extensions** icon in the left sidebar.
2. Search for **Python**.
3. Install the extension named **Python** published by **Microsoft**.
4. Search for **Jupyter**.
5. Install the extension named **Jupyter** published by **Microsoft**.

Tip

Do not choose an extension only because its name contains “Python” or “Jupyter”. For this course, use the official Microsoft Python and Jupyter extensions.

8. Step 6 – Select the correct Python interpreter in VS Code

This is one of the most important steps.

1. Open VS Code.
2. Open your course folder with **File** → **Open Folder**.

3. Open the Command Palette:

- Windows/Linux: **Ctrl+Shift+P**
- macOS: **Cmd+Shift+P**

4. Search for **Python: Select Interpreter**.

5. Select the Python installation you intend to use.

If you installed Anaconda, choose the interpreter identified as a Conda/Anaconda environment. If you installed standard Python, choose the Python 3 interpreter you installed.

Now open `hello.py` and click the triangular **Run Python File** button in the top-right area of the editor.

Verification checkpoint

The VS Code terminal should print:
Hello, Programming Techniques!

9. Step 7 – Prepare Jupyter Notebooks locally

9.1. If you installed Anaconda

Anaconda normally provides the components needed for Jupyter. In VS Code:

1. Create a new file named `test_notebook.ipynb`.
2. VS Code should open the notebook interface.
3. Click **Select Kernel** in the top-right corner.
4. Choose your Anaconda/Conda Python environment.

9.2. If you installed standard Python

Install Jupyter Notebook and the IPython kernel in your Python environment.

Windows

```
python -m pip install notebook ipykernel
```

macOS/Linux

```
python3 -m pip install notebook ipykernel
```

Then restart VS Code if necessary.

Tip

If your operating system prevents global `pip` installation, use a virtual environment instead. VS Code can select that environment as both interpreter and notebook kernel.

9.3. Test the notebook

In `test_notebook.ipynb`, create a code cell containing:

```
x = 5
y = 7
print(x + y)
```

Run the cell with the small triangular button next to it.

Verification checkpoint

The output below the cell should be:
12

10. Step 8 – Optional but recommended: create a clean local environment

This section is useful if you use standard Python without Anaconda. It keeps course tools separate from the rest of your computer.

From your course folder:

Windows

```
python -m venv .venv
.venv\Scripts\activate
python -m pip install notebook ipykernel
```

macOS/Linux

```
python3 -m venv .venv
source .venv/bin/activate
python -m pip install notebook ipykernel
```

Then select `.venv` with **Python: Select Interpreter** in VS Code and use the same environment as the notebook kernel.

To leave the environment later:

```
deactivate
```

11. Step 9 – Set up Google Colab and Google Drive

Google Colab is a hosted Jupyter Notebook service. It runs in the browser, so it does not require your local Python installation.

11.1. Create a course folder in Google Drive

1. Sign in to the Google account you will use for the course.
2. Open Google Drive.
3. Create a new folder named, for example, **Programming Techniques**.
4. Inside it, optionally create subfolders such as:
 - Lectures
 - Exercises
 - Project

11.2. Create a Colab notebook

1. Open colab.research.google.com while signed in with the same Google account.
2. Create a new notebook.
3. Rename it, for example, `setup_test.ipynb`.

- Colab notebooks are stored in Google Drive. If necessary, move the notebook in Google Drive into your **Programming Techniques** folder.

Create a code cell:

```
print("Colab is working!")
```

Run the cell.

Verification checkpoint

The output should be:
Colab is working!

11.3. Use files from your Google Drive inside Colab

Storing the notebook in Drive and accessing arbitrary files from Drive are two different things. To access files programmatically from the Colab runtime, mount Google Drive.

Run this cell:

```
from google.colab import drive
drive.mount('/content/drive')
```

Authorize access when requested. Your Drive will then be available under a path beginning with:

```
/content/drive/MyDrive/
```

For example, if your folder is called **Programming Techniques**, a file could be accessed as:

```
path = "/content/drive/MyDrive/Programming Techniques/Exercises/data.txt"
```

Important

Colab code runs on a temporary virtual machine. Files created only in the temporary runtime may disappear when the runtime is reset. Save important notebooks in Drive and save important data explicitly to Drive or download it locally.

12. Step 10 – Final self-check

Before the first practical class, verify all of the following.

- I can open a terminal.
- I can obtain a Python 3 version number.
- I can run `hello.py` from a terminal.
- VS Code is installed.
- The Microsoft Python extension is installed.
- The Microsoft Jupyter extension is installed.
- VS Code has the correct Python interpreter selected.
- I can run a `.py` file in VS Code.
- I can create and run a local `.ipynb` notebook.
- I can create and run a Colab notebook.

11. My Colab notebook is stored in my Google Drive account.
12. I know how to mount Drive in Colab when a notebook needs to read or write files.

13. Common problems

Problem: python is not found

- Windows: try `py -version` and verify the Python Install Manager installation.
- macOS/Linux: try `python3 -version` instead of `python -version`.
- If you installed software very recently, close and reopen the terminal.

Problem: VS Code uses the wrong Python

Open the Command Palette and run **Python: Select Interpreter**. Choose the intended Python/-Conda environment again.

Problem: a notebook says that no kernel is available

- Check that the Jupyter extension is installed.
- Check that you selected the intended Python interpreter.
- If you use standard Python, install `notebook` and `ipykernel` in that environment.

Problem: pip installs packages into the wrong Python

Prefer:

```
python -m pip ...
```

or on macOS/Linux:

```
python3 -m pip ...
```

This makes the Python interpreter used by `pip` explicit.

Problem: Colab cannot find a file in Drive

- Mount Drive first.
- Check the exact folder and file names.
- Remember that spaces in folder names are part of the path.
- Use the Colab file browser to inspect `/content/drive/MyDrive/`.

14. Official references

The installation steps in this guide were checked against the official documentation available in August 2026.

- Python downloads: <https://www.python.org/downloads/>
- Python on Windows: <https://docs.python.org/3/using/windows.html>
- Python on macOS: <https://docs.python.org/3/using/mac.html>
- Python on Unix/Linux: <https://docs.python.org/3/using/unix.html>

- Anaconda documentation: <https://www.anaconda.com/docs/main>
- Visual Studio Code setup: <https://code.visualstudio.com/docs/getstarted/overview>
- VS Code Jupyter notebooks: <https://code.visualstudio.com/docs/datascience/jupyter-notebooks>
- Project Jupyter installation: <https://jupyter.org/install>
- Google Colab FAQ: <https://research.google.com/colaboratory/faq.html>
- Colab and Google Drive: <https://colab.research.google.com/notebooks/io.ipynb>

If every verification checkpoint succeeds, your computer is ready for **Programming Techniques**.