

Hands-on Manual: Setting Up Your Python Environment for Advanced Hydraulics

Welcome to the hands-on session for Advanced Hydraulics. In this course, we will use Python as a powerful tool for numerical modeling, data analysis, and visualization. Python is the industry and academic standard for scientific computing, and mastering it will be an invaluable skill in your career.

This manual will guide you through setting up a professional, robust, and isolated Python environment on your personal computer. By the end of this session, you will have:

1. Installed Python.
2. Installed a modern code editor (VS Code or Cursor).
3. Understood and created a virtual environment for our project.
4. Installed the essential scientific libraries.
5. Created and opened our course project.

Let's get started!

Part 1: Installing Python

Python is the programming language we will be using. We need to install the Python interpreter, which is the program that reads and executes our Python code.

It is crucial to install Python correctly. Follow the instructions for your operating system.

For Windows Users:

1. **Download the Installer:** Go to the official Python website: <https://www.python.org/downloads/>
2. Click the "Download Python 3.13.5" button (it will detect you're on Windows).
3. **Run the Installer:** Once downloaded, run the installer.
4. **IMPORTANT:** On the first screen of the installation, check the box at the bottom that says "**Add Python 3.13 to PATH**". This is a critical step that allows you to run Python from the command line easily.
5. Choose "Install Now" for the recommended installation.
6. Once the installation is complete, close the installer.

For macOS Users:

While macOS comes with a version of Python, we recommend installing the latest version from the official website to ensure consistency.

1. **Download the Installer:** Go to the official Python website: <https://www.python.org/downloads/>
2. Click the "Download Python 3.13.5" button.
3. **Run the Installer:** Open the downloaded `.pkg` file and follow the installation prompts. The default settings are fine.

For Linux Users (Ubuntu/Debian):

Most Linux distributions come with Python 3 pre-installed. You can verify this and ensure you have the necessary tools (**pip** and **venv**) by running the following command in your terminal:

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

Verification (All Operating Systems):

Let's check that Python was installed correctly.

1. Open your terminal application:
 - **Windows:** Open the Start Menu, type **PowerShell** or **cmd**, and press Enter.
 - **macOS:** Open Finder, go to Applications -> Utilities -> Terminal.
 - **Linux:** Press **Ctrl+Alt+T** or find "Terminal" in your applications.
2. In the terminal, type the following command and press Enter:

```
python3 --version
# On some Windows systems, you might need to use 'python' instead
python --version
```

You should see an output like **Python 3.13.5**. As long as it's a version 3.6 or higher, you're good to go.

Part 2: Installing a Code Editor (VS Code / Cursor)

You can write Python in a simple text editor, but a modern code editor (or IDE - Integrated Development Environment) makes coding much easier with features like syntax highlighting, code completion, and debugging.

We recommend **VS Code** or its AI-powered fork, **Cursor**. They are nearly identical, but Cursor has powerful AI features built-in that can help you write and understand code faster. For this course, either is an excellent choice.

1. Download:

- **For Visual Studio Code (VS Code):** <https://code.visualstudio.com/>
- **For Cursor (VS Code with AI):** <https://cursor.sh/>

2. **Install:** Run the downloaded installer and follow the on-screen instructions. The default settings are fine.

3. Install the Essential Python Extension:

- Open VS Code or Cursor.
- On the left-hand side, click the **Extensions** icon (it looks like four squares).
- In the search bar, type **Python**.
- The first result should be from Microsoft. Click the **Install** button. This extension provides critical features like code analysis (IntelliSense) and Jupyter Notebook support.

Part 3: Virtual Environments - Your Project's Sandbox

This is one of the most important concepts for professional Python development.

What is a virtual environment? Imagine you are working on two different projects. Project A needs an older version of a library (e.g., **pandas 1.5**), while Project B needs the newest version (**pandas 2.1**). If you install these system-wide, they will conflict.

A virtual environment is a self-contained, isolated directory that contains a specific version of Python and all the packages required for a single project. It's like having a clean, separate workshop for each of your projects.

Benefits:

- **No Conflicts:** Prevents package version conflicts between projects.
- **Reproducibility:** Makes it easy for a colleague (or your professor!) to run your code by installing the exact same dependencies.
- **Cleanliness:** Keeps your global Python installation clean and tidy.

We will create one for our "Advanced Hydraulics" project.

Part 4: Setting Up Our Course Project Folder

Let's create a dedicated space for all our course-related work.

1. Create a Project Folder:

- On your computer, create a new folder in a location you can easily find (e.g., in your **Documents** or on your **Desktop**).
- Name the folder **Advanced_Hydraulics_Project**.

2. Open a Terminal in the Project Folder:

- **Easy Way (Windows/macOS):** Open VS Code/Cursor. Go to **File > Open Folder...** and select the **Advanced_Hydraulics_Project** folder you just created. Then, in VS Code/Cursor, go to the top menu and select **Terminal > New Terminal**. This will open a terminal panel already inside your project directory.
- **Manual Way:** Navigate to the folder using your terminal. For example:

```
# On Windows
cd C:\Users\YourUsername\Documents\Advanced_Hydraulics_Project

# On macOS/Linux
cd /Users/YourUsername/Documents/Advanced_Hydraulics_Project
```

3. Create the Virtual Environment:

- With your terminal open **inside the Advanced_Hydraulics_Project directory**, run the following command:

```
python -m venv venv
# On Linux/Mac, you might use 'python3'
python3 -m venv venv
```

- This command tells Python to run the **venv** module and create a new virtual environment in a folder named **venv**. You will now see a new **venv** folder inside your project directory.

4. Activate the Virtual Environment:

- You must "activate" the environment every time you work on the project.
- **Windows (PowerShell/cmd):**

```
.\venv\Scripts\activate
```

- **macOS/Linux (bash/zsh):**

```
source venv/bin/activate
```

- **Verification:** You'll know it's active because your terminal prompt will change to show **(venv)** at the beginning. This is your confirmation that you are now working inside the sandbox!

Part 5: Installing the Essential Scientific Libraries

Now that our virtual environment is active, we can install the packages we need for this course. We will use **pip**, Python's package installer. Because our environment is active, **pip** will install these libraries *only* inside our **venv** folder, not system-wide.

Here are the libraries we'll be using:

- **NumPy:** The fundamental package for numerical computation (arrays, matrices, linear algebra). Essential for handling data like river cross-sections or time-series data.
- **Matplotlib:** The most popular library for creating static, animated, and interactive visualizations (plots, charts, hydrographs).
- **SciPy:** Builds on NumPy and provides a large collection of algorithms for optimization, integration, statistics, and signal processing. We'll use it for more advanced numerical methods.
- **Pandas:** A powerful library for data manipulation and analysis. Perfect for reading, cleaning, and analyzing data from CSV files or Excel spreadsheets (e.g., rainfall or streamflow data).

- **SymPy:** A library for symbolic mathematics. It can solve equations, differentiate, and integrate symbolically, which is useful for deriving hydraulic formulas.
- **Jupyter:** Provides an interactive, web-based environment (JupyterLab and Jupyter Notebooks) that allows you to combine live code, equations, visualizations, and narrative text. It's an incredible tool for exploratory analysis and sharing results.

In your **activated** terminal, run this single command to install all of them:

```
pip install numpy matplotlib scipy pandas sympy jupyterlab
```

You will see **pip** download and install each package. This may take a few minutes.

Verification: To see what's installed in your environment, you can run:

```
pip list
```

You should see all the packages we just installed in the list.

Part 6: Opening the Project and Starting to Code

You are now fully set up! Let's see how to start working. You have two excellent options.

Option 1: Work inside VS Code / Cursor (Recommended)

This is a powerful, all-in-one workflow.

1. **Open the Project Folder:** If you haven't already, open your **Advanced_Hydraulics_Project** folder in VS Code/Cursor (**File > Open Folder...**).
2. **Select the Python Interpreter:** VS Code needs to know it should use the Python from our virtual environment.
 - Press **Ctrl+Shift+P** (or **Cmd+Shift+P** on Mac) to open the Command Palette.
 - Type **Python: Select Interpreter**.
 - Choose the option that includes **(`venv`: `venv`)**. It should be listed as a recommended environment. This links VS Code to your sandbox.
3. **Create Your First Jupyter Notebook:**
 - In the File Explorer panel on the left, click the "New File" icon.
 - Name the file **session1_introduction.ipynb**. The **.ipynb** extension is for Jupyter Notebooks.
 - VS Code will open the notebook interface.
 - In the first cell, type the following code:

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

print("All libraries loaded successfully!")
```

```
print(f"NumPy version: {np.__version__}")  
print(f"Pandas version: {pd.__version__}")
```

- Press **Shift+Enter** to run the cell. The output should appear directly below the cell, confirming everything is working.

Option 2: Work in JupyterLab (Classic Browser-based Interface)

1. **Launch JupyterLab:** Make sure your terminal is inside the project folder and the **(venv)** is active. Then, run the command:

```
jupyter lab
```

2. This command will automatically open a new tab in your web browser with the JupyterLab interface.
3. **Create a Notebook:**
 - The file browser on the left shows the contents of your **Advanced_Hydraulics_Project** folder.
 - Under "Launcher", click on "Python 3 (ipykernel)" to create a new notebook.
 - You can then enter and run code in the cells, just like in VS Code.

Congratulations!

You have successfully set up a complete, professional Python environment for scientific computing. You are now ready to tackle the challenges of Advanced Hydraulics using the power of code.