

# Python Setup and Environment Readiness

A student-friendly guide to installing Python, checking PATH, confirming Conda, creating a virtual environment, and running your first Flask Hello World app on localhost.

Python installed

PATH understood

pip verified

venv created

Flask installed

localhost working



## What this lesson is really about

Before writing real code, you must make sure your laptop, terminal, interpreter, package manager, and environment all point to the same setup.

# Lesson roadmap

This lesson is split into four tracks. Follow them in order. Each one removes one common category of beginner confusion.

## Track order

1) Download and install Python - 2) Check PATH and terminal commands - 3) Confirm Conda or create a venv - 4) Install Flask and run localhost

## Track 1 - Download and install Python

Install Python from the official website. On Windows, the most important installer option is the one that adds Python to PATH.

- Download the latest stable version from python.org.
- During installation, tick 'Add Python to PATH' if the option appears.
- After installation, close and reopen Command Prompt or PowerShell.

```
python --version
py --version
python -m pip --version
```

### What these commands check

They confirm whether Python, the Windows launcher, and pip are visible from the terminal.

## Track 2 - Fix PATH and understand what is running

If python is not recognized, do not panic. First check whether py works. Then confirm where the executable is being picked from.

- Use 'where python' and 'where py' to see the executable paths.
- If only 'py' works, Python is installed but PATH may need correction.
- Typical Windows folders include the main Python folder and its Scripts folder.

```
where python
where py
C:\Users\\AppData\Local\Programs\Python\Python3x\
C:\Users\\AppData\Local\Programs\Python\Python3x\Scripts
\\
```

### Best habit

Use 'python -m pip' instead of plain 'pip' so pip stays tied to the same interpreter.

## Environment setup

A project environment keeps package installs isolated. This prevents one project from breaking another.

### Track 3 - Check Conda, then create a clean project environment

Some laptops already have Anaconda or Miniconda installed. Check first. If not, use Python's built-in venv for the first lesson.

- Run 'conda --version' to see whether Conda exists.
- Run 'conda info --envs' to see which environment is active.
- For beginners, venv is the simplest path because it comes with Python.

```
conda --version
conda info --envs
mkdir hello_flask
cd hello_flask
python -m venv .venv
.venv\Scripts\activate
```

#### How to know it worked

After activation, your prompt usually shows something like (.venv). That means installs now go into this project only.

### Optional Conda path

If your class is using Conda, the equivalent idea is:

```
conda create -n hello-flask python=3.13
conda activate hello-flask
```

### Track 4 - Install Flask and build Hello World

Once the environment is active, install Flask, create a tiny app.py file, and run it locally in the browser.

- Install Flask inside the active environment, not globally.
- Create a file named app.py and add a basic route.
- Run the file and open <http://127.0.0.1:5000> in the browser.

```
python -m pip install Flask
python -m flask --version

from flask import Flask
app = Flask(__name__)

@app.route('/')
def home():
    return 'Hello, World! Flask is
    running on localhost.'

if __name__ == '__main__':
    app.run(debug=True)

python app.py
```

#### Important reminder

localhost means the app is running on your own machine. It is not deployment yet.

## Final checklist for students

A beginner should leave Lesson 1 with these setup wins completed.

<p><b>Python installed</b> The terminal shows a valid Python version.</p>	<p><b>PATH working</b> python or py is recognized from the terminal.</p>
<p><b>pip verified</b> python -m pip --version works correctly.</p>	<p><b>Environment active</b> (.venv) or an active Conda environment is visible.</p>
<p><b>Flask installed</b> Flask is installed in the project environment.</p>	<p><b>Hello World running</b> The browser opens localhost successfully.</p>

## Common mistakes to avoid

Installing Python but forgetting PATH.	Using pip from one Python and running another Python.
Installing Flask globally instead of inside the project environment.	Thinking localhost means the app is already deployed.

## What to read next

<p><b>Python on Windows</b> <a href="https://docs.python.org/3/using/windows.html">https://docs.python.org/3/using/windows.html</a></p>
<p><b>Packaging guide: pip and virtual environments</b> <a href="https://packaging.python.org/guides/installing-using-pip-and-virtual-environments/">https://packaging.python.org/guides/installing-using-pip-and-virtual-environments/</a></p>
<p><b>Flask installation</b> <a href="https://flask.palletsprojects.com/en/stable/installation/">https://flask.palletsprojects.com/en/stable/installation/</a></p>
<p><b>Flask quickstart</b> <a href="https://flask.palletsprojects.com/en/stable/quickstart/">https://flask.palletsprojects.com/en/stable/quickstart/</a></p>
<p><b>Conda installation docs</b> <a href="https://docs.conda.io/projects/conda/en/stable/user-guide/install/windows.html">https://docs.conda.io/projects/conda/en/stable/user-guide/install/windows.html</a></p>

### SAAI lesson note template

This branded template is designed for student-facing lesson notes: clear title page, track-based sections, command boxes, checklists, and concise next-step resources.