

LESSON 53

Flask Framework Introduction

40 minutes

Python Web Backend

Outcome 1.22

Outcome 1.23

Outcome 2.6

Outcome 2.7

Outcome 2.15

Outcome 2.20

Outcome 3.3

WHAT THIS LESSON DOES

Get started with the Flask framework in this hands-on lesson. You'll set up your development environment in VS Code, install Flask, and build a simple web application with basic routes to understand how Flask works.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
1.22	read, write, test, and modify computer programs
1.23	reflect and communicate on the design and development process
2.6	construct algorithms using appropriate sequences, selections/conditionals, loops and operators to solve a range of problems, to fulfil a specific requirement
2.7	implement algorithms using a programming language to solve a range of problems
2.15	explain what is meant by the World Wide Web (WWW) and the Internet, including the client server model, hardware components and communication protocols
2.20	identify and fix/debug warnings and errors in computer code and modify as required
3.3	use appropriate programming languages to develop an interactive website that can display information from a database that meets a set of users' needs

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the basics of the Flask framework and its role in web development.
- Set up a Python development environment with VS Code and necessary tools.
- Install Flask and create a virtual environment for project isolation.
- Build a simple web application with basic and dynamic routes.
- Run and test a Flask application using a development server.

BEFORE THE LESSON

- On the room's machines, confirm Python is installed and on PATH (python --version) and that VS Code has the Microsoft Python extension. Do this before class: a fresh install behind the school filter eats the whole period.
- Test pip install flask inside a virtual environment on one machine to confirm the filter does not block the package download.
- Have a known-good app.py saved somewhere students can reach in case anyone's file will not run.
- Check that the loopback address 127.0.0.1:5000 is reachable in the room's browser and not blocked.

Flask Framework Introduction

HOW THE LESSON RUNS

Introduction
Installing Python on Your Computer
Setting Up VS Code for Python
Create a Project Folder
Create the Environment
Create a New Terminal
Installing Flask
Creating Your First Flask App
Adding a Basic Route
A Route with an Argument
Wrap Up

TEACHING MOVES

- 5. Create the Environment. Explain why the isolated environment matters before they type anything: it keeps this project's Flask separate from other projects and the system Python. Show the .venv folder appearing so the idea is concrete, not abstract.
- 7. Installing Flask. Make the activation step stick. The prompt must show (.venv) before pip install flask. If it does not, Flask installs system-wide and later projects break. Have them read their own prompt back to you.
- 8. Creating Your First Flask App. Walk the three lines line by line: import, create the app instance, decorate a function with @app.route('/'). Stress that the decorator maps a URL to the function directly below it.
- 9. Adding a Basic Route. Insist on the stop-and-restart cycle: Ctrl+C, then python -m flask run. The running server holds the old code. A student swearing their edit did nothing has almost always not restarted.
- 10. A Route with an Argument. Point at the two names: <name> in the URL and name in the function signature. They must match. Have them try /hello/ their own name in the browser and watch it appear in the response.

WHAT TO WATCH FOR

- Students edit app.py, refresh the browser and see no change, concluding their code is wrong. The development server runs the code loaded when it started. Teach the reflex: save, stop with Ctrl+C, run again. Make restarting automatic before they touch the browser.
- Confusing what runs on the server with what shows in the browser: they think the returned string is a web page. Show that the function returns plain text and Flask sends it as the response. The browser is a client asking the server; the server decides what to send back.
- Running pip install flask without the virtual environment active, so the install lands in system Python. Before installing, have every student confirm (.venv) appears at the start of their terminal prompt. No prefix, no install.

DIFFERENTIATION

- Emerging: Pair a student whose environment is fighting them with one who is running, so setup errors get a second pair of eyes. Give the exact activation command for their operating system written up on the board so they are not guessing at slashes.
- Developing: Ask them to predict what `/hello/Aoife` will display before they type the URL, then check it, so they reason about the route rather than copying it.
- Proficient: Have them add a route that takes two arguments, or returns a small HTML snippet instead of plain text, and explain what changed. Ask them to describe, as project evidence, why a virtual environment is good practice and what problem it prevents.