Building a website with Python

# Glossary

* **Python:** Python is a programming language. We will be using a version of the language called Python3 (not 2!).
* **Web Server:** A web server refers to a program that can accept network requests and respond with data. This data could be web pages (HTML), assets (JavaScript, CSS) or really anything (raw video data, JSON objects, etc)
* **Flask:** Flask is a web server that is built with Python. We will be writing Python code that creates the flask application.

# Getting Started – Development Environment

Pre-requesites for this tutorial are to have Python3 and Visual Studio Code installed on the development machine (school computer/laptop).

## Installing Python3

1. Go to the Python.org websites: https://www.python.org/
2. Click on **Downloads**
3. Double-click on the downloaded file to run the installerA screenshot of a computer

   Description automatically generated
4. **Important!** Check the box next to “Add python.exe to PATH”
5. Select **Install Now**
6. You should see the following success messageA screenshot of a computer

   Description automatically generated

You can confirm that Python3 is installed by opening the Windows Start menu and typing Python. You should see Python as a program.

A screenshot of a computer

Description automatically generated

If you open this program, you will start the Python shell. This is an environment that you can directly write Python code in to. Try writing:

print("Hello, world")

You should see Hello, world printed in the shell!

A computer screen shot of a black screen

Description automatically generated

## Installing Visual Studio Code

1. Visit the Code.VisualStudio.com website: <https://code.visualstudio.com/>
2. Click on **Download for Windows**
3. Double-click on the downloaded file and run the installer

## Install the Python extension for VS Code

1. Click on **View** from the top and select **Extensions**
2. A screenshot of a computer

   Description automatically generatedThe sidebar on the left will show installed extensions. Search for **Python** and click on install

## Create a workspace (Don’t Skip!)

Code can easily get cluttered! Create a new empty folder to store the files for this project. Then open this folder in VS Code.

Create a new file (**Ctrl + N**) and name it hello.py. The .py extension is the Python language extension and tells the operating system and other programs that this file contains Python code. Add a simple Python print() statement with some content.

Perhaps: print("when is this over?\n(╯°□°）╯︵ ┻━┻")

You can now Run this Python script. Do so by clicking on the ▷ (triangle) in the top right.

A screenshot of a computer

Description automatically generated

This will open the **Terminal** view, which is also known as a command prompt. The Terminal is a mechanism for humans to issue commands to the computer through text.

A screenshot of a computer

Description automatically generated

## Quick Overview of the Terminal 🡪 How to run Python

The terminal is an important part of development. It’s a common tool used by programmers for running commands that start processes, compile code, manipulate files, and so much more. Unfortunately, the terminal that is part of the Windows operating system is not very fun to use (Mr. Doran’s opinion), but that’s what we’re stuck with today!

Consider the text that is in the terminal at the end of the previous section:

PS **C:\Users\dan\git\flask-demo>** & **C:/Users/dan/AppData/Local/Programs/Python/Python313/python.exe** **c:/Users/dan/git/flask-demo/hello.py**

when is this over?

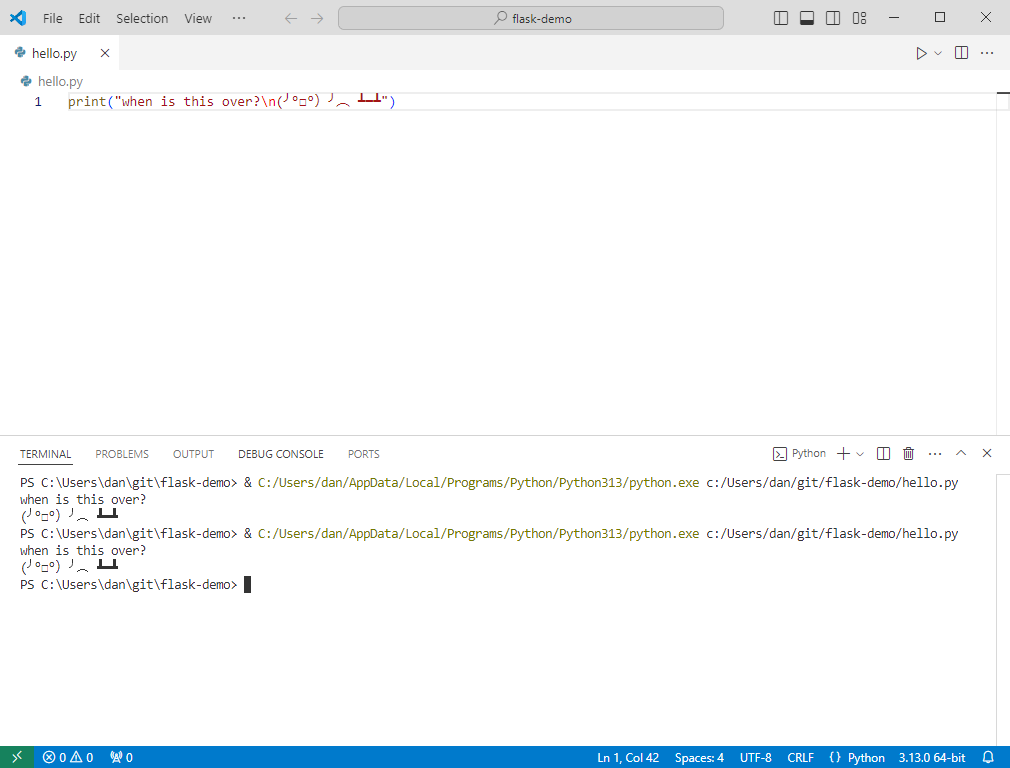
(╯°□°）╯︵ ┻━┻

PS C:\Users\dan\git\flask-demo>

|  |  |  |
| --- | --- | --- |
| Section | Text | Description |
| Location | **C:\Users\dan\git\flask-demo>** | At the very beginning of the Terminal is a path to the current folder. In this example, I’m using the Terminal inside of the “flask-demo” folder (which is inside “git”, which is inside “dan”, etc).  The final character > is used to indicate that this is the start of the user’s input. In other words, start typing and you’ll see the characters show up after >. |
| Command | **C:/Users/dan/AppData/Local/Programs/ Python/Python313/python.exe** | The first thing a user enters into the Terminal is a command. In this case, we’re entering the path to the Python program. If we enter, the computer will execute the Python program. |
| Arguments | **c:/Users/dan/git/flask-demo/hello.py** | Programs like Python are able to accept input from the command line. This is called “arguments”.  In this example, we’re passing the path (or the location) of the Python file that we created. This lets the Python program know that this is the code we want to execute. |
| Output | when is this over?  (╯°□°）╯︵ ┻━┻ | We executed a Python program that printed some text. This text is then shown back into the Terminal. |
| Repeat | PS C:\Users\dan\git\flask-demo> | The final line in the terminal output is the “Location” again. This is because once you execute a command in the Terminal, you can then execute another command. |

It’s now time to execute another command from the Terminal. This time, since we’ve already got the Terminal open, we don’t need to click on the ▷ triangle.

We’re going to use a Terminal trick: Press the ⇧ (up arrow) on your keyboard. The previous command that was executed is now visible. Hit the **Enter** key and see that we’ve executed the Python code again:



Press the ⇧ (up arrow) again, but this time delete the argument to our Python code. The Terminal should now just contain the path to the Python command. Hit **Enter** and see what happens:

A screenshot of a computer

Description automatically generated

It’s the Python shell! When you execute Python without providing an argument, it simply starts the Python shell. The shell is a great place to quickly test out Python code that isn’t too long.

## Final Installation Step – Install Flask

Flask is a library that we can install as a Python “package”. To install Python packages, we’re going to execute Python from the Terminal the way we did in the previous section.

This time, we’re going to use a special tool call pip. Enter the following arguments ***after*** the Python command:

-m pip install flask

And you should see the Flask library installed:

PS **C:\Users\dan\git\flask-demo>** & C:/Users/dan/AppData/Local/Programs/Python/Python313/python.exe **-m pip install flask**

Collecting flask

Using cached flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)

Collecting Werkzeug>=3.0.0 (from flask)

Using cached werkzeug-3.0.4-py3-none-any.whl.metadata (3.7 kB)

Collecting Jinja2>=3.1.2 (from flask)

Using cached jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)

Collecting itsdangerous>=2.1.2 (from flask)

Using cached itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)

Collecting click>=8.1.3 (from flask)

Using cached click-8.1.7-py3-none-any.whl.metadata (3.0 kB)

Collecting blinker>=1.6.2 (from flask)

Using cached blinker-1.8.2-py3-none-any.whl.metadata (1.6 kB)

Collecting colorama (from click>=8.1.3->flask)

Using cached colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)

Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask)

Using cached MarkupSafe-3.0.1-cp313-cp313-win\_amd64.whl.metadata (4.1 kB)

Using cached flask-3.0.3-py3-none-any.whl (101 kB)

Using cached blinker-1.8.2-py3-none-any.whl (9.5 kB)

Using cached click-8.1.7-py3-none-any.whl (97 kB)

Using cached itsdangerous-2.2.0-py3-none-any.whl (16 kB)

Using cached jinja2-3.1.4-py3-none-any.whl (133 kB)

Using cached werkzeug-3.0.4-py3-none-any.whl (227 kB)

Using cached MarkupSafe-3.0.1-cp313-cp313-win\_amd64.whl (15 kB)

Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)

Installing collected packages: MarkupSafe, itsdangerous, colorama, blinker, Werkzeug, Jinja2, click, flask

WARNING: The script flask.exe is installed in 'C:\Users\dan\AppData\Local\Programs\Python\Python313\Scripts' which is not on PATH.

Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.

Successfully installed Jinja2-3.1.4 MarkupSafe-3.0.1 Werkzeug-3.0.4 blinker-1.8.2 click-8.1.7 colorama-0.4.6 flask-3.0.3 itsdangerous-2.2.0

PS C:\Users\dan\git\flask-demo>

And that’s it! Flask is now installed.

# Next Up

Using flask: https://flask.palletsprojects.com/en/3.0.x/quickstart/