

Matt Coding Tutor

PYTHON
LESSON SHEET

A Moonlight Pixel Learning Publication

Generated from your question and organized for review.

DATE

August 26, 2026

LESSON TOPIC

Flask Website

Student Question

Matt, build a beginner-friendly Flask website with a home page and one additional page. Give me the complete files first, explain the folder structure, and show me how to run it.

Absolutely. Here is a beginner-friendly Flask website with:

- a home page
- one additional page
- shared layout
- simple CSS styling

Complete Program

Create a folder named `my_flask_site`.

Inside it, create these files and folders.

`app.py`

```
from flask import Flask, render_template

# Create the Flask application
app = Flask(__name__)

# Home page route
@app.route("/")
def home():
    return render_template("index.html")

# About page route
@app.route("/about")
def about():
    return render_template("about.html")

# Run the app
if __name__ == "__main__":
    app.run(debug=True)
```

`templates/base.html`

```

<!DOCTYPE html>
<html>
<head>
    <title>{{ title }}</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>

    <nav>
        <a href="{{ url_for('home') }}">Home</a>
        <a href="{{ url_for('about') }}">About</a>
    </nav>

    <main>
        {% block content %}
        {% endblock %}
    </main>

</body>
</html>

```

templates/index.html

```

{% extends "base.html" %}

{% block content %}
    <h1>Welcome to My Flask Website</h1>
    <p>This is the home page.</p>
{% endblock %}

```

templates/about.html

```

{% extends "base.html" %}

```

```
{% block content %}
    <h1>About This Website</h1>
    <p>This is a simple Flask website built for beginners.</p>
{% endblock %}
```

static/style.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f4f6f8;
    margin: 0;
    color: #222;
}

nav {
    background-color: #2c3e50;
    padding: 15px;
}

nav a {
    color: white;
    text-decoration: none;
    margin-right: 20px;
    font-weight: bold;
}

nav a:hover {
    text-decoration: underline;
}

main {
    padding: 30px;
}

h1 {
```

```
color: #2c3e50;  
}
```

Folder Structure

Your project should look like this:

```
my_flask_site/  
|  
├─ app.py  
|  
├─ templates/  
|   ├─ base.html  
|   ├─ index.html  
|   └─ about.html  
|  
└─ static/  
    └─ style.css
```

Here is what each part does.

`app.py` is the main Python file.

It creates the Flask app.

It also connects website addresses, like `/` and `/about`, to HTML pages.

The `templates` folder stores your HTML files.

Flask looks in this folder automatically when you use `render_template()`.

The `static` folder stores files that do not change while the app runs.

That usually means CSS, images, and JavaScript.

How to Run It

First, open your terminal.

Move into your project folder:

```
cd my_flask_site
```

If Flask is not installed yet, install it:

```
pip install flask
```

Then run the website:

```
python app.py
```

On some computers, especially macOS, you may need:

```
python3 app.py
```

When it runs, you should see something like this:

```
Running on http://127.0.0.1:5000
```

Open this address in your browser:

```
http://127.0.0.1:5000
```

To visit the second page, go to:

`http://127.0.0.1:5000/about`

Understand the Program

Flask App

`app.py` creates the web application.

The two routes tell Flask what to show for each page.

The home route uses `/`.

The about route uses `/about`.

So when someone visits `/about`, Flask runs the `about()` function and shows `about.html`.

Templates

The HTML files in the `templates` folder create the pages.

`base.html` is the shared layout.

That means both pages use the same navigation bar and page structure.

`index.html` and `about.html` fill in the content area.

This keeps your website organized.

Static CSS

The CSS file controls how the site looks.

It changes the background color, text style, navigation bar, spacing, and heading color.

What Can You Customize?

You can change:

- the page titles
- the text inside `index.html`
- the text inside `about.html`
- the navigation link names
- the colors in `style.css`
- the font in `style.css`

For example, changing the background color in `style.css` will update the whole site's look.

Expected Result

When you open the website, you should see a simple home page with a dark navigation bar.

The navigation bar should have two links:

- Home
- About

Clicking About should take you to the second page.

Practice Challenge

Add one more page called `Contact` .

Try creating a new route, a new template file, and a new navigation link.

Keep building your skills with another lesson from the library.

My Notes

Write one thing that helped this lesson make sense to you.

Lesson Complete

- ☐ I understand this topic.
- ☐ I need to review this lesson again.
- ☐ I completed the Practice Challenge.

Matt's Reminder

Learning Python is not about memorizing every command. It is about understanding how to solve problems one step at a time. Every lesson you complete makes the next lesson easier.

Matt Coding Tutor

A Moonlight Pixel Project

www.moonlightpixel.com/matt

© 2026 Moonlight Pixel. All rights reserved.