In Flask, static files refer to assets like CSS stylesheets, JavaScript files, images, and other resources that don't typically change during the runtime of a web application. These files are considered static because their content remains constant, and they are served directly to clients without any processing by the server.

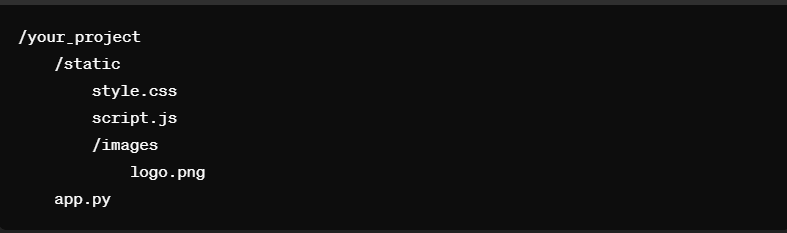
In the context of Flask, static files are typically stored in a folder named "static" within the project directory. Flask automatically recognizes this folder as the location for static files, making it easy to organize and serve them.

Here's a brief explanation of how static files work in Flask:

**Folder Structure:**

* Create a folder named "static" in your Flask project directory.
* Organize static files (e.g., CSS, JavaScript, images) within this folder.

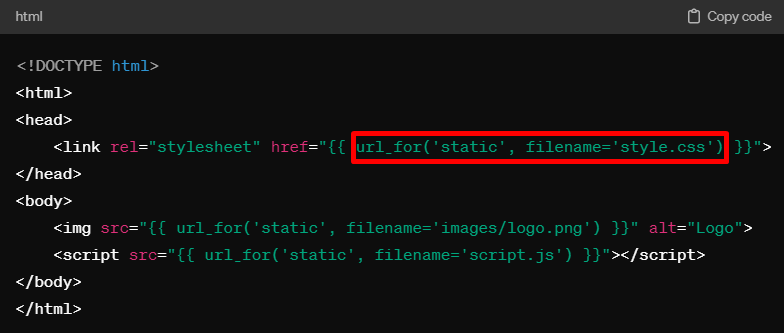
Example folder structure:



**Serving Static Files:**

* Flask automatically serves static files from the "static" folder without the need for additional route definitions.
* **You can reference these static files in your HTML templates using the url\_for function.**

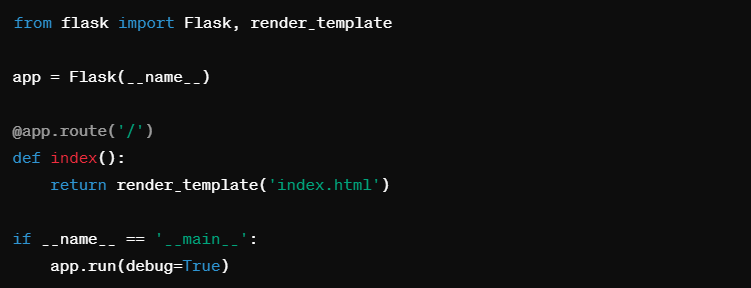
Example in a template (index.html):



In this example, **url\_for('static', filename='style.css') generates the correct URL for the style.css file in the "static" folder.**

**Configuration:**

Flask automatically configures the static file handling, and there's usually no need for explicit configuration. The "static" folder is recognized by default.



**Accessing Static Files:**

When you run your Flask application, static files will be accessible at URLs like /static/style.css or /static/images/logo.png, based on the structure of your "static" folder.

In summary, static files in Flask are resources that remain constant and are served directly to clients without processing by the server. The "static" folder is a convention for organizing these files, and Flask handles their serving automatically.