

# Building website using python and flask

Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier. It gives developers flexibility and is a more accessible framework for new developers since you can build a web application quickly using only a single Python file. Flask is also extensible and doesn't force a particular directory structure or require complicated boilerplate code before getting started.

## Prerequisites

Before you start following this guide, you will need:

- A local Python 3 programming environment
- An understanding of Python 3 concepts, such as data types, conditional statements, for loops, functions, and other such concepts.

## Installing Flask

In this step, you'll activate your Python environment and install Flask using the `pip` package installer.

- `pip install flask`

now ,we are ready to begin our code ,lets Insert the following lines of Python code inside an empty text file,

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "Hey there!"

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

Once you run the script, the website should now be up and running on your local machine and it can be viewed by visiting localhost:5000 in your browser.

So we create an instance of the *Flask* class for our web app and Python assigns the name "\_\_main\_\_" to the script when the script is executed. If the script is imported from another script, the script keeps its given name (e.g. hello.py). In our case we are executing the script. Therefore, \_\_name\_\_ will be equal to "\_\_main\_\_". That means the if conditional statement is satisfied and the *app.run()* method will be executed.

To make something more serious and more visually appealing, you would want to incorporate HTML files along with your Python file and have your Python code return HTML pages instead of plain strings. So create an empty file, name it something like home.html and put the following HTML code inside it:

```
<!DOCTYPE html>
<html>
<body>
<h1>My Personal Website</h1>
<p>Hi, this is my personal website.</p>
</body>
</html>
```

The flask framework has been written in a way so that it looks for HTML template files in a folder that should be named templates. So, you should create such an empty folder and then put all the HTML templates in there.

Now lets add about page to our website so we will modify it as following,

```
<!DOCTYPE html>
<html>
<body>
<h1>About me</h1>
<p>This is a portfolio site about anything that can be put in a portfolio.</p>
</body>
</html>
```

you need to render the HTML with Python by adding a second function to the hello.py Python script.

```
@app.route('/about/')
def about():
    return render_template('about.html')
```

### Adding a navigation menu to our website

To create that area in each page, we could add the HTML code that generates such a header to each of our HTML files. However, a smarter thing to do would be to create a parent HTML template which our two child templates could inherit the code from. Then we could simply link to that parent layout from each HTML page to inherit its code.

```

<!DOCTYPE html>
<html>
  <body>
    <header>
      <div class="container">
        <h1 class="logo">Ardit's web app</h1>
        <strong><nav>
          <ul class="menu">
            <li><a href="{{ url_for('home') }}">Home</a></li>
            <li><a href="{{ url_for('about') }}">About</a></li>
          </ul>
        </nav></strong>
      </div>
    </header>
    <div class="container">
      {% block content %}
      {% endblock %}
    </div>
  </body>
</html>

```

The two lines with curly brackets will be replaced on demand by either the home.html, or the about.html code depending on which URL the user is browsing. For that to happen, you also need to edit those two pages so that they connect to the layout.html page.

```

{% extends "layout.html" %}
{% block content %}
  <div class="home">
    <h1>A Python product</h1>
    <p>This website was built with Python via the Flask framework.</p>
  </div>
{% endblock %}

```

Same goes for about page

```

{% extends "layout.html" %}
{% block content %}
  <div class="about">
    <h1>About me</h1>
    <p>This is a portfolio site about anything that can be put in a portfolio.</p>
  </div>
{% endblock %}

```

Now we need to add some style formatting to the HTML structure using CSS (Cascading Style Sheets). That is done by creating a CSS file and connecting it to our HTML files. CSS is a style language that likewise HTML it is also very easy to learn

create an empty file inside the css and name the file something like main.css

referring to the HTML tags and div class names and applying fonts, colors, text size, padding, margins and background colors to those HTML elements.

We can link our css file to our code by the following code

```
<head>
  <title>Flask app</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='css/main.css') }}">
</head>
```

It is not advisable to even start creating a web app using your main Python installation. Because you use Python for many other things, there can be a lot going in that installation hence we will create a virtual environment using venv library The virtual environment files will be generated inside a folder that is at the same directory level with the folder where your app files are. It's a good idea to create another folder (e.g. *myblog*) and put the *app* folder inside it. Then, open a terminal/command line while you are in the *myblog* folder and type this:

```
python -m venv virtual
```

if we get an error that mean we have to install the flask in our virtual library or to avoid that we should create the virtual library first then we can code on .

Let's now run our website using our virtual environment.

```
virtual\Scripts\python app\hello.py
```

so our website will look as follows:

