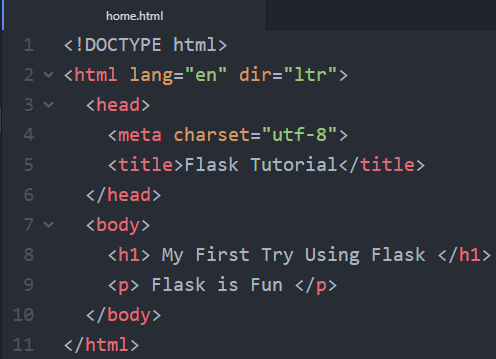
**An Introduction to Web Applications and Deploying Them to the Cloud  
Part 2: HTML, CSS, Virtual Environments**

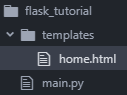
By Salvador Villalon

**HTML and Templates in Flask**

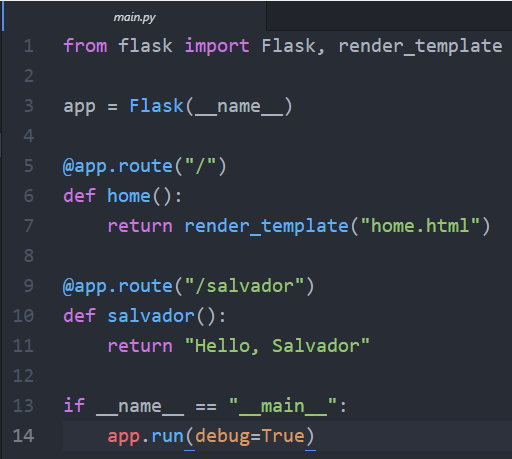
* First create a new HTML file. I called mine **home.html**
* Here is some code to get you started

****

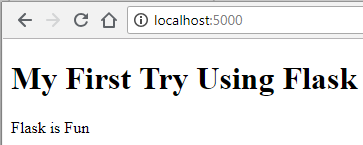
* **Important Point To Remember** 
  + The Flask Framework looks for HTML files in a folder called **templates**
  + You **need to create a templates** folder and put all your html files in there

****

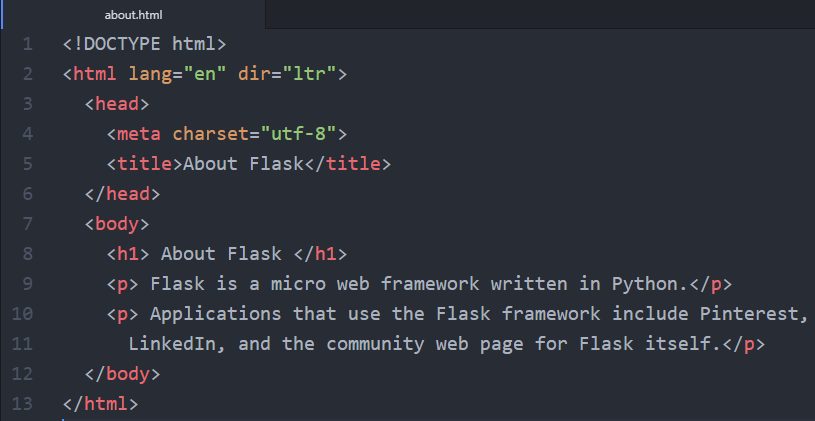
* + Remember to always keep the **main.py outside of your templates folder**
* **Now Bring it Back to main.py**
  + Now we need to change our main.py so that we can view the html file we created



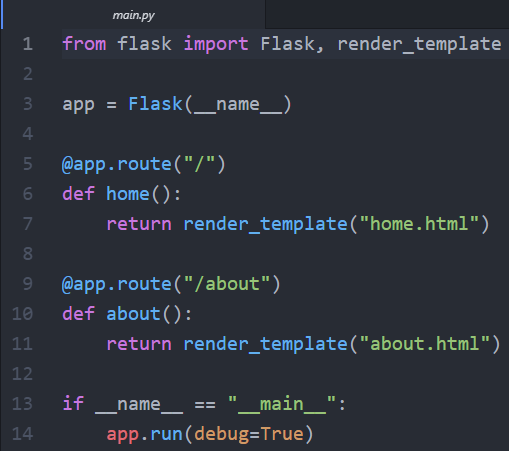
* + We made two new changes
    - **Line 1:** We imported render\_template method from the flask framework. render\_template looks for a template in the templates folder and it will render the template for which you ask
      * **More information here:** [**http://flask.pocoo.org/docs/0.12/quickstart/#rendering-templates**](http://flask.pocoo.org/docs/0.12/quickstart/#rendering-templates)
    - **Line 7:** We change the return so that now it returns render\_template(“home.html”). This will let us view our html file
  + **Now visit your localhost and see the changes: <http://localhost:5000/>**

****

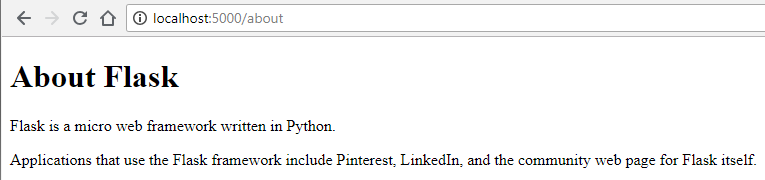
* **Let’s add more pages**
  + Let’s create an **about.html** inside the **templates folder**

****

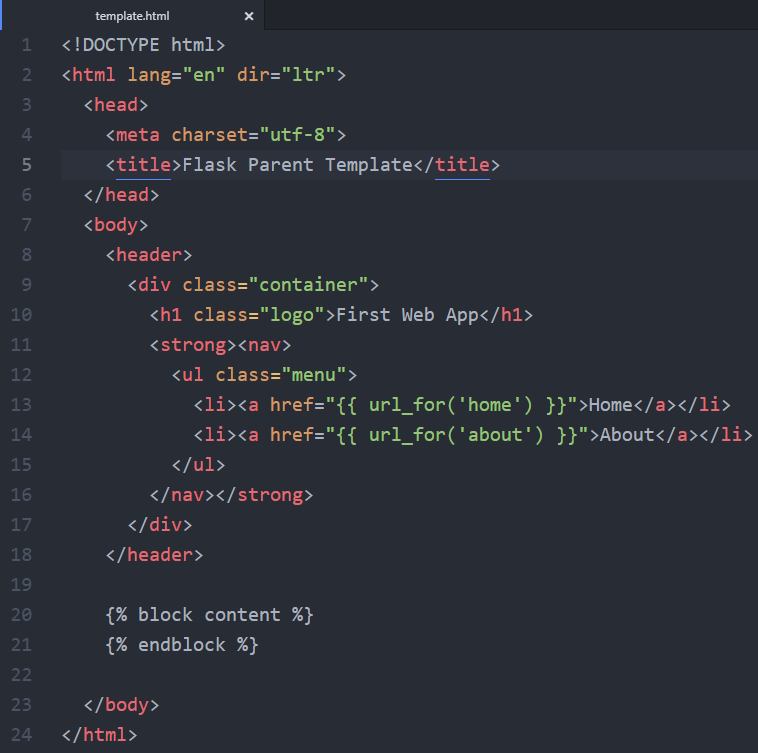
* **Now Bring it Back to main.py**
  + Let’s make a similar change from before to our main.py



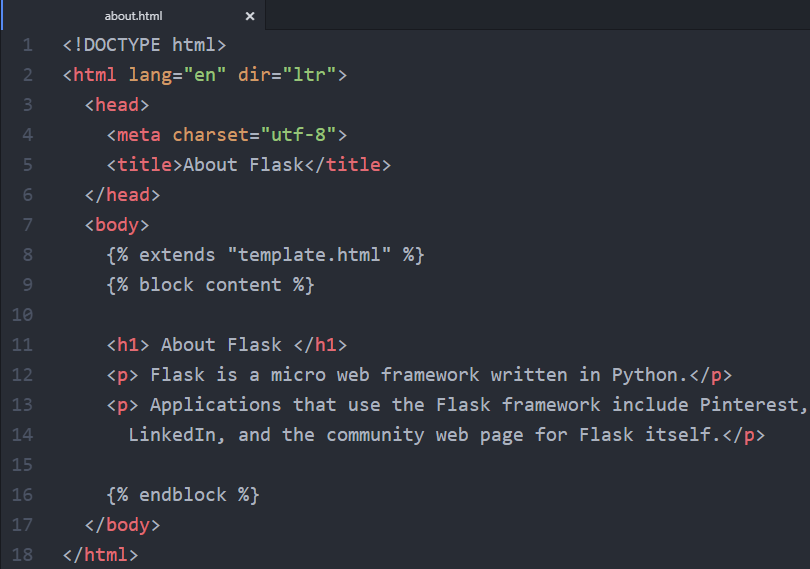
* + We made two new changes
    - **Line 9:** Change the route to “**/about”**
    - **Line 11:** Change the return so that now it returns render\_template(“about.html”)
  + **Now visit and see the changes:** [**http://localhost:5000/about**](http://localhost:5000/about)

****

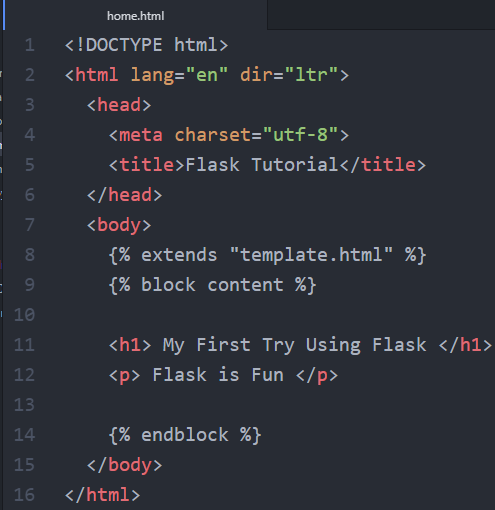
* **Let’s Connect Both Pages with a Navigation**
  + To connect both pages we can have a navigation menu on the top. We can use Flask to make the process of creating a navigation menu easier
  + First, let’s create a **layout.html.** This **layout.html** will serve as a parent template from which our two child templates can inherit code from



* + **What is going on here?**
    - **Line 13 – 14:** We use the function called **url\_for().** It accepts the name of the **function as the first argument** and a number of keyword arguments, each corresponding to the variable part of the URL rule. **Right now we just gave it the name of the function.**
      * **More info:** [**http://flask.pocoo.org/docs/0.12/quickstart/#url-building**](http://flask.pocoo.org/docs/0.12/quickstart/%23url-building)
    - The two lines with the curly brackets will be **replaced by the content of home.html and about.html** depending on the URL in which the user is browsing.
  + With these changes it allows the child pages (home.html and about.html) to connect to the parent (template.html). This allows us to not have to **copy the code for the navigation menu in the about.html and home.html**
  + **about.html**



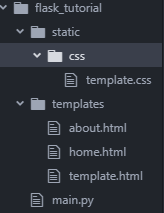
* + - **home.html**

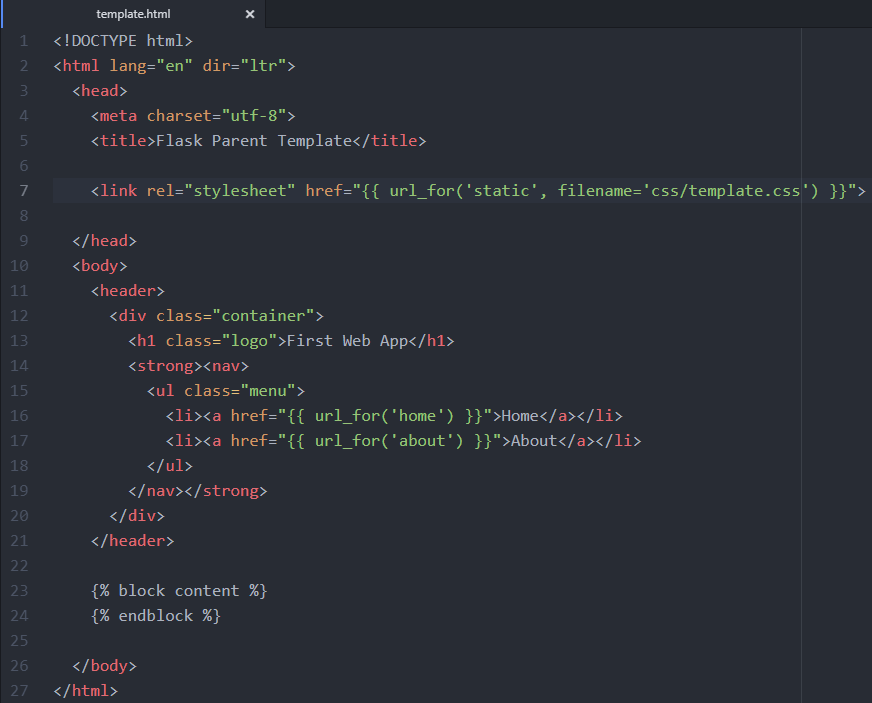
****

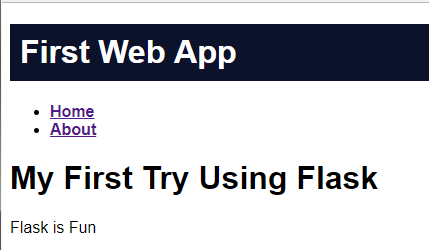
* Our website is great, but now let’s try adding some CSS

**Adding CSS to Our Website**

* + **An important note to remember**
    - Just like we did by creating a folder called **templates** to store all of our HTML templates, we need a folder called **static** to store our stylesheets
    - In **static**, we will store our CSS, JavaScript, images, and other necessary files
    - That is why it is important that you should create a **css folder to store your CSS stylesheets**
  + After you do this your project folder it should look like this:

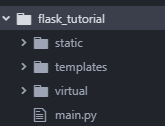


* + **Linking our CSS with our HTML file**
    - Since our template.html is the one that links all pages, we can insert the code here and it will be applied on the child pages   
        
      
    - **Line 7:** Here we are giving the path to where the template.css is located
    - **Now visit and see the changes:** [**http://localhost:5000/about**](http://localhost:5000/about)

****

**Moving Forward with Flask and virtualenv**

* + Now that you are more familiar with how to use Flask. You may start using it in your future projects. One thing to always do is use virtualenv
  + **Why use virtualenv?**
    - You may use Python for others projects besides web-development
    - The more projects you work on, the more it might mean that you will have different version of Python installed, different dependencies or different versions of libraries installed
    - We use virtualenv to create an isolated environment for your Python project. This means that each project can have its own dependencies regardless of what dependencies every other project has
  + **Start with virtualenv**
    - First, do **pip install virtualenv** on your Command prompt or Terminal
    - Second, do **virtualenv “name of virtual environment”**
      * Here you can give a name to the environment
      * I usually give it a name of virtual: **virtualenv virtual**
    - When you finished setting up the virtual environment, check your project folder and it should look like this. **The virtual environment should be created in the same directory where your app files are located**



* + **Activate the Virtual Environment**
    - For OS X and Linux Environment
      * **$ . name of virtual environmnet/bin/activate**
    - For Windows Environment
      * **$ name of virtual environment\Scripts\activate**
    - Since I am using Windows Machine, when I activate the environment it will look like this

****

* + - Next step will be to install flask on your virtual environment so that we can run the application inside our environment. Do **pip install flask**
  + **For more information about flask and virtualenv:** 
    - <http://flask.pocoo.org/docs/0.12/installation/#installation>
    - <https://realpython.com/python-virtual-environments-a-primer/>

**On the next part of the tutorial**

* We will deploy our web application so that others can see it. Check out **Part 3:** **Deploy Your Web Application to the Cloud**

**Resources**

* <https://pythonhow.com/building-a-website-with-python-flask/>
* [https://cloud.google.com/appengine/docs/standard/python/getting-started/python-standard-env](https://cloud.google.com/appengine/docs/standard/python/getting-started/python-standard-env#test_the_application)
* <https://youtu.be/j5wysXqaIV8>
* <https://realpython.com/python-virtual-environments-a-primer/>
* <http://flask.pocoo.org/docs/0.12/installation/#installation>
* <https://www.techopedia.com/definition/31267/google-app-engine-gae>
* <https://cloud.google.com/appengine/>
* <https://searchitchannel.techtarget.com/definition/cloud-services>
* <https://www.techopedia.com/definition/31267/google-app-engine-gae>