**An Introduction to Web Applications Deploying Them to the Cloud  
Part 1: Introduction to HTTP and Flask**

By Salvador Villalon

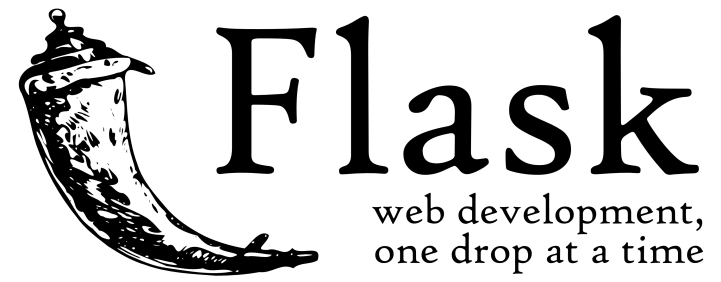
**Introduction**

* This document contains a series of several sections, each of which explains a particular aspect of Python and Flask. In each section, I will show pieces of code for you to follow along. All the code used in the tutorial is available in the [GitHub Repository](https://github.cicd.spglobal.com/salvador-villalon/flask-tutorial).

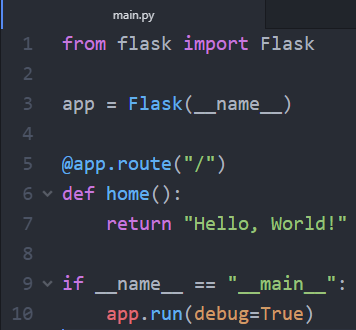
**What is HTTP and What does it have to do with Flask?**

* [**HTTP stands for Hyper Text Transfer Protocol**](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)
  + This is the protocol for websites and what the internet uses to interact and communicate with computers and servers
  + Think about it this way
    - When you are typing the name of a website in the address bar of your browser, and you hit enter, what that is really doing is it sending an HTTP Request to some server
    - For example, I go to my address bar and type google.com, then I hit enter
    - What will happen is that:
      * An HTTP Request is sent to Google Server
      * Google Server receives the request and needs to figure out some way to interpret that request
      * Google Server sends back an HTTP Response that contains the information that my web browser then receives
      * Then it displays what you asked for on a page in my browser
* **What does this have to do with Flask?**
  + We will write code that will take care of the server side processing
    - Our code will receive requests
    - It will figure out with what those requests are dealing and for what they are asking
    - It will figure out what response to send to the user
  + **To do all of this we will use Flask**

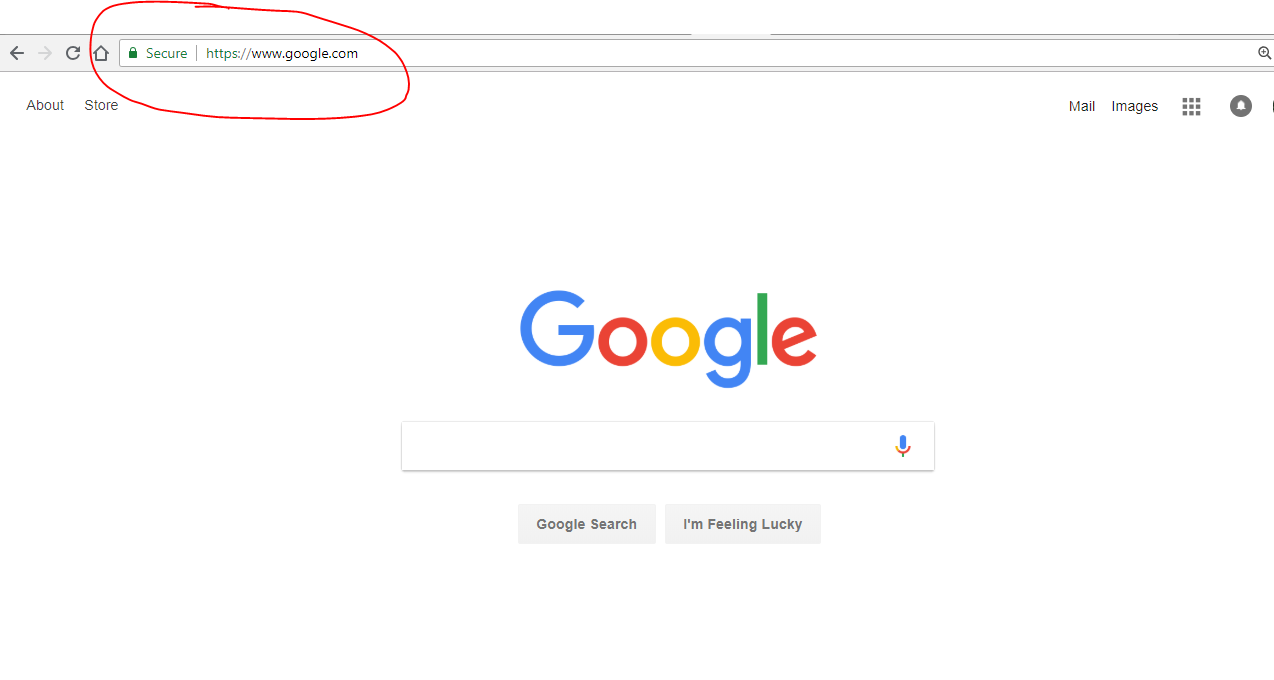
**What is Flask?**



* + [Flask is a micro framework](http://flask.pocoo.org/) written in Python
  + It makes the process of designing a Web Application more simple
  + **Flask lets us focus on what the users are requesting and what sort of response to give back**
* **How Does a Flask App Work?**
  + The Code that lets us run a basic web application that we can serve as if it were a website

******

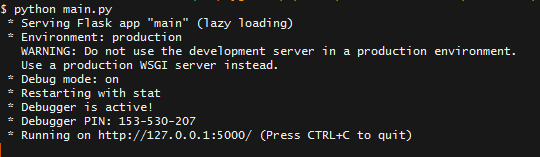
* + **Line 1:** Here we are importing the Flask module and creating a Flask web server from the Flask module
  + **Line 3:** 
    - **\_\_name\_\_ means this current file**. In this case, it will be main.py. This current file will represent my Web Application
    - We are creating an instance of the Flask class and calling it app. Here we are creating a new Web Application
  + **Line 5:**
    - **Point to Note:** Flask is designed in terms of routes where the route is part of the URL you type in order to determine which page you want to request
    - In line 5, it just represents the default page
      * For example, if I go to a website such as “google.com/” with nothing after the slash, then this will be the default page of Google

****

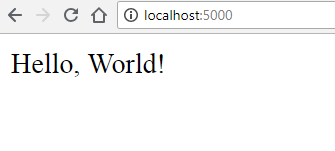
* + **Line 6 – 7:**
    - When the user goes to my website and they go to the default page (nothing after the slash), then the function below it will get activated
  + **Line 9:** 
    - When working with Python, it is important to know that when you run your Python script, Python assigns the name “\_\_main\_\_” to the script when it is executed
    - If we import another script, this **if statement will prevent other scripts from running** since when we run main.py, main.py will change its name to \_\_main\_\_ and only then will that if statement be activated
  + **Line 10:** 
    - This will run the application
    - Having debug=True allows possible Python errors to appear on the web page. This will help us trace the errors

**Let’s Try Running main.py**

* **In your Terminal or Command Prompt go to the folder that contains your main.py**
* Then do **py main.py** or **python main.py**
* In your terminal or Command Prompt you should see this output

****

* The important part is where it says “Running on <http://127.0.0.1:5000/>”
  + - 127.0.0.1 means this local computer. If you do not know the meaning of this (Just like I did when I started, [this article is really helpful](https://whatismyipaddress.com/localhost))
    - 127.0.0.1 is the same as localhost
  + Go to that address and you should see this



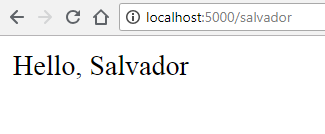
* + Congrats! You just made a simple Web Application.

**More Fun with Flask**

* Earlier you just saw what happened when we ran main.py with just one route which was app.route(“/”)  
  + Let’s add more routes so you can see the difference



* + In **lines 9 – 11**. we added a new route, this time to **/salvador**
  + Now run the main.py again and go to<http://localhost:5000/salvador>



* + **You can add as many routes as you want.**

**On the next part of the tutorial**

* So far we have just been returning text. Let’s make our website look nicer. Check out **Part 2: HTML and CSS with Flask** to learn more.

**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>