Python Flask Tutorial



Flask Tutorial provides the basic and advanced concepts of the Python Flask framework. Our Flask tutorial is designed for beginners and professionals.

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by **Armin Ronacher** who leads an international group of python enthusiasts (POCCO).

What is Flask?

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by **Armin Ronacher** who leads an international group of python enthusiasts (POCCO). It is based on WSGI toolkit and jinja2 template engine. Flask is considered as a micro framework.

## Flask Environment Setup

To install flask on the system, we need to have python 2.7 or higher installed on our system. However, we suggest using python 3 for the development in the flask.

### Install virtual environment (virtualenv)

virtualenv is considered as the virtual python environment builder which is used to create the multiple python virtual environment side by side. It can be installed by using the following command.

1. $ pip install virtualenv

Once it is installed, we can create the new virtual environment into a folder as given below.

1. $ mkdir new
2. $ cd new
3. $ virtualenv venv

To activate the corresponding environment, use the following command on the Linux operating system.

1. $ venv/bin/activate

On windows, use the following command.

1. $ venv\scripts\activate

We can now install the flask by using the following command.

1. $ pip install flask

However, we can install the flask using the above command without creating the virtual environment.

To test the flask installation, open python on the command line and type python to open the python shell. Try to import the package flask.

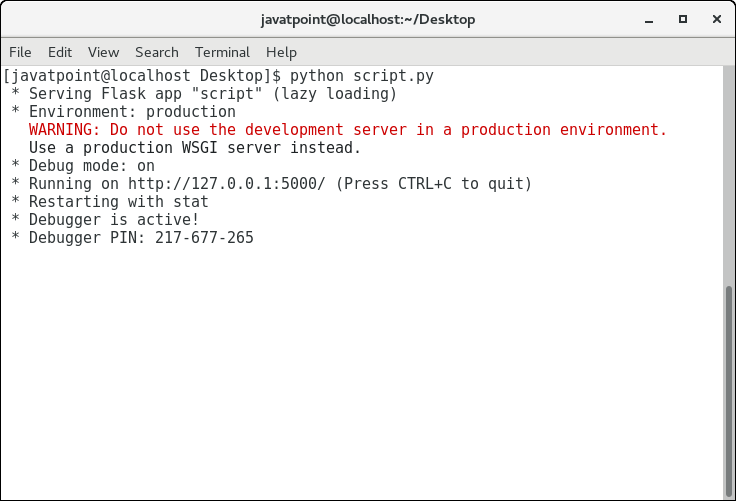
# Flask IntroductionFirst Flask application

In this section of the tutorial, we will build our first python website built using the Flask framework. In this process, open any text editor of your choice as we are using the sublime text editor in this tutorial.

Write the following lines of code and save to a file named as script.py.

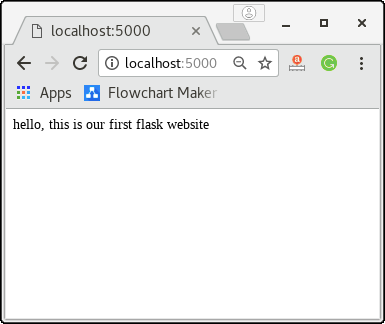
1. **from** flask **import** Flask
3. app = Flask(\_\_name\_\_) #creating the Flask class object
5. @app.route('/') #decorator drfines the
6. **def** home():
7. **return** "hello, this is our first flask website";
9. **if** \_\_name\_\_ =='\_\_main\_\_':
10. app.run(debug = True)

Let's run this python code on the command line and check the result.



Since it is a web application, therefore it is to be run to on the browser at http://localhost:5000.

Play Video[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)



To build the python web application, we need to import the Flask module. An object of the Flask class is considered as the WSGI application.

We need to pass the name of the current module, i.e. \_\_name\_\_ as the argument into the Flask constructor.

The route() function of the Flask class defines the URL mapping of the associated function. The syntax is given below.

1. app.route(rule, options)

It accepts the following parameters.

1. rule: It represents the URL binding with the function.
2. options: It represents the list of parameters to be associated with the rule object

As we can see here, the / URL is bound to the main function which is responsible for returning the server response. It can return a string to be printed on the browser's window or we can use the HTML template to return the HTML file as a response from the server.

Finally, the run method of the Flask class is used to run the flask application on the local development server.

The syntax is given below.

1. app.run(host, port, debug, options)

|  |  |  |
| --- | --- | --- |
| **SN** | **Option** | **Description** |
| 1 | host | The default hostname is 127.0.0.1, i.e. localhost. |
| 2 | port | The port number to which the server is listening to. The default port number is 5000. |
| 3 | debug | The default is false. It provides debug information if it is set to true. |
| 4 | options | It contains the information to be forwarded to the server. |

# Flask App routing

App routing is used to map the specific URL with the associated function that is intended to perform some task. It is used to access some particular page like [Flask Tutorial](https://www.javatpoint.com/flask-tutorial) in the web application.

In our first application, the URL ('/') is associated with the home function that returns a particular string displayed on the web page.

In other words, we can say that if we visit the particular URL mapped to some particular function, the output of that function is rendered on the browser's screen.

Consider the following example.

Play Video[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

### Example

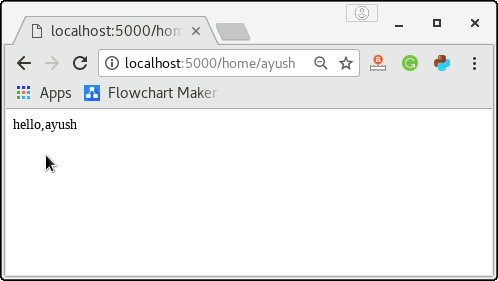
1. **from** flask **import** Flask
2. app = Flask(\_\_name\_\_)
4. @app.route('/home')
5. **def** home():
6. **return** "hello, welcome to our website";
8. **if** \_\_name\_\_ =="\_\_main\_\_":
9. app.run(debug = True)

Flask facilitates us to add the variable part to the URL by using the section. We can reuse the variable by adding that as a parameter into the view function. Consider the following example.

### Example

1. **from** flask **import** Flask
2. app = Flask(\_\_name\_\_)
4. @app.route('/home/<name>')
5. **def** home(name):
6. **return** "hello,"+name;
8. **if** \_\_name\_\_ =="\_\_main\_\_":
9. app.run(debug = True)

It will produce the following result on the web browser.

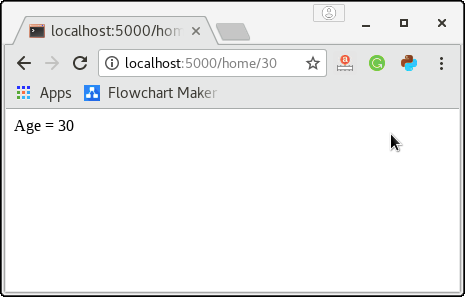


The converter can also be used in the URL to map the specified variable to the particular data type. For example, we can provide the integers or float like age or salary respectively.

Consider the following example.

### Example

1. **from** flask **import** Flask
2. app = Flask(\_\_name\_\_)
4. @app.route('/home/<int:age>')
5. **def** home(age):
6. **return** "Age = %d"%age;
8. **if** \_\_name\_\_ =="\_\_main\_\_":
9. app.run(debug = True)



The following converters are used to convert the default string type to the associated data type.

1. string: default
2. int: used to convert the string to the integer
3. float: used to convert the string to the float.
4. path: It can accept the slashes given in the URL.