Flask:

Installations:

1. pip install flask

Queries:

1. What is wisgi application?

Notes:

1. Flask is a microframework.

2. A Flask script is an app that handles input and output request.

3. When a request comes in, the app checks which route should be taken and runs the method specific to that route.

This method is known as a controller or a view.

4. The 'view' responds back with an html, json, xml, etc data.

5. Request object of the Flask is used to get the request (get and post) data sent by the user.

6. The args attribute of the request object holds all the agruments of the request.

7. Flask ‘render\_template’ object is used to return a html page as a response.

7. Flask always looks in to the ‘templates’ folder to look for html templates to be returned as response.

8. Inoder to avoid repeated html typing, we can use template inheritence that use reusable html code called ‘block statements’

Syntax:

{ % block <type> %} {% endblock %}

Where <type> can be ‘title’ and ‘content’ or ‘scripts’ for adding java scripts.

9.css and javascripts files to be used by the template can be added in ‘scripts’ directory.

FLASK REST APIs:

* Web APIs are a way for other applications (mobile and web) and even our own websites to use our data in a safe secure and sanatized manner. So they don’t have to scrape data out of our application.
* To create Web APIs in Python Flask, we can use a library know as Flask Restful.
* Flask Restful helps us create classes (similar to models) that lets us create our resources and allow http methods onto these resources.

Reqpars: To handle the request data

* Our resources should be able to accept input form the users to return some data.
* These inputs need to be in specific format (integer field, string field, etc) and should contain certain specific entries (like title, name, etc)
* Reqparse library of flask is used for parsing request.

Marshal : To handle the response data

* You can define a dict or OrderedDict of fields whose keys are names of attributes or keys on the object to render, and whose values are a class that will format & return the value for that field.
* Usage: marshal(<data> , <field dict>)