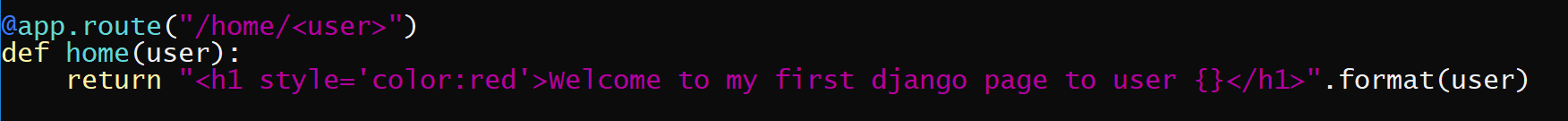
Flask

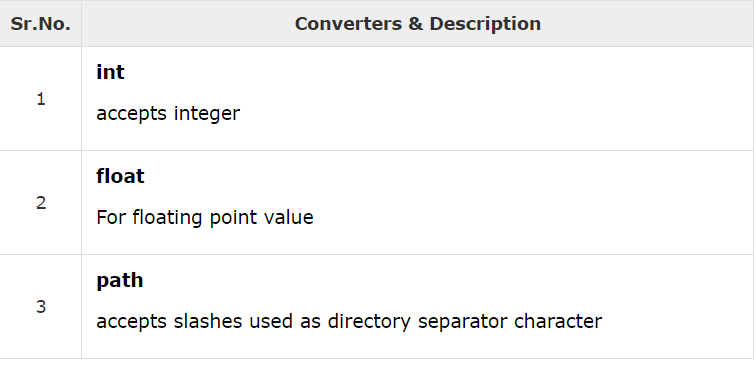
* It is possible to build a URL dynamically, by adding variable parts to the rule parameter. This variable part is marked as **<variable-name>**. It is passed as a keyword argument to the function with which the rule is associated.

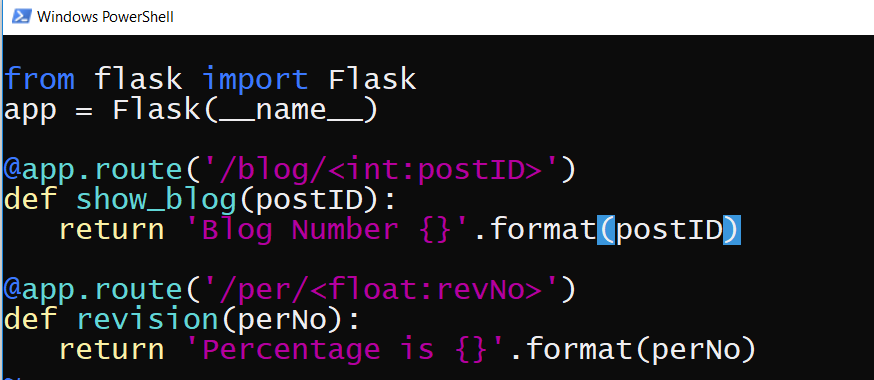
In the following example, the rule parameter of **route()** decorator contains **<name>** variable part attached to URL **‘/hello’**.



if the **localhost/home/sachin** is entered as a **URL** in the browser, **‘sachin’** will be supplied to **hello()** function as argument.

* We can have different types of data into the url.



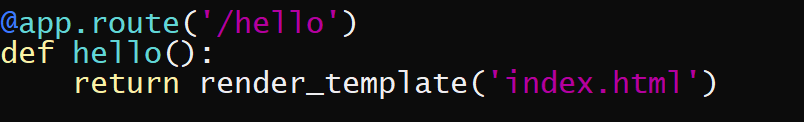


Here in the above code first url takes the int argument and second takes the float argument

So when we request “localhost/blog/1” the first url will be requested and its corresponding function will respond and if request is made for “localhost/per/1.1 the second function will respond.

* If we write like @app.route(“/flask”) and write @app.route(“/flask/”) both will give the same output the only difference is that the first url will not respond to “/flask/” while the second one will respond to both the urls.
* All the html pages will be kept in templates folder such that whenever we render the template or html page it will return from templates folder and all the css and javascripts file are kept in static folder.
* The method render\_template is imported from flask to render the template.

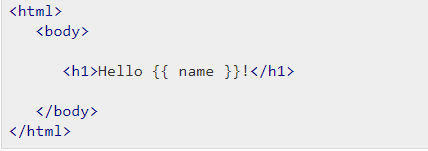
In the following code template is rendered when the request is made for /hello.



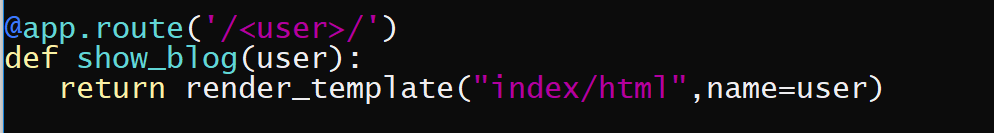
Where “index.html” can be any html page in the templates folder

* Application\_folder
  + - Templates
      * Index.html
* Flask uses **jinga2** template engine. A web template contains HTML syntax interspersed placeholders for variables and expressions (in these case Python expressions) which are replaced values when the template is rendered.

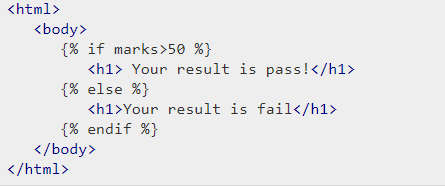
The following code is written in html page to have the dynamic value



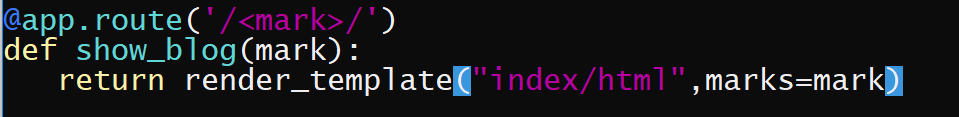
The following code is written in main file i.e app.py file



* The if statement can also be use in html page using jinja language like follow:

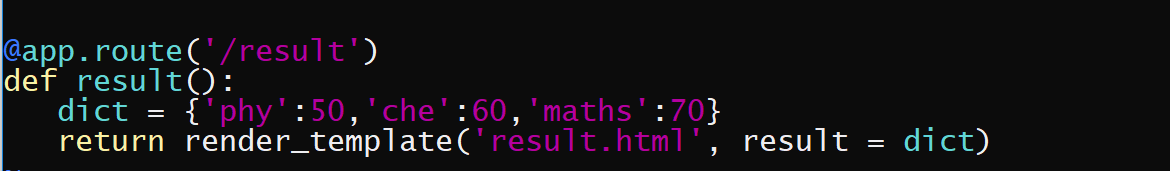


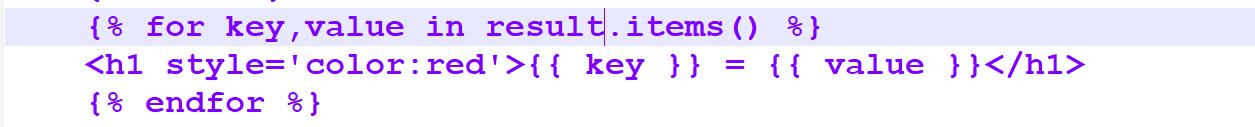
If the following code in written in app.py file:



* The for loop is also used in html pages using jinja language like follows.

If the dictionary Is returned from the main file to the html page then





* To link css or js or any static file that is kept in static folder the following syntax is used in html page:



Where static is the static folder and abcd.css is the name of file.

* Application\_folder
* Static
* abcd.css

If we render the ‘login’ form at localhost then the ‘login.html’ will be created in templates folder and views.py will render the template if request is made at ‘/’.

* Application\_folder
  + Templates
    - Login.html
* Static
* App.py

The code will be:



And login.html is: