

Flask – Variable Rules

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’**. Hence, if the **http://localhost:5000/hello/TutorialsPoint** is entered as a **URL** in the browser, **‘TutorialsPoint’** will be supplied to **hello()** function as argument.

```
from flask import Flask
app = Flask(__name__)

@app.route('/hello/<name>')
def hello_name(name):
    return 'Hello %s!' % name

if __name__ == '__main__':
    app.run(debug = True)
```

Save the above script as **hello.py** and run it from Python shell. Next, open the browser and enter URL **http://localhost:5000/hello/TutorialsPoint**.

The following output will be displayed in the browser.

```
Hello TutorialsPoint!
```

In addition to the default string variable part, rules can be constructed using the following converters –

Sr.No.	Converters & Description
1	int accepts integer
2	float For floating point value
3	path accepts slashes used as directory separator character

In the following code, all these constructors are used.

```
from flask import Flask
app = Flask(__name__)

@app.route('/blog/<int: postID>')
def show_blog(postID):
    return 'Blog Number %d' % postID

@app.route('/rev/<float: revNo>')
def revision(revNo):
    return 'Revision Number %f' % revNo

if __name__ == '__main__':
    app.run()
```

Run the above code from Python Shell. Visit the URL **http://localhost:5000/blog/11** in the browser.

The given number is used as argument to the **show_blog()** function. The browser displays the following output –

```
Blog Number 11
```

Enter this URL in the browser – **http://localhost:5000/rev/1.1**

The **revision()** function takes up the floating point number as argument. The following result appears in the browser window –

```
Revision Number 1.100000
```

The URL rules of Flask are based on **Werkzeug's** routing module. This ensures that the URLs formed are unique and based on precedents laid down by Apache.

Consider the rules defined in the following script –

```
from flask import Flask
app = Flask(__name__)

@app.route('/flask')
def hello_flask():
    return 'Hello Flask'

@app.route('/python/')
def hello_python():
    return 'Hello Python'
```

```
if __name__ == '__main__':  
    app.run()
```

Both the rules appear similar but in the second rule, trailing slash (/) is used. As a result, it becomes a canonical URL. Hence, using [/python](#) or [/python/](#) returns the same output. However, in case of the first rule, [/flask/](#) URL results in **404 Not Found** page.

Revision #1

Created 14 December 2022 11:20:39 by Admin

Updated 14 December 2022 11:21:04 by Admin