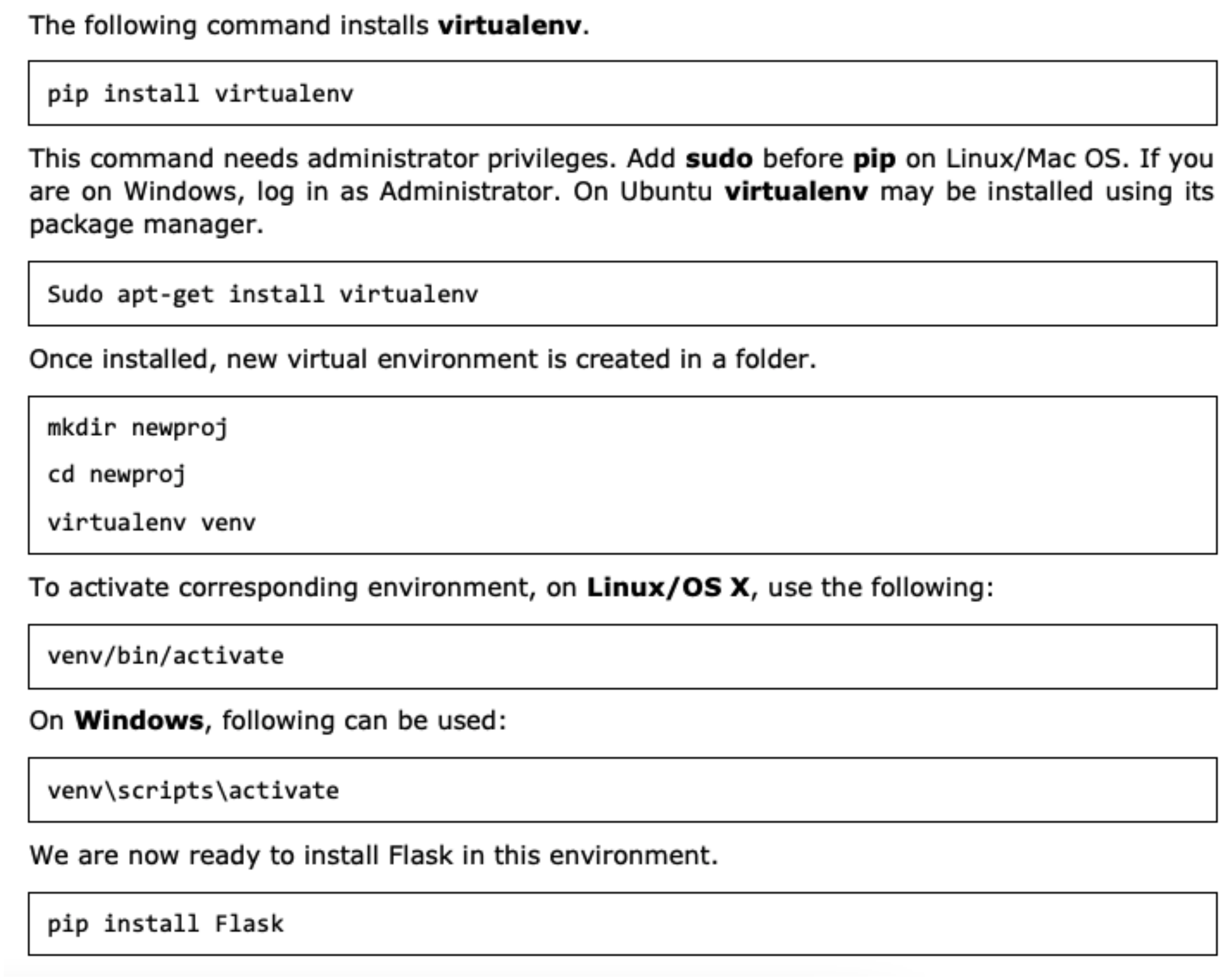
Flask

web development,
one drop at a time

Installation



The following command installs **virtualenv**.

```
pip install virtualenv
```

This command needs administrator privileges. Add **sudo** before **pip** on Linux/Mac OS. If you are on Windows, log in as Administrator. On Ubuntu **virtualenv** may be installed using its package manager.

```
Sudo apt-get install virtualenv
```

Once installed, new virtual environment is created in a folder.

```
mkdir newproj

cd newproj

virtualenv venv
```

To activate corresponding environment, on **Linux/OS X**, use the following:

```
venv/bin/activate
```

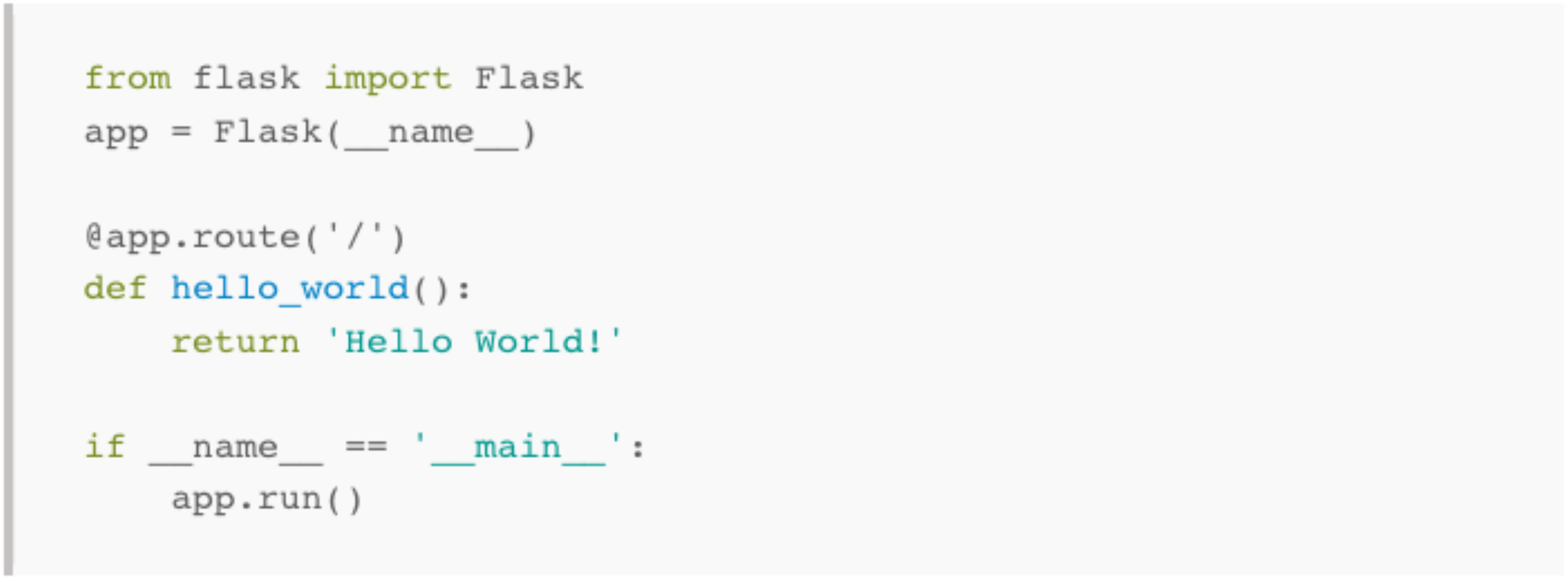
On **Windows**, following can be used:

```
venv\scripts\activate
```

We are now ready to install Flask in this environment.

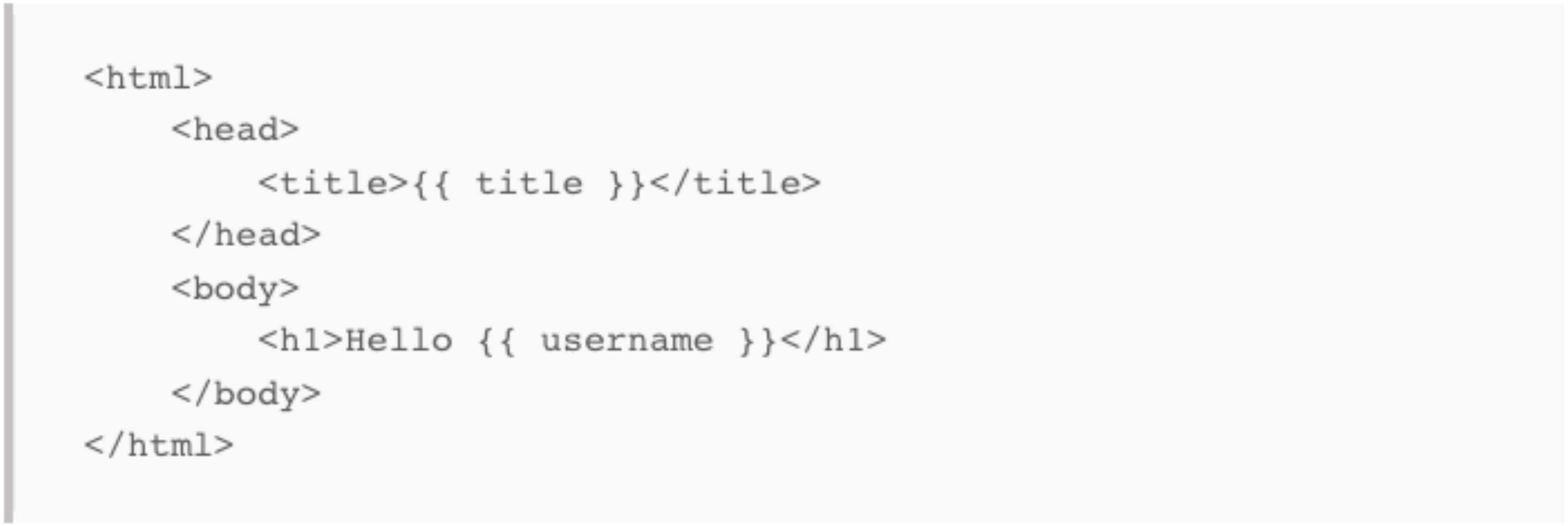
```
pip install Flask
```

Hello World! App



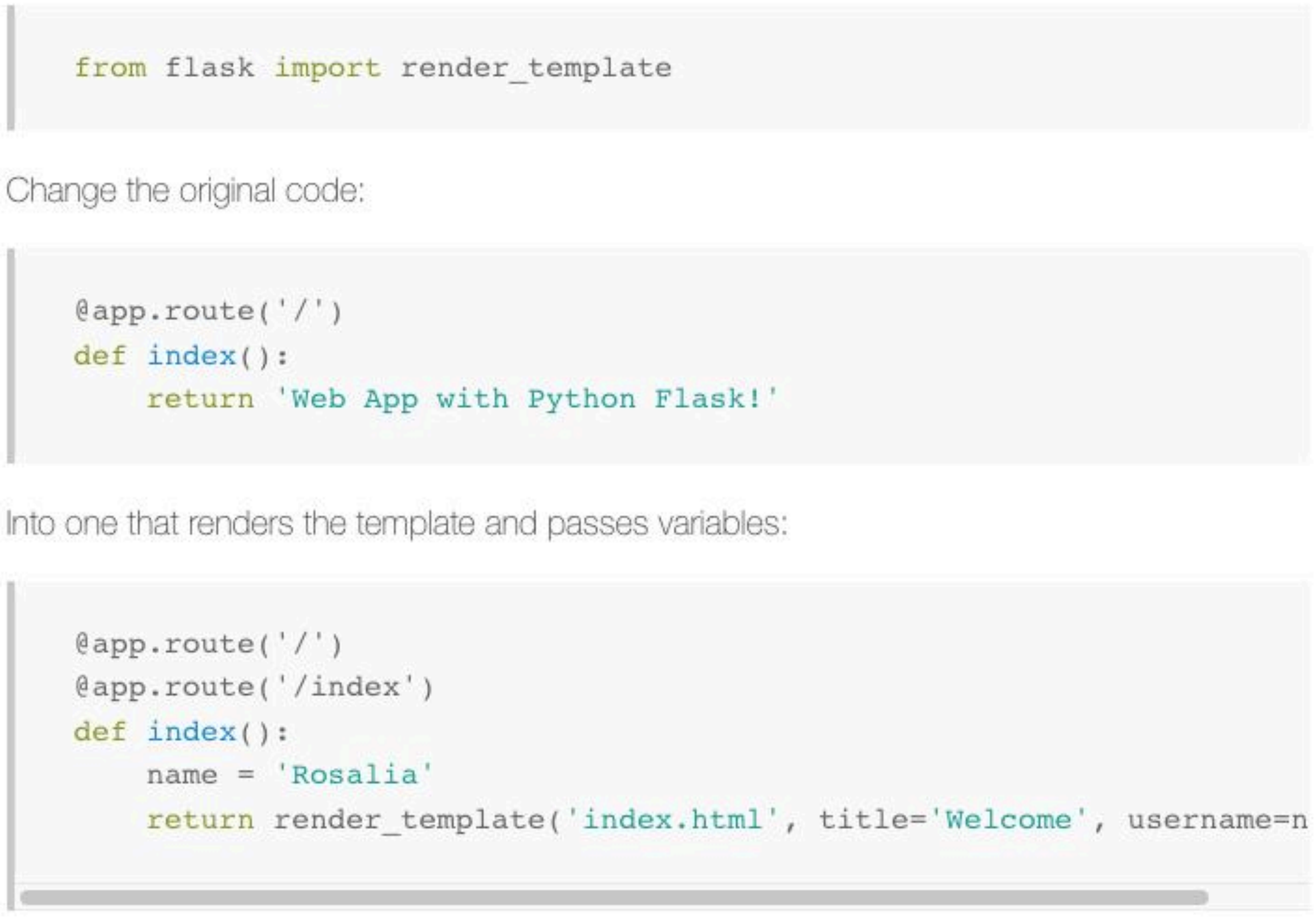
```python
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello World!'

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

Variables to HTML



```
<html>
    <head>
        <title>{{ title }}</title>
    </head>
    <body>
        <h1>Hello {{ username }}</h1>
    </body>
</html>
```

Render Template



```
from flask import render_template
```

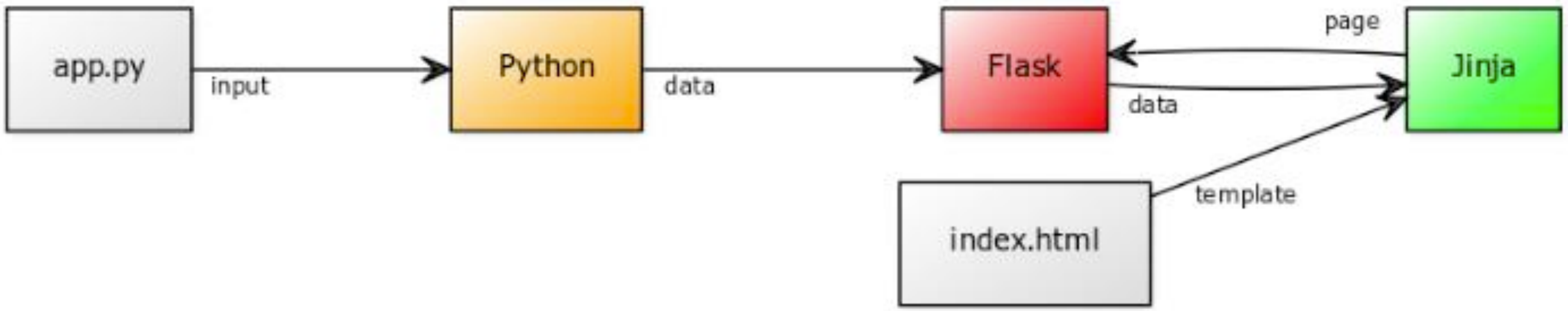
Change the original code:

```
@app.route('/')
def index():
    return 'Web App with Python Flask!'
```

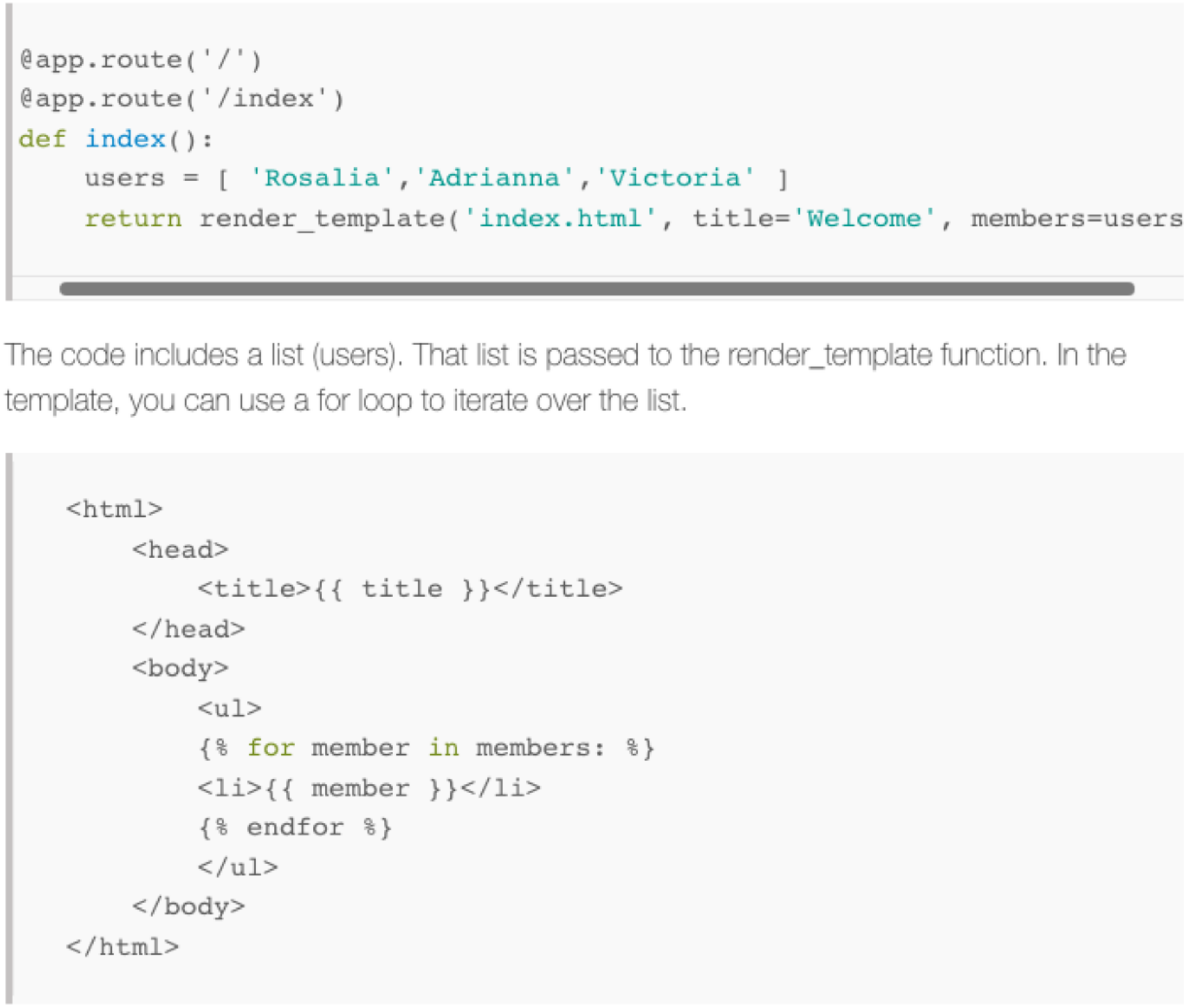
Into one that renders the template and passes variables:

```
@app.route('/')
@app.route('/index')
def index():
    name = 'Rosalia'
    return render_template('index.html', title='Welcome', username=n
```

Render Template



Loops

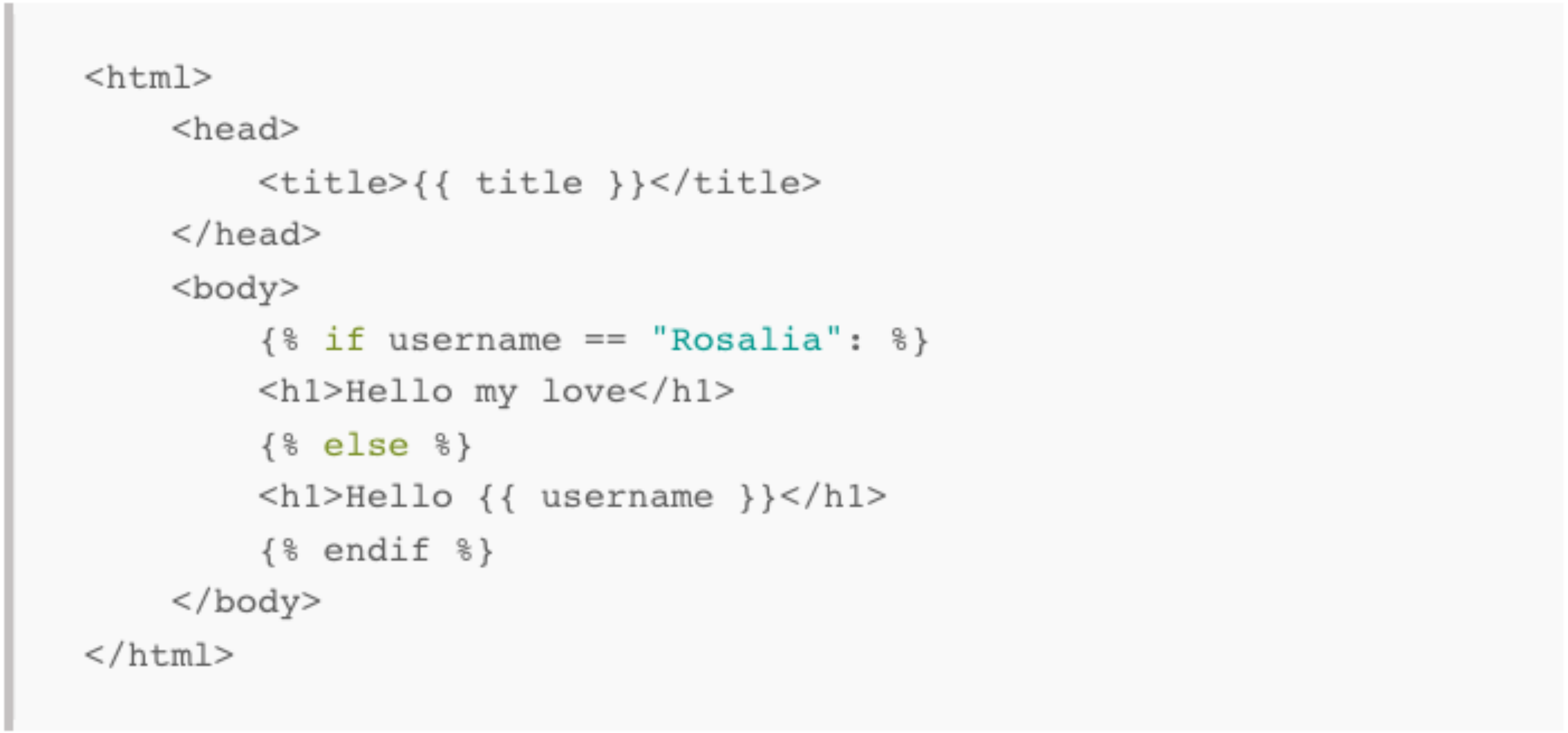


```python
@app.route('/')
@app.route('/index')
def index():
    users = [ 'Rosalia','Adrianna','Victoria' ]
    return render_template('index.html', title='Welcome', members=users
```

The code includes a list (users). That list is passed to the render_template function. In the template, you can use a for loop to iterate over the list.

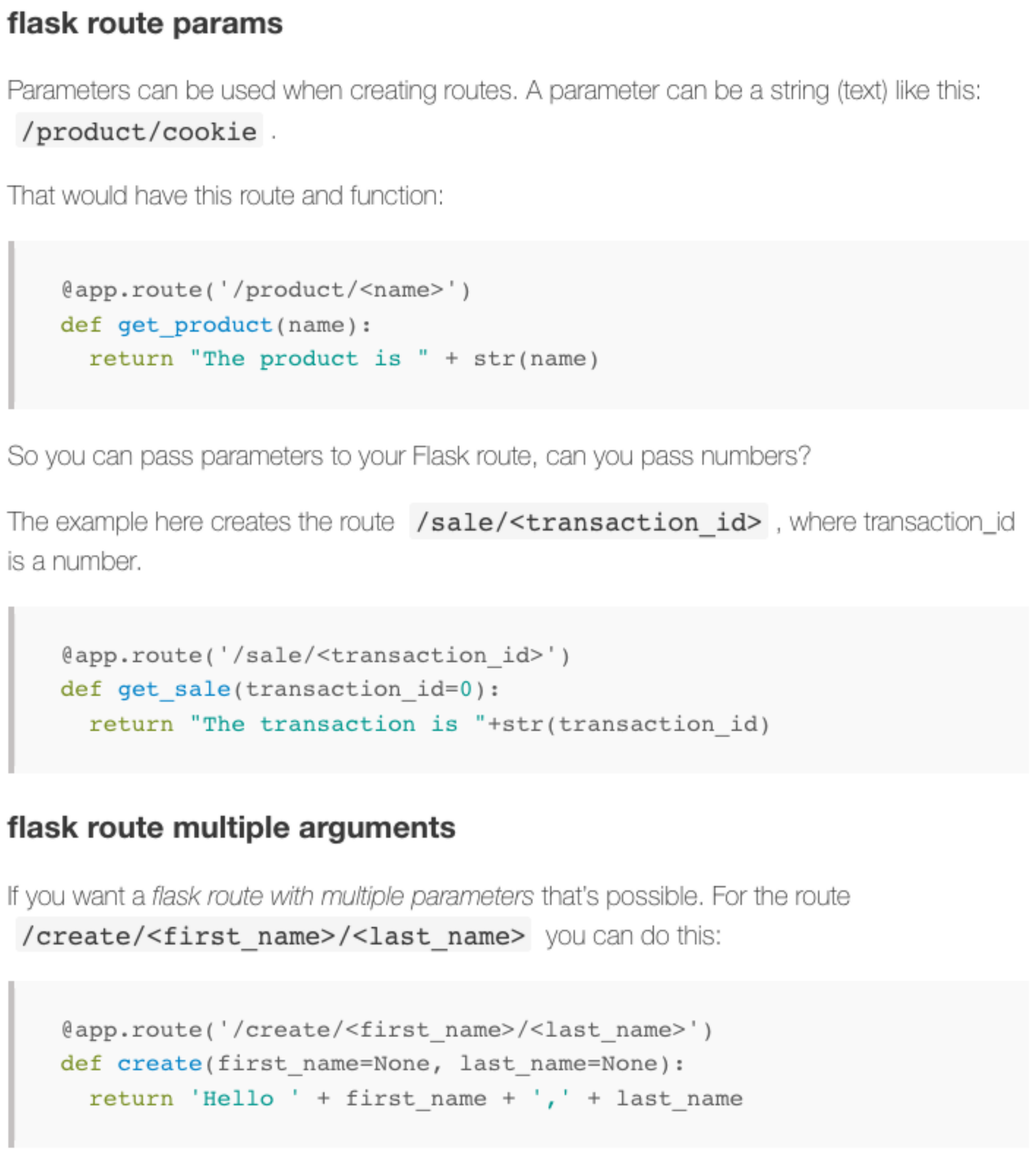
```html
<html>
    <head>
        <title>{{ title }}</title>
    </head>
    <body>
        <ul>
        {% for member in members: %}
        <li>{{ member }}</li>
        {% endfor %}
        </ul>
    </body>
</html>
```

If-else



```
<html>
    <head>
        <title>{{ title }}</title>
    </head>
    <body>
        {% if username == "Rosalia": %}
        <h1>Hello my love</h1>
        {% else %}
        <h1>Hello {{ username }}</h1>
        {% endif %}
    </body>
</html>
```

Routing



flask route params

Parameters can be used when creating routes. A parameter can be a string (text) like this: `/product/cookie` .

That would have this route and function:

```
@app.route('/product/<name>')
def get_product(name):
  return "The product is " + str(name)
```

So you can pass parameters to your Flask route, can you pass numbers?

The example here creates the route `/sale/<transaction_id>` , where transaction_id is a number.

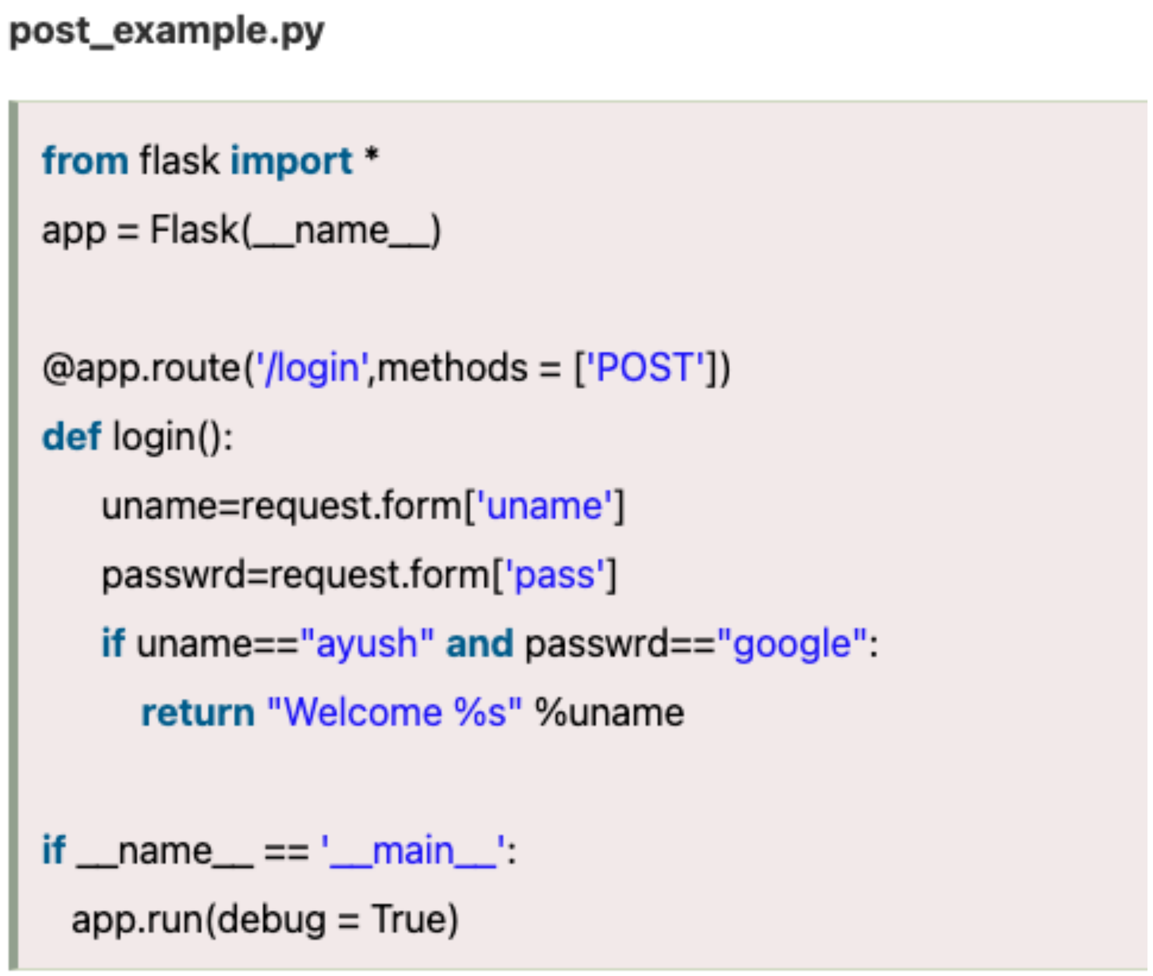
```
@app.route('/sale/<transaction_id>')
def get_sale(transaction_id=0):
  return "The transaction is "+str(transaction_id)
```

flask route multiple arguments

If you want a *flask route with multiple parameters* that's possible. For the route `/create/<first_name>/<last_name>` you can do this:

```
@app.route('/create/<first_name>/<last_name>')
def create(first_name=None, last_name=None):
  return 'Hello ' + first_name + ',' + last_name
```

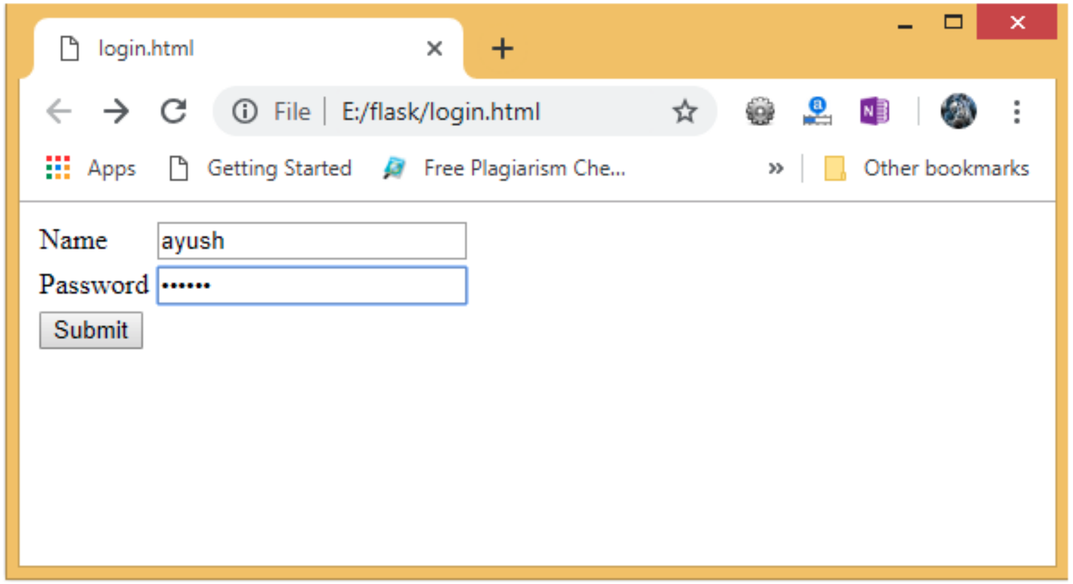
HTTP Methods: POST



post_example.py

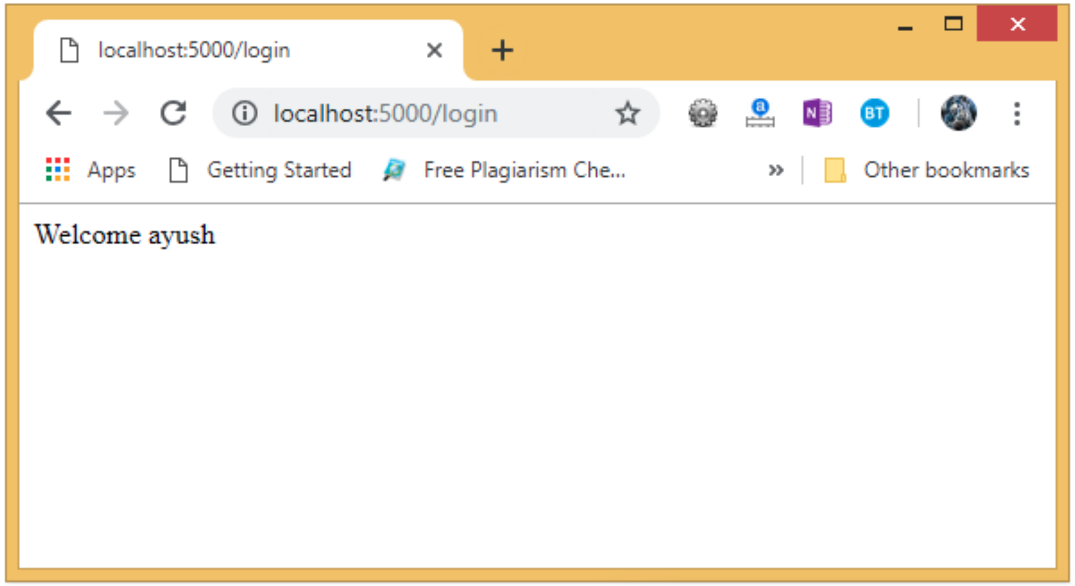
```
from flask import *
app = Flask(__name__)


@app.route('/login',methods = ['POST'])
def login():
   uname=request.form['uname']
   passwrd=request.form['pass']
   if uname=="ayush" and passwrd=="google":
      return "Welcome %s" %uname


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

HTTP Methods: POST



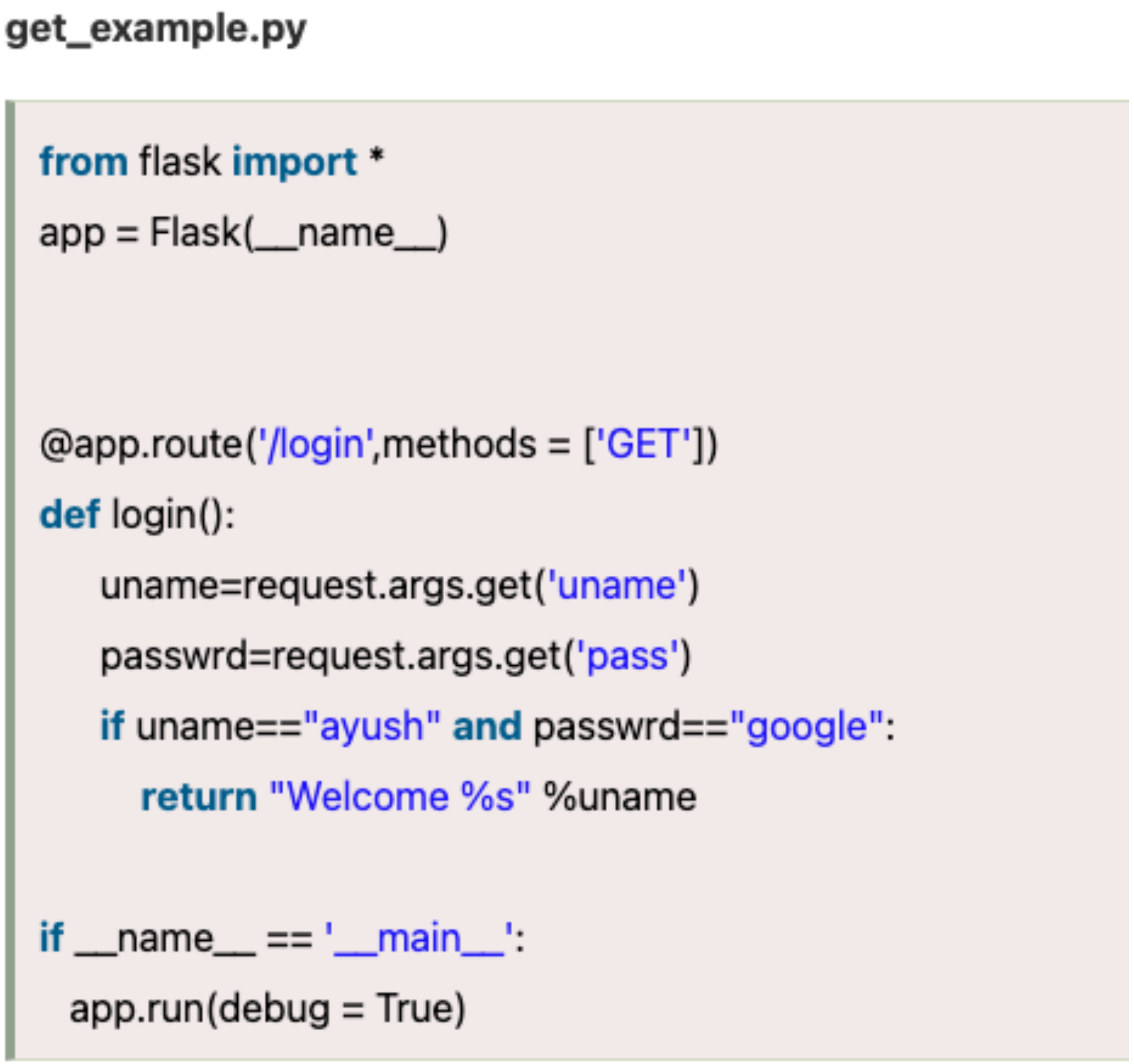
Name ayush
Password ••••••

Give the required input and click Submit, we will get the following result.



Welcome ayush

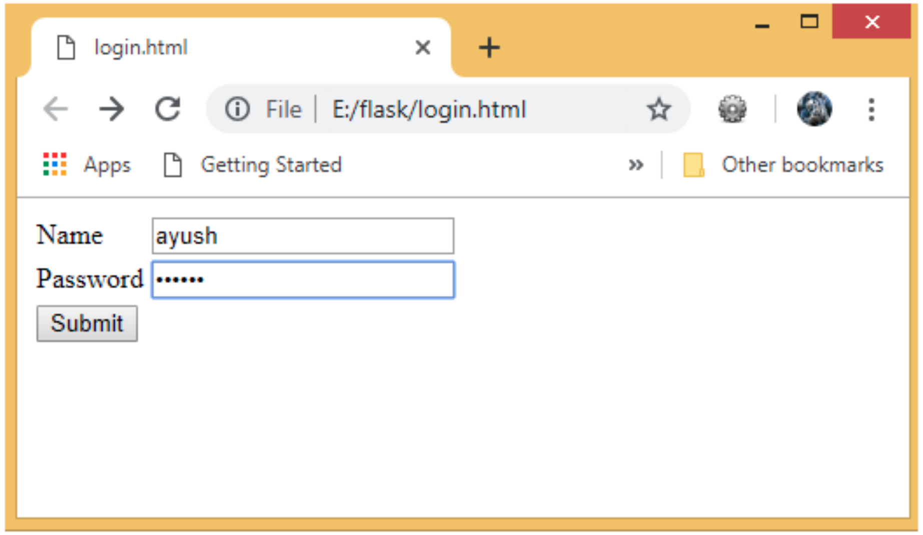
HTTP Methods: GET



get_example.py

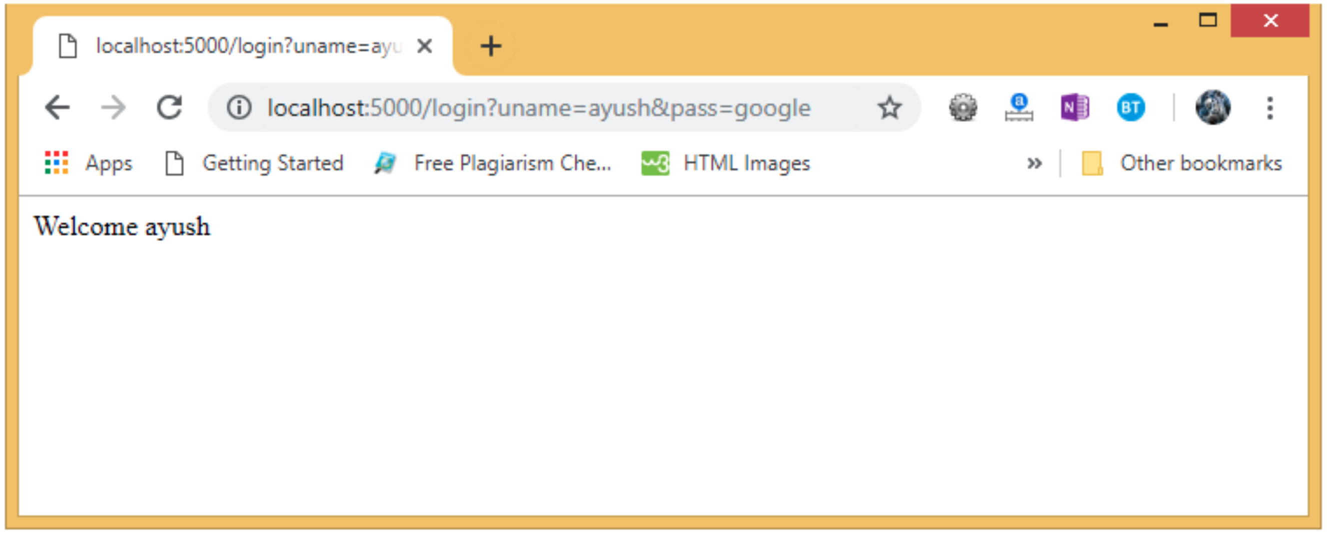
```
from flask import *
app = Flask(__name__)



@app.route('/login',methods = ['GET'])
def login():
    uname=request.args.get('uname')
    passwrd=request.args.get('pass')
    if uname=="ayush" and passwrd=="google":
        return "Welcome %s" %uname

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

HTTP Methods: GET


login.html
File | E:/flask/login.html
Apps Getting Started Other bookmarks

Name ayush
Password ••••••
Submit

Now, click the submit button.


localhost:5000/login?uname=ayu
localhost:5000/login?uname=ayush&pass=google
Apps Getting Started Free Plagiarism Che... HTML Images Other bookmarks

Welcome ayush