

Flask, continuação

Aula 12

Jinja 2

- ▶ Delimitadores Jinja 2 em templates HTML

`{% %}` para declarações

`{{ }}` para expressões a serem impressas na saída

`{# #}` para comentários

`# ##` para declarações de linha



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

```
@app.route('/hello/<int:score>')
def hello_name(score):
    return render_template('hello.html', marks = score)
```

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

```
<!doctype html>
<html>
  <body>

    {% if marks>50 %}
    <h1> Your result is pass!</h1>
    {% else %}
    <h1>Your result is fail</h1>
    {% endif %}

  </body>
</html>
```

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

@app.route('/result')
def result():
    dict = {'phy':50,'che':60,'maths':70}
    return render_template('result.html', result = dict)

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



```
<!doctype html>
<html>
  <body>

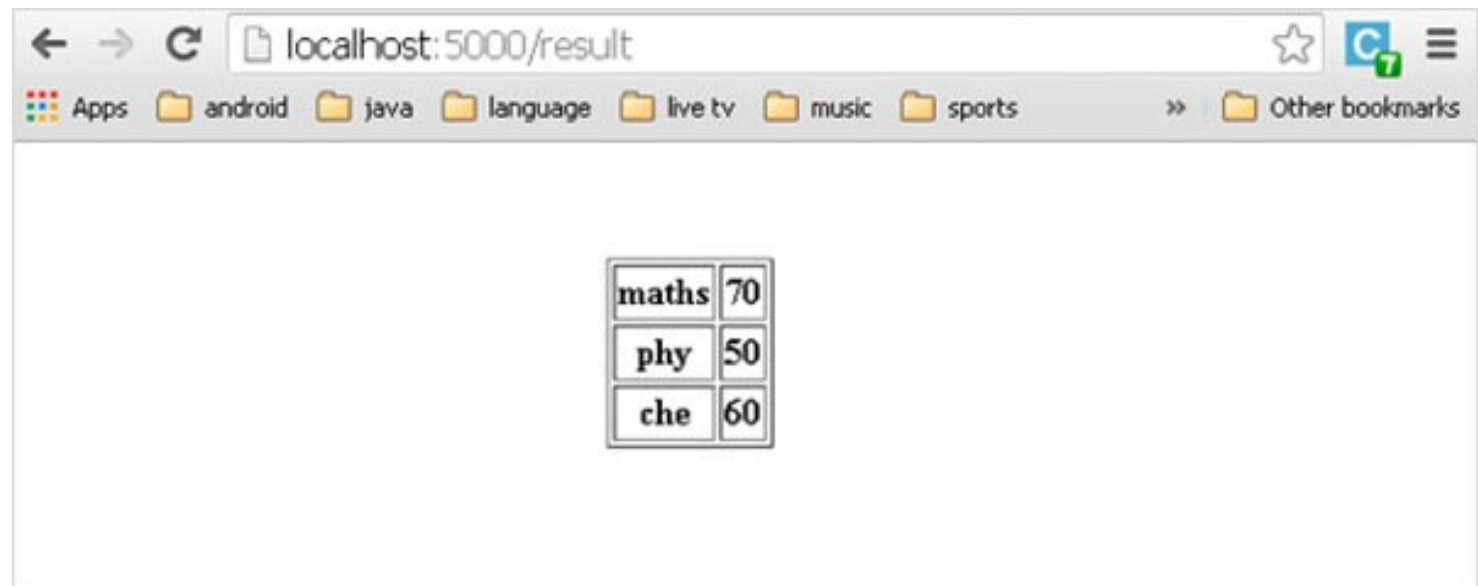
    <table border = 1>
      {% for key, value in result.iteritems()
%}

        <tr>
          <th> {{ key }} </th>
          <td> {{ value }} </td>
        </tr>

      {% endfor %}
    </table>

  </body>
</html>
```





Arquivos estáticos e templates

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

```
@app.route("/")  
def index():  
    return  
    render_template("index.html")
```

```
if __name__ == '__main__':
```

```
<html>  
  <head>  
    <script type = "text/javascript"  
      src = "{{ url_for('static', filename = 'hello.js') }}"  
></script>  
  </head>  
<body>  
  <input type = "button" onclick = "sayHello()" value = "Say  
Hello" />  
</body>  
</html>
```

hello.js

```
function sayHello() {  
  alert("Hello World")  
}
```



Objeto de requisição

- ▶ Formulário – dicionário contendo pares de chaves e valores.
- ▶ Args – conteúdo de um string que é parte de uma URL após a interrogação (?)
- ▶ Cookies – objeto de dicionário contendo nomes e valores de cookies.
- ▶ Arquivos – dados de um arquivo.
- ▶ Método – método de requisição corrente



```
from flask import Flask, render_template, request
app = Flask(__name__)
```

```
@app.route('/')
def student():
    return render_template('student.html')
```

```
@app.route('/result', methods = ['POST', 'GET'])
def result():
    if request.method == 'POST':
        result = request.form
        return render_template("result.html", result = result)
```

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



```
<html>
  <body>

    <form action = "http://localhost:5000/result" method =
"POST">
      <p>Name <input type = "text" name = "Name"
/></p>
      <p>Physics <input type = "text" name =
"Physics" /></p>
      <p>Chemistry <input type = "text" name =
"chemistry" /></p>
      <p>Maths <input type = "text" name =
"Mathematics" /></p>
      <p><input type = "submit" value = "submit" /></p>
    </form>

  </body>
</html>
```



```
<!doctype html>
```

```
<html>
```

```
  <body>
```

```
    <table border = 1>
```

```
      {% for key, value in result.iteritems() %}
```

```
        <tr>
```

```
          <th> {{ key }} </th>
```

```
          <td> {{ value }} </td>
```

```
        </tr>
```

```
      {% endfor %}
```

```
    </table>
```


```
  </body>
```


```
</html>
```




←


→

 localhost:5000



 Search

»



Name

Physics

Chemistry

Maths

submit



Cookies

```
@app.route('/')  
def index():  
    return render_template('index.html')
```

```
<html>  
  <body>  
  
    <form action = "/setcookie" method = "POST">  
      <p><h3>Enter userID</h3></p>  
      <p><input type = 'text' name = 'nm' /></p>  
      <p><input type = 'submit' value = 'Login' /></p>  
    </form>  
  
  </body>  
</html>
```



```
@app.route('/setcookie', methods = ['POST', 'GET'])
def setcookie():
    if request.method == 'POST':
        user = request.form['nm']

        resp =
make_response(render_template('readcookie.html'))
        resp.set_cookie('userID', user)

    return resp
```

```
@app.route('/getcookie')
def getcookie():
    name = request.cookies.get('userID')
    return '<h1>welcome '+name+'</h1>'
```



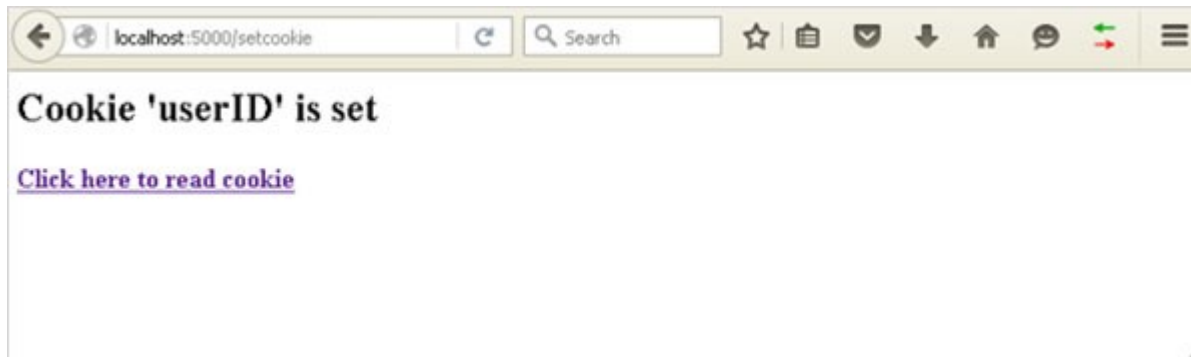


A web browser window with the address bar showing 'localhost:5000'. The page has a title 'Enter userID'. Below the title is a text input field containing the text 'admin'. Below the input field is a button labeled 'Login'.

Enter userID

admin

Login



A web browser window with the address bar showing 'localhost:5000/setcookie'. The page displays the message 'Cookie 'userID' is set'. Below this message is a blue, underlined link that says 'Click here to read cookie'.

Cookie 'userID' is set

[Click here to read cookie](#)



A web browser window with the address bar showing 'localhost:5000/getcookie'. The page displays the message 'welcome admin' in a large, bold, black font.

welcome admin

Sessão

- ▶ Um objeto sessão é um dicionário armazenado no servidor. Cada sessão de cliente é atribuído um session ID. Os dados são cifrados com uma chave secreta.

```
Session['username'] = 'admin'
```

```
session.pop('username', None)
```



```
@app.route('/')
def index():
    if 'username' in session:
        username = session['username']
        return 'Logged in as ' + username + '<br>' + \
            "<b><a href = '/logout'>click here to log out</a></b>"
    return "You are not logged in <br><a href = '/login'></b>"
+ \
    "click here to log in</b></a>"
```



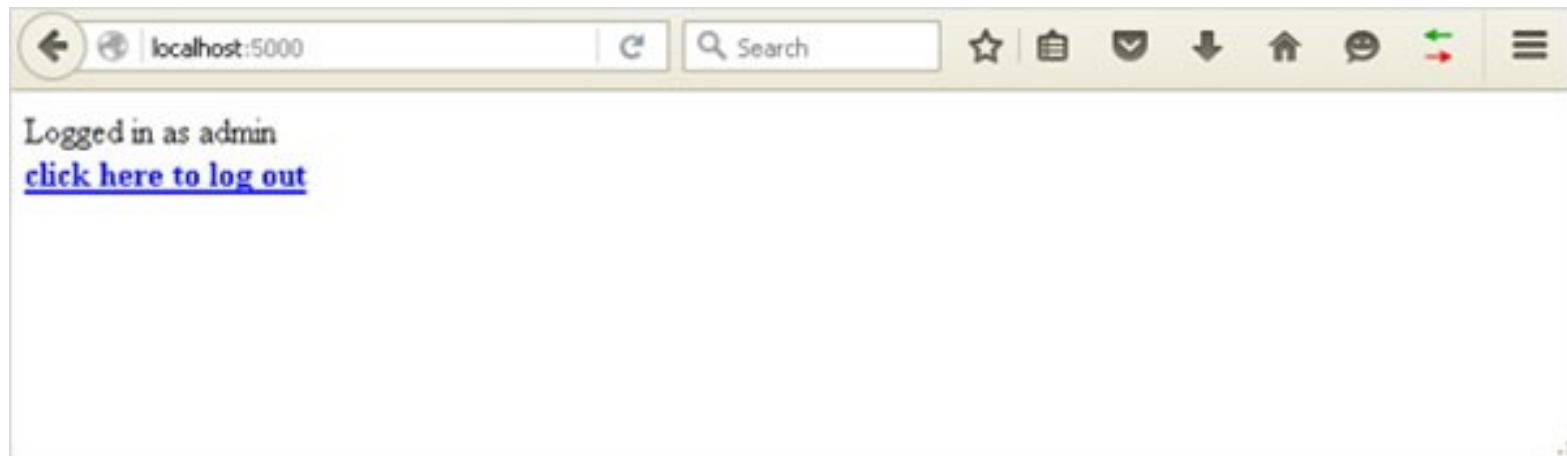
```
@app.route('/login', methods = ['GET', 'POST'])
def login():
    if request.method == 'POST':
        session['username'] = request.form['username']
        return redirect(url_for('index'))
    return ''
```

```
<form action = "" method = "post">
    <p><input type = text name = username/></p>
    <p><input type = submit value = Login/></p>
</form>
```

```
'''
```



```
@app.route('/logout')
def logout():
    # remove the username from the session if it is
    there
    session.pop('username', None)
    return redirect(url_for('index'))
```



Definir a chave secreta para sessões

```
from flask import Flask, session, redirect, url_for,  
escape, request  
app = Flask(__name__)  
app.secret_key = 'any random string'
```



Exercício

- ▶ 1) Desenvolver uma aplicação Flask usando cookies.
- ▶ 2) Desenvolver uma aplicação Flask usando sessões.
- ▶ 3) Desenvolva uma aplicação que permita carregar arquivos em um servidor, listar e acessar arquivos no servidor.

