

Flask - javascript, CSS 사용하기

자바스크립트나 CSS를 사용하려면 static이란 폴더를 생성하여 거기서 만들면 된다.

```
static/hello.js

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

```
templates/hello.html

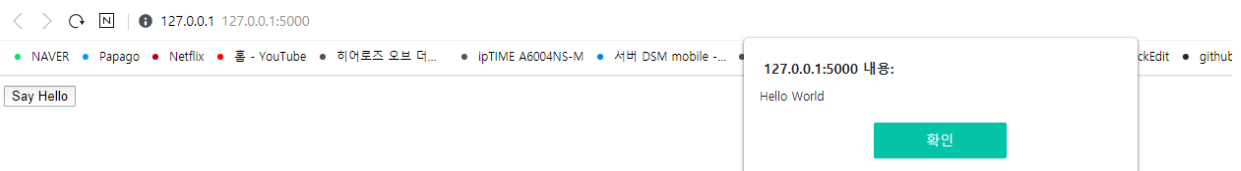
<!DOCTYPE html>
<html>
<head>
    <script type="text/javascript"
        src="{{ url_for('static', filename = 'hello.js') }}"></script>
</head>
<body>
    <input type="button" onclick="sayHello()" value="Say Hello" />
</body>
</html>
```

```
routes.py

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

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

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



CSS 사용하기

```
static\style.css

* {
  box-sizing: border-box;
}

*:focus {
  outline: none;
}

body {
  font-family: Arial;
  background-color: #3498DB;
  padding: 50px;
}

.login {
  margin: 20px auto;
  width: 300px;
}

.login-screen {
  background-color: #FFF;
  padding: 20px;
  border-radius: 5px
}

.app-title {
  text-align: center;
  color: #777;
}

.login-form {
  text-align: center;
}

.control-group {
  margin-bottom: 10px;
}

input {
  text-align: center;
  background-color: #ECF0F1;
  border: 2px solid transparent;
  border-radius: 3px;
  font-size: 16px;
  font-weight: 200;
  padding: 10px 0;
  width: 250px;
  transition: border .5s;
}
```

```

input:focus {
border: 2px solid #3498DB;
box-shadow: none;
}

.btn {
border: 2px solid transparent;
background: #3498DB;
color: #ffffff;
font-size: 16px;
line-height: 25px;
padding: 10px 0;
text-decoration: none;
text-shadow: none;
border-radius: 3px;
box-shadow: none;
transition: 0.25s;
display: block;
width: 250px;
margin: 0 auto;
}

.btn:hover {
background-color: #2980B9;
}

.login-link {
font-size: 12px;
color: #444;
display: block;
margin-top: 12px;
}

```

templates\hello.html

```

<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="/static/style.css" type="text/css">
</head>
<body>
  <form action="/login" method="POST">
    <div class="login">
      <div class="login-screen">
        <div class="app-title">
          <h1>Login</h1>
        </div>

        <div class="login-form">
          <div class="control-group">

```

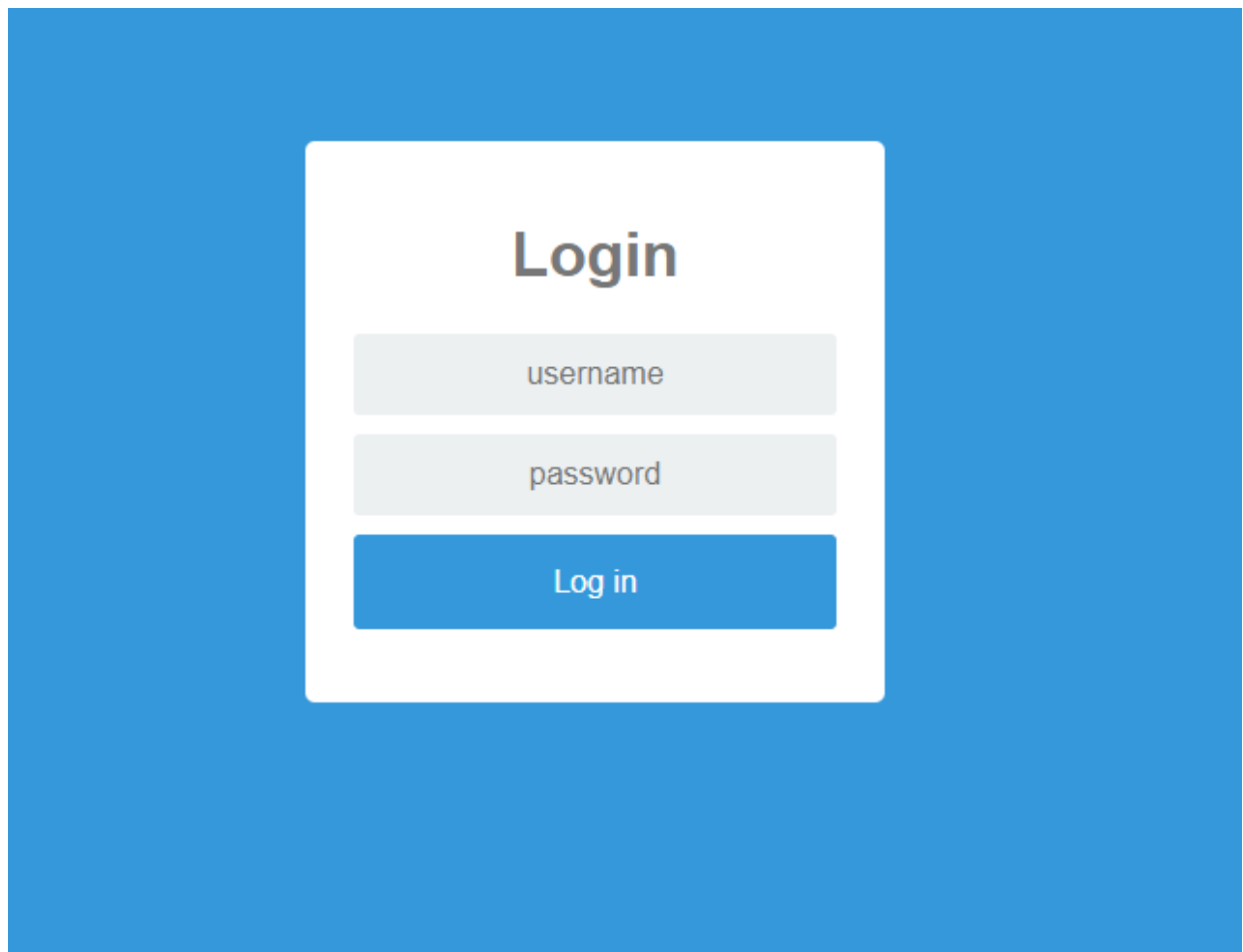
```

        <input type="text" class="login-field" value="" place
holder="username" name="username">
        <label class="login-field-icon fui-user" for="login-n
ame"></label>
    </div>

    <div class="control-group">
        <input type="password" class="login-field" value="" p
laceholder="password" name="password">
        <label class="login-field-icon fui-lock" for="login-p
ass"></label>
    </div>

    <input type="submit" value="Log in" class="btn btn-pr
imary btn-large btn-block" >
    <br>
</div>
</div>
</div>
</form>
</body>
</html>

```



폼으로 데이터 전송

routes.py

```
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)
```

result = request.form

폼 관련 데이터를 다루는 부분은 이 코드이다.

요청이 POST일 경우,

request 의 form 객체에 해당 폼에 입력된 모든 데이터가 들어있다.

만약 특정 필드 값만 뽑아내고 싶다면 request.form['name'] 와 같은 식으로 처리하면 된다.

위의 코드는 student.html 에 form 에 데이터를 작성하여 보내면 result.html에서 해당 값을 확인할 수 있는 구조이다.

templates\student.html

```
<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>
```

templates\result.html

```

<!DOCTYPE html>
<html>
<body>
  <table border="1">
    {% for key, value in result.items() %}
      <tr>
        <th> {{ key }} </th>
        <td> {{ value }} </td>
      </tr>
    {% endfor %}
  </table>
</body>
</html>

```

폼 객체의 경우 각 데이터를 딕셔너리 형태로 보관하기 때문에 기존에 파이썬에서 사용하는 `items()` 등의 메소드를 사용하여 출력할 수 있다.

< > ↺ [N] | ⓘ 127.0.0.1 127.0.0.1:5000

● NAVER ● Papago ● Netflix ● 홈 - YouTube ● 히어로

Name

Physics

Chemistry

Maths

< > ↺ [N] | ⓘ localhost localhost:5000/result

● NAVER ● Papago ● Netflix ● 홈 - YouTube ● 히어로

Name	용광순
Physics	100
chemistry	100
Mathematics	100