

Web Study

7 week

Contents

1. 과제 코드 코멘트
2. flask - static file
3. flask - http method (GET/POST)
4. SSR, CSR
5. The end.

1. 과제 코드 코멘트

2. flask - static file

- static
 - css
 - styles.css
 - images
 - hello.png
- templates
 - index.html
- app.py

```
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello world!</title>
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/styles.css') }}" />
  </head>
  <body>
    
  </body>
</html>
```

3. flask - http method (GET/POST)

get vs post

```

# application.py
import sys
from flask import Flask, render_template, request, redirect

app = Flask(__name__)

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

@app.route("/login", methods=["GET", "POST"])
def login():
    if request.method == "GET":
        return render_template("login.html")
    elif request.method == "POST":
        user = request.form["name"]
        return redirect(f"/{user}")

@app.route("/<user>")
def user(user):
    return f"<h1>{user}</h1>"

if __name__ == "__main__":
    # app.run(host="localhost", port=5000, debug=True)
    app.run(host="0.0.0.0", port=int(sys.argv[1]), debug=True)

```

- flask.redirect : flask 코드에서 특정 주소로 이동


```
<!-- templates/login.html -->
<html>
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Hello world!</title>
    <link
      rel="stylesheet"
      type="text/css"
      href="{{ url_for('static', filename='css/styles.css') }}"
    />
  </head>
  <body>
    <h1>Login Page</h1>
    <form method="post">
      <span>이름 : </span>
      <input type="text" name="name" />
      <button type="submit">제출</button>
    </form>
  </body>
</html>
```

4. SSR, CSR, json

- json
- SSR : Server Side Rendering
- CSR : Client Side Rendering

5. The end.

Web Frontend

1. Javascript
2. SPA 방식 앱 UI 기술 (react, ...)
3. CSS 라이브러리 사용 (SCSS, ...)
4. Typescript
5. babel, webpack
6. SSR 방식 앱 UI 기술 (nextjs, ...)
7. svelte (최신 기술)

Backend (python)

1. 네트워크 이론
2. Database 이론 (SQL, RDBMS, NoSQL)
3. flask
4. django
5. django rest framework
6. django graphql, fastAPI, Sanic, ...
7. OS 이론, Python 성능 최적화 이론 (비동기 프로그래밍, 컴파일 언어 포팅)
8. 인프라 이해
9. 클라우드 컴퓨팅 기술 (AWS, Azure, GCP)
10. ???

1. "어떤 기술을 학습할 때에는 그 기술이 생겨난 이유와 역사, 그리고 어떤 문제점을 해결하기 위해 그 기술이 등장했는 지를 명확하게 이해하고 상황에 맞게 유동적으로 사용하라" - 당근마켓
2. 우리나라에서의 취업 시장과 현실
3. 우리가 배운 것들과 앞으로 무엇을 하면 좋을까?
 - 데이터 엔지니어
 - 프론트엔드 개발자
 - 백엔드 개발자
 - 풀스택 엔지니어
4. 네트워킹에 대한 고찰
5. 마지막 과제 : 배운 것을 종합해서 깃허브 링크 또는 구름 ide 링크 또는 netlify 링크 또는 zip파일로 보내주세요!

