

Flask 2일

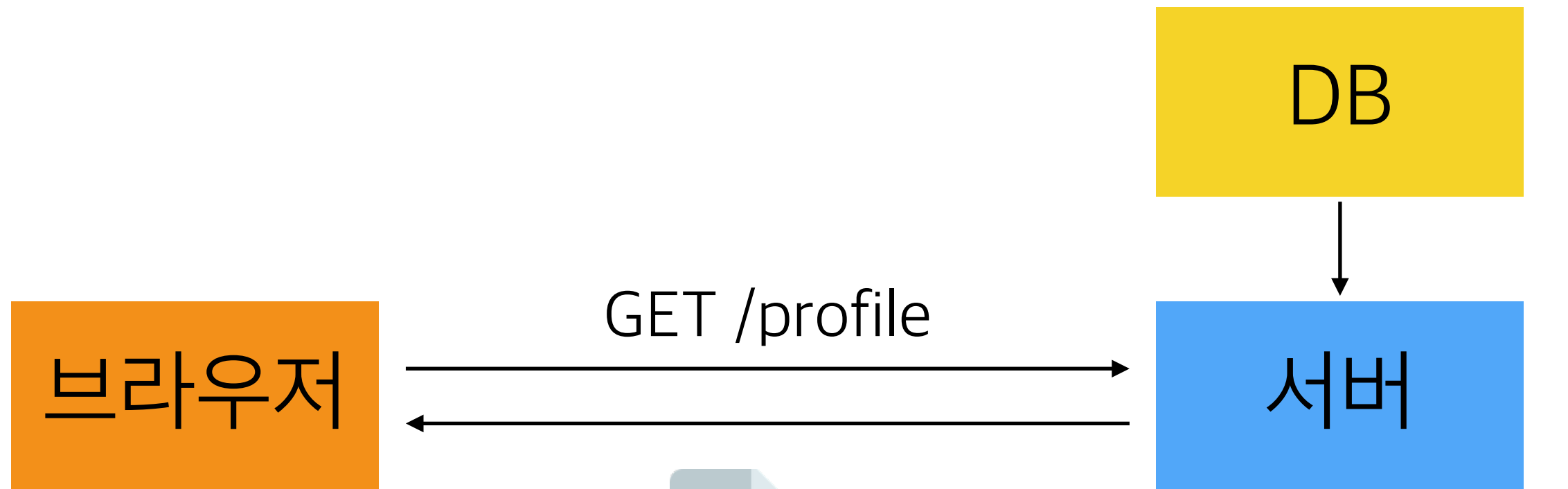
진자와 장고템플릿은 다르다!

jangjunha<jangjunha113@gmail.com>

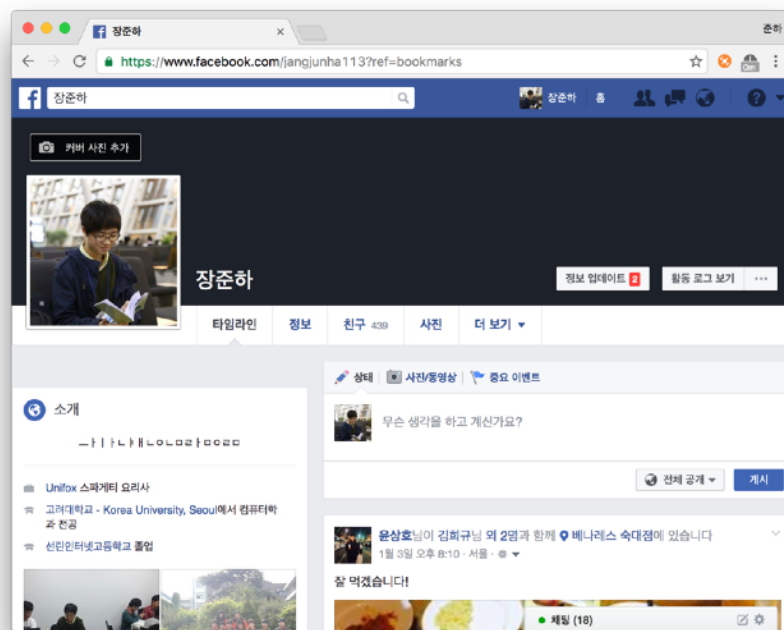
HTML 빠르게 훑어보기

- <http://www.w3schools.com>
- [https://namu.wiki/w/
%EB%8C%80%ED%95%9C%EB%AF%BC%EA%B5%AD/
%EC%A0%95%EB%8B%B9/%EB%AA%A9%EB%A1%9D
#s-1.1](https://namu.wiki/w/%EB%8C%80%ED%95%9C%EB%AF%BC%EA%B5%AD/%EC%A0%95%EB%8B%B9/%EB%AA%A9%EB%A1%9D#s-1.1)

웹 서버



```
<html>
  <h1>장준하</h1>
  <article>
    <p>배고파</p>
  </article>
</html>
```



HTML 변환

HTML파일 건네주기

```
/application.py  
/templates  
    /hello.html  
    /profile.html  
/static  
    /logo.png  
    /app.js  
    /app.css
```

```
render_template('hello.html')
```

정적 파일 건네주기

```
/application.py  
/templates  
    /hello.html  
    /profile.html  
/static  
    /logo.png  
    /app.js  
    /app.css
```

실습1

HTML파일 세 개를 만들고,
URL에 따라 각 HTML파일을 응답해봅시다.

URL	파일명
/	index.html
/users	user_list.html
/users/HeeGyu	profile.html

실습2

profile.html에 프로필사진을 삽입해봅시다.

템플릿 언어 Jinja2



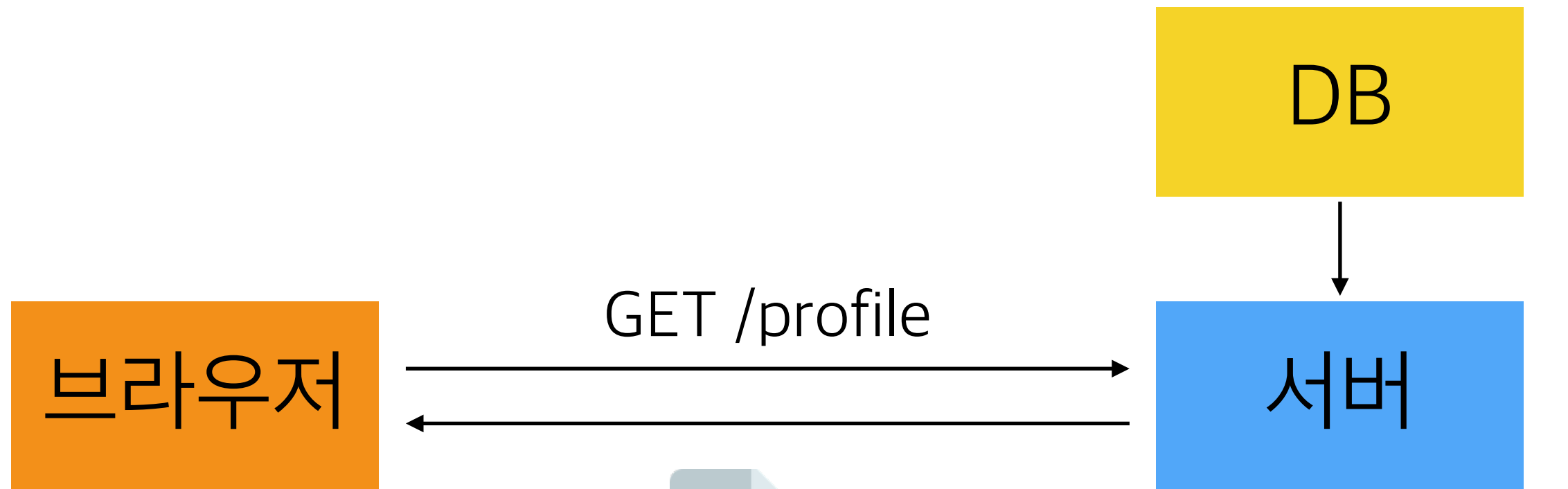
<http://jinja.pocoo.org/docs>

```
<title>{% block title %}{% endblock %}</title>
<ul>
  {% for user in users %}
    <li><a href="{{ user.url }}">{{ user.username }}</a></li>
  {% endfor %}
</ul>
```

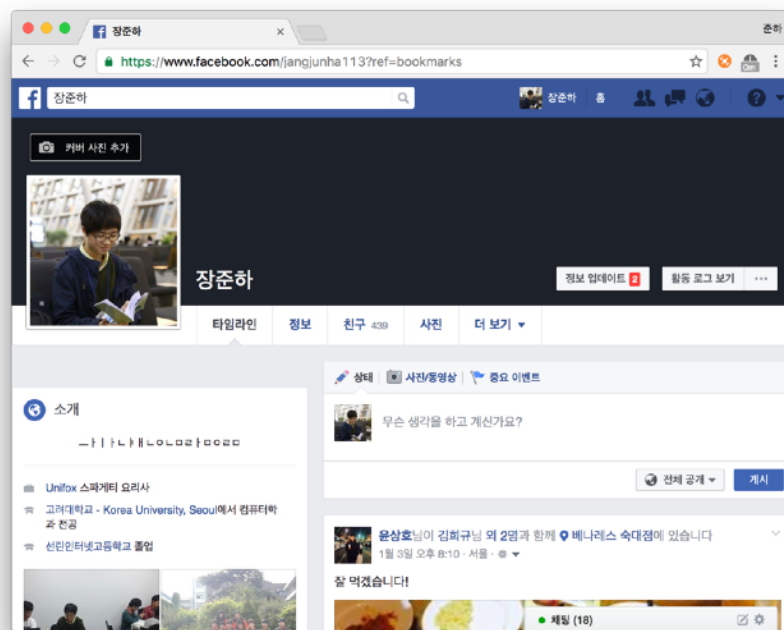
무작정 만들기

- Flaskr
- 새 프로젝트를 만듭시다.

웹 서버



```
<html>
  <h1>장준하</h1>
  <article>
    <p>배고파</p>
  </article>
</html>
```



변수 넘기기

index.html

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>HiHi</title>
</head>
<body>
  <h1>수능일까지 {{ countdown }}일</h1>
</body>
</html>
```

application.py

```
render_template('index.html', countdown=365)
```

반복문

index.html

```
<h3>덧글</h3>
<ul>
{% for comment in comments %}
    <li>
        <h4>{{ comment.writer }}:</h4>
        <p>{{ comment.content }}</p>
    </li>
{% endfor %}
</ul>
```

application.py

```
comments = [
    { 'writer': '김희규', 'content': '많이 남은 것 같지?' },
    { 'writer': '윤상호', 'content': '👑히어로즈 오브 더...' }
]

return render_template('index.html', comments=comments)
```

레이아웃, 상속

layout.html

(...)

```
<body>
  <h1>수능일까지 {{ countdown }}</h1>

  {% block content %}
  {% endblock %}
</body>
</html>
```

index.html

```
{% extends "layout.html" %}
{% block content %}
  <h3>덧글</h3>
  <ul>
    {% for comment in comments %}
      <li>
        <h4>{{ comment.writer }}</h4>
        <p>{{ comment.content }}</p>
      </li>
    {% endfor %}
  </ul>
{% endblock %}
```

write.html

```
{% extends "layout.html" %}
{% block content %}
    <form action="/" method="post">
        <input type="text" name="writer" placeholder="작성자"><br>
        <textarea name="content" placeholder="내용"></textarea><br>
        <input type="submit" value="작성"><br>
    </form>
{% endblock %}
```

Do!

실습3

사이트 소개 페이지를 하나 더 추가해봅시다.

새 페이지 역시 같은 레이아웃을 상속받읍시다.

/about으로 접근했을 때 나타나도록 해봅시다.

요청 데이터 처리하기

GET vs POST

- GET (QueryString)
 - 무언가 데이터를 가져올 때
 - 회원정보 페이지, 게시물 목록, ...
- POST (Form)
 - 새 글 작성
 - 회원가입 Form, 로그인 Form

Form (POST)

```
<form action="/new_comment" method="post">
  <input type="text" name="writer" placeholder="작성자">
  <textarea name="content" placeholder="내용"></textarea>
  <input type="submit" value="작성">
</form>
```

아이디	로그인	IP보안 ON
비밀번호		일회용 로그인
<input type="checkbox"/> 로그인 상태 유지		
회원가입 아이디/비밀번호 찾기		

제목을 입력해 주세요

자동 저장된 글 (2)

사진 동영상 파일 지도 음악 투표 외부컨텐츠

HTML

글꼴 9pt 가 간 가 과 감 가

URL ※

설정정보 스크랩 허용 복사 허용 서명 등록안함 CCL 표시할 변경

Daum Editor ?

미리보기

확인

취소

Form (POST)

request.form ← 일종의 Dictionary 비슷한 거

```
from flask import Flask, request

@app.route('/new_comment', methods=['POST'])
def post_comment():
    print(request.form['writer'])

    (...)
```


Querystring

<https://search.naver.com/search.naver?ie=UTF-8&query=2017+수능일>

ie=UTF-8
query=2017+수능일

Querystring

```
request.args
```

```
from flask import Flask, request

@app.route('/list')
def comment_list():
    print(request.args.get('page', 1))

    (...)
```

Request Object

- request.form
- request.args
- request.values
- request.cookies
- request.headers

Do!

실습4

회원가입 페이지를 만들고 넘어온 데이터를 콘솔에 출력해봅시다.

(선택1) 댓글 한 개만 보여주는 페이지를 만들어봅시다.

(선택2) 댓글 리스트에 페이지네이션을 적용해봅시다.

(선택) 실습5

세 가지 페이지가 있는 새 사이트를 만들어봅시다.

PAGE 1. 아래 두 페이지로 가는 링크가 있는 인덱스 페이지

PAGE 2. querystring으로 주어진 수의 배수 10개를 출력

PAGE 3. 랜덤한 숫자 10개를 출력

* 모든 페이지에 사이트 이름과 로고 이미지가 있습니다.

HTML태그 출력하기

XSS

Cross-Site Scripting

+ 데이터베이스