

Flask❤1일

장고 말고 플라스크 합시다

jangjunha<jangjunha113@gmail.com>



Flask

web development,
one drop at a time

[overview](#) // [docs](#) // [community](#) // [snippets](#) // [extensions](#)

Flask is a **microframework** for Python based on Werkzeug, Jinja 2 and good intentions. And before you ask: It's [BSD licensed](#)!

Flask is Fun

Latest Version: [0.11](#)

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

@app.route("/")
def hello():
    return "Hello World!"

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

And Easy to Setup

```
$ pip install Flask
$ python hello.py
* Running on http://localhost:5000/
```

Microframework

- 웹 개발에 필요한 **최소한**의 것들만 제공
 - HTTP Request , Routing , ...
- 다른 것들은 필요에 따라



express

Full-stack Framework

- 웹 개발에 필요한 **자주 사용되는 것들을 묶어서** 제공
 - 로그인 , DB 관련 라이브러리 , Form , ...

django

왜!

간단하게 시작해서 차근차근 웹을 이해해나가보아요

설치하기

- Virtualenv (옵션)
- Flask

Virtualenv ???

```
$ pip install Flask
```



Flask
0.12

Virtualenv ???

```
$ pip install Flask
```

Flask
0.12

디콘 프로젝트
Flask==0.12

외주 프로젝트
Flask==0.11

Virtualenv ???

Flask
0.12

디콘 프로젝트
Flask==0.12

Flask
0.11

외주 프로젝트
Flask==0.11

Virtualenv

MacOS / Linux

```
$ sudo pip install virtualenv
```

Ubuntu

```
$ sudo apt-get install python-virtualenv
```

<http://flask.pocoo.org/docs/0.12/installation/#virtualenv>

```
$ virtualenv venv
```

```
New python executable in venv/bin/python
```

```
Installing setuptools, pip.....done.
```

```
$ . venv/bin/activate
```

```
(venv) $
```

```
(venv) $ deactivate
```

```
$
```

Flask 설치

```
(venv) $ pip install Flask
```

Hello, World!

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

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

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

현재 위치 확인

```
$ pwd
```

```
/Users/junha/Developer/learn_flask
```

파일 리스트

```
$ ls
```

```
$ ls -al
```

```
$ ll
```

```
> dir
```

디렉토리 생성

```
$ mkdir <디렉토리 이름>
```


파일 삭제

```
$ rm <파일>
```

```
$ rm -r <디렉토리>
```

```
$ rm -f <파일> # 강제삭제
```

```
$ rm -rf <디렉토리> # 강제삭제
```

실행하기

```
$ python helloflask.py
```

```
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)  
127.0.0.1 - - [05/Jan/2017 01:22:30] "GET / HTTP/1.1" 200 -
```

HTTP 상태코드

코드	설명
200	성공
301	리다이렉트
304	수정되지 않음
400	잘못된 요청
404	찾을 수 없음
500	서버 오류

실행 옵션

```
if __name__ == '__main__':  
    app.run(host="0.0.0.0", port=8080, debug=True)
```

```
$ python helloflask.py
```

```
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)  
* Restarting with stat  
* Debugger is active!  
* Debugger pin code: 113-537-125  
127.0.0.1 - - [05/Jan/2017 01:24:10] "GET / HTTP/1.1" 200 -
```

Routing

- /
- /hello
- /user/HeeGyu
- /post/1234

string	슬래시(/)를 제외한 텍스트 (default)
int	정수
path	텍스트 - 슬래시(/) 포함

실습1

아래 각 URL에 따라 다른 결과를 출력해봅시다.

- /
- /users
- /users/HeeGyu
- /articles/3254

HTML파일 건네주기

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

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

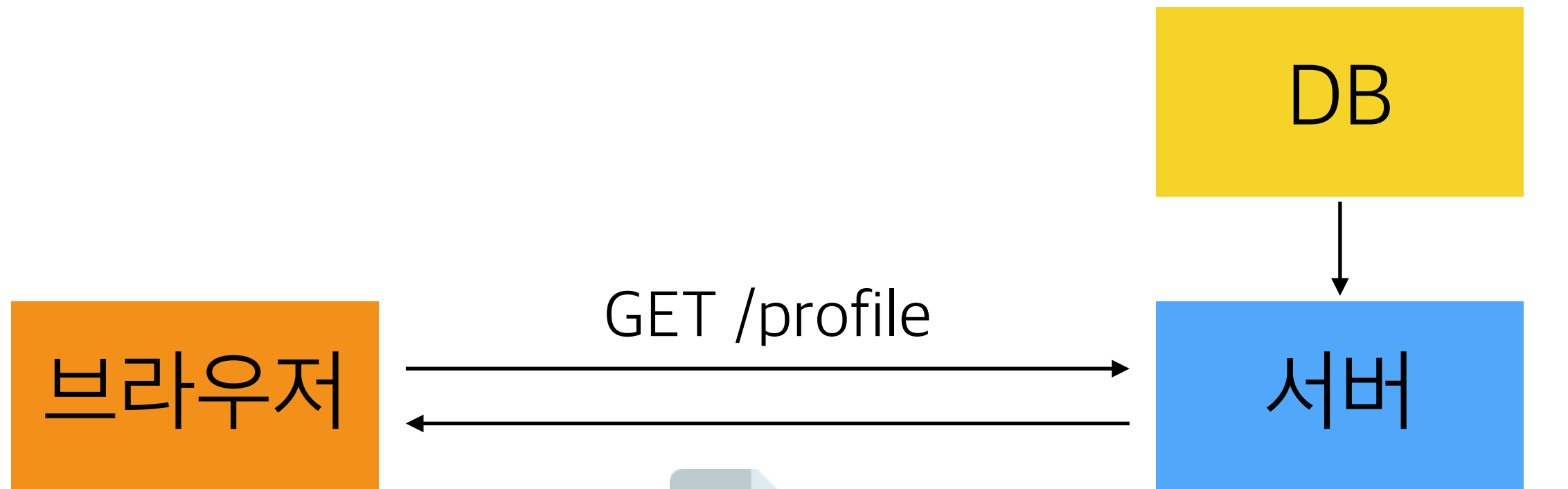
템플릿 언어 Jinja2



<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>
```


웹 서버



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



정적 파일 건네주기

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

실습2

HTML파일 세 개를 만들고,
URL에 따라 각 HTML파일을 응답해봅시다.

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

실습3

profile.html에 프로필사진을 삽입해봅시다.

- 템플릿 언어 (Jinja2)
- 요청 데이터 처리 (Form)
- 아주 간단한 블로그 만들기