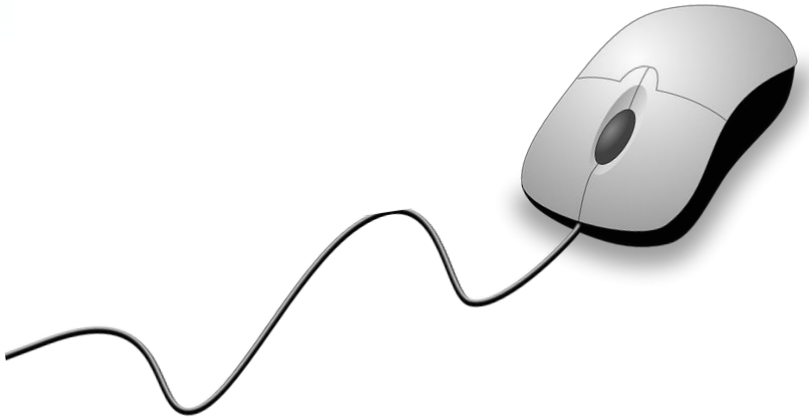
공개SW 솔루션 설치 & 활용 가이드

시스템SW -> SW공학도구

Flask



제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide

OPENUP

오픈소스 소프트웨어 통합지원센터

Open Source Software Support Center

CONTENTS

1. 개요
2. 기능요약
3. 실행환경
4. 설치 및 실행
5. 기능소개
6. FAQ
7. 용어정리

1. 개요

소개	• Python으로 작성된 오픈 소스 웹 프레임워크		
주요기능	• 마이크로 웹 프레임워크로 다양한 웹 엔진과 붙여 사용하기 용이		
대분류	• 시스템 SW	**소분류**	• SW공학도구
라이선스형태	• BSD license	**사전설치 솔루션**	• Python 3
		버전	• 2.2.3(2022년 12월 기준)
특징	• 코드도 비교적 단순하며 API 서버 구축에 편리함 • 다른 프레임 워크에 비해 비교적 확장하기 용이하고 유연함		
개발회사/커뮤니티	• Pallets TEAM		
공식 홈페이지	• https://flask.palletsprojects.com/en/2.2.x/		

1. 개요

- 간결함
  - 플라스크를 이용하면 파일 하나로 구성된 짧은 코드만으로도 완벽하게 동작하는 웹 프로그램 제작 가능

- 높은 자유도
  - Flask는 자유도가 높은 프레임워크임
  - 최소한의 규칙만 있으므로 개발의 자유도가 다른 프레임워크보다 높음

- Jinja template Language 사용 가능
  - Jinja2 문법을 이용해서 동적으로 HTML 코드 작성 가능. 웹페이지가 동적으로 반응해야 될 부분을 JS가 아닌 Python코드와 유사한 방식으로 작성하여 구현

2. 기능요약

• Flask 의 주요 기능

주요기능	기능설명
Debug mode	개발 서버를 시작하는 것 이상의 작업을 수행할 수 있음 디버그 모드를 활성화하면 코드가 변경되면 서버가 자동으로 다시 로드되고 요청 중에 오류가 발생하면 브라우저에 대화형 디버거가 표시됨
Routing	@route 데코레이터를 사용해서 쉽게 함수를 URL에 바인딩 할 수 있음
Rendering Templates	Flask 프레임워크는 Jinja2 템플릿엔진을 기본으로 사용. jinja2 를 사용하면 render_template() 함수를 이용하여 HTML 렌더링 가능

3. 실행환경

- Ubuntu 22.04.1 LTS
- 최신 버전의 Python(Flask는 Python 3.7 이상을 지원
- 코드 에디터 VS Code

4. 설치 및 실행

- 터미널에서 프로젝트 폴더 생성
- 프로젝트 폴더에 가상환경 생성

```
seojh@linuxgeek:~$ mkdir flask_project
seojh@linuxgeek:~$ cd flask_project/
seojh@linuxgeek:~/flask_project$ python3 -m venv venv
```

4. 설치 및 실행

- 가상환경 활성화

```
seojh@linuxgeek:~/flask_project$ . venv/bin/activate
```

- 가상환경에 플라스크 설치

```
(venv) seojh@linuxgeek:~/flask_project$ pip install flask
```

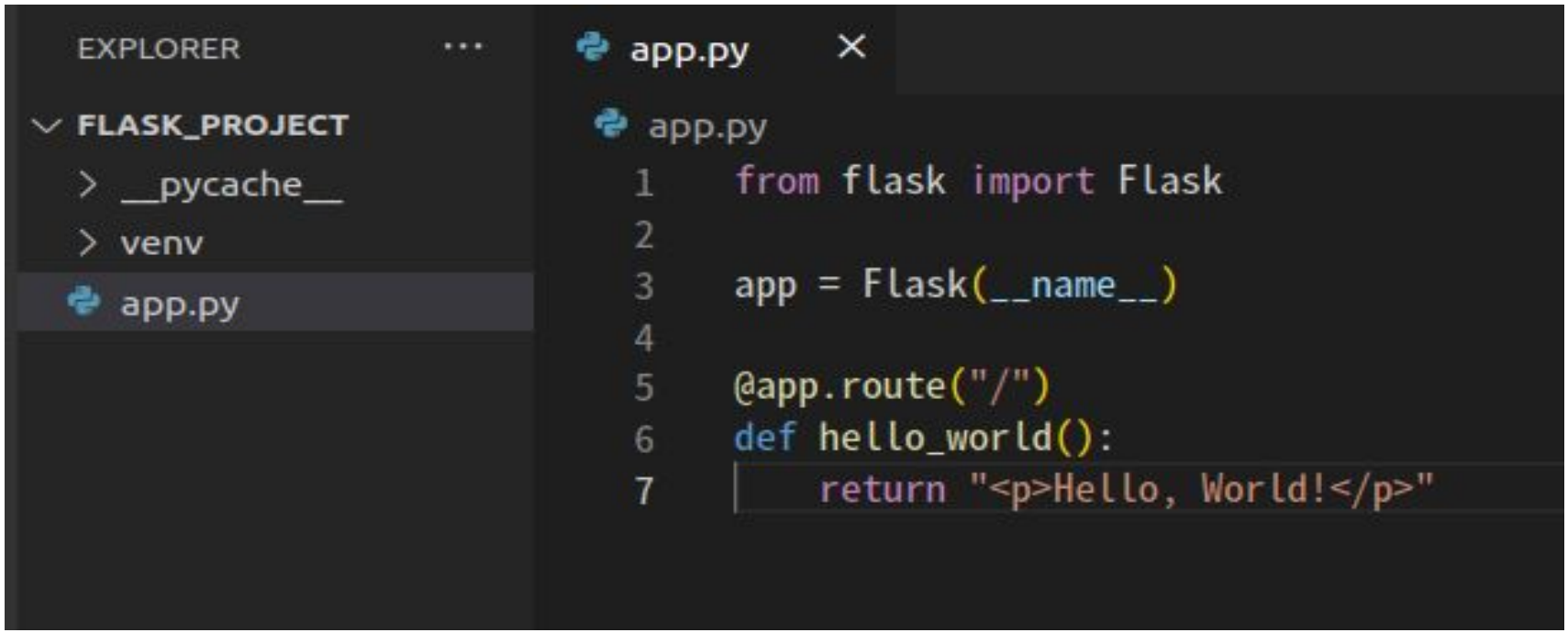
4. 설치 및 실행

- 설치완료

```
(venv) seojh@linuxgeek:~/flask_project$ pip install Flask
Collecting Flask
  Downloading Flask-2.2.2-py3-none-any.whl (101 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 101.5/101.5 KB 2.1 MB/s eta 0:00:00
Collecting itsdangerous>=2.0
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting Werkzeug>=2.2.2
  Downloading Werkzeug-2.2.2-py3-none-any.whl (232 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 232.7/232.7 KB 5.7 MB/s eta 0:00:00
Collecting Jinja2>=3.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 133.1/133.1 KB 7.3 MB/s eta 0:00:00
Collecting click>=8.0
  Using cached click-8.1.3-py3-none-any.whl (96 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, Werkzeug, Jinja2, Flask
Successfully installed Flask-2.2.2 Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 itsdangerous-2.1.2
```

5. 기능소개

- 웹서버 실행하기
  - 프로젝트 폴더 안에 app.py 생성
  - 기본 코드 작성



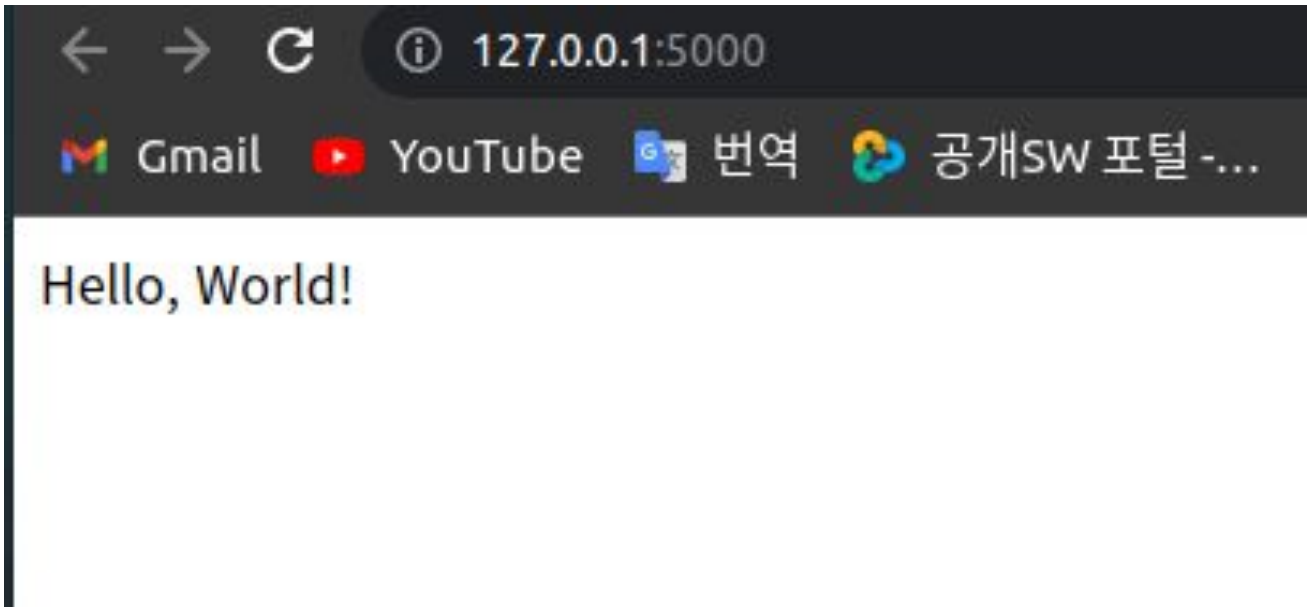
5. 기능소개

- 웹서버 실행하기
  - $flask run 명령어로 서버 실행

```
(venv) seojh@linuxgeek:~/flask_project$ flask run
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

5. 기능소개

- 웹서버 실행하기
  - $flask run 서버 실행 결과



5. 기능소개

- Debug Mode
  - Debug mode를 활성화하면 코드가 변경되면 서버가 자동으로 다시 로드되고 요청 중에 오류가 발생하면 브라우저에 대화형 디버거가 표시됨

```
(venv) seojh@linuxgeek:~/flask_project$ flask --debug run
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 131-363-590
```

5. 기능소개

Flask

- Debug Mode

TypeError

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Traceback (most recent call last)

File "/home/seojh/flask_project/venv/lib/python3.10/site-packages/flask/app.py", line *2548*, in __call__

```
return self.wsgi_app(environ, start_response)
```

File "/home/seojh/flask_project/venv/lib/python3.10/site-packages/flask/app.py", line *2528*, in wsgi_app

```
response = self.handle_exception(e)
```

File "/home/seojh/flask_project/venv/lib/python3.10/site-packages/flask/app.py", line *2525*, in wsgi_app

```
response = self.full_dispatch_request()
```

File "/home/seojh/flask_project/venv/lib/python3.10/site-packages/flask/app.py", line *1822*, in full_dispatch_request

```
rv = self.handle_user_exception(e)
```

File "/home/seojh/flask_project/venv/lib/python3.10/site-packages/flask/app.py", line *1820*, in full_dispatch_request

```
rv = self.dispatch_request()
```

File "/home/seojh/flask_project/venv/lib/python3.10/site-packages/flask/app.py", line *1796*, in dispatch_request

```
return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
```

File "/home/seojh/flask_project/app.py", line *9*, in hello_world

```
d = a + b
```

TypeError: unsupported operand type(s) for +: 'int' and 'str'

5. 기능소개

Flask

- Routing
  - @route 데코레이터를 사용해서 URL을 쉽게 함수에 바인딩 가능

    <text>와 같이 표시해 URL에 변수를 추가 가능

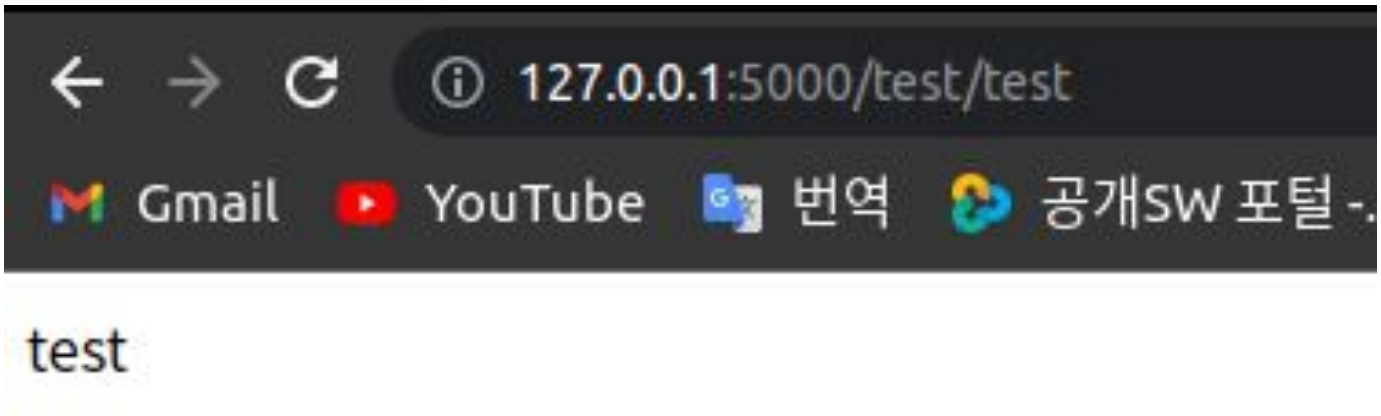
    그런 다음 text를 함수 인수로 받고 사용 가능

```
@app.route("/test/<text>")
def route_sample(text):
    return text
```

5. 기능소개

Flask

- Routing
  - 바인딩된 url로 접속한 결과 변수로 받은 test가 출력된 페이지



5. 기능소개

- templates rendering
  - 웹 서버에서 페이지를 렌더링 해주기 위해서는 render_template()라는 함수를 임포트해서 사용하면 됨

app.py

```
1    from flask import Flask ,render_template
2
```

5. 기능소개

- templates rendering

  – 템플릿 렌더 예제 코드 작성

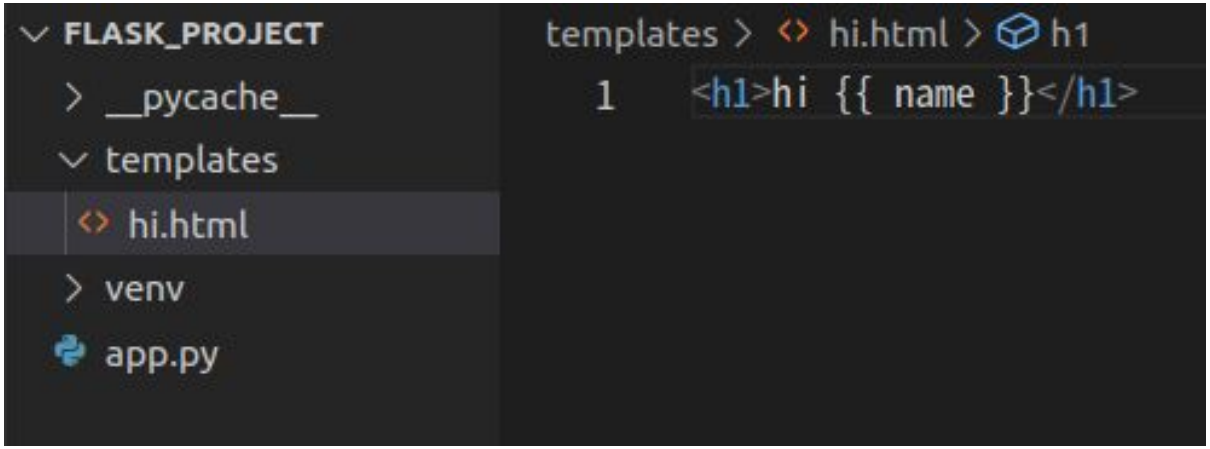
  name변수를 템플릿으로 넘겨줌

```
@app.route("/hi/<name>")
def hi_template_render(name):
    return render_template('hi.html',name=name)
```

5. 기능소개

- templates rendering

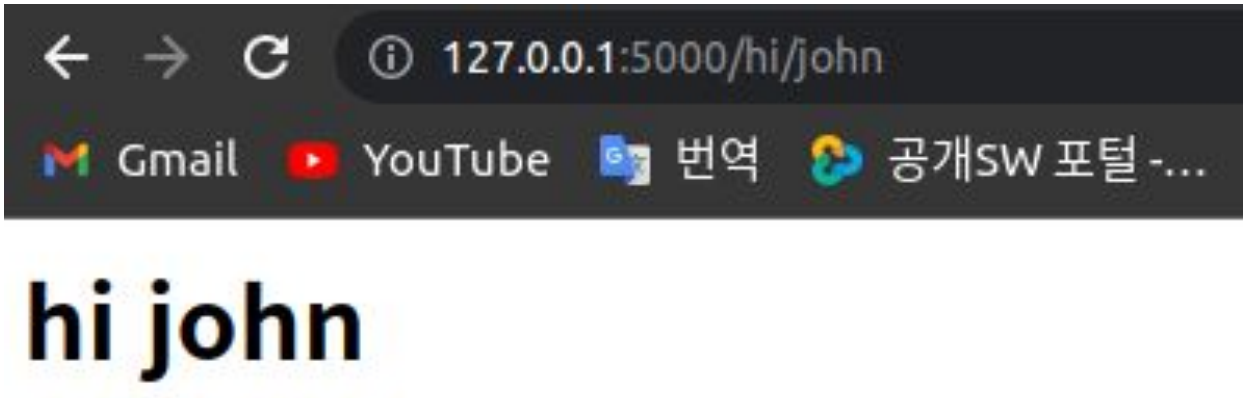
  – 템플릿은 app.py와 같은 위치에 templates 폴더를 생성해야 함
  templates 폴더 안에 html 파일을 생성
  flask는 Jinja2 템플릿의 기능 사용 가능. Jinja2의 문법으로 name 변수를 사용



5. 기능소개

Flask

- templates rendering
  - '/hi/john' URL에 연결된 html 렌더링 결과 URL에 사용한 변수가 같이 출력된 것을 볼 수 있음



6. FAQ

Q flask는 누가 만들었나요?

A 플라스크(Flask)는 2004년 오스트리아의 오픈소스 개발자 아르민 로나허(Armin Ronacher)가 만든 웹 프레임워크입니다. 플라스크는 아르민 로나허가 만우절에 장난삼아 던진 아이디어였는데 사람들의 관심이 높아져 서비스로 만들어졌다고 합니다.

Q flask가 자유도가 높다는 말은 무슨 뜻인가요?

A Flask는 Architecture도 본인이 원하는 방향에 따라 구성할 수 있게끔 되어 있고, 제공 모듈이 비교적 적기에 python의 설치형 모듈을 많이 활용하기 때문에 자기가 원하는 기능을 자유롭게 추가 수정 할 수 있습니다.

6. FAQ

Q django 와 flask 중 어느때 flask를 써야 하나요?

A 앱의 크기가 작거나, DB가 이미 생성된 NoSQL을 사용하고 있는 경우, 기본 지원 기능이 거의 없기 때문에 처음부터 모든 기능을 만들수 있는 경우 쓰는것이 좋습니다.

Q flask를 사용하는 대표적인 서비스는 어떤게 있나요?

A Linked-in, Pinterest, Netflix 가 대표적인 서비스 입니다.

7. 용어정리

용어	설 명
Rendering	마크업 언어를 브라우저(Client)에 보여주기 위한 과정
Debug	컴퓨터 프로그램 개발 단계 중에 발생하는 시스템의 논리적인 오류나 비정상적 연산(버그)을 찾아내고 그 원인을 밝히고 수정하는 작업 과정
URL	URL(Uniform Resource Locator)이란 사용자가 원하는 정보의 위치와 종류를 파악할 수 있도록 웹페이지의 정보 구조를 반영한 것
Jinja2	Python용 현대적이고 디자이너 친화적인 템플릿 언어

Open Source Software Installation & Application Guide

이 저작물은 크리에이티브 커먼즈 [저작자표시 – 비영리 – 동일조건 변경허락 2.0 대한민국 라이선스]에 따라 이용하실 수 있습니다.