

# Flask 실습 자료 (1)

- 1. Flask 란
  - 2. Flask 설치
  - 3. Hello World! with Flask
  - 4. Routing
  - 5. POST method
  - 6. ETC
- 

## 1. Flask 란

The python micro framework for building web applications  
- <https://github.com/pallets/flask>

- Micro Service Architecture (MSA) 를 위한 web app framework
  - web app framework (web framework) : 웹 서비스 개발을 위한 프레임워크
- Django 등 다른 framework 에 비해 굉장히 가벼우며, 확장성, 유연성이 뛰어난 프레임워크
  - 단, 자체 지원 기능은 적다는게 장점이자 단점
- 사용하기 쉽고, 간단한 기능을 가볍게 구현하기에 적합하기 때문에 대부분의 ML Model 의 첫 배포 Step 으로 Flask 를 자주 사용합니다.

---

## 2. Flask 설치

- Prerequisite
  - Python 가상환경
    - 3.6 이상
      - 3.8.9 사용
    - pip3
- How to Install

```
# 새로운 디렉토리를 하나 생성한 뒤, 이동해주세요
mkdir flask-tutorial
cd flask-tutorial

# python 버전 확인
python -V

# Flask 설치
pip install -U Flask==2.0.2

# Flask Version 확인
flask --version

# Python 3.8.9
# Flask 2.0.2
# Werkzeug 2.0.2
```

### 3. Hello World! with Flask

- Flask 를 사용해 여러분의 컴퓨터에 간단한 Web Server 를 띄워보겠습니다.
- `app.py` 파일을 생성하여 다음 코드를 복사 후 붙여넣습니다.

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

if __name__ == "__main__":
    app.run(debug=True, host='0.0.0.0', port=5000)
# debug 모드로 실행, 모든 IP 에서 접근 허용, 5000 포트로 사용하는 것을 의미
```

- 동일한 폴더에서 `python app.py` 을 수행하여 application server 를 로컬에 띄웁니다.
- `127.0.0.1:5000` 으로 접속하면, `Hello, World!` 라는 문자가 브라우저에 보이는 것을 확인할 수 있습니다.

### 4. Routing

- flask 의 `route()` 데코레이터는 python 함수를 web server 의 URI 에 mapping 시킬 수 있습니다.
  - 이를 활용하여 여러분이 원하는데로 Routing 을 수행할 수 있습니다.
- 간단한 예를 들면 다음과 같습니다.

- `app.py` 파일을 다음과 같이 수정합니다.

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

@app.route("/fastcampus")
def hello_fastcampus():
    return "<p>Hello, Fast Campus!</p>"

if __name__ == "__main__":
    app.run(debug=True, host='0.0.0.0', port=5000)
```

- 동일한 폴더에서 `python app.py` 을 수행하여 application server 를 로컬에 띄웁니다.
- `127.0.0.1:5000` 으로 접속하면 `Hello, World!` 라는 문자가, `127.0.0.1:5000/fastcampus` 로 접속하면 `Hello, Fast Campus!` 라는 문자가 브라우저에 보이는 것을 확인할 수 있습니다.

## 5. POST method

- flask 의 `route()` 데코레이터는 URI 뿐만 아니라 HTTP Method 도 지정할 수 있습니다.
  - 이를 활용하여 여러분이 원하대로 API 를 만들 수 있습니다.
- 간단한 예를 들면 다음과 같습니다.
- `app.py` 를 다음과 같이 수정합니다.

```
from flask import Flask
import json

app = Flask(__name__)

@app.route("/predict", methods=["POST", "PUT"])
def inference():
    return json.dumps({'hello': 'world'}), 200 # http status code 를 200 으로 반환하는 것을 의미합니다.

if __name__ == "__main__":
    app.run(debug=True, host='0.0.0.0', port=5000)
```

- 동일한 폴더에서 `python app.py` 을 수행하여 application server 를 로컬에 띄웁니다.

- curl 을 수행하여 HTTP 응답을 확인합니다.

```
curl -X POST http://127.0.0.1:5000/predict
# {"hello": "world"}

curl -X PUT http://127.0.0.1:5000/predict
# {"hello": "world"}

curl -X GET http://127.0.0.1:5000/predict
# <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
# <title>405 Method Not Allowed</title>
# <h1>Method Not Allowed</h1>
# <p>The method is not allowed for the requested URL.</p>
```

- POST, PUT 만 허용하였으므로 POST, PUT 에 대한 응답은 정상적으로 오지만, GET 에 대한 응답은 405 ERROR 가 반환됩니다.

---

## 6. ETC

- 이외에도 Flask 는 Web Application 을 구동하기 위한 다양한 기능들을 내장하고 있습니다.
- 하지만 저희는 Flask 의 모든 기능을 살펴보는 것이 목적이 아니라, 여러분의 머신러닝 모델을 API 서비스로 제공(서빙)할 때, Flask 를 사용하는 방법을 알아보는 것이 목적입니다.
- 다음 시간에는 Flask 를 활용하여 간단한 머신러닝 모델을 서빙해보겠습니다.