

- 과제 목표 (도출해야 할 결과)
 1. Flask를 이용한 GET / POST 요청 처리
 2. 이를 Docker 이미지로 만들어서 운영해보기
- 코드 설명과 과제 해결 방법

```
@app.route('/')
def hello_world():
    return 'Hello World!'
```

```
@app.route('/test_get')
def test_get():
    return render_template('get.html')

@app.route('/test_post')
def test_post():
    return render_template('post.html')
```

- ➔ @app.route 데코레이터는 URL과 사용자 함수를 연결해줌
- ➔ render_template(): template 폴더 내의 웹페이지 파일을 가져옴

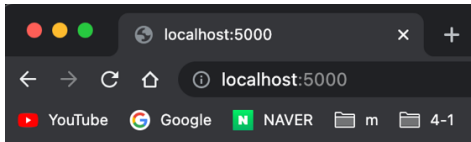
```
@app.route('/get', methods=['GET'])
def get():
    value = request.args.get('test')
    return value

@app.route('/post', methods=['POST'])
def post():
    value = request.form['test']
    return value
```

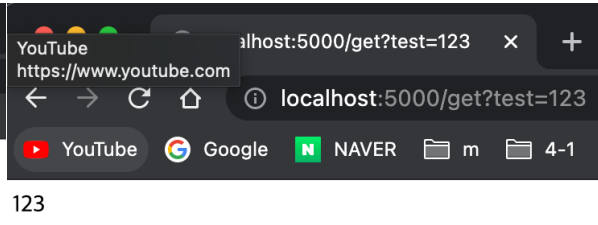
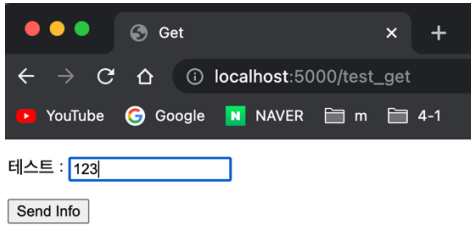
- ➔ GET 또는 POST 요청을 받았을 때에만, URL과 사용자 함수 각각 연결
- ➔ GET 일 경우 :request의 args.get으로 name이 test인 것을 가져옴
- ➔ POST 일 경우 :request의 form에서 name이 test인 것을 가져옴

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

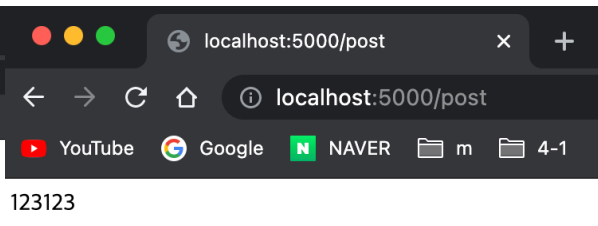
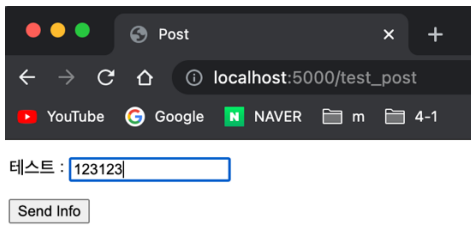
- ➔ host 를 0.0.0.0 으로 해야 모든 호스트에서의 접속 허용함



Hello World!



123



123123

→ 루트디렉토리 / GET / POST

● Dockerizing 결과 사진

```

> docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
flask-tutorial      latest     c48f931aacd2  48 minutes ago  373MB
ubuntu              latest     59ab609fab8d  2 weeks ago   65.6MB
kano@mitra ~$ docker run --rm -p 5000:5000 -d flask-tutorial
62c1174ec1ec2ab1fbecb66ae61ea198f7bc65cb7ecffcb9dd62b69d443d4da2
kano@mitra ~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
62c1174ec1ec   flask-tutorial  "python3 app.py"        4 seconds ago Up 3 seconds  0.0.0.0:5000->5000/tcp, :::5000->5000/tcp  brave_meninsky

```