

Flask - 파일 업로드

업로드 폼 만들기

파일을 업로드 하기 위해서 먼저 파일을 전송하기 위한 form을 html에 페이지에 만들어 보자.

templates\upload.html

```
<html>
  <body>
    <form action="http://localhost:5000/uploader" method="POST"
    enctype="multipart/form-data"> # multipart/form-data : 품안의 데이터를 쪼개서
      <input type="file" name="file" />
      <input type="submit" />
    </form>
  </body>
</html>
```

Flask가 성공적으로 폼으로 가져온 파일을 읽어내기 위해서는 enctype 속성에 반드시 **multipart/form-data**로 되어야 한다.

아래는 파이썬 코드

routes.py

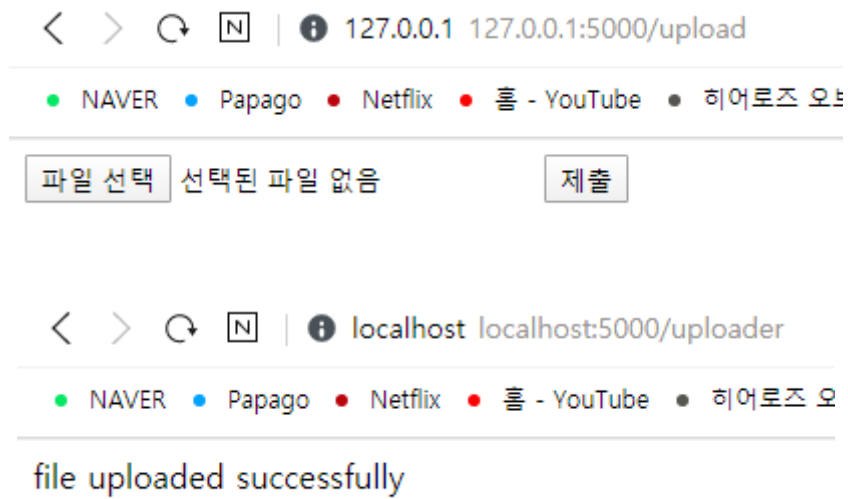
```
from flask import Flask, render_template, request
from werkzeug.utils import secure_filename # 파일을 보호해주기 위한 모듈

app = Flask(__name__)

@app.route('/upload')
def load_file():
    return render_template('upload.html')

@app.route('/uploader', methods = ['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        f = request.files['file']
        f.save(secure_filename(f.filename))
        return 'file uploaded successfully'

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



f = request.files['file']

일반적으로 폼에서 데이터를 가져올 때와, 비슷한 방식이다. 위 코드는 request 객체의 'file'이라는 이름의 폼으로 전송된 파일에 해당한다.

secure_filename(f.filename)

해당 파일명을 보호하기 위한 메소드이다. 해당 파일이 실제 시스템에 저장되기 전에 파일명을 보호하기 위한 함수이며, 항상 사용해 주는것이 좋다.

f.save()

말 그대로 해당 파일 객체를 저장하는 메소드이다. 인자로써 해당 파일명을 포함한 경로를 입력해주면 된다. 예를 들어 그냥 파일명만 인자로 주면 routes.py와 같은 경로에 저장되게 되지만, 인자를 이런 느낌으로 주면 원하는 폴더에 저장할 수 있다.

```
f.save("./img/" + secure_filename(f.filename))
```