Python Flask Tutorial: Full-Featured Web App Part 5

Flask

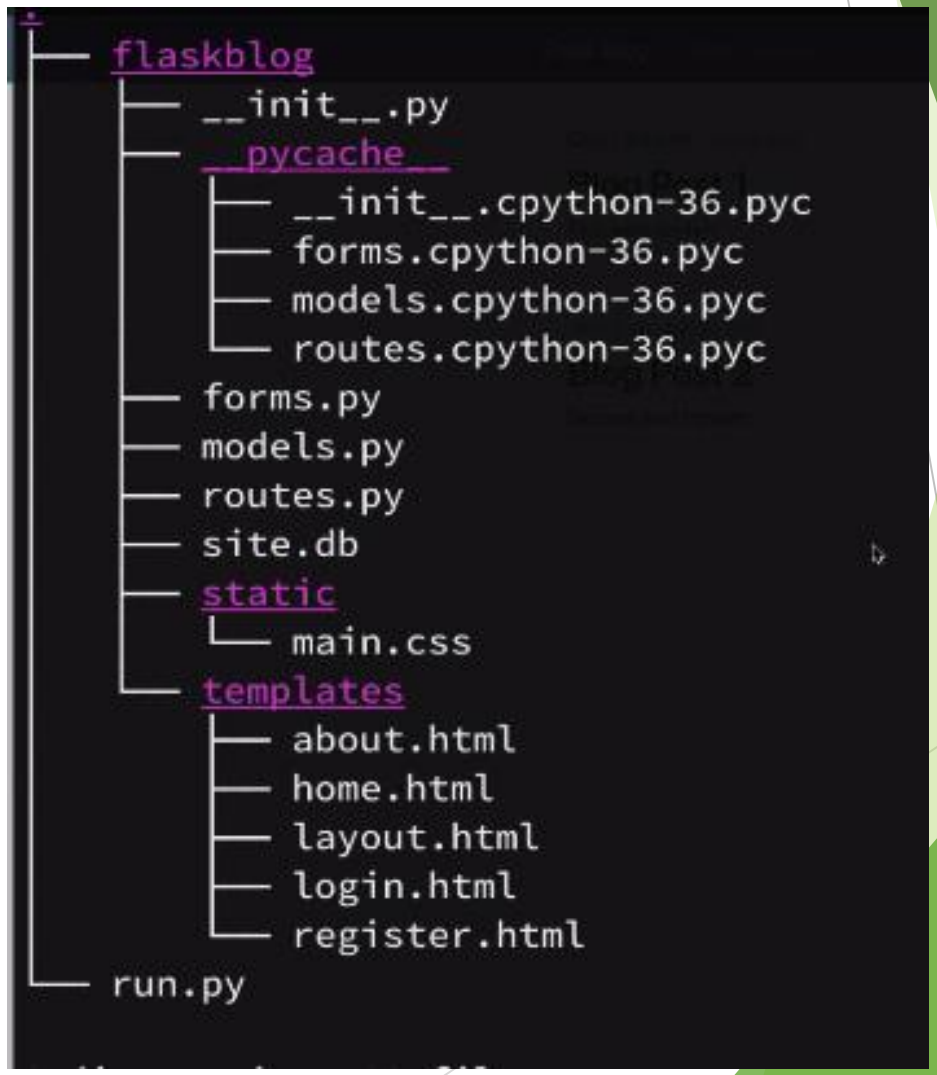
목차

- Package Structure

5. Package Structure

Flask 폴더
- templates
  - about.html
  - home.html
  - layout.html
  - login.html
  - register.html
- flaskblog.py
- forms.py

static 폴더
- main.css



5. Package Structure

Package structure

```
/yourapplication
    /yourapplication.py
    /static
        /style.css
    /templates
        layout.html
        index.html
        login.html
        ...
```

```
/yourapplication
    /yourapplication
        /__init__.py
        /static
            /style.css
        /templates
            layout.html
            index.html
            login.html
            ...
```

```
from yourapplication import app
app.run(debug=True)
```

- Flask app은 하나의 파일만으로 충분할 수 있다.
- 하지만 큰 프로젝트의 경우 모든 코드를 파일 하나에 넣기에는 불편.
- 그래서 코드 및 각종 기능들을 짜임새있게 관리하기 위해 package를 활용

- 작은 application을 큰 application 구조로 변환하기 위해서, 단지 기존에 존재하던 폴더에 새 폴더 yourapplication 을 생성하고 이 폴더로 모두 옮긴다.
- 기존 yourapplication.py를 __init__.py로 이름을 바꾼다.(flaskblog.py -> __init__.py)

- Runserver.py 새로운 파일을 루트 폴더 바로 아래 있는 yourapplication 폴더 안에 추가하면 된다.

- 그리고 from yourapplication import app
- Import만 해주면 된다.

5. Package Structure

Package structure

__init__.py

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

import yourapplication.views
```

views.py

```
from yourapplication import app

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

```
/yourapplication
    /runserver.py
    /yourapplication
        /__init__.py
        /views.py
        /static
            /style.css
        /templates
            layout.html
            index.html
            login.html
            ...
```

- 이렇게 바꾸면 좋은점은?
- 어플리케이션을 복개의 모듈들로 재구조화 가능
- Flask 어플리케이션 객체 생성은 __init__ 파일에서
- 모든 뷰 함수들은 함수의 선언부 위에 route() 데코레이터를 가진 함수는 __init__.py 파일에 import 되어야 한다.
- 객체가 아닌 함수가 있는 모듈을 import 해야함
- 어플리케이션 객체를 생성한 후에 뷰 모듈을 import

- 최종구조

- 순환 임포트(Circular Imports) :
- 모든 파이썬 프로그래머는 순환 임포트를 싫어하지만 flask 는 일부 사용한다.
- 순환 임포트는 두 모듈일 서로 의존 관계가 있는 경우인데 views.py 는 __init__.py 에 의존한다.
- Be advised that this is a bad idea in general but here it is actually fine.
- 왜냐하면 __init__.py 에 있는 뷰들은 실제로 사용하지 않고 단지 모듈들이 임포트되었는지 보장하고 그 파일의 제일 하단에서 임포트하기 때문