

# 웹 서버 심화

노기섭 교수  
[\(kafa46@hongik.ac.kr\)](mailto:kafa46@hongik.ac.kr)

# 학습 목표

## ■ 웹 서버 심화

- Flask 기반 웹 서버를 MTV 패턴으로 구성하고 역할을 이해
- 환경 설정(config.py)과 앱 초기화(server/\_\_init\_\_.py) 흐름 파악
- SQLAlchemy 모델 정의와 폼 제출 → DB 저장 라우트 구현 연습
- Jinja 템플릿으로 입력 폼을 작성하고 렌더링하는 방법 학습
- run\_server.py로 애플리케이션을 실행해 동작을 확인

# MTV 패턴 간단히 알아보기 (1/2)

## ■ MTV 패턴

- MTV 패턴은 웹 애플리케이션을 구조적으로 나누어 개발하기 위한 아키텍처 패턴
- Model, Template, View 세 가지 역할로 구성
- 일반적인 MVC(Model–View–Controller) 패턴과 유사하지만,  
웹 프레임워크에서는 MTV라는 용어를 자주 사용한다.

## ■ Model (모델)

- 데이터와 관련된 모든 것을 담당한다.
- 데이터베이스 테이블 구조를 정의한다.
- 레코드를 생성, 조회, 수정, 삭제(CRUD)하는 로직을 포함한다.
- 예:
  - User, Post, Product 등 도메인 객체
  - Django의 models.py, Flask + SQLAlchemy의 모델 클래스

# MTV 패턴 간단히 알아보기 (2/2)

## ■ Template (템플릿)

- 사용자에게 보여지는 화면(View, UI)을 담당한다.
- HTML 기반의 템플릿 파일로, 동적으로 데이터를 표시한다.
- 템플릿 엔진(예: Jinja2, Django Template Language)을 사용하여 변수 출력, 반복, 조건문 등을 처리
- 예:
  - templates/index.html
  - templates/user\_detail.html

## ■ View (뷰)

- 사용자의 요청(Request)을 받고, 응답(Response)을 생성하는 역할을 한다.
- 비즈니스 로직의 진입 지점으로, 다음과 같은 작업을 수행한다.
- 요청 파라미터 처리
- 필요한 Model 호출 (데이터 조회/저장 등)
- 적절한 Template에 데이터 전달
- 예:
  - Flask의 라우트 함수(@app.route 또는 Blueprint의 뷰 함수)

# HTML - 개념 및 문서 구조

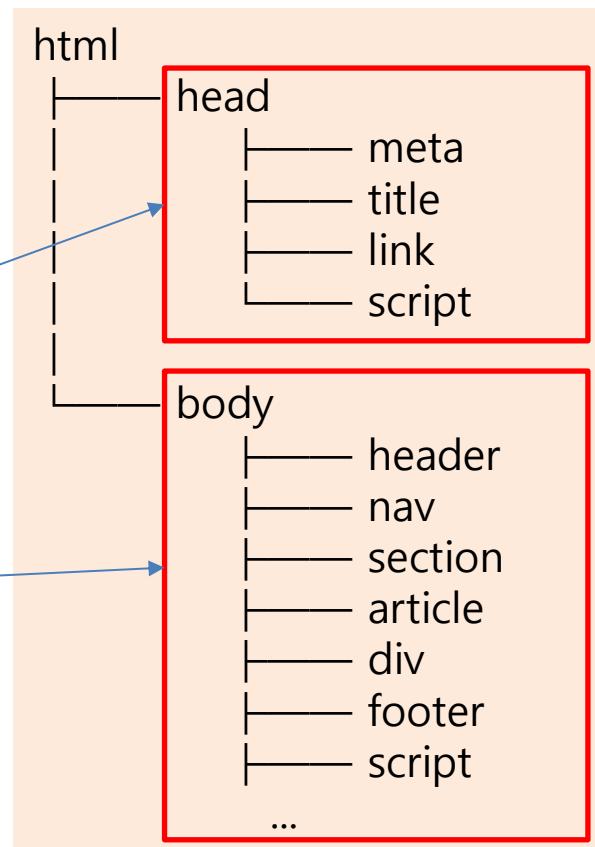
## ■ HTML

- 태그(tag)를 이용해 문서 구조를 표현하는 마크업 언어
- HTML 문서는 트리 구조(Tree Structure)로 구성되며, 가장 상위 루트는 <html> 태그이다.

## ■ HTML 문서 구조

```
<!DOCTYPE html>
<html>
  <head>
    ...
  </head>

  <body>
    ...
  </body>
</html>
```



태그	설명
<meta>	문자 인코딩, SEO, 모바일 설정 등
<title>	웹페이지 제목 (브라우저 탭에 표시됨)
<link>	CSS 파일 연결
<script>	JavaScript 연결
<style>	내부 CSS 작성

# HTML - body 영역에서 자주 사용하는 태그

태그	의미
<h1> ~ <h6>	제목(Heading)
<p>	문단(paragraph)
<a>	링크(anchor)
<img>	이미지
<ul> / <ol> / <li>	리스트
<div>	블록 단위 영역
<span>	인라인 영역
<form>	사용자 입력 양식
<input>	데이터 입력 필드
<label>	입력 필드 설명 텍스트, for-id 연결로 포커스 이동·접근성 향상
<button>	버튼
<table>	표 구조

# 실전 예제

실습 파일 ([.zip](#)) 다운로드:

[https://www.deepshark.org/courses/python\\_application/w/  
14\\_web\\_server\\_advanced#1](https://www.deepshark.org/courses/python_application/w/14_web_server_advanced#1)

# 가상환경 설정 및 패키지 설치

## ■ 가상환경 설치

```
python -m venv myenv  
source myenv/bin/activate    # macOS 및 Linux  
myenv\Scripts\activate      # Windows
```

## ■ 패키지 설치

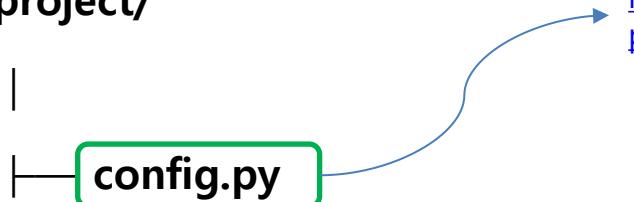
```
python -m pip install --upgrade pip  
pip install Flask Flask-SQLAlchemy
```

# 프로젝트 구조

```
project/
|
├── config.py          # 시스템 설정
├── server/
│   ├── __init__.py    # 서버 초기화
│   ├── models.py       # DB 정의/설정
│   ├── views.py        # 라우트 정의
│   └── templates/
│       └── index.html # 랜딩 페이지
└── run_server.py      # 서버 구동(시작)
```

# 웹서버 구현 - 시스템 설정

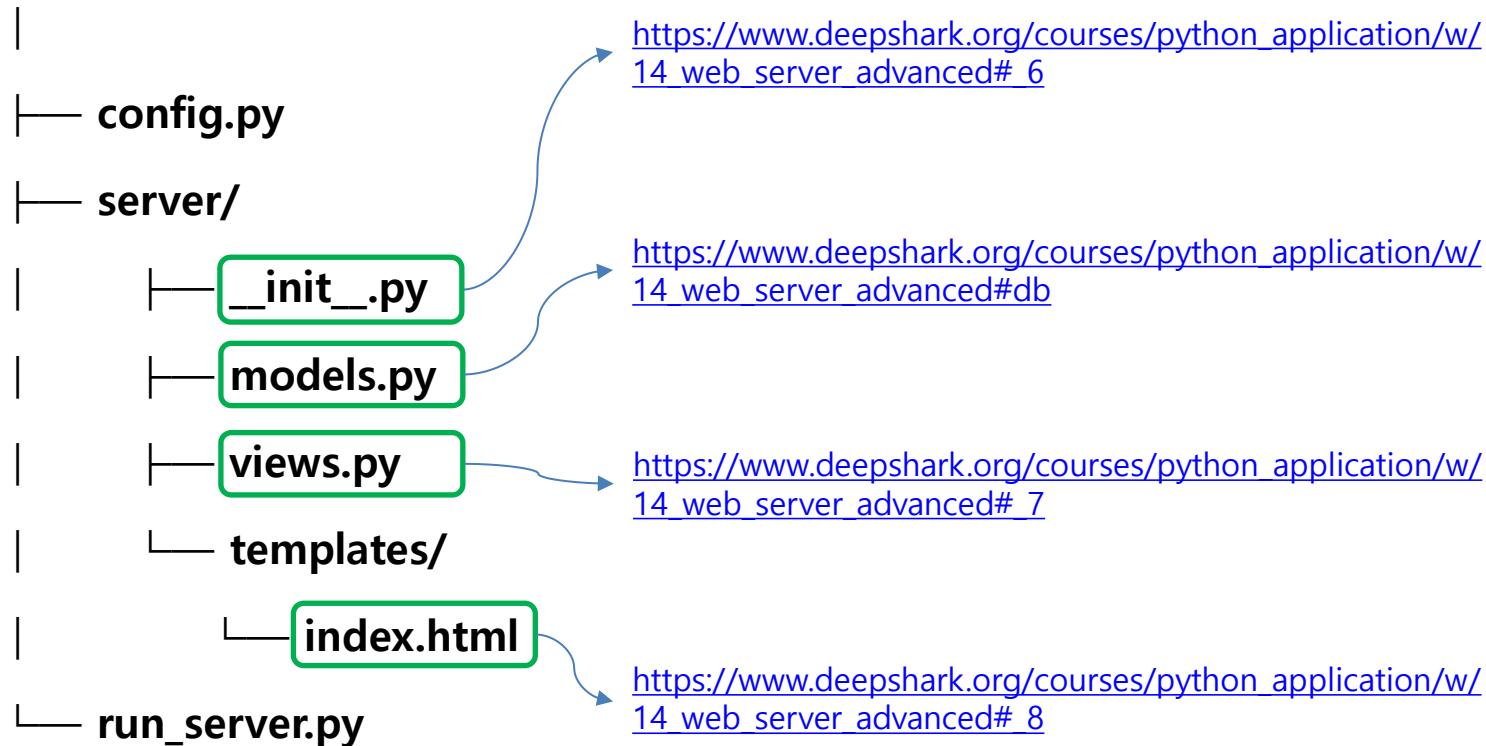
```
project/
|   config.py
+-- server/
|   |   __init__.py
|   |   models.py
|   |   views.py
|   |   templates/
|   |       index.html
+-- run_server.py
```



[https://www.deepshark.org/courses/python\\_application/w/14\\_web\\_server\\_advanced# 5](https://www.deepshark.org/courses/python_application/w/14_web_server_advanced# 5)

# 웹서버 구현 - 서버 코딩

project/



# 웹서버 구현 - 서버 구동 모듈

project/

|

  |  
  └── config.py

  |  
  └── server/

    |  
    └── \_\_init\_\_.py

    |  
    └── models.py

    |  
    └── views.py

    |  
    └── templates/

      |  
      └── index.html

  └── run\_server.py

[https://www.deepshark.org/courses/python\\_application/w/  
14\\_web\\_server\\_advanced# 9](https://www.deepshark.org/courses/python_application/w_14_web_server_advanced# 9)

# 웹 서버 실행

```
python project/run_server.py
```

# 또는

```
cd project
```

```
python run_server.py
```



수고하셨습니다 ..^^..