

# Django 기반의 웹프로그래밍

이 소 영

yisy0703@naver.com

Web Frameworks for Python(<https://wiki.python.org/moin/WebFrameworks>)

장고 공식 사이트(<https://www.djangoproject.com/>)

장고 공식 소스 저장소(<http://github.com/django/django>)

장고 참조 문서 영어(<https://docs.djangoproject.com/en/5.1/>)

# Agenda

## Django(장고) 기반의 파이썬 웹 프로그래밍

### Ch01. Django 시작하기

1. Django 란?
2. 개발 환경 구축
3. Django 구조

### Ch02. Django App

1. Django Project
2. Model
3. View

### Ch03. Model

1. Model 속성 및 옵션
2. Relationship
3. Migrations
4. Admin App

### Ch04. Django SQL

1. Django shell
2. Manager & QuerySet
3. 조회 SQL
4. 생성/수정/삭제 SQL
5. Django-Debug-Toolbar

### Ch05. Template

1. Template Loader
2. URL Dispatcher
3. Template 상속
4. Template Engines
5. Template Filter

### Ch06. Django View

1. View 기본
2. View 활용

### Ch07. Django Form

1. HTML form
2. CSRF
3. HttpRequest/HttpResponse
4. Django Form
5. Django Model Form
6. Form Validation

### Ch08. File 관리

1. Static Files
2. Media Files
3. Image Thumbnail

### Ch09. 사용자 인증

1. Auth App
2. 회원가입 구현
3. 로그인/아웃 구현
4. Oauth 라이브러리 활용

# Ch 01 Django 시작하기

---

1. Django란?
2. 개발환경구축
3. Django 구조

## Chapter 01. Django 시작하기

- Web Frameworks for Python(<https://wiki.python.org/moin/WebFrameworks>)
- 장고 공식 사이트(<https://www.djangoproject.com/>)
- **장고 공식 소스 저장소(<http://github.com/django/django>)**
- 장고 참조 문서 영어(<https://docs.djangoproject.com/en/5.2/>)
- 장고 참조 문서 한글(<https://docs.djangoproject.com/ko/5.2/intro/>)

## 개발 환경

- Python이나 Anaconda Python 설치(<https://www.anaconda.com/download/>)
- Visual Studio Code 설치(<https://code.visualstudio.com/>) 냐 PyCharm 설치(<https://www.jetbrains.com/ko-kr/pycharm/download/>)



---

# 1. Django 란?

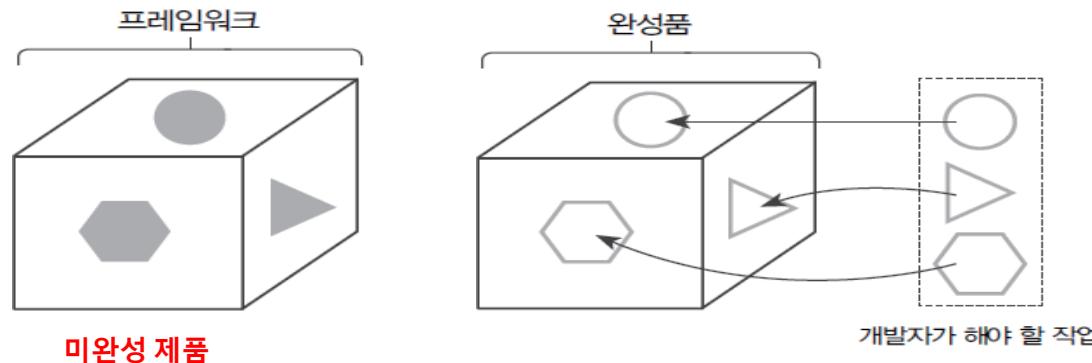


# 프레임워크

1. Django 란?

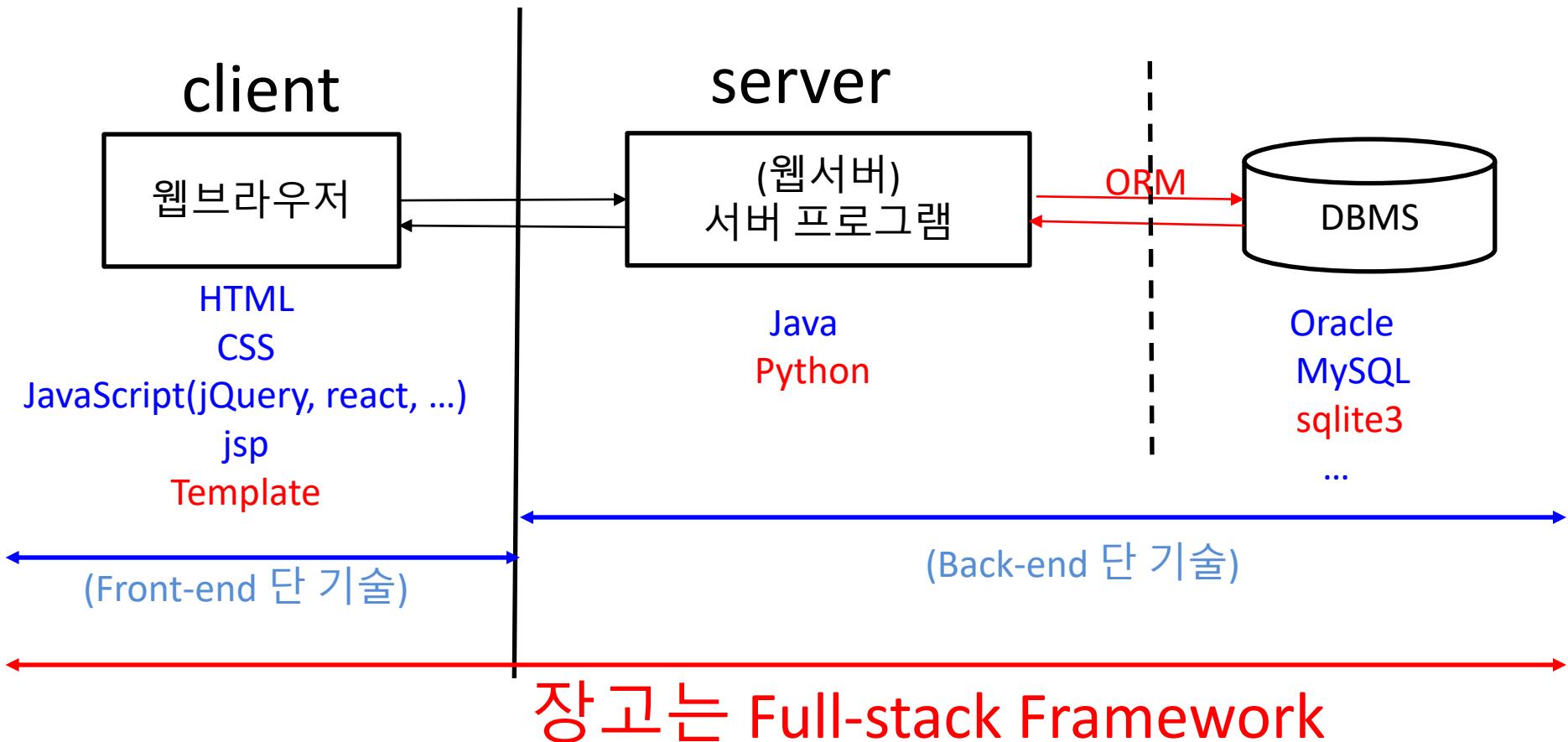
- 프레임워크와 라이브러리 차이

	프레임워크	라이브러리
코드작성	프레임워크 클래스를 서브 클래싱해서 작성	독립적으로 작성
호출흐름	프레임워크 코드가 유저 코드를 호출	유저코드가 라이브러리를 호출
실행흐름	프레임워크가 제어	유저 코드가 제어
객체연동	구조 프레임워크가 정의	독자적으로 정의



# Web Framework for Python (<https://wiki.python.org/moin/WebFrameworks>)

## 1. Django 란?



# 웹 서비스 개발 파트

## 1. Django 란?

---

- 백엔드 개발

- Django, FastAPI, Flask, Spring(Java), Node.js 등

- 서버운영

- 자체 서버, 서버/웹 호스팅
  - AWS/Azure/Google 클라우드(IaaS와 PaaS) 등

- 프론트엔드 개발

- HTML, CSS, JavaScript 기반 라이브러리 및 프레임워크
  - Vue.js, React.js, Angular.js, jQuery, Bootstrap 등

- 앱 개발

- Android, iOS 개발

# Python web Framework

## 1. Django 란?

---

- <https://wiki.python.org/moin/WebFrameworks>
- Popular **Full-stack Frameworks**
  - Django
    - The Web Framework for perfectionists with deadlines.
- Popular micro Frameworks
  - Flask
    - A microframework for Python based on Werkzeug.
  - FastAPI
    - A modern, fast (high-performance).

# Django 구성요소

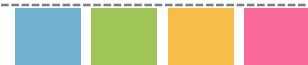
Client - web Server - DB  
Full stack

## 1. Django 란?

- Views : HTTP 요청처리 (server단 기술)
- Models : 데이터 베이스 처리 (DB 단 기술)
- Template : 사용자 인터페이스 처리 (client 측 기술)
- Form : 사용자 입력 데이터 처리 (client 측 기술)
- Admin App : 쉬운 DB 레코드 관리 UI (관리자 기능 제공)
- Loggin : 다양한 경로로 메시지 로깅
- Static files : 정적 파일 관리 (서버측에서 사용하는 파일)
- Media files : 사용자가 업로드한 파일 관리 (서버측에서 사용하는 파일)
- Message framework : 1회성 메시지 노출
- 테스팅, 국제화 & 지역화, 캐싱, Send Emails
- Geographic, Syndication Feeds(RSS/Atom), Sitemaps



## 2. 개발 환경 구축



# 파이썬 설치

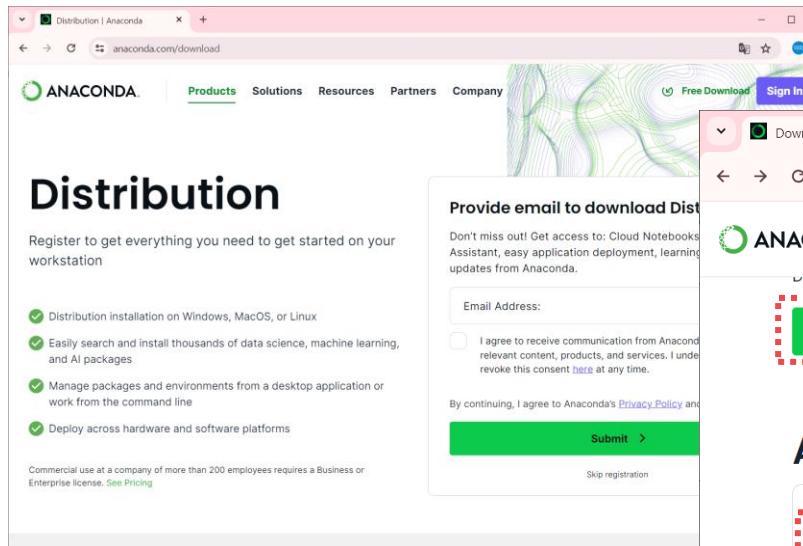
## 2. 개발 환경 구축

- 공식 배포판
  - <http://python.org>
- Anaconda Python
  - <https://www.anaconda.com>
  - 설치 시 환경변수 PATH 옵션 체크(?)
- 실제 웹서비스에서는 운영체제 배포판에서 제공하는 파이썬 사용

# Anaconda Python 설치

## 2. 개발 환경 구축

- <https://www.anaconda.com>



Anaconda3-2024.02-1-Windows-x86\_64.exe 다운로드

The top part of the image shows a browser window titled 'Download Now | Anaconda' with a green 'Download' button highlighted by a red dashed box. The bottom part shows the 'Anaconda Installers' page with three sections: 'Windows' (Python 3.11), 'Mac' (Python 3.11), and 'Linux' (Python 3.11). Each section lists different installer options with their file sizes.

Platform	Python Version	Installer Type	File Size
Windows	Python 3.11	64-Bit Graphical Installer	904.4M
		64-Bit Command Line Installer	731.2M
		64-Bit (M1) Graphical Installer	697.4M
		64-Bit (M1) Command Line Installer	700 M
		64-Bit (x86) Installer	997.2M
Mac	64-Bit Graphical Installer	728.7M	
	64-Bit Command Line Installer	731.2M	
	64-bit (Linux on IBM Z & LinuxONE) Installer	91.8M	
Linux	64-Bit (AWS Graviton2 / ARM64) Installer	798.5M	
	64-bit (Linux on IBM Z & LinuxONE) Installer	91.8M	

# Anaconda Python 설치

## 2. 개발 환경 구축

The image shows the Anaconda Python 2024.06-1 (64-bit) Setup process across six windows:

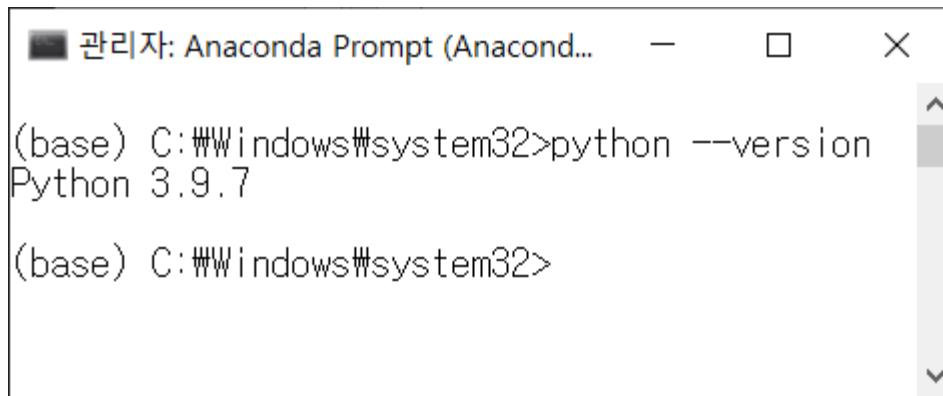
- Welcome to Anaconda3 2024.06-1 (64-bit) Setup**: A welcome screen with a brief introduction and a note to close other applications. Buttons: < Back, Next >, Cancel.
- License Agreement**: The license terms for Anaconda. It includes a scrollable "ANACONDA TERMS OF SERVICE" section, a note to press Page Down, and an "I Agree" button. Buttons: < Back, I Agree, Cancel.
- Select Installation Type**: A screen to choose the installation type. It lists "Just Me (recommended)" and "All Users (requires admin privileges)". Buttons: < Back, Next >, Cancel.
- Choose Install Location**: A screen to select the installation folder. It shows the current path "C:\Users\WYI\Anaconda3" in a "Destination Folder" input field, with a "Browse..." button. It also shows space requirements: "Space required: 5.0 GB" and "Space available: 380.2 GB". Buttons: < Back, Next >, Cancel.
- Advanced Installation Options**: A screen to customize Anaconda3 integration with Windows. It includes checkboxes for creating start menu shortcuts (selected), adding to PATH (selected), registering as default Python (selected), and clearing package cache (unchecked). A note says this is NOT recommended. Buttons: < Back, Install, Cancel.
- Installation Complete**: A confirmation screen stating "Setup was completed successfully." with a "Show details" link. Buttons: < Back, Next >, Cancel.

Red boxes highlight the "Next >" buttons in the first three windows, the "I Agree" button in the second window, the "Install" button in the fifth window, and the "Next >" button in the sixth window.

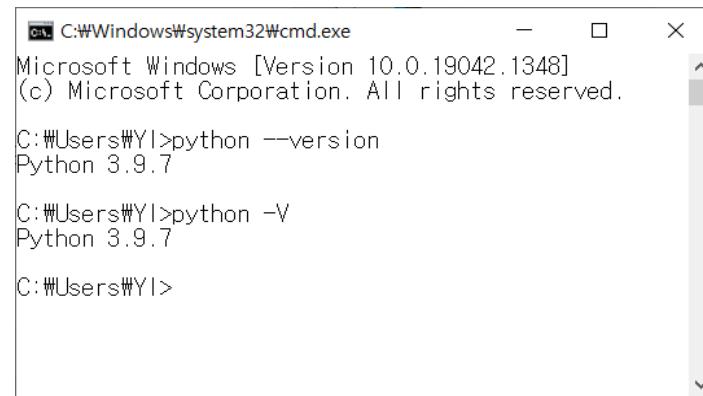
# Python 버전 확인

## 2. 개발 환경 구축

- Python --version (window 키+r -> cmd 러나 anaconda prompt)



```
관리자: Anaconda Prompt (Anacond... — □ ×
(base) C:\Windows\system32>python --version
Python 3.9.7
(base) C:\Windows\system32>
```



```
C:\Windows\system32\cmd.exe — □ ×
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

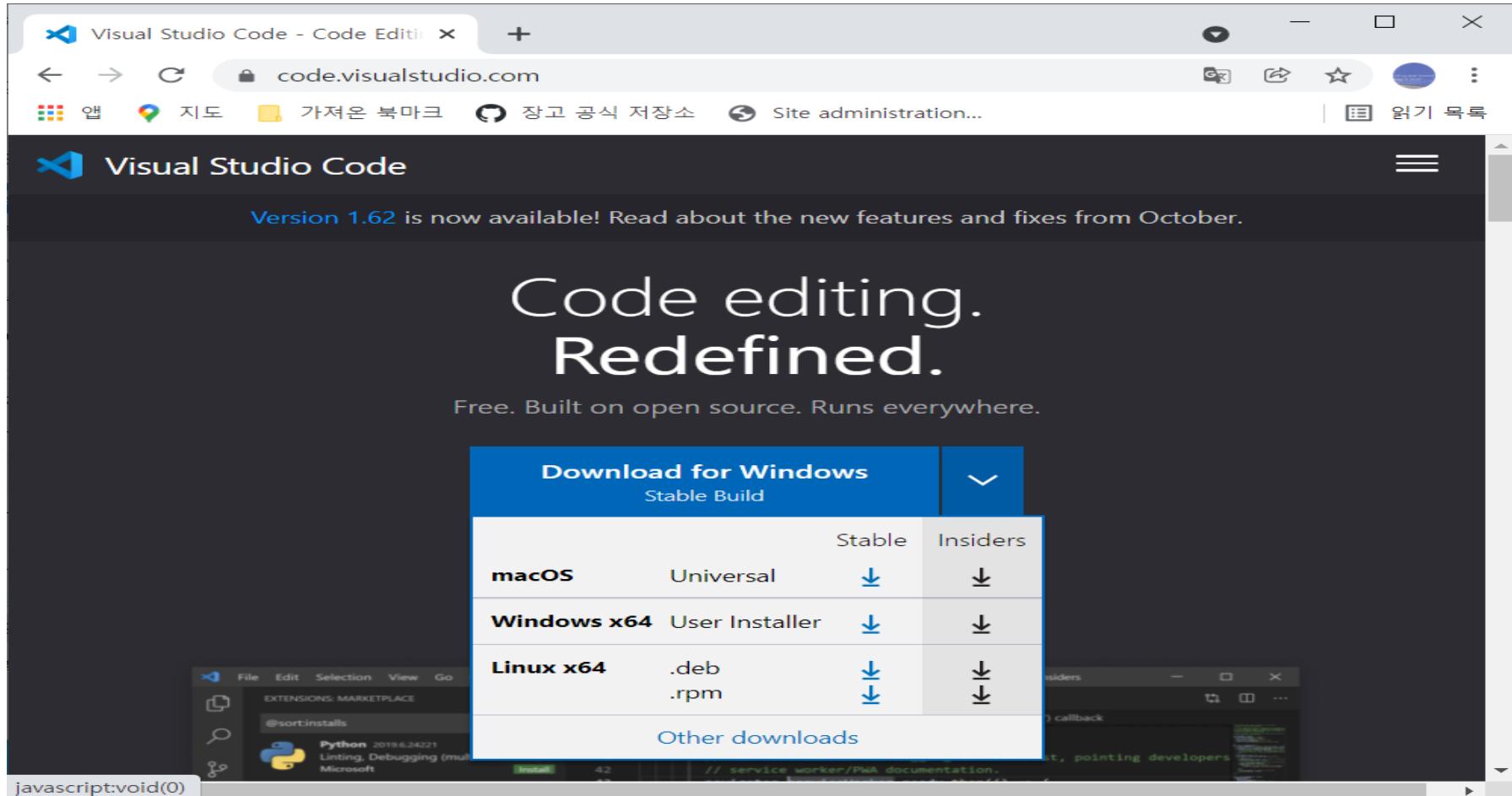
C:\Users\YI>python --version
Python 3.9.7
C:\Users\YI>python -V
Python 3.9.7
C:\Users\YI>
```

- virtual env – 가상환경 구축 설치하면 독립적으로 작업도 가능하다.  
수업시간에는 하나의 환경으로 할 예정

# Visual Studio Code(편집기)

## 2. 개발 환경 구축

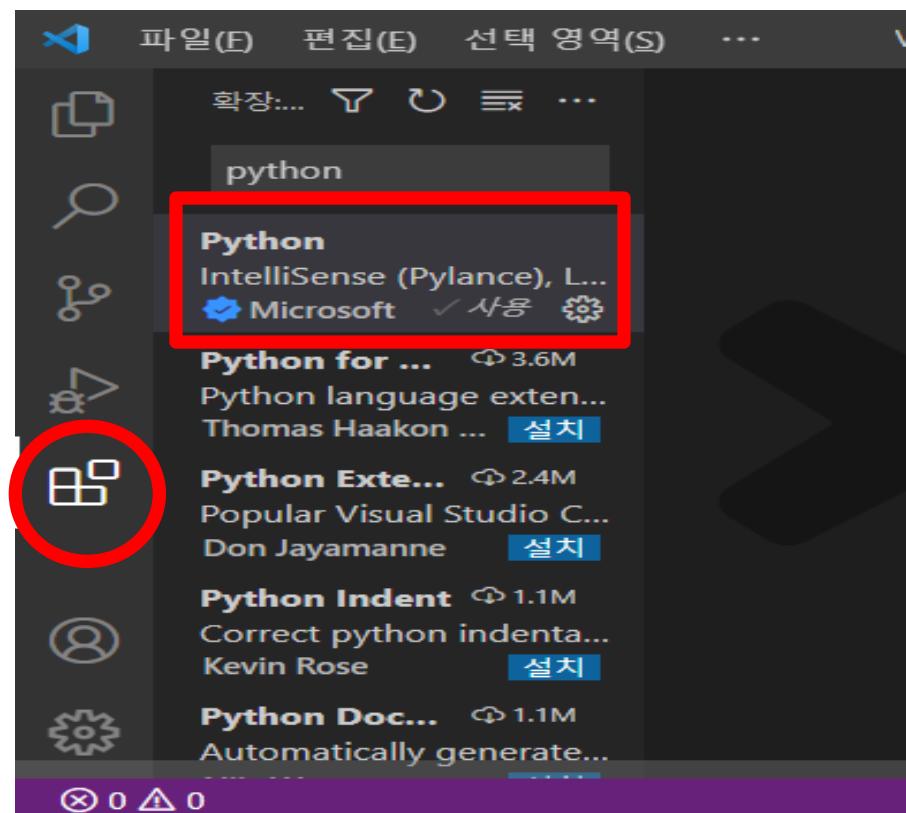
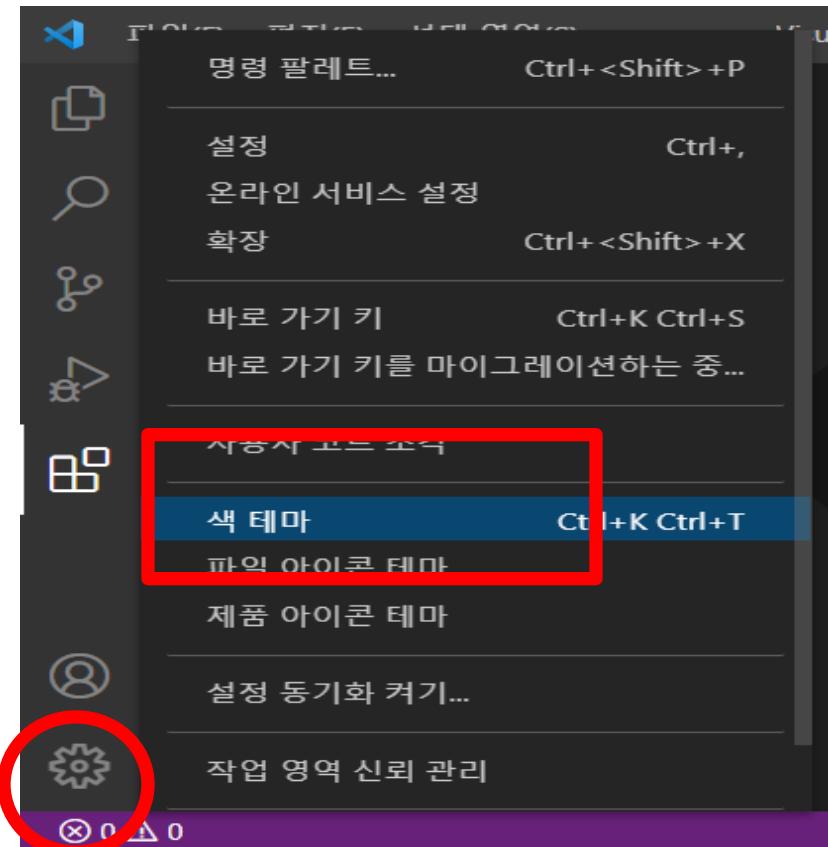
- <https://code.visualstudio.com>



# Python 확장

## 2. 개발 환경 구축

- 원하는 테마 설정
- 확장 -> python -> 설치 -> reload



# Django 설치

## 2. 개발 환경 구축

- Django 설치

```
D:\src\django>pip install django
```

최신버전

```
D:\src\django>pip install "django==5.1.1"
```

특정버전

```
D:\src\django>pip install "django~=5.1.0"
```

0 끝을 최신버전

- Django 버전 확인

```
D:\src\django>django-admin --version
```

```
D:\src\django>pip show django
```

- 장고 설치된 곳 찾기

```
D:\src\django>python -c "import django; print(django.__path__)"
```

- **Django 공식 소스 코드 저장소** - <http://github.com/django/django>

- **Django 공식 사이트** - <https://www.djangoproject.com>



### 3. Django 구조



# 실습

## ch01\_hello

### 1. 폴더 생성후, vscode에서 폴더 열고, .venv생성

- python -m venv .venv
- python -m pip install --upgrade pip
- pip --version

### 2. Django install

- pip install django

### 3. Django project 생성(현재 폴더에 :.)

- django-admin startproject ch01 .

### 4. SECRET\_KEY 숨기기(.gitignore생성)

- .env생성(**.gitignore에 .env추가**)
- pip install python-decouple
- Settings.py 를 수정하기  
from decouple import config  
SECRET\_KEY = config('SECRET\_KEY')

### 5. 장고 프로젝트 실행

- python manage.py runserver

### 6. home app 추가하기

- python manage.py startapp home
- settings.py에 home app 등록

# 실습

## ch01\_hello

- ch01/urls.py

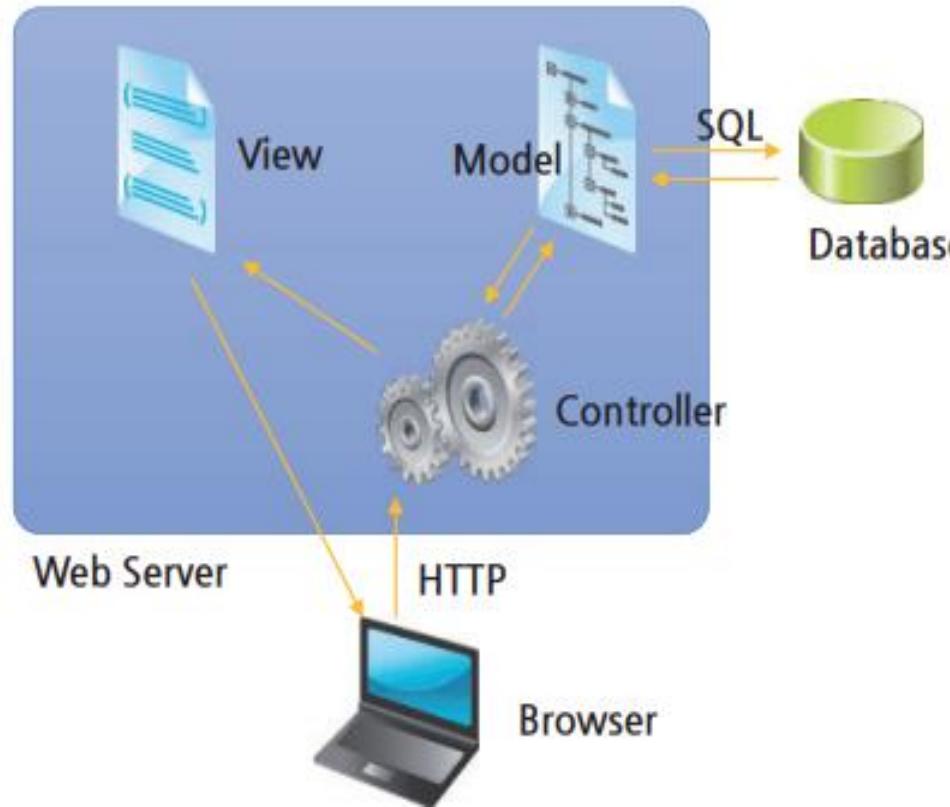
```
from django.urls import path
from home import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", views.home, name="home")
]
```

- home/views.py

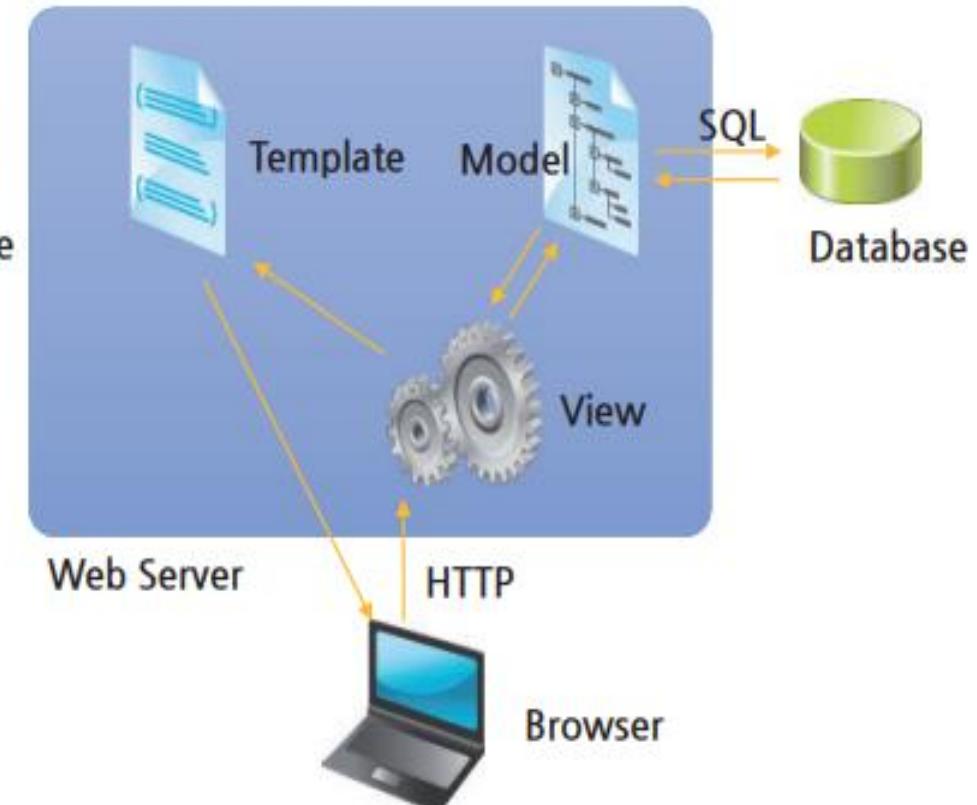
```
from django.shortcuts import render
from django.http import HttpResponse
def home(request):
    return HttpResponse('<h1>Hello, Django(장고)</h1>')
```

# MVC/MVT pattern

## 3. Django 구조



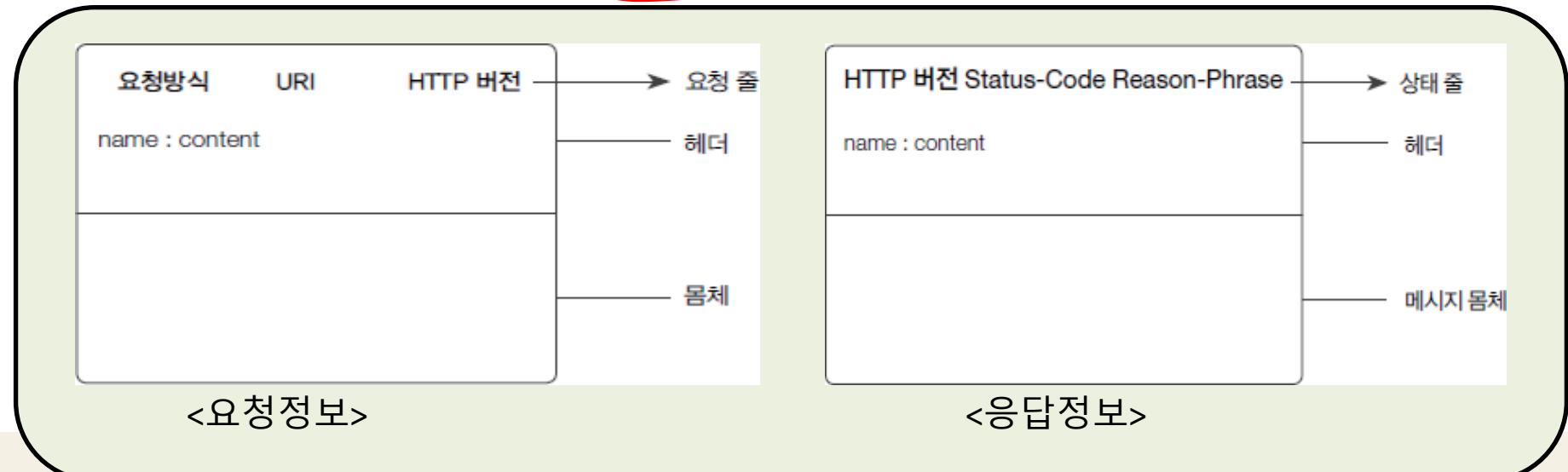
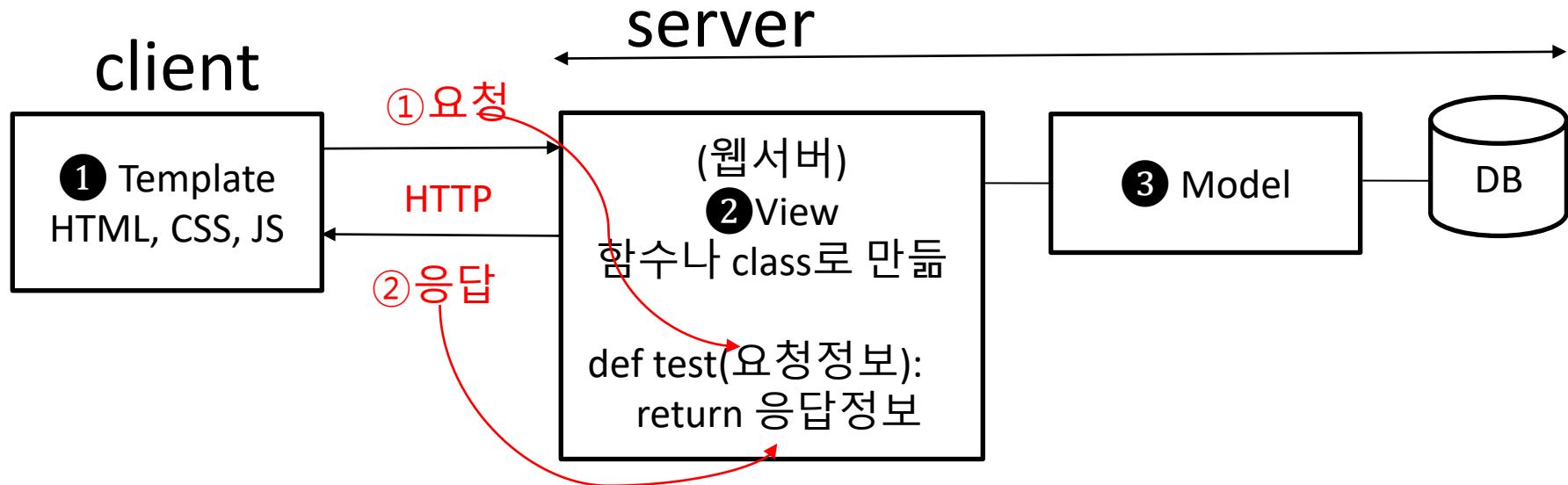
MVC (jsp, Spring)



MVT(Django)

# Django 구조 ; MVT 패턴

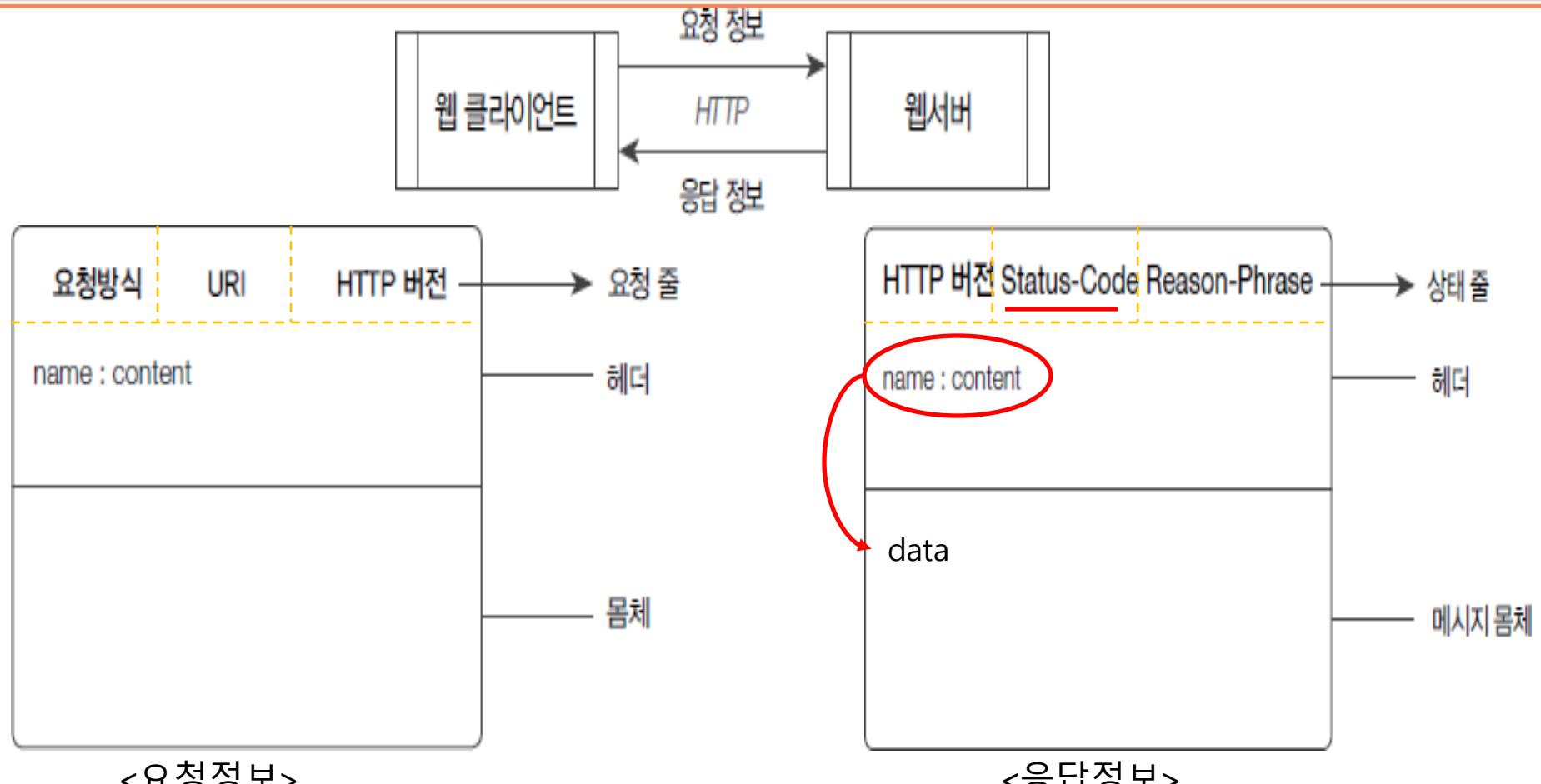
## 3. Django 구조



# HTTP Protocol

; 서버 연결 요청->클라이언트 응답->연결해제 (연결유지를 위해 쿠키 또는 세션 이용)

## 3. Django 구조

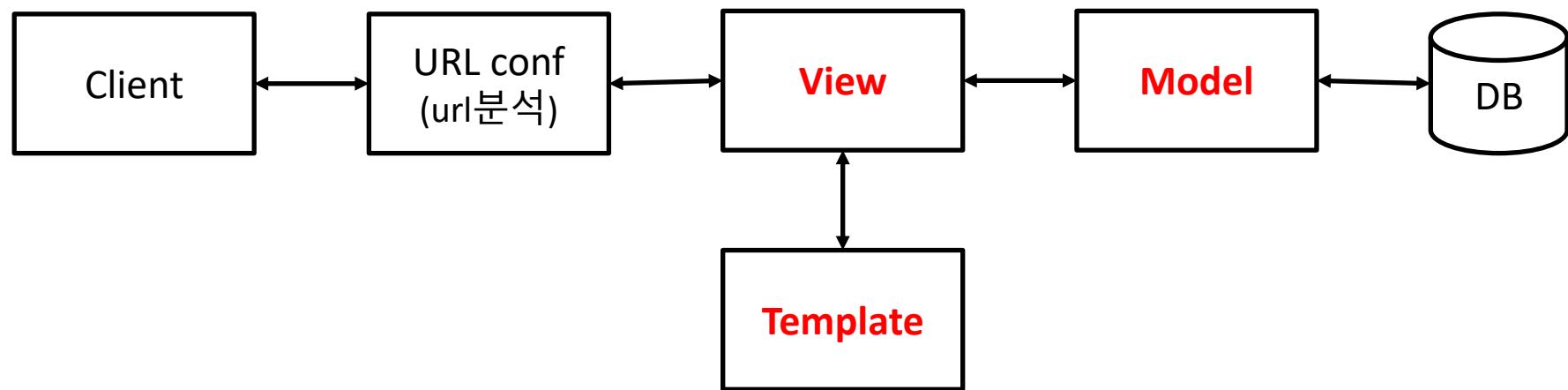


1. 요청방식 : GET/POST  
(GET방식은 몸체없이 헤더만 넘어옴)
2. URI : url 뒷부분  
url : <http://www.naver.com:80/show/index.html>  
uri : /show/index.html

1. Status-code(응답코드, 상태코드)  
200 : success code  
4xx : 인증관련error  
500 : 서버 프로그램 error

# Django Process Structure

## 3. Django 구조



# Django Process Structure

## 3. Django 구조

