

BE Week1

Django Basic

WHATEVER YOU WANT, MAKE IT REAL.

최윤한

BACK-END

Django?

Python의 가상환경

MTV Pattern

Postman

실습



What is BACKEND ??

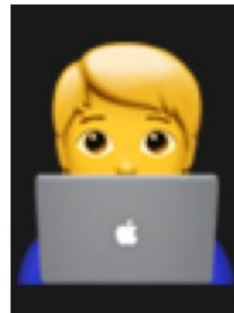
서버 개발, 웹 애플리케이션 개발, 데이터베이스 설계 등등... 의 개발을 하는 분야

=> 공부 할게 많다..(알고리즘, 네트워크, 데이터베이스, 보안, 프레임워크 기술 등등...)

3월 23일 첫 세션 기억하시나요?



Server



Backend

Client에게 제공할 수 있는
기능과
Data를 제공한다!

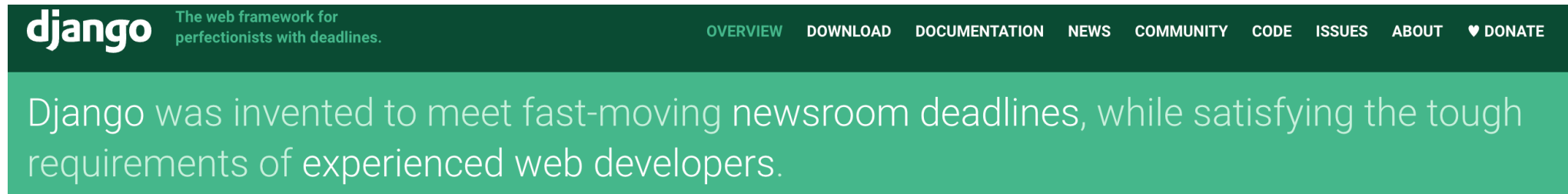


Frontend

Client가 만족하는
Interface를 만든다!
(Browser에 어떻게 보일건지..)

Client

What is Django ??

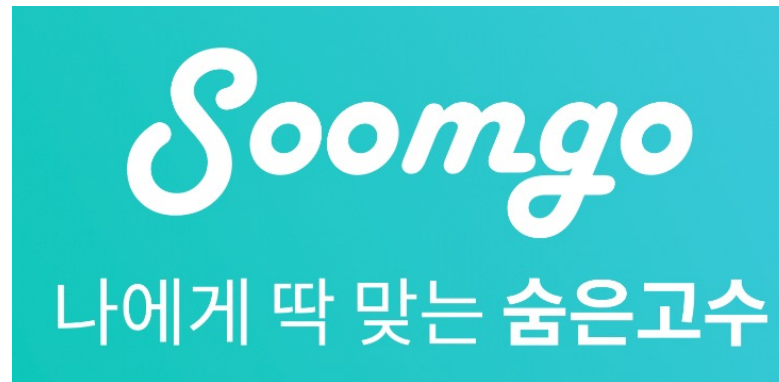


Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Python 기반의 Web Full-Stack Framework

<https://www.djangoproject.com/start/overview/>

Django를 사용하는 서비스



<https://techblog.yogiyo.co.kr/tagged/django>

Why Django ??

Why Django?

With Django, you can take web applications from concept to launch in a matter of hours. Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

1. 개발속도가 빠르다.
2. Admin, ORM, Template Engine, Queryset, etc...
3. 인증, 보안 부분들도 고려됨.
4. Database (SQLite) 제공

<https://www.djangoproject.com/start/overview/>

What is Framework ?

- 개발자가 애플리케이션을 개발할 때 필요한, 기능과 도구들을 미리 구현하여 제공하는 것들.



Framework와 Library의 차이

IoC (Inversion Of Control) : 제어의 역전

가상환경

- 가상환경(Virtual Environment)이란?
프로젝트를 진행하기 위한 가상의 환경 (cf. Global 환경)

- 가상환경이 필요한 이유는?

프로젝트, 애플리케이션 마다 요구하는 버전이 다르다.

Ex) Django 설치 => pip3 install django

A 프로젝트 : Django 3.0 버전만 실행 가능하다.

B 프로젝트 : Django 4.0 버전만 실행 가능하다.

그럼 우리는 A, B 프로젝트를 실행할 때마다, Django를 버전에 맞게 설치 했다가 지웠다가 해야 하는가?

파이썬의 가상환경 => venv, virtualenv, pipenv, poetry, anaconda / pyenv 등등 ...

파이썬의 가상환경, pip

- `pip list` : 설치된 package 확인
- `pip list -v` : 설치된 package의 version 및 경로 확인
- `pip install (package)` : package 설치
- `pip install --upgrade pip` : pip upgrade

- `pip freeze` : 설치된 package를 requirements format으로 보여준다.
- `pip freeze > requirements.txt` : 설치된 package를 requirements.txt로 저장
- `pip install -r requirements.txt` : requirements.txt에 있는 package를 설치

- `pip show` : 설치된 package 정보 확인
- `pip --help` : pip 명령어 확인

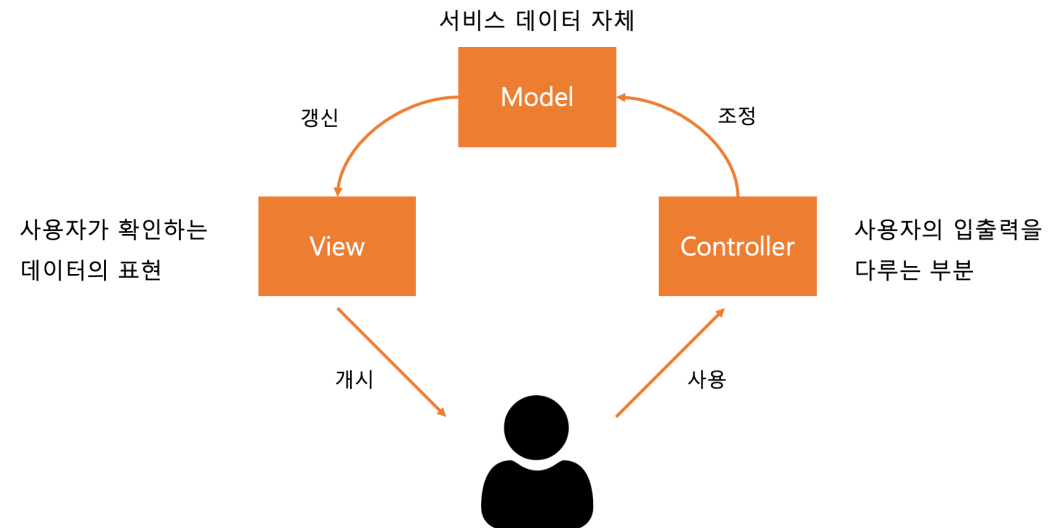
Design Pattern

- 개발할 때 공통적으로 상황에 따라 쓸 수 있는 것을 정리하여 만들어낸 '규약'

<MVC 패턴>

- Model, View, Controller
- 하나의 애플리케이션을 세가지 역할을 하는 구성요소로 분리한 패턴

Model - View - Controller

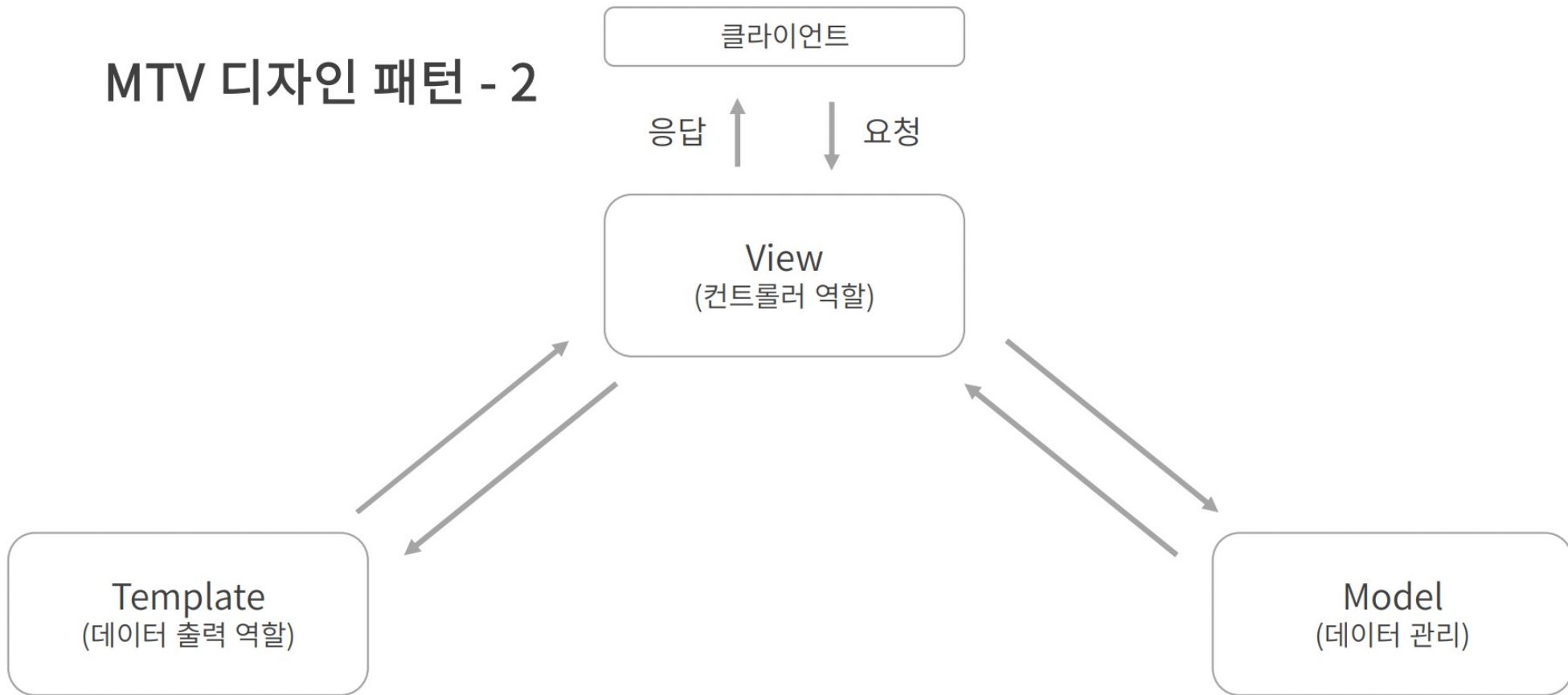


What is MTV Pattern

구분	역할	파일	비고
Model	<ul style="list-style-type: none">• 데이터 관리• 데이터베이스와 연결 및 실행	models.py	
Template	<ul style="list-style-type: none">• 데이터 출력• 사용자에게 표현 방식 정의	html	
View	<ul style="list-style-type: none">• 컨트롤러• 비즈니스 로직을 처리	views.py	

What is MTV Pattern

MTV 디자인 패턴 - 2



Postman

What is Postman?

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.



API Tools

A comprehensive set of tools that help accelerate the API Lifecycle - from design, testing, documentation, and mocking to discovery.



API Repository

Easily store, iterate and collaborate around all your API artifacts on one central platform used across teams.



Workspaces

Organize your API work and collaborate with teammates across your organization or stakeholders across the world.



Intelligence

Improve API operations by leveraging advanced features such as search, notifications, alerts and security warnings, reporting, and much more.

API와 관련된 여러가지 작업들을 할 수 있는 Platform

실습 초기 세팅

`python3 -m venv (가상환경명)` : 가상환경 생성

`source venv/bin/activate` or `source venv/Scripts/activate` : 가상환경 활성화

⋮ `pip3 install django` : 장고 설치

`django-admin startproject config .` : 프로젝트 config 파일 생성

`django-admin startapp (appname)`

실습 pip

- Mac OS의 경우 python, pip 끝에 3을 붙여준다.

`python -m venv (가상환경명)` : 가상환경 생성

`source venv/bin/activate` or `source venv/Scripts/activate` : 가상환경 활성화

`pip install django` : 장고 설치

`django-admin startproject config .` : 프로젝트 config 파일 생성

`django-admin startapp (appname)` : 앱 생성

`python manage.py runserver`

실습 Settings.py

```
DJANGO_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]  
  
PROJECT_APPS = [  
    'posts',  
]  
  
THIRD_PARTY_APPS = []  
  
INSTALLED_APPS = DJANGO_APPS + PROJECT_APPS + THIRD_PARTY_APPS
```

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [BASE_DIR / 'templates'],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    ],  
]
```


실습
.gitignore

gitignore.io

자신의 프로젝트에 꼭 맞는 .gitignore 파일을 만드세요

Django ×

생성

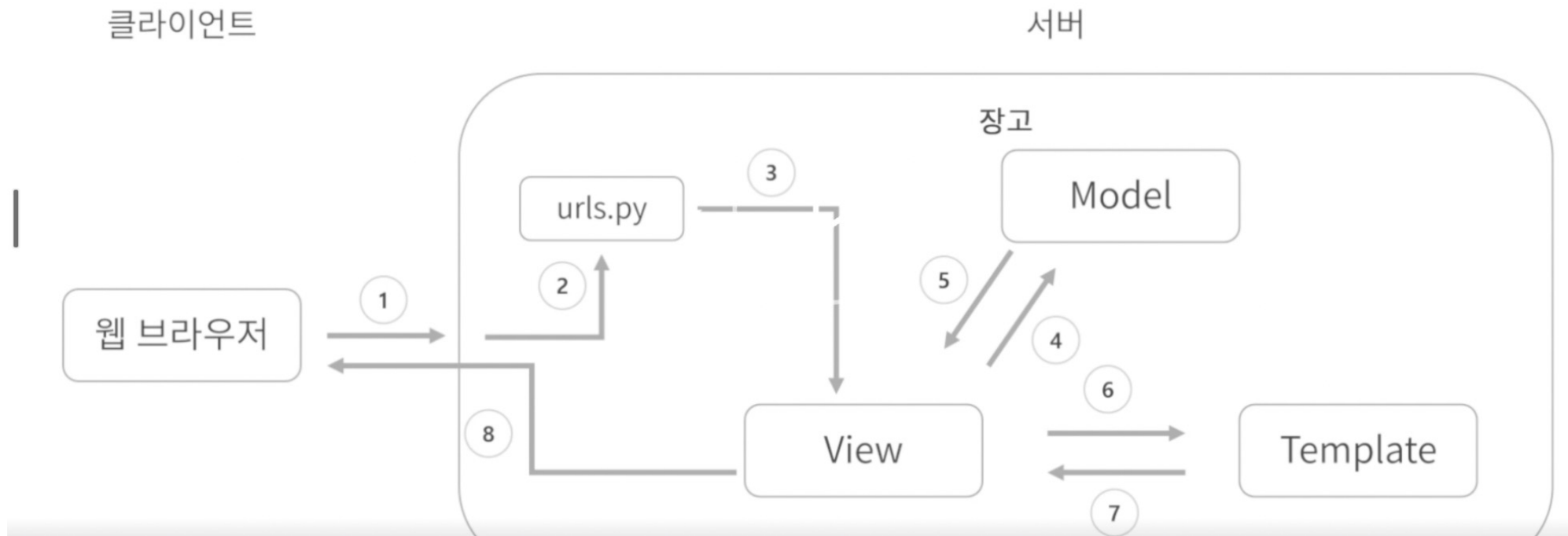
[소스 코드](#) | [커맨드라인 문서](#)

실습 secrets.json

(복습) JSON(JavaScript Object Notation): key : value 형태의 data format

```
{  
  "SECRET_KEY" : "django-insecure-123sdsdo"  
}
```

실습 Django의 흐름



실습 Convention

Convention이 중요한 이유?

Naming Convention

1. 파이썬 변수, 함수는 snake case를 따른다. (cf. Camel Case)
2. Class의 경우 Pascal case를 따른다.
3. Django에서 App 이름은 영어 소문자 복수형으로 한다.(ex. posts, accounts, users)

Snake Case

ex) `file_list`

Pascal Case

ex) `Animal`, `PostSerializer`

Camel Case

ex) `querySelector`

과제

django 프로젝트 생성 후, 멋사 10기 깃허브 본인 repository에 업로드가 아닌,
git 명령어를 이용하여 레포지토리로 push 해주세요!

<조건>

1. project 생성
2. posts 앱 생성
3. settings.py 세팅
4. .gitignore, secrets.json 설정
5. Views.py에 method 생성 후, postman으로 테스트 한 사진 캡처 후, README.md에 첨부

codelion.net 장고 강의 같이 들으면서 공부하시면,
훨씬 효과적입니다!

감사합니다!