Django 초보 가이드

목차

-MVC, MFC

-Django 개념

-Project와 App – 웹프로젝트는 다수의 앱으로 구성되어 있음

-settings.py

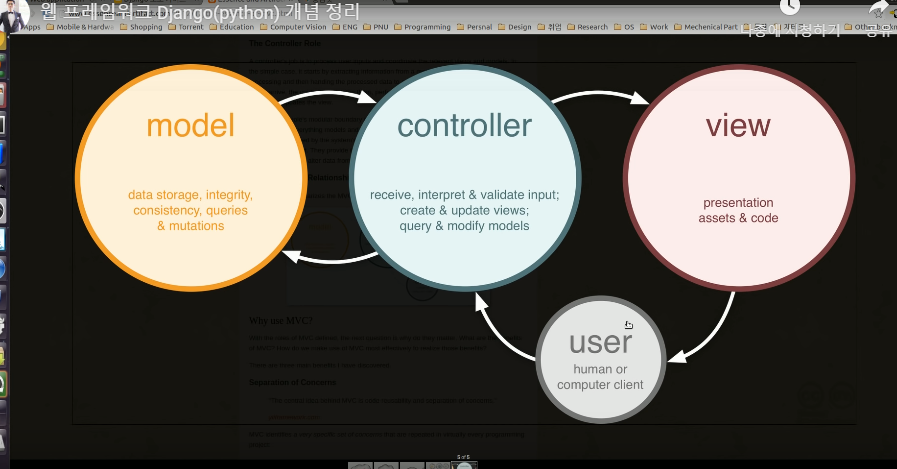
-manage.py

1.MVC & MTV

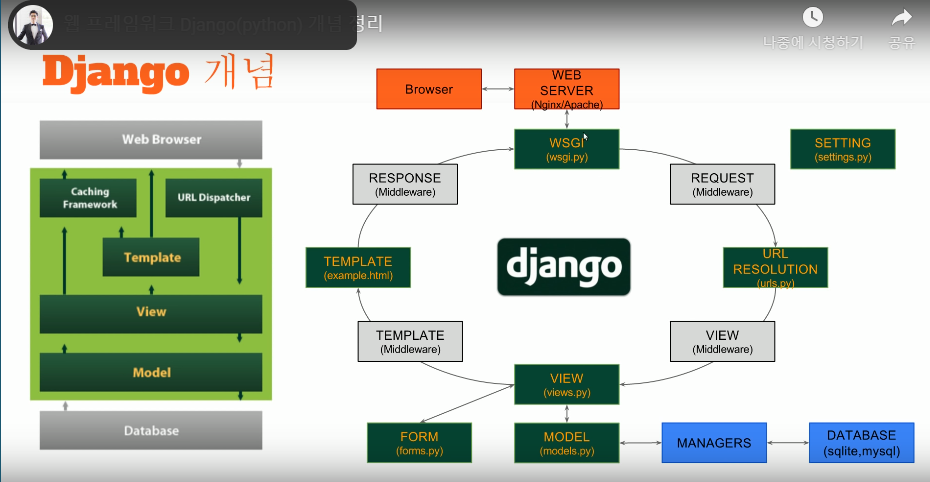
Model : 안전하게 데이터를 저장

View : 데이터를 적절하게 유저에게 보여줌

Control, Template(Django) : 사용자의 입력과 이벤트에 반응하여 Model과 View를 업데이트

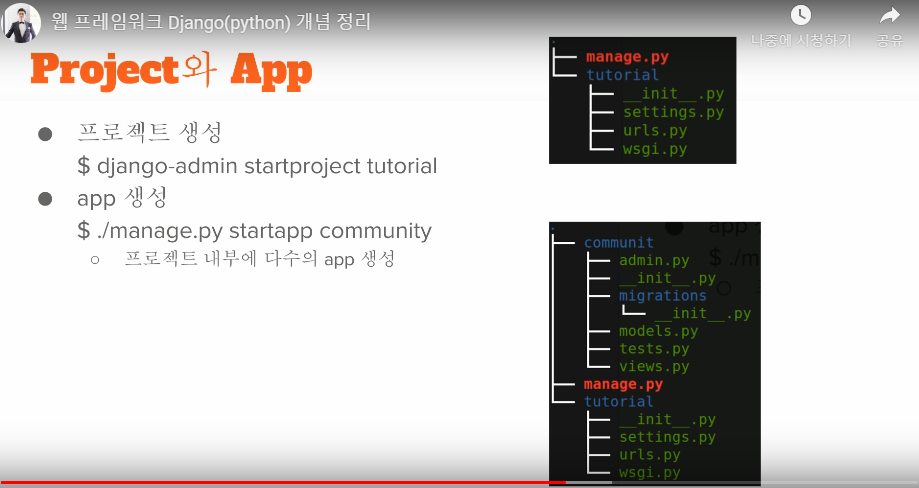


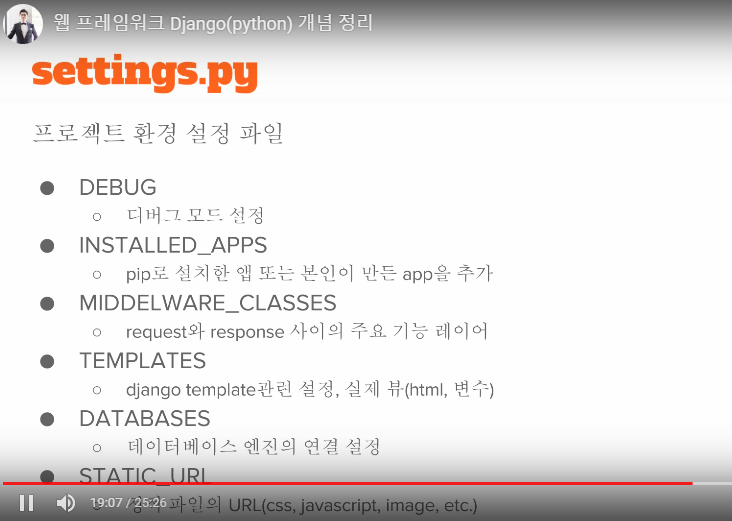
2. Django의 개념

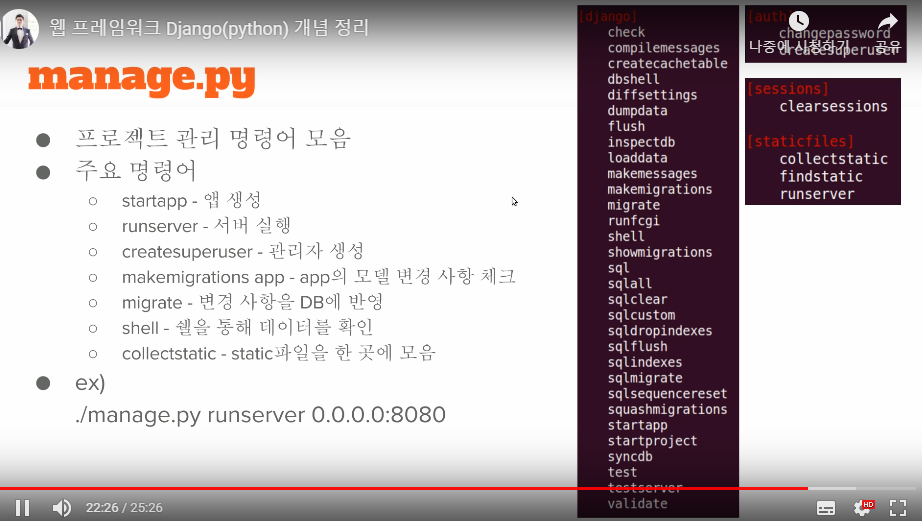


1. 브라우저에서 어떤 요청을 보내게 되면은 웹서버게이트웨이인터페이스(WSGI)에 먼저 신호가 들어온다. (이 파일은 우리가 크게 건드릴 일이 없다고 한다.)  
   이것의 역할은 웹서버와 장고를 적절히 결합시켜주는 역할이라고 한다.
2. 그 다음 신호를 urls.py에서 받게 된다. Urls.py는 정규표현식으로 구성되어 있다.  
   정규표현식에 맞게끔 특정한 VIEW로 보내주게 된다.
3. Views.py가 우리가 코드를 많이 작성하는 부분이라고 한다.  
   View에서 판단을 한다. 사용자의 요청에 의해서 데이터베이스에 데이터를 입력을 하던지, 데이터베이스로부터 데이터를 가지고와서 사용자에게 리스트를 보여줄 것인지
4. 결정을 하고 난 다음에 Model에 신호를 보내게 된다. 모델은 데이터베이스로부터 데이터를 가지고 올 때 사용
5. 모델에서 받은 데이터를 사용자에게 보여주기 위하여 UI 작업을 해야한다.
6. Template(html 파일)에서 UI를 만들어서 웹서버로 전송을 한다. Form과 같이 편리하게 화면을 만들 수 있다.

Project와 App







2강

웹 프레임워크 Django 실습

1. 프로젝트 & app 생성
2. 디렉토리 구조 확인
3. 관리자 페이지 확인
4. 글쓰기
5. 리스트
6. 글 보기

순서

Pip install Django – 으로 장고 설치

Django-admin startproject tutorial – 로 프로젝트 생성

Cd tutorial

<정보>Tutorial/urls.py에 외부에서 접근할 수 있는 url들이 표시된다.

./manage.py startapp community – 로 앱 생성 (윈도우면 앞에 python 붙여야 함)

Python ./manage.py migrate – settings에 설정되어있는 경로로 데이터베이스 생성

* 저걸 입력하면 장고가 필요한 기본적인 데이터베이스 테이블들이 생성되었음

Python ./manage.py createsuperuser - 슈퍼 유저 만들기

Id, ps : admin으로 생성하였음

Python ./manage.py runserver - 서버 실행 가능

만약에 포트번호 바꿔서 실행하고 싶다면 Python ./manage.py runserver 8080 -이렇게 적고 실행

서버를 돌린 다음 127.0.0.1:8080/admin 으로 어드민 페이지 접근 가능하다.

<정보> 어드민 페이지에서 기본적으로 user들을 관리할 수 있다. 특정 계정을 삭제할 수도 있다.

<정보> Group을 지정할 수 있다.

<정보> Model을 통해서 데이터베이스, 데이터를 만들면 어드민 계정을 통해서 그 데이터를 확인하고 수정하고 삭제할 수 있다. 따로 페이지를 만들지 않아도 되어서 편리하다.

Settings.py에서 INSTALLED\_APPS 안에 생성한 community 앱을 추가하여야 만든 앱을 사용할 수 있다.

그 다음 할 일은 community/models.py 코드를 생성하는 것이다.

<정보>Model은 필요한 데이터이고 기본적으로 class로 생성된다. 클래스는 기본적으로 models.Model을 상속하게 된다.

Models.py 부분에 아래와같이 클래스를 만들고

class Article(models.Model):

    name = models.CharField(max\_length=50)

    title = models.CharField(max\_length=50)

    contents = models.TextField()

    url = models.URLField()

    email = models.EmailField()

    cdate = models.DateTimeField(auto\_now\_add=True)

이제 데이터베이스에 적용하기 위하여

python manage.py makemigrations community - community라는 앱에 어떤 변화가 있는지 확인하는 단계

python manage.py migrate - 명령어를 주면 community라는 Table이 생성된다.

<정보> 여기까지 했으면 위 class에 대한 database 테이블을 db.sqlite3에 생성을 한 단계이다.

! 이제 글쓰기 form 을 만들어 보자

먼저 urls.py에서 User들이 접근할 수 있는 주소를 만들어야 한다.

from community.views import \*

path('write/', write, name='write'),

추가한다.