

# Python Django

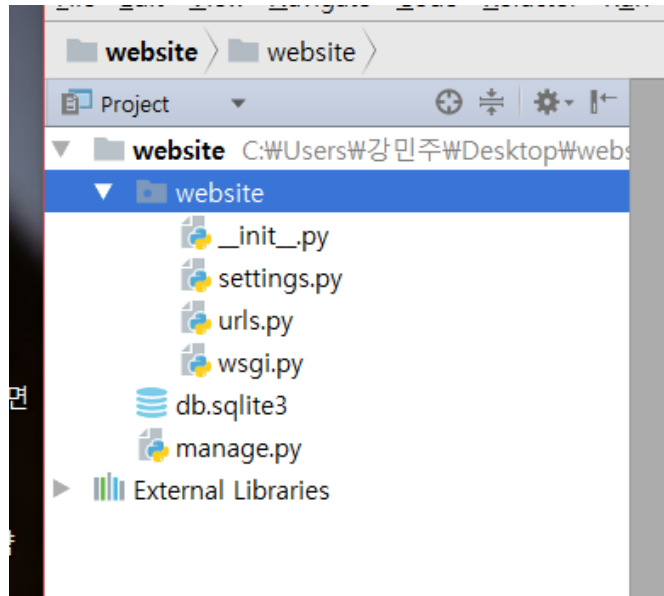
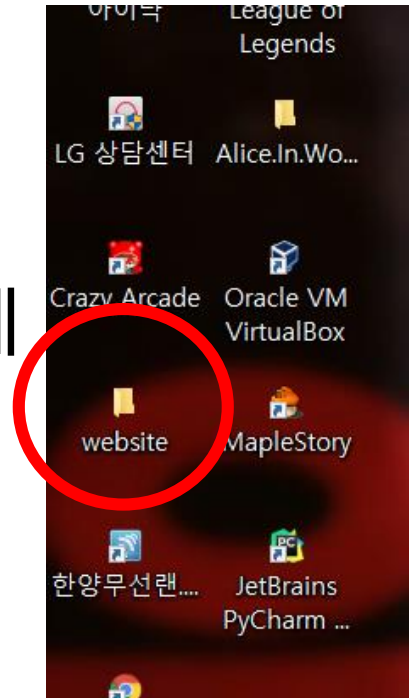
2강 Python Django 시작하기

# Python Django 시작하기

## 1. 장고를 이용해 기본 웹사이트 형식 파일 생성

cmd창을 admin으로 실행 -> 시작하고 싶은 파일 위치에

> Django-admin startproject 파일이름



\*아까 한 실행을 Pycharm이 있다면  
그 프로그램에서 바로 생성할 수 있음

\_\_init\_\_.py -> 파이썬에게 이 폴더가 프로젝트임을 알려줌

Setting.py -> 웹사이트의 설정을 보여주는 코드

Urls.py -> 사이트의 url 목록을 보여줌

Wsgl.py -> 웹서버 게이트웨이 인터페이스

Manage.py -> db관리 / user 생성 등 다양한 기능 수정 또는 삭제 절대 X

# Python Django 시작하기

## 2. 웹사이트 실행해보기

생성한 프로젝트 파일 > `python manage.py runserver`

- 간단하게 개발용 서버를 열어서 어떻게 됐는지 확인 할 수 있음
- Manage 툴을 이용해서 서버 돌림 + ctrl+c 누르면 닫힘

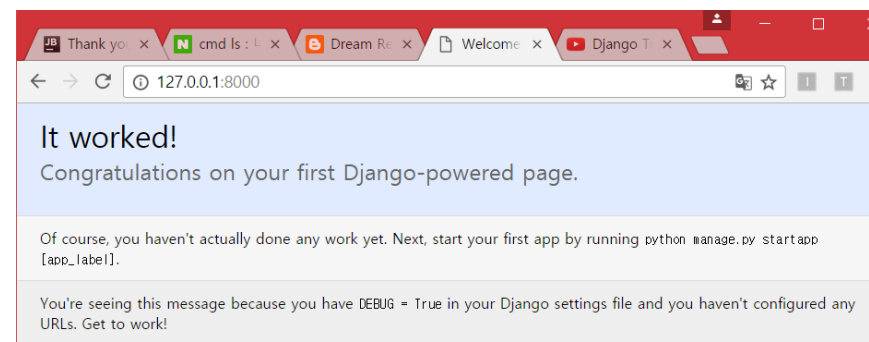
```
C:\Users\강민주\Desktop\website>doskey ls = dir
C:\Users\강민주\Desktop\website>ls
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: CEC7-36DB

C:\Users\강민주\Desktop\website 디렉터리
2017-01-14 오후 10:32 <DIR> .
2017-01-14 오후 10:32 <DIR> ..
2017-01-14 오후 10:47 <DIR> .idea
2016-12-28 오전 12:54          3,072 db.sqlite3
2016-12-28 오전 12:19          805 manage.py
2016-12-28 오전 12:54 <DIR> website
                2개 파일          3,877 바이트
                4개 디렉터리 81,759,367,168 바이트 남음

C:\Users\강민주\Desktop\website>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 13 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
January 14, 2017 - 23:00:57
Django version 1.10.4, using settings 'website.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



```
January 14, 2017 - 23:05:22
Django version 1.10.4, using settings 'website.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[14/Jan/2017 23:05:32] "GET / HTTP/1.1" 200 1767
```

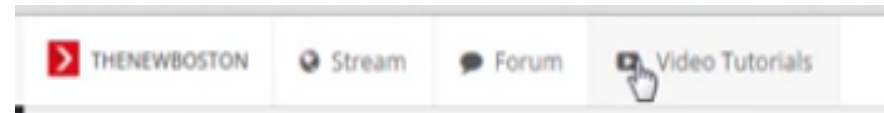
# Python Django 시작하기

## 3. App 만들기

서버를 켜 놓으면서 코드를 수정해도 서버를 다시 켤 필요가 없음! 그냥 '새로고침' 하면 적용됨

자신의 만들 사이트를 구성하는 기능의 요소들 -> App이라고 함 ex)

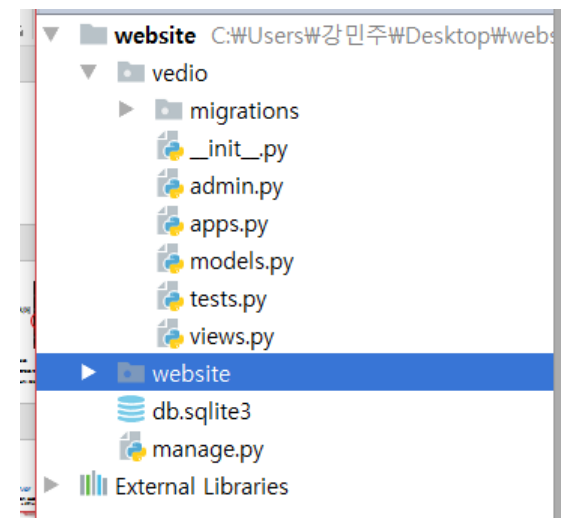
-> 각 앱은 하나의 기능만 해야함!



생성한 프로젝트 파일 > `python manage.py startapp 앱이름`

```
C:\Users\강민주\Desktop\website>python manage.py startapp vedio
```

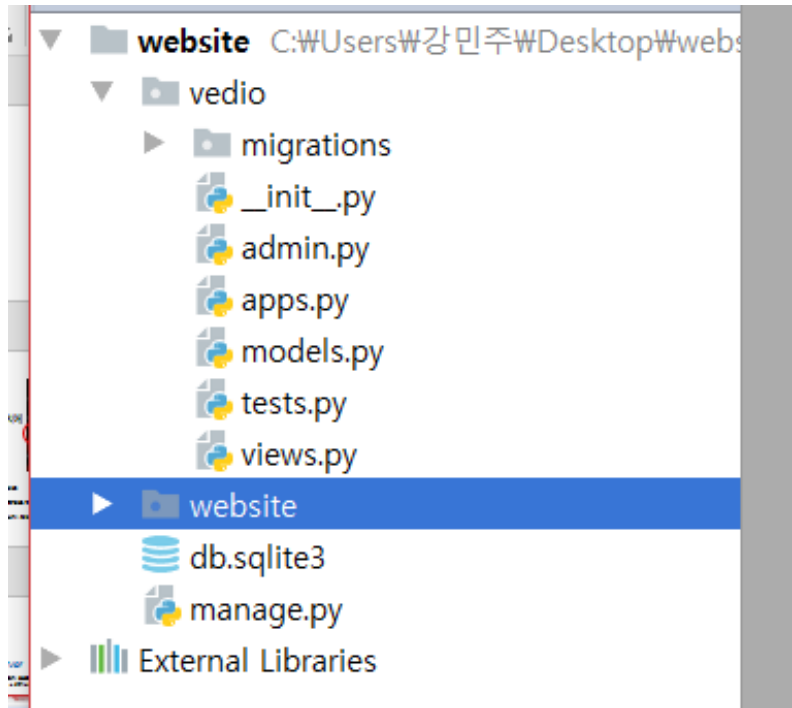
```
C:\Users\강민주\Desktop\website>
```



# Python Django 시작하기

## 3. App 만들기

## 기본적인 App 파일 구성



migrations -> 만든 app안의 코드를 사이트의 db와 연결 시켜 줌

admin -> 말그대로 관리자 section

apps -> 그 앱의 기본적인 setting, configuration

models -> 데이터 베이스의 blueprint / 어떻게 데이터를 저장시킬지 설정  
db의 entity 같은 거라고 생각하면 쉬움

예를 들어서 영화 저장 공유 사이트일 때

들어오는 영화에 대한 정보 ( 이름, 개봉연도, 장르) -> 이런 형식을 선언

tests -> 코드에 버그 없는지 테스트

views -> 웹을 통해 유저에게 얻는 respond에 대해 다시 response를 주는 것  
말 그대로 웹 페이지 그 자체..!

# Python Django 시작하기

## 4. App에 대한 Views 만들기

1) Url 정규식 연결시키기

127.0.0.1:8000/vedio/

127.0.0.1:8000/admin/

각 앱은 각 앱의 기능을 표명해주는 url을 가질 수 있도록 url을 만들어 주어야함!

Ex) vedio app에서 vedio를 시청하는 기능이라면 /vedio/watch

Ex) music app에서 음악을 올리는 기능이라면 /music/upload

```
from django.conf.urls import url
from django.contrib import admin

urlpatterns = [
    url(r'^admin/', admin.site.urls),
]
```

자신이 만든 사이트의 기본 url

여기다가 각 앱의 기능을 덧붙여 줄 수 있는 url 문구를 붙여야 한다!

Website - vedio  
- music  
- picture

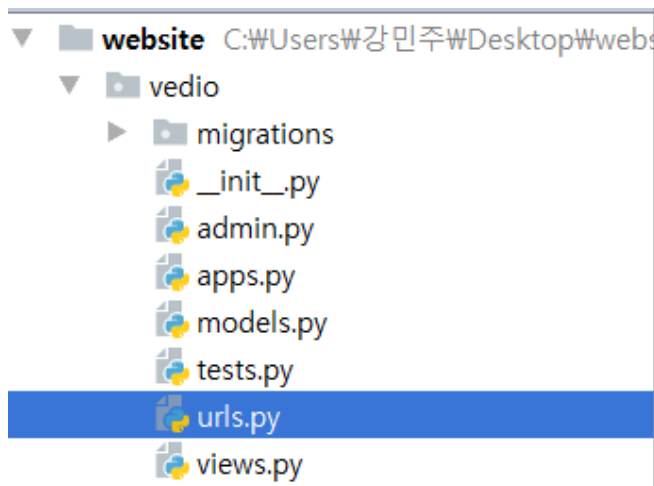
각 해당 앱에 대한 url 연결 문구!

# Python Django 시작하기

## 4. App에 대한 Views 만들기

### 1) Url 정규식 연결시키기

#### 1. 새로 만든 app 파일 안에 urls.py 파일 생성 후 아래와 같이 작성



# Python Django 시작하기

## 4. App에 대한 Views 만들기

### 1) Url 정규식 연결시키기



```
views.py x websiteUrls.py x vedioUrls.py x
1 from django.conf.urls import url
2 from . import views
3
4 urlpatterns = [
5     url(r'^$', views.index, name='index'),
6 ]
```

From . 에서의 '.' -> 해당 파일이 있는 위치  
즉, app의 views.py 파일을 import

url(r'^\$', views.index, name = 'index')

r' -> 정규표현식

Views.index -> 해당 url이 실행될 경우 실행되는 함수

name = 'index' -> url 구분을 위한 이름 나중에 되게 쓸모 있음



# Python Django 시작하기

## 4. App에 대한 Views 만들기

1) Url 정규식 연결시키기

### 2. App의 url과 본 website url과 연결

website 의 url.py 파일에 작성

```
from django.conf.urls import include, url
from django.contrib import admin
```

include 기능 사용을 위해 import

```
urlpatterns = [
    url(r'vedio/', include('vedio.urls')),
    url(r'^admin/', admin.site.urls),
```

vedio/ 라는 url문구를 website 파일 urls.py 안에 만들어주고  
후에 연결될 말을 vedio app 안의 urls 파일과 연결

# Python Django 시작하기

## 4. App에 대한 Views 만들기

### 2) App에 대한 view 작성하기

새로 만든 app의 views.py 파일에 작성

```
views.py x websiteWurls.py x vedioWurls.py x
1 from django.conf.urls import url
2 from . import views
3
4 urlpatterns = [
5     url(r'^$', views.index, name='index'),
6 ]
```

```
views.py x websiteWurls.py x vedioWurls.py x
1 from django.http import HttpResponse
2
3 def index(request):
4     return HttpResponse("<h1>This is the Vedio app homepage</h1>")
```

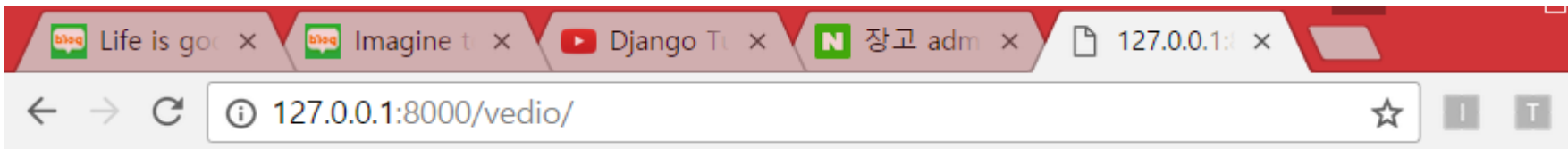
HttpResponse를 import

Views 안에 index 함수 생성  
request를 받으면 아래 return 값을 웹 창에 띄우게 됨

# Python Django 시작하기

## 4. App에 대한 Views 만들기

### 2) App에 대한 view 작성하기



**This is the Vedio app homepage**

해당 url을 입력 후 or 새로고침 하면 문구 땡

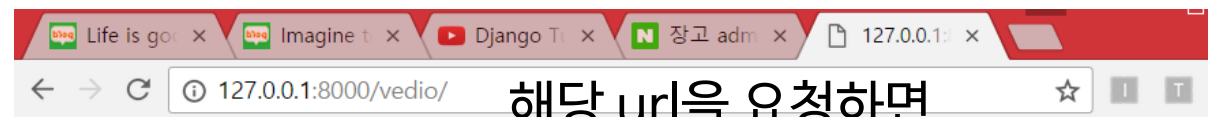
# Python Django 시작하기

## 4. App에 대한 Views 만들기

```
from django.conf.urls import include, url
from django.contrib import admin
```

```
urlpatterns = [
    url(r'vedio/', include('vedio.urls')),
    url(r'^admin/', admin.site.urls),
```

```
1 from django.conf.urls import url
2 from . import views
3
4 urlpatterns = [
5     url(r'^$', views.index, name='index'),
6 ]
```



This is the Vedio app homepage

```
1 from django.http import HttpResponse
2
3 def index(request):
4     return HttpResponse("<h1>This is the Vedio app homepage</h1>")
```

이런 순서를 거쳐서 페이지에 표시 !

# Python 정규표현식 (regular expression)

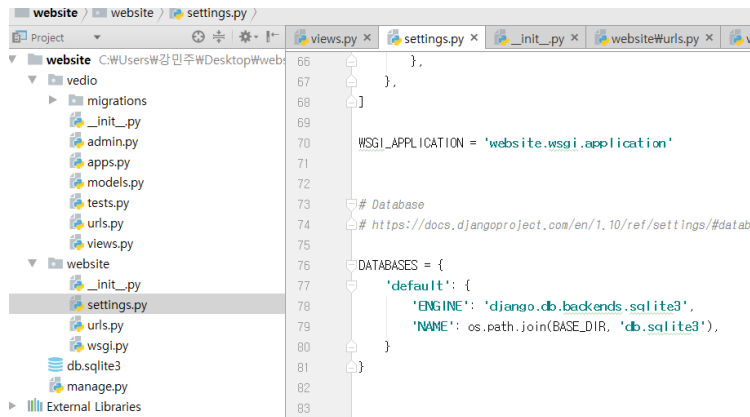
정규표현식 : 특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용

- 특정 목적을 위해 필요한 문자열 집합을 지정하기 위해 쓰임
- 많은 양의 텍스트가 있을 경우, 그 중 필요한 정보만 뽑아서 가져오는데 사용 할 수 있음
- google에 정규 표현식을 치면 자신의 정규식을 테스트 할 수 있는 홈페이지가 굉장히 많음

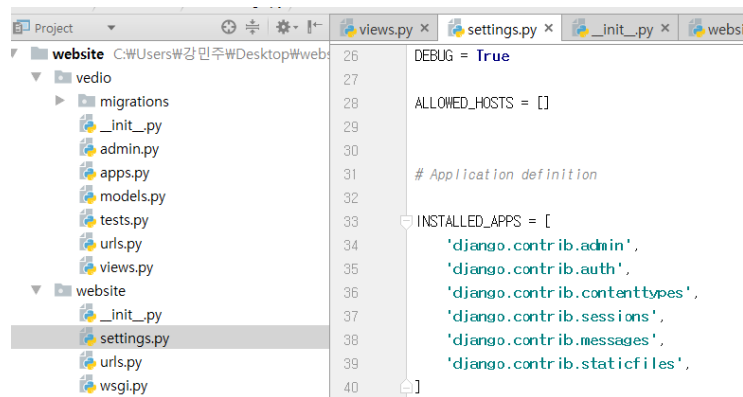
메타 문자	기능	설명
^	처음	문자열이나 행의 처음
\$	끝	문자열이나 행의 끝
.	문자	1개의 문자와 일치, 단일행 모드시 새줄 문자는 제외
[ ]	문자 클래스	‘[’와 ‘]’ 사이 문자 중 하나를 선택 ex) [abc]d = ad, bd, cd “-” 기호로 범위 지정 가능 ex) [az] = a~z 중 하나, [1-9] = 1~9 중 하나

# Python Django 시작하기

## 5. 데이터베이스와 연결하기



- parent website의 폴더를 보면  
이미 default DB가 있고 setting도 되어있음  
하지만 이 DB가 우리가 만든 app과는 연결되어 있지  
않기 때문에 코드를 통해 이것을 연결 시켜줘야 함!



-그리고 setting에는 처음부터 우리가 웹을 만들기 위한  
기본 app들이 만들어져 있음

-> 이 app들 중에서는 DB와 연결되어 각자 자기가 필요  
한 DB 테이블을 가지고 있는 app들이 존재함

# Python Django 시작하기

## 5. 데이터베이스와 연결하기

코드 수정 혹은 앱을 만든 후 서버를 다시 켜면 아래 문구와 함께 서버가 작동됨

```
"C:\Program Files (x86)\JetBrains\PyCharm 5.0.1\bin\runnerw.exe" C:\Users\Bucky\AppData\Local\Temp\1\pycharm\runnerw.exe
Performing system checks...

System check identified no issues (0 silenced).

You have unapplied migrations; your app may not work properly until they are applied.
Run 'python manage.py migrate' to apply them.
March 21, 2016 - 09:49:47
Django version 1.9.1, using settings 'website.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

-> 이것은 내가 작성한 코드가 DB와 싱크가 맞지 않다는 것을 알려줌

싱크를 맞춰 주기 위해서 cmd창에

>python manage.py migrate 라고 작성

Migrate는 setting에 있는 installed\_apps의 각 app과 각자의 DB table이 연결 될 수 있도록 해줌

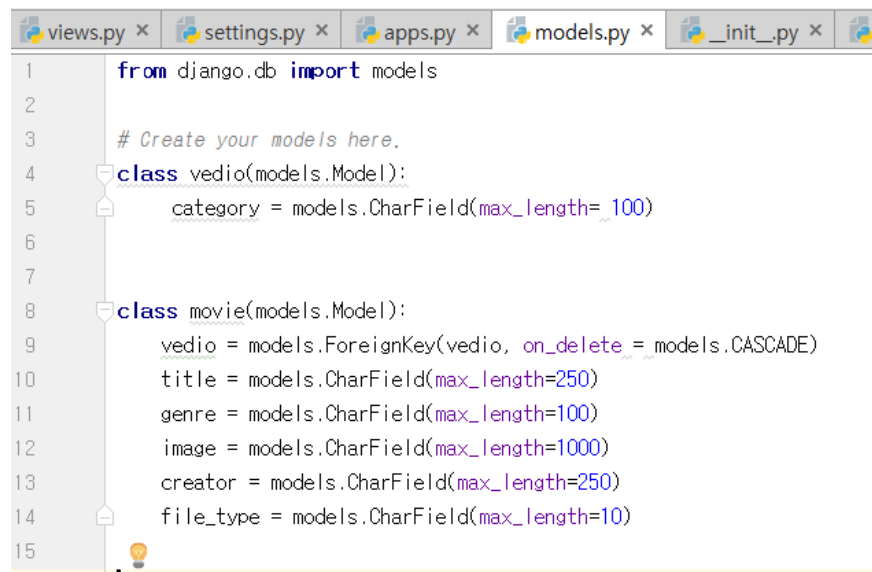
# Python Django 시작하기

## 6. 데이터베이스 Model 만들기

Model : 들어오는 데이터를 담는 일종의 형식, 체계

ex) 영화를 업로드 하는 사이트라면 영화의 정보에 대해서 가지고 있어야함

-> 영화(감독, 주연배우, 개봉연도, ...) 등의 정보들을 저장해야 하는데 이 틀을 만들어 주는 것이라고 생각하면 쉬움



```
1 from django.db import models
2
3 # Create your models here.
4 class vedio(models.Model):
5     category = models.CharField(max_length=100)
6
7
8 class movie(models.Model):
9     vedio = models.ForeignKey(vedio, on_delete=models.CASCADE)
10    title = models.CharField(max_length=250)
11    genre = models.CharField(max_length=100)
12    image = models.CharField(max_length=1000)
13    creator = models.CharField(max_length=250)
14    file_type = models.CharField(max_length=10)
15
```

자신이 만든 app의 model.py에서 그 model을 작성해줘야 함

-> 하나의 entity = 하나의 class

-> entity 안의 column = class 안의 variable

따라서 model을 만들려면 그에 맞춰 class에 variable 을 생성해주면 됨!

ForeignKey ... = 두 entity간의 연결을 의미함

여기서 vedio의 종류에 movie가 속하므로

vedio가 PK, movie가 FK / CASCADE 설정



# Python Django 시작하기

## 6. 데이터베이스 Model 만들기

이제 수정한 것을 DB에게 알려주어야 함

Parent website setting.py 에서 INSTALLED\_APP에 우리가 만든 APP을 추가  
->넣어야 할 문구는 app파일의 apps.py에 나와 있음

```
views.py x settings.py x apps.py x
1 from django.apps import AppConfig
2
3
4 class VedioConfig(AppConfig):
5     name = 'vedio'
6
```

```
INSTALLED_APPS = [
    'vedio.apps.VedioConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

# Python Django 시작하기

## 6. 데이터베이스 Model 만들기

후에 cmd창에 `>python manage.py makemigrations app이름`

내가 만들어 놓은 것을 불러옴

```
C:\Users\강민주\Desktop\website>python manage.py makemigrations vedio
Migrations for 'vedio':
  vedio\migrations\0001_initial.py:
    - Create model movie
    - Create model vedio
    - Add field vedio to movie
```

`>python manage.py sqlmigrate app이름 migration번호`

Sql에 만든 형식대로 table이 형성

```
C:\Users\강민주\Desktop\website>python manage.py sqlmigrate vedio 0001
BEGIN;
--
-- Create model movie
CREATE TABLE "vedio_movie" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "title" varchar(250) NOT NULL, "genre" varchar(100) NOT NULL, "image" varchar(1000) NOT NULL, "creator" varchar(250) NOT NULL, "file_type" varchar(10) NOT NULL);
--
-- Create model vedio
CREATE TABLE "vedio_vedio" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "category" varchar(100) NOT NULL);
--
-- Add field vedio to movie
ALTER TABLE "vedio_movie" RENAME TO "vedio_movie_old";
CREATE TABLE "vedio_movie" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "title" varchar(250) NOT NULL, "genre" varchar(100) NOT NULL, "image" varchar(1000) NOT NULL, "creator" varchar(250) NOT NULL, "file_type" varchar(10) NOT NULL, "vedio_id" integer NOT NULL REFERENCES "vedio_vedio" ("id"));
INSERT INTO "vedio_movie" ("genre", "file_type", "title", "id", "vedio_id", "creator", "image") SELECT "genre", "file_type", "title", "id", NULL, "creator", "image" FROM "vedio_movie_old";
DROP TABLE "vedio_movie_old";
CREATE INDEX "vedio_movie_eaac28d5" ON "vedio_movie" ("vedio_id");
COMMIT;
```

# Python Django 시작하기

## 6. 데이터베이스 Model 만들기

후에 `>python manage.py migrate`

내가 바꾼 것이 website에 적용됨

```
C:\Users\강민주\Desktop\website>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, video
Running migrations:
  Applying video.0001_initial... OK
C:\Users\강민주\Desktop\website>
```

앞으로 model 코드를 수정하게 된다면

`>python manage.py makemigrations app이름`

`>python manage.py migrate`

이 두가지만 cmd창에 쳐주면 변화가 정상적으로 적용됨

# Python Django 과제

오늘 한 것을 이용해 음악 스트리밍 기능이 있는 사이트를 만들기!  
오늘 배운 부분까지 ( view, model) 모두 적용해서 사이트 만들어 보기  
DB migrate 한거도 인증샷 보내기