장고는 파이썬 웹 프레임워크다
MVT (Model View Template) 구조. MVC랑 비슷하지?

장고는 파이썬 가상 환경에서 돌린다. 글로벌로 안돌린다.
버전 안맞으면 그때마다 파이썬 버전 바꿔야되는데 그럴 수가 없잖아

가상환경 만들기: $ python -m venv name
구동: $ name₩Scripts₩activate
비활성화: deactivate

이후 가상환경 내에서 pip install django로 장고를 설치해준다. 밖에서 하는거랑 다름 ㅇㅇ
파이썬 구동하고 import django print(django.get_version()) 해서 설치 확인 가능

나중에 장고 restapi부터 해서 이거저거 설치할 일 많은데 항상 잊지말고 가상환경을 실행시켜줘야된다. 로컬이랑 다르다 이말이야.

프로젝트 만드는 법: $ django-admin startproject (프로젝트이름)
앱 만드는법: $₩프로젝트 폴더₩python manage.py startapp (앱이름)
서버 켜는 법: $₩프로젝트 폴더₩python manage.py runserver (portnum)

128.0.0.1:port 로 접속 할 수 있음. 디폴트 포트는 8000

#프로젝트 구조

- 들어가자 마자 있는 manage.py: 얘가 main같은 느낌. 앱 생성, db작업, 서버구동 얘로 함

- 프로젝트 명과 동일한 디렉토리 안에 있는 놈들(이하 프로젝트). 대충 핵심부 같은 느낌이 든다.
_init__.py : 그냥 빈파일인데 얘부터 시작한다는 뜻. 파이썬이 패키지를 인식하게 함
settings.py: 이 사이트 자체의 설정을 다룸. 어플리케이션 등록, static파일 위치, db등 설정 가능
urls.py: URL(Uniform Resource Locator: 인터넷에서 자원의 위치)이랑 뷰의 연결을 다룸.
wsgi.py 장고 앱과 웹서버의 연결을 다룸. 자세힌 모르겠음.
프로젝트 폴더는 view.py가 없음. 쓰고 싶으면 만들어줘야함.

- 다른 디렉토리들은 각각의 앱들임
migration폴더: 모델 수정할 때 migration 만들면 db를 자동으로 업그레이드 해줌. 그 migration 들 저장되는 곳.

__init.py__:마찬가지로 패키지 인식용
urls.py: 처음에 앱 만들면 없다. 만들어줘야됨.
나머지는 뭐 메인 폴더랑 비슷함.

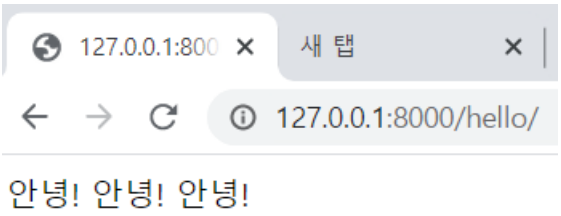
#setting.py의 내용

INSTALLED_APPS: 앱 만든 후엔 setting.py열고 INSTALLED_APPS에서 등록해줘야 함. 앱 이름이 hello라 치면 'hello.apps.HelloConfig',를 넣어줘야함. 참고로 HelloConfig는 앱 디렉토리의 apps.py 에 있는 클래스임.
DATABASES에서 DB설정 만질 수도 있다 ㅇㅇ 근데 DB가 뭔지 아직 몰라 ㅋㅋ 디폴트는 SQLite 라고 함. python에서 기본 지원된다는듯?
TIME_ZONE: 디폴트가 'UTF'일텐데, 'Asia/Seuol'로 바꿔주셈. UTF+8인가 그럼
DEBUG: 에러 발생했을 때 디버깅 로그 표시되게 함. 평소엔 보안 때문에 False해두셈. 지금은 공부땜에 True

나머지는 아직 볼 일 없다.

~~~homework_20201229~~~

디렉토리 만들고 hello앱 만들어줌

(py manage.py startapp hello)

앱 후킹하기

앱을 만들었기 때문에 프로젝트의 settings.py에서 신고식을 해줘야한다.

```
INSTALLED_APPS += ['hello.apps.HelloConfig',]
```

INSTALLED_APPS에다가 'hello.apps.HelloConfig',를 써주자. (hello/apps.py에 있는거임)

신고식은 프로젝트의 urls.py에서도 해야한다.

```
from django.urls import path
```

urls.py 그냥 열면 include가 없다. path 옆에다가 붙혀주자.

```
from django.urls import path, include
```

이캐이캐 ㅇㅇ

```
urlpatterns += [path('hello/', include('hello.urls')),]
```

그리고 path('hello/', include('hello.urls')), 해서 메인의 url이 hello의 url를 참조할 수 있도록 하자. 참고로 hello에 아직 urls 없다. 나중에 만들어줄거다.

hello/views.py

```
from django.http import HttpResponse
def printHello(request):
    return HttpResponse("안녕! 안녕! 안녕!")
```

HttpRespose를 통해 '안녕! 안녕! 안녕!'을 출력하는 printHello를 만든다. 얘를 뭐라 불러야 할까?

hello/urls.py

앱을 만들었을 때 urls.py는 없다. 따로 만들어줘야한다.

```
from django.urls import path
from . import views
```

얘도 path를 쓸거니까 선언 해준다. 그리고 from . import views 는 hello 안에 있는 view.py를 쓰겠다는거다 ㅇㅋ?

```
urlpatterns = [path('', views.printHello, name='printHello'),]
```

그리고 urlpatterns는 이렇게 작성해준다. views에서 아까 선언했던 printHello 쓰는거다.

name='printHello'는 무슨 의미인지 잘 모르겠다. 저거 이상하게 바꿔도 작동은 되던데...
~~~

#127.0.0.1:8000/hello/



잘 되는걸 확인할 수 있다. 그런데 여기서 끝이 아니다. 127.0.0.1:8000들어가면 404뜬다. 얘도 뷰를 만들어줘야한다.

project/views.py 및 urls.py

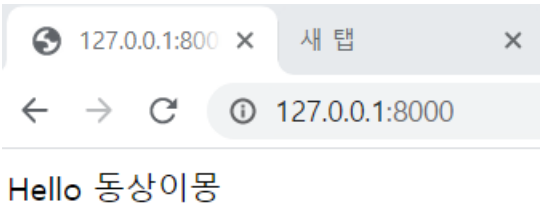
상기했듯 project/views.py 처음엔 없는거다. 만들어줘야한다. 어차피 한줄 출력이라 모양새는 아까 hello/view.py랑 비슷하다.

```
def printDongSang(request):
    return HttpResponse("Hello 동상이몽")
```

대충 이캐 해주고 project/urls.py가 지금은 hello만 후킹해주고 있다. 지꺼도 써야되니까 후킹해주자.

```
from . import views
urlpatterns += [path('', views.printDongSang, name='printDongSang'),]
```

127.0.0.1:8000



ㅇㅇ 잘 됨

@@@ 이미지 띄워주기 @@@

루트 디렉토리에 image 폴더 만들고 커여운 안녕로봇 삼형제의 gif파일을 넣어줬다.
그래서 127.0.0.1:8000/image/AnnoyoTron.gif 그냥 하면 나오느냐? 안나온다. 우리의 프로젝트는 아직 미디어파일을 사용할 준비가 안됐다!

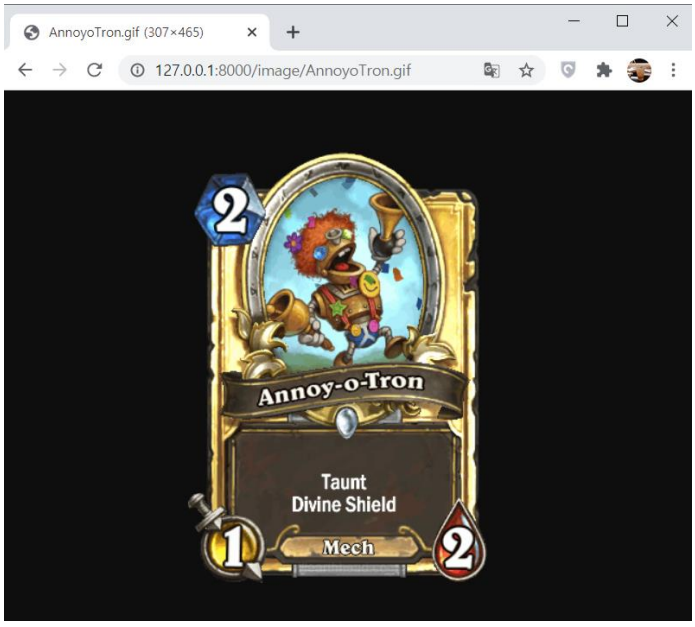
#settings.py

```
import os
MEDIA_URL = '/image/'
MEDIA_ROOT = os.path.join(BASE_DIR,'image')
```

#project/urls.py

```
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [. . .
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

http://127.0.0.1:8000/image/AnnoyoTron.gif



ㅋㅋ 교회누나마렵네 ㅋㅋ

이 외에 models 관련 공부도 했는데 djangorestframework에서 용법이 확 바뀌고 헷갈리고 난리부르스이므로 장고 찍먹은 여기까지 하도록 하겠다.