<Django 개발환경 만들기> 관련 사이트 참고

python3 —version 을 통해 설치되어 있는지 확인

sudo pip3 install virtualenvwrapper

.bash\_profile 파일 끝에 특정 코드 붙여넣기

which python

which virtualenvwrapper.sh 을 통해 위치 찾아서 코드 수정

source ~/.bash\_profile

mkvirtualenv 가상환경 이름

가상환경으로 들어가 pip3 install django

**deactivate**: 활성화된 파이썬 가상환경을 비활성화

**workon**: 사용가능한 가상환경 목록을 보여줌

**workon name\_of\_environment**: 특정 파이썬 가상환경을 활성화

**rmvirtualenv name\_of\_environment**: 특정 환경을 제거한다.

가상 개발환경에서 실행

플젝 생성: django-admin startproject 플젝이름

이때 플젝이름 yousite으로 짓기

yousite에 들어가서 윤진 깃허브 파일 clone 하기

https://github.com/skewondr/my-first-website.git

application 만들기: python3 manage.py startapp 앱이름

이때 앱이름 classifier로 짓기

setting.py 수정: INSTALLED APPS에 앱이름 추가

플젝 urlpattern, 앱 urlpattern(새로 생성) 채우기. 플젝 urlpattern에서는 요청 url을 앱urls.py로 넘기는 부분을, 앱 urlpattern은 요청 url을 view로 넘기는 부분을 담당한다.

model 만들기

**\_\_init\_\_.py**: 파이썬에게 이 디렉토리를 하나의 파이썬 패키지로 다루도록 지시

**settings.py**: 웹사이트의 모든 설정을 포함. app등록, static files위치, database 세부설정

**urls.py**: url-view의 연결을 지정해줌

**wsgi.py**:장고 어플리케이션이 웹서버와 연결 및 소통하는 것을 도움.

**manage.py**:app생성, 데이터베이스와 작업, 개발 웹서버 시작하기 위해 사용

DEBUG: 에러 발생시 디버깅 로그 표시 - True는 해킹에 유용하기 때문에 상품화 후 False로 바꾼다.

path('catalog/', include('catalog.urls')) : www.xxxx.com/catalog로 시작되는 request가 오면 catalog/urls.py

를 참조해서 관련된 파일을 mapping하겠다.

python3 manage.py makemigrations 앱이름: 적용하기 전 모든 app에 대한 migration들 생성. model 수정 후에도 적용

python3 manage.py migrate 앱이름: migration들을 실제로 데이터 베이스에 적용. model 수정 후에도 적용

admin.py에 model 등록

**blog/models.py** : 모델 객체를 정의함.

class 이름 첫글자는 무조건 대문자.

웹 서버 실행하기: python manage.py runserver

view.py: 요청(request)을 받으면 특정 html을 리턴

<https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python-keras/>

<https://suwoni-codelab.com/django/2018/03/25/Django-FBV/>

<https://docs.djangoproject.com/en/2.2/topics/db/queries/>

atom indentation 단축키: cmd+alt+l