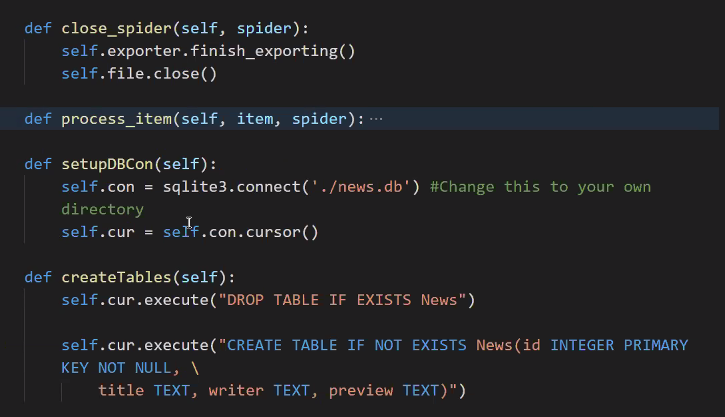
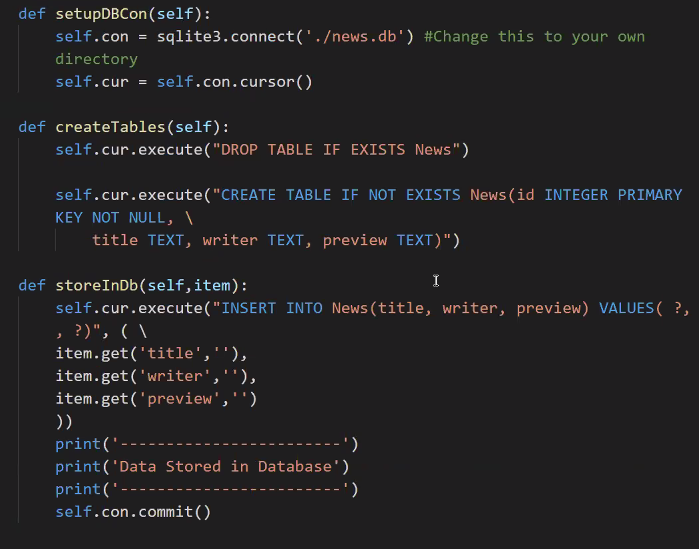
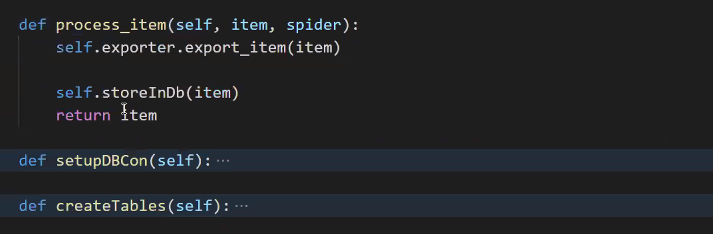
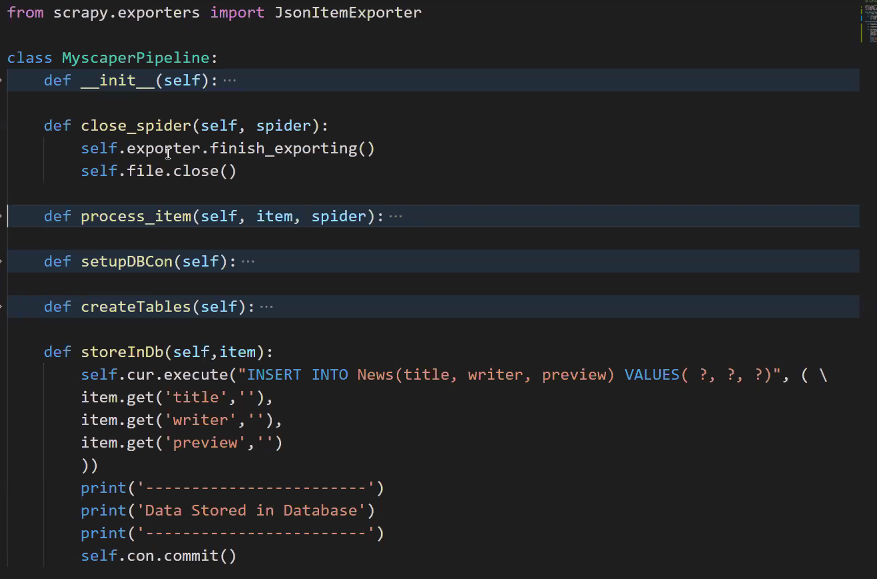


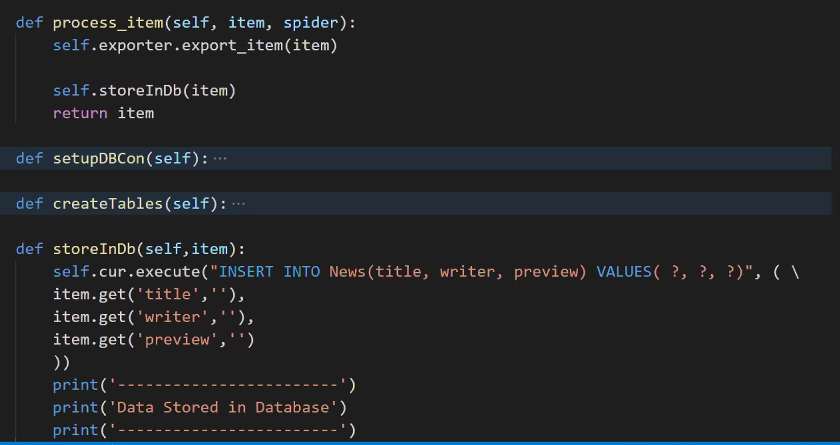
사진 있/없 버전에 따라 dt가 다름🡪dt없으면 사진이 없음

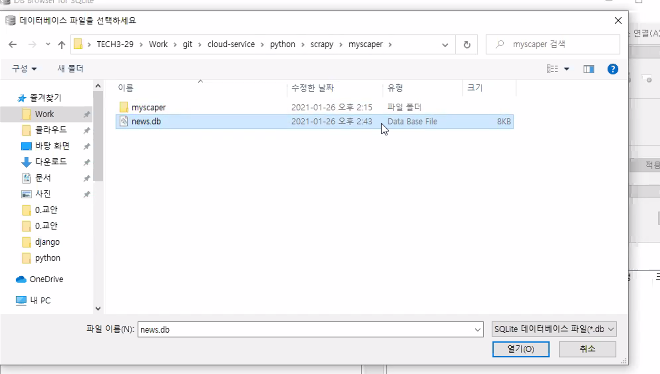
Scrapy pipeline

e

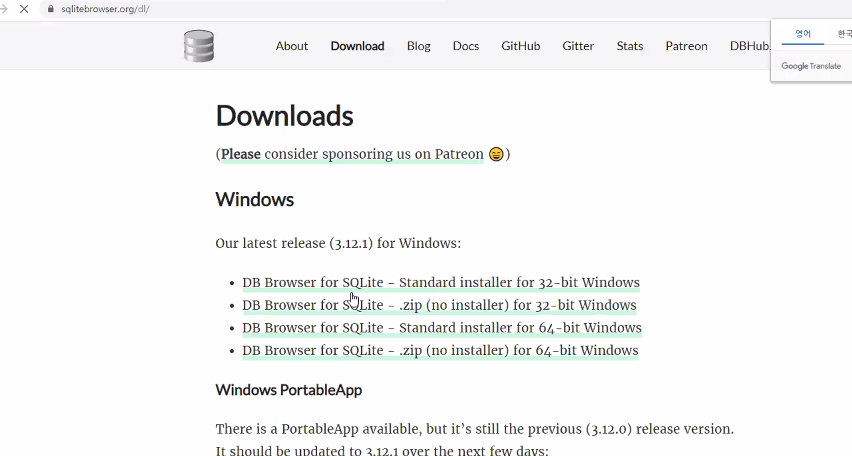




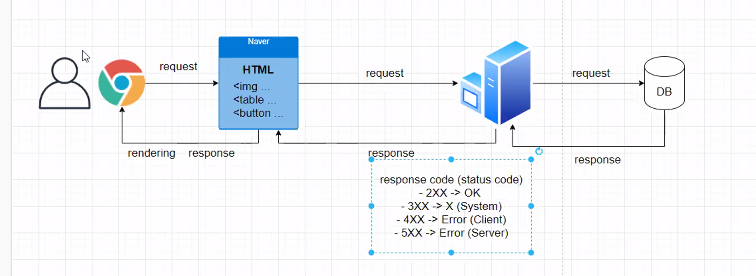
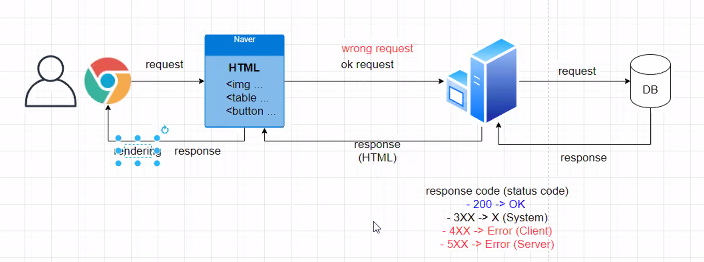


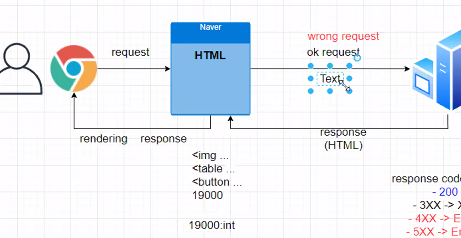






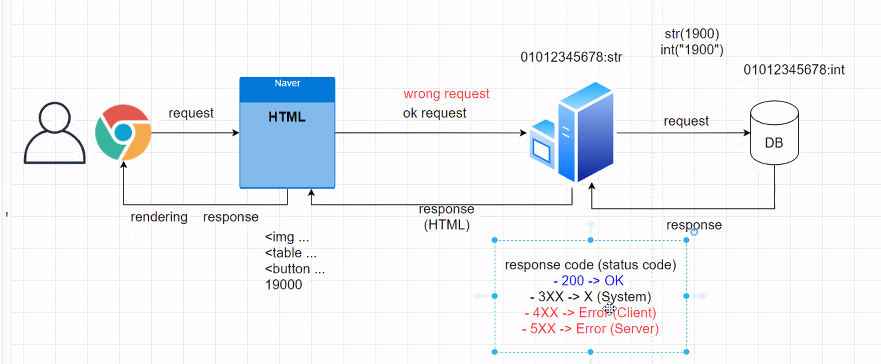
Code 뒤에 파일 이름 적으면 없는 파일이면 파일 새로 만들어준다

html의 대부분 데이터는 문자데이터타입이다!

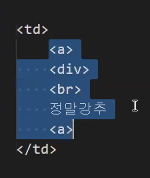


String 은 포매팅 통해서 바꿀 수 있다

서버/클라/DB가 전달하는 데이터 모두 문자형태로 저장되기 때문에 바꾸고싶으면 캐스팅작업 필요!

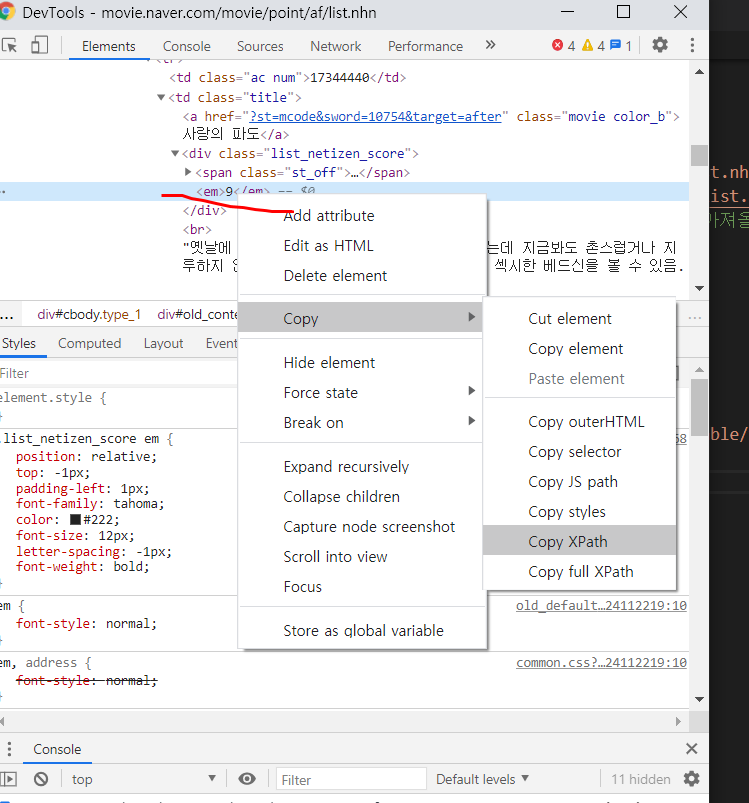
 

🡪

td태그 안에 <a>태그 얘기하는거임;;

//\*[@id="old\_content"]/table/tbody/tr/td[2]/a[1]/text()



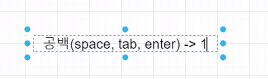


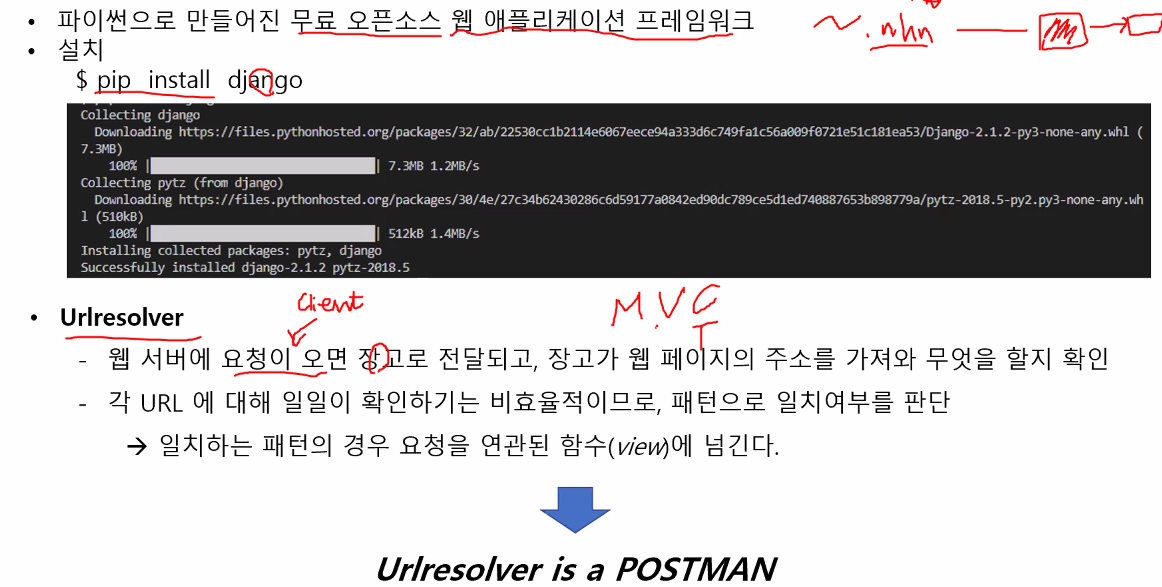
//\*[@id="old\_content"]/table/tbody/tr[1]/td[2]/div/em/text

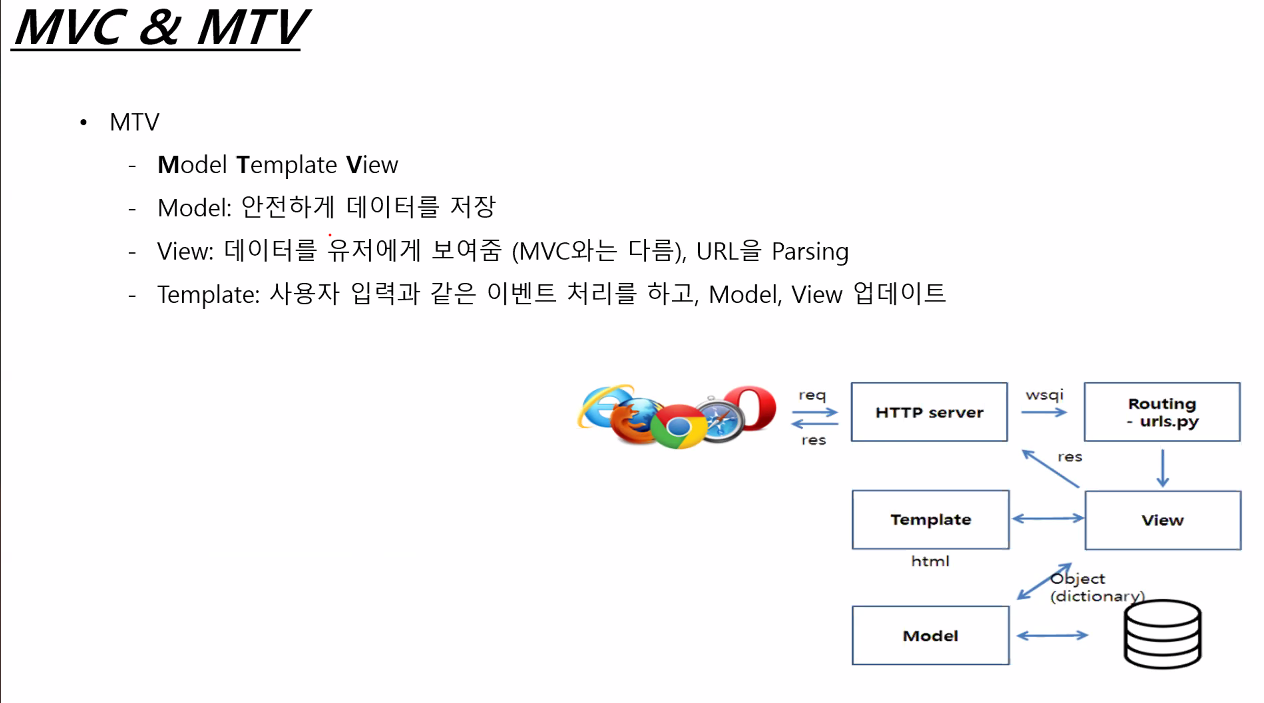
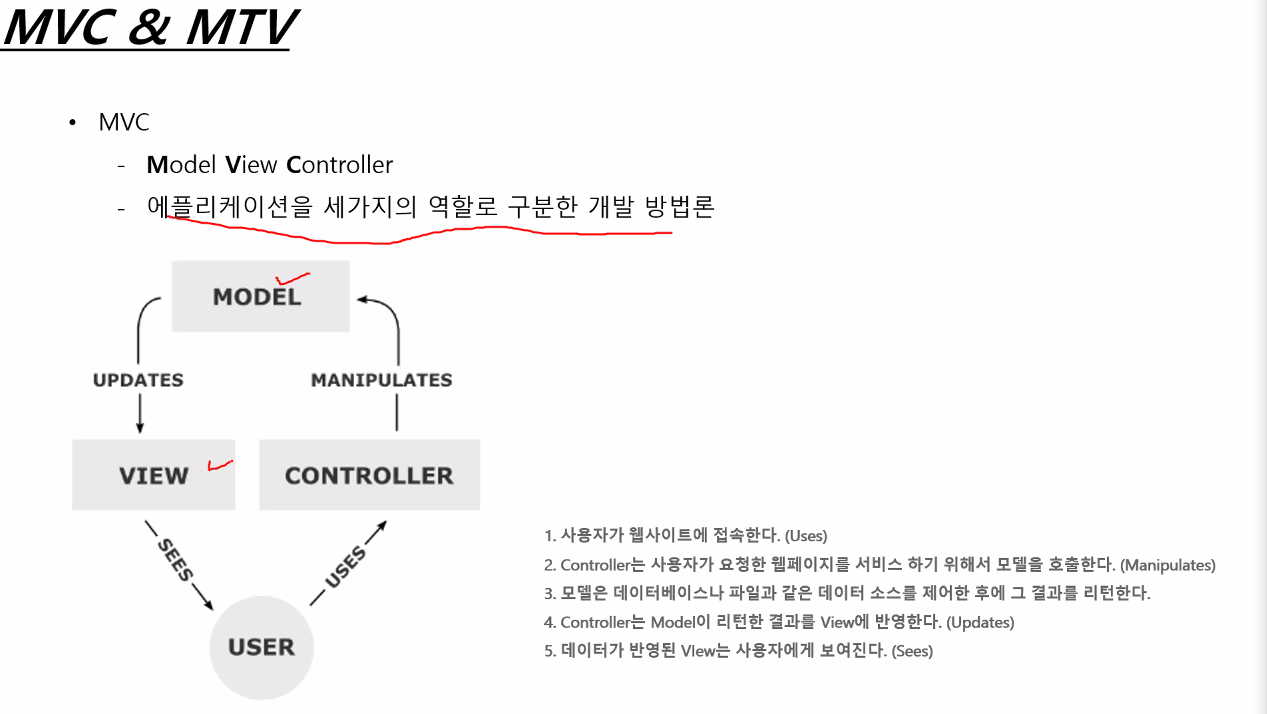
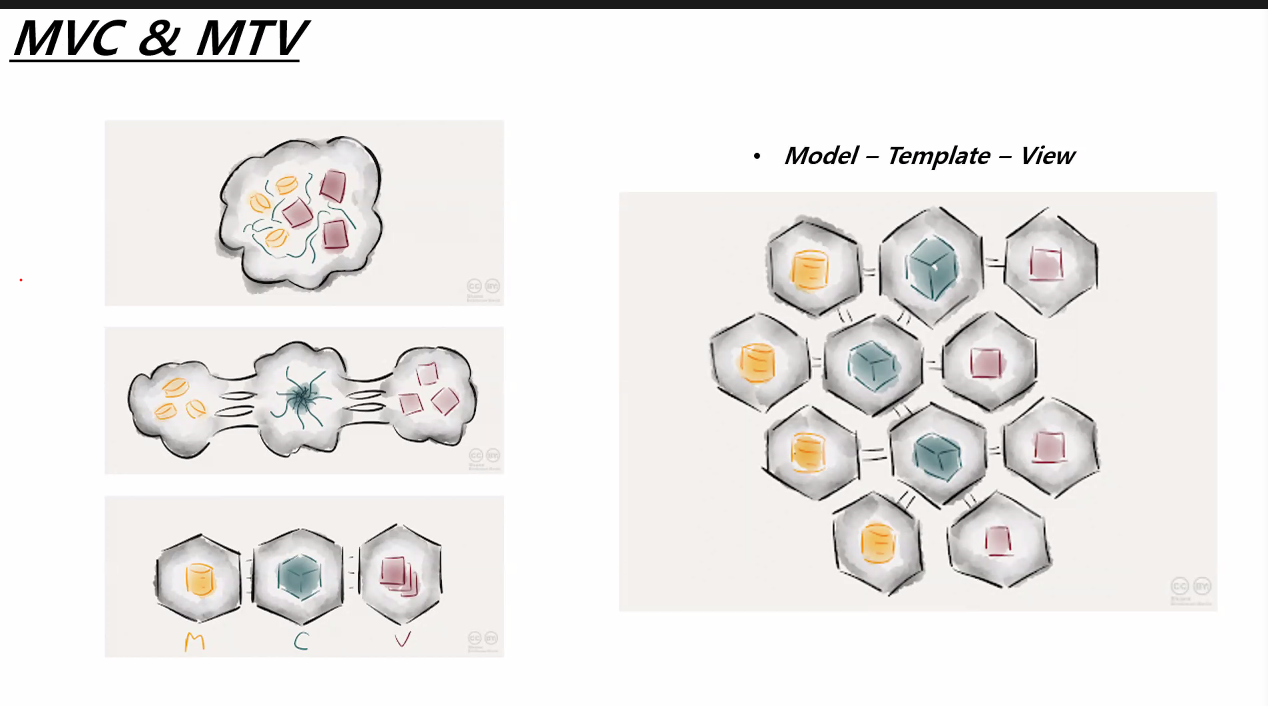
em까지 쓰면 em까지 가져오니까/text해줘야됨

//\*[@id="old\_content"]/table/tbody/tr[1]/td[2]/br

//\*[@id="old\_content"]/table/tbody/tr/td[2]/br/text()

공백은 무조건 1글자만 처리됨





라우팅-> 교통정리하는것

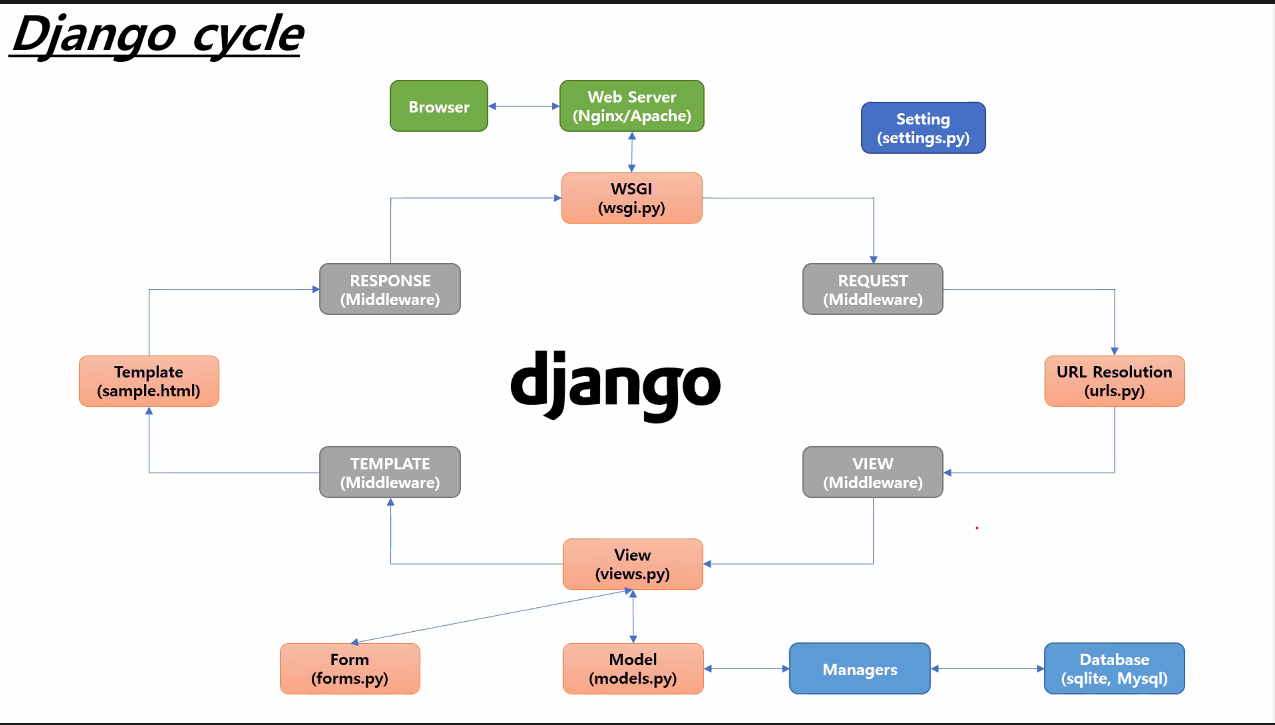
Urls.py->view-> model 데이터 뭘 만들지 호출-> model에서 데이터연동 발생-> 뷰로 반환

데이터베이스 값을html형태로 만들어서 다시 사용자한테 돌려준다.

Mvc/mtv

View 에서 처리된 값은 model생성해서 처리

model에서 가져온 데이터값이 있다면 ㅊ템플릿형태로 만들어서 보여준다.



라우팅 역할로 sum.py 게이트는wsgi

설정에 관련된건 settings

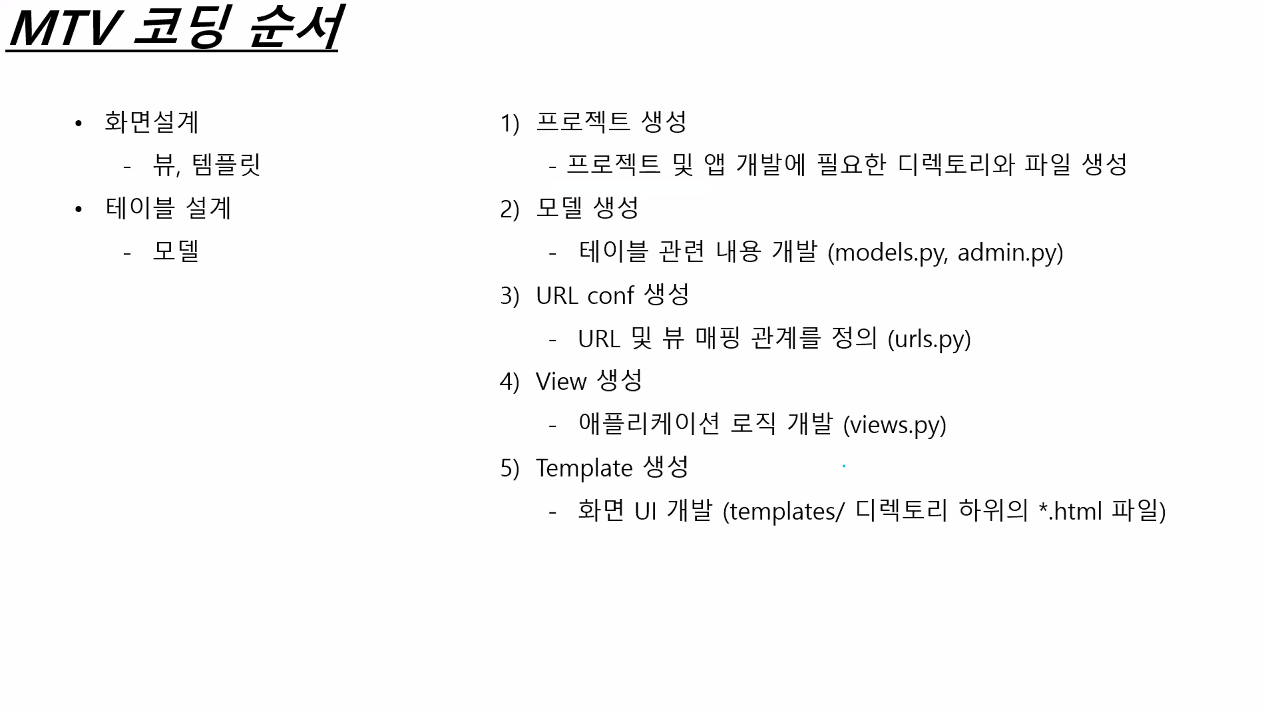
Tmplate는 html 형태

View 안에는 폼데이터,폼 파이썬 파이/모델즈 파이 사용

모델-파랑

뷰 클라이언트 쪽에서 처리

템플릿-파랑

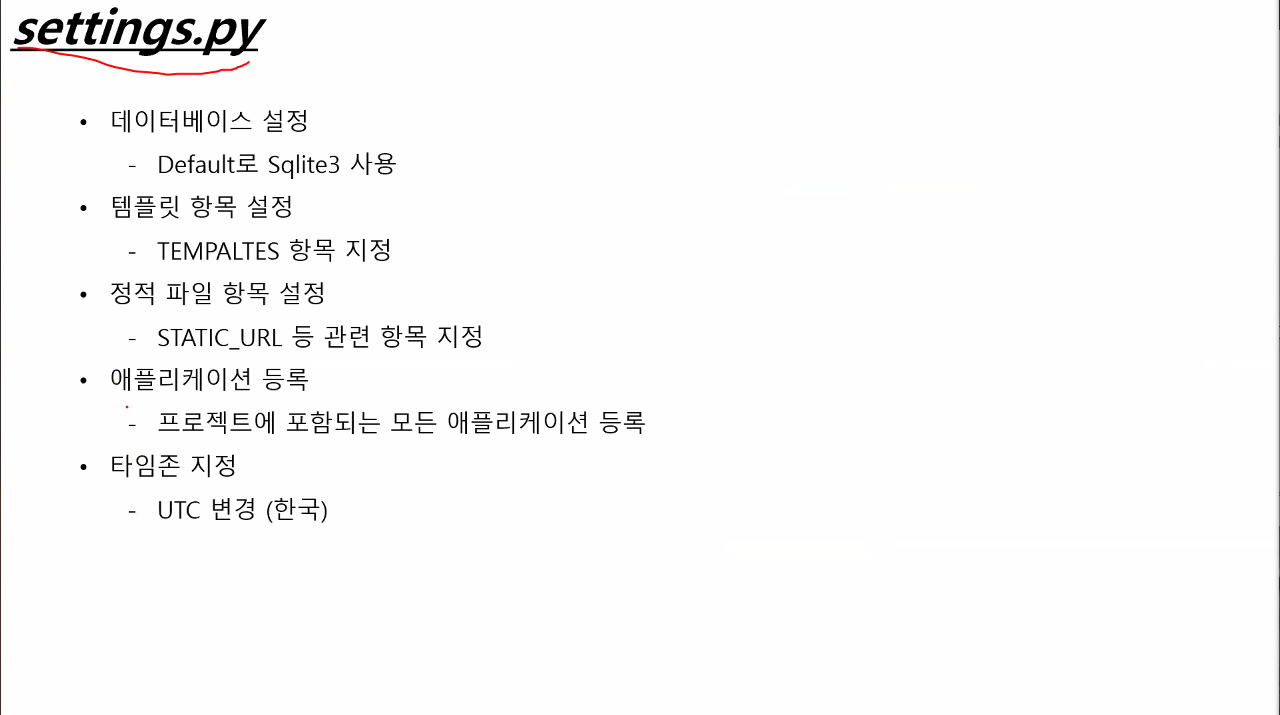


데이터베이스 생성-> 모델생성

url 사용자로부터 요청되는 주소값/매핑관계 정의해놓은 것이 url

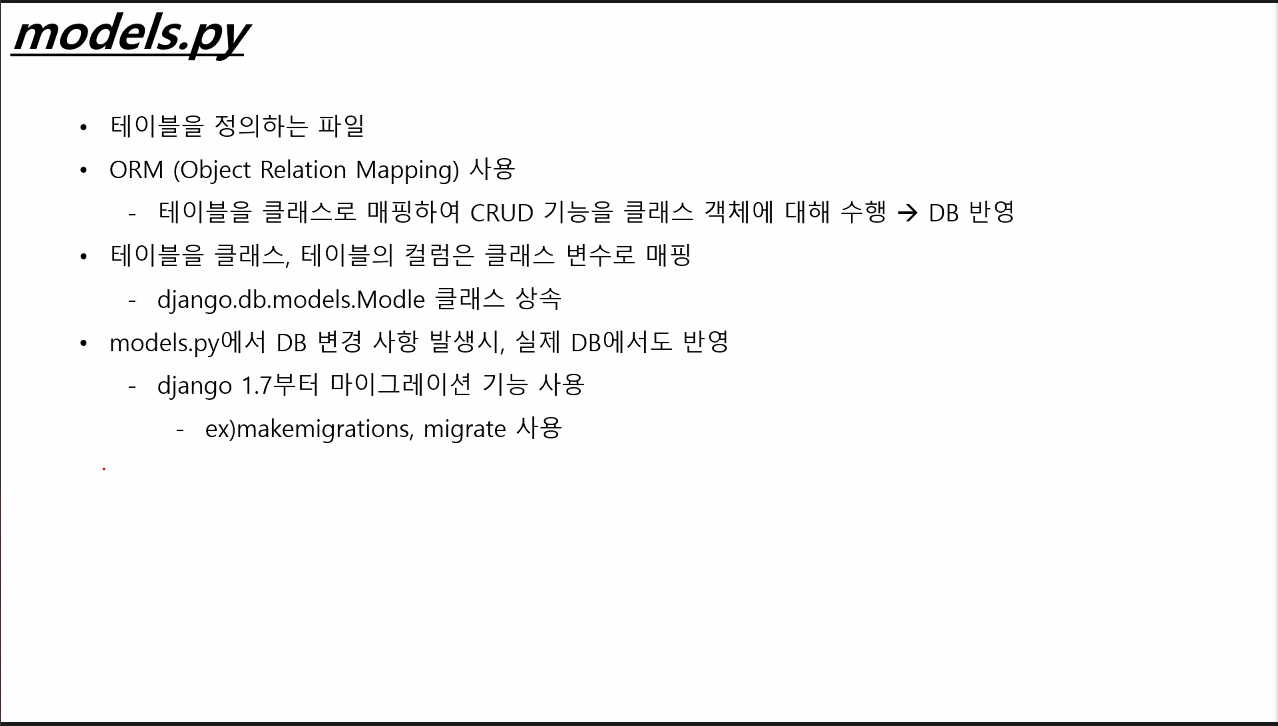
view 어떤 작업/모델 사용할것인지 결정

template 🡪 UI를 가진 형태의 결과물(html)파일로 만들어준다. 서버가 클라에게 응답할때는 모든모든 것 html이라는 파일로 묶어서 전달한다. 그 안에 들어가있는 데이터들의 사용여부는 html문법과 python 섞어서 보냄



프로젝트 하나 만들어서 그 안에 어플리케이션 관리하게 될 것

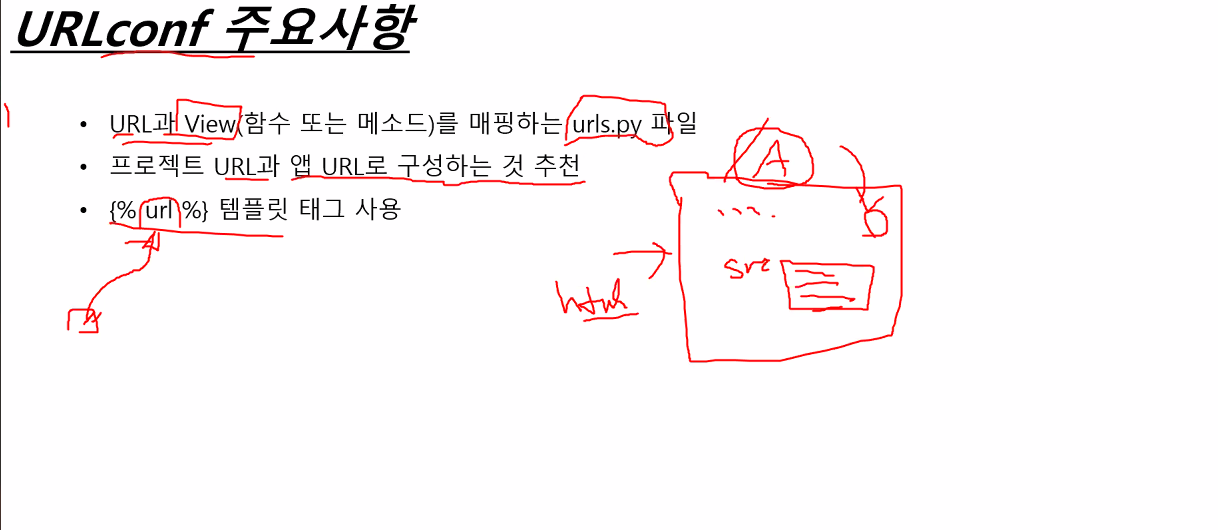
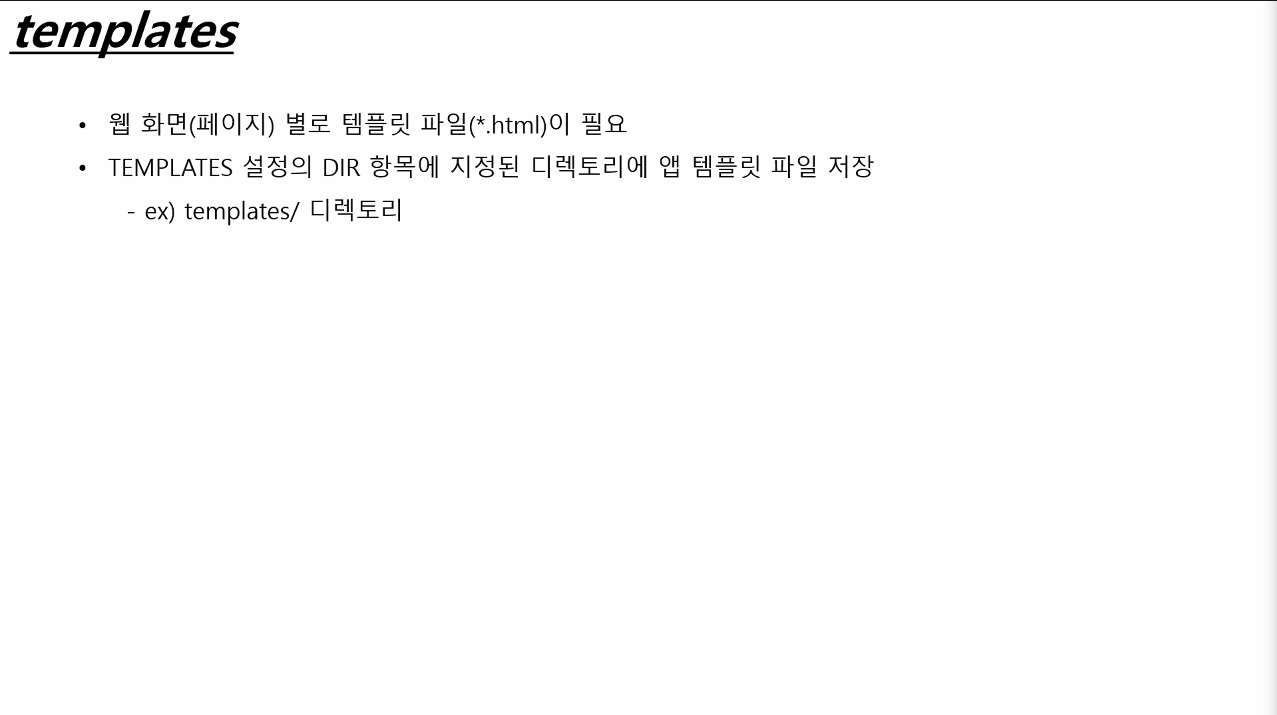
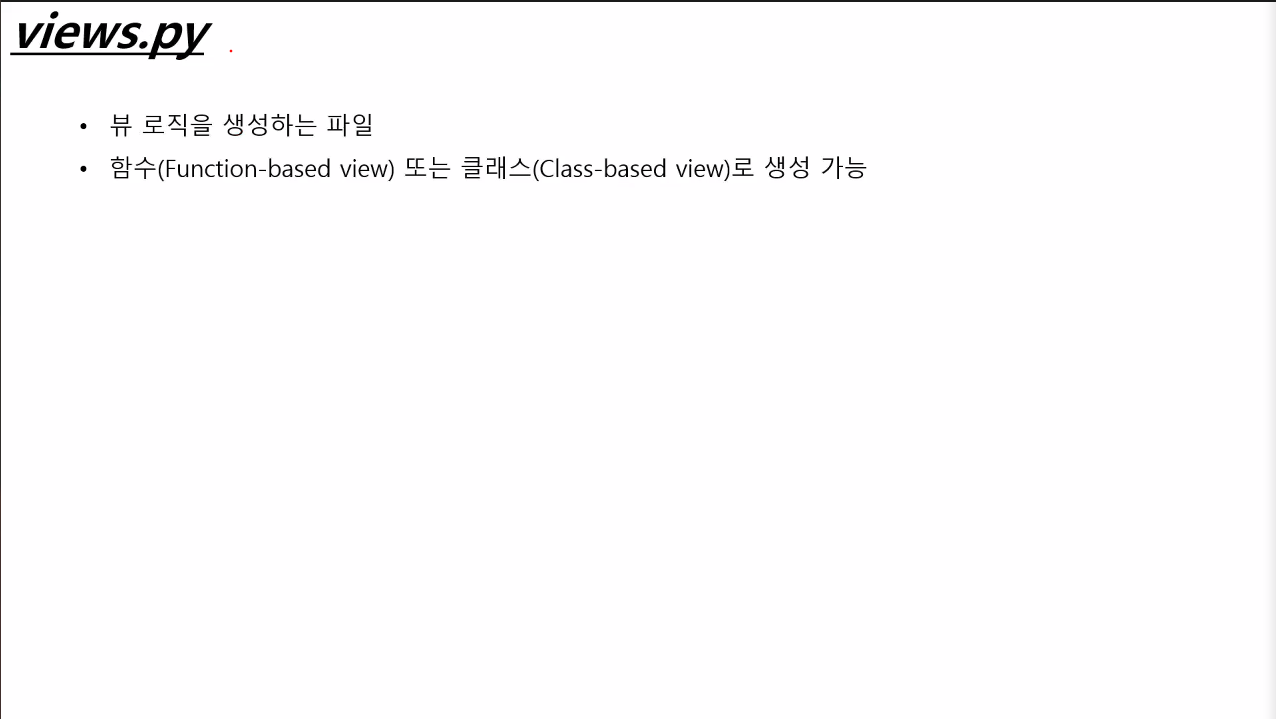
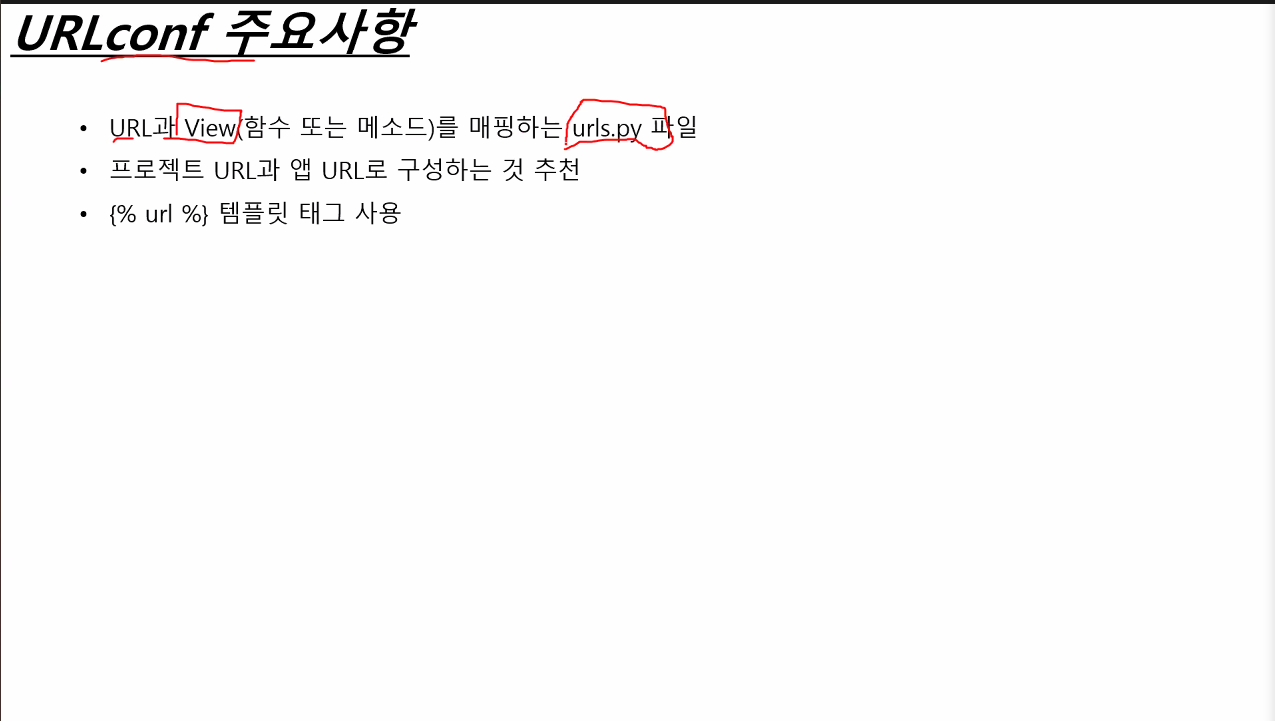
어플리케이션 등록해야 등록되어진 컨텐츠 사용할 것

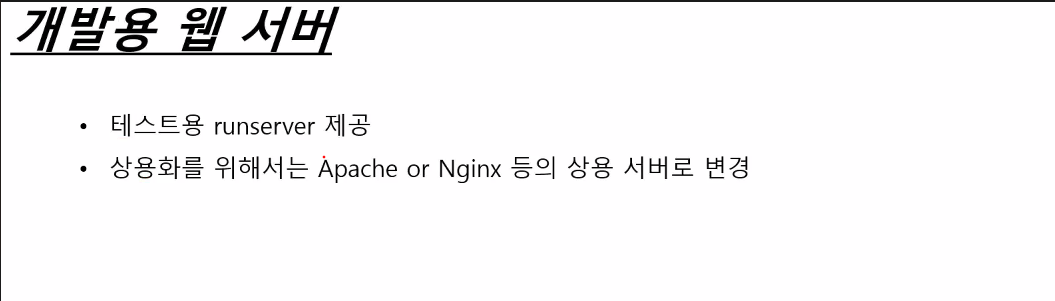
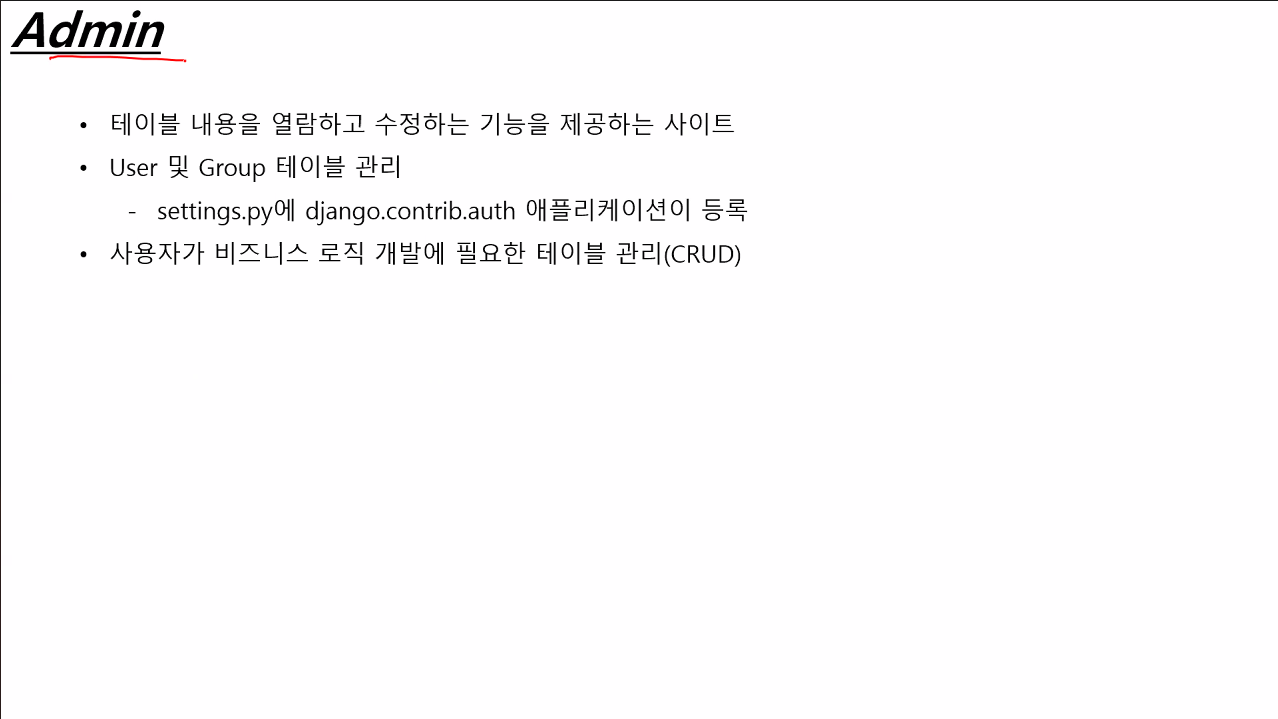


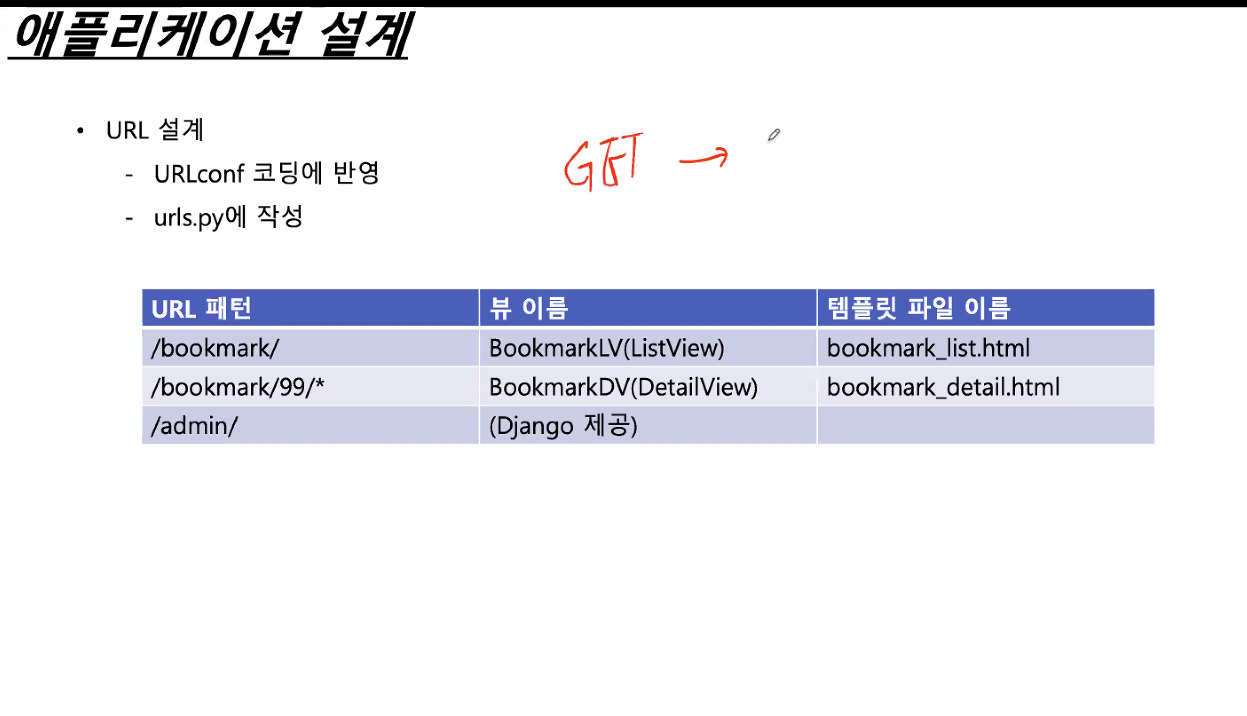
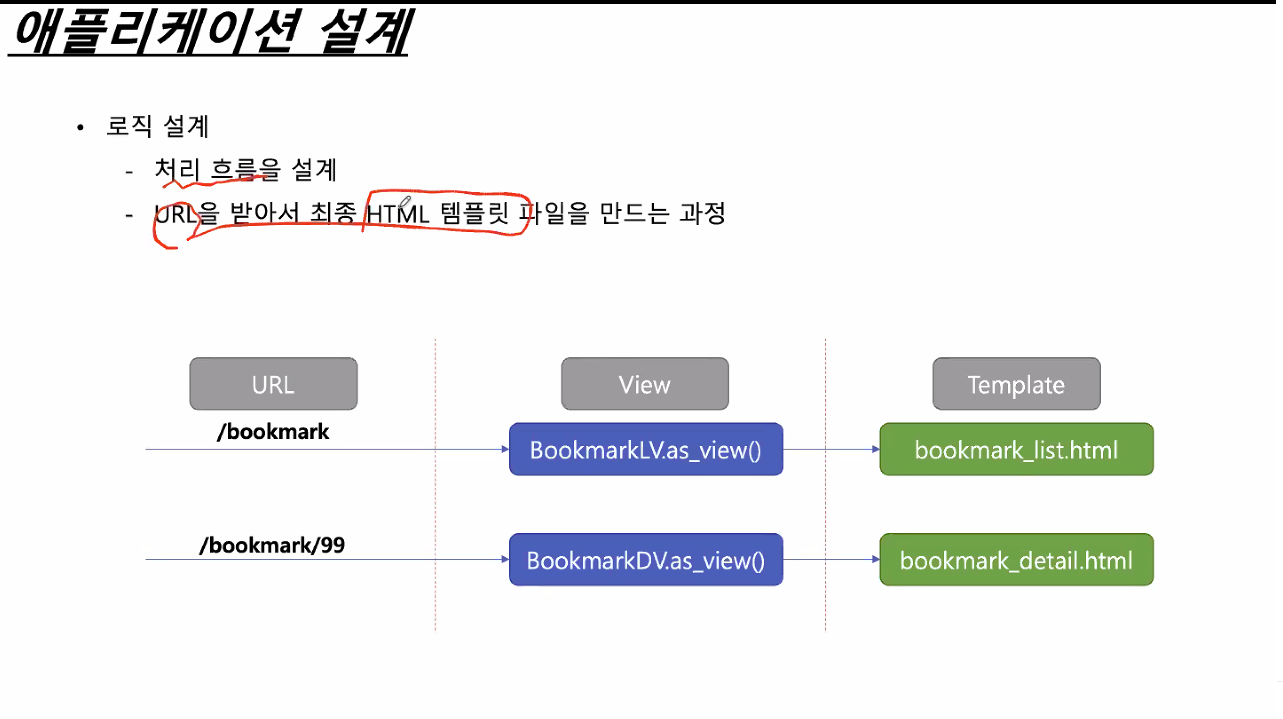
모델안에 포함되어야 할 내용은 테이블 정의할 파일.

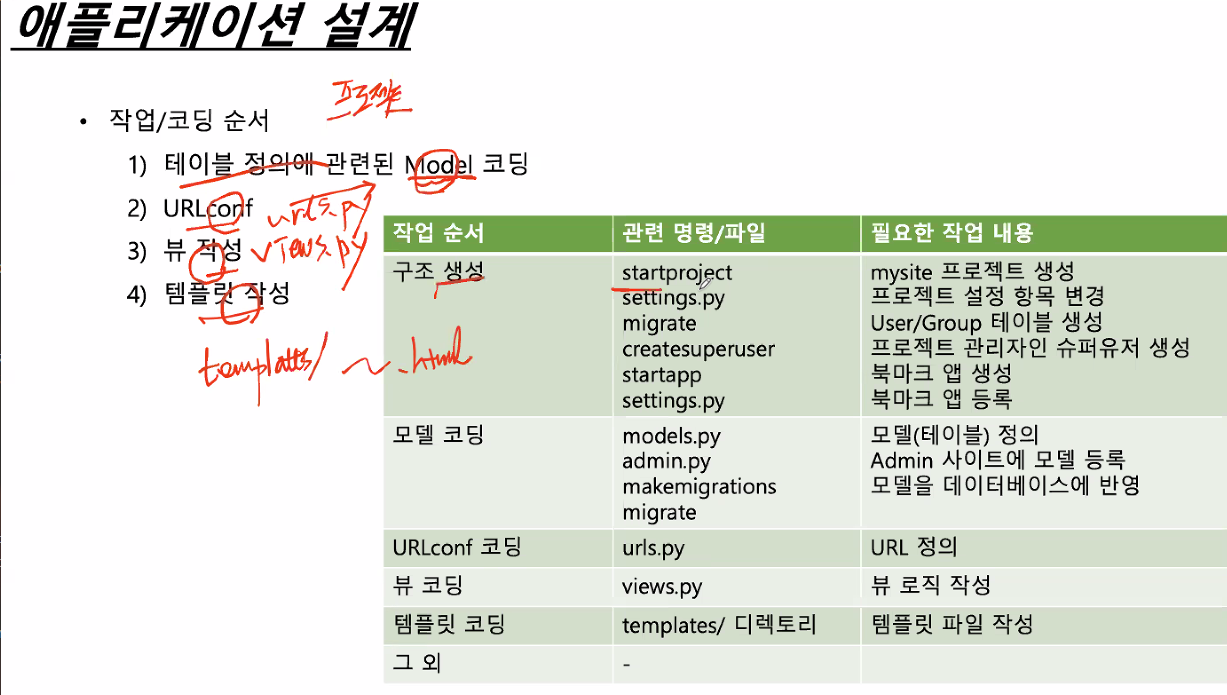
모델은 데이터베이스랑 반드시 연결되어있어야 함

테이블 정의는 데이터타입 같이 지정

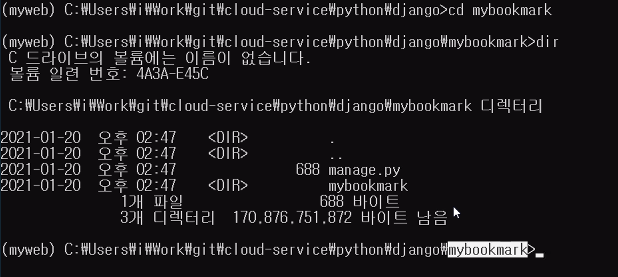
업데이트가 계속되는페이지-> 동적인 페이지

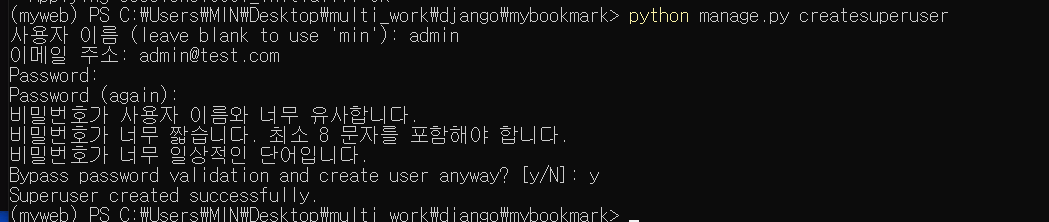
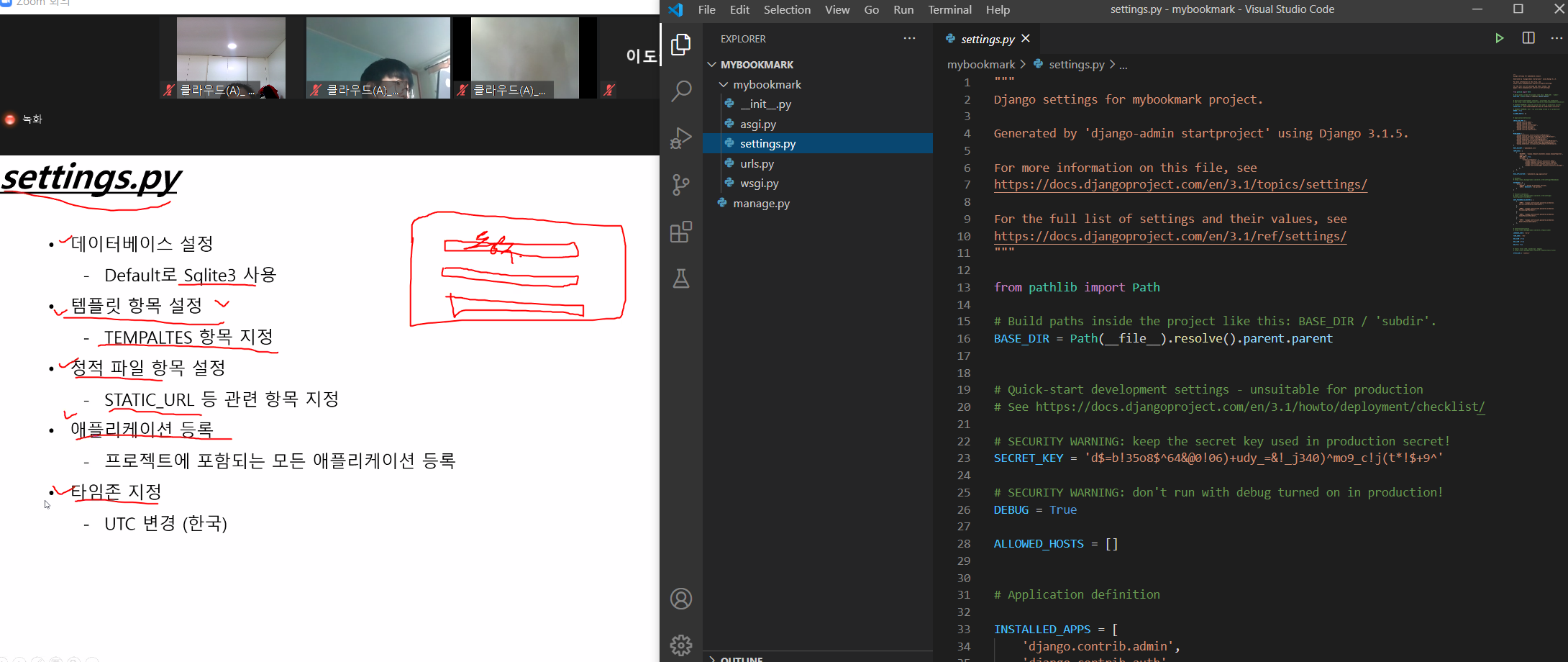
myweb 가상환경 만들어서 django다운받음

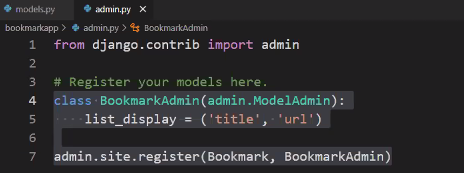
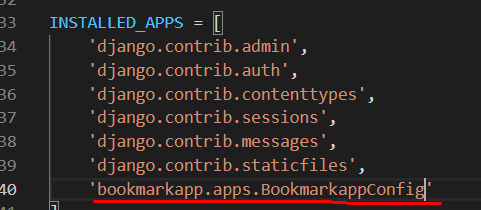


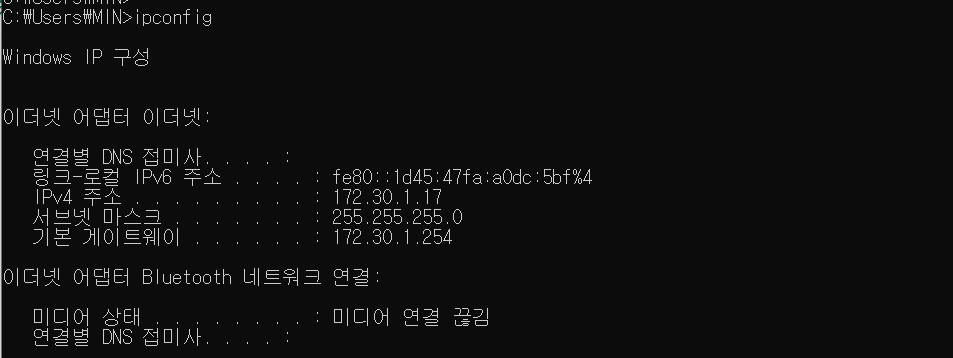




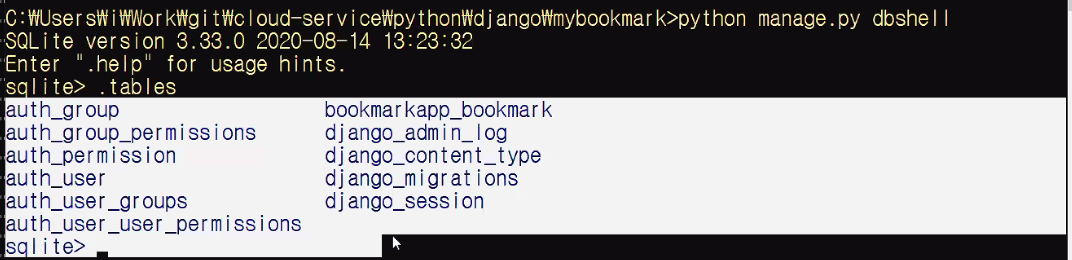


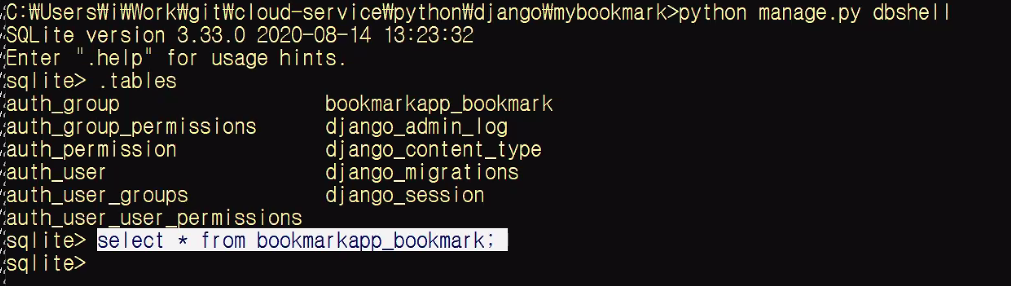
migrate 한 다음에 하는거

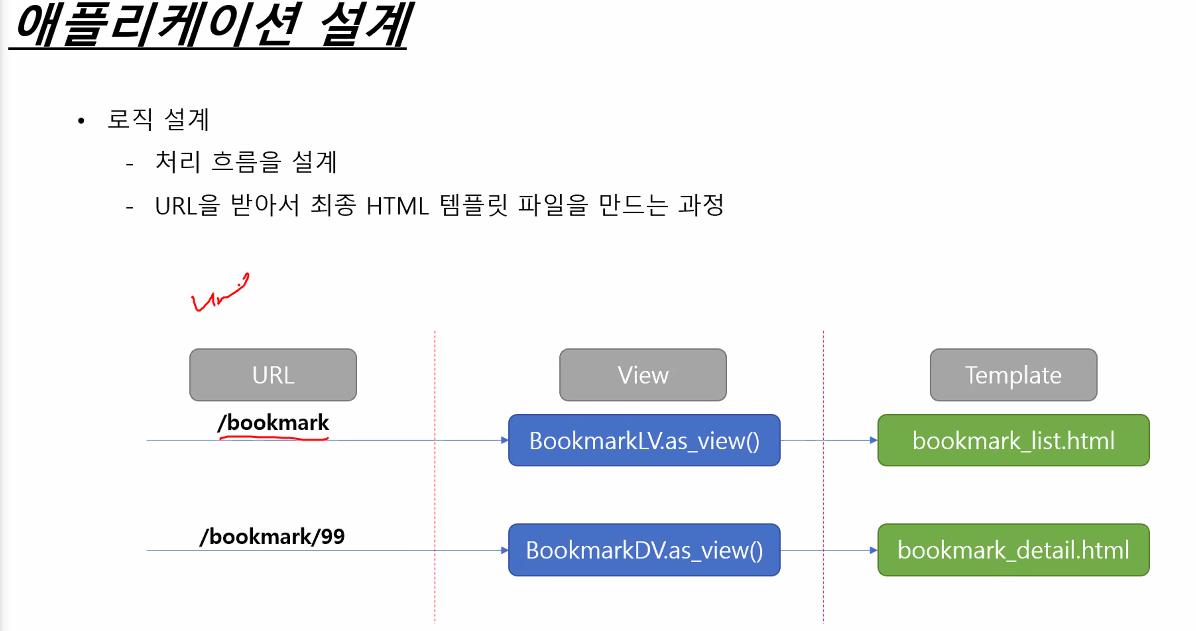


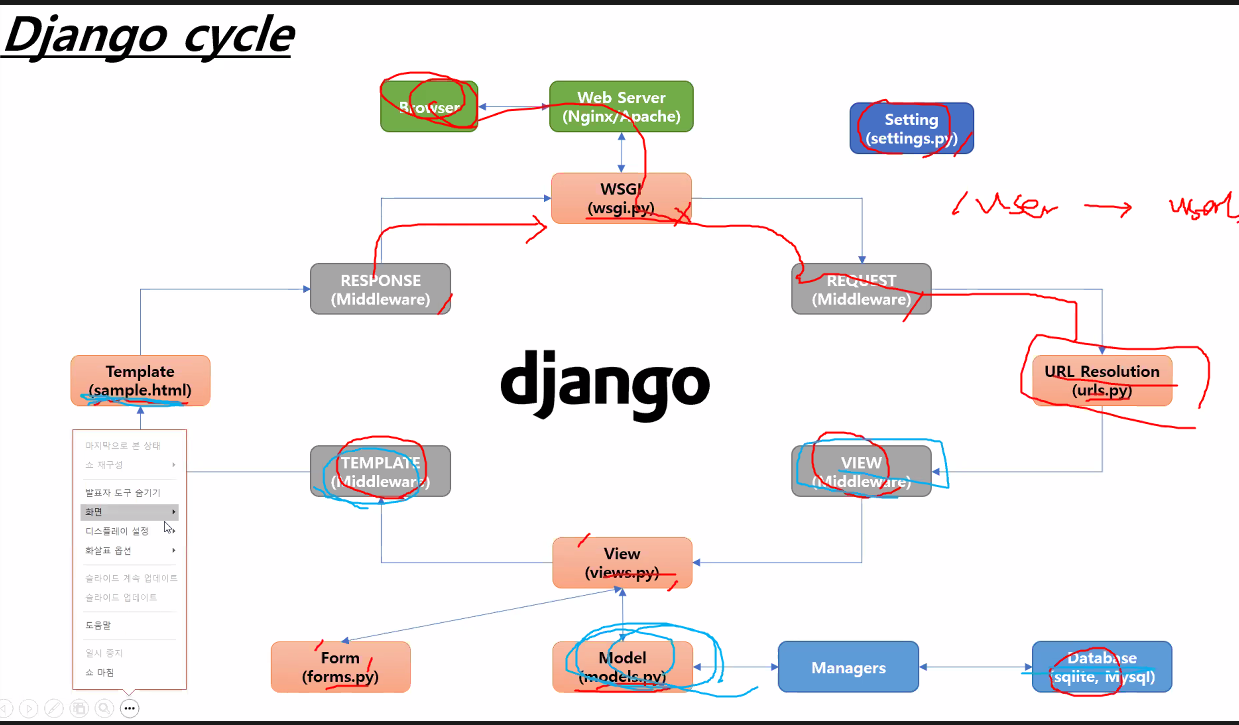


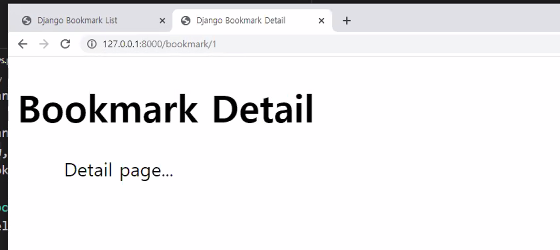


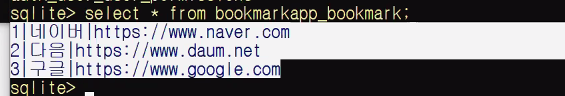


콘솔에서 데이터확인

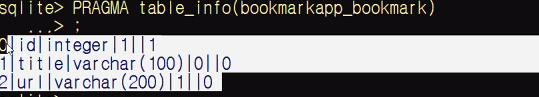
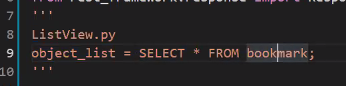


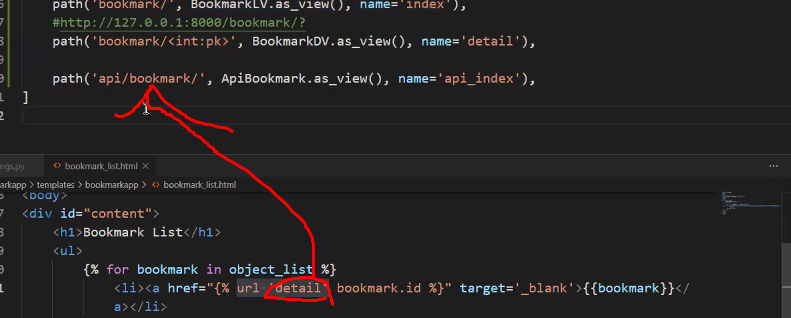


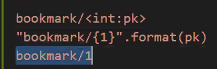


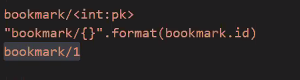


Manage.py있는 폴더 들어가서 dbshell들어간담에 저거하면됨







/html/body/div/div[4]/div/div/div/div/div[1]/table/tbody/tr[1]/td[2]/br

//\*[@id="old\_content"]/table/tbody/tr[1]/td[2]/br