DB & django CR

6번째 세션

NEXT X LIKELION 정성원

MTV 패턴 복습

Model – 데이터베이스 설계

- 데이터베이스에 저장되는 데이터의 영역

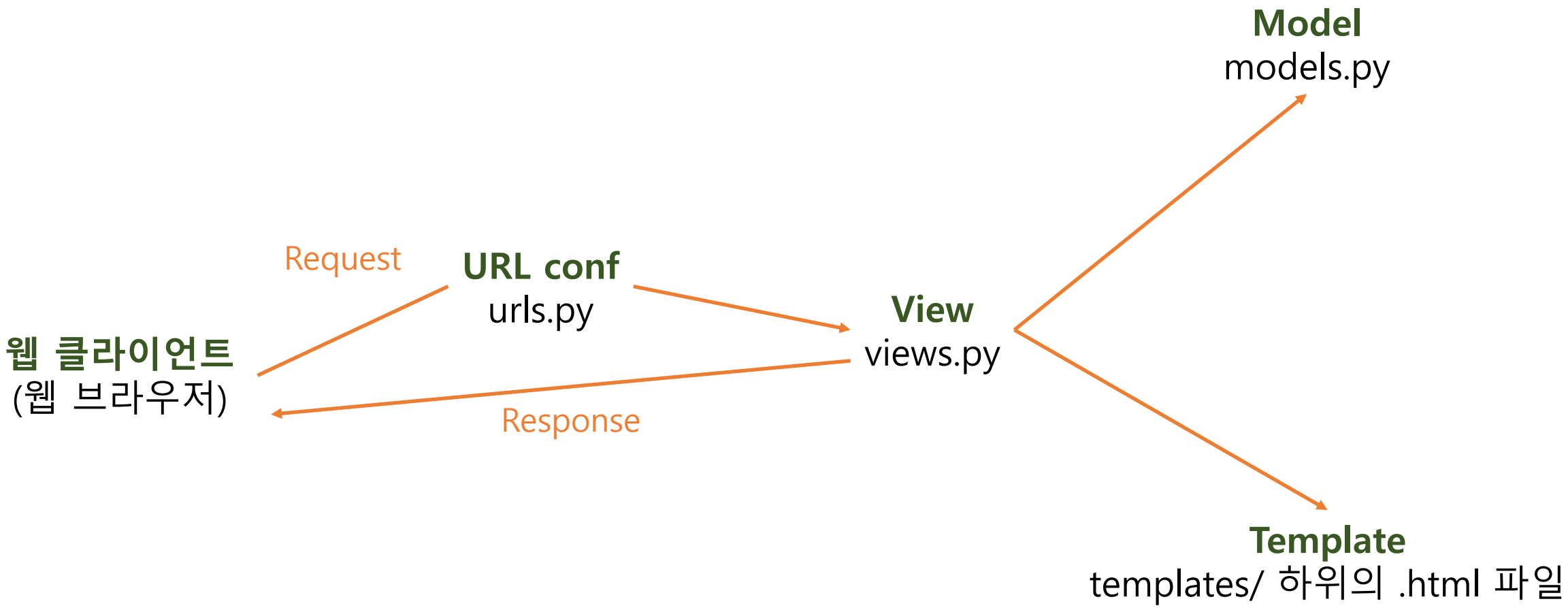
Template – 화면 UI 설계

- 사용자에게 보여지는 영역, 화면
- HTML

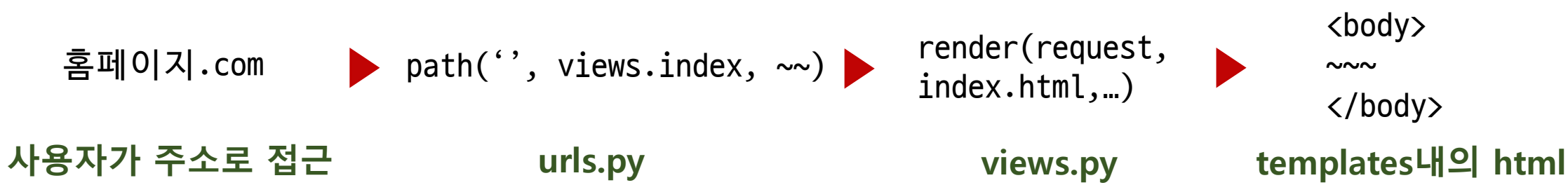
View – 프로그램 로직 설계

- 요청에 따라 Model에서 필요한 데이터를 가져와 처리
- 처리 결과를 Template에 전달

MTV 패턴 복습


Model
models.py
Request
URL conf
urls.py
View
views.py
웹 클라이언트
(웹 브라우저)
Response
Template
templates/ 하위의 .html 파일

MTV 패턴 복습


홈페이지.com
path(‘’, views.index, ~~)
render(request, index.html,…)
<body>
~~~
</body>
사용자가 주소로 접근
urls.py
views.py
templates내의 html
~~~

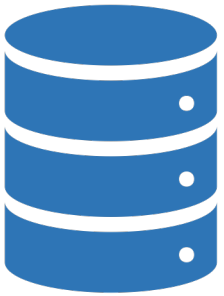
웹사이트에 글을 쓰려면 무엇이 필요할까?

☐ 총 10건의 게시물 제목 ▼ 검색

번호	제목	첨부파일	작성일	조회수
10	게시판 제목이 표출됩니다.		2111-09-15	11
09	게시판 제목이 표출됩니다.		2111-09-15	11
08	게시판 제목이 표출됩니다.		2111-09-15	11
07	게시판 제목이 표출됩니다.		2111-09-15	11
06	게시판 제목이 표출됩니다.		2111-09-15	11
05	게시판 제목이 표출됩니다.		2111-09-15	11
04	게시판 제목이 표출됩니다.		2111-09-15	11
03	게시판 제목이 표출됩니다.		2111-09-15	11
02	게시판 제목이 표출됩니다.		2111-09-15	11
01	게시판 제목이 표출됩니다.		2111-09-15	11

《 〈 1 · 2 · 3 · 4 · 5 〉 》 등록

글의 내용



글을 저장할 공간

데이터베이스

DB의 개념

글을 저장할 공간 => Database(DB)

🧐 데이터베이스란?

📌 여러 사람이 공유하여 사용할 목적으로 체계화해 통합,관리하는 데이터의 집합

📌 SQLite, MySQL, PostgreSQL 과 같은 DBMS(데이터베이스 관리 시스템)을 통해서 관리

📌 여러 테이블 들의 관계 집합 (Relational Database)

테이블 구성요소

데이터베이스

글을 저장할 공간 => Database(DB)

<학생 테이블(릴레이션) 의 예시>

어트리뷰트(Attribute)

스키마(모델)

튜플

학번	이름	학과이름	학점(예시)
2016130404	정성원	심리학부	3.5
2018250224	김범진	보건환경융합과학부	4.5
2014120299	정상윤	경영학과	4.3

Primary Key

데이터베이스

테이블에서 중복이 있을 수 없는 어트리뷰트는?

Primary Key 는 unique하다!!

<학생>

학번	이름	학과이름	학점(예시)
2016130404	정성원	심리학부	3.5
2018250224	김범진	보건환경융합과학부	4.5
2014120299	정상윤	경영학과	4.3

Foreign Key

데이터베이스

고려대학교 DB에서 아래 예시의 문제점은?

<학생>

학번	이름	학과이름	학점(예시)
2016130404	정성원	심리학부	3.5
2018250224	김범진	보건환경융합과학부	4.5
2014120299	정상윤	경영학과	4.3
2020135959	오징어	오징어학과	1.1

Foreign Key

데이터베이스

<학생>의 “학과이름”은 이미 존재하는 학과의 이름이어야 한다.

<학생>

학번	이름	학과이름	학점(예시)
2016130404	정성원	심리학부	3.5
2018250224	김범진	보건환경융합과학부	4.5
2014120299	정상윤	경영학과	4.3
2020135959	오징어	**오징어학과**	1.1

Foreign Key

데이터베이스

<학생>의 “학과이름”은 이미 존재하는 학과의 이름이어야 한다.

<학생>

학번	이름	학과이름	학점(예시)
2016130404	정성원	심리학부	3.5
2018250224	김범진	보건환경융합과학부	4.5
2014120299	정상윤	경영학과	4.3

<학과>

학과이름	빌딩	인원 수
심리학부	법학관구관	300
보건환경융합과학부	하나과학관	200
경영학과	Lg posco 경영관	400

Foreign Key

데이터베이스

<학생>의 “학과이름”는 <학과> 테이블을 참조하는 foreign key이다.

<학생>

학번	이름	학과이름	학점(예시)
2016130404	정성원	심리학부	3.5
2018250224	김범진	보건환경융합과학부	4.5
2014120299	정상윤	경영학과	4.3

<학과>

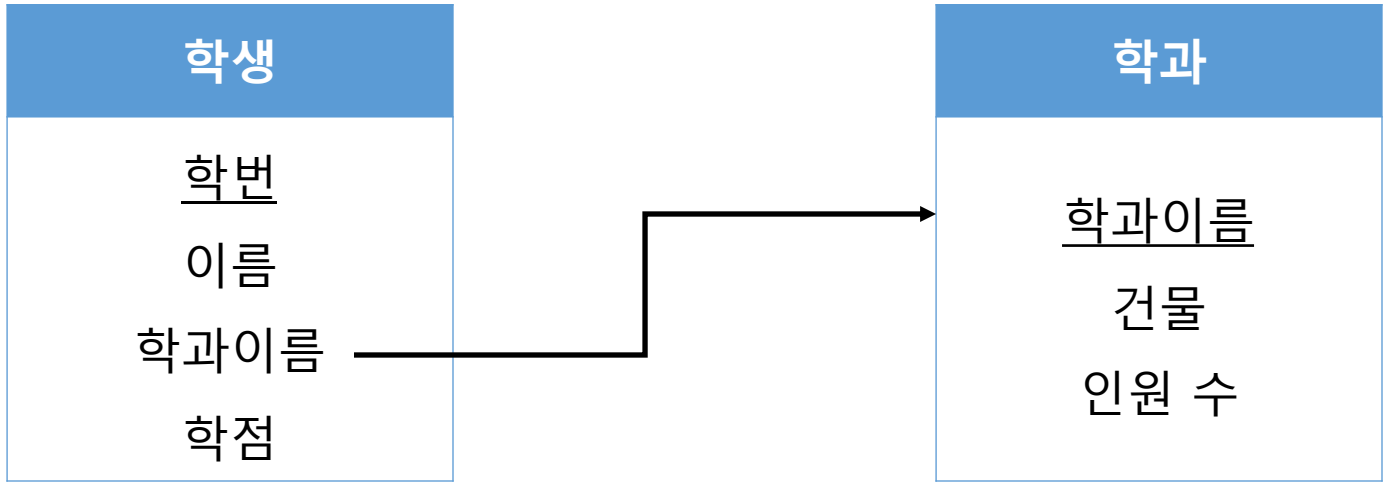
학과이름	빌딩	인원 수
심리학부	법학관구관	300
보건환경융합과학부	하나과학관	200
경영학과	Lg posco 경영관	400

💡 <학생> 과 <학과> 테이블이 각각 존재한다!

Foreign Key

데이터베이스

밑줄: Primary key
화살표: Foreign key 참조



💡 “학과이름”은 <학과>의 Primary key, <학생> 테이블에서는 <학과>를 참조하는 Foreign key.

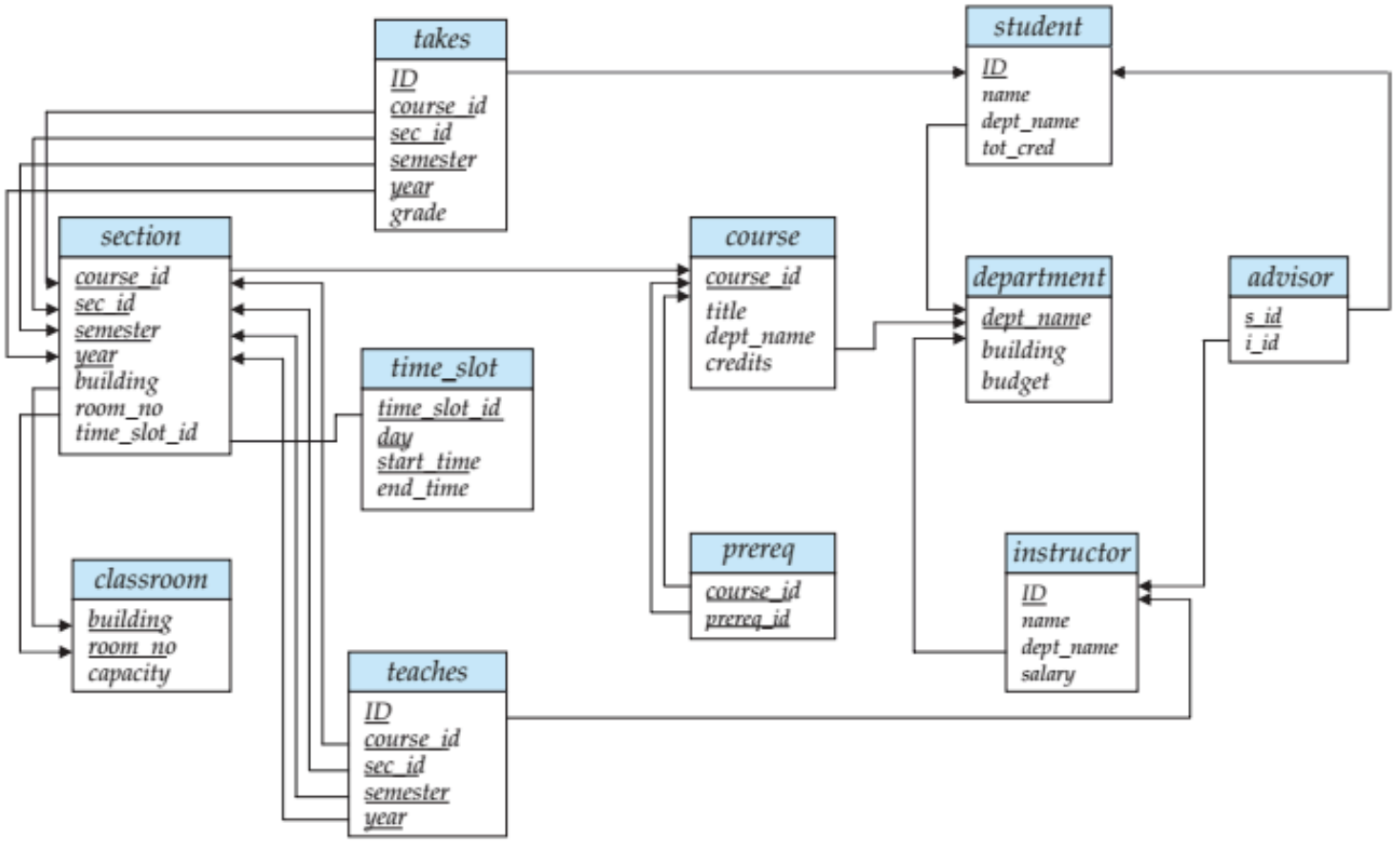
데이터베이스 예시

데이터베이스

밑줄: Primary key
화살표: Foreign key 참조

실제 학교 데이터베이스
(학생, 교수, 학과, 강의, 선수과목, 지도교수, 강의실, 등등)

LAM.pdf



블로그

데이터베이스

블로그에 필요한 데이터베이스는 무엇일까요?

글 ID	제목	내용	작성자 ID
1	...	...	1
2	...	...	2
3	...	...	2

<글 테이블>

댓글 ID	내용	작성자 ID	글 ID
1	무	2	1
2	야	3	1
3	호	3	2

<댓글 테이블>

유저 ID	유저 이름
1	ㄱ
2	ㄴ
3	ㄷ

<유저 테이블>

블로그

데이터베이스

앞장을 참고해 빈 칸을 채워보세요

글 ID	작성자 이름	댓글 내용
1		
2		
3		

<글 테이블>

블로그

데이터베이스

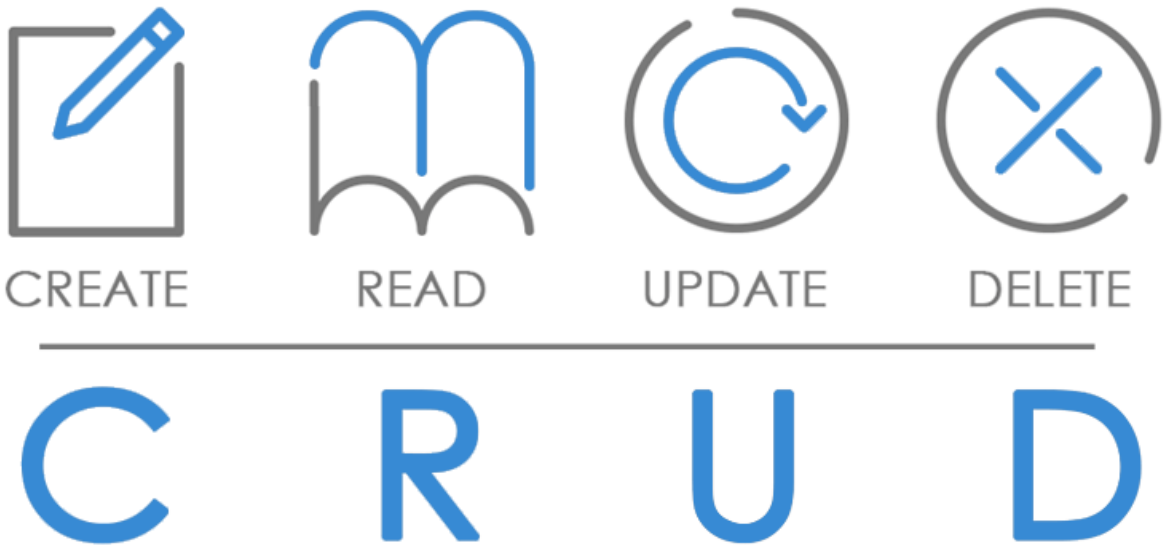
정답

글 ID	작성자 이름	댓글 내용
1	ㄱ	무, 야
2	ㄴ	호
3	ㄴ	

<글 테이블>

CRUD란?

CRUD소개



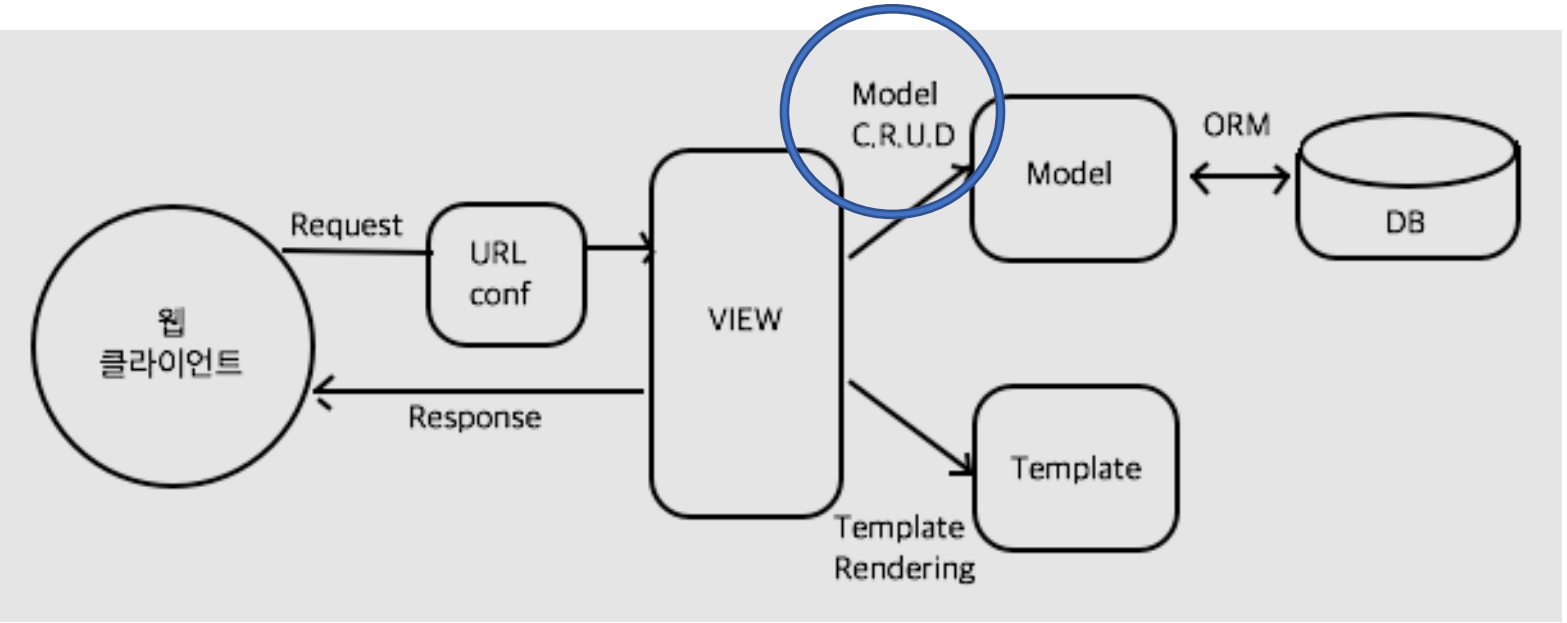
CRUD란?

CRUD소개

CRUD는 대부분의 컴퓨터 소프트웨어가 가지는 기본적인 데이터 처리 기능인 **Create(생성), Read(읽기), Update(갱신), Delete(삭제)**를 묶어서 일컫는 말

CRUD란?

CRUD소개



Django 의 Model 을 통해 DB의 데이터를
Create(생성), Read(조회), Update(수정), Delete(삭제)

오늘의 실습👨🏻‍💻

블로그 만들기

오늘의 목표는?

<블로그 만들기>

CRUD의 **Create, Read**를 사용해
새로운 글을 작성하고,
작성된 글들을 읽을 수 있는 블로그를 만들어보자!

프로젝트 세팅 To-do

1. 작업할 폴더 생성, 가상환경 세팅 및 장고 설치
2. 장고 프로젝트, 앱 생성
3. DB생성, 관리자 계정 생성

실습 프로젝트 세팅

작업할 폴더 생성, 가상환경 세팅 및 장고 설치

1. 터미널에서 작업할 폴더 생성 및 이동

```
$ mkdir session6, $ cd session6
```

2. Pipenv 로 가상환경 생성(Django 프로젝트 생성할 경로에서)

```
$ pipenv shell
```

3. 가상환경 내에 장고 설치

```
$ pipenv install django
```

실습 프로젝트 세팅

장고 프로젝트, 앱 생성

1. 장고 프로젝트 생성: django-admin startproject <프로젝트명>

    $ django-admin startproject blogProject

2. 프로젝트 폴더 내에 장고 앱 생성: python manage.py startapp <앱명>

    $ cd blogProject, $ python manage.py startapp blogApp

3. settings.py의 INSTALLED_APPS에 생성한 앱 추가

```
INSTALLED_APPS = [
…
'blogApp',
]
```

실습 프로젝트 세팅

DB생성, 관리자 계정 생성

1. DB 생성

```
$ python manage.py makemigrations
```

```
$ python manage.py migrate
```

2. 관리자 계정 생성

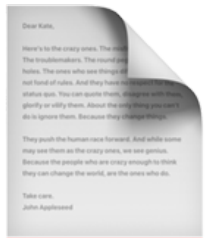
```
$ python manage.py createsuperuser
```

💡 앱 생성 후 이 과정은 이후에 할 작업 완료 후에 해줘도 괜찮다!

모델 어트리뷰트

장고의 데이터베이스

Django는 models.py에서 db 테이블을 생성하는데, **어트리뷰트**를 지정해줘야 한다.



TextField()

CharField(max_length=)

FileField()

DateTimeField()

💡 Primary Key를 따로 만들지 않아도 장고에서 알아서 1번부터 번호를 붙여준다.

🔍 https://docs.djangoproject.com/en/3.1/ref/models/fields/

모델 어트리뷰트

장고의 데이터베이스

<Name> 테이블 (예시)

ID	first_name	last_name
1	...	...
2	...	...
3	...	...

```python
from django.db import models

# Create your models here.
class Name(models.Model):
    first_name = models.CharField(max_length=10)
    last_name = models.CharField(max_length=10)
```

👉 models.py 에서 models 라는 이름의 장고 모듈을 import해서 **클래스 형태**로 작성한다.

💡 models.Model 이라는 클래스를 상속하여 테이블을 구성한다! (클래스 상속)

모델 생성

장고의 데이터베이스

실습: 조건에 맞게 Article 모델을 생성해보자!

<Article> 테이블

ID	title	content
1	…	…
2	…	…
3	…	…

조건:

"title" 은 200자로 제한을 두고 싶고,
"content"에는 긴 글을 저장하고 싶어…! 🤔

모델 생성

장고의 데이터베이스

<Article> 테이블

ID	title	content
1	...	...
2	...	...
3	...	...

정답

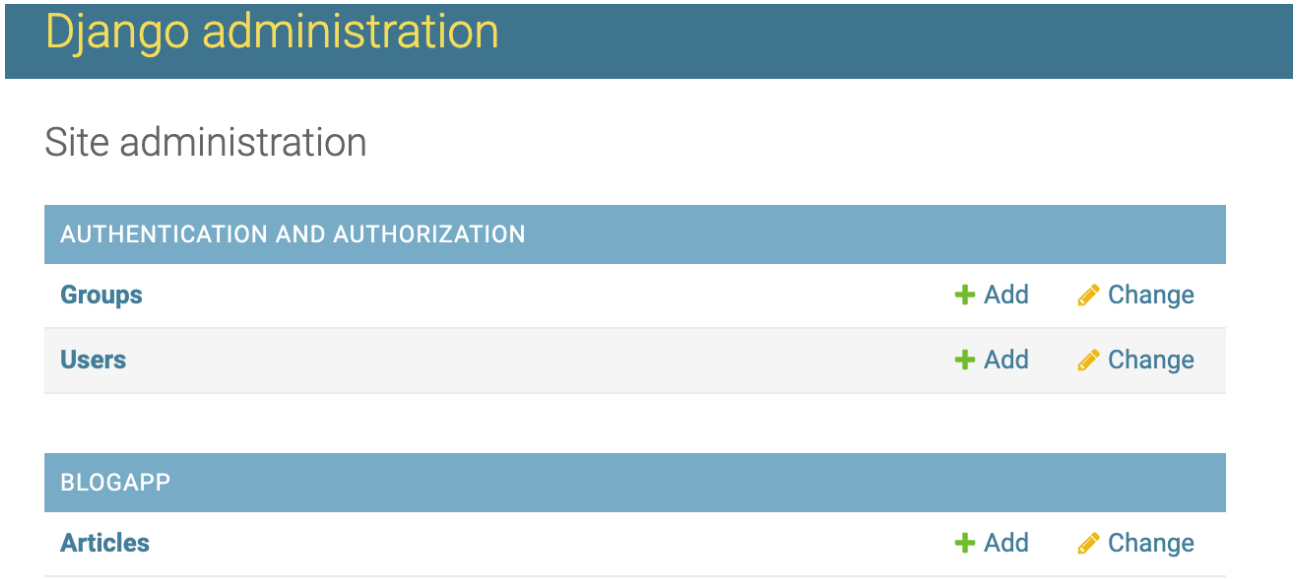
```python
# Create your models here.
class Article(models.Model):
    title = models.CharField(max_length=200)
    content = models.TextField()
```

💡 Primary Key 에 해당하는 ID 어트리뷰트는 장고에서 자동으로 생성해준다!

모델 관리

장고의 어드민 페이지

Django Admin



글 테이블 안에 생성된 튜플들(Article 클래스로 생성된 인스턴스들) 확인 가능!

http://127.0.0.1:8000/admin

모델 관리

장고의 어드민 페이지

생성한 모델을 어드민에서 확인하려면...

앱 폴더 내 admin.py 에 다음과 같이 작성

```
from django.contrib import admin
from .models import Article
# Register your models here.
admin.site.register(Article)
```

모델 관리

장고의 어드민 페이지

어느 쪽이 더 편할까?

Select article to change

Action: --------- Go 0 of 1 selected

ARTICLE
Article object (1)

1 article

Select article to change

Action: --------- Go 0 of 1 selected

ARTICLE
첫글

1 article

모델 관리

클래스 메소드

test.py 에 다음과 같이 작성 후 실행해보자

```
class Test():
    def __str__(self):
        return "hello"

test_instance = Test()
print(test_instance)
```

👉 __str__ 메소드에서 반환해준 값이 출력된다!

모델 관리

클래스 메소드

__str__ 메소드를 이용하여 title 이 보이도록 수정해보자!

```
# Create your models here.
class Article(models.Model):
    title = models.CharField(max_length=200)
    content = models.TextField()
```

Hint: Article 클래스로 생성한 인스턴스의 title을 가져오려면?

모델 관리

클래스 메소드

정답

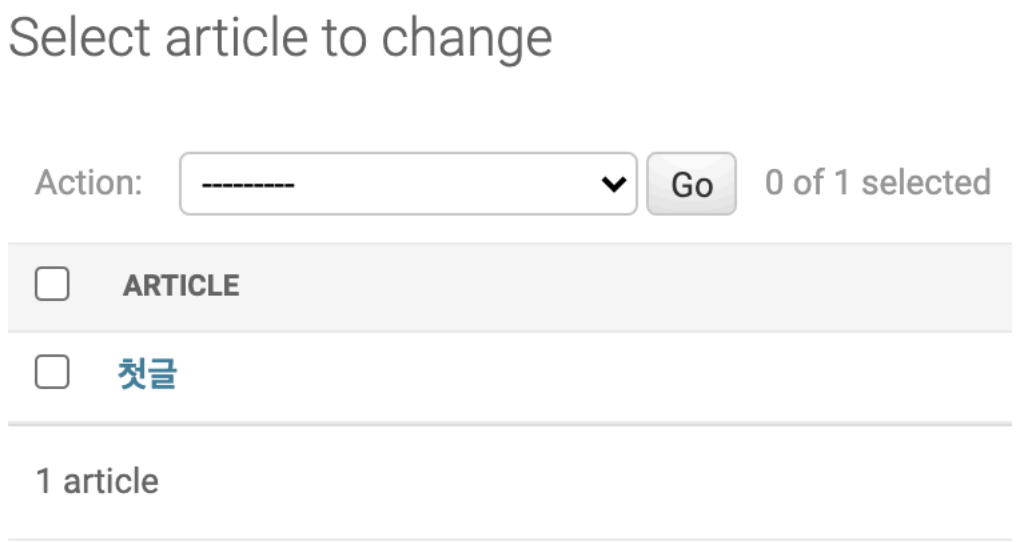
```
# Create your models here.
class Article(models.Model):
    title = models.CharField(max_length=200)
    content = models.TextField()


    def __str__(self):
        return self.title
```

self.title 로 글 인스턴스의 제목 값에 접근한다!

모델 관리

클래스 메소드

Admin에서 확인해 보면 다음과 같이 보일 것!



모델 사용

블로그 페이지 만들기

글의 생성, 조회를 위한 블로그에는 어떤 **페이지**가 필요할까?

Index

New

Detail

모델 사용

블로그 페이지 만들기

글의 생성, 조회를 위한 블로그에는 어떤 **페이지**가 필요할까?

Index

- 전체 글 리스팅
- 새 글 작성하러 가기

New

- 글 작성 양식
- 작성 버튼(저장)

Detail

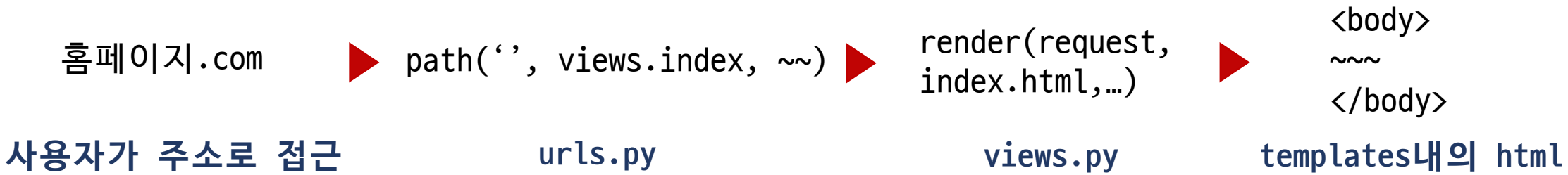
- 글 하나의 세부내용 보여주기

👉 메인페이지, 글을 쓰는 페이지, 글을 보는 페이지 가 필요!

장고 작동 순서

다시 한 번 remind~

Index, New, Detail 페이지에 대해 다음과 같은 과정을 거친다.



URL 설정

urls.py

각 페이지에 접근 위해 url 설정, views에 작성해줄 함수와 연결한다.

```
from blogApp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index, name='index'),
    path('new', views.new, name='new'),
    path('detail', views.detail, name='detail'),
]
```

사용자가 주소로 접근(request)하면, views.py 에서 해당 함수가 실행될 것!

DB 내용 가져오기

Django queryset methods

장고에서 DB 테이블의 내용을 가져오는 방법은 다음과 같다.

전부

```
<모델명>.objects.all()
```

하나

```
<모델명>.objects.get(pk=1)
```

* ( )안의 조건에 맞는 인스턴스가 여러 개이거나 없으면 에러 발생

몇 개

```
<모델명>.objects.filter(title="어떤 제목")
```

갯수

```
<모델명>.objects.count()
filter 된 변수.count()
```

🔍 https://docs.djangoproject.com/en/3.1/topics/db/queries/

HTML 연결

views.py

views의 함수에서 두가지 방식의 return!
보여주고자 하는 것을 마지막에 return 한다.

모델 오브젝트 + HTML 파일

return render(request, html파일,
모델에서 가져온 오브젝트/변수)

모델 오브젝트는
{'1':1, '2':2, } 와 같이 dict형태로
여러 개를 넘길 수 있음.

해당 주소로 리다이렉트

return redirect('urls.py 에서 정의한 name')

💡 views.py에서 import해서 사용

```
from django.shortcuts import render,redirect
```

HTML 연결

urls.py

여기서 정의한 name값이 Html에서 뿐 아니라, views에서의 redirect에도 사용된다!

```
from blogApp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index, name='index’),
    path('new', views.new, name='new’),
    path('detail', views.detail, name='detail'),
]
```

Index 함수 예시

views.py

이름 테이블에 저장된 모든 이름들을 보여주고 싶다면?

```
from .models import Name
# Create your views here.
def index(request):
    names = Name.objects.all()
    return render(request, 'index.html', {'names': names})
```

Index 함수 작성

views.py

글 테이블에 저장된 모든 이름들을 보여주고 싶다면?

Index 함수 작성

views.py

글 테이블에 저장된 모든 이름들을 보여주고 싶다면?(정답)

```python
from .models import Article
# Create your views here.
def index(request):
    articles = Article.objects.all()
    return render(request, 'index.html', {'articles': articles})
```

Detail 함수 예시

views.py

이름 테이블에서 하나의 데이터만 보여주고 싶다면?

```python
from .models import Name
# Create your views here.

def detail(request, name_pk):
    name = Name.objects.get(pk=name_pk)
    return render(request, 'detail.html', {'name': name})
```

💡 *name_pk* 는 Name 테이블의 Primary Key를 뜻한다!

Detail 함수 작성

views.py

글 테이블에서 특정 글 하나만 보여주고 싶다면?

Detail 함수 작성

views.py

글 테이블에서 특정 글 하나만 보여주고 싶다면?(정답)

```python
from .models import Article
# Create your views here.

def detail(request, article_pk):
    article = Article.objects.get(pk=article_pk)
    return render(request, 'detail.html', {'article': article})
```

각 글의 주소

urls.py

detail/1 이라는 주소에 들어가면 pk=1인 글을 보여준다! (urls.py)

```python
from blogApp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index, name='index'),
    path('detail/<int:article_pk>', views.detail, name='detail'),
]
```

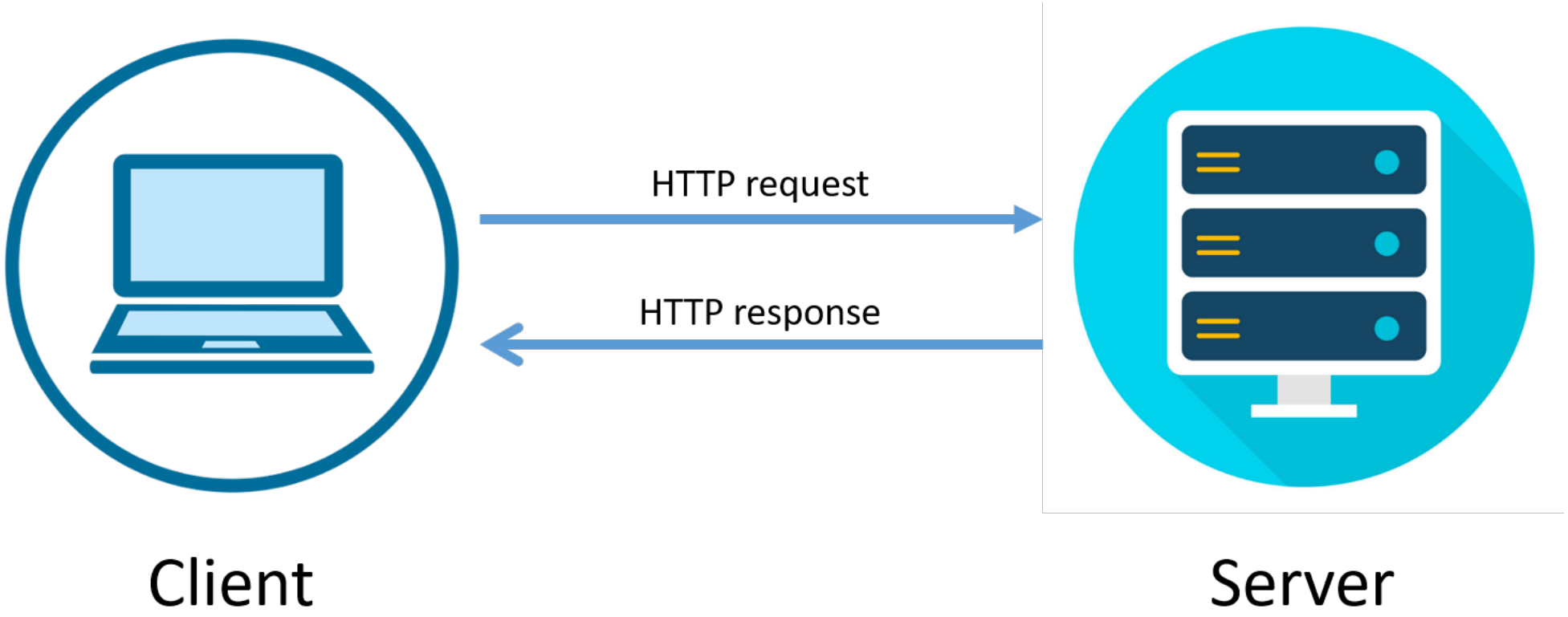
int: 다음에 들어가는 것은 views.py 에서 인자로 넣은 변수명과 동일해야함!

```python
from .models import Article
# Create your views here.

def detail(request, article_pk):
    article = Article.objects.get(pk=article_pk)
    return render(request, 'detail.html', {'article': article})
```

<= views.py

요청과 응답

서버-클라이언트



Http method – GET/POST

GET

존재하는 자원을 조회할 때

읽기

어떤 내용을 가지고 오고 싶을 때!

URL에 변수(데이터)를 포함시켜 요청

글 내용을 가지고 올 때

POST

새로운 자원을 생성

저장

새로운 내용을 쓸 때

URL에 변수(데이터)를 노출하지 않고 요청

글을 쓸 때(새로운 글 내용을 포함시켜서 보낼 때)

New 함수

기능별로 method 나누기

New – 두가지 기능!

새 글 작성할 수 있는 양식 보여주기

글을 작성할 Form 태그가 있는
new.html을 보여준다.

GET 방식 사용!

작성한 글 내용을 DB에 저장

FormData를 가져와서 DB에 저장한다.

POST 방식 사용!

New 함수 예시

views.py

```
from django.shortcuts import render,redirect
```

```
def new(request):
    if request.method == 'POST' :
    #POST 요청일 경우
        print(request.POST) # data확인
        new_article = Article.objects.create(
            title = request.POST['title' ],
            content = request.POST['content' ]
        )
        return redirect('detail', article_pk=new_article.pk)

    #POST 요청이 아닐 경우
    return render(request, 'new.html')
```

New 함수 예시

views.py

```python
from django.shortcuts import render,redirect
```

redirect() 사용 위해 import

```python
def new(request):
    if request.method == 'POST':
    #POST 요청일 경우
        print(request.POST) # data확인
        new_article = Article.objects.create(
            title = request.POST['title'],
            content = request.POST['content']
        )
        return redirect('detail', article_pk=new_article.pk)

    #POST 요청이 아닐 경우
    return render(request, 'new.html')
```

request.method 로 method 구분.

Model에 새로운 인스턴스 생성시
<모델이름>.objects.create()

New 함수 예시

views.py

```python
def new(request):
    if request.method == 'POST':
    #POST 요청일 경우
        print(request.POST) # data확인
        new_article = Article.objects.create(
            title = request.POST['title'],
            content = request.POST['content']
        )
        return redirect('detail', article_pk=new_article.pk)

    #POST 요청이 아닐 경우
    return render(request, 'new.html')
```

urls.py에서 선언한 name인 'detail'
사용하여 detail페이지로 연결

```python
from .models import Article
# Create your views here.

def detail(request, article_pk):
    article = Article.objects.get(pk=article_pk)
    return render(request, 'detail.html', {'article': article})
```

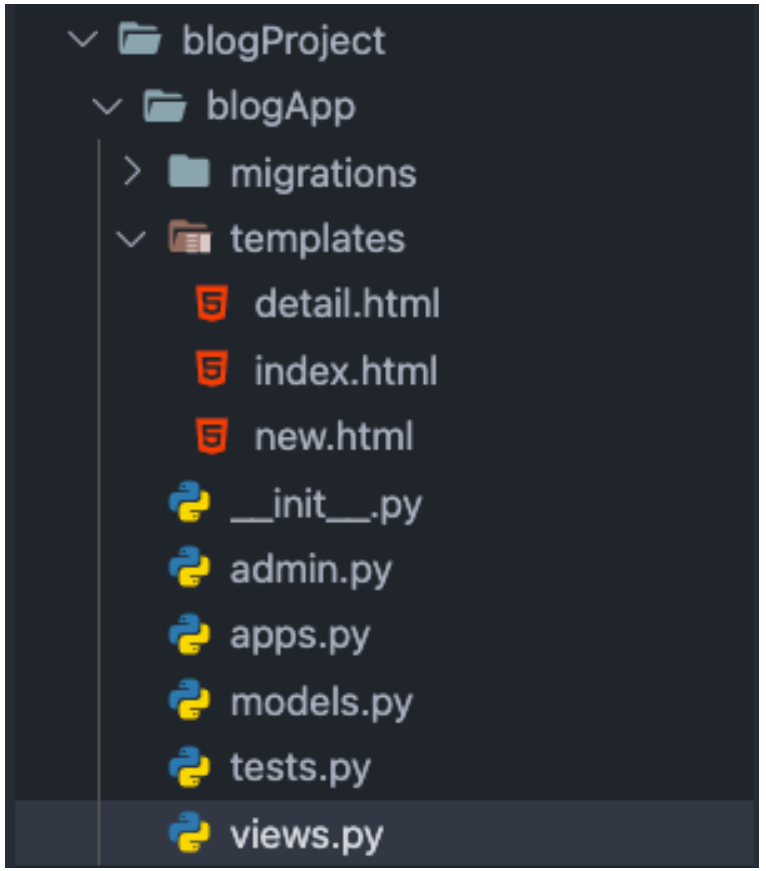
detail 로 redirect시에
새로 생성한 글의 pk전달.

Templates 작성

Html파일 만들기

1. 앱 폴더 안에 templates 폴더 생성

2. templates 폴더 안에 index.html, new.html, detail.html 생성



Templates 작성

Django Template Language

HTML 내에서 사용 가능한 장고만의 문법

```
{% if (조건문) %}
    ~~~~
{% endif %}
```

```
{% for~ in ~ %}
    ~~~~
{% endfor %}
```

```
{% url 'index' %}
```

```
{{객체.어트리뷰트}}
```

기본 문법은 {% %}로 감싸주고, 모델 객체(글)/변수와 관련된 것은 {{}}로 감싸준다!

🔍 https://docs.djangoproject.com/en/3.1/ref/templates/language/

index.html 예시

templates

앞서 들었던 이름 예시의 경우

```python
from .models import Name
# Create your views here.
def index(request):
    names = Name.objects.all()
    return render(request, 'index.html', {'names': names})
```

<= views.py

index.html=>

```
<body>
    {% for name in names %}
    <ul>
        <li>
            <a href="{% url 'detail' name.pk %}">{{ name.first_name }}</a>
        </li>
    </ul>
    {% endfor %}
    <a href="{% url 'new' %}">글 쓰러 가기</a>
</body>
```

/detail/name.pk
라는 url로 연결

detail.html 예시

templates

앞서 들었던 이름 예시의 경우

```python
from .models import Name
# Create your views here.

def detail(request, name_pk):
    name = Name.objects.get(pk=name_pk)
    return render(request, 'detail.html', {'name': name})
```

<= views.py

detail.html=>

```html
<body>
    <h2>{{name.first_name}}</h2>
    <p>{{name.last_name}}</p>
</body>
```

하나만 가져와서 보여준다!

new.html 예시

templates

앞서 들었던 이름 예시의 경우

```python
# Create your models here.
class Name(models.Model):
    first_name = models.CharField(max_length=10)
    last_name = models.CharField(max_length=10)
```

<= models.py

new.html=>

```html
<body>
    <form action="" method="post">
        {% csrf_token %}
        <input name="first_name" placeholder="이름을 입력하세요">
        <input name="last_name" placeholder="성을 입력하세요">
        <button type="submit">저장하기</button>
    </form>
</body>
```

태그의 name에 어트리뷰트 명 지정해서 form 생성

Templates 작성

실습

글 모델을 이용해 index.html, detail.html, new.html 을 작성해봅시다!

앞의 예시들을 참고하여 templates를 직접 작성해보세요!

Templates 작성

실습정답

index.html (정답)

```
<body>
    {% for article in articles %}
        <ul>
            <li>
                <a href="{% url 'detail' article.pk %}">{{ article.title }}</a>
            </li>
        </ul>
    {% endfor %}
    <a href="{% url 'new' %}">글 쓰러 가기</a>
</body>
```

Templates 작성

실습정답

detail.html (정답)

```
<body>
    <h2>{{article.title}}</h2>
    <p>{{article.content}}</p>
</body>
```

Templates 작성

실습정답

new.html (정답)

```
<body>
    <form action="" method="post">
        {% csrf_token %}
        <input name="title" placeholder="제목">
        <br>
        <textarea name="content" cols="30" rows="10" placeholder="내용"></textarea>
        <button type="submit">저장하기</button>
    </form>
</body>
```

마이그레이션

DB변경사항 적용

서버 실행 전, DB 변경 사항을 적용한다.

1.(프로젝트 경로에서) $ python manage.py makemigrations

- models.py 에서 만든 사용자의 글을 저장할 공간을 실체화 하는 준비작업

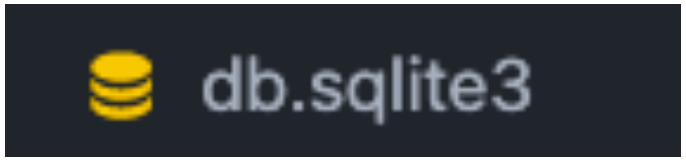
2. (프로젝트 경로에서) $ python manage.py migrate

- makemigrations가 했던 준비작업 완료시켜 models.py의 공간을 실체화

DB 파일

DB변경사항 적용

프로젝트 폴더 안에 DB 파일 생성됨



db.sqlite3라는 파일에 db가 저장된다.

=> DBMS 에서 확인도 가능!

과제🦁

과제타임

블로그 프로젝트 수정하기

- 게시글에 "카테고리" 추가- 영화(movie), 드라마(drama), 프로그래밍(programming)
- Index 페이지에서는 각 카테고리로 이동하는 링크와 각 카테고리 게시글 갯수 보여주기
- Detail 페이지에서는 글 제목, 내용, 글 작성시각 보여주기
- 각 카테고리별 페이지에서 해당 카테고리의 게시글만 리스팅

~04/15 (목) 까지

과제🦁

과제타임

🍯 과제 TIP

- 카테고리는 글 작성시 select box로 선택하게 하기
- 필요한 페이지는 총 6개 – index.html, new.html, detail.html, movie.html, drama.html, programming.html
- 각 페이지 마다 urls.py, views.py 작성 필요
- 글 작성시각은 datetimefield 사용하지 않고도 views.py에서 처리 가능(hint: python의 time module)

~04/15 (목) 까지

끝~ 👏👏👏