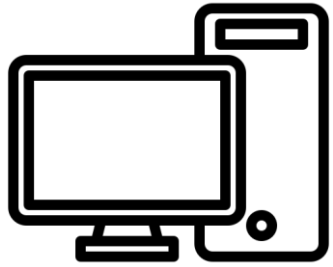


HICC 개발 입문 세미나

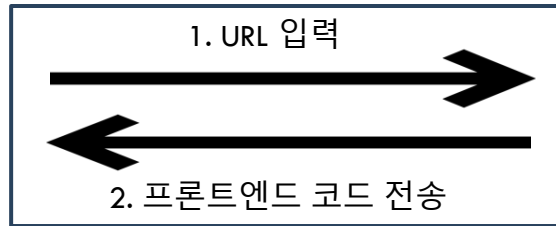
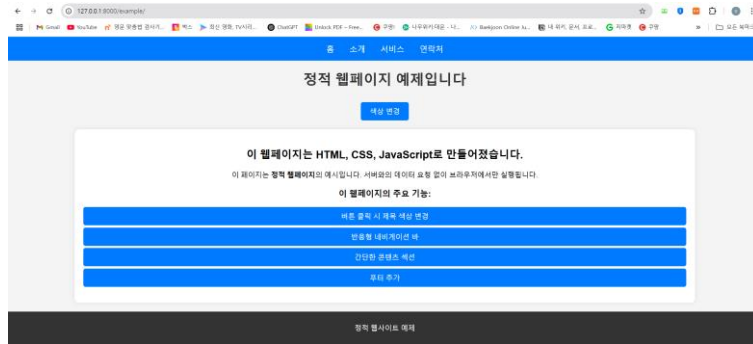
5주차. MVC 패턴 + 데이터베이스 기초

웹 세계에서 일어나는 일 (정적)

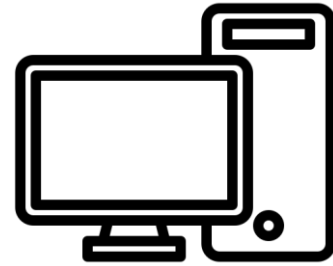


내 컴퓨터

3. 브라우저에서 코드 실행

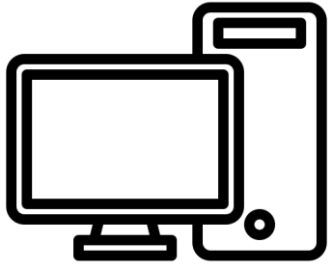


내 컴퓨터가 서버에 요청(request)을 보내고,
서버는 내 컴퓨터로 응답(response)을 보냅니다.



서버

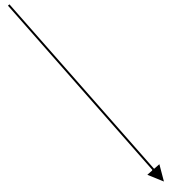
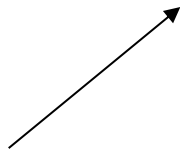
웹 세계에서 일어나는 일 (동적)



내 컴퓨터

3. 브라우저에서 코드 실행

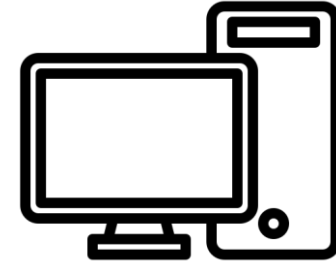
7. 실행된 결과 띄우기



4. 서버에 api 요청

5. 서버에서 코드 실행

6. api 응답 데이터 전송



서버

질문 게시판		
번호	제목	작성일시
질문 등록하기		

질문 게시판		
번호	제목	작성일시
3	HICC는 무엇의 약자인가요?	2025. 3. 11. 오전 3:53:20
4	HICC 동방은 몇호인가요?	2025. 3. 11. 오전 4:25:32
질문 등록하기		

백엔드와 DB



이와 같이 애플리케이션 서버가 실행하는 코드를 **백엔드 코드**라고 부릅니다.



이들의 역할은 보이지 않는 곳에서 클라이언트(프론트엔드)의 요청을 받아, 로직을 실행하여 **데이터베이스**와 상호작용한 후 응답을 반환하는 것입니다.



데이터베이스(DB)란 백엔드 코드가 데이터를 저장, 관리, 조회, 수정, 삭제(CRUD)할 수 있도록 구조화된 방식으로 제공하는 시스템을 의미합니다.

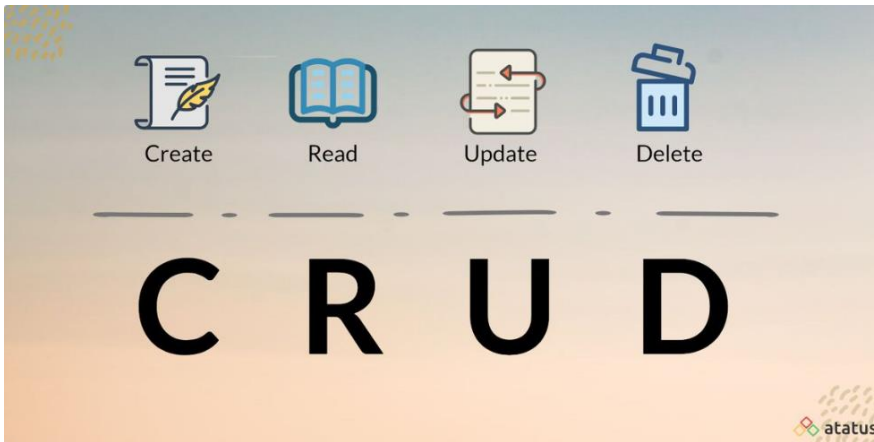
쉽게 말해서 **디지털 데이터들의 창고** 정도로 이해하면 된다.

백엔드와 DB

	id	category	closed_days	lat	lng	name	opening_hours	phone_num	place_img_url
▶	1	문예회관	전시공연에 따라 변동	35.22581853	128.577106	315아트센터	전시공연에 따라 변동	055-719-7800	https://lh3.googleusercontent.com/...
	2	문예회관	법정공휴일, 매주 월요일	35.10968396	129.028028	부산민주공원	화~일 09:00~18:00	051-790-7400	https://lh3.googleusercontent.com/...
	3	미술관	1월1일, 설, 추석 당일, 매주 월요일	37.32507855	126.80999	경기도미술관	화~일 10:00~18:00	031-481-7000	https://lh3.googleusercontent.com/...
	4	미술관	연중무휴	35.8322527	129.288867	경주솔거미술관	매일 10:00~18:00	054-740-3990	https://lh3.googleusercontent.com/...
	5	미술관	매주 월요일	35.86235646	129.205996	경주예술의전당 알천미술관	화~일 10:00~18:00	1588-4925	https://lh3.googleusercontent.com/...
	6	미술관	매주 토, 일	35.10590365	126.89617	광주대학교 호심미술관	월~금 10:00~17:00	062-670-2173	https://lh3.googleusercontent.com/...
	7	미술관	매주 월, 화	37.60305157	127.351144	구하우스	수~일 10:30~18:00	031-774-7460	https://lh3.googleusercontent.com/...
	8	미술관	매주 월요일	37.9045119	127.472275	남송미술관	화~일 10:00~18:00	031-581-0772	https://lh3.googleusercontent.com/...
	9	미술관	추석연휴, 매주 월요일	37.57750388	126.973364	대림미술관	화~일 10:00~18:00	02-720-0667	https://lh3.googleusercontent.com/...
	10	미술관	매주 월요일	36.82015665	127.186286	리각미술관	화~일 11:00~18:00	070-4111-3463	https://lh3.googleusercontent.com/...
	11	미술관	1월1일, 설, 추석 당일, 매주 월요일	35.13418204	126.953724	무등현대미술관	화~일 10:00~18:00	062-223-6677	https://lh3.googleusercontent.com/...
	12	미술관	매주 월, 화	37.71664923	126.689097	미메시스아트뮤지엄	수~일 10:00~18:00	031-955-4100	https://lh3.googleusercontent.com/...
	13	미술관	1월1일, 매주 월요일	35.10933193	128.942694	부산현대미술관	화~일 10:00~18:00	051-220-7400	https://lh3.googleusercontent.com/...
	14	미술관	매주 월요일	37.63012857	126.929276	사비나미술관	화~일 09:30~18:30	02-736-4371	https://lh3.googleusercontent.com/...
	15	박물관	1월1일, 설, 추석 연휴, 매주 월요일	37.41847997	126.657603	가천박물관	화~일 09:00~17:00	032-833-4747	https://lh3.googleusercontent.com/...

이런식으로 데이터베이스에는 백엔드에서 프론트엔드로 렌더링할 때 보내줘야할 정보들이 저장되어있음

기본적인 데이터 처리 기능



DB의 데이터를 만들고(CREATE), 읽고(READ), 변경하고(UPDATE), 삭제하고(DELETE)

127.0.0.1:8000/questions

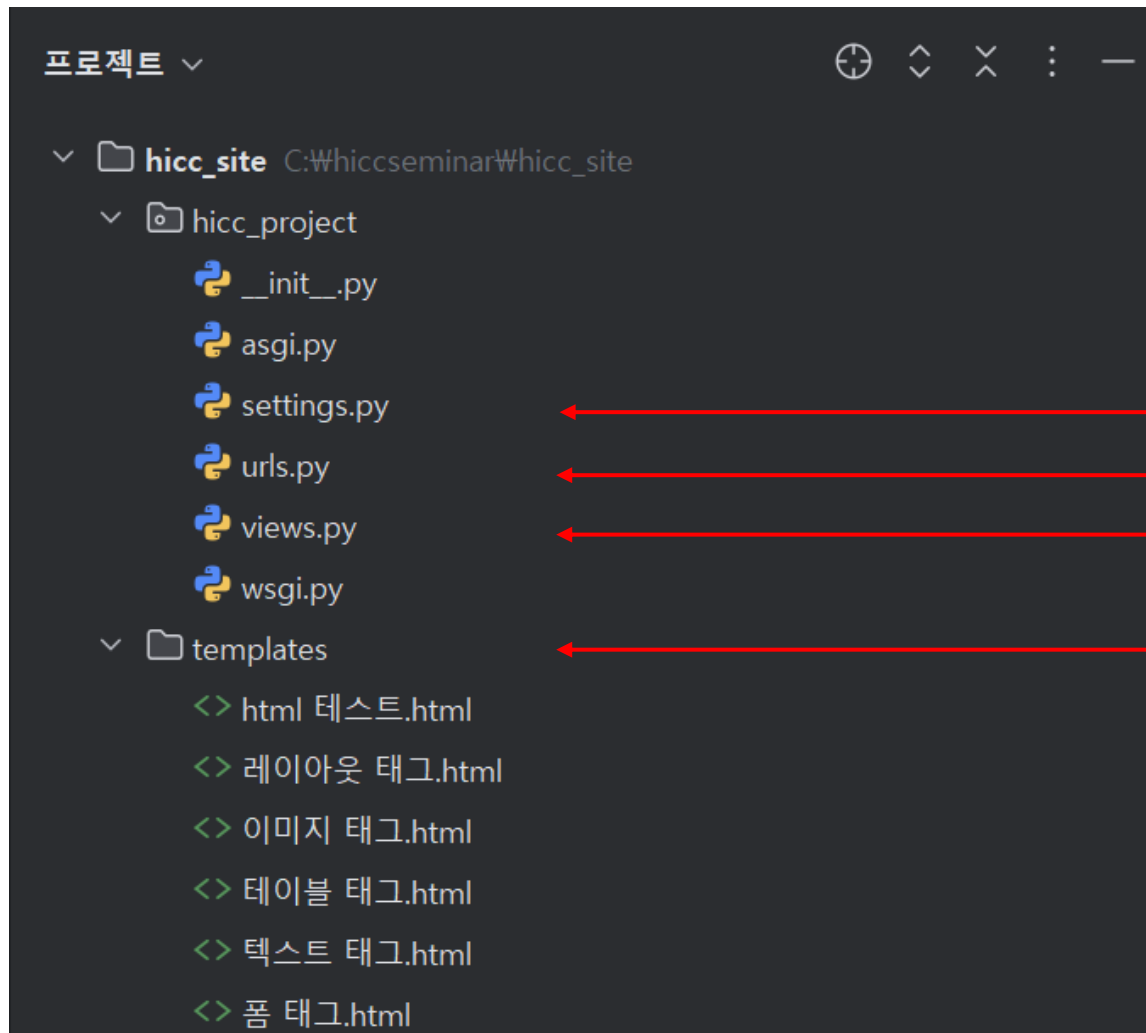
질문 게시판

번호	제목	작성일시
3	HICC는 무엇의 약자인가요?	2025. 3. 11. 오전 3:53:20
4	HICC 동방은 몇호인가요?	2025. 3. 11. 오전 4:25:32

질문 등록하기

위 페이지는 DB의 CRUD 중 READ 기능이 구현된 페이지라고 볼 수 있음

Django 프로젝트 구조



← 프로그램 설정을 담아둔 파일

← url 경로를 설정하는 파일

← 특정 url을 입력했을 때 브라우저로 코드를 전송하는 파일

← 브라우저 쪽으로 보내줄 html 코드들을 모아둔 폴더

웹 프레임워크



- Django는 파이썬으로 작성되는 웹 프레임워크입니다.
- 다른 프레임워크에 비해 난이도가 매우 낮은 것이 특징이며, 프로젝트 입문자들이 사용하기에 적합합니다.
- Instagram, Pinterest 같은 유명한 프로그램도 Django를 통해 개발되었습니다.
- Model – Template – View 의 구조를 따르고 있습니다.

Python의 클래스

클래스를 프로그래밍적 언어로 표현하자면 어떠한 물체의 변수(variable) , 함수(method)의 집합입니다.

```
class Dog:
    def __init__(dog, name, age):
        dog.name = name
        dog.age = age

    def bark(dog):
        print(dog.name + "이(가) 멍멍!")

    def show_info(dog):
        print(dog.name + "은(는)", dog.age, "살입니다.")

# 강아지 만들기
choco = Dog("초코", 3)

# 강아지 행동 시키기
choco.bark()      # 초코이(가) 멍멍!
choco.show_info() # 초코은(는) 3 살입니다.
```

해당 코드에서 `__init__` 생성자의 인자로 받는 `name`, `age`가 변수, `bark`, `show_info`가 함수입니다.

그리고 클래스를 통해 마치 생명을 불어넣듯이 특정 물체를 나타내는 "클래스 변수"(객체)를 만들 수 있고 해당 강아지 안에는 그 물체의 이름, 나이(변수)와 짖기, 정보보기(함수) 등이 들어 있습니다.

Django의 MTV 구조

Model(모델)

- 데이터베이스에 저장되는 데이터와 View 사이의 연결고리 역할을 합니다. 모델은 클래스로 정의되며 **하나의 클래스가 하나의 DB Table**입니다.
원래 DB를 조작하기 위해선 SQL이라는 DB 전용 언어를 다룰 줄 알아야 하지만 **Django는 ORM(Object Relational Mapping)기능을 지원하기 때문에 파이썬 코드로 DB를 조작할 수 있습니다.**

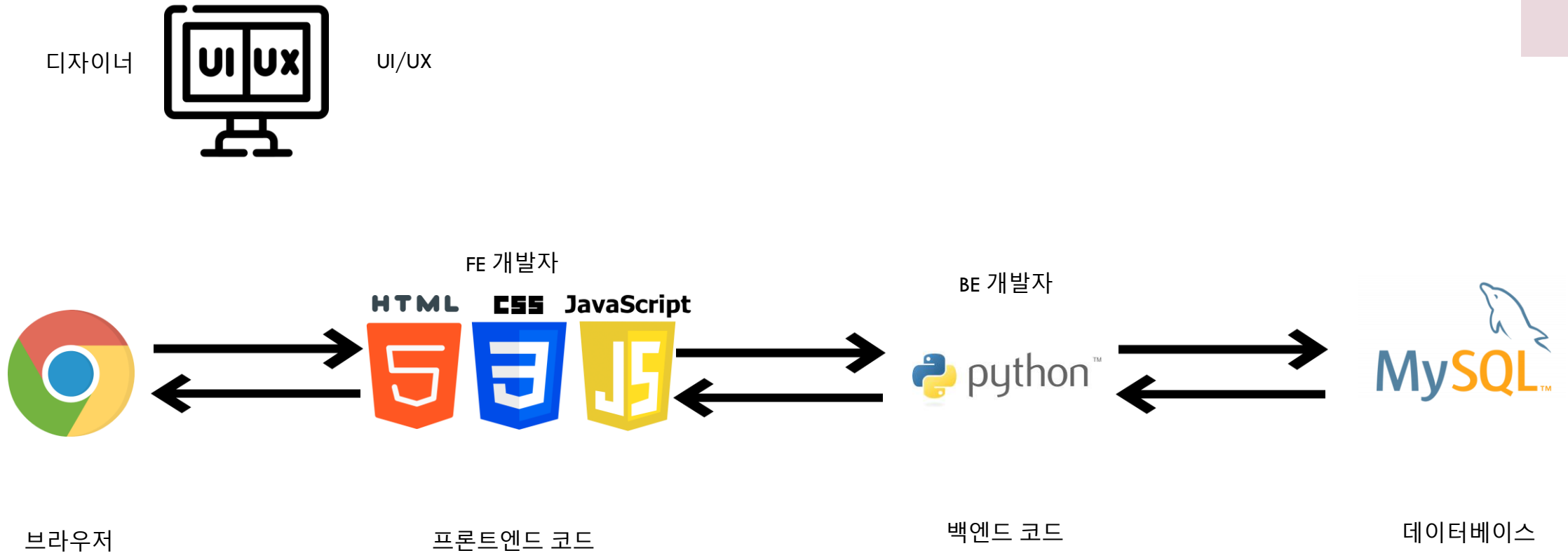
View(뷰)

- 요청에 따라 적절한 로직을 수행하여 결과를 템플릿으로 렌더링하며 응답합니다. (다만 항상 템플릿을 렌더링 하는 것은 아니고 서버에서 데이터만 보내주는 경우가 대부분입니다. 해당 내용은 6~7주차때 깊게 다룹니다.)

Template(템플릿)

- 유저에게 보여지는 화면을 의미합니다. 장고는 뷰에서 로직을 처리한 후 html 파일을 context와 함께 렌더링하는데 이 때의 html 파일을 템플릿이라 칭합니다.

일반적 프로그램 개발 프로세스

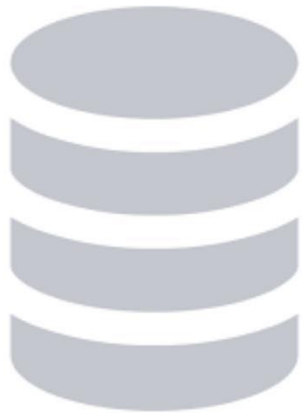


MTV 구조

• Model Template View 구조

데이터베이스와 연결

Model 영역



Models.py
DB 데이터

3. DB 정보 제공

2. DB에 요청

Controller 영역



Views.py
가공

4. HTML에
정보 삽입

5. 렌더링

1. Http 요청

6. Http 응답
Html 데이터



프론트엔드와 연결

View 영역

HTML



CSS

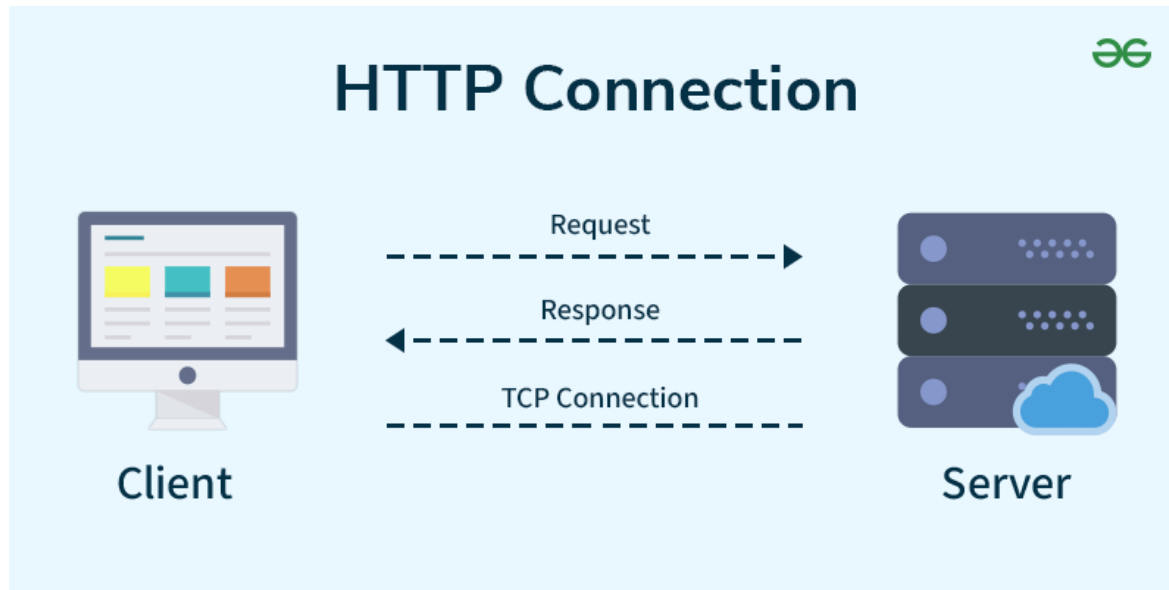


Template
보여지는 화면

HTTP 의미

HTTP란?

HTTP는 클라이언트(웹 브라우저, 애플리케이션 등)와 서버 간에 **요청(request)**과 **응답(response)**을 통해 HTML 문서, 이미지 등과 같은 데이터를 주고 받기 위한 프로토콜을 의미합니다



HTTP 요청, 응답 확인

urls.py에

```
from django.contrib import admin
from django.urls import path
from hicc_project.views import text_tag_render, html_test_render, image_tag_render, form_tag_render, table_tag_render, \
layout_tag_render, http_test

urlpatterns = [
    path('admin/', admin.site.urls),
    path('html_test/', html_test_render),
    path('text_tag/', text_tag_render),
    path('image_tag/', image_tag_render),
    path('form_tag/', form_tag_render),
    path('table_tag/', table_tag_render),
    path('layout_tag/', layout_tag_render),
    path('test/', http_test)
]
```

views.py에

```
from django.http import HttpResponse

def http_test(request):
    return HttpResponse("안녕하세요 hicc 세미나에 오신 것을 환영합니다.")
```

MVC 패턴

Django의 MTV 구조처럼, 주요 웹 프레임워크는 MVC(Model – View – Controller) 패턴을 지닙니다.

Model(Django의 Model) 영역에서 데이터베이스와 백엔드 코드를 연결하고,

View(Django의 Template) 영역에서 백엔드 코드와 프론트엔드 코드를 연결하고,

Controller(Django의 View) 영역에서 Model, View, 클라이언트(웹 브라우저) 사이를 연결합니다.

관계형 데이터베이스

관계형 데이터베이스의 기본 용어

- 테이블 – 하나의 개체에 관한 데이터를 2차원 구조로 정의한 것
- 튜플 – 테이블의 행, 쉽게 말해 실제 데이터
- 속성 – 테이블의 열, 쉽게 말해 실제 데이터들의 카테고리
- 도메인 – 실제 데이터 값의 종류, 쉽게 말해 **자료형**

열(속성, 애트리뷰트)

고객아이디	고객이름	나이	등급	직업	적립금
CHAR(20)	CHAR(20)	INT	CHAR(10)	CHAR(10)	INT
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

행(튜플)

도메인

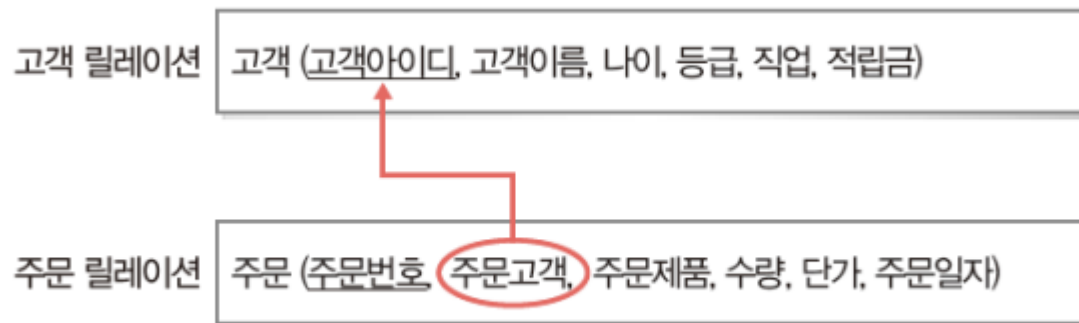
관계형 데이터베이스

기본키(primary key)

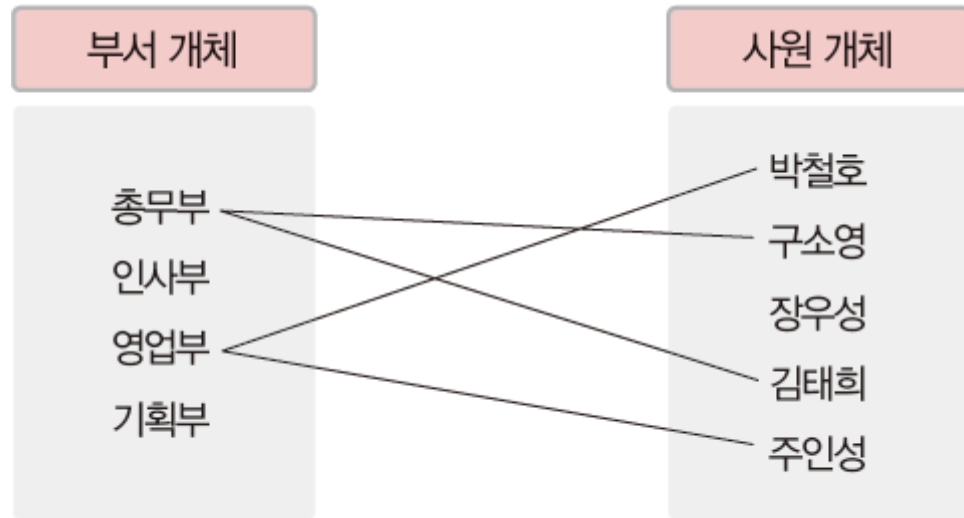
- 테이블에서 튜플 하나가 기본적으로 사용하는 고유의 아이디
- Ex) 고객 테이블의 기본키 : 고객아이디

외래키(foreign key)

- 다른 테이블의 기본키를 참조하는 속성
 - 테이블들 간의 관계를 표현
- ex) 주문 테이블의 외래키 : 주문고객 (고객아이디를 참조하기 때문에 고객 테이블을 참조함)



관계형 데이터베이스

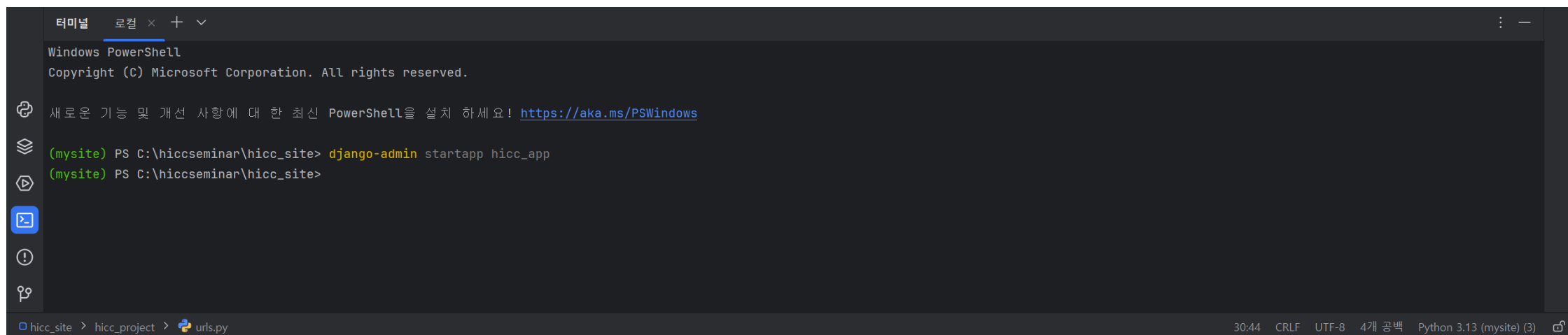


이와같은 1 : N 관계에서 N쪽이 1쪽을 참조함
(즉 N 테이블이 1 테이블의 기본키를 외래키로 지님)

Model로 DB 생성해보기

이제부터 Django의 Model을 활용해서 데이터베이스를 다뤄보도록 합시다.

그 전에 파이참에서 터미널을 켜서 아래의 명령어를 입력해줍니다.



```
터미널  로컬 x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 기능 및 개선 사항에 대한 최신 PowerShell을 설치하세요! https://aka.ms/PSWindows

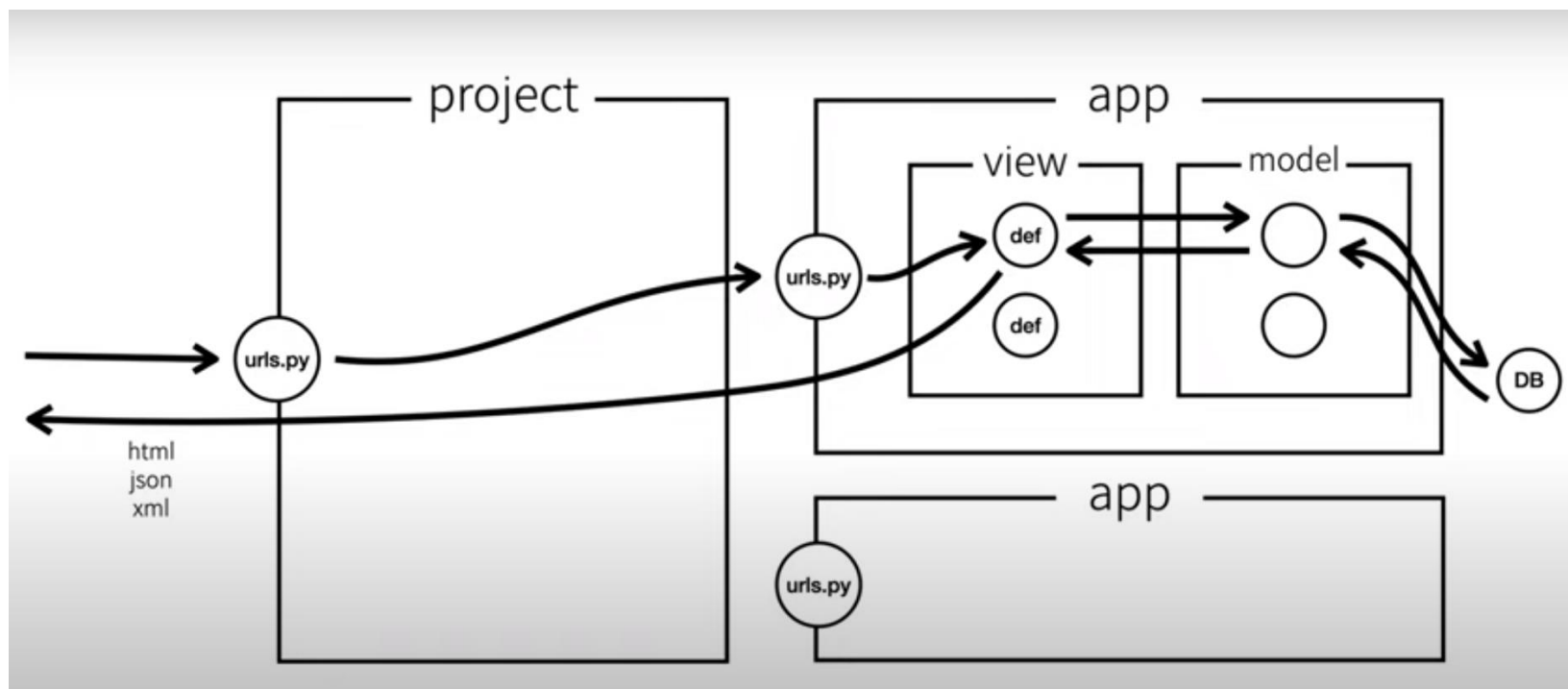
(mysite) PS C:\hiccseminar\hicc_site> django-admin startapp hicc_app
(mysite) PS C:\hiccseminar\hicc_site>
```

hicc_site > hicc_project > urls.py 30:44 CRLF UTF-8 4개 공백 Python 3.13 (mysite) (3)

Model로 DB 생성해보기

Django에서는 아래와 같이 하나의 프로젝트와 그 프로젝트를 구성하는 여러 개의 앱을 만들 수 있습니다.

쉽게 말하면 프로젝트가 회사고 앱이 각 부서들이라고 생각하면 됩니다



Model로 DB 생성해보기

프로젝트 – 앱 구조를 적용하기 위해 `urls.py`에 해당 코드를 추가해줍니다.

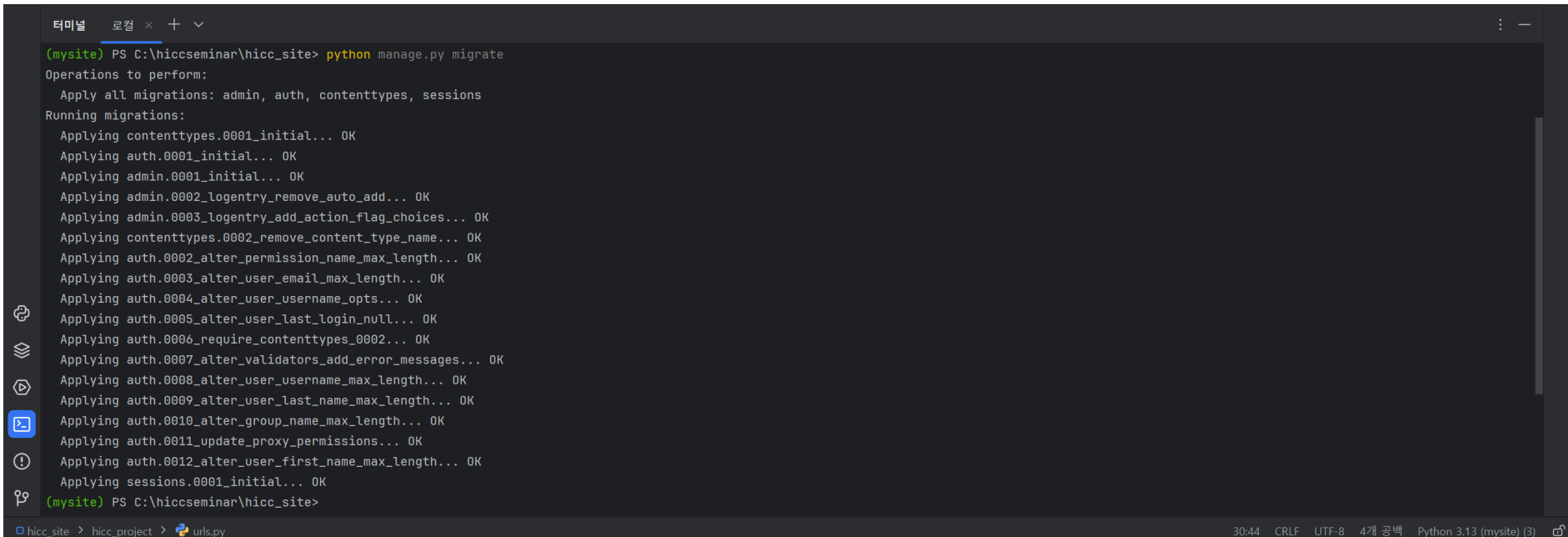
```
from django.contrib import admin
from django.urls import path, include
from hicc_project.views import (text_tag_render, html_test_render, image_tag_render,
form_tag_render, table_tag_render, layout_tag_render, http_test)

urlpatterns = [
    path('admin/', admin.site.urls),
    path('html_test/', html_test_render),
    path('text_tag/', text_tag_render),
    path('image_tag/', image_tag_render),
    path('form_tag/', form_tag_render),
    path('table_tag/', table_tag_render),
    path('layout_tag/', layout_tag_render),
    path('test/', http_test),
    path("", include('hicc_app.urls')),
]
```

추가로 `hicc_app` 디렉토리에 `urls.py` 파일을 따로 생성해줘야합니다.

Model로 DB 생성해보기

Django에는 미리 구현된 여러 기능들이 있습니다.
이를 서버에 적용시키려면 migrate 명령어를 사용해야 합니다.
이번 세미나에선 Model 기능을 사용하기 위해 migrate 명령어를 사용하겠습니다.

A screenshot of a Windows terminal window with a dark background. The prompt is (mysite) PS C:\hiccseminar\hicc_site>. The command python manage.py migrate has been entered. The output shows the operations to be performed (admin, auth, contenttypes, sessions) and a list of 12 migrations being applied, each followed by 'OK'. The terminal window has a title bar with '터미널' and standard window controls. On the left side of the terminal, there are icons for file explorer, search, and other utilities. At the bottom, a status bar shows the current directory, file name, and encoding information.

```
(mysite) PS C:\hiccseminar\hicc_site> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(mysite) PS C:\hiccseminar\hicc_site>
```

터미널에 python manage.py migrate 명령어를 입력하고 위와 같이 뜨면 성공

Model로 DB 테이블 생성해보기

간단한 질문 페이지, 질문 생성 페이지를 구현하기 위해 Model을 통해 데이터들을 저장할 DB의 테이블들을 생성합니다.

hicc_app 폴더의 models.py 파일에 해당 코드를 입력합니다.

```
from Django.db import models
```

```
class Question(models.Model):  
    subject = models.CharField(max_length=200)  
    content = models.TextField()  
    create_date = models.DateTimeField()
```

```
class Answer(models.Model):  
    question = models.ForeignKey(Question, on_delete=models.CASCADE)  
    content = models.TextField()  
    create_date = models.DateTimeField()
```

Model로 DB 테이블 생성해보기

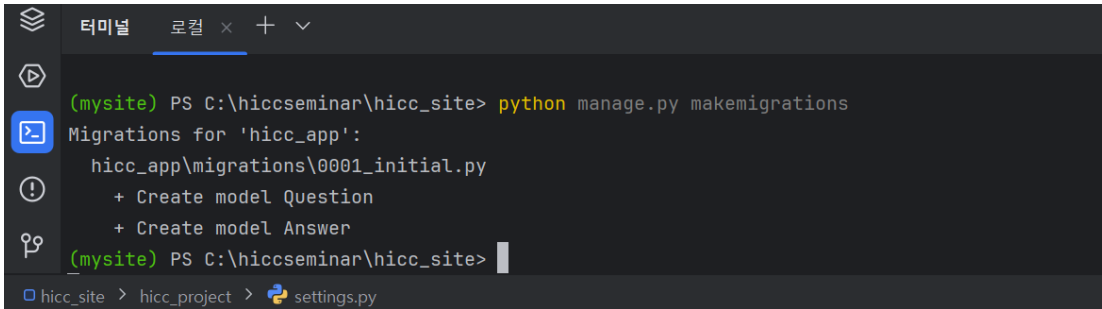
models.py 파일을 작성했으니, 이제 sqlite에 테이블을 실제로 생성합니다
그러기 위해 settings.py 파일에서 INSTALLED_APPS에 해당 부분을 추가해줍니다
Hicconfig 클래스는 apps.py 파일에 처음부터 구현되어 있습니다

```
# Application definition

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


Model로 DB 테이블 생성해보기

실제로 DB에 테이블을 생성하기 위해 터미널에 `python manage.py makemigrations` 명령어를 입력합니다.



```
(mysite) PS C:\hiccseminar\hicc_site> python manage.py makemigrations
Migrations for 'hicc_app':
  hicc_app\migrations\0001_initial.py
    + Create model Question
    + Create model Answer
(mysite) PS C:\hiccseminar\hicc_site>
```

위와 같은 상태가 나오고, 앱 폴더의 `migrations` 폴더에 아래와 같은 파일이 생성되면 성공입니다.
우리는 DB에 실제로 `models.py` 파일의 테이블을 구현할 준비를 마쳤다.



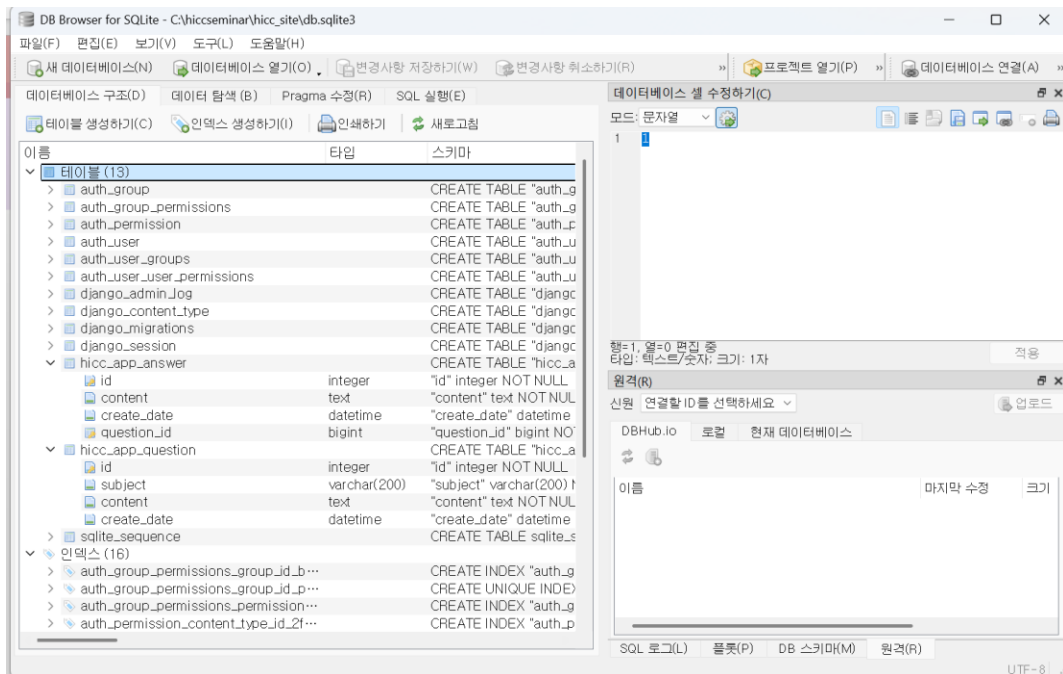
Model로 DB 테이블 생성해보기

다시 터미널에서 `python manage.py migrate` 명령어를 실행시키면, 실제로 DB에 우리가 작성한 테이블이 적용됩니다.

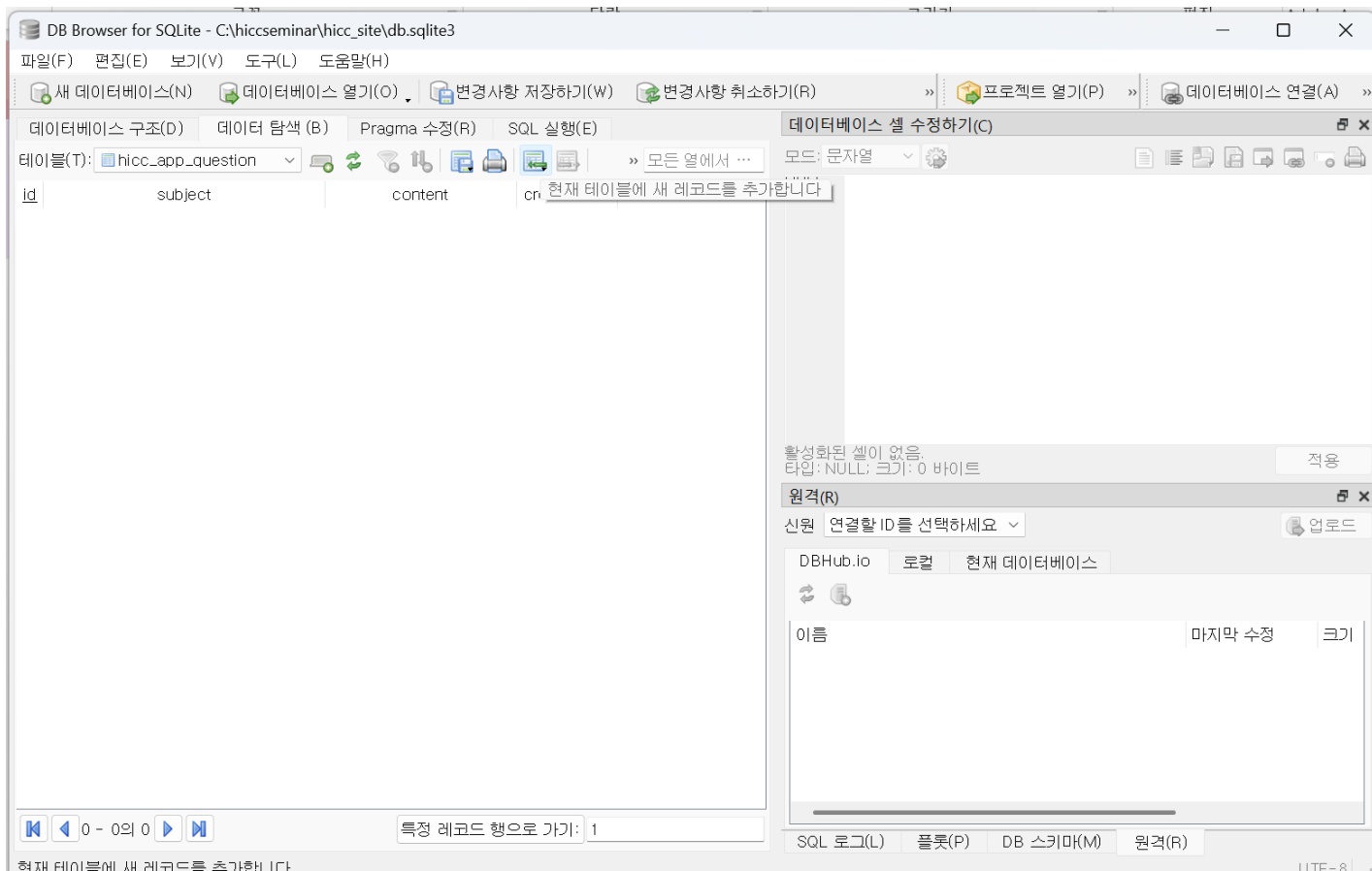


```
터미널 로컬 x + v
+ Create model Answer
(mysite) PS C:\hiccseminar\hicc_site> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, hicc_app, sessions
Running migrations:
  Applying hicc_app.0001_initial... OK
(mysite) PS C:\hiccseminar\hicc_site>
```

다운 받아왔던 `sqlite`로 프로젝트 폴더의 `db.sqlite3` 파일을 열어보면 실제 테이블들이 db에 구현되었음을 볼 수 있습니다.



SQLite에서 직접 데이터 넣기



SQLite의 해당 기능을 통해 데이터를 직접 넣어볼 수 있습니다.

Model 코드 다시보기

```
from django.db import models
```

```
class Question(models.Model): # Question은 테이블 subject, content, create_date는 속성
    subject = models.CharField(max_length=200) # CharField는 데이터 타입, 글자 타입을 의미하고 최대 길이는 200
    content = models.TextField() # TextField는 데이터 타입, 글자 타입이지만 최대 길이가 정해져 있지 않으면 TextField
    create_date = models.DateTimeField() # DateTimeField는 데이터 타입, 날짜 데이터를 저장하는 속성
```

```
class Answer(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    #ForeignKey는 외래키를 지정하는 함수, Question 클래스를 참조함, on_delete=models.CASCADE는 Question 클래스의 데이터가
    삭제됐을 때, Answer 데이터도 같이 삭제됨을 의미하는 옵션
    content = models.TextField()
    create_date = models.DateTimeField()
```

해당 코드와 같이 클래스에서 primary key를 따로 지정하지 않으면, **id**라는 **primary key** 속성이 자동으로 생성됨.

더 많은 필드와 옵션은

<https://github.com/dkyou7/TIL/blob/master/%ED%8C%8C%EC%9D%B4%EC%8D%AC/Django/5.%20%5BDjango%5D%20Model%20%E D%95%84%EB%93%9C%ED%83%80%EC%9E%85%20%EC%A0%95%EB%A6%AC.md> 참고

DB와 웹페이지 연동 확인

실제로 해당 DB의 데이터들이 페이지에 표시되는지 확인해봅시다.

코드 내용들 설명은 다음주 분량이므로, 현재 github의 코드들을 그대로 가져와서 확인만 해보겠습니다.

hicc_app/urls.py

https://github.com/hyeonhakjang/HICC-2025-Web-Dev-Seminar/blob/main/hicc_app/urls.py

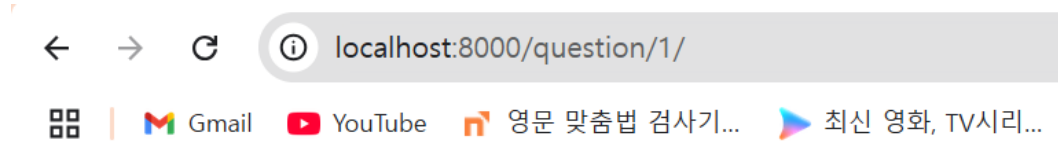
hicc_app/views.py

https://github.com/hyeonhakjang/HICC-2025-Web-Dev-Seminar/blob/main/hicc_app/views.py

templates/question_detail_ssr.html

https://github.com/hyeonhakjang/HICC-2025-Web-Dev-Seminar/blob/main/templates/question_detail_ssr.html

DB와 웹페이지 연동 확인



hicc 세미나는 언제인가요?

모르겠네요.

1개의 답변이 있습니다.

- 매주 수요일 6시입니다.

터미널에 `python manage.py runserver`를 실행후

`http://localhost:8000/question/1/`

해당 url에 접속하면 위와같이 db와 잘 연동됨을 확인할 수 있다.

다음주 예고

```
1 from django.shortcuts import render
2
3 # Create your views here.
4
5 from django.shortcuts import render, get_object_or_404
6 from .models import Question, Answer
7
8 def question_detail(request, question_id): 1개의 사용 위치  장현학#gusgk
9     question = get_object_or_404(Question, id=question_id)
10    answers = Answer.objects.filter(question=question)
11    return render(request, template_name: 'question_detail_ssr.html', context: {
12        'question': question,
13        'answers': answers,
14    })
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>질문 게시글</title>
6 </head>
7 <body>
8     <h1 id="subject">{{ question.subject }}</h1>
9     <div id="content">{{ question.content }}</div>
10    <p></p>
11
12    <h5>{{ answers|length }}개의 답변이 있습니다.</h5>
13    <div>
14        <ul>
15            {% for answer in answers %}
16                <li class="answer">{{ answer.content }}</li>
17            {% endfor %}
18        </ul>
19    </div>
20 </body>
21 </html>
```

다음주 예고

1주차. 웹 통신과 개발의 이해 (OT)

2주차. HTML, CSS, JS 개념 기초

3주차. 웹 프레임워크 이해 (Django를 활용한 페이지 랜더링) -- 정적 웹 서비스 구현

4주차. GIT 사용 실습 -- 필수 개발자 도구 익히기

5주차. MVC 패턴 (MTV) + 데이터베이스 기초

6주차. 네트워크 기초 + REST API (1) (Django ORM 하드코딩) -- 동적 웹 서비스 구현

7주차. REST API (2) (템플릿 프론트엔드 연결)

8주차. 최신 웹 개발 스택 맛보기 (DRF + React를 통한 -- 메인스트림 개발 스택을 활용하여 간편하게 개발하기
웹페이지 개발)