

# 03/31 (화) Django



## | Background

- ✓ 데이터베이스와의 상호작용 방식에 대한 이해

## | Goal

- ✓ 게시글을 작성하고 읽는 기본적인 동작 원리 학습

## | Problem

- ❖ 아래 제시된 정보를 참고하여 장고 프로젝트를 만들고 결과 사진과 코드를 제출하시오.
- ❖ Django 프로젝트 구조
  - **crud/**는 startproject 명령어로 생성되는 project 디렉토리이다.
  - **articles/**는 startapp 명령어로 생성되는 application 디렉토리이다.

```
crud/  
  templates/  
    base.html  
  __init__.py  
  settings.py  
  urls.py  
  wsgi.py  
articles/  
  migrations/  
  templates/  
    articles/  
      index.html  
      new.html  
      create.html  
  __init__.py  
  admin.py  
  apps.py  
  models.py  
  test.py  
  views.py  
manage.py
```



## | Problem

### 1. 기본 페이지 설정

- 결과 사진

## Articles

---

### 게시글 모음

- 해당 페이지의 접속 경로는 `/articles/` 이다.
- `settings.py`에서 앱 등록, 언어 설정 등의 기본적인 설정을 진행한다.
- `base.html`은 `articles` application 템플릿에서 모두 사용되는 부모 템플릿이다. 이를 위해, `settings.py`의 `TEMPLATES` 변수에 추가적인 설정을 진행한다.
- 해당 템플릿을 상속받아 사용하는 다른 템플릿에서 `content`라는 이름으로 `block`을 사용할 수 있도록 설정한다.
- `crud/urls.py`는 `articles/`로 시작하는 주소로 요청이 들어올 경우, `articles/urls.py`로 요청을 넘겨줄 수 있도록 해당 `urls`를 `include` 한다.
- `articles/urls.py`는 `articles/`로 시작하는 주소가 정의되어 있으며, 요청이 들어온 주소를 확인하여 그에 맞는 `views.py` 내의 함수를 호출한다.



## | Problem

### 2. Model 설정

- 결과 사진

```
Migrations for 'articles':  
articles/migrations/0001_initial.py  
- Create model Article
```

```
Applying sessions.0001_initial... OK
```

- models.py에 작성하는 class 이름은 **Article** 이다.
- 해당 클래스는 models.Model을 상속받는다.
- 해당 테이블에 지정되는 필드는 title과 content이다.
  - title의 글자수를 20자로 제한한다.
  - content는 글자수의 제한이 없다.
- 클래스 작성 후 터미널에서 models.py에 작성한 클래스를 기반으로 마이그레이션 작업을 실행한다.



## | Problem

### 3. 게시물 작성 form 생성

- 결과 사진

---

## Articles

---

### 글 작성

제목:

내용:

---

[BACK](#)

- 해당 페이지의 접속 경로는 /articles/new/ 이다.
- views.py에 new 함수를 작성한다.
- new.html은 base.html을 상속 받는다.
- 사용자에게 제목과 내용을 입력 받아 /articles/create/로 전송한다.
  - 이 때, 요청 방식은 GET 방식으로 진행한다.
  - 제목과 내용을 적는 각각의 input 태그는 "title", "content"라는 name 속성값을 가진다.
- BACK 링크를 클릭하면 index 페이지로 돌아간다.



## | Problem

### 4. 게시물 쓰기 (Create)

- 결과 사진

---

## Articles

---

성공적으로 글이 작성되었습니다.

[BACK](#)

- 해당 페이지의 접속 경로는 /articles/create/ 이다.
- new.html에서 사용자가 보낸 제목과 내용 데이터를 받아 title과 content 변수에 저장한다.
- models.py로부터 Article 클래스를 불러와 form을 통해 전달 받은 데이터를 삽입하고 저장한다.
- 저장이 완료되면 create.html을 렌더링해 '성공적으로 글이 작성되었습니다.'를 출력한다.
- BACK 링크를 클릭하면 index 페이지로 돌아간다.



## | Problem

### 5. 게시글 읽기 (Read)

- 결과 사진

## Articles

[NEW](#)

### 게시글 모음

- 제목: 제목 테스트  
내용: 내용 테스트
- 제목: 제목 테스트2  
내용: 내용 테스트2

- 해당 페이지의 접속 경로는 /articles/ 이다.
- views.py의 index 함수에서는 데이터베이스로부터 작성한 모든 게시글을 가져온다.
- index.html은 base.html을 상속 받는다.
- NEW 링크를 통해 index 페이지로 돌아갈 수 있다.



## | Problem

### ❖ 결과 제출

- 각각의 결과 사진과 아래 제시된 파일들의 코드를 마크다운 파일에 옮겨 제출하시오.

1. crud/settings.py

- TEPLATES 변수에 해당하는 코드만 작성하시오.

2. crud/urls.py

3. articles/urls.py

4. articles/views.py

5. crud/templates/base.html

6. templates/articles/index.html

7. templates/articles/new.html

8. templates/articles/create.html