

HomeWork

Django Model

- posts앱 안의 models.py 파일에 다음과 같은 코드를 작성하였다.

```
from django.db import models

class Post(models.Model):
    title = models.CharField(max_length=100)
    content = models.TextField()
```

- models.py를 작성한 후 마이그레이션 작업을 위해 터미널에 작성해야 하는 두 개의 핵심 명령어를 작성하시오.

```
$ python manage.py makemigrations #설계도 생성
$ python manage.py migrate #db에 반영 => table name : posts_post
```

- 다음 중 새로운 Post를 저장하기 위하여 작성한 코드 중 옳지 않은 것을 고르시오.

```
# 1
post = Post()
post.title = 'a'
post.content = 'b'
post.save()

#2
post = Post(title='가', content='나')
post.save()

#3
post = Post('제목', '내용')
post.save()

#4
Post.objects.create(title='1', content='2')
```

- 3번 : 인스턴스 생성시, 필드명도 함께 표기해 주어야 한다.

- Post가 10개 저장되어 있고 id의 값이 1부터 10까지라고 가정할 때 가장 첫 번째 Post를 가져오려고 한다. 다음 중 옳지 않은 코드를 고르시오.

```
#1
post1 = Post.objects.all()[0]
#2
post2 = Post.objects.all()[-10]
#3
post3 = Post.objects.all().first()
#4
post4 = Post.objects.all().get(id=1)
```

- 2번 : QuerySet은 list처럼 작동하는 것으로 보이지만 정확하게 List 인 것은 아니다.
- 음수 인덱싱은 지원하지 않는다.

4. my_post 변수에 Post 객체 하나가 저장되어 있다. title을 “안녕하세요” content를 “반갑습니다” 로 수정하기 위한 코드를 작성하시오.

```
# my_post = Post.objects.all().first()

my_post.title = '안녕하세요'
my_post.content = '반갑습니다'
my_post.save()
```

5. 만들어진 모든 Post 객체를 QuerySet형태로 반환 해주기 위해 빈칸에 들어갈 코드를 작성하시오.

```
posts = Post.__(a)__.__(b)__.()
```

- A : objects (manager)
- B : all (DB API)