

4. Photo 모델로 Admin 영역에서 데이터 다루기

1. Photo 모델로 데이터 넣기

1. Admin에서 Photo 모델에 데이터 넣기

- `admin.py`

```
from django.contrib import admin
from .models import Photo

admin.site.register(Photo) # Photo app 등록
```

- `models.py`

```
class Photo(models.Model):
    image = models.ImageField()
    filtered_image = models.ImageField()
    content = models.TextField(max_length=500, blank=True) #최대 길이 500자, 빈칸 허용
    created_at = models.DateTimeField(auto_now_add=True)
```

2. 파일 업로드 경로 지정

- `model.py`

```
class Photo(models.Model):
```

```
    image = models.ImageField(upload_to='uploads/%Y/%m/%d/orig') #파일 업로드 경로 지정
    filtered_image = models.ImageField(upload_to='uploads/%Y/%m/%d/filtered', blank=True)
    content = models.TextField(max_length=500, blank=True)
    created_at = models.DateTimeField(auto_now_add=True)
```

3. 첨부 파일 삭제하기

- 모델 객체 삭제할때 파일도 삭제
 - Django는 `delete` 메소드를 통해 모델 객체를 지움
 - `model.py`

```
class Photo(models.Model): # Photo: Model 클래스 상속
# delete 메소드 오버라이딩
    def delete(self, *args, **kwargs): # instance method's 1st parameter: self
        # *args, **kwargs: parameter를 미리 알지 못하는 경딩, 넘겨 받은 인자 그대로 전달하기 위
        함
        self.image.delete() # self.image속성 접근
        self.filtered_image.delete()
        super(Photo, self).delete(*args, **kwargs) # 부모 클래스(Model)의 delete 메소드 호
        출
```