



Static과 Media

11기 운영진 김민지

멋쟁이사자처럼 대학 11기

Static과 Media

Static

css, js, image 등과 같이 개발자가 사전에 미리 서버에 저장 해둔 파일

Media

사용자가 업로드한 파일(웹상에서 동적으로 부여되는 파일)

관리자가 admin 페이지에서 올리는 파일도 포함

Static - settings.py 수정

```
116 # Static files (CSS, JavaScript, Images)
117 # https://docs.djangoproject.com/en/4.1/howto/static-files/
118 import os
119
120 STATIC_URL = 'static/'
121 STATICFILES_DIRS = [
122     os.path.join(BASE_DIR, 'static'),
123 ]
```

STATIC_URL : static 파일에 접근하기 위한 url 설정

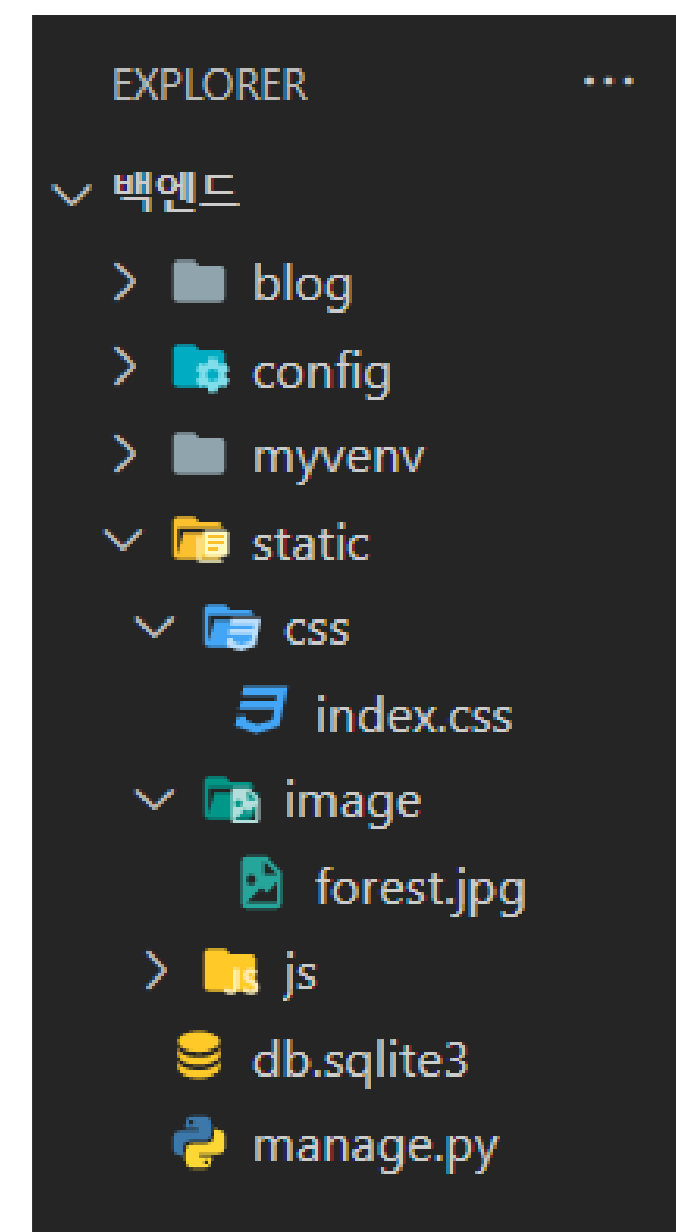
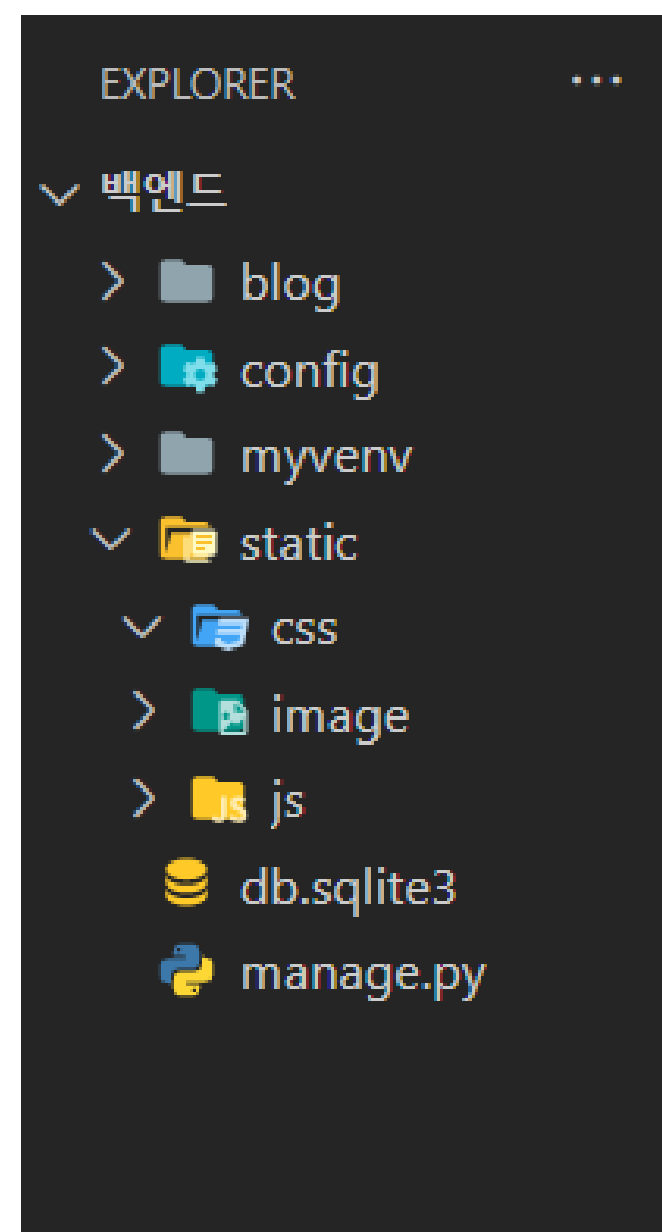
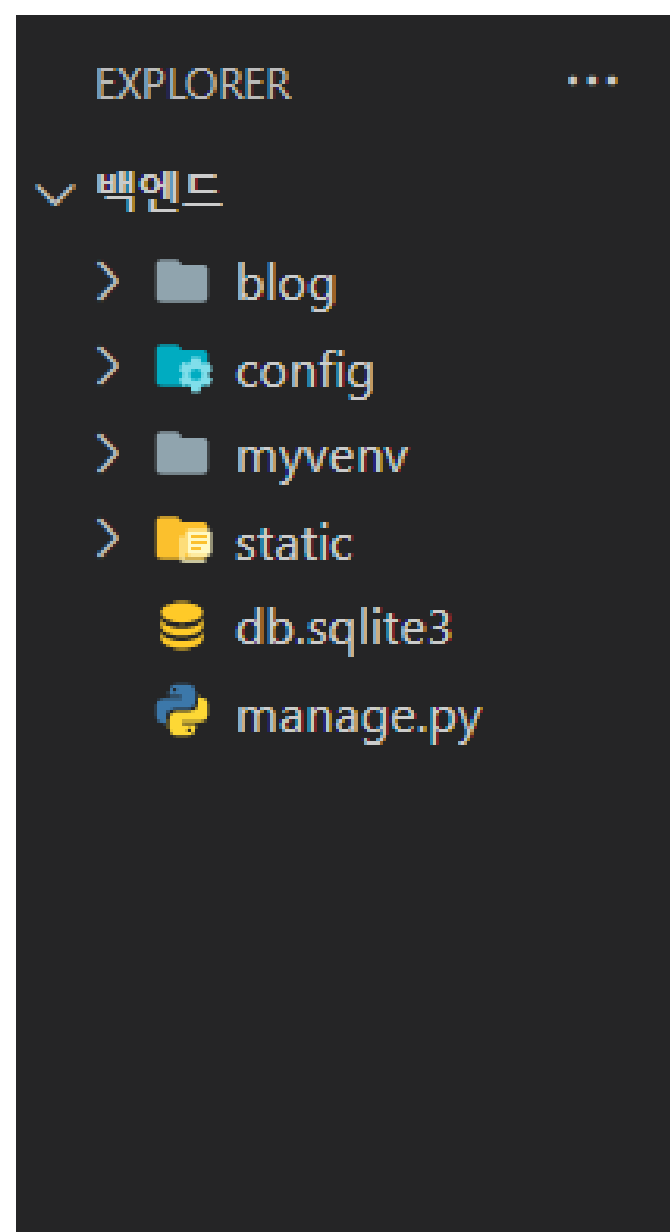
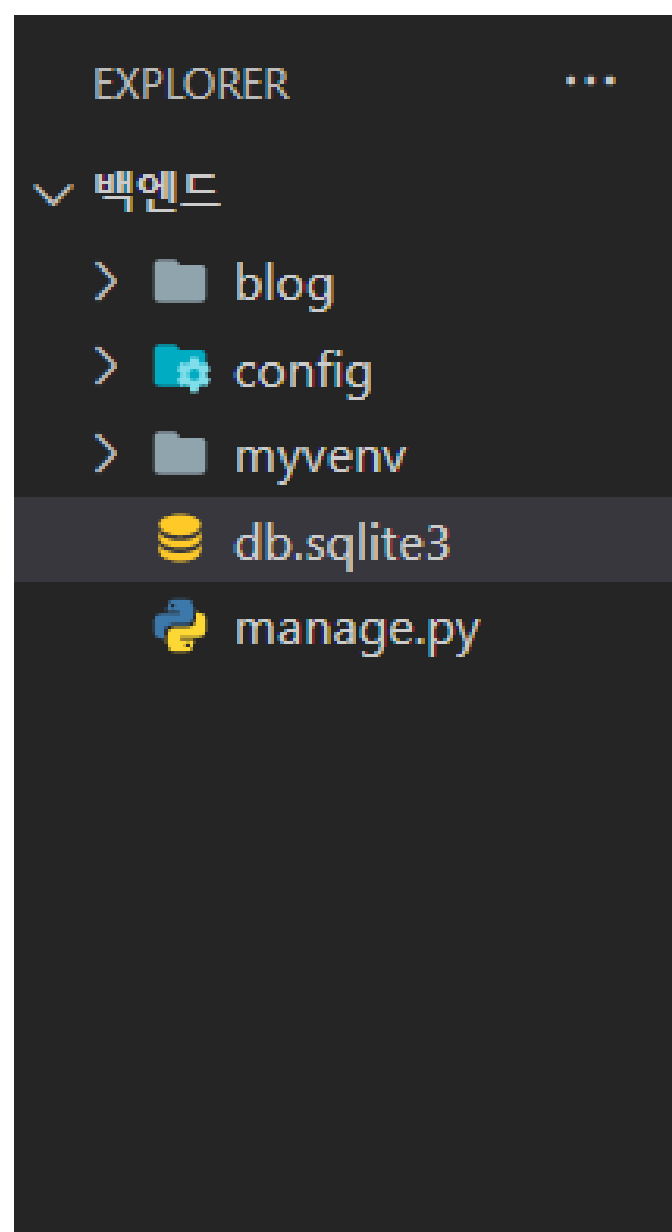
ex) {% static 'img/sample.jpg' %} 에 대해서 해당 url이 STATIC_URL+'경로'로 바뀌게 되고
이는 '/static/img/sample.jpg'로 바뀌게 되어 참조

STATICFILES_DIRS : static 파일들이 현재 어느 위치에 있는지 알려줌

⇒ BASE_DIR(루트 디렉터리)의 static 폴더에 있음

Static - 폴더 생성

1. 루트 디렉터터리에 New Folder를 선택해 이름을 static으로 생성
2. static 폴더에 css, image, js 폴더 생성
3. css, image, js파일들을 각각 해당하는 폴더에 넣기



Static - html 파일 수정

link, image, script 태그에 {% static '경로' %}로 static 경로를 작성

ex) CSS 파일

```
<link rel="stylesheet" href="{% static 'css/index.css' %}">
```

ex) Image 파일

```

```

```
<head>
|   <link rel="stylesheet" href="{% static 'css/index.css' %}" type="text/css">
</head>
<body>
|   
```

Media - setting.py, urls.py 수정

```
125 MEDIA_URL = 'media/'
126 MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

MEDIA_URL : media 파일에 접근하기 위한 url 설정

MEDIA_ROOT : 실제 사용자가 업로드한 파일들이 저장되는 경로

```
17 from django.contrib import admin
18 from django.urls import path, include
19 from django.conf import settings
20 from django.conf.urls.static import static
21 from blog import views
22
23 urlpatterns = [
24     path('admin/', admin.site.urls),
25     path('', include('blog.urls')),
26     path('', views.index, name="index"),
27 ] + static(settings.MEDIA_URL, document_root = settings.MEDIA_ROOT)
28
```

Media - models.py 수정

Image Field 추가하기

```
3  # Create your models here.
4  class Post(models.Model):
5      title = models.CharField(max_length=50)
6      pub_date = models.DateTimeField()
7      body = models.TextField()
8      photo = models.ImageField(upload_to='blog/', null=True, blank=True)
9
10     def __str__(self):
11         return self.title
```

upload_to : 어디에 저장할지 설정하는 옵션

따로 설정 안하면 media폴더에 저장되고, 다음과 같이 설정하면 media/blog 폴더에 저장

null과 blank : 이미지 첨부 안하고, 빈 값이 와도 상관 없음

이미지 필드를 사용하기 위해 pillow 라이브러리 설치

pip install pillow

model이 변경되었으니 makemigrations, migrate 다시 하기

Media - create.html 수정

<form>태그에 `enctype="multipart/form-data"` 추가
모든 문자를 인코딩 하지 않고, 파일이나 이미지를 서버로 전송

이미지 파일을 받을 수 있도록 `<input type="file">` 태그 추가

```
5  <form action="{% url 'create' %}" method="post" enctype="multipart/form-data">
6      {% csrf_token %}
7      제목: <input type="text", name="title">
8      내용: <textarea name="body"></textarea>
9      <input type="file" class="imaUploadBtn_icon" name="photo">
10     <input type="submit", value="제출">
11 </form>
```


Media - views.py 수정

html에서 보낸 사진을 views.py가 받을 수 있도록 `request.FILES['photo']` 추가

```
14  ✓ def create(request):
15  ✓     if request.method == 'POST':
16         post = Post()
17         post.title = request.POST['title']
18         post.body = request.POST['body']
19         post.pub_date = timezone.datetime.now()
20         post.photo = request.FILES['photo']
21         post.save()
22         return redirect('/detail/' + str(post.id))
23  ✓     else:
24         return render(request, 'create.html')
```

Media - detail.html 수정

태그의 src부분에 `{{post.photo.url}}`를 작성
`{{post.photo}}`만 적으면 사진의 url만 표시되고 정상적으로 표시되지 않음

```
5  <h1>{{ post_detail.title }}</h1>
6  <p>{{ post_detail.body }}</p>
7  
8  <p>{{ post_detail.pub_date }}</p>
9  <a href="{% url 'update' post_detail.id %}"><button>수정</button></a>
10 <a href="{% url 'delete' post_detail.id %}"><button>삭제</button></a>
```

과제

블로그 css, image 적용하기
글쓰기에 사진 첨부 추가해서 디테일 페이지에서 보여주기