

정적 파일

- HTML을 제외하고 웹 페이지를 렌더링 할 때 필요한 추가적인 파일을 의미한다.
- 서버의 데이터들이 어느정도 가공된 후 보여지는 동적파일과 달리, 서버에 미리 저장되어 있어야하며, 서버에 저장된 그대로를 서비스한다.
- 장고에서 정적파일은 아래의 두가지 파일로 분류될 수 있다.
 - **Static 파일** : 이미지, Javascript, CSS처럼 개발자가 서버를 개발할 때 미리 넣어 준비해 놓은 파일로, 서비스 중에 추가되거나 수정되지 않고 고정되어 있음
 - **Media 파일** : 사용자가 업로드 할 수 있는 파일로, 파일 자체는 고정되어 있지만 언제 어떤 파일이 제공될 지는 예측할 수 없음

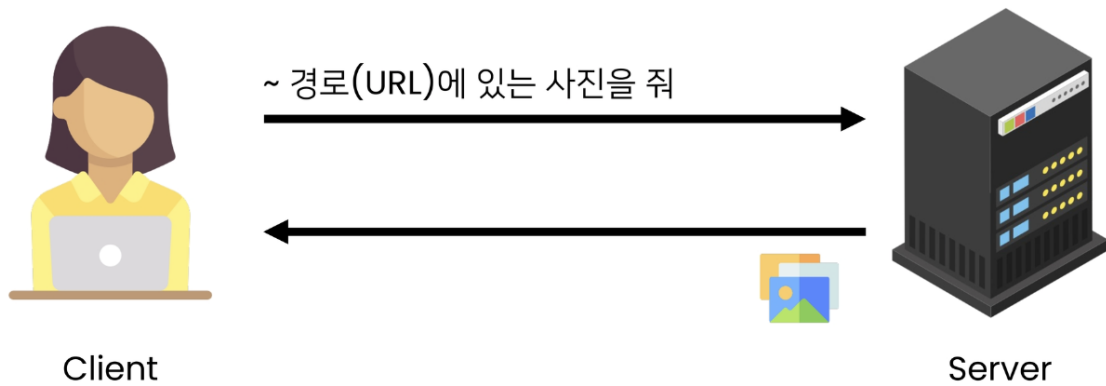
Static File

- 응답할 때 별도의 처리 없이 파일 내용을 그대로 보여주면 되는 파일
 - 사용자의 요청에 따라 내용이 바뀌는 것이 아니라 요청한 것을 그대로 보여주는 파일
- 파일 자체가 고정되어 있고, 서비스 중에도 추가되거나 변경되지 않고 고정되어 있음
 - 예를 들어, 웹사이트는 일반적으로 이미지, 자바 스크립트 또는 CSS와 같은 미리 준비된 추가 파일(움직이지 않는)을 제공해야 함
- Django에서는 이러한 파일들을 “static file”이라 함
 - Django는 `staticfiles` 앱을 통해 정적 파일과 관련된 기능을 제공

Media File

- 미디어 파일
- 사용자가 웹에서 업로드하는 정적 파일(user-uploaded)
- 유저가 업로드 한 모든 정적 파일

웹 서버와 정적 파일



- 웹 서버의 기본 동작은
 - 특정 위치(URL)에 있는 자원을 요청(HTTP request) 받아서
 - 응답(HTTP response)을 처리하고 제공(serving)하는 것
- 이는 '자원과 자원에 접근 가능한 주소가 있다'라는 의미
 - 예를 들어, 사진 파일은 자원이고, 해당 사진 파일을 얻기 위한 경로인 웹주소 (URL)가 존재함
- 즉, 웹 서버는 요청 받은 URL로 서버에 존재하는 정적 자원(static resource)을 제공함

Static files 구성하기

Django에서 정적 파일을 구성하고 사용하기 위한 몇가지 단계

1. `INSTALLED_APPS` 에 `django.contrib.staticfiles` 가 포함되어 있는지 확인하기
2. `settings.py` 에서 `STATIC_URL` 을 정의하기

```
# settings.py

INSTALLED_APPS = [
    'articles',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
```

```

]

...

STATIC_URL = '/static/'
STATICFILES_DIRS = [
    BASE_DIR / 'static',
]

```

3. 앱의 static 폴더에 정적 파일 위치 하기

- ex) `my_app/static/sample_img.jpg`

4. 템플릿에서 static 템플릿 태그를 사용하여 지정된 경로에 있는 정적 파일의 URL 만들기

```

{% load static %}



```

Django template tag

{% load %}

- load tag
- 특정 라이브러리, 패키지에 등록된 모든 템플릿 태그와 필터 로드

{% static ' ' %}

- static tag
- STATIC_ROOT에 저장된 정적 파일에 연결

Static files 관련 Core Settings

STATIC_ROOT

- Default : None
- Django 프로젝트에서 사용하는 모든 정적 파일을 한 곳에 모아 넣는 경로
- `collectstatic` 이 배포를 위해 정적 파일을 수집하는 디렉토리의 절대 경로
- 개발 과정에서 settings.py의 DEBUG 값이 True로 설정되어 있으면 해당 값은 적용되지 않음

- 실 서비스 환경(배포 환경)에서 Django의 모든 정적 파일을 다른 웹 서버가 직접 제공하기 위해 사용
- 배포 환경에서는 Django를 직접 실행하는 것이 아니라, 다른 서버에 의해 실행되기 때문에 실행되는 다른 서버는 Django에 내장되어 있는 정적 파일들을 인식하지 못함
(내장되어 있는 정적 파일들을 밖으로 꺼내는 이유)

[참고] collectstatic

- STATIC_ROOT에 Django 프로젝트의 모든 정적 파일을 수집

```
# settings.py

STATIC_ROOT = BASE_DIR / 'staticfiles'
```

```
$ python manage.py collectstatic
```

STATICFILES_DIRS

- Default : [] (empty list)
- `app/static/` 디렉토리 경로를 사용하는 것(기본 경로) 외에 추가적인 정적 파일 경로 목록을 정의하는 리스트
- 추가 파일 디렉토리에 대한 전체 경로를 포함하는 문자열 목록으로 작성되어야 함

```
# 작성 예시

STATICFILES_DIRS = [
    BASE_DIR / 'static',
]
```

STATIC_URL

- Default : None
- STATIC_ROOT에 있는 정적 파일을 참조 할 때 사용할 URL
- 개발 단계에서는 실제 정적 파일들이 저장되어 있는 `app/static/` 경로(기본경로) 및 `STATICFILES_DIR`에 정의된 추가 경로들을 탐색
- 실제 파일이나 디렉토리가 아니며, URL로만 존재

- 비어 있지 않은 값으로 설정한다면 반드시 slash(/)으로 끝나야 함

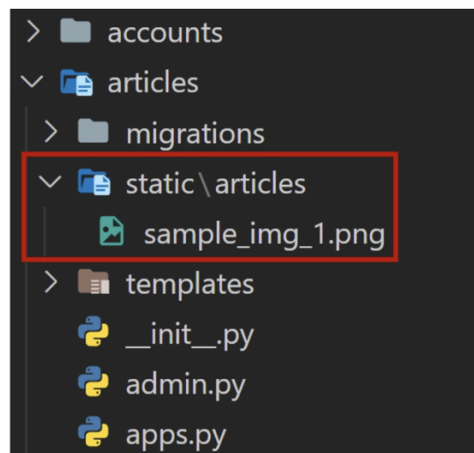
```
# 작성 예시
```

```
STATIC_URL = '/static/'
```

Static files 사용하기

방법 1 - 기본 경로에 있는 static file 가져오기

1. `articles/static/articles` 경로에 이미지 파일 배치하기



2. static tag를 사용해 이미지 파일 출력하기

```
{% load static %}
```

```

```

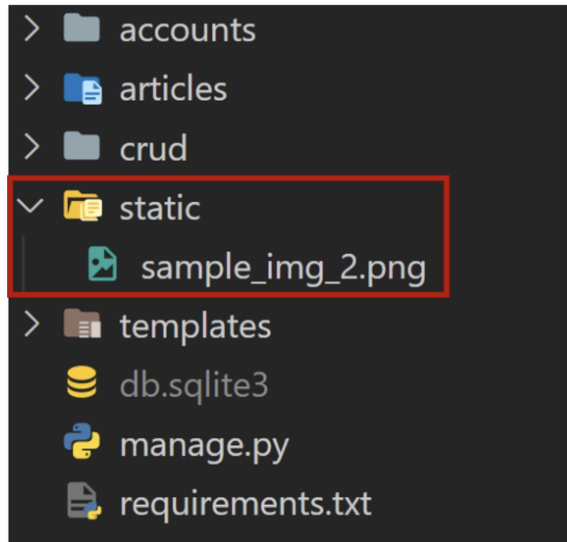
방법 2 - 추가 경로에 있는 static file 가져오기

1. 추가 경로 작성

```
# settings.py
```

```
STATICFILES_DIRS = [  
    BASE_DIR / 'static',  
]
```

2. `static/` 경로에 이미지 파일 배치하기



3. static tag를 사용해 이미지 파일 출력하기

```
{% load static %}


```

STATIC_URL 확인하기

- Django가 해당 이미지를 클라이언트에게 응답하기 위해 만든 image url 확인하기
 - 개발자 도구 - inspect 버튼을 통해 확인
- `STATIC_URL + static file 경로` 로 설정됨