

# Lecture 5.

## 정적(static) 파일 및 미디어 파일 관리



# Agenda

Ⅰ — 5주차 이론 강의

Ⅱ — 5주차 실습



# 정적(static) 파일 및 미디어 파일 관리



## II \_ 들어가기 앞서

### 1 대문페이지와 자기소개 페이지 만들기

- ✓ FBV 방식을 활용하여, `single_pages` 앱의 대문 페이지/자기소개 페이지 만들기
  - 대문페이지 만들기 (127.0.0.1:8000/) 로 접속 시 가장 처음 보이는 페이지
  - 현재는 아무 페이지도 없기 때문에 접속시 404 오류 도출
  - 순서
    - 1. `do_it_django_prj/urls.py` 수정
    - 2. 대문 페이지와 자기소개 페이지의 URL 지정하기
    - 3. `single_pages/view.py` 수정
    - 4. 템플릿 파일 만들기(`landing.html` , `about_me.html`)

## II \_ 들어가기 앞서

### 1 대문페이지와 자기소개 페이지 만들기

- ✓ FBV 방식을 활용하여, single\_pages 앱의 대문 페이지/자기소개 페이지 만들기
  - 순서
    - 1. do\_it\_django\_prj/urls.py 수정

실습 파일: do\_it\_django\_prj/urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('blog/', include('blog.urls')),
    path('admin/', admin.site.urls),
    path('', include('single_pages.urls')),
]
```

## II \_ 들어가기 앞서

### 1 대문페이지와 자기소개 페이지 만들기

- ✓ FBV 방식을 활용하여, single\_pages 앱의 대문 페이지/자기소개 페이지 만들기
  - 순서
    - 2. 대문 페이지와 자기소개 페이지의 URL 지정하기

실습 파일: single\_pages/urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('about_me/', views.about_me),
    path('', views.landing),
]
```

## II \_ 들어가기 앞서

### 1 대문페이지와 자기소개 페이지 만들기

- ✓ FBV 방식을 활용하여, single\_pages 앱의 대문 페이지/자기소개 페이지 만들기
  - 순서
    - 3. single\_pages/view.py 수정

실습 파일: single\_pages/views.py

```
from django.shortcuts import render

def landing(request):
    return render(
        request,
        'single_pages/landing.html'
    )

def about_me(request):
    return render(
        request,
        'single_pages/about_me.html'
    )
```

## II \_ 들어가기 앞서

### 1 대문페이지와 자기소개 페이지 만들기

- ✓ FBV 방식을 활용하여, single\_pages 앱의 대문 페이지/자기소개 페이지 만들기
  - 순서
    - 4. 템플릿 파일 만들기(landing.html , about\_me.html)

실습 파일: single\_pages/templates/single\_pages/landing.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>달타냥입니다!</title>
</head>
<body>
<nav>
  <a href="/blog/">Blog</a>
  <a href="/about_me/">About me</a>
</nav>

<h1>안녕하세요. 달타냥입니다.</h1>
<h2>대문 페이지</h2>
<h3>아직 만들지 않음</h3>
</body>
</html>
```

실습 파일: single\_pages/templates/single\_pages/about\_me.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>개발자 달타냥입니다.</title>
</head>
<body>

<nav>

  <a href="/blog/">Blog</a>
  <a href="/about_me/">About me</a>
</nav>

<h1>안녕하세요. 달타냥입니다.</h1>
<h2>이력</h2>
<h2>Portfolio</h2>
<h3>아직 공사중입니다. </h3>
</body>
</html>
```



## II 정적 / 미디어 파일 관리

### 2 정적 파일 관리

✓ HTML 파일에 CSS 파일이 적용되는 과정



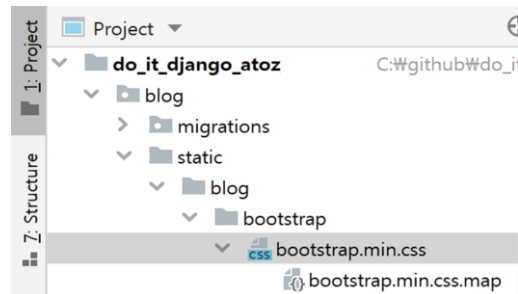
- Django에서는 JS, CSS, 이미지 등의 파일들을 정적(static)파일이라 함
- Templates 의 HTML 파일은 정적 파일이 아님
- 클라이언트에게 CSS를 제공하기 위해서 정적 파일을 따로 제공해야함

## II 정적 / 미디어 파일 관리

### 2 정적 파일 관리

✓ App 폴더에 static 폴더 만들고, CSS 파일 넣기

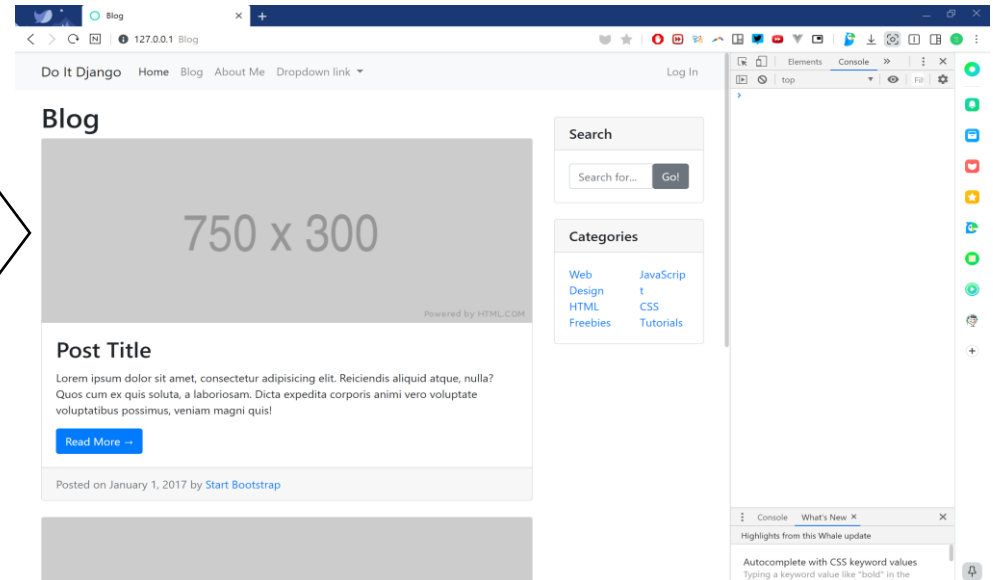
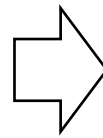
- App 폴더 안에 static 폴더 만든 뒤, 그 안에 다시 App 이름으로 된 폴더 생성
- 셋팅 후, 다시 runserver 실행하면, static 파일 인식



실습 파일: blog/templates/blog/post\_list.html

```
<!DOCTYPE html>
{% load static %}
<html lang="ko">
<head>
  <title>Blog</title>
  <link rel="stylesheet" href="{% static 'blog/bootstrap/bootstrap.min.css' %}"
media="screen">

  <script src="https://kit.fontawesome.com/*****.js"
crossorigin="anonymous"></script>
</head>
(...생략...)
```



## II \_ 정적 / 미디어 파일 관리

### 2 정적 파일 관리

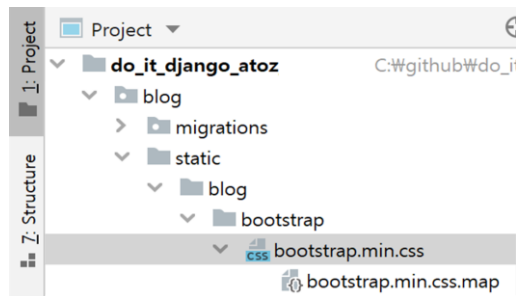
- ✓ App 폴더에 static 폴더 만들고, CSS 파일 넣기
  - <https://getbootstrap.com/docs/5.1/getting-started/download/>
  - Compiled CSS and JS 에서 zip 파일 Download 이후, CSS 폴더 내의 bootstrap.min.css 파일과 bootstrap.min.css.map 파일을 프로젝트로 복사
  - 복사 위치 : blog 폴더(최상위 App) -> static 폴더 생성 -> blog 폴더 생성 -> bootstrap 폴더 생성 -> 그 아래 파일 붙여 넣기

## II 정적 / 미디어 파일 관리

### 3 Bootstrap template 가져오기

#### ✓ 2주차 실습한 Bootstrap 파일 가져오기

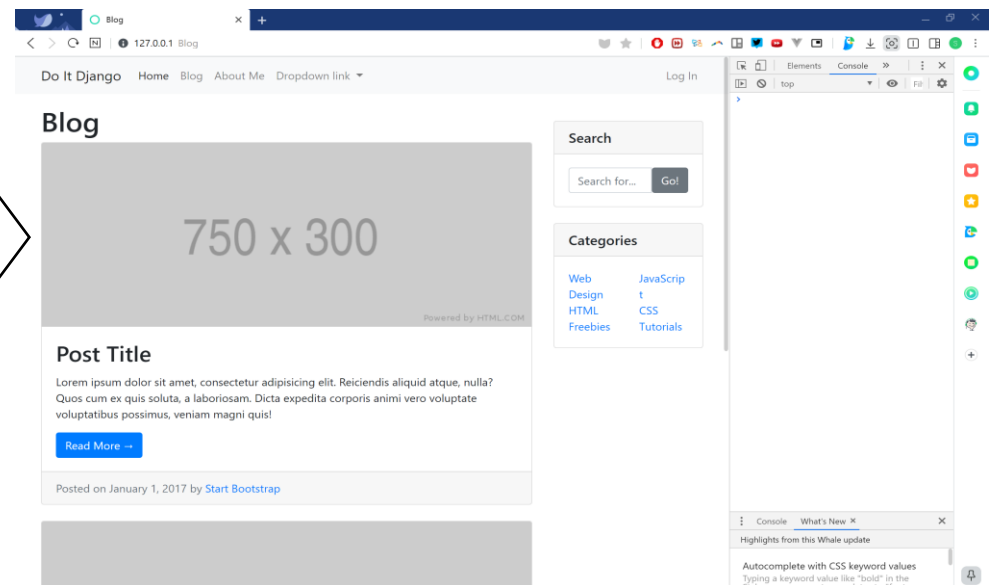
- Lms 에 업로드된 '2주차 실습코드 2' 파일 다운
- Blog\_list.html 파일을 열고, post\_list.html 에 bootstrap 실습 내용을 붙이기
- Bootstrap 템플릿이 잘 적용되었는지 확인



실습 파일: blog/templates/blog/post\_list.html

```
<!DOCTYPE html>
{% load static %}
<html lang="ko">
<head>
  <title>Blog</title>
  <link rel="stylesheet" href="{% static 'blog/bootstrap/bootstrap.min.css' %}"
media="screen">

  <script src="https://kit.fontawesome.com/*****.js"
crossorigin="anonymous"></script>
</head>
(...생략...)
```



## II 정적 / 미디어 파일 관리

### 4 미디어 파일 관리하기

#### 업로드한 파일이 저장될 폴더 지정하기

실습 파일: do\_it\_django\_prj/settings.py

```
import os
from pathlib import Path
(...생략...)
STATIC_URL = '/static/'

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

#### 이미지 파일을 저장할 수 있는 ImageField

실습 파일: blog/models.py

```
from django.db import models

class Post(models.Model):
    title = models.CharField(max_length=30)
    content = models.TextField()

    head_image = models.ImageField(upload_to='blog/images/%Y/%m/%d/', blank=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    # author: 추후 작성 예정
    (...생략...)
```

Settings.py에서 설정한 \_media 폴더 아래,  
blog/images/ 폴더 아래, 연도, 월, 일 순서로 저장

한 폴더에 너무 많은 파일을 저장하면,  
서버에서 파일을 찾을 때 많은 부담을 주게 됨

blank=True 라고 하면,  
필수항목은 아니라는 뜻

#### 데이터베이스에 변경사항 반영하기

Cmdr

```
C:\github\do_it_django_a_to_z (main -> origin)
(venv) λ pip install Pillow
C:\github\do_it_django_a_to_z (main -> origin)
(venv) λ python manage.py makemigrations
C:\github\do_it_django_a_to_z (main -> origin)
(venv) λ python manage.py migrate
```

Python에서 이미지 처리  
를 위해 필요

Head image: Currently: blog/images/2020/05/25/world\_time.jpg Clear

Change: 파일 선택 선택된 파일 없음

1 원하는 이미지 선택

2 클릭

Delete

Save and add another

Save and continue editing

SAVE

## II 정적 / 미디어 파일 관리

### 4 미디어 파일을 위한 URL 처리하기

#### 미디어 파일을 위한 URL 지정하기

실습 파일: do\_it\_django\_prj/urls.py

```
from django.contrib import admin
from django.urls import path, include

from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('blog/', include('blog.urls')),
    path('admin/', admin.site.urls),
    path('', include('single_pages.urls')),
]

urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

#### head\_image의 이미지 파일을 템플릿에서 출력

실습 파일: blog/templates/blog/post\_list.html

```
(...생략...)
{% for p in post_list %}
<!-- Blog Post -->
<div class="card mb-4">
    
(...생략...)
```

## II 정적 / 미디어 파일 관리

### 4 미디어 파일을 위한 URL 처리하기

head\_image의 이미지 파일이 없는 경우, 템플릿 문법으로 처리

```
<div class="container my-3">
  <div class="row">
    <div class="col-md-8 col-lg-9">
      {% for p in post_list %}
        <!-- Blog Post -->
        <div class="card mb-4">
          {% if p.head_image %}
            
          {% else %}
            
          {% endif %}
          <div class="card-body">
            <h2 class="card-title">{{ p.title }}</h2>
            <p class="card-text">{{ p.content }}</p>
            <a href="{{ p.get_absolute_url }}" class="btn btn-primary">Read More &arr;</a>
          </div>
          <div class="card-footer text-muted">
            Posted on {{ p.create_at }}
            <a href="#">개발예정</a>
          </div>
        </div>
      {% endfor %}
    </div>
  </div>
</div>
```

## II 정적 / 미디어 파일 관리

### 5 파일 업로드를 위한 FileField

실습 파일: blog/models.pyl

```
from django.db import models

class Post(models.Model):
    title = models.CharField(max_length=30)
    content = models.TextField()

    head_image = models.ImageField(upload_to='blog/images/%Y/%m/%d/', blank=True)
    file_upload = models.FileField(upload_to='blog/files/%Y/%m/%d/', blank=True)

    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    # author: 추후 작성 예정
    (...생략...)
```



# Q & A

