

Djagno 기반의 웹프로그래밍

이 소 영

yisy0703@naver.com

Web Frameworks for Python(<https://wiki.python.org/moin/WebFrameworks>)

장고 공식 사이트(<https://www.djangoproject.com/>)

장고 공식 소스 저장소(<http://github.com/django/django>)

장고 참조 문서(<https://docs.djangoproject.com/ko/5.2/>)

Agenda

Django(장고) 기반의 파이썬 웹 프로그래밍

Ch01. Django 시작하기

1. Django 란?
2. 개발 환경 구축
3. Django 구조

Ch02. Django App

1. Django Project
2. Model
3. View

Ch03. Model

1. Model 속성 및 옵션
2. Relationship
3. Migrations
4. Admin App

Ch04. Django SQL

1. Django shell
2. Manager & QuerySet
3. 조회 SQL
4. 생성/수정/삭제 SQL

Ch05. Template

1. Template Loader
2. URL Dispatcher
3. Template 상속
4. Template Engines
5. Template Filter

Ch07. Django View

1. View 기본
2. View 활용

Ch06. Django Form

1. HTML form
2. CSRF
3. HttpRequest/HttpResponse
4. Django Form
5. Django Model Form
6. Form Validation

Ch08. File 관리

1. Static Files
2. Media Files

Ch09. 사용자 인증

1. Auth App
2. 회원가입 구현
3. 로그인/아웃 구현
4. OAuth 라이브러리 활용

Ch 08 File 관리

1. Static Files
2. Media Files

Chapter 08. File 관리

- Managing static files
(<https://docs.djangoproject.com/ko/5.2/howto/static-files/>)
- Deploying static files
(<https://docs.djangoproject.com/en/5.2/howto/static-files/deployment/>)
- 개발서버에서 media 파일 서빙
(<https://github.com/django/django/blob/master/django/conf/urls/static.py>)
- Pillow
(<https://pillow.readthedocs.io/>)
- strftime 포매팅
(<https://docs.python.org/3/library/time.html#time.strftime>)
- django-imagekit
(<https://github.com/matthewwithanm/django-imagekit/blob/develop/imagekit/generatorlibrary.py#L13>)



1. Static Files



Static and Media Files

1. Static Files

● Static Files

- 개발 리소스 파일 (images, JavaScript, CSS 등)
 - 앱 단위로 저장/서빙
 - 프로젝트 단위로 저장/서빙

● Media Files

- 유저가 업로드한 모든 정적인 파일 (image, etc)
 - 프로젝트 단위로 저장/서빙

Static Files

1. Static Files

- <https://docs.djangoproject.com/ko/5.2/howto/static-files/>
- settings.py

```
STATIC_URL = '/static/'

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'myproject', 'static'),
]

STATIC_ROOT = os.path.join(BASE_DIR, '_staticfiles') #운영시 경로
```

- Static file의 위치
 - AppDirectoriesFinder : '앱/static' 경로
 - FileSystemFinder : settings.STATICFILES_DIRS = [] 의 경로
 - static 파일의 검색 대상은 장고 서버 초기 시작시 지정됨

Static Files URL

1. Static Files

- 방법1

```

```

- 방법2

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

```

```

```
<link rel="stylesheet" href="/static/style.css" />
```

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

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


개발 서버 static 파일 서빙

1. Static Files

- 개발서버를 통해 settings.DEBUG = True 설정에 한해 서빙 지원
- 프로젝트/urls.py 에 Rule이 명시되어 있지 않아도 자동 Rule 추가
- 장고를 통해 직접 static 파일 서빙하는 것은 개발목적으로만 제공
- 실제로는 nginx, apache, AWS S3, CDN, Azure storage 등의 서버에서 파일이 서빙되어야 함

collectstatic

1. Static Files

- <https://docs.djangoproject.com/ko/5.2/howto/static-files>
- 여러 디렉토리에 나눠 저장된 static 파일들은 개발 환경에서만 인식됨
- 외부 웹서버에서 서빙하기 위해 한 디렉토리로 복사 (settings.STATIC_ROOT)

```
STATIC_ROOT = os.path.join(BASE_DIR, '_staticfiles')
```
- D:\src\myproject> `python manage.py collectstatic`



2. Media Files



Media Fields

2. Media Files

- 사용자가 업로드한 모든 정적인 파일
- 프로젝트 단위로 저장/서빙
- HttpRequest.FILES을 통해 View로 전달
- settings.MEDIA_ROOT 디렉터리에 저장
- settings.py

```
MEDIA_URL = '/media/'  
# media 파일에 대한 URL Prefix  
MEDIA_ROOT = os.path.join(BASE_DIR, '_media')  
# 업로드된 파일을 저장할 디렉터리 경로
```

FileField / ImageField

2. Media Files

- 파일은 settings.MEDIA_ROOT 경로에 저장
- DB 필드에는 경로만 저장

```
# settings.py
```

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

```
>> pip install pillow
```

```
# article/models.py
```

```
class Article(models.Model):  
    title = models.CharField(max_length=100)  
    body = models.TextField()  
    status = models.CharField(max_length=1, choices=STATUS_CHOICES)  
    photo = models.ImageField(blank=True)
```

form enctype

2. Media Files

- 파일 업로드시 <form>태그 속성

- method = "POST"
- enctype = "multipart/form-data"

```
# article_form.html
```

```
<form method="post" enctype="multipart/form-data">
  {% csrf_token %}
  {{ form.as_p }}
  <input type="submit" value="Save" />
</form>
```

파일 필드

2. Media Files

● FileField

- File Storage API(FileSystemStorage)를 통해 파일을 저장
- blank=True 지정하여 옵션 필드로 지정함

● ImageField

- FileField를 상속함
- Pillow 이미지 처리 라이브러리를 내부적으로 사용해서 이미지를 처리함

● 파일 저장 경로

- settings.MEDIA_ROOT/파일명
- DB 필드는 파일명만 저장

upload_to

2. Media Files

- 파일 저장 경로 커스텀

- 필드별로 다른 디렉터리 저장

```
class Article(models.Model):  
    photo = models.ImageField(blank=True, upload_to='article')
```

- 업로드 시간대별로 다른 디렉터리 저장

```
class Article(models.Model):  
    photo = models.ImageField(blank=True, upload_to='article/%Y/%m/%d')
```

- strftime 포매팅 지원

<https://docs.python.org/ko/3/library/time.html#time.strftime>

개발 서버 media 파일 서빙

2. Media Files

- 개발 서버에서 media 파일 미지원
- 개발 편의 목적으로 서버 urls.py 추가

```
# 사용자가 업로드한 파일(MEDIA_ROOT)을 url(MEDIA_URL)과 연결
# 장고는 static은 자동 연결해 주나, media는 개발자가 연결
from django.conf.urls.static import static
from . import settings
urlpatterns += static(settings.MEDIA_URL,
                      document_root=settings.MEDIA_ROOT)
```

템플릿의 media 파일 URL

2. Media Files

- media 파일 URL 직접 지정하기

```

```

- 모델 필드 값 사용하기

```

```

- FileField/ImageField의 URL 속성 사용하기

```
  
➔ photo 필드가 없는 글은 오류 발생  
  
{% if article.photo %}  
    <img src="{{article.photo.url{% endif %}
```