

Django 보조자료

Django 설치-1

1. 가상 환경 생성 (venv는 가상 환경 이름, 원하는 이름으로 변경 가능)

```
python -m venv venv
```

2. 가상 환경 활성화

Windows

```
.\venv\Scripts\activate
```

Linux/macOS

```
source venv/bin/activate
```

Django 설치-2

pip를 최신 버전으로 업데이트하는 것도 좋습니다.

```
python -m pip install --upgrade pip
```

Django 설치

```
pip install Django
```

```
python -m django --version
```

Django –프로젝트, App 생성

<프로젝트명>은 원하는 이름으로 지정 (예: myproject)

```
django-admin startproject <프로젝트명>
```

예시:

```
django-admin startproject mysite
```

프로젝트 폴더로 이동

```
cd mysite
```

앱 생성 (<앱 이름>은 원하는 이름으로 지정, 예: main)

```
python manage.py startapp <앱 이름>
```

예시:

```
python manage.py startapp main
```

Django –프로젝트, App 생성

1.생성한 앱을 Django 프로젝트에 등록해야 합니다.

mysite/settings.py 파일을 엽니다.

INSTALLED_APPS 리스트에 생성한 앱 이름('main')을 추가합니다.

```
# mysite/settings.py
```

```
INSTALLED_APPS = [
```

```
    # ... (기존 설정)
```

```
    'main', # 새로 생성한 앱 이름 추가
```

```
    # ...
```

```
]
```

Django –프로젝트, App 생성

2.View(뷰)작성(main/views.py)

mysite/views.py

간단한 "Hello, Django!" 메시지를 반환하는 함수를 작성합니다.

main/views.py

from django.http import HttpResponse

def index(request):

HTTP 요청을 받아 HTTP 응답을 반환합니다.

return HttpResponse("<h1>Hello, Django! Welcome to my first page.</h1>")

Django –프로젝트, App 생성

3.URL 매핑(URLconf)

mysite/views.py

사용자의 URL요청과 작성한 View함수를 연결하는 작업입니다.

3-1 앱 RULconf 생성 및 설정(main/urls.py)

앱폴더(main/)내부에 urls.py파일을 생성하고 View 함수를 연결합니다.

main/urls.py (새로 생성)

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

```
from . import views
```

URL 패턴 리스트

```
urlpatterns = [
```

```
    # 사용자가 'http://.../main/'으로 요청하면 views.index 함수가 실행됩니다.
```

```
    path(' ', views.index, name='index'),
```

```
]
```

Django –프로젝트, App 생성

3.URL 매핑(URLconf)

3-2 프로젝트 URLconf 설정(mysite/urls.py)

프로젝트 레벨의 urls.py에 앱의 URLconf를 연결

```
# mysite/urls.py
```

```
from django.contrib import admin
```

```
from django.urls import path, include # 'include'를 import 함
```

```
urlpatterns = [
```

```
    path('admin/', admin.site.urls),
```

```
    # 'http://.../main/'로 시작하는 모든 요청을 main 앱의 urls.py로 보냄
```

```
    path('main/', include('main.urls')),
```

```
]
```


Django –프로젝트, App 생성

4.서버 실행

프로젝트 폴더(mysite/)에서 개발 서버를 실행3-2 프로젝트 URLconf 설정(mysite/urls.py)

```
python manage.py runserver
```

서버가 성공적으로 실행되면 터미널에 다음과 같은 메시지 출력

Starting development server at http://127.0.0.1:8000/ Quit the server with CONTROL-C.

5.동작확인

웹브라우저 열고 다음 주소 접속

Http://127.0.0.1:8000/main/

Django –프로젝트, App 생성

6.요약 명령어

파일 / 명령어	역할
<code>pip install django</code>	Django 프레임워크 설치
<code>django-admin startproject</code>	Django 프로젝트 초기 폴더 생성
<code>python manage.py startapp</code>	프로젝트 내부에 기능 모듈(앱) 생성
<code>settings.py</code>	프로젝트의 전반적인 환경 설정 (앱 등록)
<code>views.py</code>	사용자의 요청을 처리하는 함수 작성 (핵심 로직)
<code>urls.py</code>	URL 경로와 View 함수 연결 (라우팅)
<code>python manage.py runserver</code>	개발용 웹 서버 실행