

# Django Study 01. Django 이해 및 프로젝트 만들기.

---

## Django

웹 프로그래밍을 할 수 있게 해준다.

Frontend (프론트엔드) : 웹 브라우저에서 동작하는 코드, 보여지는 화면의 코드를 작성한다. ( HTML, CSS, JS )

Backend(백엔드) : 데이터를 입출력하고, 계산을 하거나 서버에서 동작하는 코드를 작성한다. ( Python, Ruby, Java, PHP, JS, C# )

웹프로그래밍을 잘하기 위한 조건:

### HTTP

- 웹 사이트의 동작 방식을 이해할 수 있다.
- 각각의 무언가를 page를 요청할 때, 우리가 만든 웹 어플리케이션이 이를 해석해 해당하는 코드를 동작하도록 한다.
- 이 코드의 동작 결과는 응답으로 돌려받고, 서버가 웹 브라우저로 데이터를 보내준다.
- 웹 브라우저는 응답받은 데이터를 화면에 표시한다. 여기까지의 메커니즘을 잘 숙지해야 한다.

### 백엔드 코드

- 웹 어플리케이션이 URL을 해석해 해당하는 코드가 동작하고 응답하게 하려면, 많은 양의 백엔드 코드가 필요하다.
- 모든 화면마다 소스코드 1개 이상의 준비가 필요해 양이 방대하다.

### Framework

- 어떤 일을 할 때 자주 사용되는 기능을 미리 준비해 방대한 코드 양의 번거로움을 덜어준다.
- Micro Framework(최소한의 기능, Flask), FullStack(대부분의 기능, Django)

### 디자인 패턴

- 역할을 분담하지 않으면 개발의 설계상 문제가 발생할 수 있다.
- 디자이너, 프론트, 백엔드로 역할을 최대한 분담해 효율적으로 일을 수행할 수 있도록 한다.
- MVC : Model( 데이터베이스), View(화면-프론트), Controller(계산, 처리 - 백엔드)
- MTV : Model(데이터베이스), Template(화면-프론트), View(계산,처리 - 백엔드)

## 장고 프로젝트의 과정 순서 알아보기.

1. 파이참(vscode) 프로젝트 만들기
2. 장고 설치
3. 장고 프로젝트 만들기
4. 설정하기 (데이터베이스, S3)
5. 데이터베이스 초기화
6. 관리자 계정 만들기
7. 앱 만들기
8. 모델 설계
9. 뷰 만들기
10. 템플릿 만들기 (화면에 표시될 내용, 양식)
11. URL 만들기. (필수 기능 : Create, Read, Update, Delete)


프로젝트는 이 틀을 벗어나지 않고, 필요시에 중간중간 추가되는 부분이 있을 수 있다.

오늘 실습해볼 내용은 1번, 2번, 3번에 관한 기본적인 부분이다.

그렇다면 이제 본격적으로 Django의 세계에 빠져보자!

## 1. PyCharm 내에서 프로젝트 생성하기

활기찬 Python 커뮤니티에 대한 감사의 마음을 담아 Python 에코시스템을 지원하는 오픈소스 기여 활동으로 PyCharm Community Edition을 무상으로 제공합니다.

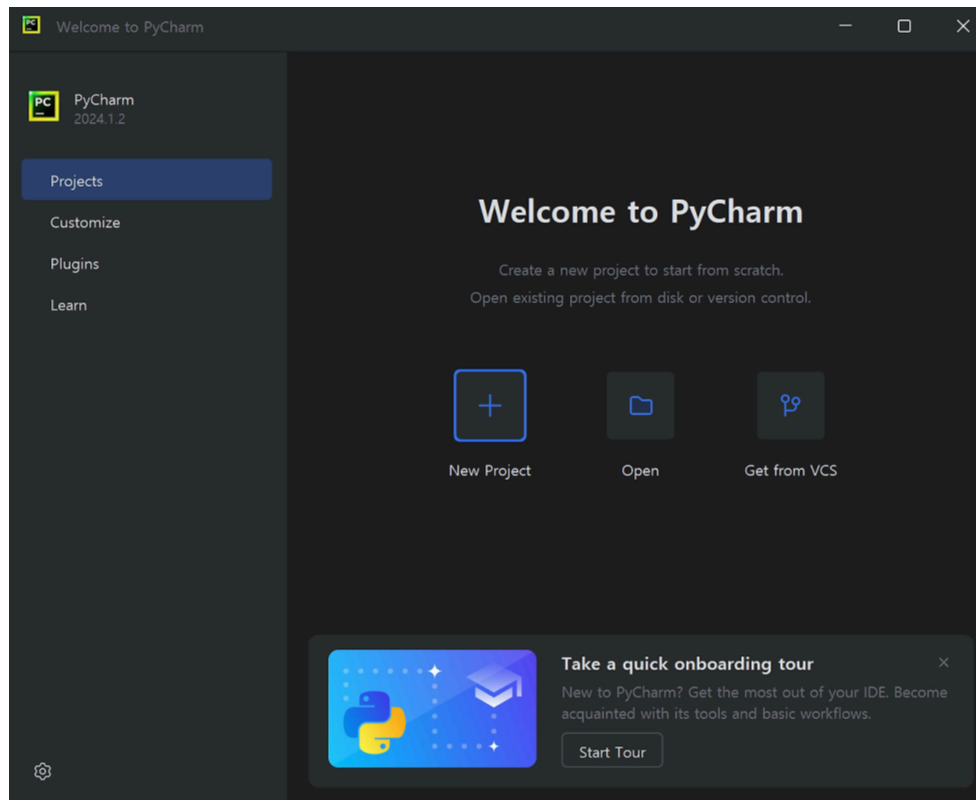
 **PyCharm Community Edition**  
순수 Python 개발용 IDE

[다운로드](#) .exe (Windows) ▼

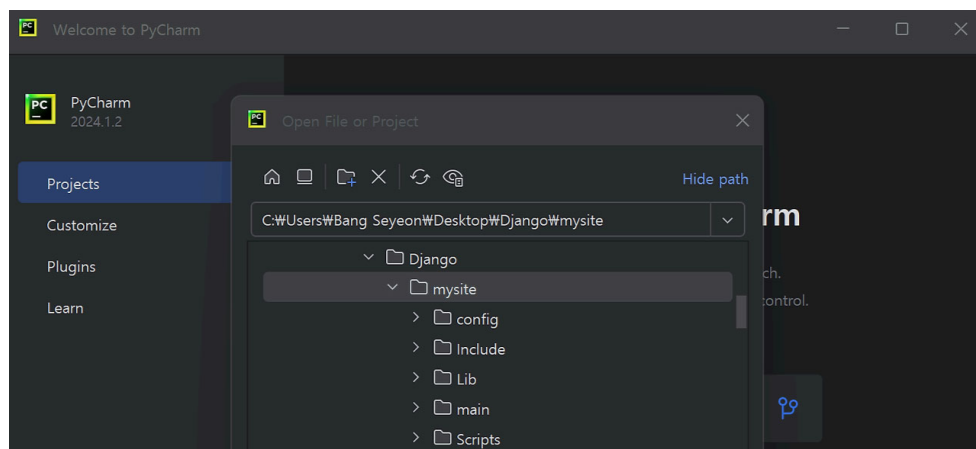
무료, 오픈 소스로 빌드됨

Feedback

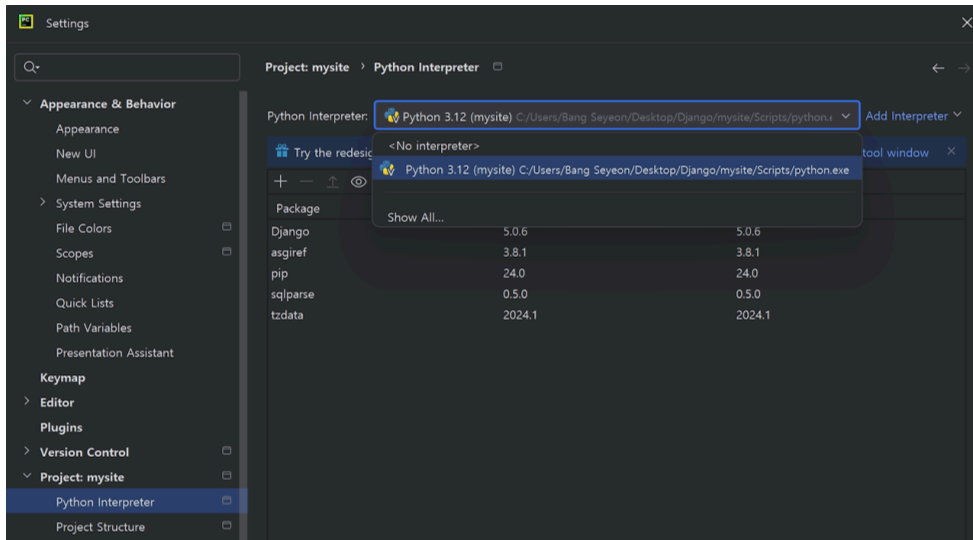
해당 이미지를 클릭하면 PyCharm 다운로드 사이트로 이동할 수 있다.



PyCharm을 설치해주었다.



사전에 실습을 진행했던 프로젝트인 mysite를 선택해주었다.



file -> Settings 에서 Project: mysite를 접근했을 때, ~/mysite/Scripts/python.exe를 선택해주는 과정은 가상 환경을 사용하려는 과정에서 필요한 사항이다.

위 사진은 이미 있던 프로젝트를 열어준 것이지만, 처음이라면 python -m venv '가상환경명'을 통해 프로젝트를 생성해줘야 한다.

##

새로운 가상환경을 설치하기

```
python -m venv mysite
```

```
django-admin startproject config .
```

mysite 가상환경에 진입하기

```
PS C:\Django\mysite> cd C:\Django\mysite\Scripts
PS C:\Django\mysite\Scripts> .\activate
```

```
cd C: mysite/Scripts/
```

```
.\wactivate
```

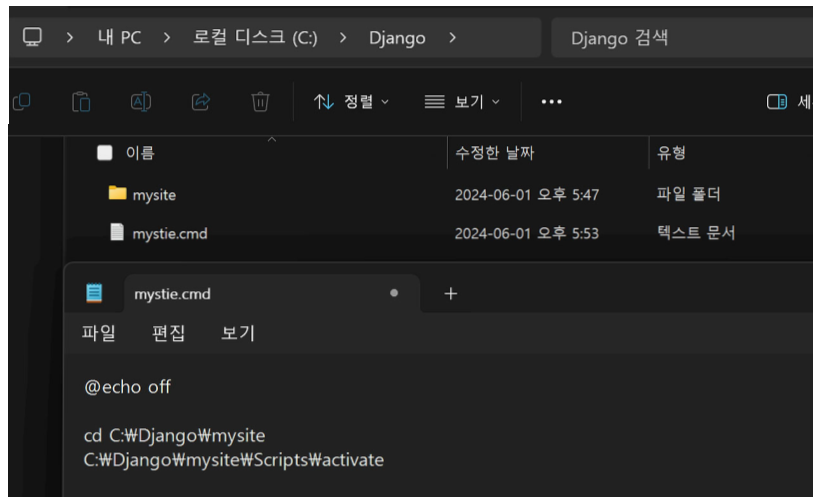
...

```
(mysite) PS D: :....
```

해당 과정이 완료되면 가상환경에 진입한 것이다.  
(mysite)는 우리가 진입한 가상환경, 즉 mysite를 나타낸다.

가상환경에서 벗어나기 위해서는 deactivate라는 명령을 실행하면 된다.  
어디서든 벗어날 수 있다.

## 더 쉽게 activate에 접속하기.



해당 내용을 입력해주고, 파일을 저장할 때 '모든 파일'을 선택해 저장한다.

제어판 -> 시스템 및 보안 -> 시스템 -> 고급시스템 설정 -> 환경변수 경로로 이동

시작 및 복구

시스템 시작, 시스템 오류 및 디버깅 정보

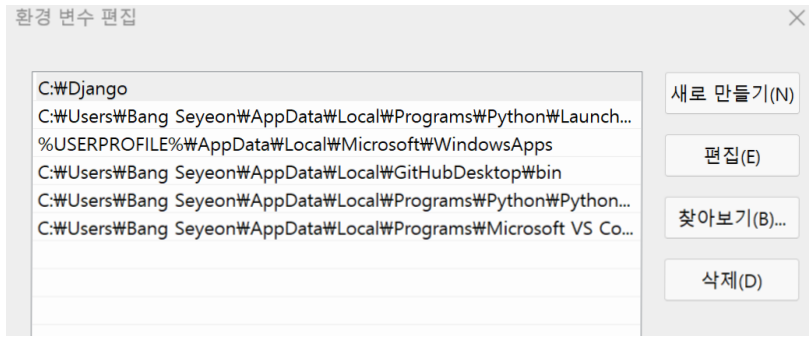
설정(T)...

환경 변수(N)...

확인

취소

적용(A)



시스템 변수(S)에 Path 경로 추가해주기.

이후 터미널에 `mysite`를 입력해준다면, `activate`가 쉽게 작동할 것이다.

## 2. 장고 설치하기.

```
pip install django
```

## 3. 장고 프로젝트 만들기.

```
System check identified no issues (0 silenced).
June 01, 2024 - 17:38:36
Django version 5.0.6, using settings 'config.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

```
(mysite) >django-admin startproject config .
```

```
(mysite) >python manage.py migrate
```

```
(mysite) >python manage.py runserver
```

이 과정을 진행하면 DB를 생성하고, 개발 서버에 구동하고 웹 사이트에 접속할 수 있게 된다.

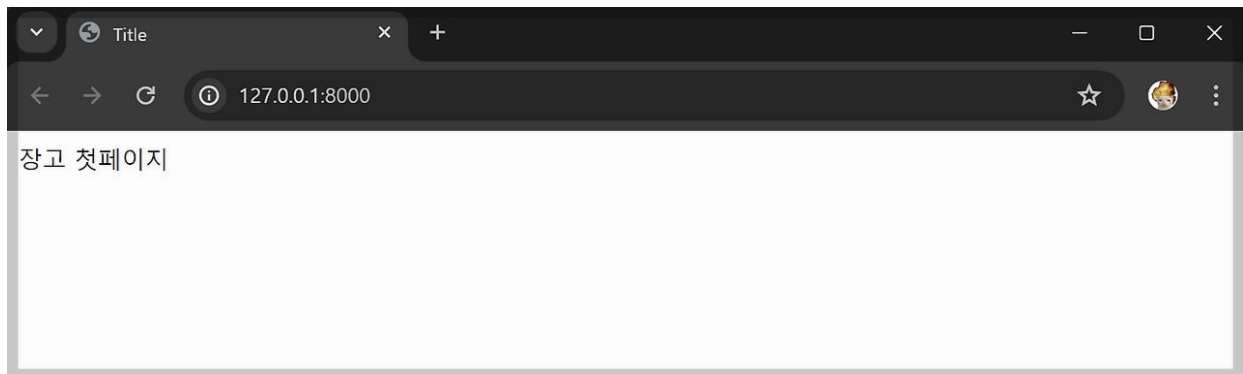
장고 서버는 `http://127.0.0.1:8000/`에서 구동되었으며 서버를 중지하기 위해 `CONTROL-c`를 입력해야 함을 알 수 있다.

이는 로컬 호스트라고 부르며, 로컬 서버 또는 개발 서버라고 칭할 수도 있다.

```
https://127.0.0.1:8000/  
https://localhost:8000/
```

이 둘 중 어떤 주소를 입력해도 서버에 접속할 수 있다.

이 주소는 자신만 입장할 수 있으며, 아직 다른 상대방이 이 사이트에 접속할 수는 없다.



스터디 참고 :

장고를 공부할 때 필요한 기본 지식 - Django Basic : [https://youtu.be/Nv\\_9uZ4ld9U](https://youtu.be/Nv_9uZ4ld9U)

점프 투 장고 : <https://wikidocs.net/book/4223>