# 2.1 Getting started with Django

| | |
|---|---|
| Courses | Python |
| Done | □ |

## 1. Why Django is so popular among web developers

Django is popular because it allows developers to build secure, scalable web applications quickly without reinventing common features like authentication or databases. It follows a “batteries-included” approach, which means many tools are built in and ready to use. This makes Django especially appealing for teams that want to move fast (can create quick prototypes) while following best practices.

## 2. Five large companies that use Django

- **Instagram**

  Product/Service: Social media platform

  Use of Django: Handles backend logic, APIs, and scaling for millions of users.

- **Pinterest**

  Product/Service: Visual discovery and bookmarking

  Use of Django: Supports backend services and content management at scale.

- **Mozilla**

  Product/Service: Web browsers and developer tools

  Use of Django: Powers internal tools and public-facing web services.

- **Spotify**

  Product/Service: Music and podcast streaming

  Use of Django: Used for backend services and data-driven features.

- **Dropbox**

  Product/Service: File storage and collaboration

Use of Django: Supports backend systems and internal tools.

## 3. When would you use Django?

### You need to develop a web application with multiple users

Yes. Django is a great choice because it includes built-in authentication, user management, and security features that are used in  multi-user applications.

### You need fast deployment and the ability to make changes as you proceed

Yes. Django’s structure and built-in tools allow developers to prototype quickly and make iterative changes without a lot of setup overhead due to what is included when using Django.

### You need to build a very basic application with no database or file operations

Probably not. Django may be overkill for a very simple app, and a lighter framework or even plain Python might be more appropriate.

### You want to build an application from scratch and want a lot of control

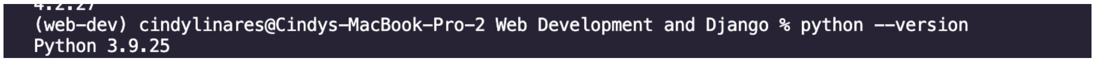
Not ideal. Django is opinionated and provides a lot out of the box, which trades some low-level control for speed and consistency.

### You’re starting a big project and are worried about getting stuck

Yes. Django has excellent documentation and a large community, making it easier to find help, tutorials, and solutions when problems come up.

## 4. Python & virtual environment verification

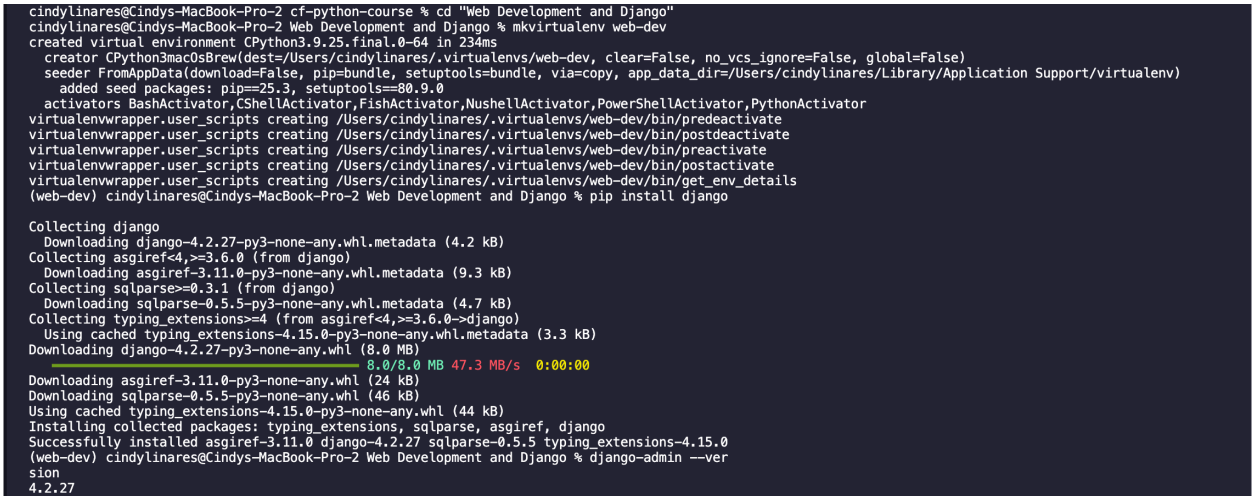
- Python was installed and verified using the `python3 --version` command.



```
4.2.27
(web-dev) cindylinares@Cindys-MacBook-Pro-2 Web Development and Django % python --version
Python 3.9.25
```

- A virtual environment named `web-dev` was created and activated successfully.

  - *** I DID NOT USE “achievement2-practice” i used web-dev per following the reading.***



```
cindylinares@Cindys-MacBook-Pro-2 cf-python-course % cd "Web Development and Django"
cindylinares@Cindys-MacBook-Pro-2 Web Development and Django % mkvirtualenv web-dev
created virtual environment CPython3.9.25.final.0-64 in 234ms
  creator CPython3macOsBrew(dest=/Users/cindylinares/.virtualenvs/web-dev, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, via=copy, app_data_dir=/Users/cindylinares/Library/Application Support/virtualenv)
    added seed packages: pip==25.3, setuptools==80.9.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /Users/cindylinares/.virtualenvs/web-dev/bin/predeactivate
virtualenvwrapper.user_scripts creating /Users/cindylinares/.virtualenvs/web-dev/bin/postdeactivate
virtualenvwrapper.user_scripts creating /Users/cindylinares/.virtualenvs/web-dev/bin/preactivate
virtualenvwrapper.user_scripts creating /Users/cindylinares/.virtualenvs/web-dev/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/cindylinares/.virtualenvs/web-dev/bin/get_env_details
(web-dev) cindylinares@Cindys-MacBook-Pro-2 Web Development and Django % pip install django

Collecting django
  Downloading django-4.2.27-py3-none-any.whl.metadata (4.2 kB)
Collecting asgiref<4,>=3.6.0 (from django)
  Downloading asgiref-3.11.0-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from django)
  Downloading sqlparse-0.5.5-py3-none-any.whl.metadata (4.7 kB)
Collecting typing_extensions>=4 (from asgiref<4,>=3.6.0->django)
  Using cached typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
Downloading django-4.2.27-py3-none-any.whl (8.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.0/8.0 MB 47.3 MB/s  0:00:00
Downloading asgiref-3.11.0-py3-none-any.whl (24 kB)
Downloading sqlparse-0.5.5-py3-none-any.whl (46 kB)
Using cached typing_extensions-4.15.0-py3-none-any.whl (44 kB)
Installing collected packages: typing_extensions, sqlparse, asgiref, django
Successfully installed asgiref-3.11.0 django-4.2.27 sqlparse-0.5.5 typing_extensions-4.15.0
(web-dev) cindylinares@Cindys-MacBook-Pro-2 Web Development and Django % django-admin --ver
sion
4.2.27
```

- Django was installed inside the activated virtual environment and verified by checking its version.