

Why Django is Popular Among Web Developers

Django is popular because it allows developers to build web applications quickly and with less code. It provides a lot of built-in tools for common web development tasks and adheres to the "Don't Repeat Yourself" (DRY) principle, which keeps the code base clean and maintainable. Additionally, its robust security features are highly valued in developing secure websites.




Large Companies Using Django

- Instagram: Social media platform - Instagram uses Django to handle massive user data and traffic because Django efficiently manages user authentication and data models.
- Spotify: Music streaming service - Spotify employs Django to handle the backend of its browse and search services, utilizing its rapid development capabilities.
- YouTube: Video sharing service - YouTube uses Django for certain features like video processing, viewing, and management due to Django's ability to scale with high traffic.
- Dropbox: Cloud storage service - Dropbox uses Django for handling the backend of their web application, managing file storage, and user authentication.
- Mozilla: Software community - Known for the Firefox browser, Mozilla uses Django for many of its sites to manage user data and site content effectively.

Scenarios for Using Django

1. Developing a web application with multiple users: Use Django because it has built-in support for handling user authentication, sessions, and user roles, which are essential for a multi-user application.
2. Need fast deployment and the ability to make changes: Use Django as it offers rapid development with its extensive libraries and templating engine, making it ideal for projects that need to evolve over time.
3. Building a very basic application without database or file operations: Do not use Django. It might be overkill for such simple needs. A lightweight framework or even a simple static site might be more appropriate.
4. Building an application from scratch with full control: Use Django if you are comfortable with its structure; it's very customizable and extensible. However, if you need control over every aspect of the system, a micro-framework might offer more flexibility.
5. Starting a big project with potential need for support: Use Django because it has a large and active community, which can be a great resource when you encounter challenges. The extensive documentation and community support can help you troubleshoot and learn throughout the project.

```
activators BashActivator, BatchActivator, FishActivator, NushellActivator, PowerShellActivator, PythonActivator
PS C:\python> workon achievement2-practice
PS C:\python> cd .\achievement2-practice\
PS C:\python\achievement2-practice> █
```

```
PS C:\python\achievement2-practice> pip install django
● Collecting django
  Downloading Django-5.0.6-py3-none-any.whl.metadata (4.1 kB)
  Collecting asgiref<4,>=3.7.0 (from django)
    Downloading asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
  Collecting sqlparse>=0.3.1 (from django)
    Downloading sqlparse-0.5.0-py3-none-any.whl.metadata (3.9 kB)
  Collecting tzdata (from django)
    Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
  Downloading Django-5.0.6-py3-none-any.whl (8.2 MB)
     8.2/8.2 MB 18.7 MB/s eta 0:00:00
  Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
  Downloading sqlparse-0.5.0-py3-none-any.whl (43 kB)
     44.0/44.0 kB ? eta 0:00:00
  Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
     345.4/345.4 kB 20.9 MB/s eta 0:00:00
  Installing collected packages: tzdata, sqlparse, asgiref, django
  Successfully installed asgiref-3.8.1 django-5.0.6 sqlparse-0.5.0 tzdata-2024.1

[notice] A new release of pip is available: 24.0 -> 24.1
[notice] To update, run: python.exe -m pip install --upgrade pip
○ PS C:\python\achievement2-practice> █
```