

**1. Write two to three sentences on why Django is so popular among web developers.**

Developers like to use Django because it uses the DRY principles to avoid redundant and repetitive code. It also assures development is fast and easy, as well as helps the application run faster.

**2. After some research, list five large companies that use Django. Specify what the company's product or service is and what they use Django for.**

Instagram: A video and photo sharing social media platform, uses Django for user interactions, data management, and image processing.

Mozilla: A web browser, uses Django to leverage its flexibility to manage complex content and user interactions

Pinterest: An engine for finding ideas, uses Django for its scalability and performance capabilities for user interactions, image processing, and recommendation algorithms.

Disqus: A commenting platform for websites, uses Django to handle high traffic and real time updates

Spotify: A music streaming platform, uses Django for its flexibility and rapid development capabilities.

**3. For each of the following scenarios, explain if you would use Django (and why or why not):**

**- You need to develop a web application with multiple users**

Yes because it has built in authentication and authorization. It can handle many users and sessions at the same time.

**- You need fast deployment and the ability to make changes as you proceed.**

Yes because its built in server allows for quick testing and prototyping, as well as you can easily make changes as the project evolves

- **You need to build a very basic application, which doesn't require any database access or file operations**

It may be overkill to use Django because it has many built-in features for database access that would not be necessary. A simpler framework like flask would be more reasonable for this application

- **You want to build an application from scratch and want a lot of control over how it works.**

Django could be a good option because it provides a solid foundation that allows customization. But if you want absolute control over everything in the project, a smaller framework or building it from scratch may be a better option because using Django means you have to follow the Django rule.

- **You're about to start working on a big project and are afraid of getting stuck and needing additional support.**

Yes because Django is made to handle complex applications with large code bases. It has a large active community so there is strong support for building and maintaining large-scale projects.

```
(achievement2-practice) C:\Users\dkoun> python --version
Python 3.8.7
```

```
(web-dev) C:\Users\dkoun> mkvirtualenv achievement2-practice
created virtual environment CPython3.8.7.final.0-64 in 272ms
creator CPython3Windows(dest=C:\Users\dkoun\Envs\achievement2-practice, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users\dkoun\AppData\Local\pypa\virtualenv)
added seed packages: pip==24.1, setuptools==70.1.0, wheel==0.43.0
activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

(achievement2-practice) C:\Users\dkoun> py -m pip install django
Collecting django
  Using cached Django-4.2.14-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.6.0 (from django)
  Using cached asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from django)
  Using cached sqlparse-0.5.1-py3-none-any.whl.metadata (3.9 kB)
Collecting backports.zoneinfo (from django)
  Using cached backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl.metadata (4.7 kB)
Collecting tzdata (from django)
  Using cached tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting typing-extensions>=4 (from asgiref<4,>=3.6.0->django)
  Using cached typing_extensions-4.12.2-py3-none-any.whl.metadata (3.0 kB)
Using cached Django-4.2.14-py3-none-any.whl (8.0 MB)
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.1-py3-none-any.whl (44 kB)
Using cached backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl (38 kB)
Using cached tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Using cached typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: tzdata, typing-extensions, sqlparse, backports.zoneinfo, asgiref, django
Successfully installed asgiref-3.8.1 backports.zoneinfo-0.2.1 django-4.2.14 sqlparse-0.5.1 typing-extensions-4.12.2 tzdata-2024.1

[notice] A new release of pip is available: 24.1 -> 24.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip

(achievement2-practice) C:\Users\dkoun> django-admin --version
4.2.14

(achievement2-practice) C:\Users\dkoun>
```