

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

Answer: Django is a popular framework amongst web developers because it “removes the hassle of web development” and lets the developer focus on their project’s business logic. Django is also popular because it allows for a faster web development process, secure deployment and easier scalability.

2. List five large companies that use Django. Specify what the company’s product or service is and what they use Django for.

Answer: Here are five large companies that use Django and what they use it for:

- Spotify is a music platform where you can listen to millions of available songs and podcasts from various artists across the world. They use Django because of its fast backend processes and its machine learning possibilities. I am sure Django easy scalability also plays a big part in this decision since Spotify has a large community of users.
- Instagram is a social media where users can post pictures and short videos that they want to share with their followers. Instagram uses Django because of its scalability benefits. Instagram has over 2 billion users and Django allows them to handle this with ease, as well as allowing Instagram developers to focus on UI/UX rather than the technology behind their features.
- Pinterest is a social media app focused on pictures for creative moodboards. They are known for using Django due to its machine learning possibilities that allows users to get personalized recommendations of posts (much like Instagram).
- Dropbox is a cloud storage platform where you can keep your files safely stored. They use Django for their backend and client desktop software. Django lets them synchronize your account across different devices, add user history and file sharing options.
- Youtube - Youtube is a video sharing platform with an ever-growing large database of users. Youtube is said to have chosen Django not only because of their large user community but also due to Django’s ability to implement new features in a fast manner that allows Youtube to keep improving their platform consistently.

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

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

I would use Django for this due to its scalability. Some of the world's biggest apps are proof that this framework can handle an absurd amount of users accessing their platforms everyday.

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

I would also use Django for this scenario since Django is known for being able to implement new features in a quick and easy manner.

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

I would not use Django for this scenario. Django is recommended for more complex applications and a basic application without database access would not require such an intensive, heavy framework.

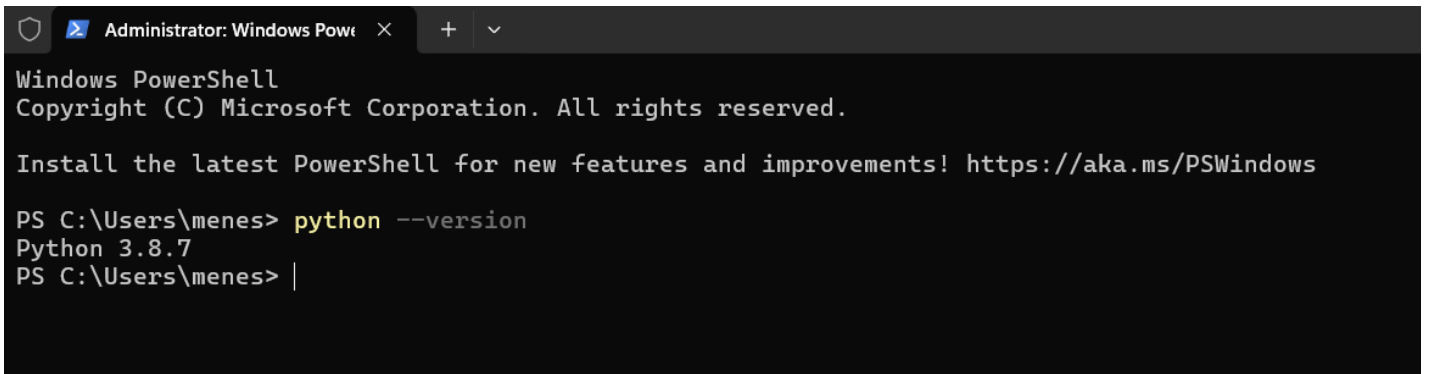
d. You want to build an application from scratch and want a lot of control over how it works.

I would not use Django in this scenario since this framework requires the developer to follow a strict set of rules and therefore removes some of the control of how it works.

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

I would definitely consider using Django for this scenario since this is a big project that will most likely require a good, scalable framework. Django also has a large community online which will be able to provide me with documentation and support in case I encounter any issues along the way.

4. Download and install Python; and check Python's current version:

A screenshot of a Windows PowerShell terminal window. The title bar at the top reads "Administrator: Windows PowerShell" with a shield icon on the left and window control buttons on the right. The terminal content shows the standard PowerShell startup text: "Windows PowerShell" and "Copyright (C) Microsoft Corporation. All rights reserved." followed by a message to install the latest PowerShell. The user then enters the command "python --version" at the prompt "PS C:\Users\menes>". The output is "Python 3.8.7". The prompt "PS C:\Users\menes> |" is visible on the next line, indicating the command execution is complete.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\menes> python --version
Python 3.8.7
PS C:\Users\menes> |
```

5. Set up and create a virtual environment and name it **achievement2-practice**.

Then activate the virtual environment. Install Django and verify the installation by checking the version.

```
Command Prompt
(web-dev) C:\Users\menes>mkvirtualenv achievement2-practice
created virtual environment CPython3.8.7.final.0-64 in 1552ms
creator CPython3Windows(dest=C:\Users\menes\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\menes\AppData\Local\pypa\virtualenv)
added seed packages: pip==24.3.1, setuptools==75.3.0, wheel==0.45.1
activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

(achievement2-practice) C:\Users\menes>activate.bat

(achievement2-practice) C:\Users\menes>if defined _OLD_VIRTUAL_PYTHONPATH (set "PYTHONPATH=" ) else (set "_OLD_VIRTUAL_PYTHONPATH=" )

(achievement2-practice) C:\Users\menes>py -m pip install Django
Collecting Django
  Using cached Django-4.2.19-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-0.5.3-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-2025.1-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-4.2.19-py3-none-any.whl (8.0 MB)
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.3-py3-none-any.whl (44 kB)
Using cached backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl (38 kB)
Using cached tzdata-2025.1-py2.py3-none-any.whl (346 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 Django-4.2.19 asgiref-3.8.1 backports.zoneinfo-0.2.1 sqlparse-0.5.3 typing_extensions-4.12.2 tzdata-2025.1

[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(achievement2-practice) C:\Users\menes>django-admin --version
4.2.19

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