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

Django is popular among web developers due to its ease of use, versatility, and the wide range of functionality it provides out of the box. Its powerful object-relational mapper and built-in admin interface make it easy to build and manage complex web applications, while its template system allows for flexible and responsive user interfaces. Additionally, Django's community is highly active, which means that developers can easily access documentation, tutorials, and support.

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

1. Instagram: Django framework helps Instagram process large amounts of data and perform user interaction fast every minute of the day.
2. Youtube: Django ensures that new features and updates are added in a short period of time and eliminates the risk of making massive flaws in implementation.
3. Spotify: Spotify has a large library with a huge amount of data accessible everywhere on any device, and to handle this data it uses Django.
4. Pinterest: Pinterest uses Django to ensure the best performance for its heavy-load users of 250 million people monthly.
5. Mozilla: it uses Django to manage high traffic and improve efficiency from millions of searches daily.

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

⍰ **You need to develop web application with multiple users.**
I would use Django in this case as it would help deploy the app quickly and scale it up in the future to improve user experience.

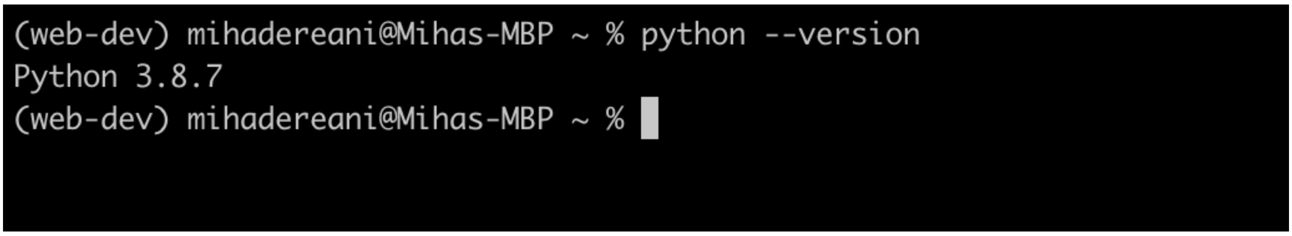
⍰ You need fast deployment and the ability to make changes as you proceed.
One of the main features of Django is its fast deployment speed and easy, flawless scalability, so I would use it.

⍰ You need to build a very basic application that doesn’t require any database access or file operations.
I wouldn’t use Django here because its strength is seen when most of its features are used. In small applications like this, though, it will, in fact, be an unnecessary heavy burden for the server.

⍰ You want to build an application from scratch and want a lot of control over how it works.
Again I wouldn’t use Django here as Django is a batteries-included framework; I will definitely lose some control over how I can build it.

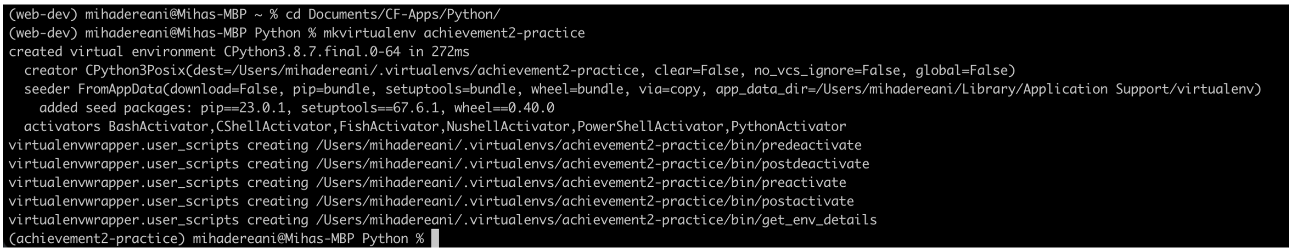
⍰ You are about to start working on a big project and are afraid of getting stuck and needing additional support.
I would use Django here. It would help with its built-in libraries and tools if I got stuck, and I could always scale down without/less flaws.

Download and install Python



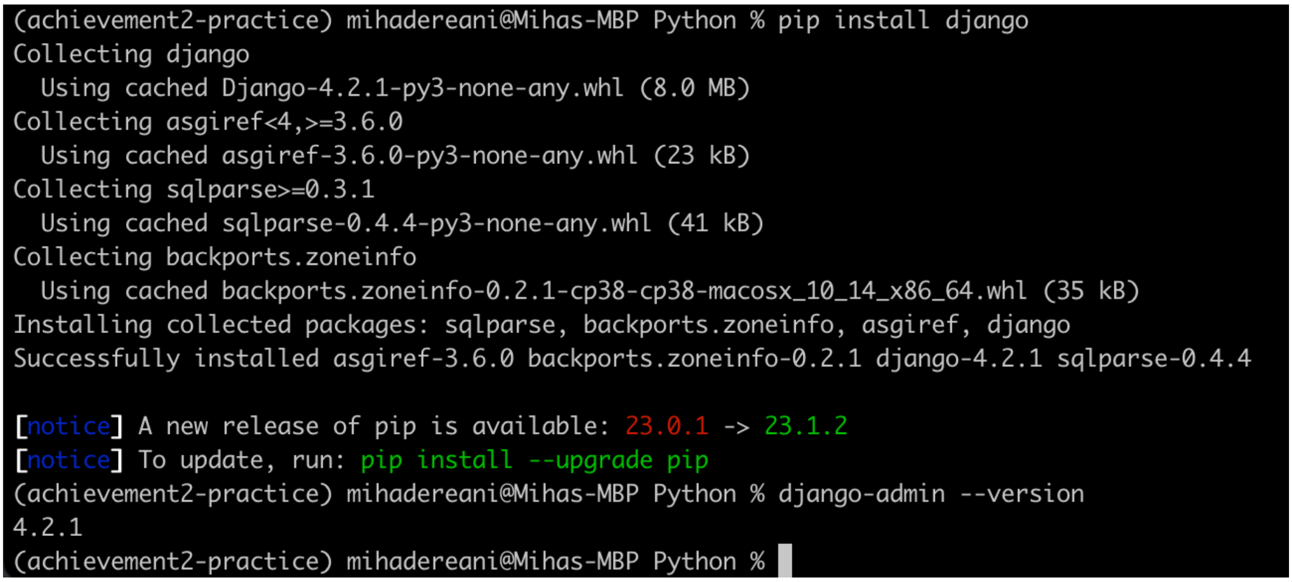
```
(web-dev) mihadereani@Mihas-MBP ~ % python --version
Python 3.8.7
(web-dev) mihadereani@Mihas-MBP ~ % 
```

Create a virtual environment and name it



```
(web-dev) mihadereani@Mihas-MBP ~ % cd Documents/CF-Apps/Python/
(web-dev) mihadereani@Mihas-MBP Python % mkvirtualenv achievement2-practice
created virtual environment CPython3.8.7.final.0-64 in 272ms
  creator CPython3Posix(dest=/Users/mihadereani/.virtualenvs/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=/Users/mihadereani/Library/Application Support/virtualenv)
    added seed packages: pip==23.0.1, setuptools==67.6.1, wheel==0.40.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /Users/mihadereani/.virtualenvs/achievement2-practice/bin/predeactivate
virtualenvwrapper.user_scripts creating /Users/mihadereani/.virtualenvs/achievement2-practice/bin/postdeactivate
virtualenvwrapper.user_scripts creating /Users/mihadereani/.virtualenvs/achievement2-practice/bin/preactivate
virtualenvwrapper.user_scripts creating /Users/mihadereani/.virtualenvs/achievement2-practice/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/mihadereani/.virtualenvs/achievement2-practice/bin/get_env_details
(achievement2-practice) mihadereani@Mihas-MBP Python % 
```

Install Django and verify the installation by checking the version.



```
(achievement2-practice) mihadereani@Mihas-MBP Python % pip install django
Collecting django
  Using cached Django-4.2.1-py3-none-any.whl (8.0 MB)
Collecting asgiref<4,>=3.6.0
  Using cached asgiref-3.6.0-py3-none-any.whl (23 kB)
Collecting sqlparse>=0.3.1
  Using cached sqlparse-0.4.4-py3-none-any.whl (41 kB)
Collecting backports.zoneinfo
  Using cached backports.zoneinfo-0.2.1-cp38-cp38-macosx_10_14_x86_64.whl (35 kB)
Installing collected packages: sqlparse, backports.zoneinfo, asgiref, django
Successfully installed asgiref-3.6.0 backports.zoneinfo-0.2.1 django-4.2.1 sqlparse-0.4.4

[notice] A new release of pip is available: 23.0.1 -> 23.1.2
[notice] To update, run: pip install --upgrade pip
(achievement2-practice) mihadereani@Mihas-MBP Python % django-admin --version
4.2.1
(achievement2-practice) mihadereani@Mihas-MBP Python %
```