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

   Django is a *batteries-included* framework, meaning it “arrives” with most of the essentials for effective development—including important additional plugins for features such as user-friendly admin panels and authentication. Django’s fast development capabilities help save developer time, and it increases the overall cost-effectiveness of the project. It helps developers avoid common security issues that could otherwise incur additional costs later. It’s also scalable, which makes it very simple to add new hardware and components.

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: Use Django for managing user data and interactions.
   - Spotify: For backend and data analysis. Django helps manage user data, playlists, music preference, and data handling.
   - Eventbrite: Helps manage the extensive event listings and ticketing processes efficiently.
   - Pinterest: Django is used to process large amounts of and handle thousands of user requests simultaneously. Django helps Pinterest to scale and perform well.
   - Dropbox: Django is used for file handling, user authentication and database management.

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 of its built-in user authentication and admin interface is well-suited for multiuser applications.

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

Yes, Django supports fast development and deployment through its modular structure. It comes with built in features to help with to help with fast development, and Django's battery included philosophy means developers can make changes quickly.

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

  No, for such applications, frameworks like flask would be a good fit, since they have minimal set up and are easier to use.

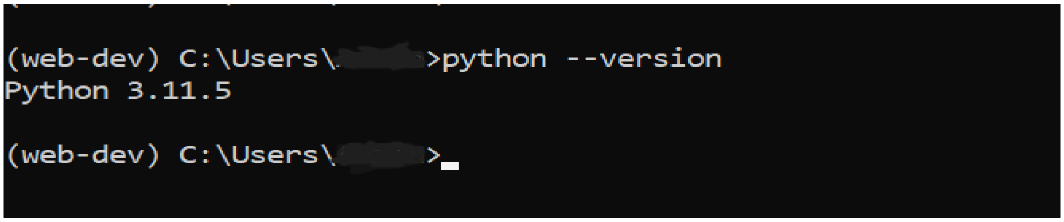
- You want to build an application from scratch and want a lot of control over how it works.

  No, using Django is highly structured and does things a certain way. Certain steps have to be taken in an exact manner, so there is less control for the developer.

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

  Yes, Django is a widely used framework with extensive documentation, and has a large community. So, it makes it easier to find solutions to problems and get support from other developers.

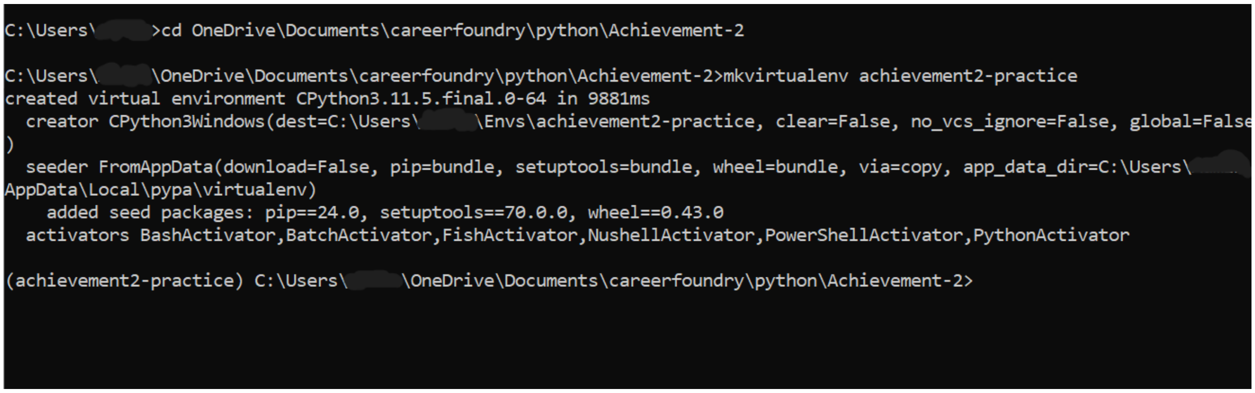
Python Version:



```
(web-dev) C:\Users\[redacted]>python --version
Python 3.11.5

(web-dev) C:\Users\[redacted]>_
```

New Virutal Environment:

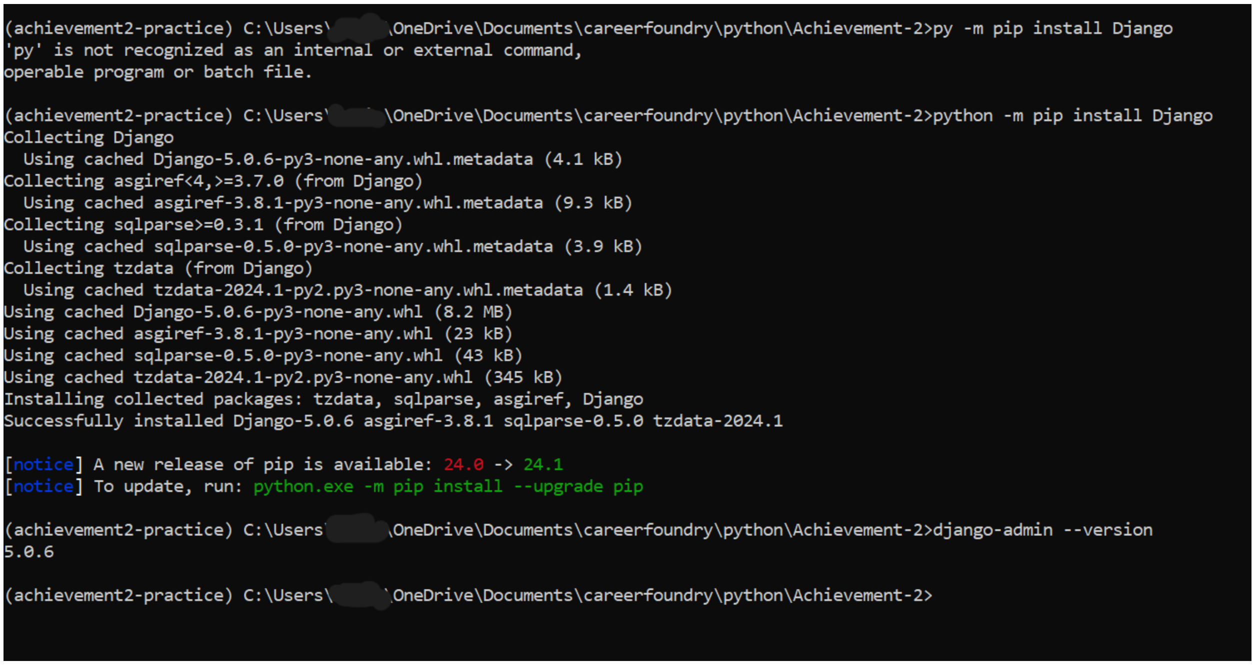


```
C:\Users\[redacted]>cd OneDrive\Documents\careerfoundry\python\Achievement-2

C:\Users\[redacted]\OneDrive\Documents\careerfoundry\python\Achievement-2>mkvirtualenv achievement2-practice
created virtual environment CPython3.11.5.final.0-64 in 9881ms
  creator CPython3Windows(dest=C:\Users\[redacted]\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\[redacted]
AppData\Local\pypa\virtualenv)
    added seed packages: pip==24.0, setuptools==70.0.0, wheel==0.43.0
  activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

(achievement2-practice) C:\Users\[redacted]\OneDrive\Documents\careerfoundry\python\Achievement-2>
```

Django Installation:



```
(achievement2-practice) C:\Users\[redacted]\OneDrive\Documents\careerfoundry\python\Achievement-2>py -m pip install Django
'py' is not recognized as an internal or external command,
operable program or batch file.

(achievement2-practice) C:\Users\[redacted]\OneDrive\Documents\careerfoundry\python\Achievement-2>python -m pip install Django
Collecting Django
  Using cached Django-5.0.6-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.7.0 (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-0.5.0-py3-none-any.whl.metadata (3.9 kB)
Collecting tzdata (from Django)
  Using cached tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Using cached Django-5.0.6-py3-none-any.whl (8.2 MB)
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.0-py3-none-any.whl (43 kB)
Using cached tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Installing collected packages: tzdata, sqlparse, asgiref, Django
Successfully installed Django-5.0.6 asgiref-3.8.1 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

(achievement2-practice) C:\Users[redacted]\OneDrive\Documents\careerfoundry\python\Achievement-2>django-admin --version
5.0.6

(achievement2-practice) C:\Users\[redacted]\OneDrive\Documents\careerfoundry\python\Achievement-2>
```