

Exercise 2.1: Task Questions

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

Developers like to use Django as it works fast and is secure and scalable. It also follows DRY principles, so it helps to optimize code structure and avoid repetitive code.

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

YouTube, a video-sharing platform, uses Django to ensure features and upgrades are implemented as fast as possible.

Spotify, an audio-streaming platform, uses Python in its backend combined with Django to ensure optimal functionality.

Instagram, a video and photo-sharing platform, uses Django to help the app's features function properly in the midst of the large amount of data and user interactions it has to process.

Disqus, a network-based comments system used in blogging communities, uses Django's scalability and wide variety of ready-to-implement solutions to meet the needs of its users.

Dropbox, a cloud storage service, uses Django for synchronizing data, enabling sharing, and allowing storage of large files.

Source: <https://careerkarma.com/blog/companies-that-use-django/>

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 can handle multiple users and sessions at the same time and has built-in authentication and authorization.

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

Yes, because it can handle rapid-paced development and allows making changes on the fly.

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

Given Django has many built-in features that are useful for database access and file operations, as well as many others, it would be overkill to use something like Django. Flask would be a good alternative as it is more lightweight.

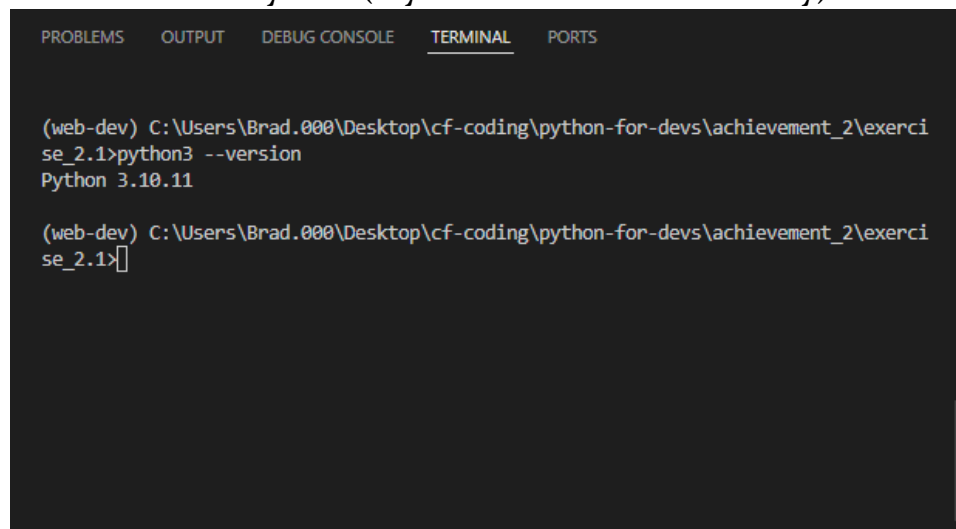
- **You want to build an application from scratch and want a lot of control over how it works.**

Given the “Django Way”, it can be difficult to decide how things in the app will work if you want it some features to work outside the Django norm. The app can still be made from scratch, but it has to follow Django rules and structures. For this reason, it depends on *how much* control you want. If you want total control of app customization, middleware and other custom systems can be added to the app to accommodate. If you don't want to have to install more frameworks or templates into the app, Django would not be the way to go.

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

Django is open source and has a large and active community, so it would be feasible to reach out to other devs whenever you need help with something in a project.

4. Download and install Python (if you haven't done so already).



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

(web-dev) C:\Users\Brad.000\Desktop\cf-coding\python-for-devs\achievement_2\exercise_2.1>python3 --version
Python 3.10.11

(web-dev) C:\Users\Brad.000\Desktop\cf-coding\python-for-devs\achievement_2\exercise_2.1>
```

5. If you haven't done so already, set up and create a virtual environment and name it achievement2-practice.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Version 10.0.19045.4529]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Brad.000\Desktop\cf-coding>cd ../../

C:\Users\Brad.000>mkvirtualenv achievement2-practice
created virtual environment CPython3.8.7.final.0-64 in 20187ms
creator CPython3Windows(dest=C:\Users\Brad.000\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\Brad.000\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\Brad.000>
(achievement2-practice) C:\Users\Brad.000>
```

6. Install Django and verify the installation by checking the version.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(achievement2-practice) C:\Users\Brad.000\Desktop\cf-coding\python-for-devs\achievement_2\exercise_2
.1>py -m pip install Django
Collecting Django
  Using cached Django-4.2.13-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-0.5.0-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-4.2.13-py3-none-any.whl (8.0 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 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 Django-4.2.13 asgiref-3.8.1 backports.zoneinfo-0.2.1 sqlparse-0.5.0 typing-extensions-4.12.2 tzdata-2024.1

(achievement2-practice) C:\Users\Brad.000\Desktop\cf-coding\python-for-devs\achievement_2\exercise_2
.1>django-admin --version
4.2.13

(achievement2-practice) C:\Users\Brad.000\Desktop\cf-coding\python-for-devs\achievement_2\exercise_2
.1>
```