

# Exercise 2.1

## 1. Why is Django so popular among web developers?

Firstly Django is a “batteries-included” framework meaning it contains a comprehensive toolkit right out of the box. This allows developers to spend more time on development and less time juggling external dependencies. Django also has impressive scalability, letting it handle both personal projects as well as large scale applications.

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

- **Youtube:** Youtube is the world’s most popular video-sharing platform, and uses Django to keep up with its rapid audience growth and push updates quickly and efficiently.
- **Mozilla:** Mozilla is the world’s second most popular web browser, and uses Django to keep the browser running well even with the incredibly high traffic the browser experiences daily.
- **Boeing:** A manufacturing company specializing in aviation (Airplanes, Rockets, Helicopters, etc.). Being a direct link to the US Government and military, Boeing appreciates the built in security and authentication functions Django provides along with its scalability.
- **Spotify:** Spotify is an incredibly popular music streaming platform, and uses Django to keep its backend fast even with heavy server demands.
- **DropBox:** DropBox is a cloud-based storage platform for documents, pictures, videos, and more. The platform was always built using python, but they added Django for its versatility in creating and maintaining new features for the platform.

**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, as Django offers built in user authentication and allows multiple concurrent sessions at the same time.
- **You need fast deployment and the ability to make changes as you proceed.**  
Yes, because Django is incredibly fast and its scalability will allow for growth and change without issue.
- **You need to build a very basic application, which doesn't require any database access or file operations.**  
Probably not. You can absolutely make small applications with Django, but generally it would be a bit overkill to install and use an entire framework if you don't need most of its features.
- **You want to build an application from scratch and want a lot of control over how it works.**  
I wouldn't use Django for a project like this. Django has the majority of its features and functions built in, and as such allows for a lot less flexibility if you are building something completely from scratch.
- **You're about to start working on a big project and are afraid of getting stuck and needing additional support.**  
Yes, Django has a large and active community and it would have a lot of options for getting help if I ran into problems while working on a project.

## 4. Python Version Check

```
(achievement2-practice) C:\Users\hoore\Documents\CareerFoundry\Python\CF-Python\Achievement 2\Exercise 2.1>python --version
Python 3.8.7

(achievement2-practice) C:\Users\hoore\Documents\CareerFoundry\Python\CF-Python\Achievement 2\Exercise 2.1>_
```

## 5. Virtual Environment Test

```
(web-dev) C:\Users\hoore\Documents\CareerFoundry\Python\CF-Python\Achievement 2\Exercise 2.1>mkvirtualenv achievement2-practice
created virtual environment CPython3.8.7.final.0-64 in 5426ms
creator CPython3Windows(dest=C:\Users\hoore\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\hoore\AppData\Local\pypa\virtualenv)
added seed packages: pip==24.1.2, setuptools==70.1.1, wheel==0.43.0
activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

(achievement2-practice) C:\Users\hoore\Documents\CareerFoundry\Python\CF-Python\Achievement 2\Exercise 2.1>
```

## 6. Django Install

```
(web-dev) C:\Users\hoore>py -m pip install Django
Collecting Django
  Downloading Django-4.2.14-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.6.0 (from Django)
  Downloading asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from Django)
  Downloading sqlparse-0.5.1-py3-none-any.whl.metadata (3.9 kB)
Collecting backports.zoneinfo (from Django)
  Downloading backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl.metadata (4.7 kB)
Collecting tzdata (from Django)
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting typing_extensions>=4 (from asgiref<4,>=3.6.0->Django)
  Downloading typing_extensions-4.12.2-py3-none-any.whl.metadata (3.0 kB)
Downloading Django-4.2.14-py3-none-any.whl (8.0 MB)
----- 8.0/8.0 MB 11.4 MB/s eta 0:00:00
Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
Downloading sqlparse-0.5.1-py3-none-any.whl (44 kB)
----- 44.2/44.2 kB 1.3 MB/s eta 0:00:00
Downloading backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl (38 kB)
Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
----- 345.4/345.4 kB 20.0 MB/s eta 0:00:00
Downloading 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-4.2.14 asgiref-3.8.1 backports.zoneinfo-0.2.1 sqlparse-0.5.1 typing-extensions-4.12.2 tzdata-2024.1

[notice] A new release of pip is available: 24.0 -> 24.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip

(web-dev) C:\Users\hoore>django-admin --version
4.2.14
```