

Exercise 2.1

Questions & answers

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

Answer: Django is popular among web developers for many reasons. First, Django arrives with most of the essentials for effective development, which allow developers to access most of the features they could need without necessarily having to import everything piece by piece. Django also follows the DRY principles, which help keep code non-repetitive, non-redundant, and efficient. Finally, this framework can also count on a very large community of users, so it's usually very easy to get support when needed. Many other benefits could be said about Django, such as fast development, secure deployment, and scalability.

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.

Answer:

Spotify : Spotify is a music streaming app that allows users to listen music and other related products (ex: podcasts) via an app installed on their phone. The company uses Django. There are mainly two reasons Spotify chose this framework: fast backend and machine learning options. To provide their users with customized auto-generated playlists, Spotify benefits from Machine Learning with Python while Django provided the full range of Python features to handle the web application.

Eventbrite: Eventbrite is a website that provides event management and ticketing service. Initially, Eventbrite was not relying on Django to support his system. However, after rapid audience growth, the platform started facing scalability issues. The developer's team therefore decided to move on to Django to help to scale their website and better handle high traffic.

The Washington Post: The Washington Post is a newspaper company, offering products both physically and digitally. The company uses Django mainly to handles the huge traffic their site receive every day / every month with a very well, fast and

efficient performance. Django is very useful to scale their website and handle large amounts of data generated by their audience.

Prezi: Prezi is a website that allows users to build interactive presentations, similarly to Powerpoint but usually in a more dynamic way. Prezi runs on Django CMS, a content management system (CMS) written using the Django framework. The company choose to use Django mainly to give its user base dynamic and fully responsive designs.

National Geographic: National Geographic is a television network and magazine that focuses on delivering educational content in subject areas like science, culture, and history. Its online version uses a Django-powered content management system to allow developers to easily manage the website's content, from modules and advertisements to individual images. Django is also useful in the case of National Geographic, since it eases the development of data-driven complex websites.

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.

Answer: Yes, I would use Django in this case, because this framework is very scalable and offer fast processing. Many large companies having a large user base use Django to handle their traffic and it's working very well.

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

Answer: Yes, I would use Django in this case, because Django's MVT architecture ensures development is fast and easy. Secure deployment is another advantage of Django. Similarly, due to its MVT architecture, it is very simple to add new hardware and components along the way with Django.

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

Answer: No, I wouldn't use Django in this case, because Django's offers a lot of out-of-the-box solutions that are very useful in projects that need it, but not really in a very basic

application that doesn't require it (it's too much for the needs). Also, Django's use of prewritten code means it's more server intensive, making it heavy on low-bandwidth systems, which again is not necessarily suitable for a very basic application.

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

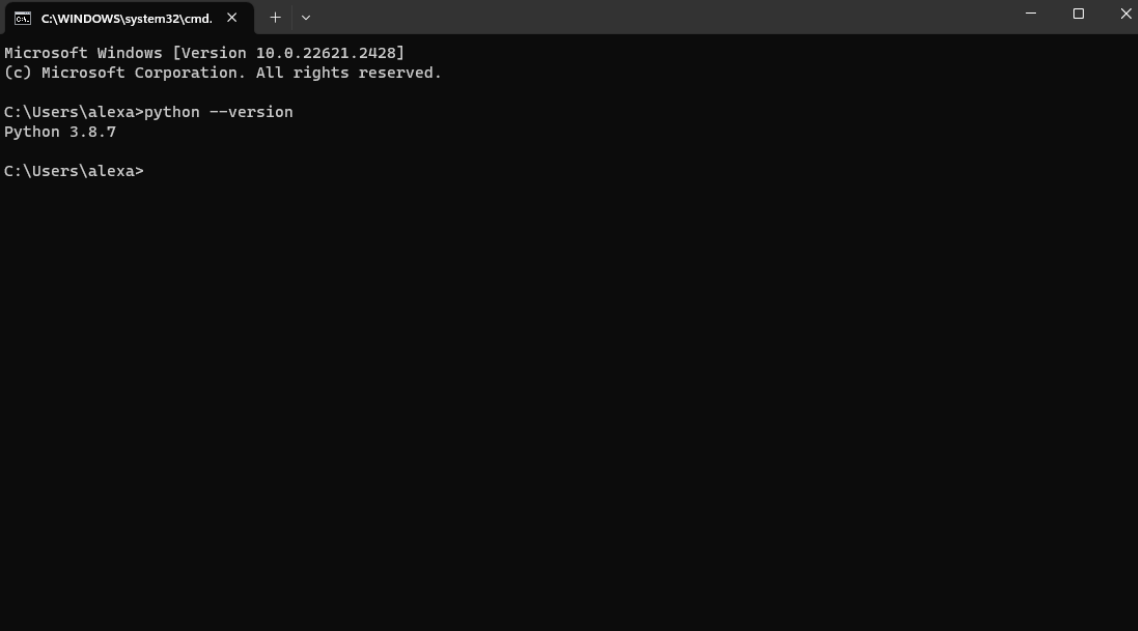
Answer: It depends on the context, but the answer here would be no - Django wouldn't be suitable.

Because Django is a *batteries-included* framework, it could help making the development from the scratch faster and more efficient. However, if a developer wants to have a lot of control over how the application will work / is designed, Django is not the best tool. The reason is that Django is a highly structured web framework that does things a certain way. In a Django project, certain steps have to be taken in an exact manner, so Django's firm structure doesn't really provide room to diverge from the rules and have a lot of control over the app development.

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

Answer: Yes, I would use Django in this case, because Django is open source and has a huge community of contributors. As such, it's usually very easy to get support when needed.

Python installation proof

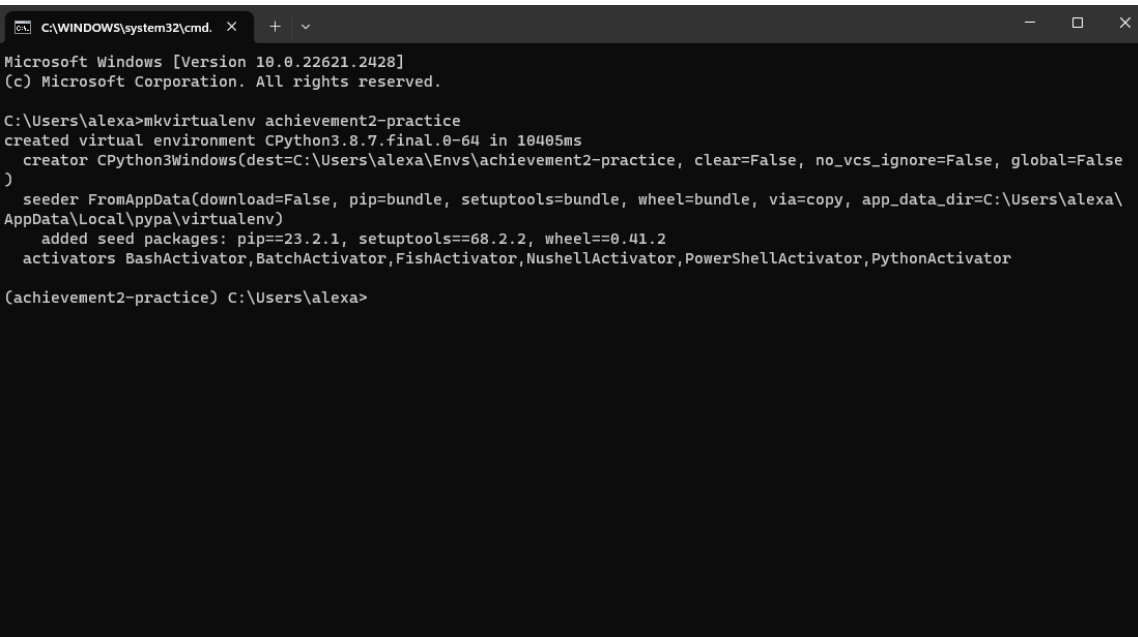


```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\alexa>python --version
Python 3.8.7

C:\Users\alexa>
```

Creation and activation of the virtual environment



```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\alexa>mkvirtualenv achievement2-practice
created virtual environment CPython3.8.7.final.0-64 in 10405ms
  creator CPython3Windows(dest=C:\Users\alexa\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\alexa\AppData\Local\pypa\virtualenv)
    added seed packages: pip==23.2.1, setuptools==68.2.2, wheel==0.41.2
    activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

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

Django installation

```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\alexa>cd Envs

C:\Users\alexa\Envs>cd achievement2-practice

C:\Users\alexa\Envs\achievement2-practice>cd Scripts

C:\Users\alexa\Envs\achievement2-practice\Scripts>.\activate.bat

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

(achievement2-practice) C:\Users\alexa\Envs\achievement2-practice\Scripts>py -m pip install Django
Collecting Django
  Obtaining dependency information for Django from https://files.pythonhosted.org/packages/b9/45/707dfc56f381222c1c798503546cb390934ab246fc45b5051ef66e31099c/Django-4.2.6-py3-none-any.whl.metadata
  Using cached Django-4.2.6-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.6.0 (from Django)
  Obtaining dependency information for asgiref<4,>=3.6.0 from https://files.pythonhosted.org/packages/9b/80/b9051a4a07ad231558fcd8ffc89232711b4e618c15cb7a392a17384bbeef/asgiref-3.7.2-py3-none-any.whl.metadata
  Using cached asgiref-3.7.2-py3-none-any.whl.metadata (9.2 kB)
Collecting sqlparse>=0.3.1 (from Django)
  Using cached sqlparse-0.4.4-py3-none-any.whl (41 kB)
Collecting backports.zoneinfo (from Django)
  Using cached backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl (38 kB)
Collecting tzdata (from Django)
  Using cached tzdata-2023.3-py2.py3-none-any.whl (341 kB)
Collecting typing-extensions>=4 (from asgiref<4,>=3.6.0->Django)
```

```
C:\WINDOWS\system32\cmd. X + v
3546cb390934ab246fc45b5051ef66e31099c/Django-4.2.6-py3-none-any.whl.metadata
  Using cached Django-4.2.6-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.6.0 (from Django)
  Obtaining dependency information for asgiref<4,>=3.6.0 from https://files.pythonhosted.org/packages/9b/80/b9051a4a07ad231558fcd8ffc89232711b4e618c15cb7a392a17384bbeef/asgiref-3.7.2-py3-none-any.whl.metadata
  Using cached asgiref-3.7.2-py3-none-any.whl.metadata (9.2 kB)
Collecting sqlparse>=0.3.1 (from Django)
  Using cached sqlparse-0.4.4-py3-none-any.whl (41 kB)
Collecting backports.zoneinfo (from Django)
  Using cached backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl (38 kB)
Collecting tzdata (from Django)
  Using cached tzdata-2023.3-py2.py3-none-any.whl (341 kB)
Collecting typing-extensions>=4 (from asgiref<4,>=3.6.0->Django)
  Obtaining dependency information for typing-extensions>=4 from https://files.pythonhosted.org/packages/24/21/7d397a4b7934ff4028987914ac1044d3b7d52712f30e2ac7a2ae5bc86dd0/typing_extensions-4.8.0-py3-none-any.whl.metadata
  Using cached typing_extensions-4.8.0-py3-none-any.whl.metadata (3.0 kB)
Using cached Django-4.2.6-py3-none-any.whl (8.0 MB)
Using cached asgiref-3.7.2-py3-none-any.whl (24 kB)
Using cached typing_extensions-4.8.0-py3-none-any.whl (31 kB)
Installing collected packages: tzdata, typing-extensions, sqlparse, backports.zoneinfo, asgiref, Django
Successfully installed Django-4.2.6 asgiref-3.7.2 backports.zoneinfo-0.2.1 sqlparse-0.4.4 typing-extensions-4.8.0 tzdata-2023.3

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(achievement2-practice) C:\Users\alexa\Envs\achievement2-practice\Scripts>django-admin --version
4.2.6

(achievement2-practice) C:\Users\alexa\Envs\achievement2-practice\Scripts>
```