

Exercise 2.1

1. Why is Django popular among developers?

Django is popular among developers for many reasons including scalability, security, and community. It has built in libraries and templates and removes the hassle of web development. This is a huge resource saver and lets you create apps in a timely manner.

2. Five companies that use Django:

- **Instagram** - Instagram is one of the world's most popular social media apps where users share photos and videos. Django helped Instagram to scale the application, process large amounts of data, and manage large numbers of interactions between users. Django's ready to implement solution allowed the company to focus on UI and UX of the app instead of having to worry about the backend technology.
- **Spotify** – Spotify is one of the most popular streaming music apps available. Due to its large library of music and data, the application uses Python alongside Django to handle it all. The two reasons for choosing the framework were a fast backend and machine learning options.
- **Pinterest** - Pinterest is an app that allows users to find ideas like recipes, decorations, and home design inspiration. The site uses Django which provides the ability to scale effectively without affecting its speed. Django has helped its developers manage the website, allowing users to follow each other and share boards and pins.
- **Dropbox** - Dropbox is a cloud-based storage platform for documents, video, and pictures which allows users to access the files anywhere from any device. Dropbox uses Django to add the user history option, synchronize an account across different devices, and to add various types of file-sharing options.

- YouTube – YouTube is a video-sharing platform. YouTube decided to use Django due to its growing popularity and the number of users. Django is used to implement new features and to maintain the speed of the website.

3. Scenarios - Would you use Django?

- You need to develop a web application with multiple users.

Django would be useful for this app. It includes a plugin for features such as user-friendly admin panels and authentication which would work well for multiple users.

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

Django would be a good choice. Since Django removes the hassle of web development you can focus on creating your project logic. This helps in fast development and secure deployment. Since Django is based on MVT, it is suitable for a lot of changes.

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

I would not use Django in this case, since it is basic and doesn't require a database. Django might be too large of a framework to use for this type of application.

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

Django would not be the best to use here. Since Django works the specific Django way, you would lose some control over the internals of your system.

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

Django would be a good fit. Django is very popular and has a huge community of contributors. Due to this it's usually very easy to get support if you need it.

Python Version:

```
Microsoft Windows [Version 10.0.22621.1992]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\amart>python --version  
Python 3.8.7  
  
C:\Users\amart>
```

Activate Environment:

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

Django Version:

```
(achievement2-practice) C:\Users\amart\Envs> django-admin --version  
4.2.4
```