

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

- Django is popular among web developers for a variety of reasons. First, it offers fast prototyping capabilities. Second, Django is scalable and secure. Third, the template handles all of the controller code, so you don't have to spend all of the extra time writing it yourself. And fourth, Django boasts a variety of built-in tools such as ORM, an admin interface, and authentication systems.

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 (Meta)
 - Product/Service: Social media platform for sharing photos and videos.
 - Usage of Django: Originally built on Django, Instagram relies on the framework for its backend development, handling various aspects like user interactions, content delivery, and data management.
- Spotify
 - Product/Service: Music/podcast streaming service
 - Usage of Django: Django is used in some parts of Spotify's backend, particularly with user authentication and playlist management.
- Pinterest

- Product/Service: A platform for users to discover and save ideas for various projects and interests
 - Usage of Django: Pinterest has employed Django for managing user data, boards, and interactions on the platform.
- Dropbox
 - Product/Service: Cloud-based file storage and collaboration platform.
 - Usage of Django: Django has been used in aspects of Dropbox's file management and user accounts.
- Eventbrite
 - Product/Service: Event management and ticketing platform.
 - Usage of Django: Eventbrite relies on Django for its user authentication, event creation, and management of event-related data.

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.
 - I would use Django for this scenario. Django provides a robust and secure framework for building apps with user authentication, and it allows for the management of multiple users. It has many built-in features such as the authentication system and ORM that streamline the

development process and ensure the app's scalability and security.

- You need fast deployment and the ability to make changes as you proceed.
 - Yes, Django would be a suitable choice for this scenario. Django's "batteries-included" philosophy provides a comprehensive set of tools and features, allowing for rapid development and deployment. The Django framework comes with a built-in development server and a straightforward deployment process, making it convenient for continuous development and quick iterations.

- You need to build a very basic application, which doesn't require any database access or file operations.
 - I would not use Django. Django is designed for apps that interact with a database and require file management. Such a simple app would be bogged down by Django's server

intensiveness. Flask might be the way to go since it is lightweight and flexible.

- You want to build an application from scratch and want a lot of control over how it works.
 - You definitely would not want to use Django here. Django uses MVT, where the template handles all of the controller code. Also, Django is highly structured and opinionated, so one must follow 'the Django way'.

- You're about to start working on a big project and are afraid of getting stuck and needing additional support.
 - If you're concerned about potential roadblocks and might need additional support, Django can be an excellent choice. Django has a huge and active community, extensive documentation, and tons of tutorials online.

- Python Installation/Version Verification

```
7.2.0
(web-dev) jpcody@MacBook-Air ~ % python --version
```

```
Python 3.8.7
(web-dev) jpcody@MacBook-Air ~ % █
```

- Activated Virtual Environment (command uses the alias: a2p)

```
jpcody@MacBook-Air ~ % echo "alias a2p='source achievement2-practice/bin/activate'" >> ~/.bashrc
source ~/.bashrc

jpcody@MacBook-Air ~ % a2p
source: no such file or directory: achievement2-practice/bin/activate
jpcody@MacBook-Air ~ % cd Desktop/CareerFoundry
jpcody@MacBook-Air CareerFoundry % a2p
(achievement2-practice) jpcody@MacBook-Air CareerFoundry % █
```

- Django Installation/Version Verification

```
(achievement2-practice) jpcody@MacBook-Air CareerFoundry % pip install django

Collecting django
  Downloading Django-4.2.8-py3-none-any.whl (8.0 MB)
    |████████████████████| 8.0 MB 6.8 MB/s
Collecting backports.zoneinfo; python_version < "3.9"
  Downloading backports.zoneinfo-0.2.1-cp38-cp38-macosx_10_14_x86_64.whl (35 kB)
Collecting asgiref<4,>=3.6.0
  Downloading asgiref-3.7.2-py3-none-any.whl (24 kB)
Collecting sqlparse>=0.3.1
  Downloading sqlparse-0.4.4-py3-none-any.whl (41 kB)
    |████████████████████| 41 kB 1.1 MB/s
Collecting typing-extensions>=4; python_version < "3.11"
  Downloading typing_extensions-4.9.0-py3-none-any.whl (32 kB)
Installing collected packages: backports.zoneinfo, typing-extensions, asgiref, sqlparse, django
Successfully installed asgiref-3.7.2 backports.zoneinfo-0.2.1 django-4.2.8 sqlparse-0.4.4 typing-extensions-4.9.0
WARNING: You are using pip version 20.2.3; however, version 23.3.2 is available.
You should consider upgrading via the '/Users/jpcody/Desktop/CareerFoundry/achievement2-practice/bin/python3 -m pip install --upgrade pip' command.
(achievement2-practice) jpcody@MacBook-Air CareerFoundry % python3 -m django --version

4.2.8
(achievement2-practice) jpcody@MacBook-Air CareerFoundry % █
```

