

Why is Django so popular among web developers?

Django is popular among web developers due to its high productivity, built-in scalability and security features, and the support of an active community and ecosystem. Its robust framework reduces development time using built-in authentication and ORM. Its active community provides an abundance of 3rd party packages and documentation.

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

1. Instagram uses Django for its web application. Django helps manage massive amounts of user-generated content to ensure a responsive and reliable platform.
2. Spotify utilizes Django for its web backend. It handles user accounts, playlists, and content delivery, contributing to the platform's stability and user experience.
3. Pinterest uses Django to manage its content and user interactions. Django helps handle the creation, organization, and discovery of pins and boards.
4. NASA's website, which provides information about space exploration missions and scientific research, uses Django. It enables content management, user authentication, and data presentation on their web platform.
5. Dropbox uses Django for various aspects of its web application. Django helps with user account management, file sharing, and collaboration features.

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 I would use Django in this situation because of its robust user authentication and authorization features allow for easy and safe management of user account info.

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

YES I would use Django because it has of its "batteries included" approach and built in tools allow for fast and flexible development.

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

NO I would not use Django because it Django is more than is needed here. There are simpler, more appropriate choices like Flask.

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

NO I would not use Django because it forces me not to diverge from the Django rules.

You're about to start working on a big project and are afraid of getting stuck and needing additional support. YES I would use Django because it has a large community and lots of documentation.

```
(cf-python-base) PS C:\Users\User> python --version  
Python 3.8.7
```

```
C:\Users\User\Envs\achievement2-practice>cd Scripts  
C:\Users\User\Envs\achievement2-practice\Scripts>activate.bat  
(achievement2-practice) C:\Users\User\Envs\achievement2-practice\Scripts>if defined _OLD_VIRTUAL_PYTHONPATH (set "PYTHONPATH=" ) else (set "_OLD_VIRTUAL_PYTHONPATH=" )  
(achievement2-practice) C:\Users\User\Envs\achievement2-practice\Scripts>
```

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