

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

Django's MVT architecture ensures development is fast and easy. Thanks to its MVT architecture, Django handles network transmission and content delivery at a high speed. Additionally, Django—due to MVT—is based on loosely coupled architecture, which makes it very simple to add new hardware and components. As such, scaling a Django-based system isn't a problem.

2. 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.
 1. Pinterest: this company uses a website where users can post images / recipes/ videos and can also get inspired by them to create their own projects. Django framework helps Pinterest users subscribe to other users and share their boards.
 2. Eventbrite: Eventbrite is an event management and ticketing service. The benefits of moving to Django according to Eventbrite were, unit-test framework, database model abstraction (Django ORM), request middleware, URL routing, form building.
 3. Mozilla, Firefox browser. With Django, Firefox can tackle large amounts of traffic and API hits in a more efficient manner.
 4. Spotify: audio streaming provider for music and podcasts. Python is used for Spotify's back-end services and data analysis. So, Spotify uses Django app to increase functionality.
 5. Instagram: Instagram is a social media application for sharing photos and videos. Using Django, Instagram Engineering was able to build custom tools for platform scaling and efficiency.
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.
I would use Django because Django is good when there is large number of users both for scalability and using a large database using its MVT architecture.
- You need fast deployment and the ability to make changes as you proceed.
Django is based on MVT, making it suitable for fast prototyping, a lot of changes, and high-speed applications. So, Django would be good here.
- You need to build a very basic application, which doesn't require any database access or file operations.
Since Django is good for large application with a large database not using this function will be for waste so Django would not be good to use here.

- You want to build an application from scratch and want a lot of control over how it works.
Django is a *batteries-included* system and works the specific Django way, so you'll lose some control over the internals of your system. So Django may not be good in this scenario.
- You're about to start working on a big project and are afraid of getting stuck and needing additional support.
Django is open source and has a huge community of contributors. As such, it's usually very easy to get support if you need it.

Achievement 2 environment creation

```
(achievement2-practice) karenhakobyan@Karens-MacBook-Air Task %
```

Django Version

```
(achievement2-practice) karenhakobyan@Karens-MacBook-Air Task % django-admin --version
4.2.6
(achievement2-practice) karenhakobyan@Karens-MacBook-Air Task %
```