

Answers:

2.

Django's popularity among web developers can be attributed to its emphasis on simplicity, efficiency, and robustness. Its high-level abstractions and built-in features, such as an ORM system, admin interface, and authentication system, significantly accelerate the development process. Additionally, Django follows the "Don't Repeat Yourself" (DRY) principle and promotes best practices, providing developers with a powerful and well-organized framework for building scalable and maintainable web applications.

3.

1. Instagram (Meta)

- Product/Service: Social media platform for sharing photos and videos.

- Use of Django: Django was the web framework initially used by Instagram. It provided a solid foundation for building and scaling the platform.

2. Pinterest

- Product/Service: Visual discovery and bookmarking platform.

- Use of Django: Pinterest has utilized Django for its backend development, leveraging its features to handle user interactions and data processing.

3. Dropbox

- Product/Service: Cloud-based file storage and collaboration platform.

- Use of Django: Dropbox has employed Django for certain parts of its web infrastructure, benefiting from Django's rapid development capabilities.

4. Disqus

- Product/Service: Blog comment hosting service.
- Use of Django: Disqus, known for its scalable comment system, has used Django for its web development, taking advantage of the framework's simplicity and versatility.

5. Eventbrite

- Product/Service: Event management and ticketing platform.
- Use of Django: Eventbrite has been reported to use Django for its web applications, facilitating the creation and management of events on their platform.

4.

1. You need to develop a web application with multiple users:

- Use of Django: Yes, Django is well-suited for developing web applications with multiple users. It includes built-in features for user authentication, authorization, and session management, making it efficient for handling user-related functionalities.

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

- Use of Django: Yes, Django supports fast deployment with its built-in development server, and it provides tools like Django's development environment, which allows for quick iterations. The Django framework follows the "Don't Repeat Yourself" (DRY) principle, aiding in efficient development and changes.

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

- Use of Django: It might be overkill. Django is a full-stack web framework designed for projects that involve databases, templates, and other web-related functionalities. For a very basic application without database or file operations, a micro-framework like Flask, which is lightweight and minimalistic, might be more suitable.

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

- Use of Django: It depends. Django follows the "batteries-included" philosophy, providing a lot of built-in features. If you want more control and prefer a more modular approach, you might consider a micro-framework like Flask. Django is powerful but might come with more predefined structure.

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

- Use of Django: Yes, Django is a good choice for large projects. It has a large and active community, extensive documentation, and many third-party packages. The Django framework's structure and conventions can provide a solid foundation and help prevent common pitfalls. The community support can be beneficial when tackling complex issues in a big project.

5.

```
(web-dev) (cf-python-base) python --version  
Python 3.8.18
```

7.

```
• (achievement2-practice) (cf-python-base) workon achievement2-practice
```

8.

```
(achievement2-practice) (cf-python-base) django-admin --version  
4.2.10
```