

## **Difference Between Django And Flask Python Framework**

Flask and Django are two important web development frameworks for Python. Python is a versatile programming language and it provides a wide variety of web frameworks, where the developer takes full advantage of the full-stack Python web framework. It enhances the development of complex web applications.

**Framework:** Framework is a collection of modules that helps programmers to develop codes for web applications. Frameworks are fully developed packages that make the web development process easier, faster, scalable, and reliable.

Every framework has its unique quality which can be used as per the project requirements.

### **What is Flask?**

**Flask** was created by Armin Ronacher. Flask is considered a micro web framework is written in Python programming. The word “micro” means focusing to keep the core extensible but straightforward. It is not dependent upon external libraries to perform the tasks of a framework. Flask framework is more independent, flexible, and simple; so many developers prefer to start with Flask.

**Companies that are using Flask:** MIT, Uber, Reddit, Netflix, Mozilla, Airbnb, Lyft, etc.

### **What is Django?**

**Django** is an open-source, flexible, accessible, and follows Model View Controller patterns. Django is commonly called a “batteries-included” system approach or the “framework for fussy budgets with deadlines”. It is a high-level framework that speeds up the web development process and increases efficiency. This framework has been inspired by the old frameworks like CherryPy, Plone, Zope, etc. The framework was named after the famous jazz guitarist Django Reinhardt.

**Companies that are using Django:** Instagram, Pinterest, Bitbucket, Disqus, Spotify, National Geographic, etc.

## **Features of Flask**

- **Lightweight:** Flask is a lightweight framework because it is independent of external libraries and it gives a quick start for web development having complex applications.
- **Compatible:** Flask is compatible with the latest technologies such as machine learning, agile development, cloud technologies, etc.
- **Independent:** Flask allows full control to the developers for creating web applications. A developer can do the experiment with the libraries and architecture of the framework.
- **Integrated Unit Testing:** Flask offers an integrated unit testing feature that helps in faster debugging, robust development, and independence to do experiments.
- **Flexible and Scalable:** Flask supports WSGI templates that help in flexibility and scalability in the web development process.

- **Secure Cookies:** Secure cookie is an attribute of an HTTP request that enables the security of channels and ensures no unauthorized person has access to the text. Flask supports the feature of secure cookies.

## Features of Django

- **Fast:** Django is an extremely fast framework, the working process from concept to completion is insanely fast.
- **Versatile:** Django is a versatile framework that helps developers to work on different platforms like WordPress, LinkedIn, Youtube, CNN, The New York Times, etc.
- **Portable:** It is a python based framework hence it is portable.
- **Secure:** This framework guarantees security with powerful authentication systems and protocols to avoid unauthorized access, cyber attack, clickjacking, etc.
- **Adaptable:** It is adaptable to different formats such as HTML, JSON, XML, and many more.

Parameter	Flask	Django
<b>About Framework</b>	Flask is a lightweight framework with minimalist features.	Django is a full-stack web framework that includes a batteries-included approach
<b>Project</b>	Flask is suitable for	Django is suitable for

<b>Layout</b>	single-page application	multiple-page application
<b>Working Style</b>	Flask working style is diversified	Django's working style is Monolithic
<b>Debugger</b>	Flask provides an in-built debugger that offers virtual debugging	Django does not provide any virtual debugging
<b>HTML Page</b>	Dynamic HTML pages are not supported by the Flask	Dynamic HTML pages are supported by the Django
<b>Built-in Templates</b>	Flask built-in template engine Jinja2 is based on the Django template engine	Django offers built-in templates that are ready to use for the developers
<b>Lines of Code</b>	For a simple functionality, Flask requires fewer lines of code	For the similar functionality, Django requires 2 times more lines of codes than Flask
<b>Flexibility</b>	Flask offers flexibility to the developers as it is a micro-based framework with extensible libraries	Django offers in-built features so it is less flexible and developers cannot make changes to the modules
<b>Database</b>	Flask doesn't offer	Django offers popular

<b>Support</b>	a database management system and it uses SQLAlchemy for database requirements	relational database management systems such as Oracle, MySQL, etc
<b>Admin Tool</b>	Flask does not offer any ready-to-use administrator tool for the developers	Django offers ready to use admin framework that can be customized as per the need
<b>Dynamic Behavior</b>	In Flask, a developer can change the application features latterly as per the need, as flask provides flexibility to expand the application quickly	Django is not suitable for a project whose requirement changes dynamically
<b>Structure</b>	Flask has a random web frame structure	Django has a conventional structure
<b>Security</b>	Flask offers in-built security features for common threads like CSRF, SQL Injection XSS, etc	Django is more secure than other frameworks because it has a small codebase so the possibility of unauthorized access

		is less. However, we can use third-party libraries and extensions to make it more secure.
<b>Usage</b>	Flask is used by the companies that need to do an experiment with their modules and architecture such as Reddit, MIT, Netflix, Lyft, etc	Django is used for high-end performance and it is used by Udemy, Instagram, Coursera, etc

- **Scalable:** This framework can be easily scalable and easy to maintain.

## Flask vs Django

### Pros of Flask

- It is a lightweight framework that offers hassle-free development
- Provide flexibility to the developer to experiment with their modules or architecture
- It is suitable for small projects
- Offers a built-in development server and fast debugger
- Easily scalable for the applications
- Support for secure cookies
- Uses Ninja2 Template engine
- It has integrated support for unit testing
- APIs are coherent and neat
- Provide strong WSGI support

### Pros of Django

- It offers MVC (Micro View Controller) Architecture
- It supports dynamic HTML pages
- Preferred for big projects
- Highly secure framework
- Pre-defined templates for developers
- Has integrated unit testing for web applications
- It is portable
- It supports multi-language and multi-character
- Have support for frontend tools such as jquery, Ajax, Pyjamas, etc
- Supports relational databases

### **Cons of Flask**

- Not suitable for big applications and projects
- It does not offer an in-built admin site for maintaining records for insertion, deletion, and modification
- Lack of Database and ORM
- Complex maintenance for large applications
- Libraries must be manually installed
- Limited community support and compare to Django
- The Minimum Viable Product (MVP) development process is slow

### **Cons of Django**

- Monolithic style makes things difficult and fixed
- Codebase size is much higher as compared to the flask
- Templates failed silently
- Developers need broad knowledge to work with Django
- Only allow handling a single request at a time
- The large size of the code

- Auto reload restarts the entire server

## **Which is Better? Flask or Django**

Both frameworks have their own functionality and features. Where Django is an open-source framework for the rapid development of web applications and Flask is a light-end framework offering standard functionality.

If a developer is looking for a small web-based application then they can prefer Flask else for a bigger project they can move to Django, as Django offers extensive community support to create a unique website.

Django is complex and requires vast knowledge to design a sophisticated application. Django is better to design excellent applications but it is a little hard for beginners to understand about it, whereas flask requires fewer lines of code and is easy to understand for the beginners but not feasible for larger applications.

## **Conclusion**

So, in this article, we have got a brief knowledge of both the frameworks along with their functionality, features, and how they differ from one another. A learner can choose whatever suits them the best but always make sure to master it like a pro as they have a surging demand and are indispensable to the industry of web development. Having fundamental knowledge of both frameworks can put you on the map prior to other candidates while applying for a job.