**# What is Flask?**

Flask is an micro framework offering basic features of web app. This framework has no dependencies on external libraries. The framework offers extensions for form validation, object-relational mappers, open authentication systems, uploading mechanism, and several other tools. Flask may be a generally youthful framework, as it were in utilize since 2010. Flask is considered more “Pythonic” than Django is basically since Flask web application code is, in most cases, more unequivocal.

## # What is Django?

Django is a web development framework for Python. This framework offers a standard method for fast and effective website development. It helps you in building and maintaining quality web applications. It enables you to make the development process smooth and timesaving. It is a high-level web framework which allows performing rapid development. The primary goal of this web framework is to create complex database-driven websites. Django is commonly called a “batteries-included” system approach—or the “framework for fussbudgets with deadlines.” This implies that Django makes it simple for Python designers to jump into web applications rapidly without requiring planning into the app’s framework ahead of time.

**# Django vs Flask**

Django and Flask are two of the most popular web frameworks for Python. Flask showed up as an alternative to Django, as designers needed to have more flexibility that would permit them to decide how they want to implement things, while on the other hand, Django does not permit alteration of their modules to such degree. Flask is truly so straightforward and direct that working in it permits an experienced Python designer making ventures inside truly tight timeframes. Here are a few Flask and Django use cases: Flask is extraordinary for building basic locales with inactive substance, like blogs; it gives all the usefulness you would like and permits for customization to a tremendous degree whereas Django is incredible for building complex destinations with energetic substance, with adaptability in intellect; enormous ventures that require out-of-the-box arrangements can be sent truly quick.

**# Admin Interface**

Not at all like Flask, Django incorporates a ready-to-use admin system that empowers clients to carry out the extend organization errands consistently. The useful admin interface is what makes Django a capable web system. Based on the venture, it naturally creates admin modules. Engineers can customize the admin interface to meet the trade needs.

**# Database**

Django has bolstered for the ORM framework.

Advantage of ORM framework includes:

1. Developers can take advantage of the ORM framework to work with an assortment of databases, including PostgreSQL, SQLite, Prophet, MySQL and more.
2. Developers don’t have to type in long SQL inquiries to execute common database operations. Whereas Flask doesn’t support the ORM framework.
3. Designers are required to type in SQLAlchemy (Protest Social Mapper and SQL toolkit for Python) in arrange to perform common database operations.

**# Built-in template engine**

Not at all like Django, Flask doesn’t have a built-in layout motor. Flask is based on the Jinja2 format motor. Jinja2 is itself impacted by the Django format motor. Its employments coordinate a sandboxed execution environment, permitting engineers to speed up the advancement handle for energetic web applications. While Django incorporates a built-in format motor that permits engineers to make user-facing layers for web applications consistently and quickly.

**# KEY DIFFERENCES:**

* Flask provides support for API while Django doesn’t have any support for API.
* Flask does not support dynamic HTML pages and Django offers dynamic HTML pages.
* Flask is a Python web framework built for rapid development whereas Django is built for easy and simple projects.
* Flask offers a diversified working style while Django offers a Monolithic working style.
* URL dispatcher of the Flask web framework is a RESTful request on the other hand, URL dispatcher of Django framework is based on controller-regex.
* Flask is WSGI framework while Django is a Full Stack Web Framework.

Flask and Django are the two most popular Python- frameworks. Here are some important differences between Django and Flask.

|  |  |
| --- | --- |
| **\*\*Flask\*\*** | **\*\*Django\*\*** |
| Created in 2010 | Created in 2005 |
| Python web framework built for rapid development. | Python web framework built for easy and simple projects. |
| Flask is WSGI framework. | Django is a Full Stack Web Framework. |
| Flask provides support for API. | Django doesn’t have any support for API. |
| Support Visual Debug. | No support for Visual Debug. |
| Flask allows you to use multiple types of databases. | Django doesn’t offer multiple types of databases. |
| Flask has no default support for forms, but you can use WTForms to fill the gap. | Django comes Form with which can be integrate with the ORM and the admin site. |
| Flask does not offer dynamic HTML pages. | Django offers dynamic HTML pages. |
| The request-based object is imported from the flask module, which is a global variable in Flask. | All views are set as an individual parameter in the Django. |
| Flask is much younger platform compared to Django. | Django is a very mature framework. |
| Flask offers a diversified working style. | Django offers a Monolithic working style. |
| It supports an extension which could be implemented in the framework. | Django has its own module library. So, it stores several prewritten codes. |
| The structure of the project layout for Flask web framework is random. | The structure of the project layout for the Django is conventional. |
| Flask web framework uses a Ninja2 template design. | Django web framework helps you to utilizes the View web templating system. |
| URL dispatcher of the Flask web framework is a RESTful request. | URL dispatcher of this Django framework is based on controller-regex. |
| Flask does not offer a built-in bootstrapping tool. | Django-admin enables developers to start building web applications without any external input. |
| Flask is a good choice if you want a lightweight codebase. | The best feature of Django is Robust documentation. |
| Flask framework is suitable for single application. | Django framework allows developers to divide a project into multiple page application. |
| Flask Web Framework doesn’t offer support for third-party applications. | Django Web Framework supports many third-party applications. |
| Git hub stars 48.8 K | Git hub stars 47.1 K |
| Best features of the flask are it is lightweight, open source, and offer minimal coding for developing an application. | The best features of Django are Rapid development, Open source, Great Community, Easy to learn. |
| \*\*Famous companies using Flask are Netflix, Reddit, Lyft, MIT\*\* | \*\*Famous companies using Django are Instagram, Coursera, Udemy.\*\* |

**# Advantages of Flask**

Here, are pros/benefits of using Flask

* Higher compatibility with latest technologies
* Technical experimentation
* Easier to use for simple cases
* Codebase size is relatively smaller
* High scalability for simple applications,
* Easy to build a quick prototype
* Routing URL is easy
* Easy to develop and maintain applications
* Database integration is easy
* Small core and easily extensible
* Minimal yet powerful platform
* Lots of resources available online especially on GitHub

**# Advantages of Django**

Here, are pros/benefits of Django framework:

* Django is easy to set up and run
* It provides an easy-to-use interface for various administrative activities.
* It offers multilingual websites by using its built-in internationalization system
* Django allows end-to-end application testing
* Allows you to document your API with an HTML output
* REST Framework has rich support for several authentication protocols
* It is used for rate-limiting API requests from a single user.
* Helps you to define patterns for the URLs in your application
* Offers built-in authentication system
* Cache framework comes with multiple cache mechanisms.
* High-level framework for rapid web development
* A complete stack of tools
* Data modelled with Python classes

**# Disadvantage of Flask**

Here, are cons/drawback of Flask

* Slower MVP development in most cases,
* Higher maintenance costs for more complex systems
* Complicated maintenance for larger implementations.
* Async may be a little problem
* Lack of database and ORM
* Setting up a large project requires some previous knowledge of the framework
* Offers limited support and smaller community compared to Django

**# Disadvantage of Django**

Here, are cons/drawback of the Django framework

* It is a monolithic platform.
* High dependence on Django ORM. Broad Knowledge required.
* Fewer Design decisions and Components.
* Compatibility with the latest technologies
* A higher entry point for simple solutions
* The larger size of the code
* Too bloated for small projects
* Underpowered templating and ORM
* Templates failed silently
* Auto reload restarts the entire server
* High learning curve
* Documentations does not cover real-world scenarios
* Only allows you to handle a single request per time.
* Routing requires some knowledge of regular expressions
* Internal subcomponents coupling
* You can deploy components together, which can create confusion.

**# Which is Better?**

* You should prefer flask if you want the granular level of control while a Django developer relies on an extensive community to create unique website.
* Django combined with the REST Framework helps you to build powerful APIs, whereas Flask requires more work, so there are high chances to make a mistake.
* The best method is to build a few basic CRUD apps with both frameworks and decide which framework fits your project style better.