Flask vs Django: Must-Know Differences

Python is a general-purpose programming language. It can be used to create games, desktop applications, script as well as websites. Flask and Django are two popular frameworks of Python for web development. In this tutorial, we take a look at key differences between Django and Flask.

## Django

Django is a web development framework of Python that takes covers multiple conventional functionalities to build safe and maintainable websites. As a website developer, all you require is to make your business philosophy. Django is free and open-source. Django has a very dynamic and helpful community assistance with excellent documentation.

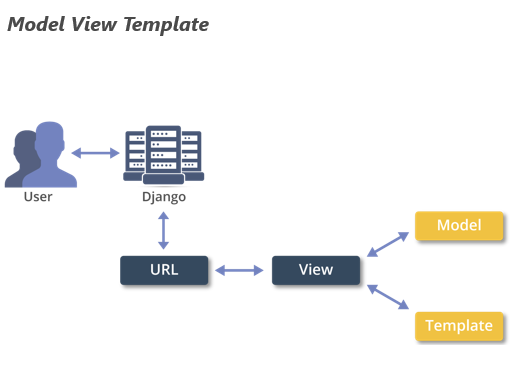
### Features of Django

* You can create almost any type of website using Django, whether it is a social network site, content management, document management, video sharing and the list goes on.
* When you create a simple Django project, it auto-creates the Admin Panel and provides features like user management, cross-site request forgery etc.
* Django keeps your files in a specific organized way. Django code is very reuseable and easy to maintain.
* Django is movable. It is written in Python, currently the most popular programming language that can run anywhere.

### 

### Django Architecture

Django follows a model view template architecture. Models keep track of the database tables, template holds the frontend files and view connects the model and template.



## Flask

Flask is also known as a micro framework, is a Python's lightweight framework. Flask comes with several plugins and libraries that makes all type of tasks easy. If you want to take complete control of your website, then Flask should be your choice. If you are new to Python, start your web development with Flask to get the feel of backend and frontend both and learn the core concepts well.

### Features of Flask

`

* In Flask, developers have full control over the application during the development stage.
* Flask has a fast debugger and built-in development server.
* Flask has a neat API
* Flask is easy to configure.

|  |  |
| --- | --- |
| **Django** | **Flask** |
| Full-stack web framework that supports the batteries-included approach. | Light-weight framework with minimum peculiarities. |
| Fast Development | Need to create from scratch |
| Comes with Admin Panel. | No Admin Panel |
| Comes with built-in template engine | Flask’s template engine Jinja2 is based on Django’s template engine. |
| The project can be divided into small pieces that makes development easy. | No such functionality. |
| Comes with built-in bootstrapping tool | Admin features are not as noticeable as in Django. |
| Comes with built-in ORM system. No need to write SQL queries | SQL queries have to be written for common tasks. |
| Django is considered to be more popular because it provides many out of box features and reduces time to build complex applications. | Flask is a good start if you are getting into web development. There are many websites built on the flask and gain heavy traffic, but not as much compared to the ones in Django. |
| Django is not suitable for projects where requirements change dynamically. | With Flask, a simple application can be later changed to add more functionality and make it complex. It provides flexibility to expand the application quickly. |

## 