

# Flask vs. Django in 2022 – Which Python Framework To Choose and When?

When you're looking for a [Python framework to build a Web app](#), two options stand out: lightweight Flask and comprehensive Django. Both are successful high-level Python Web frameworks. Both are widely used by successful businesses. But their philosophies, strengths, and weaknesses are completely different.

In this text, we'll explain what Django and Flask are and when you should choose each of them, focusing on the business perspective.

Not sure if Python is a way to go? Check out [when to use Python](#) or [compare it to other languages](#).

## Which is Better? Flask vs Django

Flask and Django are the most popular Python web frameworks, according to [JetBrains Python Developers Survey 2021](#). But does being slightly more popular mean that Flask is better than Django?

Unfortunately, the answer to the *Flask Vs Django* debate is not that simple. Django's long-time popularity comes from its stability, powerful features, and exquisite documentation. Flask has a bigger talent pool and gives developers more creative freedom.

Don't ask what is better - Django or Flask. Ask what's better for your project.

## Django – For Perfectionists With Deadlines

Django's general goal is to make it easy for developers to create quality Web apps quickly, using less code.

It's an open-source high-level Python Web framework with built-in dependable security measures, designed to bring quick results and be easily scalable. That means you can get your MVP fast, and add more to your app later with relative ease.

The Django approach is to equip the framework with everything a web app developer will need, at the cost of some creative freedom. This usually makes Django better than Flask for businesses with a strong grasp of what the final product should be like.

## Why use Django?

The framework comes as a bundle of standard functionalities that you can pick and choose from.

The developers don't have to seek the best solution or reinvent the wheel themselves – the proper component may already be there. This means your developers can focus on creating the app as a whole and improving elements unique to your app. Thanks to its modular nature, it's easier to work on each functionality independently and scale up your project.

Django's all-inclusive approach can also positively impact the security of your app. Third-party extensions used in Flask are a potential risk and need regular monitoring and upkeep.

Django's long presence comes with excellent documentation and great popularity. You can always count on a huge, passionate community already having a solution to the problem your developers encountered.

As of August 2022, Django's [official forum](#) hosts thousands of topics, there are 23,000+ questions tagged [django] on [Stack Overflow](#), and the [community on Reddit](#) has more than 104,000 members.

## Django cons to consider

Less flexibility can be a plus in some circumstances, but it's limiting to developers, especially those more creative ones. Django's extended nature leads to a bigger codebase (that includes unused features and extra code).

The framework is also a little bit more challenging to grasp at the beginning than Flask.

## What apps are built with Django?

[Bitbucket](#) is a powerful Django success story. This Git code management system for professional teams started in May 2008 and launched only one month later. The developers [praised](#) Django's native elements and orderly structure that allowed working quickly and scaling up easily. In 2010, Bitbucket was bought by Atlassian.

Other well-known businesses that use Django:

- [Disqus](#)
- [Instagram](#)
- [Pinterest](#)
- [Spotify](#)

## **When to use Django?**

Enjoy Django's organized system and prepackaged goods to quickly develop an MVP and then add new functionalities.

Stop reinventing the wheel and create a complex app in less time using Django's standard features.

## **Flask – Web Development, One Drop at a Time**

Flask takes a radically different approach to creating web apps. It's a microframework, equipped only with the web app development essentials. Its strength lies in its customizability.

Flask is flexible, allowing you to experiment and switch directions easily. This usually means Flask is better than Django if you're going for an open-ended solution or want to use a variety of independent features.

## **Why is Flask good?**

Flask is a lightweight web application framework designed to get results fast and leave room to make the app more detailed in the future. With Flask, your project's code always consists only of what developers put in it, with no unnecessary code responsible for features you don't use.

Many developers love this framework because of its simplicity, resulting in clean, concise code. Active communities (like those on [Facebook](#) and [Reddit](#)) and useful documentation make it easy to find relevant information, appropriate extensions, and simple solutions to encountered difficulties.

Others choose Flask for the freedom to choose from multiple solutions or even create a new one. Not having to follow a strict convention could accelerate the development process for experienced developers who enjoy the added flexibility.

## **Disadvantages of Flask**

Despite Flask having virtually every Django-like security measure in form of a downloadable extension, the usage of third-party additions comes with more risk than choosing the monolithic Django.

In Django, all security features are already there. In Flask every third-party extension is developed by various teams, making updates harder and potentially increasing the upkeep time for the developer.

## What is Flask used for?

When talking about Flask use, Airbnb's workflow management platform Airflow [adopted Flask for its Web application](#).

Other popular companies that use Flask include:

- [Lyft](#)
- [Netflix](#)
- [Red Hat](#)
- [Uber](#)

## When to use Flask?

Flask works best when you take advantage of its flexibility and customization capabilities. If you're delaying some technological decisions for later, it can be beneficial to depend on the easily customizable Flask rather than Django.

You can also use Flask for simple Web apps that have limited functionalities.