**DJANGO VS FLASK**

**WHY DJANGO/FLASK MIGHT BE BETTER**

Django & Flask are two of the most popular web frameworks for Python (although there are many more).

If you decide you want to build a [web application](https://www.codementor.io/learn-development/speed-up-web-development-process), and you would like to develop it in Python, you'll probably want a so-called web framework. There are a lot of repetitive and boring parts of building backend logic, user interface, and hooking everything up to the Internet so that users can navigate your app in their browser. A web framework aims to implement all the functionality common to most web applications, such as mapping URLs to chunks of Python code.

* **Django** on the other hand, adopts an all inclusive “batteries included” approach, providing an admin panel, ORM (Object Relational Mapping), database interfaces, and directory structure straight out of the box.
* **Flask** is a microframework that is focused on simplicity, minimalism and fine grained control. It implements the bare minimum, leaving the developer with complete freedom of choice in terms of modules and add-ons.

**QUICK COMPARISON**

If you're here for quick answers, this section is for you!

* **Main contrasts:**
  + Flask provides *simplicity*, *flexibility* and *fine-grained control*. It is *unopinionated* (it lets you decide how you want to implement things).
  + Django provides an *all-inclusive* experience: you get an *admin panel*, *database interfaces*, an *ORM*, and directory structure for your apps and projects out of the box.
* **You should probably choose:**
  + Flask, if you're focused on the experience and learning opportunities, or if you want more control about which components to use (such as what databases you want to use and how you want to interact with them).
  + Django, if you're focused on the final product. Especially if you're working on a straight-forward application such as a news site, an e-store, or blog, and you want there to always be a single, obvious way of doing things.
* **More information:**
  + Django has been around for longer — it was first released in 2005, while Flask debuted in 2010.
  + Flask has a much lighter footprint, comprising under 10000 source lines of code to Django's approximately 240000 lines.

**PROS AND CONS OF FLASK AND DJANGO**

Before we look at whether Flask or Django is more suitable in particular cases, let’s take a look at the pros and cons of each framework.

**1- Development speed**

Django was created for fast development of complex web apps. Fully featured right out of the box, developers have all the tools they need to implement and develop easily scalable, reliable, and maintainable web apps, in record time. Flask’s simplicity also allows experienced developers to create smaller apps in short timeframes.

**2- Learning curve**

For those not already familiar with Python or web frameworks, it is generally accepted that Django has a slightly steeper learning curve than Flask. Django does provide benefits if you need to change your development team mid-way through the process, or scale the app with a new team after project completion. New, experienced developers should be able to understand the project’s architecture and conventions more easily in Django, rather than taking over a project created in Flask.

**3- Flexibility and control**

One of Flask’s greatest strengths, is its minimalism and simplicity. No restrictions means that the developer can implement everything exactly as they want it, using a huge range of external libraries and add-ons, making it flexible and extensible. Django on the other hand, with its built-in features and modules, offers far less freedom and control.

**4- Community**

One of the main advantages of Django, is that it has a huge active developer community. This means that if you need help, or when it’s time to scale your app, you will have an easier job finding other developers to join in and start contributing, plus a wealth of useful content already in the public domain. The Flask community is currently not as big, and so information may be harder to come by.

**5- Maturity**

Django is a very mature framework, having seen its first release in 2005. This means that it has gathered numerous extensions, plugins and third party apps covering a wide range of needs. Flask by comparison, is much younger, having been introduced in 2010, so doesn’t have quite the same range of options available to it.

**6- Which framework is best for your app**

When deciding which framework will perform better for your app, as well as each framework’s pros and cons, it is important to consider the individual needs of your project:

**7- Size of the project**

The size of your project is a good starting point when selecting a framework. Flask is more suited to smaller, less complicated applications, while Django is designed for larger, more complex, and high-load applications. The future growth plans of your project should also be factored in. While Flask is flexible and highly extensible, you will need to be prepared to create most of the functionality yourself. Django, on the other hand will be easier to scale as it is fully featured out of the box.

**8- Structure**

The structure of your app may dictate which framework you should use. If you have a need for particular tools and libraries, or want a highly customizable framework, Flask will be the best choice. If, however, you don’t require the granular tuning that Flask allows, Django will save you a lot of time and money in development hours with its myriad of built-in features and reasonable defaults.

**TYPE OF PROJECT**

Certain types of apps lend themselves better to each framework:

* An online store would be well served by Django, as the framework includes e-commerce modules and the ORM possibilities should be sufficient.
* Online media would be suited to Django as this is the purpose for which the framework was originally created.
* A simple app with static content could be created with Flask as it provides all the functionality needed and allows vast scope for customization, whereas a complex app with dynamic content would be better developed using Django.

**Wrapping up**

When comparing Flask and Django, the aim is not to declare that one framework is better than the other. Rather, it is to examine the key points that will enable you to make an informed decision, against the needs of your individual project.

Both frameworks are suitable for rapid development of web apps, although Django is considered to have the steeper learning curve. Flask provides the most flexibility in terms of customization, however Django has a larger existing community of users and so offers more support. Django is also the older of the two frameworks which means that a wider choice of third party apps, plugins, and extensions are available to developers.

The main thing to remember when choosing a framework, is that they are not a one-size-fits-all solution; the individual project needs should take precedence. Consider the size of the app you wish to build plus plans for future growth, how much customization will be required, and the type of app you are building.