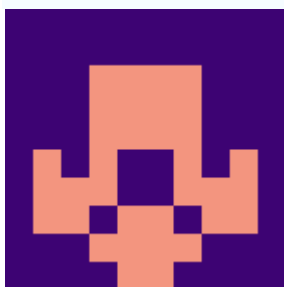
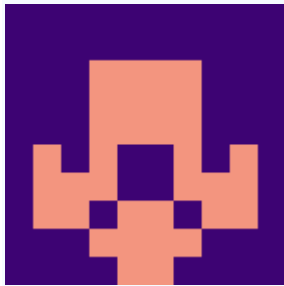


Framework



+1

Published Apr 1, 2022 • **Updated Apr 2, 2023**

[Contribute to Docs](#)

Frameworks are tools that provide reusable boilerplate code and functionality for building applications. They are commonly associated with a specific programming language and are suited to different types of tasks.

Using frameworks can save time and reduces the risk of hard-to-find errors. Plus, frameworks have already been tested, so there's less to worry about. Other advantages include:

- Boilerplate code that is clean, secure, and scalable.
- Better tests and debugging methods.
- The ability to customize the code for a specific project.

Unlike libraries, which are pieces of reusable code, frameworks provide supporting structures that the code must follow.

Types of Frameworks

Frameworks can be used for developing many types of software including the following.

Web Application Frameworks

These frameworks are used for developing any software for the Web, including websites, web applications, and Web APIs. A few common examples of web frameworks include:

- [Angular](#), an open-source [TypeScript](#)-based framework.
- Django, a [Python](#)-based framework that is fast, secure, and has a great admin interface.
- [Rails](#), a full-stack framework written in [Ruby](#).
- Gatsby, a [React](#)-based framework used for building static websites like landing pages, portfolios, and blogs.

Mobile Development Frameworks

These frameworks are used for building applications that can be used on various mobile devices including phones, tablets, and wearables. A few common examples of mobile frameworks include:

- [Flutter](#) is an open-source framework supported on iOS and Android and features customizable widgets.
- [React Native](#), a framework that is cross-platform and written in [JavaScript](#).
- [NativeScript](#), an JavaScript framework that features robust backend-end support.