What is a Framework?

Since they are often built, tested, and optimized by several experienced software engineers and programmers, software frameworks are versatile, robust, and efficient.

Using a software framework to develop applications lets you focus on the high-level functionality of the application. This is because any low-level functionality is taken care of by the framework itself.

Why do we use Frameworks?

Developing software is a complex process. It necessitates a plethora of tasks, including coding, designing, and testing. For only the coding part, programmers had to take care of the syntax, declarations, garbage collection, statements, exceptions, and more.

Software frameworks make life easier for developers by allowing them to take control of the entire software development process, or most of it, from a single platform.

Advantages of using a software framework:

* Assists in establishing better programming practices and fitting use of [design patterns](https://hackr.io/tutorials/learn-software-design-patterns)
* Code is more secure
* Duplicate and redundant code can be avoided
* Helps consistent developing code with fewer bugs
* Makes it easier to work on sophisticated technologies
* One could create their software framework or contribute to open-source frameworks. Hence, there is a continuous improvement in the functionality
* Several code segments and functionalities are pre-built and pre-tested. This makes applications more reliable
* Testing and debugging the code is a lot easier and can be done even by developers who do not own the code
* The time required to develop an application is reduced significantly

What goes in a Framework?

When you install a software framework, the first thing that you need to take care of is the system requirements. Once a framework is installed and configured, it creates a directory structure.

For example, fig. (i) illustrates the directory structure of the [Laravel Framework](https://laravel.com/" \t "_blank). Each of these folders could have additional directories. Directories can further have files, classes, test routines, templates, and more.

fig. (i)

Difference between a Library and a Framework

Some may assume that a software framework is a collection of libraries just as libraries are a collection of precompiled routines. However, this is not true as not all software frameworks use or depend on libraries.

The difference between a library and a framework is that the latter calls the code. Opposite to this, the code calls the software library. Let us understand this with an example:

curl is a library in PHP. When you use one of the curl functions, the PHP code calls that particular function in the curl library. Your code is the caller, and the library code is the callee.

When you use a[PHP framework](https://hackr.io/blog/best-php-frameworks), such as Laravel, the relationship gets inverted, and so the software framework calls the application code written in the framework. This is technically known as [Inversion of Control (IoC)](https://en.wikipedia.org/wiki/Inversion_of_control).

Programming language vs Frameworks

A programming language tells the computer what it should do. Every programming language features a syntax and a particular set of rules, which need to be followed every time the code is written.

A software framework is built on top of a programming language. For example,

Rails, also known as Ruby on Rails, is a web framework built on top of the Ruby programming language.

Django and Flask are two different web frameworks built on top of the [Python programming](https://hackr.io/blog/python-programming-language) language. Hence, they are also known as [Python frameworks](https://hackr.io/blog/python-frameworks). React, and Angular are front-end web frameworks built on top of the JavaScript programming language.

Types of Frameworks

As a developer, you should be on the lookout for frameworks that best suit your needs. Whether it is working on a website, data science, database management, or mobile applications, software frameworks exist for all genres of software programming.

There are many types of software frameworks to make it easier for developing applications for a wide range of application development domains. Let us dive into some of the software frameworks that are in vogue today:

Web Application Frameworks

1. Angular

Angular is a typescript-based, open-source JS framework that makes it easy to build applications on the web. Angular solves application development challenges by combining declarative templates, dependency injection, end-to-end tooling, and much more.

Angular empowers developers to build applications that live on the web, mobile, and desktop.

The popular [JavaScript framework](https://hackr.io/blog/best-javascript-frameworks) is used in public-facing applications and sites such as Google Cloud Platform and AdWords, as well as many internal tools of Google.

Some popular websites developed using AngularJS are:

* Netflix
* Paypal
* Upwork
* Youtube
* Django

Django is a free and open-source web application framework written in Python. Built by a team of experienced developers, Django takes care of web development so that developers can focus on writing applications without reinventing the wheel.

Large organizations actively use Django in its development. Some popular websites developed using Django are:

* Disqus
* Instagram
* Mozilla
* Pinterest