### **Framework**

A framework is a platform or a collection of programs for developing software applications. It is built on top of a programming language. A programming language is a specified method of communication between the programmer and computer. It provides a foundation on which software developers can build programs for a specific [platform](https://techterms.com/definition/platform). Framework is a set of pre-written code libraries designed to be used by developers. For example, a framework may include predefined [classes](https://techterms.com/definition/class) and [functions](https://techterms.com/definition/function) that can be used to process [input](https://techterms.com/definition/input), manage hardware devices, and interact with [system software](https://techterms.com/definition/systemsoftware). This reduces the pressure on the the development process since programmers don't need to do everything manually each time they develop a new application.

A framework may also include code libraries, a [compiler](https://techterms.com/definition/compiler), and other programs used in the software development process.

Popular framework examples- [ActiveX](https://techterms.com/definition/activex) , .NET for [Windows](https://techterms.com/definition/windows) development, Cocoa for [Mac OS X](https://techterms.com/definition/os_x), Cocoa Touch for [iOS](https://techterms.com/definition/ios), and the Android Application Framework for [Android](https://techterms.com/definition/android). Software development kits ([SDKs](https://techterms.com/definition/sdk)) are available for each of these frameworks and include programming tools designed specifically for the corresponding framework.

Sometimes a software framework is supported only by an [operating system](https://techterms.com/definition/operating_system), like a program written for the Android Application Framework will run on an Android device without requiring other additional files to be installed. However, some applications require a specific framework in order to run. For example, a Windows program may require Microsoft .NET Framework 4.0, which is not installed on all Windows machines (especially on older versions of Windows). In this case, the Microsoft .NET Framework 4.0 installer package must be [installed](https://techterms.com/definition/install) in order for the program to run.

Apple has created several specific frameworks that can be accessed by Mac OS programs. These frameworks are saved with a [.FRAMEWORK](https://fileinfo.com/extension/framework) file extension. Examples of Mac OS frameworks include

AddressBook.framework,

CoreAudio.framework,

 CoreText.framework, and QuickTime.framework.

* **Frameworks are built with different dependencies.** Some frameworks require very specific versions of other libraries, which may be incompatible with the library versions used by another framework. Some frameworks even require specific versions of the language.
* **Frameworks are developed by different people**. In order to consolidate all these projects into a single unified whole, you’d need to turn them into a single organization reporting in at different levels. That’s a lot of added management overhead; too much for what are often open source and personal projects to most contributors.
* **Frameworks develop at a different speed.** If the core language wants to release every six months, but some framework wants to release every two weeks. The language releases more slowly because it takes longer to test, so if you release every two weeks, you’re possibly shipping unstable code. But if you release every six months, you could be needlessly preventing users from making use of new features of that framework.
* **Frameworks can be heavy**. They can pull into tons of dependencies, drivers, applications, and end up costing you a lot of disk space. If all that needs to be packaged into every application, your users will complain.

### **Why should Developer use a framework?**

A framework is not absolutely necessary: it is “just” one of the tools that is available to help a developer develop better and faster!A framework provides developer with the certainty that an application which is being developed is in full compliance with the business rules, that is structured, and that is both maintainable and upgradable.

It allows developers to save time by re-using generic modules in order to focus on other areas.

### **Open Source**

Open source means that frameworks are cost effective for both the customer and the developer. That doesn’t say that they aren’t of good quality. Most of the well-known web frameworks used by developers are free for use.

### **Efficiency**

Frameworks eliminate the need to write a lot of repetitive code which allows the developers to build websites and applications much quicker. A web framework also comes with advanced security features and a support team that you can be rest assured knowing that if an issue arises, it will be handled by that team. And that’s how the efficiency of developers increase.

### **Integration**

This is one of the most useful features that have the ability to permit developers to link with other tools such as databases to the framework.

### **Investing in the task, not in the technology**

This is the basic principle of a framework: Not having to repeat everything and do it manually. And doing low value added tasks (for example, the development of generic components) in order to fully focus on the business rules.

A framework will keep the developer from having to spend 2 or 3 days creating an authentication form (which is not a specific task). The time that is saved can be dedicated to more specific components as well as to the corresponding unit tests; giving high quality code.

### **Guaranteed upgradability and maintenance**

A framework ensures the longevity of your applications. If a development team works as they please, only that particular team will be able to maintain and upgrade the application with ease. On the other hand, the structure that a framework provides for the application makes it possible to avoid this problem altogether and it gives any developer - whether they participated in its development or not – the ability to easily “adopt” an application, to maintain it over time and to upgrade it both quickly and neatly, whenever needed.

### **Advantages of using a software framework:**

### **SUPPORT**

As any other distributed tool, a framework generally includes documents, a support group, or huge community online forums where you can acquire quick responses from experts and other users who have been in that group for a long time. You can get tips on making your application into a huge hit. The testing and debugging can be done pretty quickly.

### **Integration**

For building almost any type of application (including a website) where you want to store some data, you will typically use a database. There also exist many other tools that link to web development. Many frameworks will thus make it easier to link to these tools and also communicate with them

### **EFFICIENCY**

Jobs that generally would take you hours and hundreds of lines of code to compose, can now be done in minutes with pre-built functions. Instead of generating code from scratch, you can save a considerable amount of time if you find another developer who will be willing to share code with you. Development becomes a lot easier, so if it’s much easier it’s quicker, and subsequently effective.

### **SECURITY**

An extensively utilized framework has big security applications. The big benefit is users end up being long-lasting testers. If you find a vulnerability or a security hole, you can go to the framework’s web site and let the team understand so they can fix it. Most frameworks feature about 10-100+ expert developers, it means that many people will lay their hands and minds on your code and identify any security loopholes. They will identify any potential security threats to its overall integrity, and provide insights that will help you fix them before they turn out to be a disaster in the near future

### **EXPENSE**

Most popular structures are complimentary, and considering that it likewise helps the developer to code faster, the expense for the final client will certainly be smaller. Most of the popular frameworks are open-source (or available to use for free). They often also come with licensing that is not restrictive and allows to build a commercial product

### **ROBUSTNESS**

When you develop software by yourself, you might not come up with something as efficient as one that you can develop collectively with help from expert developers. Development frameworks create networks between developers, and this enhances the robustness of the end product. This, in turn, results in highly efficient software that serves the needs of the end user to the maximum.

### **Types of Frameworks**

As a developer, you should be on the search 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 common today:

### **Web Application Frameworks**

Web Application Framework  is a software framework that is designed to support the development of web applications including web services, web resources and web APIs. Frameworks are, in short, libraries that help you develop your application faster and smarter. Web frameworks provide a standard way to build and deploy web applications on the World Wide Web.

### **1. Angular**

Angular is a typescript-based, open-source JS framework by Google (originally developed by Misko Hevery and Adam Abrons) that makes it easy to build applications on the web. It is a framework to build large scale and high performance web application while keeping them as easy-to-maintain. 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.

### **2. Django**

Django is a free and open-source web application framework written in Python that helps in building quality web applications. 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.

### **3. Laravel**

Laravel is a PHP-based web application framework created by Taylor Otwell in 2011 with an expressive, elegant syntax. The open-source framework and follows a model-view-controller design pattern that is robust and easy to understand. One can rightaway start learning and developing Laravel with Laracasts which has hundreds of tutorials in it.

According to Google Trends, Laravel has secured the position for the most powerful PHP framework, which offers a standardized and feature-packed platform for high-performing PHP web application development.

### **4. Ruby on Rails**

Ruby on Rails is an extremely productive web application framework written by David Heinemeier Hansson. One can develop an application at least ten times faster with Rails than a typical Java framework. Moreover, Rails includes everything needed to create a database-driven web application, using the Model-View-Controller pattern.

### **DataScience Frameworks**

A Machine Learning Framework is an interface, library or tool which allows developers to build [machine learning models](https://www.edureka.co/blog/machine-learning-classifier/) easily, without getting into the depth of the underlying algorithms. Let’s discuss the Top 10 Machine Learning Frameworks in detail:

### **1. Apache Spark**

Apache Spark is a unified analytics engine for large-scale distributed data processing and machine learning. You can write applications quickly in Java, Scala, Python, R, and SQL using the Apache Spark.On top of the Spark core data processing engine are libraries for SQL, machine learning, graph computation, and stream processing. These libraries can be used together in many stages in modern data pipelines and allow for code reuse across batch, interactive, and streaming applications. Spark is useful for ETL processing, analytics and machine learning workloads, and for batch and interactive processing of SQL queries, machines learning inferences, and artificial intelligence applications.

### **2. TensorFlow**

TensorFlow is an end-to-end open-source framework for machine learning library developed at Google. It was developed for numerical computation using data flow graphs. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers dive in ML, and developers quickly build and deploy ML-powered applications.It’s handy for creating and experimenting with deep learning architectures, and its formulation is convenient for data integration such as inputting graphs, SQL tables, and images together.

### **3. PyTorch**

PyTorch is an open-source machine learning framework developed by Facebook’s artificial intelligence research group. It accelerates the process from research and prototyping to production deployment. It is the primary software tool for deep learning after Tensorflow. It allows you to make changes to the architecture in the process. PyTorch can be used with Python as well as C++. PyTorch is used for Computer Vision and Natural Language Processing (NLP).

### **4.Scikit-learn**

[**Scikit-learn**](http://scikit-learn.org/stable/) is a very popular open-source machine learning library of algorithms, with the goal of providing a set of common algorithms to Python users through a consistent interface. With constant updations in the product for efficiency improvements coupled with the fact that its open-source makes it a go-to framework for machine learning in the industry. It’s quickly becoming a go-to framework for machine learning, as it’s constantly [**evolving**](http://scikit-learn.org/0.16/whats_new.html) with new models, efficiency improvements on speed and memory, and large data capabilities

### **Mobile Development Frameworks**

A mobile development framework is a software framework that is designed to support mobile app development. It is a software library that provides a fundamental structure to support the development of applications for a specific environment.

### **1. Ionic**

Ionic is a free, open-source mobile UI toolkit for developing high-quality, cross-platform native applications for Android, iOS, and the Web—all from a single codebase.It is usually a client-side framework that helps in building native-based apps with a combination of HTML, CSS3, and JavaScript.The element of HTML5 helps in building hybrid apps as well. There is no doubt that it is regarded as one of the best assets for the development of the Progressive Web Apps.Ionic is a development platform for the entire application lifecycle that allows teams to build better and faster applications.

### **2. React Native**

It is one of the other popular cross-platform frameworks that have been launched by social media king Facebook in 2013. React Native brings what developers are used to from React on the web — declarative self-contained UI components and fast development cycles — to the mobile platform, while retaining the speed and feel of native application. React Native is basically an open-source framework offering ample support to the IDEs and the other mobile app development tools.

### **3. Xamarin**

Xamarin is a free, open-source application development platform introduced by Microsoft for building Android, iOS applications with .NET, and C#. Xamarin is part of the [.NET platform](https://dotnet.microsoft.com/platform/open-source) that has an active community of over 60,000 contributors from more than 3,700 companies. You can even share these codes with the help of Windows and Mac OS platforms. Since it allows the benefit of code sharing, the developers can save a lot of their time.

### **4. Flutter**

Flutter is Google’s UI toolkit for building beautiful, natively compiled applications for mobile, web, desktop from a single codebase. It is an open-source mobile application development SDK.. It has an expressive and flexible UI and gives a native performance on iOS and Android platforms. It is a popular cross-platform app development and is written in the Dart language.

However, when it comes to the choice of framework, it is best suited for the development of the hybrid apps. There is no need to restart your project if there is any mistake.

A framework is not an absolute necessity, but nonetheless it is very useful. It is a pledge of quality, upgradability and maintainability of applications at lower cost. Applications developed using a framework are interoperable with market standards. It is definitely good to use frameworks initially to get a hang of it, then you will have times where you need to edit the frameworks to suit your needs. Then you might feel like writing your own code from scratch. It all depends on the project you are working on.

Hope this article was helpful