

The Difference Between a Framework and a Library



Last updated: December 23, 2022



Written by: Sandip Roy

(<https://www.baeldung.com/cs/author/sandiproj>)

Core Concepts (<https://www.baeldung.com/cs/category/core-concepts>)

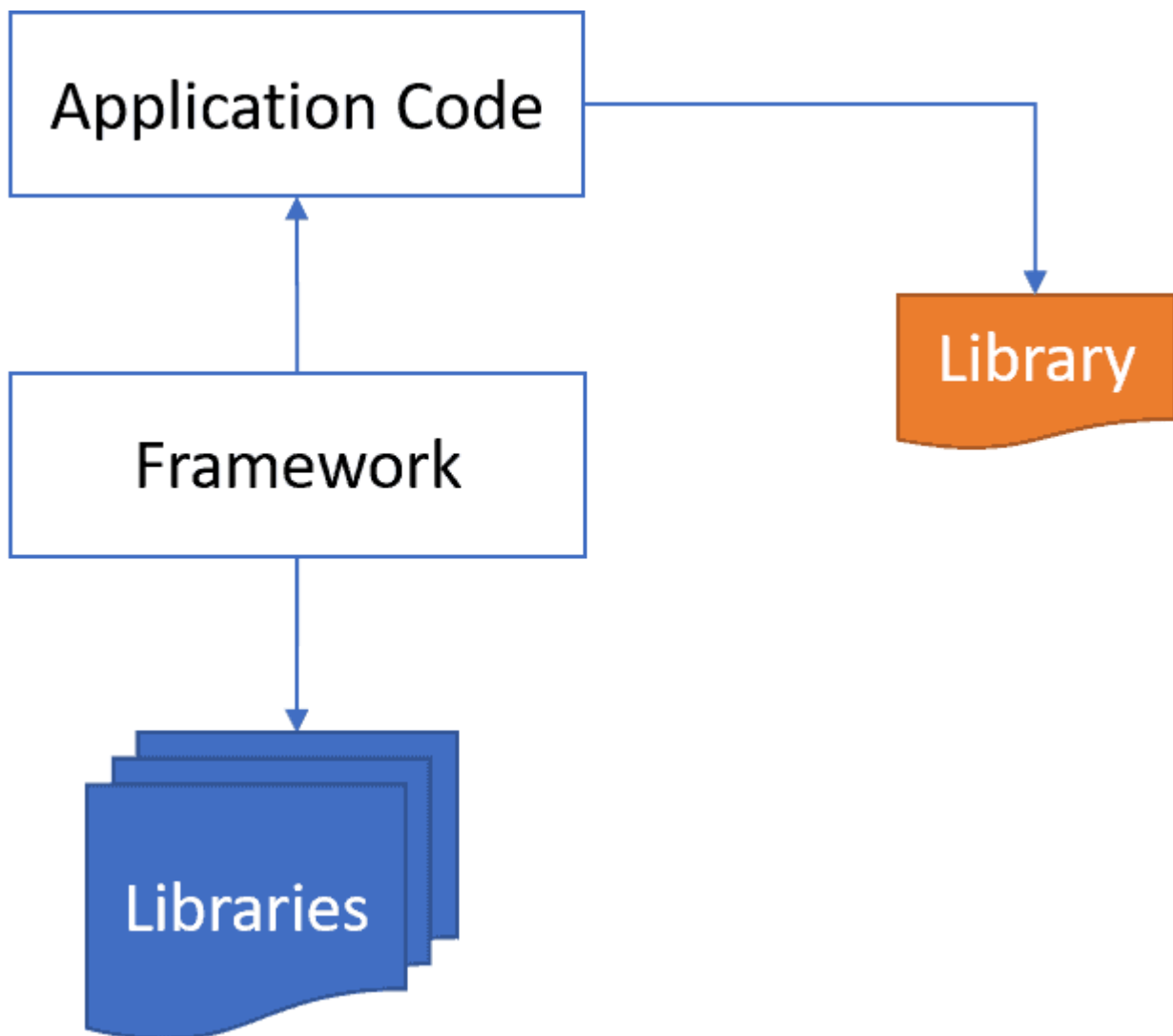
1. Introduction

Developers generally use the terms “library” and “framework” interchangeably. In this tutorial, we'll learn the differences between them, as well as explore various aspects of each. **We can say that a library implements a particular function.** Some examples of popular libraries are

React (<https://reactjs.org/>), and JQuery (<https://jquery.com/>). **We can define a framework as a collection of libraries implementing a particular methodology.** Some of the more commonly known frameworks are Angular, (<https://angular.io/>) Vue (<https://vuejs.org/>), Model View Controller ([/spring-mvc](https://spring-mvc)), and Model View Presenter ([/mvc-vs-mvp-pattern](https://spring-mvc)).

2. Inversion of Control

Let's consider the technical difference between a framework and a library due to the inversion of control:



The above diagram shows that when an application code uses a library, the developer writing the code is in charge of the application flow. **This means the developer decides when to call the library. However, when we use a framework, the framework decides when to call the library.** This shift in control of calling the library from the application code to the framework is an inversion of control ([/inversion-control-and-dependency-injection-in-spring](https://spring)).

3. How Does a Framework Work?

A framework enables a coding environment that contains low-level libraries to address conventional coding issues. **The objective of a framework is to deliver faster development of an application.** This includes everything we need to build large-scale applications, such as templates based on best practices. Let's consider the inner workings of a framework. If we're using a graphical user interface (GUI) framework, it calls the code through event handlers

(<http://www.cs.fsu.edu/~jtbauer/cis3931/tutorial/ui/overview/events.html>). If we're using a web framework, it calls the code through a request-response model

(http://books.gigatux.nl/mirror/beaweblogic8.1/0672324873_ch06lev1sec2.html). **For a web application, JavaScript (<https://www.javascript.com/>) frameworks provide a skeleton with the tools for fast development.**

Internally, such a framework contains a large collection of libraries that provides the developer with inbuilt functionalities, which helps to develop an application without extensive coding knowledge.

3.1. Advantages of a Framework

There are several benefits of using a framework in general, such as ease of debugging, improved coding, easy code re-usability, and accelerated development. Let's discuss the advantages of a web development framework:

- **A change in one part of the application doesn't affect the entire application.**
- Provides caching and optimized processes for the network traffic
- **Enables faster methods for web development with less code**
- Supports cross-platform application development
- Provides a superior user experience by creating rich and dynamic content
- Some JavaScript frameworks, such as Angular, are based on MVC, hence the use of **data binding (/spring-mvc-custom-data-binder).**

3.2. Limitations of a Framework

Some of the issues associated with frameworks in general are:

- The ready-to-use features of a framework prevent programmers from gaining an in-depth understanding of the programming language.
- **Options to tweak functionalities are limited.**
- Sometimes application development is complex using a framework.
- We need to choose the right framework for the scale of the application, or else performance and user experience may be adversely impacted.
- Clear separation of business logic from the presentation layer in MVC is sometimes difficult.
- **We have to be up-to-date with new/deprecated features in every version.**

In addition to the above, the issues associated with web development frameworks are:

- Browser version dependencies
- **JavaScript frameworks can only run with a JavaScript-enabled browsing environment.**
- Chance of a security breach if the prescribed framework guideline isn't followed during development

3.3. Use of a Framework

Although the framework is a broad term, its main uses include:

- Web Development – This uses different languages, such as PHP (CodeIgniter, Laravel), Python (Django), and JavaScript (**Angular** ([/spring-boot-angular-web](https://spring-boot-angular-web/))).
- Artificial Intelligence – Popular frameworks include **Apache Spark** ([/java-spark-dataframe-dataset-rdd](https://java-spark-dataframe-dataset-rdd/)), PyTorch, and **Tensorflow** ([/tensorflow-java](https://tensorflow-java/)).
- Mobile App Development – Some of the popular frameworks are **Native Script** (<https://nativescript.org/>), **React Native** (<https://reactnative.dev/>), and **Flutter** (<https://flutter.dev/>). Flutter is Google's UI toolkit for building beautiful, natively compiled applications.

4. How Does a Library Work?

A library is a collection of reusable, compiled, and tested code that can facilitate the automation or augmentation of application functionalities. It's designed to support both the code developer and code compiler during the build process and the running of the application. A **library implements many functions, variables, and parameters.**

4.1. Advantages of a Library

Let's discuss some of the benefits of a library:

- **Improves performance of a program by the selective inclusion of a library by the compiler during run time**
- **Provides reusable functions that can be referred to within the code without defining them explicitly**
- Eliminates the need for writing code for complex functions
- **Prevents us from having to write code to solve the same problem over and over again**
- Focuses only on the feature that the library implements, without worrying about global state management, like HTTP and routing
- **Provides us with the opportunity to pick libraries that we want**
- Reduces application development cost
- **Encourages programmers to focus on configurable and reusable library development**
- Provides pre-tested code for multiple environments and use cases

4.2. Limitations of a Library

Now let's look at some of the limitations of a library:

- Using a library means that our code is tied to that library.
- To change libraries, our code might have to undergo changes to use the new library.
- Using a library in an unsupported environment needs a wrapper, resulting in a performance impact on the application.
- Lack of support by the developer of a library may result in incompatibility issues with the new version of the application.
- Using several libraries might impact the performance of an application adversely because of dependency conflicts.
- Sometimes libraries are vulnerable to malicious attacks.

4.3. Use of a Library

We tell libraries what to do, so they have a vast area of use. Since software development started involving complex functionalities, the need for reusable libraries increased. JQuery is one such example because it makes event handling, animation, and Ajax ([https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming))) simpler, and works across a large number of browsers.

5. Framework vs. Library

There are several differences between frameworks and libraries. Let's look at few important ones:

Framework	Library
Provides ready to use tools, standards, templates, and policies for fast application development	Provides reusable function for our code
The framework controls calling of libraries for our code	Our code controls when and where to call a library
To leverage the benefit of a framework, a fresh application can be developed following the framework's guideline	Library can be added to augment the features of an existing application
Easy to create and deploy an application	Facilitates program binding
Helps us to develop a software application quickly	Helps us to reuse a software function
Intent of a framework is to reduce the complexity of the software development process	Intent of a library is to provide reusable software functionality

6. Conclusion

In this article, we learned the concepts behind frameworks and libraries. A framework is used for building and deploying an application quickly. **When we use a framework, we can use resources to facilitate faster development and a greater user experience.** A library is used to enhance the functionality of an application. **If we develop our own library, we can use the functions in many applications.**

CATEGORIES

[ALGORITHMS \(/CS/CATEGORY/ALGORITHMS\)](#)
[ARTIFICIAL INTELLIGENCE \(/CS/CATEGORY/AI\)](#)
[CORE CONCEPTS \(/CS/CATEGORY/CORE-CONCEPTS\)](#)
[DATA STRUCTURES \(/CS/CATEGORY/DATA-STRUCTURES\)](#)
[LATEX \(/CS/CATEGORY/LATEX\)](#)
[NETWORKING \(/CS/CATEGORY/NETWORKING\)](#)
[SECURITY \(/CS/CATEGORY/SECURITY\)](#)

SERIES

[GRAPHS TUTORIAL \(HTTPS://WWW.BAELDUNG.COM/CS/GRAPHS-SERIES\)](https://www.baeldung.com/cs/graphs-series)
[NEURAL NETWORKS SERIES \(HTTPS://WWW.BAELDUNG.COM/CS/NEURAL-NETWORKS-SERIES\)](https://www.baeldung.com/cs/neural-networks-series)
[LATEX SERIES \(HTTPS://WWW.BAELDUNG.COM/CS/LATEX-SERIES\)](https://www.baeldung.com/cs/latex-series)

ABOUT

[ABOUT BAELDUNG \(/ABOUT\)](#)
[BAELDUNG ALL ACCESS \(/COURSES\)](#)
[THE FULL ARCHIVE \(/CS/FULL_ARCHIVE\)](#)
[EDITORS \(/EDITORS\)](#)
[OUR PARTNERS \(/PARTNERS/\)](#)
[PARTNER WITH BAELDUNG \(/PARTNERS/WORK-WITH-US\)](#)
[EBOOKS \(/LIBRARY/\)](#)
[FAQ \(/LIBRARY/FAQ\)](#)

[BAELDUNG PRO \(/MEMBERS/\)](#)

[TERMS OF SERVICE \(/TERMS-OF-SERVICE\)](#)

[PRIVACY POLICY \(/PRIVACY-POLICY\)](#)

[COMPANY INFO \(/BAELDUNG-COMPANY-INFO\)](#)

[CONTACT \(/CONTACT\)](#)

[PRIVACY MANAGER](#)