LU04a - Framework vs. Library

Learning objectives

1. Definition what a software library is
2. Definition what a framework is
3. Names the differences of both

Introduction

Libraries and frameworks streamline software development by providing reusable code and predefined structures. Libraries focus on specific tasks, enabling modularity, while frameworks establish an overall application architecture, promoting consistency. The choice between them depends on project needs and desired flexibility. Together, they reduce development time, enhance quality, and simplify the integration of advanced features across various platforms.

First, however, it is important to distinguish between the terms. "Library" and 'framework' are often used interchangeably - which, strictly speaking, is wrong. Even if the transition is indeed fluid in many respects, there are fundamental differences between the two.
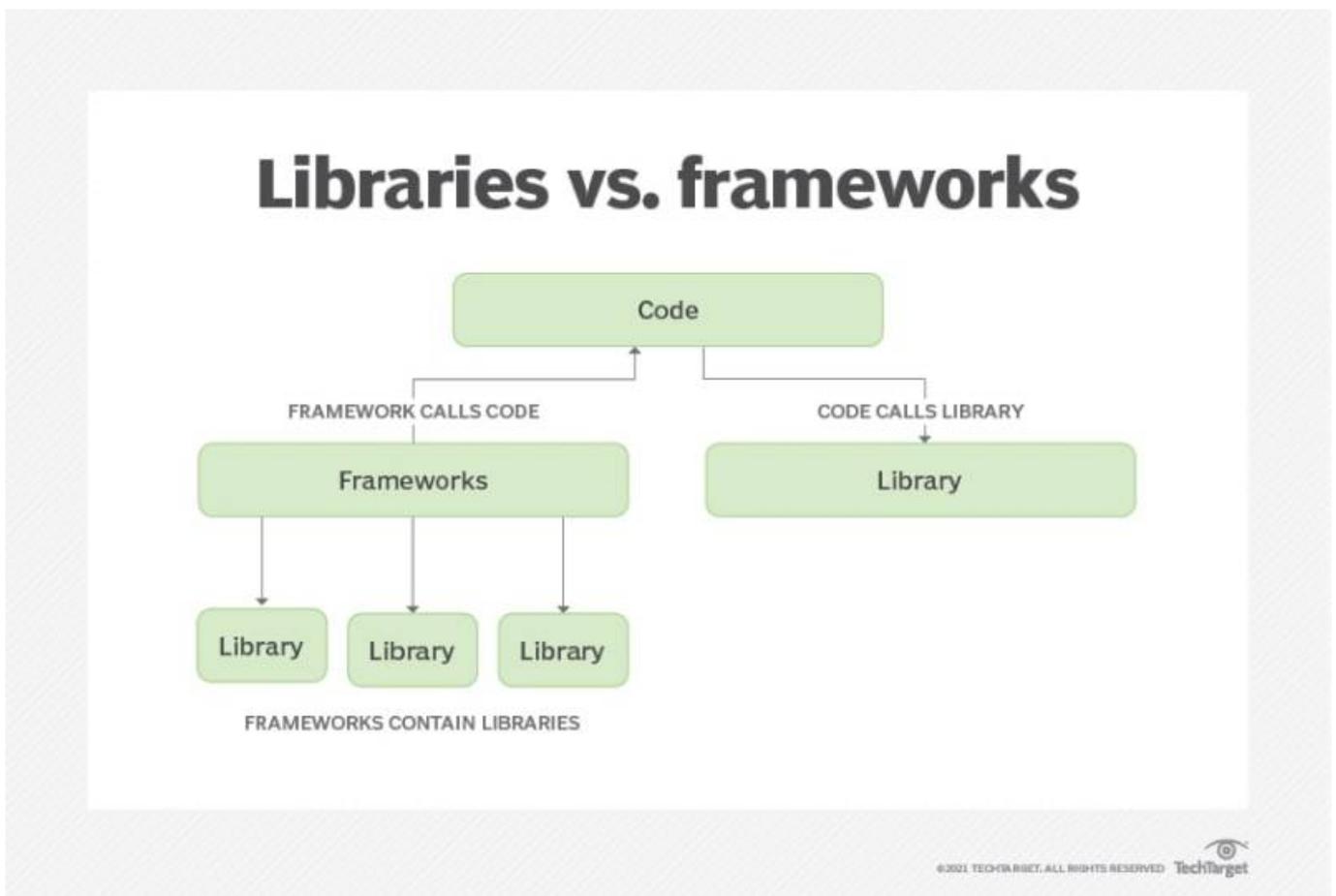
Library

A software library is a collection of pre-written, reusable code that provides specific functionality for developers. Libraries simplify programming tasks by offering modules or functions for common operations, such as handling data, managing files, or creating user interfaces. Unlike frameworks, libraries are less prescriptive, allowing developers to call and integrate them as needed without imposing strict design patterns.

Framework

A framework is a structured platform or foundation for developing applications, offering pre-written code, tools, and guidelines to streamline the process. It provides a standardized way to build and deploy software, saving time and ensuring consistency. Frameworks support specific languages or domains, such as web development (Django, Angular) or data analysis (TensorFlow), enabling efficient and organized coding practices.

Summary

Libraries and frameworks are tools that simplify software development by providing pre-written code. Libraries offer reusable functions for specific tasks, giving developers control over when and how to use them. Frameworks, on the other hand, define a structured architecture, requiring developers to work within their rules. The „who-calls-who" distinction highlights that in libraries, the developer calls the library, while in frameworks, the framework calls the developer's code.

Who-Calls-Who Point-OF-View

There is also another aspect, which claims to explain the difference between frameworks and libraries: The Who-Calls-Who approach.

Who-Calls-Who-Approach

The „who-calls-who" principle is a key distinction between libraries and frameworks, focusing on the flow of control:

1. **Library**: In a library, **you are in control**. Your code explicitly calls the library's functions or methods when needed, and the control is then returned to your code.

2. **Framework**: In a framework, **it is in control**. The framework defines the overall structure, and it calls your code at specific points, such as predefined methods or callbacks, following its architectural rules.

This distinction highlights the differing roles and levels of control in using libraries versus frameworks.

Frameworks vs. Library

Library	Framewor
Library is a set of reusable functions used by computer programs.	Framework is a piece of code that dictates the architecture of your project and aids in programs.
You are in full control when you call a method from a library, and the control is then returned.	The code never calls into a framework; instead, the framework calls you.
It's incorporated seamlessly into existing projects to add functionality that you can access using an API.	It cannot be seamlessly incorporated into an existing project. Instead, it can be used when a new project is started.
They are important in program linking and binding process.	They provide a standard way to build and deploy applications.
Example: jQuery is a JavaScript library that simplifies DOM manipulation.	Example: AngularJS is a JavaScript-based framework for dynamic web applications.

Additional Material

You will find a list with JavaScript-Libraries und Frameworks including an explaination and a rathing as follows: here .