

# LIBRARY

---

특정 기능을 수행하기 위한 클래스 또는 함수의 집합체

library는 주로 소프트웨어를 개발할 때 컴퓨터 프로그램이 사용하는 비휘발성 자원의 모임이다. 여기에는 구성 데이터, 문서, 도움말 자료, 메시지 틀, 미리 작성된 코드, 서브루틴(함수), 클래스, 값, 자료형 사양을 포함할 수 있다.

수학 라이브러리 애니메이션 라이브러리 문자열 라이브러리

소프트웨어 개발 시 사용되는 프로그램의 구성요소로, 공통으로 사용될 수 있는 특정한 기능(들)을 모듈화한 것이다. 대상 환경(플랫폼)에서 바로 실행될 수 있는 형태로 제공되며, 실제 실행되어 기능을 수행하는 단편화된 프로그램이다.

라이브러리는 동작하는 완전한 프로그램이 아닌, 특정한 부분 기능만을 수행하도록 제작된, 컴파일되어 기계어의 형태로 (또는 대상 플랫폼에 따라서는 바이트코드로) 존재하는 프로그램이다. 이 자체로는 사용자가 직접 일반적인 조작으로 실행할 수 없으며, 해당 라이브러리의 기능을 직접 호출하는 프로그램을 실행하거나, 사용자가 해당 라이브러리의 기능을 실행하는 API(Application Programming Interface)를 사용하는 프로그램을 직접 개발하면 실행할 수 있다.

표준 라이브러리 : 특정 언어의 개발 환경에 기본적으로 포함된 것들은 대부분 표준 라이브러리라고 불린다. 기본적인 기능 수행과 더불어 디버깅, 성능측정 등을 위한 별도의 API가 존재한다.

런타임 라이브러리 : 프로그램이 실제 환경에서 실행되기 위해 필요한 모듈들이다. 대부분 위의 표준 라이브러리에서 기능 수행에 필요한 것들만 제공되거나, 스크립트의 실행기 등을 말한다.

# FRAMEWORK

---

“당신이 애플파이를 맨 처음부터 만들고자 한다면, 세상을 먼저 만들어야 할 것이다. If you want to make an apple pie from scratch, you must first create the universe.” —칼 세이건, Carl Sagan

프로그래밍에서 특정 운영 체제를 위한 응용 프로그램 표준 구조를 구현하는 클래스와 라이브러리 모임이다. 간단하게 프레임워크라고도 부른다. 자주 개발자들이 사용할 만한 기능을 미리 모아두고 개발자가 나름대로 기능을 확장, 설계, 변형하면서 사용해 나갈 수 있는 집합체이다. 재사용할 수 있는 수많은 코드를 프레임워크로 통합함으로써 개발자가 새로운 애플리케이션을 위한 표준 코드를 다시 작성하지 않아도 같이 사용된다.

## 어플리케이션 프레임워크란?

어플리케이션 프레임워크란 어플리케이션을 개발할 때 '어떻게 하면 맨손으로 개발하지 않을 수 있을까?' 의 고민으로 부터 시작된 것이다. 맨손으로 시작하지 않기 위해서는 구조화된 '무엇'을 개발자에게 제공해야 하며, 그 '무엇' 이 바로 '어플리케이션 프레임워크' 라고 Xin Chen이 제안한 용어(selling idea). 닷넷, 자바 개발자들은 닷넷프레임워크 와 자바프레임워크 위에서 개발을 한다.

## 어플리케이션 프레임워크 왜 사용하는가?

우리가 개발을 할때 마다 거의 같은 일을 하는 코드들을 개발자 개개인이 매번 같은 일을 맨손으로 만든다면, 그런데 매번 만드는 '맨땅에 헤딩'이 모두 다른 스타일로 만들어진다면 '개발' 과 '관리'는 그야말로 악몽일 것이다.

어플리케이션 프레임워크는 일관된 개발을 지원하고, 경계설정을 강제함으로써 모듈성 Modularity, 재사용성 Reusability, 확장성 Extensibility, 단순성 Simplicity, 유지관리성 Maintainability 을 얻을 수 있다.

비유해보자면, 어떠한 사업을 시작할 때

- 프레임워크 - 국가 (기반시설. 뼈대. 근간. 법, 도로, 시설 등등)
- 프레임워크의 하위 프레임워크 - 정부부처
- API - 국가기관
- 라이브러리 - 직무 전문가
- 프로그래밍 언어 - 국가 통용 언어
- 레퍼런스 문서 - 국가 조직도 및 기능설명서

## Framework Layers

