

프레임워크(Framework)와 라이브러리(Library)의 차이

프레임워크의 정의

원하는 기능 구현에 집중하여 개발할 수 있도록 일정한 형태와 필요한 기능을 갖추고 있는 골격, 뼈대를 의미합니다.

앱/서버 등의 구동, 메모리 관리, 이벤트 루프 등의 공통된 부분은 프레임워크가 관리하며, 사용자는 프레임워크가 정해진 방식대로 class, method들을 구현하면 됩니다.

프레임워크의 장단점

장점

- 가이드를 제공함으로써 프로그램을 체계적으로 관리할 수 있다
- 기본적인 설계와 라이브러리를 제공하여 개발 속도를 향상시킨다
- 코드의 재 사용성이 높고 확장성이 좋다

단점

- 각 프레임워크에 따른 공부가 필요하다
- 자유롭고 유연한 코드를 개발이 불가능하다(프레임워크가 짜놓은 틀 안에서 개발해야 하기 때문에)
- 프로젝트의 용량이 증가한다

라이브러리의 정의

소프트웨어를 개발할 때 컴퓨터 프로그램이 사용하는 비휘발성 자원의 모임, 즉 특정 기능을 모아둔 코드, 함수들의 집합이며 코드 작성 시 활용 가능한 도구들을 의미합니다.

라이브러리의 장단점

장점

- 코드를 재사용하기가 쉽다
- 코드의 내용을 숨겨서 기술 유출을 방지할 수 있다
- 이미 구현되어 있는 기능(함수)들을 사용할 수 있어서 개발 시간이 단축된다
- 컴파일 시간을 단축할 수 있다(라이브러리는 이미 컴파일 되어 있기 때문에 바로 사용가능하다)

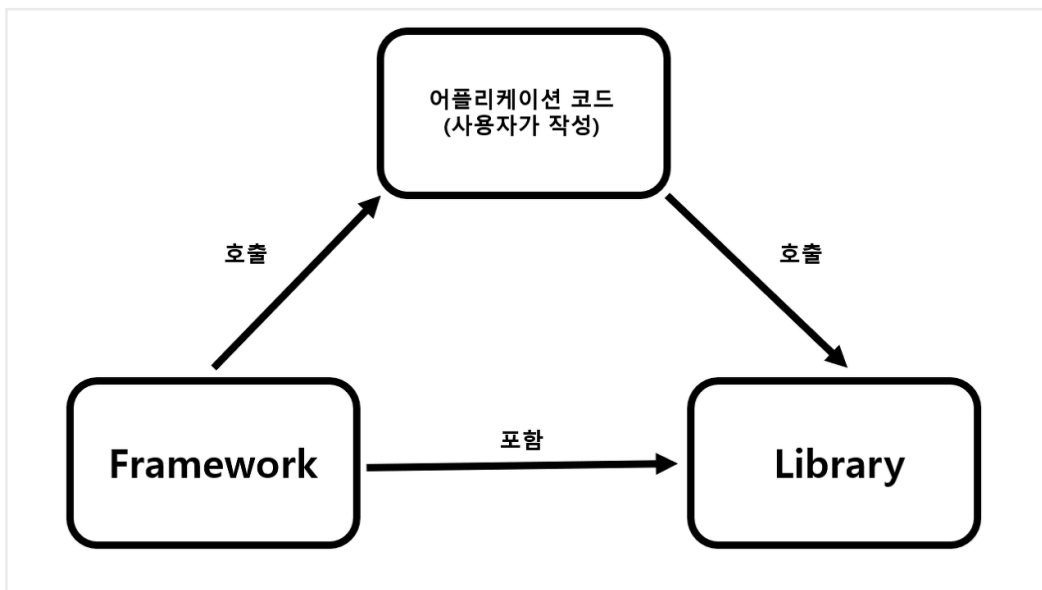
그렇다면 뭐가 다른걸까요?

위의 정의들을 살펴보면 공통점은 개발자가 개발할때 편리하게 사용하려고 사용하는 측면이 존재합니다.

그래서 저 또한 프레임워크와 라이브러리의 차이를 질문 받았을때 대답을 제대로 하지 못했었습니다.

프레임워크와 라이브러리의 가장 큰 차이점은 "제어 흐름"의 권한이 어디에 있는가입니다.

라이브러리는 애플리케이션 코드의 흐름을 직접 제어해야하지만, 프레임워크는 애플리케이션의 코드가 프레임워크에 의해 사용됩니다. 프레임워크가 짜 놓은 뼈대 안에서 수동적으로 동작 해야 하기 때문에 제어의 흐름은 프레임워크가 가지고 있습니다.



참고블로그

<https://eroul-ri.tistory.com/5>

<https://cococon1787.tistory.com/745>

<https://bradbury.tistory.com/224>

