

라이브러리와 프레임워크

비유하자면 프레임워크란 일을 하는 기반이 되는 환경(도구를 포함하는)이고 라이브러리는 각각의 도구다.

예를 들어 자동차를 구매해서 타고 다닌다고 해 보자. 그 자동차를 운전, 유지, 보수하는 데는 그 나름대로의 방식이 정해져 있어서 이를 따라야 한다. 자동차 운용법이라는 거대한 사회적인 규칙에서부터 자동차 자체의 원리에 의한 활용법까지 총체가 바로 프레임워크다. 쉽게 말해, 프레임워크를 어기게 되면 법적으로 문제가 생긴다. 그 문제가 실제로 사고로 이어지느냐에 상관 없이 그렇게 한 행동 자체의 책임이 운전자에게 있다는 점이 중요하며, 적발되면 불이익이 있다. 한편 라이브러리는 자동차 수리에 쓰는 공구나 스페어 타이어에 비유할 수 있겠다. 기본적으로 구매한 자동차에 응급처치 등을 위한 키트가 포함되어 있는 것처럼, 프레임워크에서도 기본적인 도구, 즉 기본 라이브러리를 제공해준다. (순정 타이어가 규격상 불법적일 확률은 거의 없다.) 그러나 법률(프레임워크)에 저촉되지 않는 범위에서 유지, 보수를 위해 더 좋은 도구를 사서 쓴다면, 일을 좀 더 손쉽게 할 수 있을 것이다. (물론 그 도구를 사서 쓰다가 문제가 발생할 수도 있지만, 이것은 법적 책임의 영역이라기보단 도구 선택을 잘못된 것 뿐이다. 해당 프레임워크에서 잘 작동하는 라이브러리인지 체크하고 써야 한다는 얘기.)

특정 라이브러리가 싫다면 그것을 쓰지 않는 것은 꽤나 자유롭다. 대체제도 얼마든지 찾아볼 수 있다. 하지만 프레임워크는 그렇게 쉽사리 옮겨다닐 수 없다. 자동차가 싫다고 해서 자동차 운전 자체를 포기하고 자전거나 기차만 이용하는 것과 같은 이치로, 분명히 그 선택권이 사람에게 있는 것은 맞지만, 대체제를 쉽게 찾을 수 없으며 대개의 경우 분야별로 프레임워크가 이미 정해져있다. (기차를 선호한다고 해서 기차로 자동차 경주에 나갈 수는 없다.)