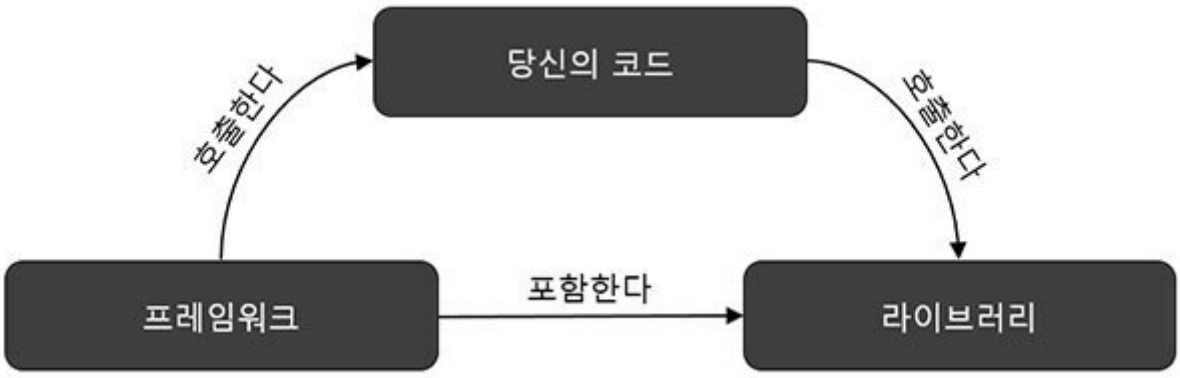
✔️ 2023.04.09 리액트 기초(라이브러리 or 프레임워크, 사용 이유, Virtual DOM)

라이브러리와 프레임워크

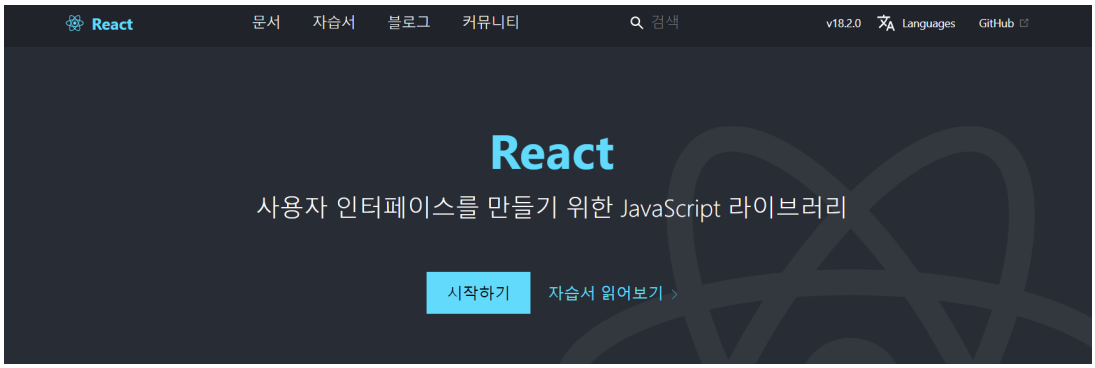
1. **프레임워크(Framework)**는 '구조, 뼈대'를 의미합니다. 프로그래밍에서는 '소프트웨어의 특정 문제를 해결하기 위해서 상호 협력하는 클래스와 인터페이스의 집합'이라고 할 수 있습니다.
2. **라이브러리(Library)**는 '도서관'을 의미합니다. 도서관에서 필요한 책을 골라 내듯이, 프로그래밍에서도 '활용 가능한 도구들의 집합'을 의미합니다.
3. 프레임워크 vs 라이브러리
    a. 프레임워크와 라이브러리는 **제어의 흐름에 대한 주도권**이 어디에 있는지를 파악하면 구분할 수 있습니다.
    b. 프레임워크는 미리 짜여진 구조에 맞게 사용자가 코드를 작성합니다. 전체적인 흐름을 사용자가 아닌 프레임워크가 가지고 있습니다. 이를 **제어의 역전**이라고 합니다.
    c. 라이브러리는 사용자가 필요할 때마다 호출하여 사용합니다. 흐름을 사용자가 가지고 있습니다.



4. 리액트는 프레임워크인가?

a. 위 정의에 따르면 리액트는 프레임워크처럼 보입니다. 미리 짜여진 구조에 맞게 사용자가 코드를 작성해야 하기 때문입니다. 특히, CRA를 사용하지 않고 앱을 빌드한 경험이 없다면 더욱 프레임워크에 가깝게 느껴질 것입니다. 또한, 모던 SPA 구축을 위한 프레임워크를 논할 때 주로 Vue, Angular와 함께 언급되고는 합니다.

b. 그러나 리액트 공식 홈페이지에서는 리액트를 아래와 같이 라이브러리라고 소개하고 있습니다.



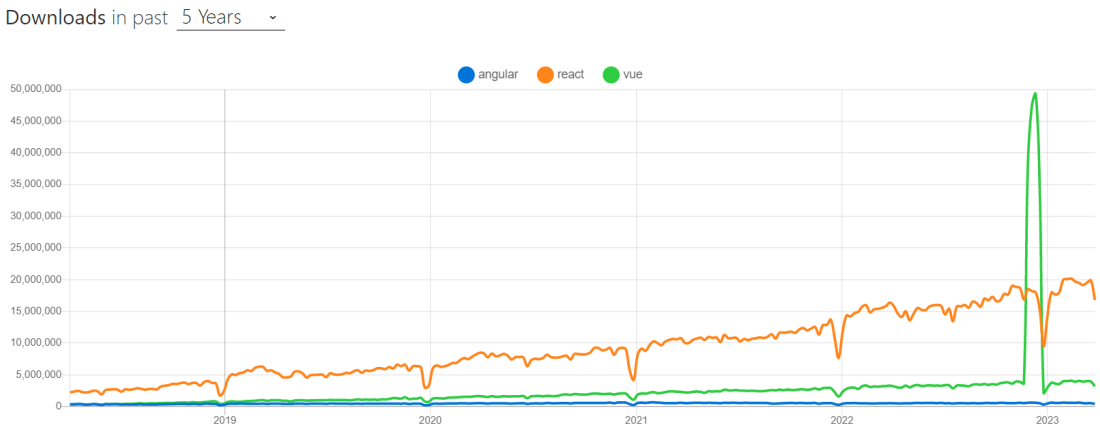
c. 리액트가 라이브러리인 이유에 대해 공식적으로 메타에서 밝힌 바는 없습니다만, 리액트는 프로젝트를 위해 필요한 모든 툴셋 대신, UI와 관련된 일부 도구만을 포함하고 있다는 점이 그 이유로 거론되고는 합니다.

d. 리액트를 프레임워크로서 사용하기 위해 기본적으로 제공되는 것이 create-react-app이며, Next.js 또한 리액트를 포함하여 SSR을 지원하기 위한 프레임워크로 분류됩니다.

e. 리액트가 라이브러리이기 때문에, 비교군인 Vue나 Angular보다 유연하고 개발자의 취향에 맞게 사용이 가능합니다. 다만, 리액트 그 자체로는 지원되지 않는 기능이 많기 때문에 완전한 프로젝트를 구축하기 위해서는 서드 파티 라이브러리에 대해서도 알아야 할 필요성이 생겨 러닝 커브가 높다고 할 수 있겠습니다.

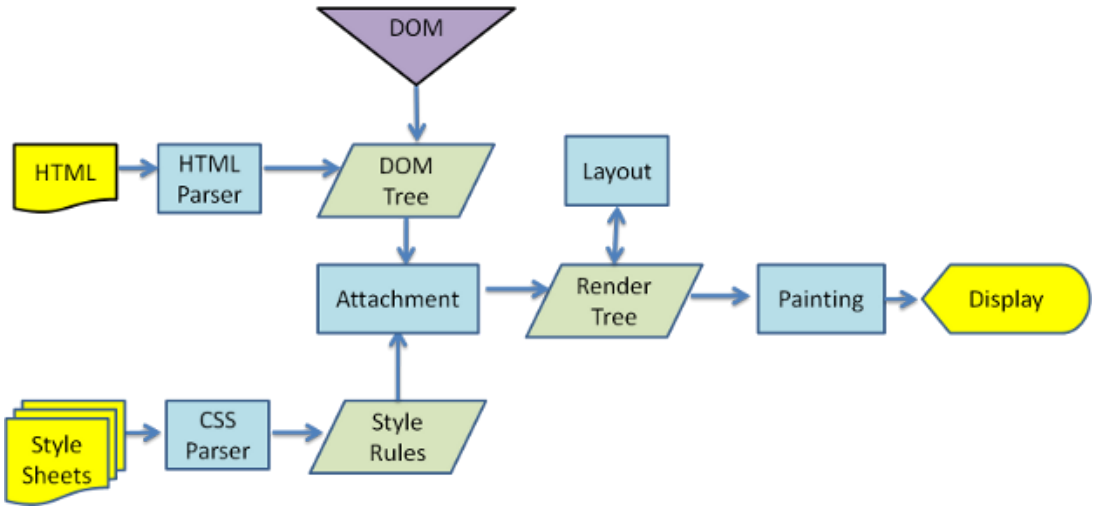
리액트를 사용하는 이유

1. 페이지 전체를 새로고침하지 않고 변하는 부분만을 리렌더링하여 사용성을 향상시켜 줍니다.
2. 컴포넌트 단위로 개발이 가능하여 가독성, 확장성, 재사용성을 향상시켜 줍니다.
3. 라이브러리이므로 다른 라이브러리 및 프레임워크와 함께 사용할 수 있습니다.
4. 현실적인 이유로는 가장 많이 사용되는 SPA 툴이라는 점이 있습니다.



Virtual DOM

1. DOM을 직접 조작하면 안 되나요?
    a. 리액트를 사용하면서, 직접적인 DOM 조작을 피하라는 이야기를 자주 듣습니다. Virtual DOM을 사용해 렌더링 속도를 향상시키는 것이 리액트를 사용하는 주된 목적이기 때문에, DOM을 직접 조작하면 리액트를 사용하는 의미가 없어지기 때문입니다.
    b. 개인적으로 개발의 편의성만을 생각하면 JQuery만큼 편리한 것이 없다고 생각하는데, 이는 DOM을 직접 조작하는 것을 편리하게 만들어 주는 도구이므로 성능 면에서는 많이 떨어지죠. 대부분의 프로젝트에서 이제는 더 이상 사용하지 않는 이유이기도 합니다.
2. DOM을 직접 조작하면 안 되는 이유
    a. 브라우저는 이전에 학습한 바와 같이, 아래의 순서에 따라 동작합니다. (HTML과 CSS로부터 각각 DOM Tree와 CSSOM Tree를 생성하여 결합(Attachment)하고, 이를 통해 Render Tree를 생성합니다.)



b. DOM을 조작하면, 브라우저는 **매번** 스타일을 계산하고 실제 화면에 렌더링하는 작업을 수행합니다.

c. 그런데, Render Tree를 생성하는 과정에서, **DOM의 모든 요소의 스타일이 계산**됩니다.

d. 즉, DOM이 조작될 때마다 **변화하지 않는 요소를 포함한 모든 요소의 스타일이 다시 계산**되어 불필요한 비용이 발생하는 것입니다.

3. Virtual DOM의 역할

a. Virtual DOM은 말하자면 **오프라인 DOM**과 같습니다. 먼저 오프라인 DOM에서 연산을 수행하고, **최종적인 변화를 실제 DOM에 한 번만 전달**합니다.

b. 위 과정은 사실 직접 수행할 수도 있습니다. 변화가 발생할 때, DOM fragment에 해당 변화를 전달하고, 이후 기존 DOM에 이를 보내면 된다고 합니다. 그러나 이를 직접 시도하면 기존 요소들 중 **어떤 것들이 변화되었고 어떤 것들이 변화되지 않았는지를 항상 파악**하고 있어야 합니다. Virtual DOM은 이러한 작업을 자동으로 해주는 역할 또한 수행하고 있습니다.

※ 번외: Shadow DOM

a. Virtual DOM과는 다르게, 컴포넌트를 캡슐화하는데 특화된 개념입니다. 글로벌 스타일이 적용되지 않는 것이 특징입니다.