

FrontEnd 라이브러리가 필요한 이유?

요즘의 웹은 단순히 웹 페이지가 아니라, 웹 애플리케이션이다. 유저인터페이스를 동적으로 나타내기 위해서는 정말 많은 상태를 관리 해주어야한다.

Javascript로 직접 구현하는 것도 가능한 하지만 프로젝트 규모가 커지고, 다양한 유저 인터페이스와 인터랙션을 제공하게 된다면, 많은 DOM 요소들을 직접 관리하고 코드를 정리 하는 것은 힘든 일이다. 웹 개발을 하게 될 때, DOM 관리와 상태 값 업데이트 관리를 최소화 하고, 오직 기능 개발, 그리고 사용자 인터페이스를 구현하는 것에 집중 할 수 있도록 하기 위해서 여러 라이브러리들 혹은 프레임워크들이 만들어졌다. 대표적으로 Angular, Vue, React , Ember, Backbone 등이 있다.

웹 사이트를 구축할 때 사용하는 대표적인 자바스크립트 UI라이브러리는

1) AngularJS

- : 구글에서 만든 오픈소스 프로젝트로 자바스크립트 기반의 웹 개발 프레임워크
- : 2010년에 출시
- : 2018년에 LTS모드 진행되다 2022년 1월 LTS마져 중단선언. 공식적인 지원이 종료

2) ReactJS

- : 메타(구 페이스북)에서 만든 오픈소스 자바스크립트 UI 라이브러리
- : 2013년 처음 출시되어 사용률이 점점 증가하고 있음

3) VueJS

- : Evan You라는 중국인 개발자 한명이 시작한 오픈소스 프로젝트
- : 2014년 처음 출시
- : 현재 React와 함께 항상 언급되는 자바스크립트 대표 프레임워크이다.

일반적으로 Angular와 Vue는 프레임워크라 하고 React는 라이브러리라고 한다.

React 공식 사이트 - <https://react.dev/>



“웹과 네이티브 사용자 인터페이스를 위한 라이브러리”

라이브러리란 자주 사용되는 기능을 정리해서 모아 놓은 것으로 React는 대표적인 자바스크립트 UI라이브러리라고 한다.

React는

- 1) 사용자와 웹사이트의 상호작용을 돕는 인터페이스를 만들기 위한 자바스크립트 기능 모음집이다.
- 2) 복잡한 사이트를 쉽고 빠르게 만들고 관리하기 위해 만들어 졌다
- 3) Single Page Application를 쉽고 빠르게 만들 수 있도록 해주는 도구이다.

“SPA란 하나의 페이지(HTML파일이 하나)만 존재하는 웹 애플리케이션으로 처음에는 HTML의 `<body>`태그의 내부가 텅 비어 있다가 해당페이지에 접속할 때 그 페이지에 해당하는 콘텐츠를 가져와서 동적으로 `<body>` 태그 내부를 채워 넣는 것으로 이러한 부분을 React가 담당 해준다.”

React 장점

1) 빠른 업데이트와 렌더링 속도

내부적으로 **Virtual DOM(가상DOM)**이라는 것을 사용해서 화면을 빠르게 업데이트 한다.

Virtual DOM은 웹페이지와 실제 DOM사이에서 중간 매개체역할을 한다.

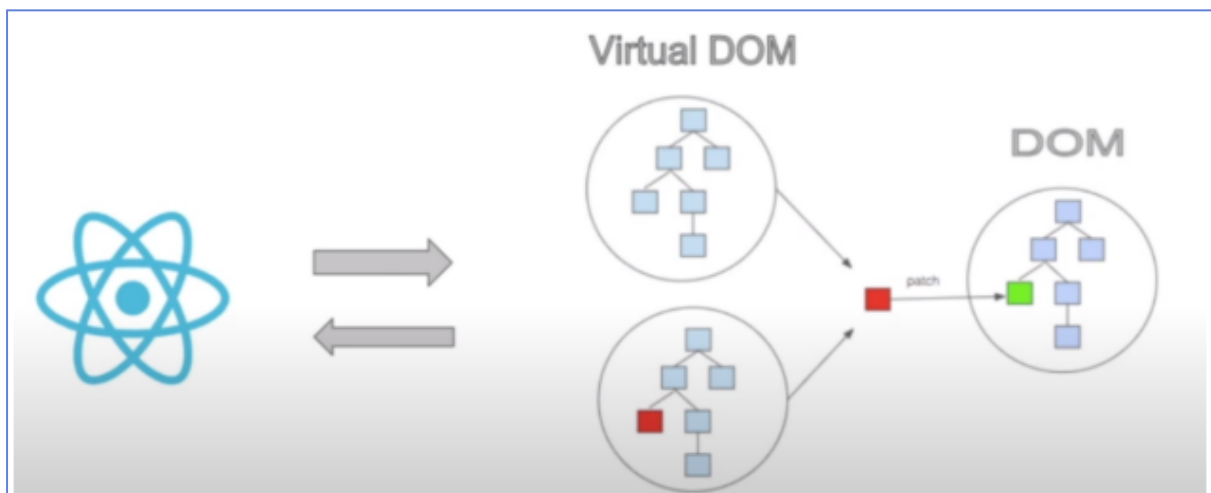
React는 DOM을 직접 수정하는 것이 아니라 업데이트 해야 할 최소한의 부분만 찾아서 업데이트 한다. DOM(Document Object Mode)은 웹 페이지를 정의하는 하나의 객체로 하나의 웹사이트에 대한 정보를 모두 담고 있는 큰 그릇과 같다.

Virtual DOM이란?(<https://www.youtube.com/watch?v=BYbgopx44vo>)

Virtual DOM은 가상의 DOM 이다. 변화가 일어나면, 실제로 브라우저의 DOM에 새로운 것을 바로 넣지 않고, 자바스크립트로 이루어진 가상 DOM 에 한번 렌더링을 하고, 기존의 DOM과 비교를 한 다음에 변화가 필요한 곳에만 업데이트를 해준다. 이 Virtual DOM 을 사용함으로써, 데이터가 바뀌었을 때 업데이트 할 지를 고려하는게 아니라, 그냥 일단 바뀐 데이터로 일단 그려놓고 비교를 한 다음에, **바뀐 부분만 찾아서 변경 시켜 준다.**

Virtual DOM 동작순서를 다시 정리하면,

1. 변화가 일어났다.
2. 변화된 버전을 Virtual DOM으로 바꾼다.
3. 데이터가 업데이트 되면 전체 UI를 Virtual Dom에 Re렌더링한다.
4. Virtual DOM 끼리 비교하자. 변화 전의 Virtual Dom 과 변화된 Virtual Dom 을 비교한다.
5. 비교하여 바뀐 부분만 적용하자. 바뀐 부분만 실제 DOM에 적용을 함으로서 레이아웃 계산은 한 번만 이행된다



Virtual DOM을 정리하면

리액트의 가상돔은 실제 돔과 같은 내용을 담고 있는 객체 복사본이다.

그리고 이 복사본은 자바스크립트 객체 형태로 메모리상에 저장된다.

리액트는 항상 두 개의 가상돔을 갖고 있으며 첫 번째 가상돔은 변경 이전의 내용을 담고 있고 두번째 가상 돔은 변경 이후에 보여질 내용을 담고 있다.

변경된 내용이 화면에 새롭게 그려지기 이전, 곧 실제 돔이 변경되기 이전에 리엑트는 두 개의 가상 돔을 비교해서 정확히 어떤 부분이 바뀌었는지 효율적으로 비교하여 파악한다. 그리고 이러한 과정을 **Diffing** 이라고 한다.

Diffing을 통해서 변경된 부분들을 파악 한 이후에 리엑트는 **Batch Update**를 수행 함으로 실제 돔에 한번에 적용 시켜준다. 이러한 과정을 **Reconciliation**(재조정)이라고 한다.

2) 컴포넌트 기반의 구조

React에서는 모든 페이지가 컴포넌트(구성요소)로 구성된다.

하나의 컴포넌트는 또 다른 여러개의 컴포넌트 조합으로 구성될 수 있다.

마치 레고블록들이 모여서 하나의 완성된 모형이 되는 것과 비슷하다.

3) 재 사용성

재 사용성이란 다른 모듈에 의존성을 낮추고 호환성 문제가 발생하지 않도록 개발하여 유지보수가 용이하게 만드는 것으로 React는 컴포넌트를 개발할 때 항상 쉽고 재사용이 가능한 형태로 개발한다.

4) 든든한 지원군

전세계 최대 IT기업 중 하나인 메타(구 페이스북)에서 시작한 프로젝트로 꾸준히 버전 업데이트가 이루어지며 발전하고 있다.

큰 스폰서를 갖고 있는 만큼 꾸준히 성장하고 유지되고 있다.

5) 활발한 지식공유 와 커뮤니티

현재 React가 영향력을 가지게 된 이유는 개발생태계와 커뮤니티 덕분이라 할 수 있다.

굉장히 큰 개발자 커뮤니티를 형성하고 있고 오픈소스 플랫폼인 깃허브 React프로젝트에 대한 관심이 높다.

생태계 규모를 판단하는 지표 stackoverflow(<https://stackoverflow.com/>)에 react가 개발자로부터 엄청난 관심을 받고 있다. 이것은 실제로 React를 사용하는 개발자가 많다는 것으로 도움을 청할

커뮤니티가 아주 잘 구축되어 있다는 뜻이다.

<https://github.com/facebook/react>

<https://stackoverflow.com/tags>

스택오버플로는 개발자 간의 개발과 관련된 질문과 답변을 공유하는 커뮤니티

6) 모바일 앱 개발 가능

리액트네이티브라는 모바일 환경 UI프레임워크를 사용하여 모바일 앱 개발 가능하다.

React의 단점

1) 방대한 학습량

기존과 다른 방식의 UI라이브러리이기 때문에 배워야 할게 많다.

2) 높은 상태관리 복잡도

React는 state라는 중요한 개념을 이용하여 상태관리를 한다.

React는 Virtual DOM이 바뀐 부분만을 찾아서 업데이트를 하게되는데, 이 때바뀐 부분이라는 것이 state가 바뀐 컴포넌트를 뜻한다. 이것은 컴포넌트를 다시 호출하는 것으로 성능에 저하를 가져온다. 성능 최적화를 위해 state를 잘 관리하는 것이 매우 중요하다.

규모가 큰 프로젝트에서는 상태관리를 위해 Redux, MobX, Recoil등의 외부 상태관리 라이브러리를 사용하기도 한다.

정리하면 , React의 장점은

- 1) 빠른 업데이트와 렌더링 속도
- 2) 컴포넌트 기반의 구조
- 3) 재 사용성
- 4) 든든한 지원군
- 5) 활발한 지식공유 와 커뮤니티
- 6) 모바일 앱 개발 가능