

1. 리액트를 배우는 이유

① 컴포넌트 기반 구조

- UI를 작은 단위의 컴포넌트로 나누어 개발할 수 있어, 코드의 재사용성과 유지보수가 뛰어납니다.

② 빠른 렌더링과 성능

- Virtual DOM을 활용해 실제 DOM 조작을 최소화하여, 빠르고 효율적인 화면 업데이트가 가능합니다.

③ React Native로 모바일 앱 개발 확장

- 리액트의 개념을 그대로 사용해 모바일 앱(Android, iOS) 개발도 할 수 있습니다.

④ 풍부한 생태계와 커뮤니티.

⑤ 대기업 및 다양한 프로젝트에서 사용

2. DOM이란 무엇일까?

춤추는 로봇을 만든다고 가정할때,

로봇의 도면(HTML) → 공장(브라우저)

① 공장(브라우저)에서는 도면(HTML)을 받아서 로봇(DOM)을 생성

② 도면(HTML)인해 로봇(DOM)으로 실체화됨

③ JS(자바스크립트)로 로봇(DOM)에게 명령을 내려 춤을 추게 만든다

실제 동작 과정

HTML → 사용자 화면 (X)

- HTML이 단순히 화면을 그리는 것이라면, DOM은 필요가 없다

HTML → JS → 사용자 화면 (O)

- 웹페이지는 JS에 의해 내용이 변경되거나 요소들에 기능을 부여한다
- JS로 움직일 수 있는 DOM이 필요함

※ 브라우저는 HTML 문서를 읽어 낸 다음 **트리구조로 DOM이라는 실체를 만들어낸다** (parsing 과정)

3. CSSOM이란 무엇일까?

HTML로 부터 DOM이 만들어지듯, CSS로부터는 CSSOM이 만들어진다

CSSOM 역시, 트리구조로 만들어진다

4. 브라우저 (DOM + CSSOM) = Render Tree

브라우저는 DOM과 CSSOM을 결합하여 Render Tree란 것을 만들어 냅다

Render Tree는



DOM과 CSSOM은 JS로 작성된 정보로 **webAPI**의 일부이다
webAPI는 브라우저들이 제공하는 표준화 되어 있는 기능들이다

[정리]

HTML,CSS → DOM,CSSOM → 사용자 화면 출력

JS를 통해서 DOM이나 CSSOM에 접근하여 동적요소 및 기능을 부여한다