




24.03.22

```
name: my-app ,
"version": "0.1.0",
"private": true,
"homepage": ".",
"dependencies": {
  "@testing-library/jest-dom": "^5.17.0",
  "@testing-library/react": "^13.4.0",
  "@testing-library/user-event": "^13.5.0",
  "react": "^18.2.0",
  "react-dom": "^18.2.0"
}
```




React

The library for web and native user interfaces

리액트란?

- 사용자 인터페이스를 만들기 위한 JavaScript 라이브러리
- 사용자와 웹 사이트의 상호작용을 돕는 인터페이스를 만들기 위한 자바스크립트 기능 모음집
- 2013년에 페이스북에서 발표한 오픈소스 자바스크립트 프레임워크
- 가상 DOM(Document object model)과 JSX(JavaScript XML) 방식으로 동작하는 프레임워크
- SPA(Single Page Application) 개발을 위한 프레임워크
- [사용자 인터페이스를 구축하기 위한 선언적이고 효율적이며 유연한 JavaScript 라이브러리](#)
- "컴포넌트"라고 불리는 작고 고립된 코드의 파편을 이용하여 복잡한 UI를 구성



반응

웹 및 기본 사용자 인터페이스를 위한 라이브러리The library for web and native user interfaces

[React 배우기](#)[API 레퍼런스](#)

리액트 - JS 라이브러리

리액트는 프론트엔드 담당

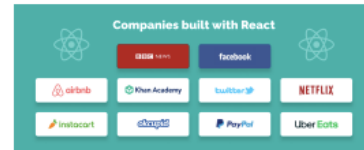
컴포넌트 - UI 구성 (주의)

클래스형과 함수형 존재



React 장점

1. 완성도 높고 이해하기 쉬운 개발 워크플로우
2. 뛰어난 유연성과 호환성
3. 손쉬운 컴포넌트 재사용성
4. Virtual DOM으로 한층 강화된 고성능
5. Flux와 Redux의 힘
6. 다양한 툴 제공
7. React Native의 강력한 기능
8. 시장에 영향을 미치는 거대한 커뮤니티와 리소스
9. HTML 확장을 위한 JSX 구문
10. React Hook



방대한 학습량
버전업
높은 상태 관리 복잡도
...

3번이 제일 뛰어남

리액트에서는 Virtual DOM을 도입

5는 중간고사 이후에

7번은 app이라 생각하면 쉽다

하나의 프레임워크 생성 → 제작 하는 회사에 스폰서 존재 → 스폰서들에 의해 상당히 다양한 코드들과 문서들, 개발자들이 참여하고있어 리액트가 인기



브라우저 로딩 과정

1. 파싱
 - ✓ 브라우저가 HTML을 파싱하고 읽어들이는 과정
 - ✓ DOM tree
 - ✓ CSSOM tree
2. 스타일
 - ✓ DOM Tree, CSSOM Tree가 생성되면 스타일을 매칭시켜주는 과정을 거쳐 렌더 트리를 구성.
 - ✓ 렌더링 트리에는 페이지를 렌더링하는 데 필요한 노드만 포함.
 - ✓ Render tree
3. 레이아웃
 - ✓ 기기의 뷰포트 내에서 노드의 정확한 위치와 크기를 계산하는 과정
4. 페인트
 - ✓ 렌더링 트리의 각 노드를 화면의 실제 픽셀로 변환하는 마지막 단계
5. 합성(Compositing)
 - ✓ 화면에 표시하기 위해 페이지에서 페인트된 부분을 합치는 과정
 - ✓ Transform, opacity, will-change 등을 사용했을 때 합성 과정을 거친다

파싱과정에서 컴파일러나 트리가 만들어짐

파싱 - 트리를 읽고 구문에 맞는지 안맞는지 확인

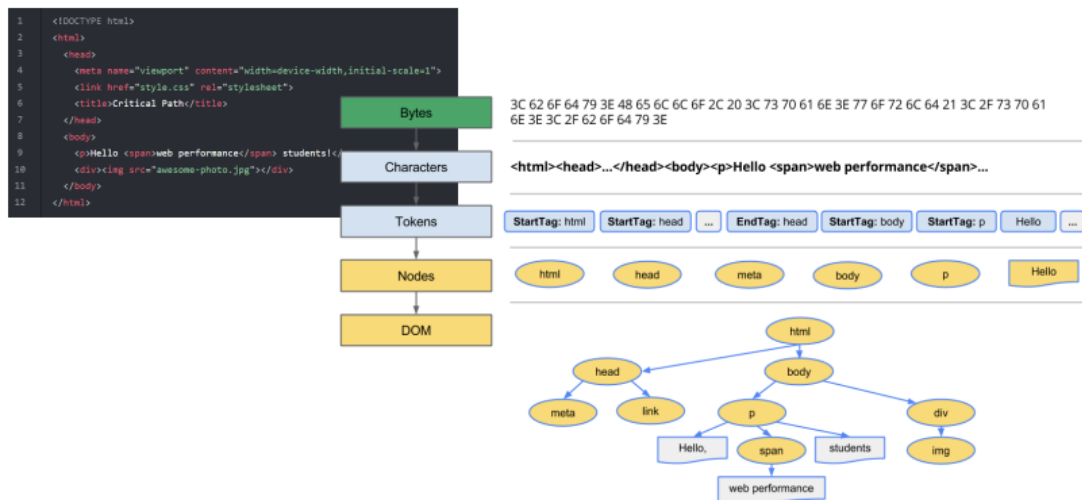
스타일 - 렌더 트리 구성 / 돔에는 헤더택 안에 있는 내용들이 함께 첨부

레이아웃 - 문서의 내용이 디바이스의 위치를 잘 잡아주는기 위해 사용

합성 - 애니메이션할때 사용, CSS



DOM tree



파싱처리하는 함수를 브라우저가 가지고 있다

파싱처리가 끝나면 스타일 처리 → ... → 합성 ○ 처리

서버에 파일들이 존재

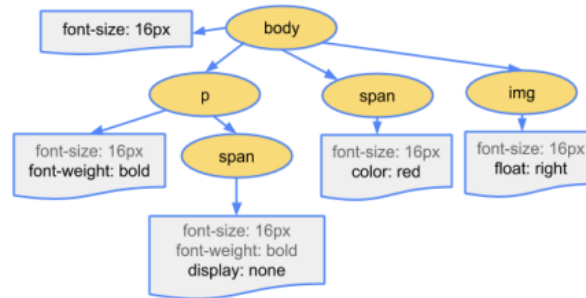
파일들을 이진수 받아 문자로 변환 후 문자를 토큰으로 변환

그것들을 노드로 만들어서 DOM 처리



CSSOM tree

```
1 body { font-size: 16px }
2 p { font-weight: bold }
3 span { color: red }
4 p span { display: none }
5 img { float: right }
```



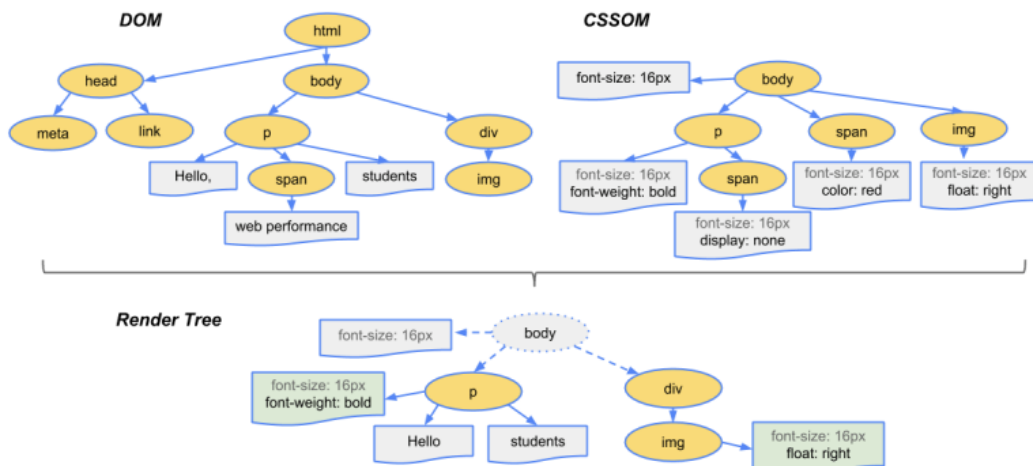
셀렉트 - html 태그 중에 어느 것

각 노드에 이와같은 스타일들을 적용

돔과 CSSOM이 합쳐져서 렌더트리 생성



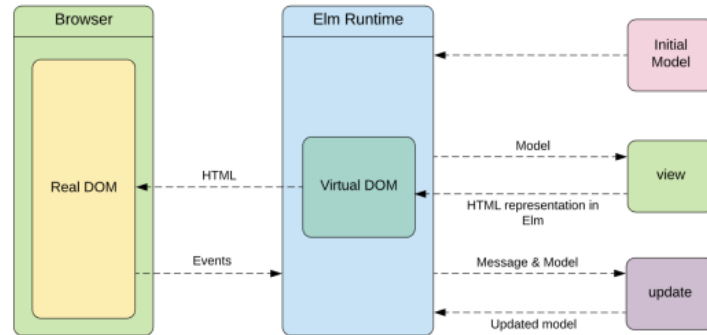
Render tree



화면구성에 아무런 영향을 주지않는 head는 렌더트리에서 빠진다



Virtual DOM



DOM에 변경이 있을 경우 렌더트리를 재생성하고(모든 요소들의 스타일이 다시 계산됨) 레이아웃을 만들고 페인팅을 하는 과정이 다시 반복되는 것을 알 수 있었다. 즉, 브라우저가 연산을 많이 해야한다는 이야기이며, 전체적인 프로세스를 비효율적으로 만든다는 것이다.

그래서 등장한 개념이 가상돔(VirtualDOM) 이다.

DOM이 JS코드로 변경될 수 있다

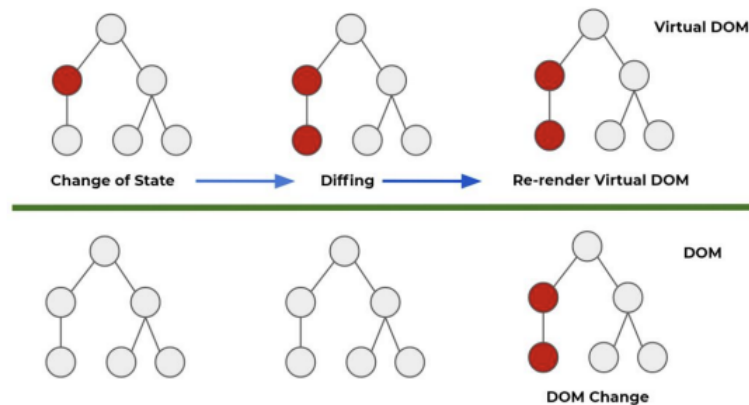
JS코드가 DOM트리를 변경 할 수 있다

JS가 DOM을 다룬다

이벤트가 생성될때마다 렌더트리가 새로 만들어짐



Virtual DOM



뷰(HTML)에 변화가 있을 때, 구 가상돔(Old Node)과 새 가상돔(New Node)을 비교하여 변경된 내용만 DOM에 적용한다. 이를 통해 브라우저 내에서 발생하는 연산의 양(정확히는 렌더링 과정)을 줄이면서 성능이 개선되는 것 이다.

노드 전체가 아닌 변경된 부분만 DOM트리에 변경



Real DOM vs Virtual DOM

```

1 <div id="app">
2   <ul>
3     <li>
4       <input type="checkbox" class="toggle" />
5       todo list item 1
6       <button class="remove">삭제</button>
7     </li>
8     <li class="completed">
9       <input type="checkbox" class="toggle" checked />
10      todo list item 2
11      <button class="remove">삭제</button>
12    </li>
13  </ul>
14  <form>
15    <input type="text" />
16    <button type="submit">추가</button>
17  </form>
18 </div>

```

Real DOM

```

1 function virtualDom(type, props, ...children) {
2   return { type, props, children: children.flat() }
3 }
4
5 virtualDom('div', { id: 'app' },
6   virtualDom('ul', null,
7     virtualDom('li', null,
8       virtualDom('input', { type: 'checkbox', className: 'toggle' }),
9       'todo list item 1',
10      virtualDom('button', { className: 'remove' }, '삭제')
11    ),
12    virtualDom('li', { className: 'completed' },
13      virtualDom('input', { type: 'checkbox', className: 'toggle', checked: true }),
14      'todo list item 2',
15      virtualDom('button', { className: 'remove' }, '삭제')
16    ),
17  ),
18  virtualDom('form',
19    virtualDom('input', { type: 'text' }),
20    virtualDom('button', { type: 'submit' }, '추가'),
21  )
22 );

```

Virtual DOM

보통 virtualDom 대신 h 로 표현

리엑트는 virtualDOM을 사용 → 상태변화는 버추얼 돔에서 먼저 발생 → 실제 돔과 버추얼 돔의 차이를 비교하여 실제 돔의 부분만 변경

개발 환경

리엑트 개발에 필요한 도구 모음

프로그램명	용도	운영체제	기타
Node.js	웹 서버 개발 플랫폼	윈도우/macOS	필수
VSCode	소스 코드 편집기	윈도우/macOS	권장
scoop	윈도우용 설치 프로그램	윈도우	권장
Homebrew	macOS용 설치 프로그램	macOS	권장
touch	파일 관리 유틸리티	윈도우	선택
Chrome	웹 브라우저	윈도우/macOS	권장

Node 14.0.0 혹은 상위 버전 및 npm 5.6 혹은 상위 버전이 필요



Package manager

Linux에서 사용하던 커맨드라인 패키지 매니저인 apt(apt-get), yum, Mac에서 사용하던 Homebrew 처럼 패키지 설치/업데이트/삭제 등에 사용

Chocolatey

- Chocolatey (약칭 Choco) : 윈도우에서 사용할 수 있는 커맨드 라인 패키지 매니저

Scoop

- 윈도우의 패키지 매니저
- 윈도우용 CLI (command-line installer)로 윈도우에서 좀 더 쾌적한 설치를 위해 개발

Winget

- Windows 10 및 Windows 11 컴퓨터에서 애플리케이션을 검색, 설치, 업그레이드, 제거 및 구성할 수 있음

WingetUI

- GUI를 제공하면서 winget, chocolatey, scoop 3개의 윈도우 패키지 관리자를 한번에 사용할 수 있는 오픈 소스 앱



npm, npx, nvm, yarn

npm(Node Package Manager)

- nodejs에서 사용하는 모듈들을 관리하고 배포하는 패키지 관리자
- nodejs에서 자주 쓰이는 코드들을 패키지로 만들어 배포하는 저장소



npx(Node Package Execute)

- npm을 더 편하게 사용하기 위한 도구
- npm이 패키지 설치와 관리를 돕는다면, npx는 패키지를 더 쉽게 실행할 수 있게 도와줌
- 외부 패키지에 포함된 실행 파일을 실행할 때 사용

nvm(Node Version Manager)

- nodejs 버전을 여러 개 사용할 때 편리

Yarn(Yet Another Resource Negotiator)

```
npm install yarn --global
```

- Node.js 자바스크립트 런타임 환경을 위해 페이스북이 2016년 개발한 소프트웨어 패키지 시스템



Babel

JavaScript에 컴파일러가 필요한 이유

모든 브라우저가 최신 문법, 기술(ES6)을 지원하지 않기 때문에 구 기능(ES5)으로 변환하는 작업이 필요

Babel이 하는 일

- Transform syntax (구문 변환)
 - 트랜스파일링은 최신의 자바스크립트 문법을 오래된 브라우저가 이해할 수 있도록 오래된 문법으로 변환해 준다.
- babel-polyfill을 통해서 폴리필 기능을 지원
 - 폴리필은 오래된 브라우저에 네이티브로 지원하지 않는 사용자가 사용하는 메서드, 속성, API가 존재하지 않을 때 추가해 준다.
 - 바벨은 최신 문법을 오래된 문법으로 변환해 주는 트랜스파일러 역할만 할 뿐 최신 함수를 사용할 수 있는 건 아니다.
 - 폴리필은 프로그램이 처음 시작할 때, 지원하지 않는 기능들을 지원해 준다.
 - 바벨은 컴파일 때 실행되고 폴리필은 런타임에 실행된다.
- JSX and React
 - 바벨은 JSX 문법을 변환한다.

Babel is a JavaScript compiler.
Use next generation JavaScript, today.

(중요)

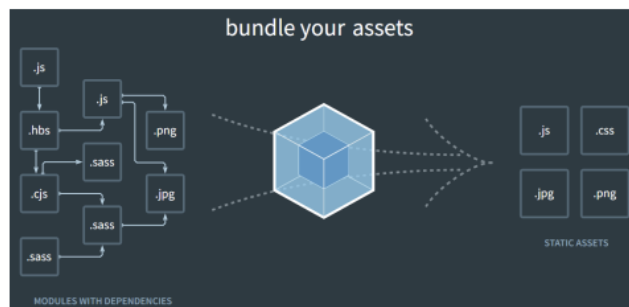
JS컴파일러

옛날것도 지금 사용할 수 있도록 만들어줌

JSX변환시켜주는 변환기



Webpack



웹팩(Webpack) 기본 사용법 (CLI)

웹팩(Webpack) 기본 설정법
(Entry/Output/Loader/Plugins)

웹팩(Webpack) 개발 편의 기능

Webpack

자바스크립트로 만든 프로그램을 배포하기 좋은 형태로 묶어주는 도구

모듈

프로그래밍 관점에서, 특정 기능을 갖는 작은 코드 단위

번들링

js, .css, .png ... 각각의 파일들을 모두 모듈로 보고, 이 모듈들을 배포용으로 하나의 파일로 통합시켜주는 작업

배포하기 쉽게 파일들을 묶어줌


```
x create-react-app m
mv-ann
```

CRA

모듈

- 프로시저 함수 매크로 등의 이름으로 불리기도 함

```
npx create-react-app my-app
```

```
cd my-app
```

```
dir
```

```
open 폴더
```

```
리액트 - my app 폴더선택
```

```
npm start
```



폴더 구조

node_modules

- 현재 프로젝트에 포함된 라이브러리들이 설치되어 있는 폴더
- 보통 깃과 같은 저장소에 올릴 때는 이 폴더를 함께 올리지 않음.

public

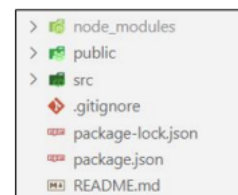
- index.html과 같은 정적 파일이 포함되는 곳으로 컴파일이 필요 없는 파일들이 위치하는 폴더

src

- 리액트 내부에서 작성하는 거의 모든 파일들이 이 폴더 내부에서 작성되며 이 폴더에 있는 파일들은 명령어에 따라 JS로 컴파일이 진행

.gitignore

- 깃에 포함하고 싶지 않은 파일의 이름 혹은 폴더 등을 입력하는 파일



src = 바벨 생각하면 됨

.gitignore 중요

```

name : my-app ,
"version": "0.1.0",
"private": true,
"homepage": ".",
"dependencies": {
  "@testing-library/jest-dom": "^5.17.0",
  "@testing-library/react": "^13.4.0",
  "@testing-library/user-event": "^13.5.0",
  "react": "^18.2.0",
  "react-dom": "^18.2.0"
}

```

You, 1 minute ago •

추가해서 작성

[네트워크/기본] URI, URL 및 URN의 차이점 - 하나몬

수정 보완 중에 있습니다... 🙏 그래도 의견 있으시면 주세요. ⚡ URI
 과 그 하위 개념 URL, URN 개념 이해하기 ! URI 이란? URI는
 Uniform Resource Identifier, 통합 자원 식별자의 줄임말이다. 브라
 🌐 <https://hanamon.kr/%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-%EA%B8%B0%EB%B3%B8-url-uri-urn-%EC%B0%A8%EC%9D%B4%EC%A0%90/>

