

Web Programming

React programming



React



React

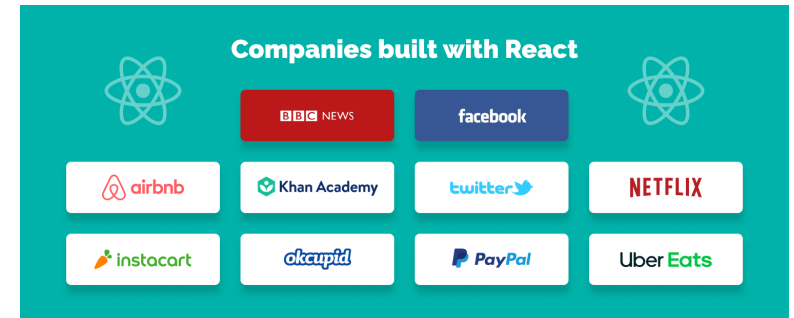
The library for web and native user interfaces

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

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
...



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)

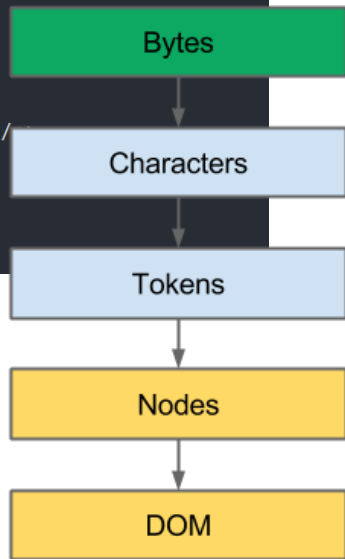
- ✓ 화면에 표시하기 위해 페이지에서 페인트된 부분을 합치는 과정
- ✓ Transform, opacity, will-change 등을 사용했을 때 합성 과정을 거친다





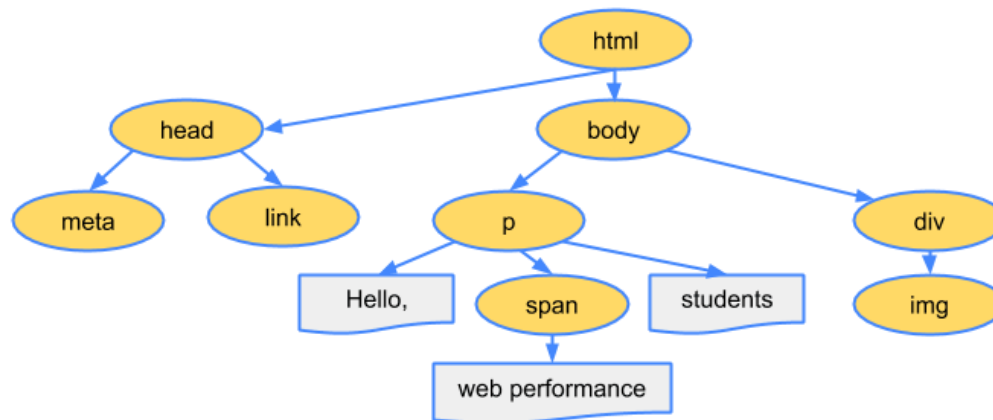
DOM tree

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta name="viewport" content="width=device-width,initial-scale=1">
5     <link href="style.css" rel="stylesheet">
6     <title>Critical Path</title>
7   </head>
8   <body>
9     <p>Hello <span>web performance</span> students!</p>
10    <div></div>
11  </body>
12 </html>
```



3C 62 6F 64 79 3E 48 65 6C 6C 6F 2C 20 3C 73 70 61 6E 3E 77 6F 72 6C 64 21 3C 2F 73 70 61
6E 3E 3C 2F 62 6F 64 79 3E

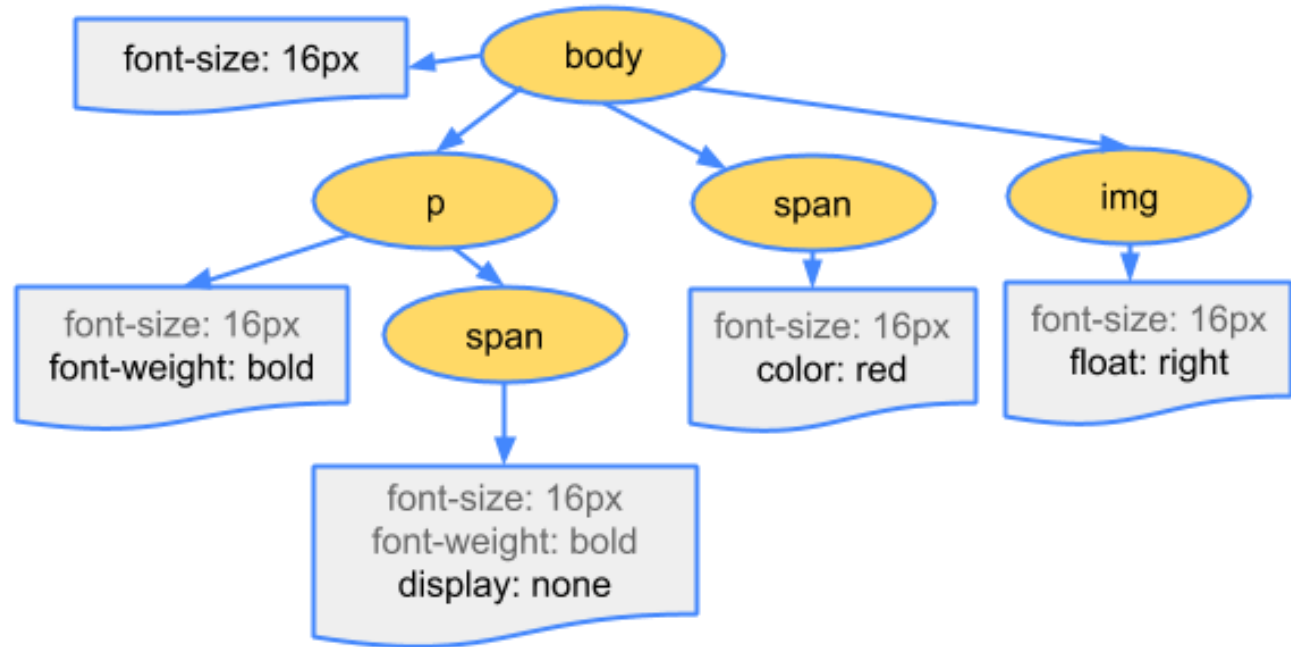
<html><head>...</head><body><p>Hello web performance...





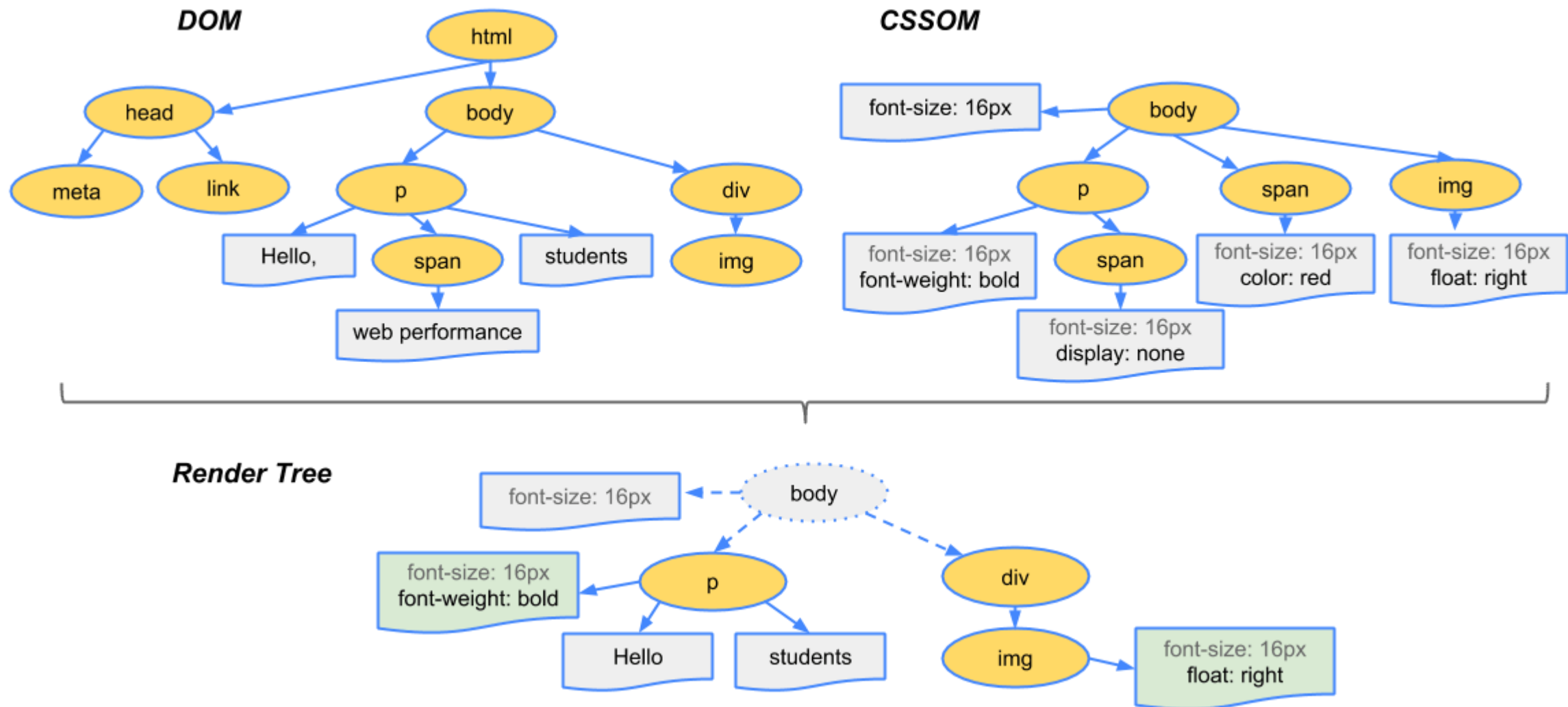
CSSOM tree

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



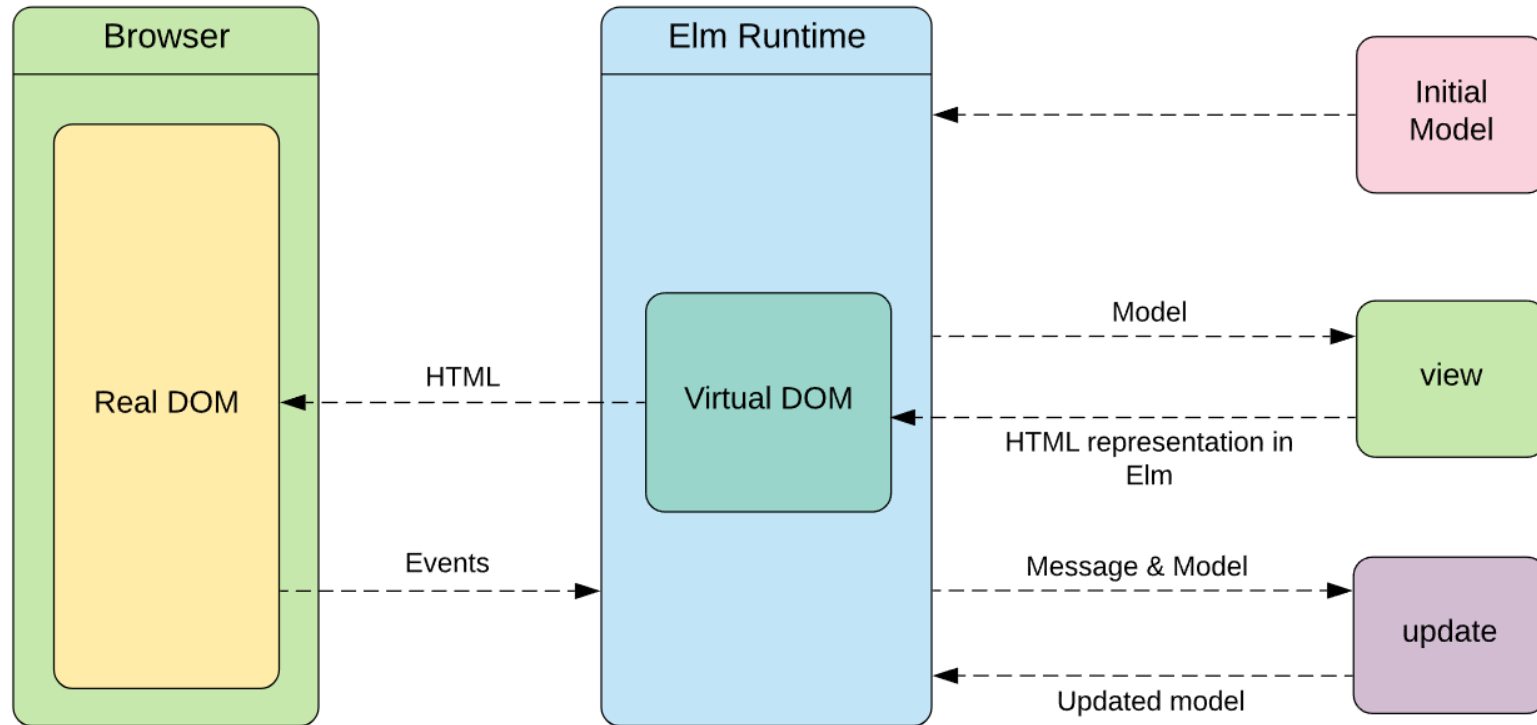


Render tree





Virtual DOM

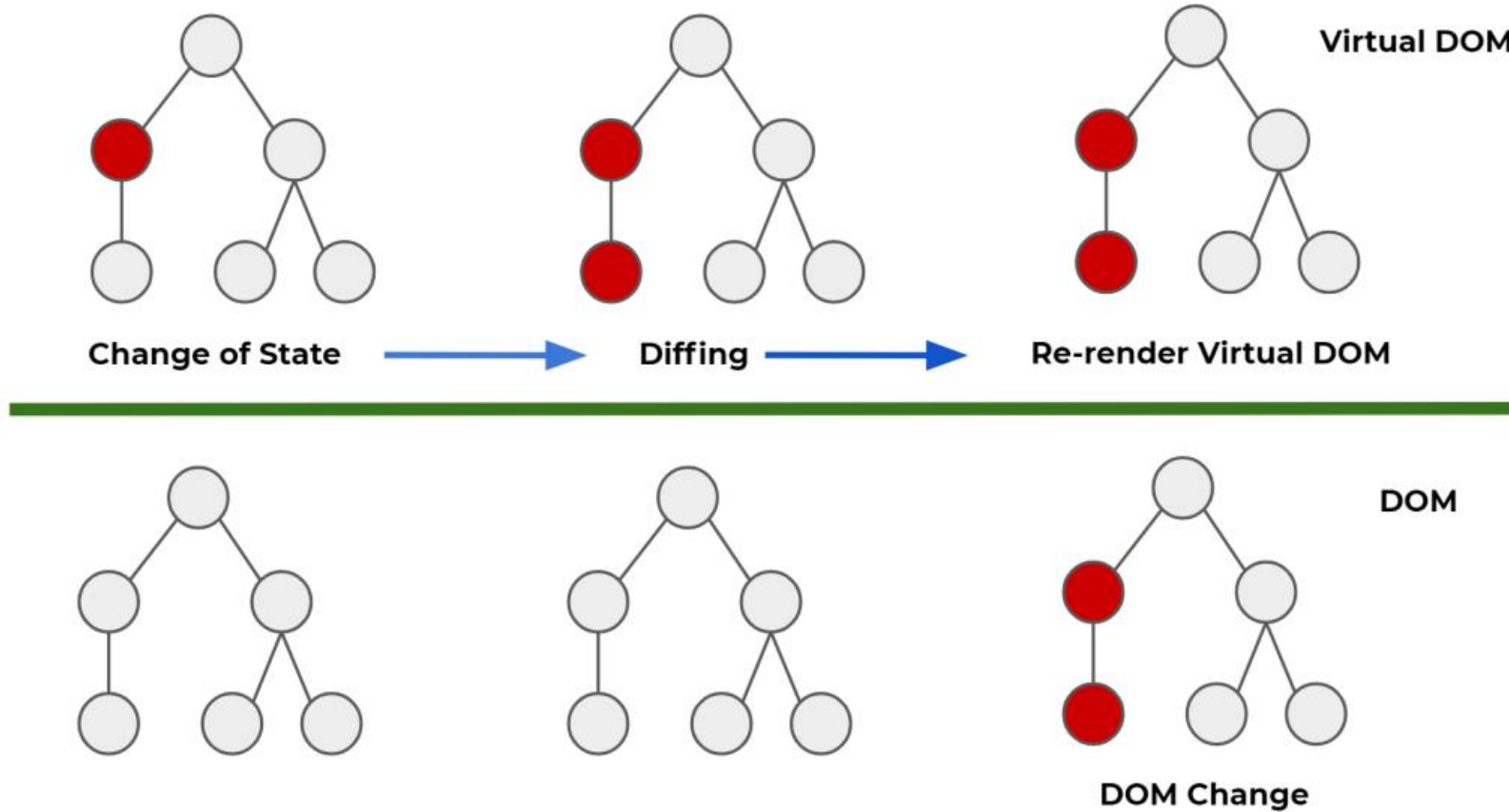


DOM에 변경이 있을 경우 렌더트리를 재생성하고(모든 요소들의 스타일이 다시 계산됨) 레이아웃을 만들고 페인팅을 하는 과정이 다시 반복되는 것을 알 수 있었다. 즉, 브라우저가 연산을 많이 해야한다는 이야기이며, 전체적인 프로세스를 비효율적으로 만든다는 것이다.

그래서 등장한 개념이 가상돔(VirtualDOM) 이다.



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.



Real DOM vs Virtual DOM

Real 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>
```

Virtual 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 );
```

보통 virtualDom 대신 h 로 표현



개발 환경

리액트 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



VS code 확장 프로그램

- ES7+ React/Redux/React-Native snippets
- Import Cost
- GitLens - Git supercharged
- React Native Tools
- Stylelint
- npm IntelliSense
- JavaScript Debugger (Nightly)
- Reactjs code snippets
- VSCode React Refactor
- Babel javascript
- Prettier - Code formatter
- Headwind
- PostCSS Language Support



Visual Studio Code



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

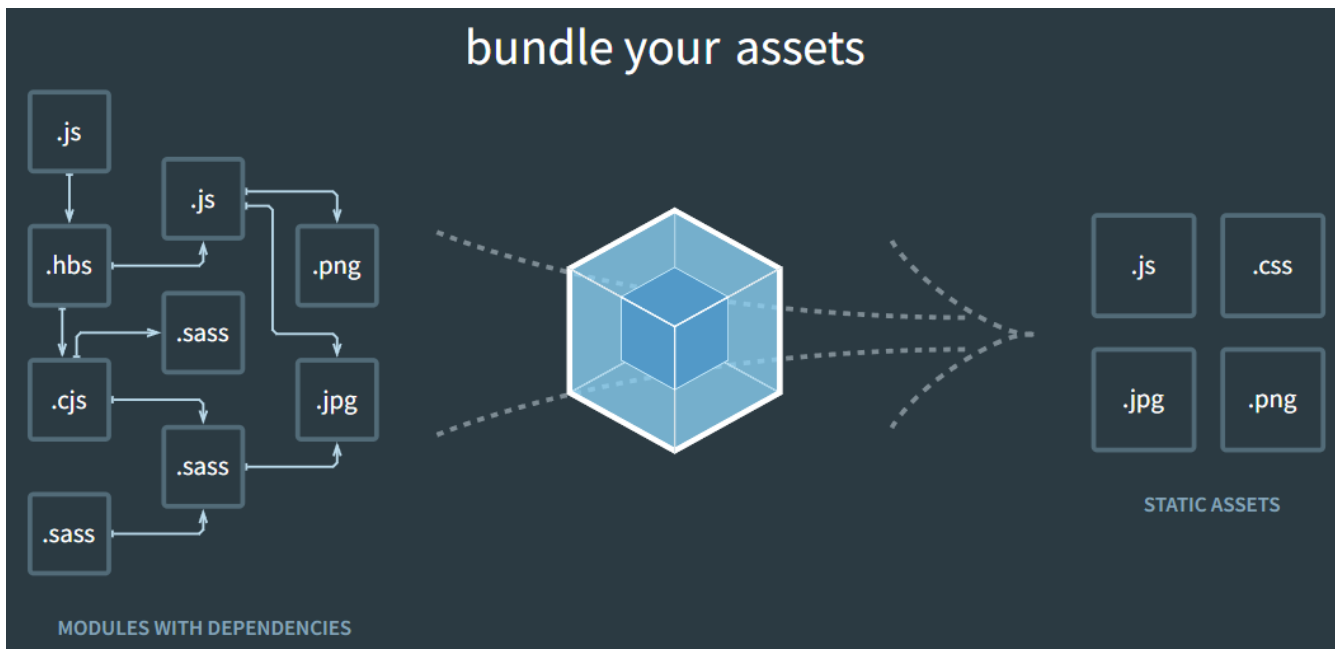
- Transform syntax (구문 변환)
 - 트랜스파일링은 최신의 자바스크립트 문법을 오래된 브라우저가 이해할 수 있도록 오래된 문법으로 변환해 준다.
- babel-polyfill을 통해서 폴리필 기능을 지원
 - 폴리필은 오래된 브라우저에 네이티브로 지원하지 않는 사용자가 사용하는 메서드, 속성, API가 존재하지 않을 때 추가해 준다.
 - 바벨은 최신 문법을 오래된 문법으로 변환해 주는 트랜스파일러 역할만 할 뿐 최신 함수를 사용할 수 있는 건 아니다.
 - 폴리필은 프로그램이 처음 시작할 때, 지원하지 않는 기능들을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



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

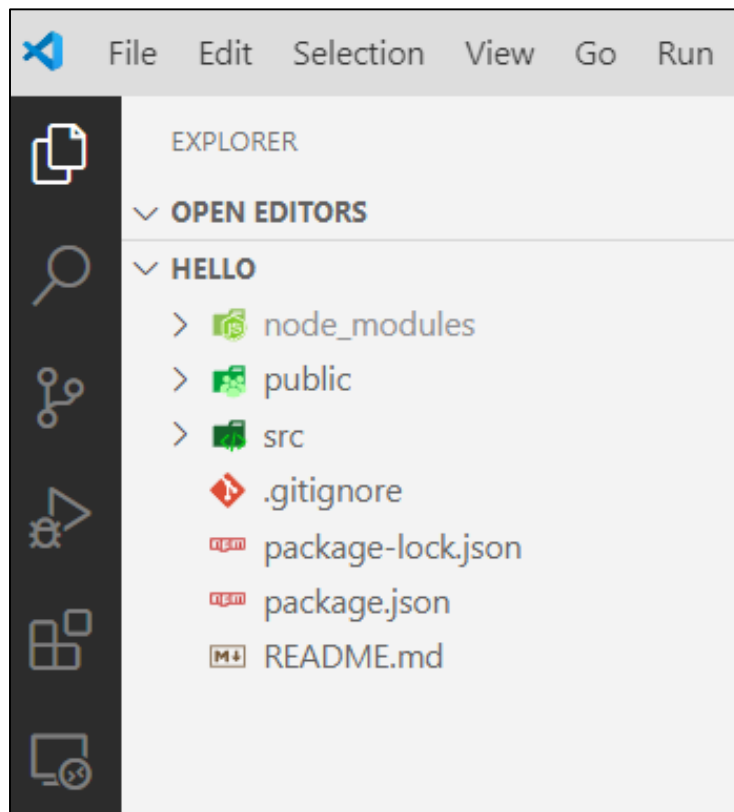
번들링

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



React 시작하기

```
> npx create-react-app my-app  
> cd my-app  
> npm start
```



Start a New React Project



폴더 구조

node_modules

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

public

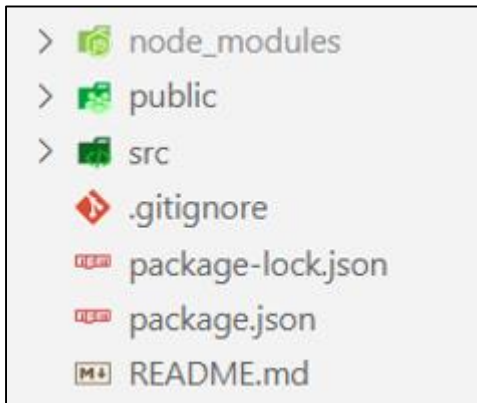
- index.html과 같은 정적 파일이 포함되는 곳으로 컴파일이 필요 없는 파일들이 위치하는 폴더

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





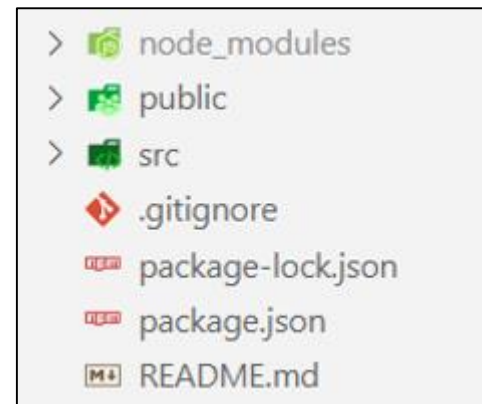
폴더 구조

package.json

- 프로젝트에 관련된 기본적인 내용(프로젝트의 이름, 버전 등)과 라이브러리들의 목록이 포함
- 라이브러리가 설치된 node_modules 대신에 이 package.json을 깃에 포함하여 올리게 되며 후에 누군가가 프로젝트를 클론할 때 이 package.json에 적혀있는 라이브러리의 목록을 기준으로 npm에서 설치

README.md

- 프로젝트에 대한 설명을 작성하는 곳





src 내부 폴더 구조

assets

- 이미지 혹은 폰트와 같은 파일들이 저장되는 폴더

components

- 재사용 가능한 컴포넌트들이 위치하는 폴더

config

- 여러 개의 config 파일들이 위치하는 폴더

constants

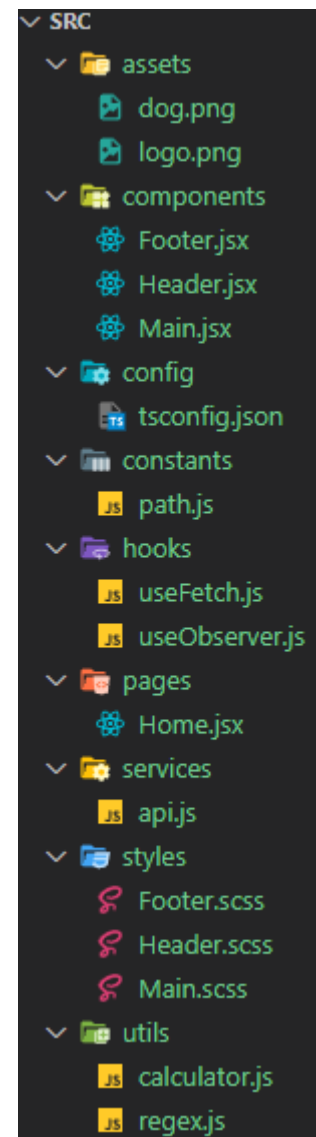
- 공통적으로 사용되는 상수들을 정의한 파일들이 위치하는 폴더

hooks (= hoc)

- 커스텀 훅이 위치하는 폴더

pages

- react router등을 이용하여 라우팅을 적용할 때 페이지 컴포넌트를 이 폴더에 위치





src 내부 폴더 구조

services (= api)

- 보통 api관련 로직의 모듈 파일이 위치하며 auth와 같이 인증과 관련된 파일이 포함

styles

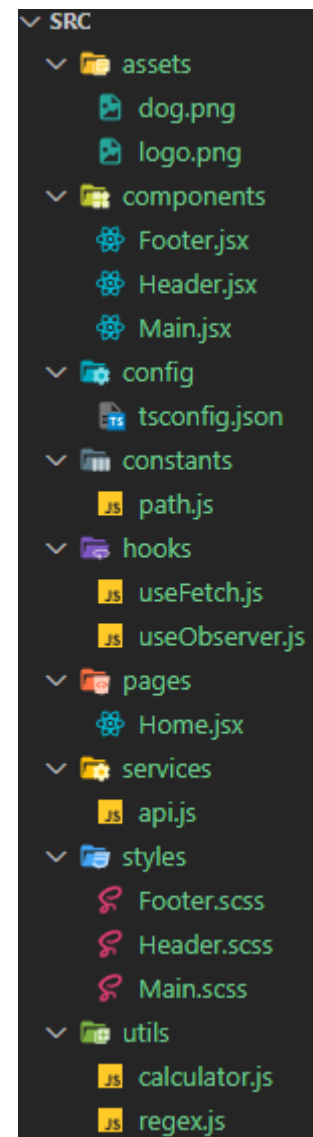
- css 파일들이 포함되는 폴더

utils

- 정규표현식 패턴이나 공통함수 등 공통으로 사용하는 유틸 파일들이 위치하는 폴더

contexts

- contextAPI를 사용할 때 관련 파일들이 위치하는 곳으로 상태 관리를 위해 contextAPI 대신 redux를 사용 할 경우 폴더 이름을 store로 사용





Development vs Production Mode

개발 모드

프로젝트 개발 시 적용



```
T 터미널  
> npm start
```

프로덕션 모드

개발 완료 후 사용자에게 실제 서비스를 하려고 할 때 적용



```
T 터미널  
> npm run build
```



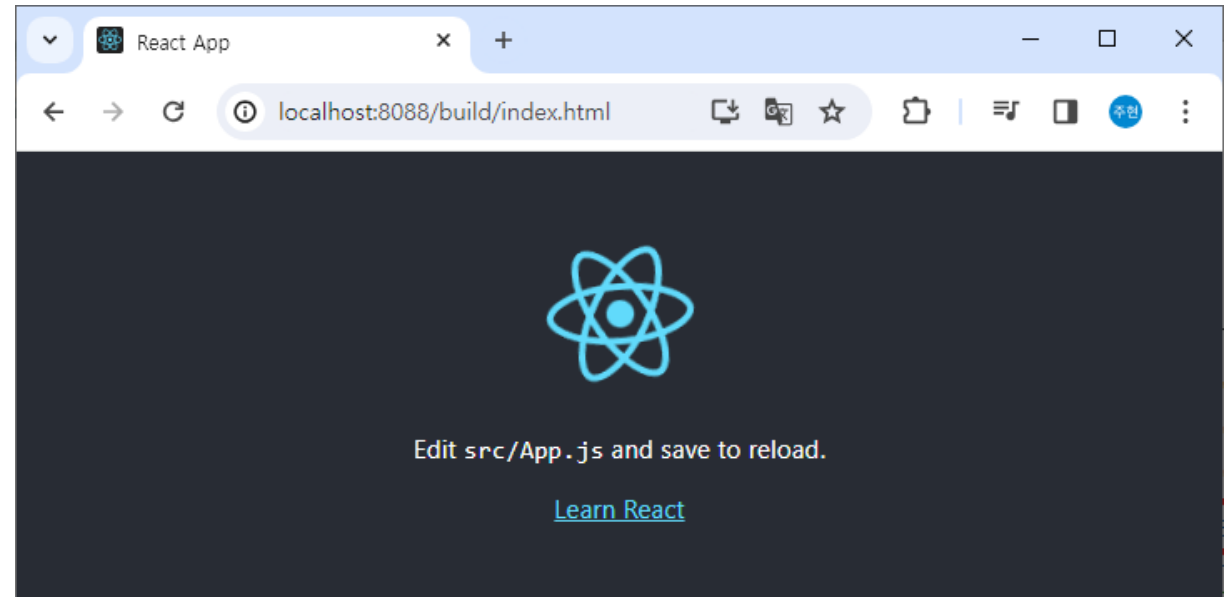
Publishing

Deploy

1. package.json 편집
 - ✓ "homepage": ".", 추가
2. npm run build
3. [npm install -g serve]
4. serve -s build
5. build 폴더 사용

serve라는 웹서버를 생성해서 실행시킬 때 build를 document root로 하겠다는 뜻

```
1 {  
2   "name": "hello",  
3   "version": "0.1.0",  
4   "private": true,  
5   "homepage": ".",  
6   "dependencies": {}  
7   "@testing-library/jest-dom": "^5.17.0",  
8   "@testing-library/react": "^13.4.0",
```



- Release: 같은 제품을 새롭게 만드는 것(예: 새로운 버전을 배포, 새로운 아이피 번호 부여)
- Deploy: 프로그램 등을 서버와 같은 기기에 설치하여 서비스 등을 제공하는 의미
- Distribute: 제품을 사용자들이 사용할 수 있도록 서비스 등을 제공하는 의미



Index.html, index.js, App.js

index.html

- public 폴더에 있다.
- 메인 프로그램인 index.js에 대응되는 것으로, HTML 템플릿 파일이다.
- 이 파일이 직접 표시되는 것은 아니고, index.js에 의해 일어 와서 렌더링된 결과가 표시된다.
(index.html 이름을 바꿀 시 오류발생)

index.js

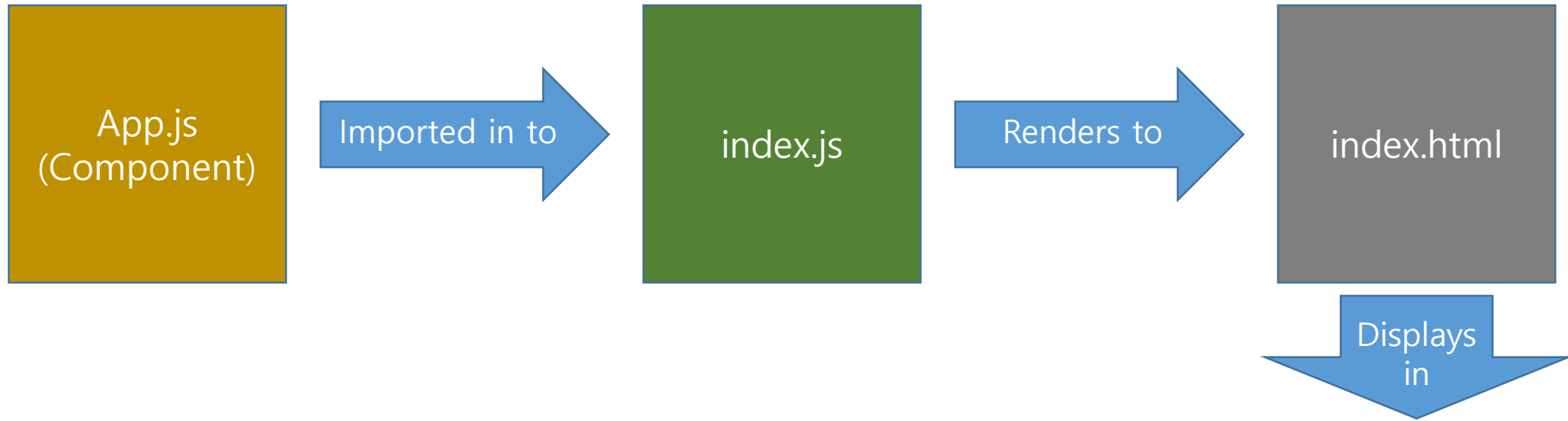
- src 폴더에 포함되어 있다. 메인 프로그램이라고 할 수 있다.
- 여기에서 HTML 템플릿 및 JavaScript의 컴포넌트를 조합하여 렌더링하고 실제 표시한다.

App.js

- src 폴더에 있다.
- 이것은 컴포넌트를 정의하는 프로그램이다.
- 실제로 화면에 표시되는 내용 등은 여기에서 정의된다.



Index.html, index.js, App.js



- ✓ App.js 에서는 실제로 사용자에게 보여줄 화면을 정의한다.
- ✓ index.js 에서는 'root' 에 App.js 를 렌더링 시킨다.
- ✓ index.html 에는 'root' 라는 div가 하나 있다.

The idea is to divide the web page in to UI Components that are easily updated and re-used.





App.js

```
function MyButton() {  
  return (  
    <button>  
      I'm a button  
    </button>  
  );  
}  
  
export default function MyApp() {  
  return (  
    <div>  
      <h1>Welcome to my app</h1>  
      <MyButton />  
    </div>  
  );  
}
```

Welcome to my app

I'm a button



Report : 깃허브 디렉토리에 build를 호스팅(편집)

발표 주제 : git & source

명령어 설명

commit(소스)

commit(소스 변경)

branch

merge

commit

commit

github(위의 작업을 깃허브에서 협업하는 방법)

vs code & github



Reference

- <https://react.dev/>
- [https://developer.mozilla.org/ko/docs/Learn/Tools and testing/Client-side JavaScript frameworks/React getting started](https://developer.mozilla.org/ko/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started)
- <https://ko.legacy.reactjs.org/docs/create-a-new-react-app.html>
- <https://joong-sunny.github.io/react/react2/>
- <https://www.howdy-mj.me/dom/what-is-dom>
- <https://velog.io/@ye-ji/DOM-vs-Virtual-DOM>
- https://junilhwang.github.io/TIL/Javascript/Design/Vanilla-JS-Virtual-DOM/#_2-jsx
- 명품 웹 프로그래밍, 황기태, 생능출판사
- 리액트 & 리액트 네이티브 통합 교과서, 아담 보두치, 강경일, 신희철, 에이콘
- Do it! 리액트 모던 웹 개발 with 타입스크립트, 전예홍, 이지스퍼블리싱