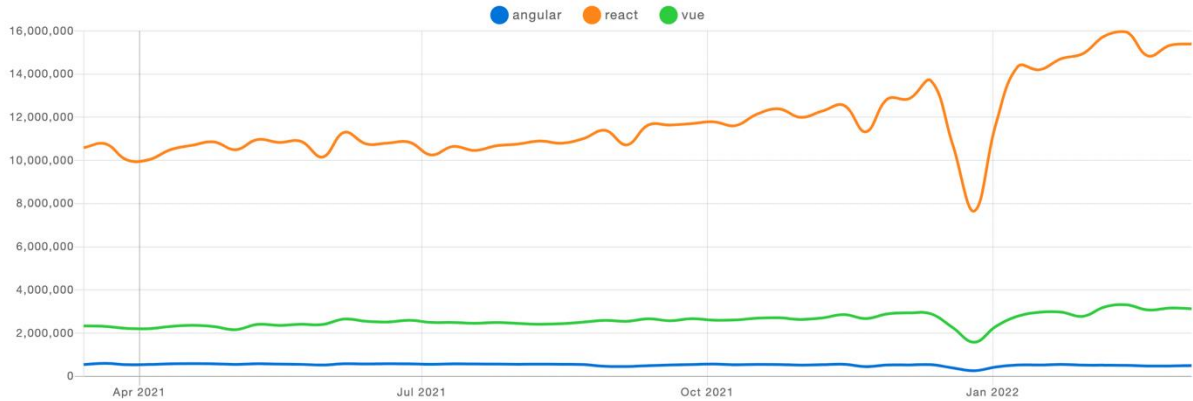


# 왜 리액트인가? (리액트 튜토리얼)

정보통신공학부 2020037069 김지원

## 시작에 앞서 사진을 하나 봐보자



이는 npm trend에서 볼 수 있는 1년간의 spa 비교이다. Angular와 vue 프레임워크의 다운로드수를 다 합쳐도 리액트에 한참 모자른 것을 확인할 수 있다.

또 웹개발 채용 현황에서 원티드, 프로그래머스, 랠릿등을 다들 리액트를 잘 다루는 프론트엔드 개발자를 다수 채용하는 것을 알 수 있다.

리액트가 뭔데 이렇게 핫한걸까? 왜 리액트를 알아야 하는거고 리액트는 무엇을 해결하는가?

## 리액트란?

리액트는 페이스북이 만든 사용자 인터페이스를 만드는 것을 돕는 라이브러리이다. (리액트 공식문서에서 발췌)

사용자 인터페이스 만드는 것을 돕는다고 치면 무엇을 돕는 것인가?

1. 동적으로 DOM을 컨트롤 하여 내용을 다룰 수 있게 함.
2. Virtual Dom을 이용하여 브라우저의 워크플로우를 관리한다. (모바일 앱같은 느낌을 줌)
3. 데이터를 가져와 내가 원하는 데로 데이터를 그리는 작업에 효율적인 프로세스를 유도한다.

이 얘기를 읽다가 벌써 이해가 안 될 수 있다. **DOM이 뭔데?**

Dom을 요약하자면 Document Object Model로 웹 페이지는 사실상 문서인데 이를 object화하여 사용할 수 있게 해준다. 즉 웹 페이지에서 우리가 볼 수 있는 모든 것을 생성, 변형, 삭제할 수 있게 하는 Model이고 개발자는 이를 통해 웹페이

지를 다룬다.

ㅈ

DOM보다는 리액트의 사용이유와 튜토리얼에 대한 얘기를 하는 글이기에 더 필요한 내용이 있다면 [mozilla](#)를 참고하길 바란다.

즉 리액트는 사용자가 DOM을 다루는 것을 도와준다.

그러면 이런 리액트는 어떻게 사용할 수 있는가?

```
<!-- ... 다른 HTML ... -->
```

```
<!-- React 를 실행. -->
```

```
<!-- 주의: 사이트를 배포할 때는 "development.js"를 "production.min.js"로  
대체하세요. -->
```

```
<script src="https://unpkg.com/react@17/umd/react.development.js"  
crossorigin></script> <script src="https://unpkg.com/react-dom@17/umd/react-  
dom.development.js" crossorigin></script>
```

```
<!-- 만든 React 컴포넌트를 실행. -->
```

```
<script src="like_button.js"></script>  
</body>
```

이런 식으로 html에 script 추가만 해줘도 리액트를 사용할 수 있다. 하지만 우리가 하는 프로젝트를 굳이 script로 관리할 필요는 없다. React가 만들어놓은 create-react-app이란 기능이 있기 때문이다.

Create-react-app을 이용하는 순서를 정리하자면

1. Node와 npm을 설치한다.

이는 운영체제에 따라 설치방법에 차이가 있기 때문에 따로 설명하지 않겠다.

2. 터미널에서 내가 이 앱을 만들려 하는 경로로 이동한다.

```
~ (0.07s)  
cd Documents/dev  
~/Documents/dev
```

3. npx create-react-app 내가-원하는-폴더-이름을 입력한다.

```
~ (0.07s)
cd Documents//dev

~/Documents/dev
npx create-react-app react-tutorial
```

#### 4. 잠깐의 설치시간이 지난 후

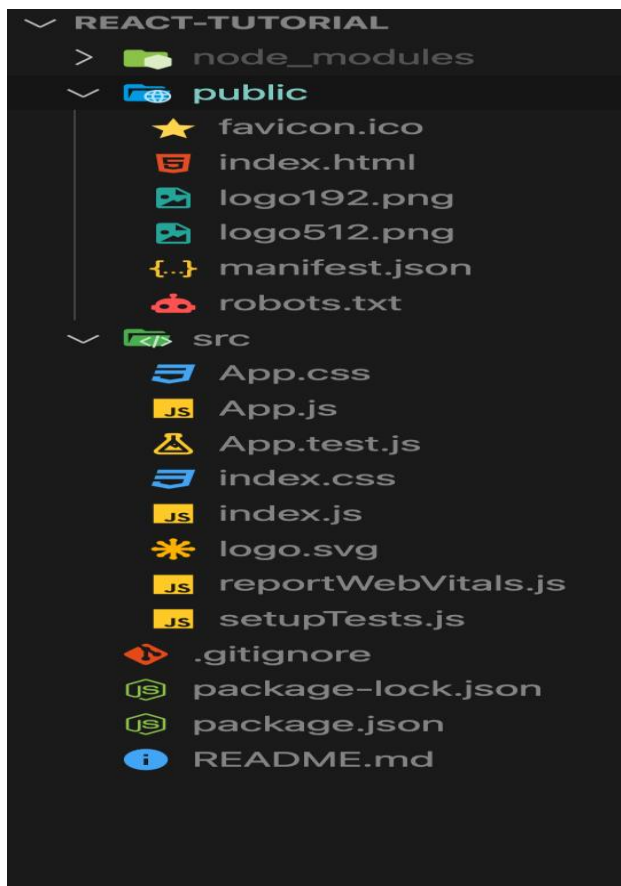
```
~ (0.07s)
cd Documents//dev
~/Documents/dev
npx create-react-app react-tutorial

Creating a new React app in /Users/jiwonkim/Documents/dev/react-tutorial.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

: idealTree:workbox-webpack-plugin: timing idealTree:node_modules/workbox-webpack-plugin Completed in 596ms
```

#### 5. 이런 구조의 폴더가 완성된다.



우리는 이런 구조를 기본으로 나의 방식에 맞게 커스텀하여 코드를 작성하면 된다.

**그러면 어떻게 실행하냐?**

모든 실행, 테스트, 배포 에 관한 것은 package.json파일이 중심이 된다.

package.json 파일을 살펴보자.

```
{
  "name": "react-tutorial",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.2",
    "@testing-library/react": "^12.1.4",
    "@testing-library/user-event": "^13.5.0",
    "react": "^17.0.2",
    "react-dom": "^17.0.2",
    "react-scripts": "5.0.0",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [

```

저기에 scripts 부분이 명령어를 미리 지정해 주는 부분이다.

즉 package.json파일이 있는 경로에서

1. npm start를 입력하면 실행
2. npm build를 입력하면 빌드
3. npm test를 입력하면 테스트(테스트 코드를 기반하여 .test나 .spec파일들)
4. npm eject를 입력하면 숨겨져 있는 설정들을 꺼낼 수 있다.

즉 cra가 잘 됐는 지 확인하기 위해 npm start를 입력하면

```
Compiled successfully!

You can now view react-tutorial in the browser.

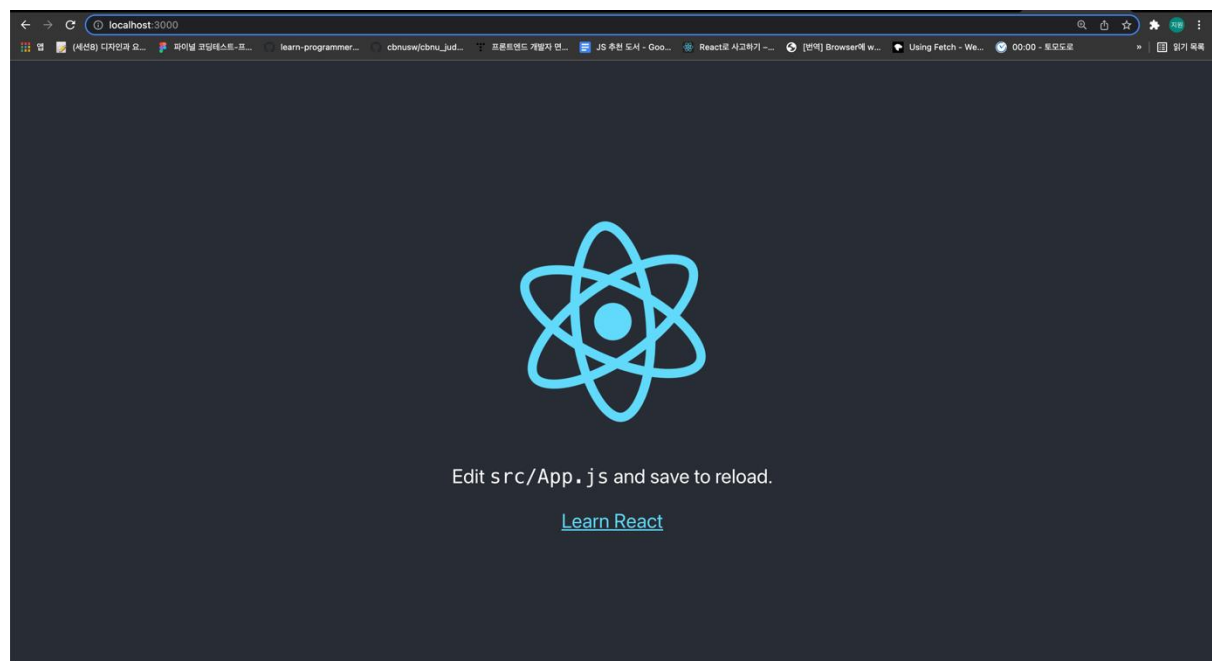
Local:      http://localhost:3000
On Your Network:  http://172.30.24.2:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

assets by path static/ 1.49 MiB
  asset static/js/bundle.js 1.48 MiB [emitted] (name: main) 1 related asset
  asset static/js/node_modules_web-vitals_dist_web-vitals_js.chunk.js 6.93 KiB [emitted] 1 related asset
  asset static/media/logo.6ce24c58023cc2f8fd88fe9d219db6c6.svg 2.57 KiB [emitted] (auxiliary name: main)
asset index.html 1.67 KiB [emitted]
asset asset-manifest.json 546 bytes [emitted]
runtime modules 31.3 KiB 15 modules
modules by path ./node_modules/ 1.35 MiB 99 modules
modules by path ./src/ 18.1 KiB
  modules by path ./src/*.css 8.82 KiB
    ./src/index.css 2.72 KiB [built] [code generated]
    ./node_modules/css-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[5].use[1]!./node_modules/postcss-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[5].use[2]!./node_modules/source-map-loader/dist/cjs.js!./src/index.css 1.37 KiB [built] [code generated]
    ./src/App.css 2.72 KiB [built] [code generated]
    ./node_modules/css-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[5].use[1]!./node_modules/postcss-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[5].use[2]!./node_modules/source-map-loader/dist/cjs.js!./src/App.css 2 KiB [built] [code generated]
  modules by path ./src/*.js 5.71 KiB
```

이런 메시지가 뜨며 3000번 포트를 확인하라고 나온다.

확인해보자



잘 실행되는 것을 확인할 수 있다.

이제 react를 배워 사용하실 분들이 react를 왜 배워야하는 지 더 직관적으로 이해할 수 있게 react를 오래 사용한 사용자로서 말하자면

- 페이스북이 페이스북이라는 초대형 서비스에 직접 사용하며 증명하고 있다.
- 커뮤니티의 크기가 제일 크다.
- 다른 라이브러리들의 기준이 된다.

리엑트는 규모가 제일 큰 인터페이스 라이브러리의 역할을 정말 잘하고 있고 많은 개발자들은 리엑트의 철학에 공감한다. 실제로 react가 없던 시절부터 react의 방식대로 컴포넌트를 만들던 사람들이 좋은 평가를 받는 경우가 많다.

React에서 데이터를 관리하고 데이터를 렌더링하는 방식, 렌더링을 효율적으로 관리하는 방법등은 다음 튜토리얼을 쓸 기회가 있다면 게시하도록 하겠다.

혹시 그런 내용들이 더 궁금하다면 리엑트 공식문서를 찾아보는 것을 굉장히 추천한다. 또 내가 풀이한 리엑트의 생각들이 궁금하다면 [개발나무 심기 블로그](#)를 들어오셔도 좋을 것 같다.