


## 01장 리액트 시작하기

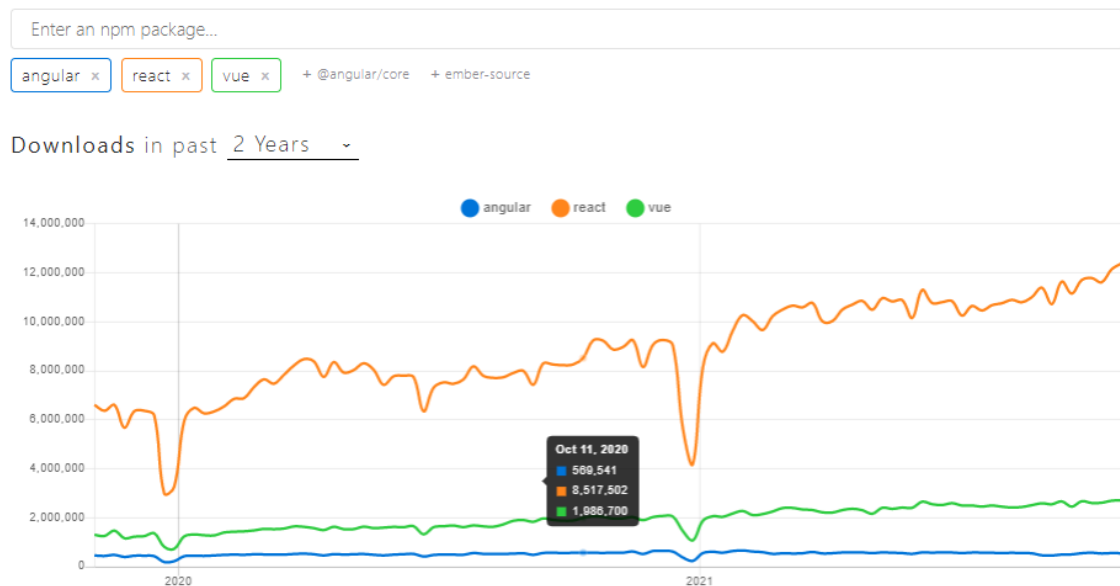
### 1.1 리액트의 정체를 알아보자!

#### 1.1.1 어떤 프레임워크를 공부해야 할까?

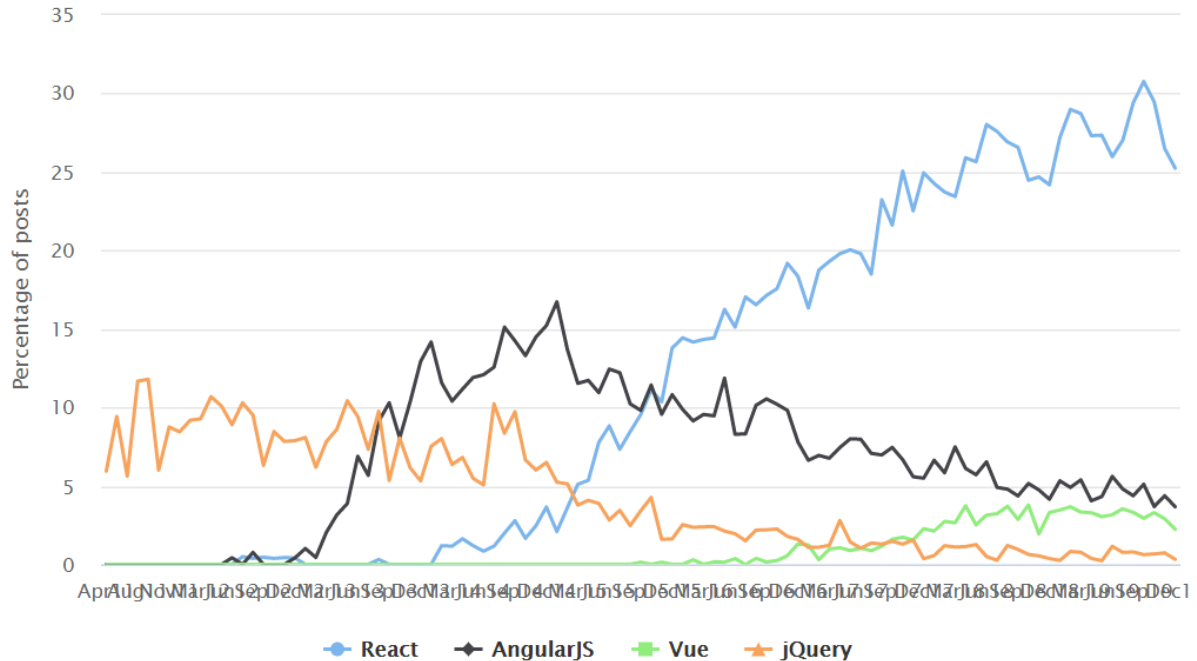
- 리액트는 프론트엔드 프레임워크 중 하나이다. 현재 가장 주목받는 프레임워크는 react, angular, vue를 꼽을 수 있다.
- 프레임워크 다운로드와 개발자 수요 현황 그래프
  - <https://www.npmtrends.com/angular-vs-react-vs-vue>
  - <https://bit.ly/2Rmx1Zn>

 npm trends

angular vs react vs vue



## 리액트 프로그래밍 (프론트엔드 개발)



### 1.1.2 리액트의 개념과 장점은 무엇일까?

- 리액트는 페이스북을 개발할 때 사용한 기술이며, 공개 소프트웨어이다. 리액트의 가장 큰 특징은 ‘ 화면 출력에 특화된 프레임워크 ’ 라는 것이다.
- 컴포넌트로 화면 구성을 효율적으로 할 수 있다.
- 게임 엔진 원리를 도입하여 화면 출력 속도가 빠르다.

### 1.1.3 노드 패키지 매니저란 무엇일까?

- npm은 <https://www.npmjs.com>에서 필요한 자바스크립트 라이브러리를 내려받아 설치하고 삭제하는 등의 관리를 해주는 프로그램이다.
- npm은 실제로 node\_modules 폴더에 라이브러리를 내려받아 저장하고 package.json이라는 파일에 설치된 라이브러리의 정보를 적어 저장한다.

### 1.1.4 웹팩이란 무엇일까?

- 웹팩은 프로젝트에 사용된 파일을 분석하여 기존 웹 문서 파일로 변환하는 도구이다. 웹 브라우저가 해석할 수 없는 .hbs, .cjs, .sass 등의 파일을 웹팩이 분석하여 .js, .css 등의 파일로 변환해 준다.
- 웹팩은 간단한 노드 기반의 웹 서버를 구동하기도 한다.

## 1.2 리액트 개발 환경 설치하기

### 1.2.0 ~~노드 버전 매니저로 노드제이에스 설치하기~~

- 노드제이에스는 구글에서 공개한 소프트웨어로 V8 엔진을 기반으로 만든 자바스크립트 런타임 도구이다. 쉽게 말해 웹 브라우저가 아닌 서버(컴퓨터)에서 자바 스크립트를 실행할 수 있게 해준다.
- NVM은 아래의 주소에 nvm-setup.zip을 내려받아 압축을 해제하고 설치 파일을 실행한다.
  - <https://github.com/coreybutler/nvm-windows/releases>

```
# NVM 버전 확인하기
d:\prod\nvm>nvm -v
Running version 1.1.8.

# NVM으로 노드제이에스 설치하기
d:\prod\nvm>nvm install 10.10.0
Downloading node.js version 10.10.0 (64-bit)...
Complete
```

### 1.2.1 노드제이에스 설치하기

- 노드제이에스(Node.js)는 서버 사이드 자바스크립트로 서버 측에서 실행되는 자바스크립트 실행 환경이다. 뷰 CLI(Command Line Interface)를 이용하여 쉽게 뷰 프로젝트를 구성하려면 노드제이에스가 설치되어야 한다.
- 설치
  - <https://nodejs.org/dist/v16.13.0/> 공식 사이트에서 "node-v16.13.0-x64.msi" 파일을 다운로드하여 설치한다.

```
# node 버전 확인하기
C:\Users\admin01>node -v
v16.13.0
```

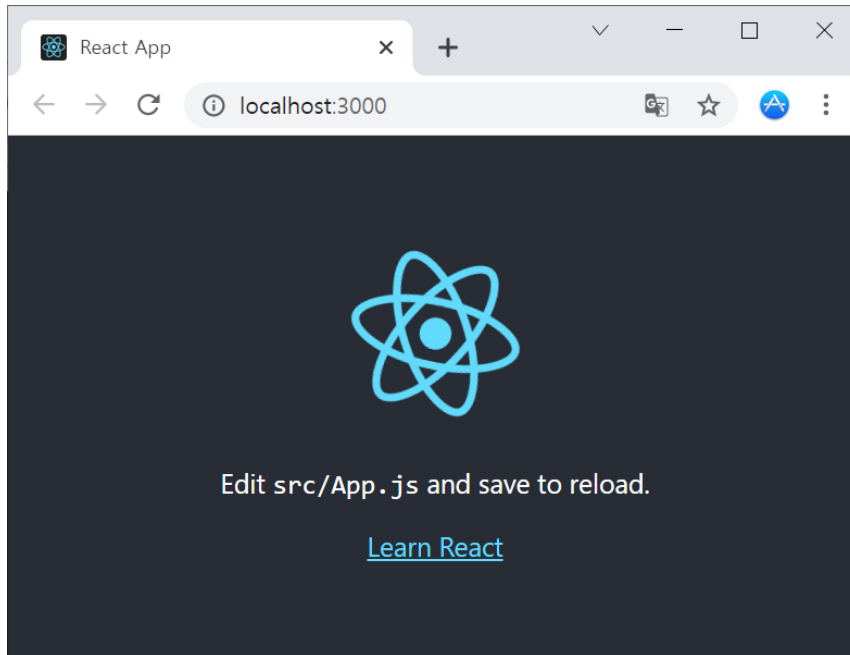
### 1.2.2 yarn과 create-react-app 설치하기

- npm으로 yarn을 설치한다. 앞으로 각종 설치는 npm 대신 yarn을 사용하도록 한다.

```
# 1. yarn 설치하기
C:\Users\admin01>npm install -g yarn

# 2. 리액트 앱 생성하기
D:\dev\workspace\react>yarn create react-app do-it-example --scripts-version 2.1.7

# 3. 리액트 앱 구동하기
D:\dev\workspace\react>cd do-it-example
D:\dev\workspace\react\do-it-example>yarn start
```



### 1.2.3 예제에 필요한 라이브러리 미리 설치하기

- package.json을 통해 프로젝트에 필요한 라이브러리를 미리 설치한다.
- package.json 수정하기
  - 프로젝트 루트 폴더(do-it-example)에 있는 package.json 파일을 다음과 같이 수정한다.

[./package.json]

```
01  {
02    "name": "do-it-example",
03    "version": "0.1.0",
04    "private": true,
05    "dependencies": {
06      "axios": "0.18.1",
07      "enzyme": "^3.8.0",
08      "enzyme-adapter-react-16.3": "^1.4.1",
09      "moment": "^2.24.0",
10      "next": "^8.1.0",
11      "react": "^16.7.0",
12      "react-dom": "^16.7.0",
13      "react-redux": "^6.0.0",
14      "react-router-dom": "^5.0.0",
15      "react-scripts": "2.1.7",
16      "react-test-renderer": "^16.7.0",
17      "react-with-styles": "^3.2.1",
18      "recompose": "^0.30.0",
19      "redux": "^4.0.1",
20      "redux-pack": "^0.1.5",
21      "redux-thunk": "^2.3.0",
22      "reselect": "^4.0.0",
23      "selector-action": "^1.1.1"
24    },
25    "scripts": {
26      "dev": "next",
27      "predeploy": "yarn build-all",
28      "deploy": "firebase deploy",
```

```
29   "build-all": "yarn ssrbuild && yarn build-firebase",
30   "build-firebase": "cd \"./functions\" && yarn --ignore-engines",
31   "ssrbuild": "next build",
32   "storybook": "start-storybook -p 9001 -c .storybook",
33   "start": "react-scripts start",
34   "build": "react-scripts build",
35   "test": "react-scripts test",
36   "mockserver": "json-server --watch --delay 500 --port 4000 mock/create.js",
37   "errorserver": "node mock/fake.js",
38   "eject": "react-scripts eject"
39 },
40 "eslintConfig": {
41   "extends": "react-app"
42 },
43 "browserslist": [
44   ">0.2%",
45   "not dead",
46   "not ie <= 11",
47   "not op_mini all"
48 ],
49 "devDependencies": {
50   "@babel/core": "7.5.5",
51   "@babel/plugin-syntax-object-rest-spread": "^7.2.0",
52   "@storybook/addon-actions": "^5.2.6",
53   "@storybook/addons": "^5.2.6",
54   "@storybook/react": "^5.2.6",
55   "aphrodite": "^2.2.3",
56   "babel-loader": "^8.0.5",
57   "json-server": "^0.14.2",
58   "node-sass": "^4.11.0",
59   "react-with-styles-interface-aphrodite": "^5.0.1",
60   "redux-devtools-extension": "^2.13.8",
61   "sass-loader": "^7.1.0",
62   "storybook-addon-jsx": "^7.1.13"
63 }
64 }
```

#### ■ package.json에 적힌 라이브러리 모두 설치하기

```
D:\dev\workspace\react\do-it-example>yarn
```

#### ■ babel-loader 호환성 오류 해결하기

- babel-loader 버전 오류가 발생할 경우 루트 폴더에 npm 환경변수 파일을 만들어 아래와 같이 저장한다.

```
[./.env]
```

```
01 SKIP_PREFLIGHT_CHECK=true
```

### 1.2.4 비주얼 스튜디오 코드와 플러그인 설치하기

#### ■ Reactjs code snippets 플러그인 설치하기

- 리엑트에서 자주 사용하는 코드 뭉치를 자동으로 완성해 준다.
- Extensions에서 'reactjs code snippets' 를 검색한 다음 플러그인을 설치한다.

#### ■ RCC.jsx 파일 생성

- 편집 화면에서 rcc라고 입력하여 목록이 나오면 rcc를 선택하여 자동으로 코드를 생성한다.
- 파일 이름이 RCC.jsx이므로 리액트 컴포넌트의 클래스 이름을 RCC로 만든다.

[./src/01/RCC.jsx]

```
01 import React, { Component } from 'react';
02
03 class RCC extends Component {
04   render() {
05     return (
06       <div>
07
08       </div>
09     );
10   }
11 }
12
13 export default RCC;
```

#### ■ Prettier 플러그인 설치하기

- 코드의 줄바꿈 등의 스타일을 자동으로 변환하여 프로젝트의 코드 입력 스타일을 동일하게 유지시켜준다.

## 1.3 리액트 앱 수정하기

#### ■ App.css 수정하기

- App.css 맨 아래에 다음 코드를 추가한다.

[./src/App.css의 일부]

```
01 ...(생략)...
02
03 .title {
04   font-style: italic;
05 }
```

#### ■ 스타일 반영하기

- 05: render() 함수는 HTML을 반환한다. HTML의 스타일 클래스 이름은 자바스크립트 클래스(class) 키워드와 같으므로 리액트에서는 class가 아니라 className으로 정의하여 사용한다.

[./src/App.js]

```
01 import React from 'react';
02 //import './App.css';
03 //import MyComponent from './01/RCC';
04 import MyComponent from './01/App';
05
06 class App extends React.Component {
07   render() {
08     return (
09       <div className="body">
10         <MyComponent />
```

## 리액트 프로그래밍 (프론트엔드 개발)

```
11     </div>
12   );
13 }
14 }
15
16 export default App;
```

- 리액트 핫 리로딩으로 변경된 화면 확인하기
  - 리액트 앱을 구동한 상태라면 파일을 저장한 즉시 화면이 바뀐다.

