2. 작업환경 준비

참고: 만약, 여러분이 현재 사용중인 컴퓨터에 프로그램을 설치 할 수 없는 환경이라면, Codesandbox 의 리액트 샌드박스 를 사용하여 개발을 하시면 됩니다.

앞으로 계속해서 튜토리얼을 진행하기 전에, 다음 항목들을 설치해주어야 합니다.

- **Node.js**: Webpack 과 Babel 같은 도구들이 자바스크립트 런타임인 Node.js 를 기반으로 만들어져있습니다. 그렇기에 해당 도구들을 사용하기 위해서 Node.js 를 설치합니다.
- **Yarn**: Yarn 은 조금 개선된 버전의 npm 이라고 생각하시면 됩니다. npm 은 Node.js 를 설치하게 될 때 같이 딸려오는 패키지 매니저 도구입니다. 프로젝트에서 사용되는 라이브러리를 설치하고 해당 라이브러리들의 버전 관리를 하게 될 때 사용하죠. 우리가 Yarn 을 사용하는 이유는, 더 나은 속도, 더 나은 캐싱 시스템을 사용하기 위함입니다.
- **코드 에디터**: 그리고, 코드 에디터를 준비하세요. 여러분이 좋아하는 에디터가 있다면, 따로 새로 설치하지 않고 기존에 사용하시던걸 사용하셔도 됩니다. 저는 주로 VSCode 를 사용합니다. 이 외에도, Atom, WebStorm, Sublime 같은 훌륭한 선택지가 있습니다.
- **Git bash**: 윈도우의 경우, Git for Windows 를 설치해서 앞으로 터미널에 무엇을 입력하라는 내용이 있으면 함께 설치되는 Git Bash 를 사용하세요. 윈도우가 아니라면 설치하지 않으셔도 상관없습니다. 설치는 기본 옵션으로 진행하시면 됩니다.

Webpack, Babel 은 무슨 용도인가요?

리액트 프로젝트를 만들게 되면서, 컴포넌트 를 여러가지 파일로 분리해서 저장 할 것이고, 또 이 컴포넌트는 일반 자바스크립트가 아닌 JSX 라는 문법으로 작성하게 됩니다. 여러가지의 파일을 한개로 결합하기 위해서 우리는 Webpack 이라는 도구를 사용하고, JSX 를 비롯한 새로운 자바스크립트 문법들을 사용하기 위해서 우리는 Babel 이라는 도구를 사용합니다.

Node.js

Windows 의 경우엔, Node.js 공식 홈페이지 에서 좌측에 나타나는 LTS 버전을 설치해주세요.

macOS / Linux 의 경우엔, nvm 이라는 도구를 사용하여 Node.js 를 설치하시는 것을 권장드립니다.

```
$ curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.8/install.sh
| bash
$ nvm install --lts
```

Yarn

만약 npm 이 이미 익숙하고, yarn 을 설치하기 귀찮다면 생략을 하셔도 상관없습니다.

yarn 설치는 Yarn 공식 홈페이지의 Install Yarn 페이지를 참고하세요.

VS Code

VS Code 를 설치 할 땐 VS Code 공식 홈페이지 를 참고하세요.

새 프로젝트 만들어보기

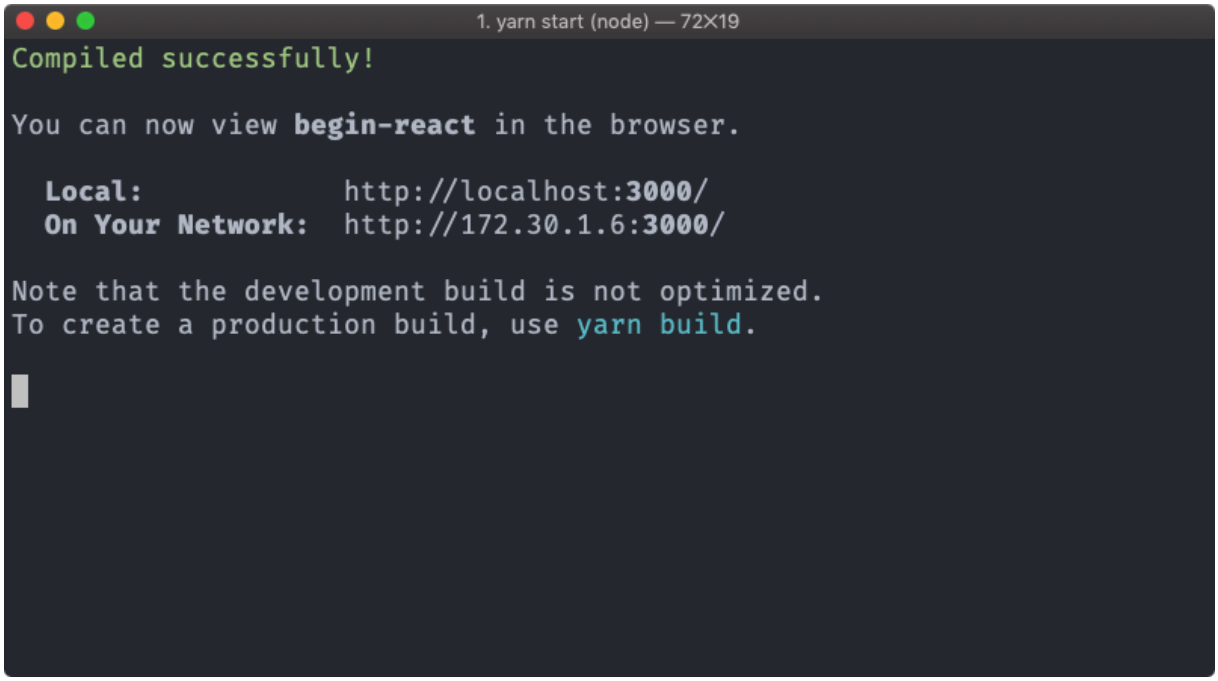
새로운 리액트 프로젝트를 만들어봅시다. 터미널을 열은 뒤, 다음 명령어를 실행해보세요. (윈도우 사용자는 Git Bash 를 사용하세요)

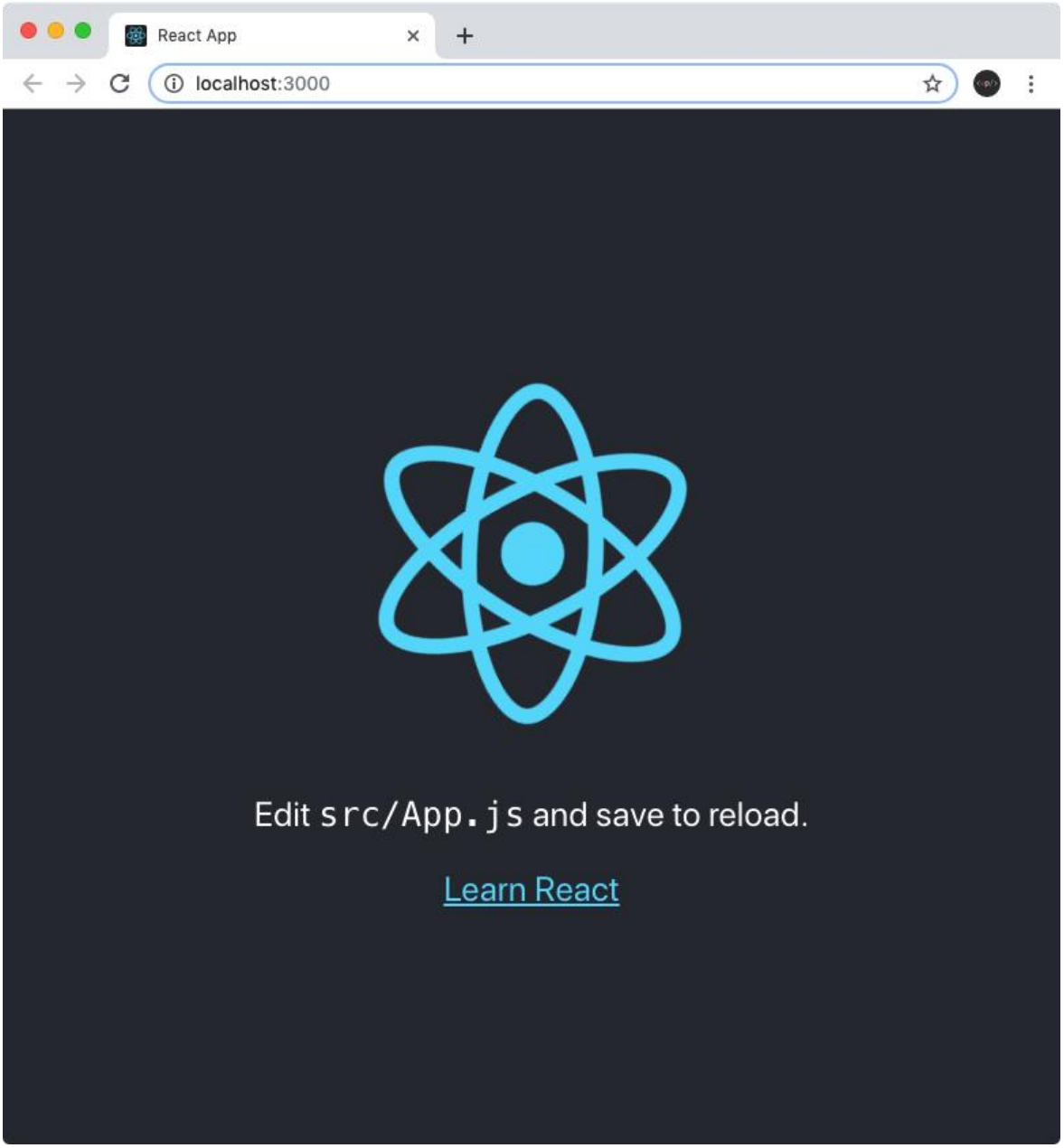
```
$ npx create-react-app begin-react
```

그러면 begin-react 라는 디렉터리가 생기고 그 안에 리액트 프로젝트가 생성됩니다. 생성이 끝나면 cd 명령어를 사용하여 해당 디렉터리에 들어간 다음 `yarn start` 명령어를 입력해보세요 (yarn 이 없다면 `npm start`).

```
$ cd begin-react
$ yarn start
```

이 명령어를 실행하고 나면 다음과 같이 브라우저에 http://localhost:3000/ 이 열리고, 돌아가는 리액트 아이콘이 보일 것입니다. 자동으로 페이지가 열리지 않는다면 브라우저에 주소를 직접 입력하여 들어가세요.

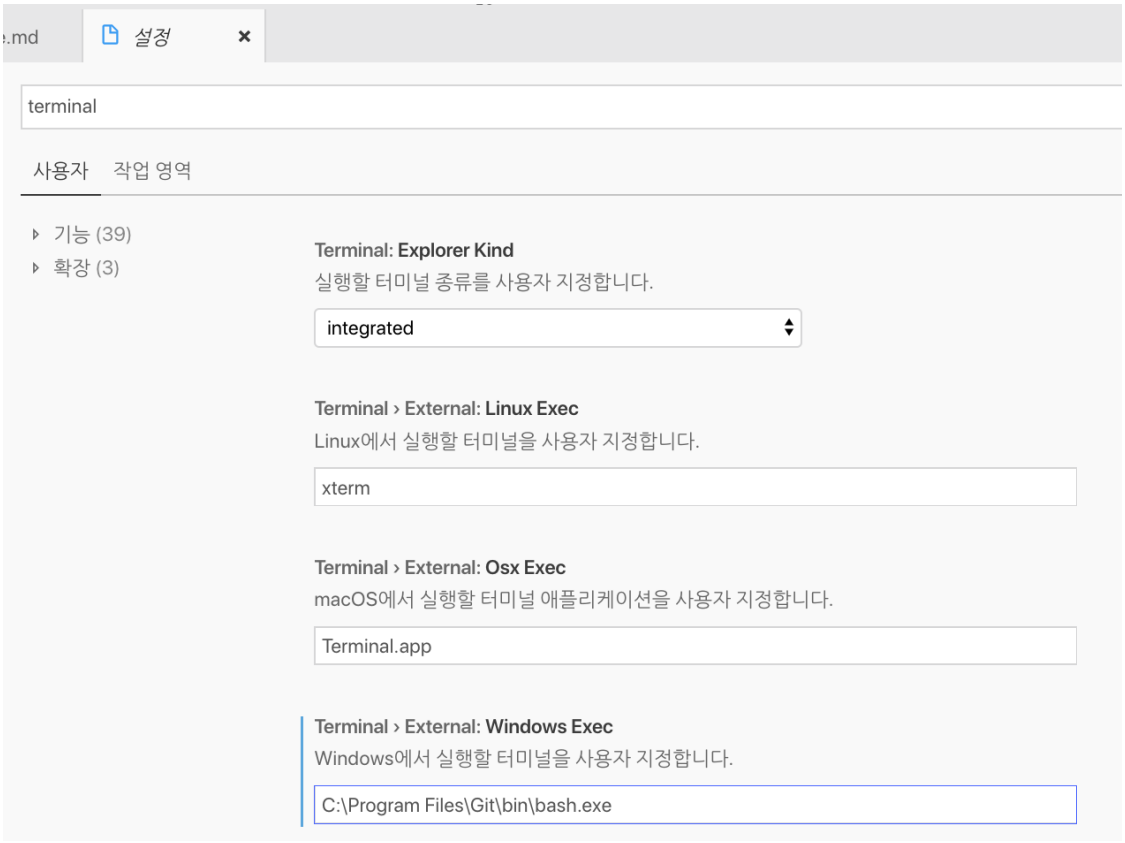

1. yarn start (node) — 72×19
Compiled successfully!
You can now view begin-react in the browser.
Local: http://localhost:3000/
On Your Network: http://172.30.1.6:3000/
Note that the development build is not optimized.
To create a production build, use yarn build.



VS Code 에서 터미널 띄우기

VS Code 내부에서 터미널을 띄울 수도 있습니다. VS Code 로 해당 디렉터리를 열은 뒤, 상단 메뉴의 View > Terminal 을 열으세요. (한글 메뉴의 경우 보기 > 터미널).

윈도우 사용자의 경우엔, 위 작업을 하기 전에 VS Code 에서 cmd 대신 Git Bash 를 사용하기 위하여 VS Code 에서 Ctrl + , 를 눌러 설정에 들어간 후, terminal 을 검색 후 Terminal > External > Windows Exec 부분에 Git Bash 의 경로인 C:\Program Files\Git\bin\bash.exe 를 넣어주세요.



Git Bash 를 열었을 때 기본 경로

Git Bash 에서의 ~/ 경로가 어디인지 모르신다면 `pwd` 명령어를 입력해보세요.