

React 환경설정하기

- Node.js 설치
- VS code 준비
- Chrome Browser 준비

1) Node.js 설치 – JavaScript 실행환경(Run Time)

공식홈페이지 : <https://nodejs.org/en>

The screenshot shows the Node.js website with the heading "Run JavaScript Everywhere". Below the heading, it states: "Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts." There is a green button that says "Download Node.js (LTS)". Below the button, it says: "Downloads Node.js v22.11.0¹ with long-term support. Node.js can also be installed via package managers. Want new features sooner? Get Node.js v23.1.0¹ instead." On the right side, there is a code editor showing a JavaScript code snippet for creating an HTTP server. The code is as follows:

```
1 // server.mjs
2 import { createServer } from 'node:http';
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:3000');
12 });
13
14 // run with 'node server.mjs'
```

Below the code editor, there is a "Copy to clipboard" button. At the bottom of the page, it says: "Learn more what Node.js is able to offer with our Learning materials."

디폴트로 설치한 후 node버전과 npm 버전을 확인한다.

Node.js를 설치하면 자동으로 npm은 함께 설치가 된다.

npm(Node Package Manager)는 프로젝트에서 필요한 다양한 외부 라이브러리들의 의존성을 관리 하고 편하게 설치 및 삭제까지 도와주는 역할을 한다.

설치완료 후 명령 프롬프트(CMD)에서 아래와 같이 확인 해 본다.

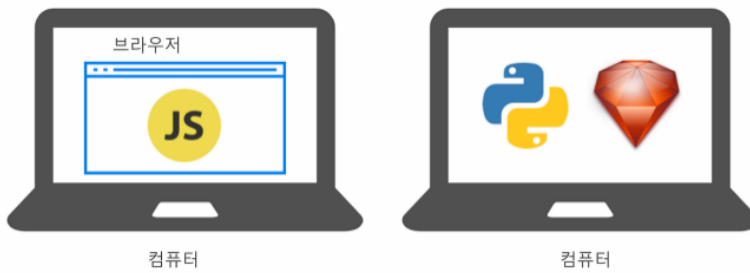
```
C:\Users\Admin>node -v
v20.11.0

C:\Users\Admin>npm -v
10.2.4

C:\Users\Admin>node --version
v20.11.0

C:\Users\Admin>npm --version
10.2.4
```

● Node.js설치이유

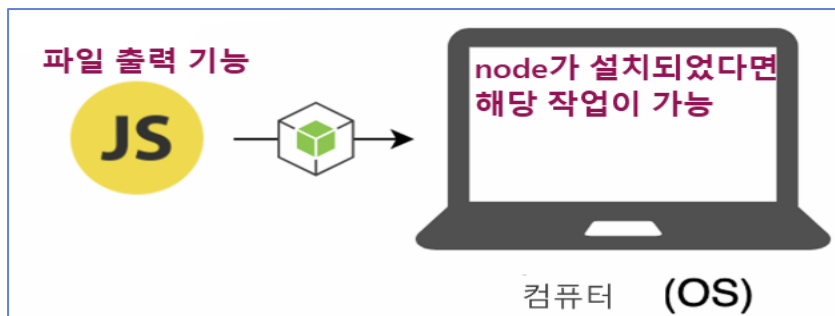


자바스크립트는 브라우저에서 동작하는 언어.

파이썬, Django는 컴퓨터의 OS기반(서버 사이드)에서 동작하는 언어.

브라우저라는 제한된 환경에서만 동작하던 자바스크립트를 파이썬 처럼 브라우저 바깥에서 동작하도록 하기 위해서는 Node.js 를 설치해야 한다.

브라우저에서 동작할 때는 가능하지 않았던 네트워크 작업등을 다룰 수 있게 된다.(파일 업로드)



Node.js는 서버 사이드의 JavaScript 실행환경

JavaScript로 Network Application을 만들수 있게 해주는 개발환경

Node.js

(JavaScript runtime)

Node.js를 위한 패키지 매니저
프로젝트에서 필요로하는 다양한 외부 라이브러리 버전과 의존성을 관리. 설치및 삭제

npm

(node package manager)

정리를 하면

Node.js는 자바스크립트로 네트워크 애플리케이션을 개발 할 수 있게 해주는 환경이고

npm은 node package manager의 약자로 Node.js를 위한 패키지 매니저이다.

패키지 매니저는 프로젝트에서 필요로 하는 다양한 외부 패키지들의 버전과 의존성을 관리하고
편하게 설치 및 삭제 할 수 있게 도와 주는 역할을한다.

👉 npm과 npx의 차이점

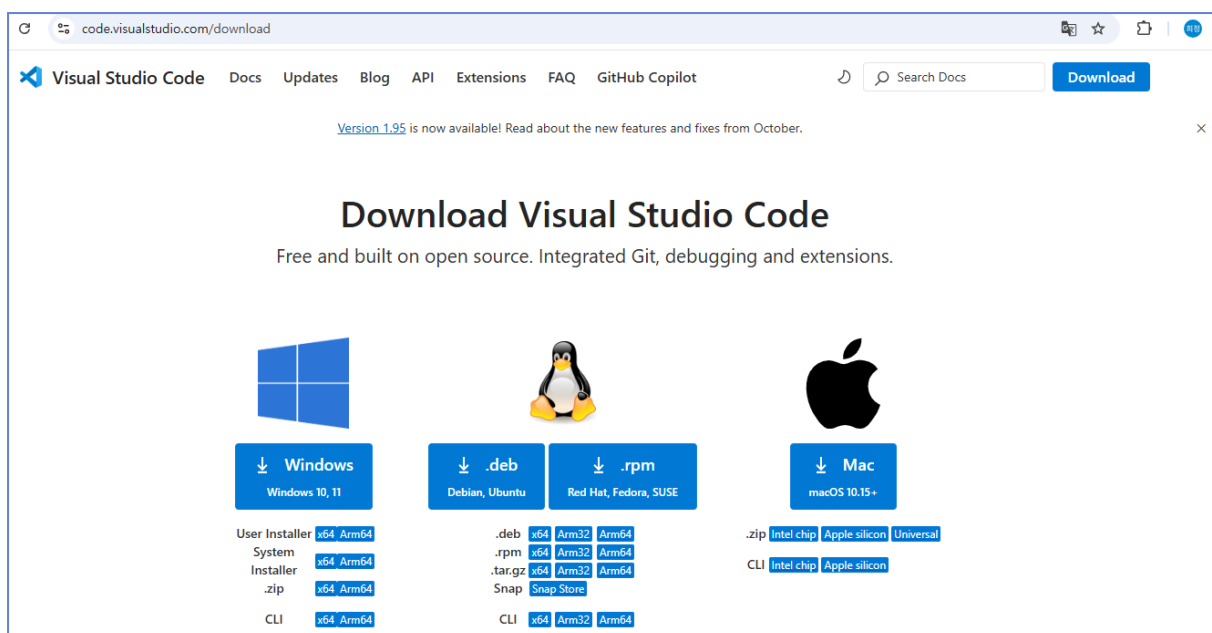
npm은 내 컴퓨터 안에 글로벌한 공간에 모듈을 설치해 프로젝트 마다 같은 모듈을 공유해서
사용 할 수 있다. 이는 모듈이 업데이트 되었는지 확인이 어렵고 변동사항이 잦은 프로젝트에 영
향을 미칠 수 있다.

: npm 5.2 버전부터 npx가 기본 패키지로 제공되기 시작 했으며 모듈을 로컬에 저장하지 않고 매
번 최신버전의 파일만을 임시로 불러와 실행 시킨 후 다시 그 파일은 없어지는 방식이다.

2) VSCode 설치하기

마이크로소프트사에서 개발했고 무료로 사용 가능한 IDE

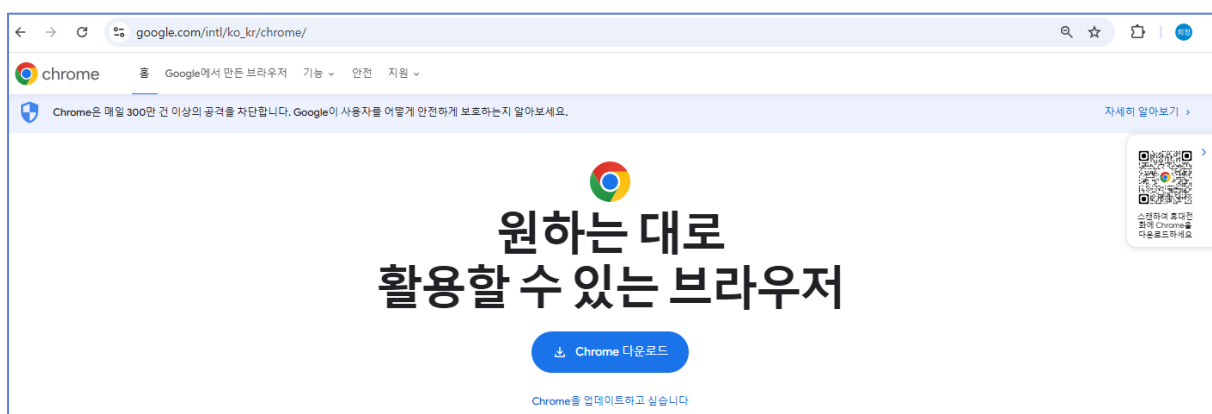
<https://code.visualstudio.com/download>



VS Code의 확장 프로그램(플러그인) 설치

- 1) ESLint - 컴파일 시 에러를 바로 확인할 수 있게 하는 것
- 2) Reactjs code snippets - react코드 자동완성 되게 하는 것
- 3) JavaScript (ES6) code snippets - js코드 자동완성 하는 것
- 4) Prettier - Code formatter - 코드를 원하는 스타일에 맞춰 다듬어 주는 도구
- 5) Relative Path - 상대 경로에 있는 파일 경로를 편하게 작성할 수 있는 도구
- 6) Guides - 들여쓰기 가이드라인을 그려 주는 도구
- 7) vscode-js-import - import 기능, 제작자가 wangtao0101)
- 8) CSS Formatter - 정해진 규칙(코딩 컨벤션)에 맞추어서 코드를 정렬하는 도구
- 9) vscode-styled-components - ctrl+spacebar를 통해 해당 스타일 코드(CSS)를 어시스트해주는 도구
- 10) Thunder Client - VSCode 내에서 HTTP 요청을 생성하고 테스트 할수 있는 도구 (postman 유사)

3) Chrome Browser 준비



React 프로젝트 생성하기 , 실행하기

React 애플리케이션을 빠르고 효율적으로 시작 할 수 있도록 도와주는 도구로 CRA와 Vite가 있다. CRA와 Vite는 프로젝트 설정 및 빌드 도구이다.

비교 항목	Create React App (CRA)	Vite
설정	자동으로 설정 제공, 설정을 변경하려면 <code>eject</code> 필요	설정이 간단하고, 커스터마이징이 쉽다.
개발 서버 속도	상대적으로 느림	빠른 HMR(핫 모듈 교체), 즉각적인 반응성
빌드 시간	상대적으로 느림	빠른 빌드, 최신 모듈 기반으로 성능 최적화
빌드 도구	웹팩(Webpack)을 사용	Rollup 사용, 빌드 최적화 성능 우수
브라우저 호환성	모든 브라우저에서 잘 작동	최신 브라우저에서 우수, 구형 브라우저 호환성 문제 가능
플러그인	웹팩 플러그인 사용	Vite 플러그인 시스템, 다양한 플러그인 지원

CRA : 안정적이고 보편적인 React 프로젝트 설정이 필요할 때. 특히 프로젝트가 복잡하지 않거나, 기존의 React 생태계와 잘 호환되는 설정을 원할 때 좋다.

Vite : 빠른 개발 환경과 빌드를 원할 때 최신 웹 기술을 활용하여 성능을 최적화하고자 할 때, 개발 속도나 빌드 시간이 중요한 경우 Vite가 좋다.

VS Code에서 CRA 도구로 React프로젝트 만들기

터미널 창을 연다 (View>Terminal혹은 **Ctrl + `**)

경로를 React_Study 로 이동한다.

```
PS C:\React_Study> 
```

-현재 디렉토리에서 리액트 개발 프로젝트 생성

👉 프로젝트 생성방법

```
>> npx create-react-app [project-name]
```

주의 : 프로젝트명(패키지이름) 대문자허용 안함.

<https://stackoverflow.com/questions/59159553/why-cant-react-project-name-contain-capital-letters>

<https://github.com/facebook/create-react-app/issues/2165>

```
PS C:\React_Study> npx create-react-app my-app
Need to install the following packages:
create-react-app@5.0.1
Ok to proceed? (y) y
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported,
updates. Please upgrade asap.

Creating a new React app in C:\React_Study\my-app.

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

create-react-app은 React의 Boiler Plate 이다. 초기환경을 일일이 설정하지 않고도 React프로젝트를 시작 할 수 있도록 셋업을 완료해 놓은 틀이다.

만약,

npx create-react-app my-app 명령 후 아래와 같이 에러메시지가 나온 경우 create-react-app 제거 후 재설치 한다.

```
npm ERR! code ENOENT
npm ERR! syscall lstat
npm ERR! path C:\Users\WKOSTA\AppData\Roaming\npm
npm ERR! errno -4058
npm ERR! enoent ENOENT: no such file or directory, lstat
'C:\Users\WKOSTA\AppData\Roaming\npm'
npm ERR! enoent This is related to npm not being able to find a file.
npm ERR! enoent
npm ERR! A complete log of this run can be found in: C:\Users\WKOSTA\AppData\Local\npm-cache\logs\2024-01-29T14_09_10_696Z-debug-0.log
```

```

명령어 > npm uninstall -g create-react-app
명령어 > npm install -g create-react-app
명령어 > npx create-react-app project-name

```

만약(node version 22이후 이슈) ,

```
npx create-react-app step01-cra
```

명령어 실행 이후 아래와 같은 오류가 발생한다면,

```

PS C:\Edu\React\Work> npx create-react-app step01-cra
npx : 이 시스템에서 스크립트를 실행할 수 없으므로 C:\Program Files\nodejs\npx.ps1 파일을 로드할 수 없습니다. 자세한 내용은 about_Execution_Policies(https://go.microsoft.com/fwlink/?LinkID=135170)를 참조하십시오.
위치 줄:1 문자:1
+ npx create-react-app step01-cra
+ ~~~
+ CategoryInfo          : 보안 오류: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess

```

📖 chatgpt 설명

문제 설명

PowerShell은 보안상의 이유로 기본적으로 스크립트 실행을 제한하는 **Execution Policy**를 가지고 있습니다. 이 설정 때문에 외부 스크립트를 실행하려고 하면 위와 같은 오류가 발생할 수 있습니다. **npx** 명령어는 Node.js에서 사용하는 명령어인데, **npx.ps1** 파일이 PowerShell에서 실행되지 않으므로 문제가 발생한 것입니다.

해결 방법

1. PowerShell을 관리자 권한으로 실행

PowerShell을 관리자 권한으로 실행해야 합니다. 아래 방법을 따라 진행하세요:

- 시작 메뉴에서 **PowerShell**을 검색하고, 관리자 권한으로 실행을 클릭합니다.

2. Execution Policy 변경

PowerShell에서 **npx**를 실행할 수 있도록 **Execution Policy**를 변경합니다. **Set-ExecutionPolicy** 명령어를 사용하여 **RemoteSigned** 또는 **Bypass**로 설정할 수 있습니다.

- PowerShell에서 아래 명령어를 입력하고 Enter 키를 누릅니다:

📖 해결 방법 - 아래의 명령어를 실행한 후 다시 프로젝트를 생성한다.

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

만약,

사용자 이름이 한글 이어서 React 프로젝트 생성이 안된다면,

아래의 블로그를 참고해주세요.

<https://blog.naver.com/rkdalstj7504/222173490548>

리액트 프로젝트 생성이 끝나면 애플리케이션을 실행시킬 수 있는 명령어가 보인다

```
npm start
  Starts the development server.

npm run build
  Bundles the app into static files for production.

npm test
  Starts the test runner.

npm run eject
  Removes this tool and copies build dependencies, configuration files
  and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

cd my-app
npm start

Happy hacking!
```

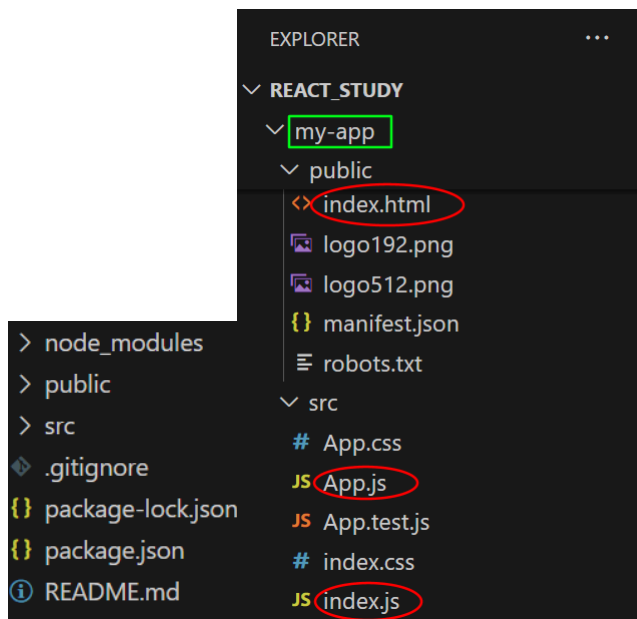
프로젝트 폴더이동

cd 프로젝트이름

프로젝트 실행

npm start

VS Code Explore에서 생성된 프로젝트 구조와 핵심 파일



my-app <

public < index.html

src < App.js , index.js

우리가 아무것도 하지 않아도

기본적인 것들이 생성

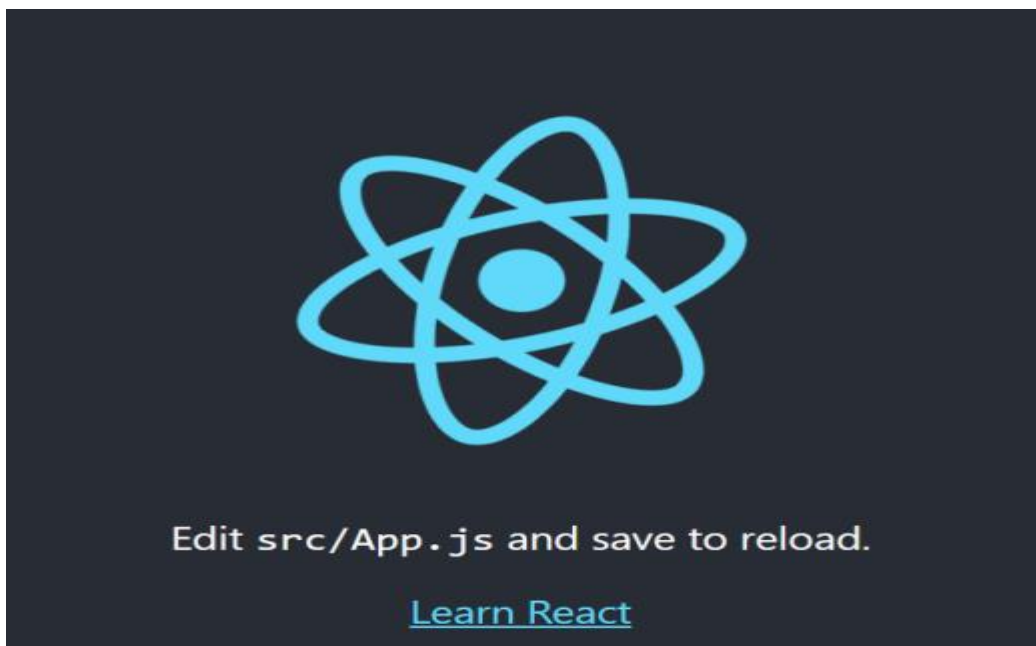
이 상태에서 바로 프로젝트 실행할 수 있다.

-VS Code 터미널에서 생성된 프로젝트 my-app으로 경로를 옮긴 후 애플리케이션을 실행

```
$ cd my-app
```

```
$ npm start
```

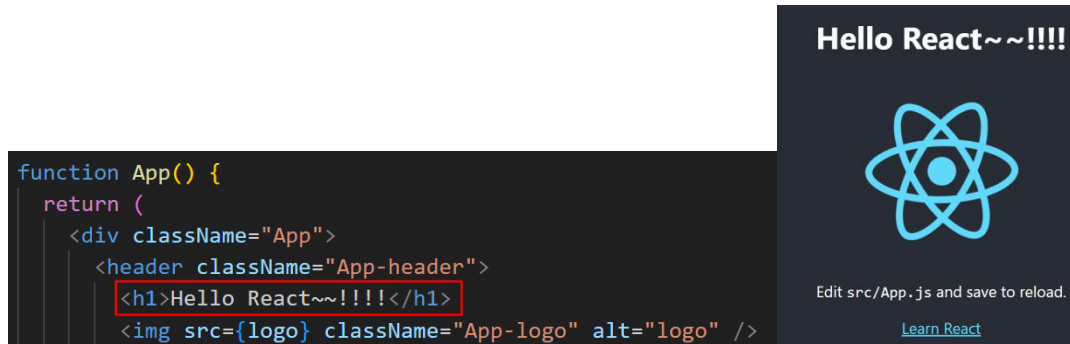
브라우저에 <http://localhost:3000>



실행화면 수정 (실행화면 흐름 이해)

1) App.js

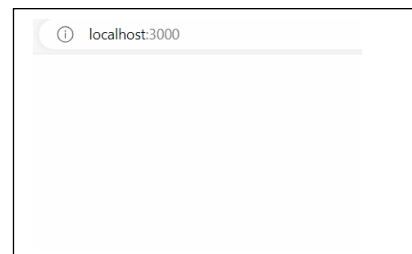
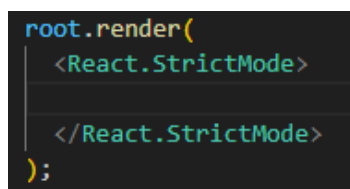
function App() 함수 부분이 바로 클라이언트 UI 화면을 구성하는 실제 내용



- ➔ 화면 세로고침 하지 않아도 갱신
- ➔ App.js파일의 디자인은 App.css에서 관장
- ➔ npm start는 프로젝트 실행과 동시에 수정된 부분을 다시 반영하는 기능도 동시에 함.

2) index.js

index.js파일의 <App />를 지웠다가 다시 살펴보자

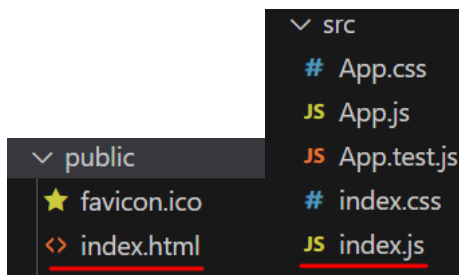


UI가 변하는 것을 알 수 있다. UI의 실제 화면은 <App />을 따라가보면

`import App from './App';` 이 부분이다.(./APP 뒤에는 js확장자 생략)

4. 프로젝트 안의 각 파일 역할

index.html | index.js | App.js



-코드에서 가장 먼저실행

-브라우저에서 실행

-main 역할을 함

-index.html파일 다음으로 실행

-리액트 코드중 가장 먼저 실행

index.html ---- DOM 생성(root)

index.js ----- **html파일의 root부분에 App.js의 UI를 연결함**

App.js ----- 화면 UI 구성

React의 중요한 파일 3가지

1. index.js
2. App.js
3. index.html

1. index.js

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

2. App.js

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

3. 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 바꿀시 오류발생)

개발모드 종료하기

터미널 창을 보면 멈춰있는 것처럼 보이지만 사실은 개발모드가 계속 실행 중이다.

3000이라는 포트로 우리 개발 모드에 언제든지 접근하도록 실행되고 있는 상태

터미널 창에서 Ctrl + C 로 개발모드를 종료

개발모드가 종료되면 웹 브라우저 페이지가 더 이상 동작하지 않는 것을 확인

(브라우저 새로고침 후 확인!!)

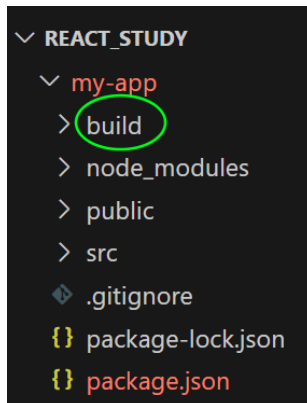
배포하기 npm run build

아래 작업을 하기 전에 index.js를 원래대로 해놓는다.<App/>

```
PS C:\React_Study\my-app> npm run build

> my-app@0.1.0 build
> react-scripts build
```

→ build폴더가 생성



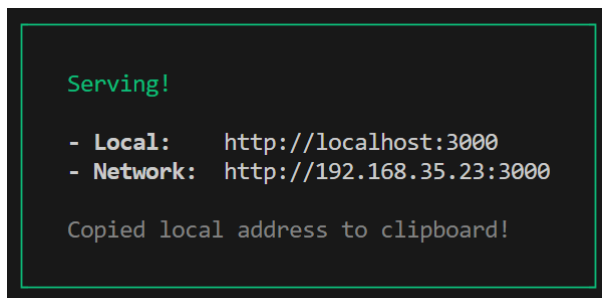
Index.html 파일에는 공백조차 없다 개발과 다르게 배포 시에는 용량을 최적화 시키기 위함이다.

이 빌드 된 파일을 이제 실행해보자

```
PS C:\React_Study\my-app> npx serve -s build
Need to install the following packages:
serve@14.2.1
Ok to proceed? (y) y
```

serve는 웹 서버를 의미

옵션 -s는 사용자가 어떤 요청의 경로로 들어오든 간에 무조건 index.html을 요청하도록 해준다.



웹 서버 실행(접속주소가 뜬다.)

➔ 개발환경의 서비스가 아니라 실제 서비스 환경이 만들어진다.

VS Code에서 Vite 도구로 React프로젝트 만들기

기존 방식인 CRA는 세팅에 시간이 오래 걸리고 다소 무겁다는 특징이 있다

Vite를 이용하면 세팅도 빠르고 개발 단계에서도 수정사항을 빠르게 반영하여 실행 화면을 업데이트 해주기 때문에 매우 유용하다

Vite 사용을 위해선 Node.js가 설치되어 있어야 한다.

아래 명령어로 프로젝트 생성

```
>> npm create vite@latest [프로젝트이름]
```

```
C:\Edu\React_2867\Reactwork>npm create vite@latest step02_vite
Need to install the following packages:
create-vite@5.5.5
Ok to proceed? (y) █
```

framework 선택화면에서 위아래 방향키로 React 선택

```
? Select a framework: » - Use arrow-keys. Return to submit.
  Vanilla
  Vue
>  React
  Preact
  Lit
  Svelte
  Solid
  Qwik
  Others
```

다음 단계에서 JavaScript 선택

```

✓ Select a framework: » React
? Select a variant: » - Use arrow-keys. Return to submit.
  TypeScript
  TypeScript + SWC
>  JavaScript
   JavaScript + SWC
   Remix ↗

```

프로젝트 생성이 완료되면 아래와 같이 실행할 명령어를 보여주며, 순서대로 실행

```

✓ Select a framework: » React
✓ Select a variant: » JavaScript

Scaffolding project in J:\development\react\react_class\ex02\ex03...

Done. Now run:

  cd ex03
  npm install
  npm run dev

```

```

프로젝트 폴더로 이동
>> cd [프로젝트이름]

의존성 설치
>> npm install

실행
>> npm run dev

```

실행하면 아래와 같이 보이며, 브라우저가 자동으로 실행되지 않으므로
브라우저 주소창에 `http://localhost:5173` 입력하여 실행
또는
콘솔 창에서 `o+ enter`

```
VITE v5.3.4 ready in 444 ms

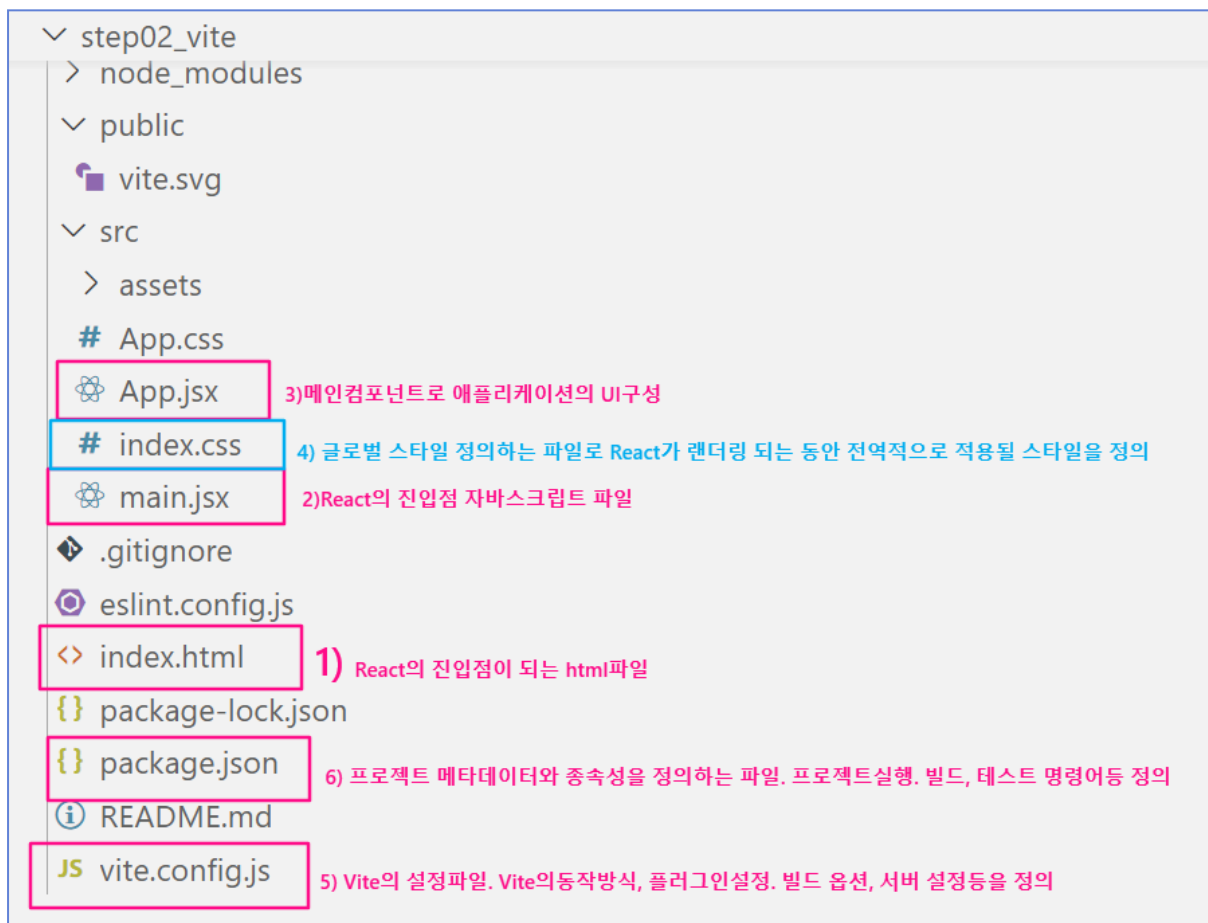
→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

접속 화면



프로젝트 구동 순서

src/index.html > src/main.jsx > src/App.jsx



Vite 로 React 프로젝트를 생성하면, 다음과 같은 파일들이 자동으로 생성된다.

- **index.html** : React 애플리케이션의 진입점인 HTML 파일
- **src/main.jsx** : React 애플리케이션의 진입점인 JavaScript 파일
- **src/App.jsx** : 기본적인 App 컴포넌트
- **src/index.css** : 전역 CSS 스타일
- **vite.config.js** : Vite 설정 파일
- **package.json** : 프로젝트 설정 및 종속성
- **node_modules/** : 설치된 npm 패키지
- **public/** : 정적 파일 (이미지, favicon 등)
- **.gitignore**: 버전 관리에서 제외할 파일 설정

이 파일들 덕분에 Vite 는 React 애플리케이션을 매우 간단하고 빠르게 설정할 수 있도록 도와준다. Vite 는 성능과 개발 경험을 개선하는 데 중점을 두고 있기 때문에, 기본적으로 제공되는 설정을 활용하여 빠르게 개발을 시작할 수 있다.

Vite 프로젝트 배포하기

명령어 > npm run build

The screenshot shows the VS Code interface with the `vite.config.js` file open. The `build` script in the `scripts` object is highlighted with a pink box. Below the editor, the terminal shows the output of the `npm run build` command, which is also highlighted with a pink box and an arrow pointing to it. The terminal output shows the Vite build process for production, including the transformation of 34 modules and the final build time of 725ms.

```

4  "version": "0.0.0",
5  "type": "module",
6  "scripts": {
7    "dev": "vite",
8    "build": "vite build",
9    "lint": "eslint .",
10   "preview": "vite preview"
11 },
12 "dependencies": {

```

```

PS C:\Edu\React\TestWork\step02-vite> npm run build → Vite 프로젝트 빌드 명령어

> step02-vite@0.0.0 build
> vite build

vite v5.4.10 building for production...
✓ 34 modules transformed.
dist/index.html                0.46 kB | gzip: 0.30 kB
dist/assets/react-CHdo91hT.svg 4.13 kB | gzip: 2.05 kB
dist/assets/index-n_ryQ3BS.css 1.39 kB | gzip: 0.71 kB
dist/assets/index-CYUN_lC9.js 142.55 kB | gzip: 45.80 kB
✓ built in 725ms

```

명령어 > npm run preview

The screenshot shows the terminal output of the `npm run preview` command. The command is highlighted with a pink box and an arrow pointing to it. The terminal output shows the Vite preview process, including the local URL `http://localhost:4173/` and instructions on how to use the `h` key and `enter` to show help.

```

PS C:\Edu\React\TestWork\step02-vite> npm run preview → vite프로젝트 실행 명령어

> step02-vite@0.0.0 preview
> vite preview

→ Local:   http://localhost:4173/
→ Network: use --host to expose
→ press h + enter to show help

```

리액트 개발자도구 설치(크롬의 확장 프로그램 - react developer tools)

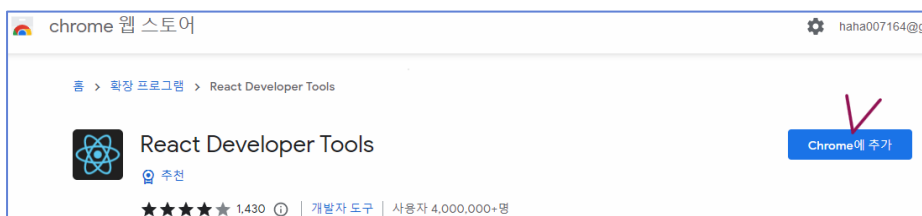
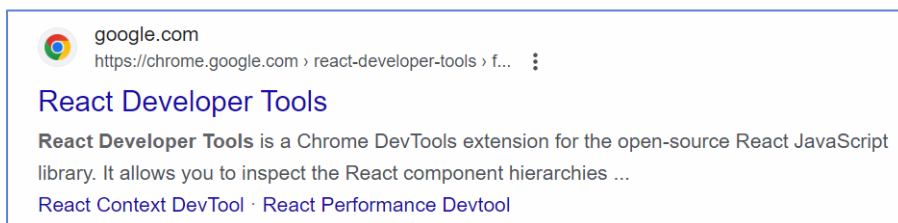
이 도구는 React 애플리케이션의 구조와 상태를 실시간으로 시각화하고, 컴포넌트의 동작을 추적할 수 있는 강력한 기능을 제공한다. 이를 통해 개발자는 애플리케이션의 성능을 개선하고, 버그

를 효율적으로 디버깅할 수 있다

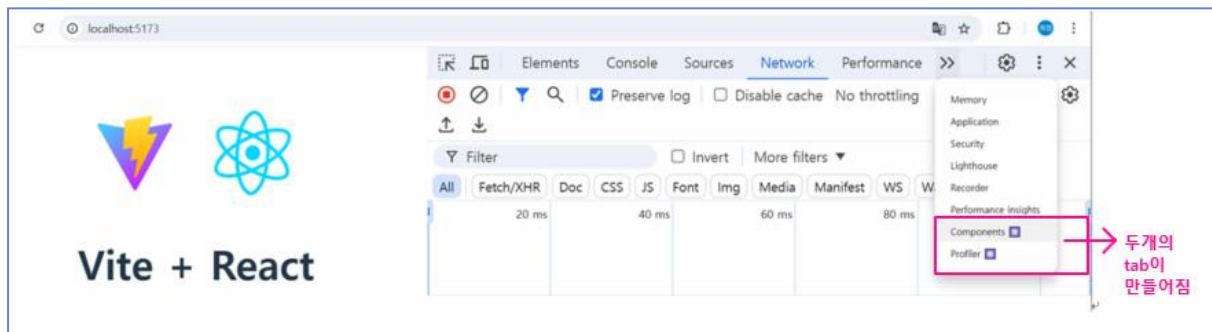
주요기능

- React 컴포넌트 트리 탐색
- Props와 State 확인
- Hooks 디버깅
- Component Re-render 추적
- Profiler (성능 분석)
- Context 확인
- 컴포넌트 렌더링 최적화
- Error Boundaries 추적

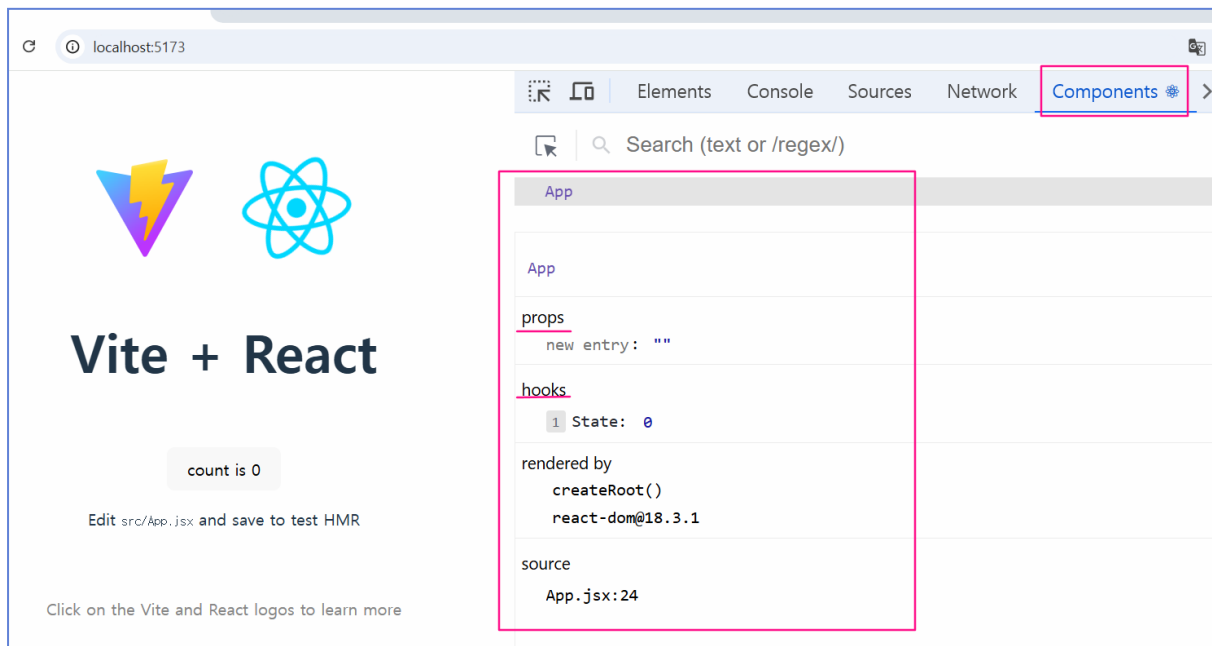
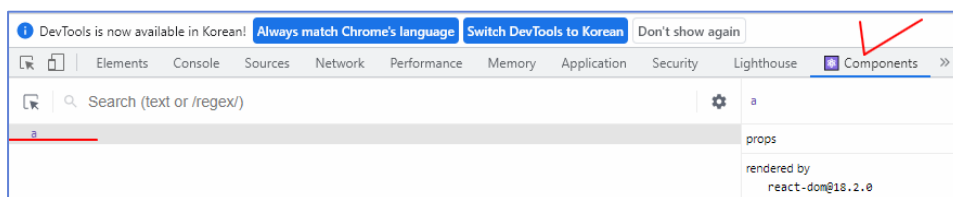
구글검색 < react developer tools



설치가 정상적으로 끝나면 , 다시 리액트 사이트 돌아가서 새로 고침 / 개발자도구도 닫았다 다시 열어본다.



2개의 새로운 탭이 만들어졌다.



2022년 3월 8일 이후로 리액트의 18버전이 나오면서 Concurrent Rendering, Automatic Batching, Suspense for Data Fetching, Server Components 등 여러 혁신적인 기능을 도입하여 React 애플리케이션의 성능을 크게 향상시켰다. 특히 사용자 경험과 성능을 중요한 요소로 다룬

이번 업데이트는 더 빠르고 효율적인 웹 애플리케이션 개발을 가능하게 한다.

Tip.

React프로젝트에서 컴포넌트 만들 때 파일이름의 확장자는 **Xxx.js orXxx.jsx** 상관이 없다.

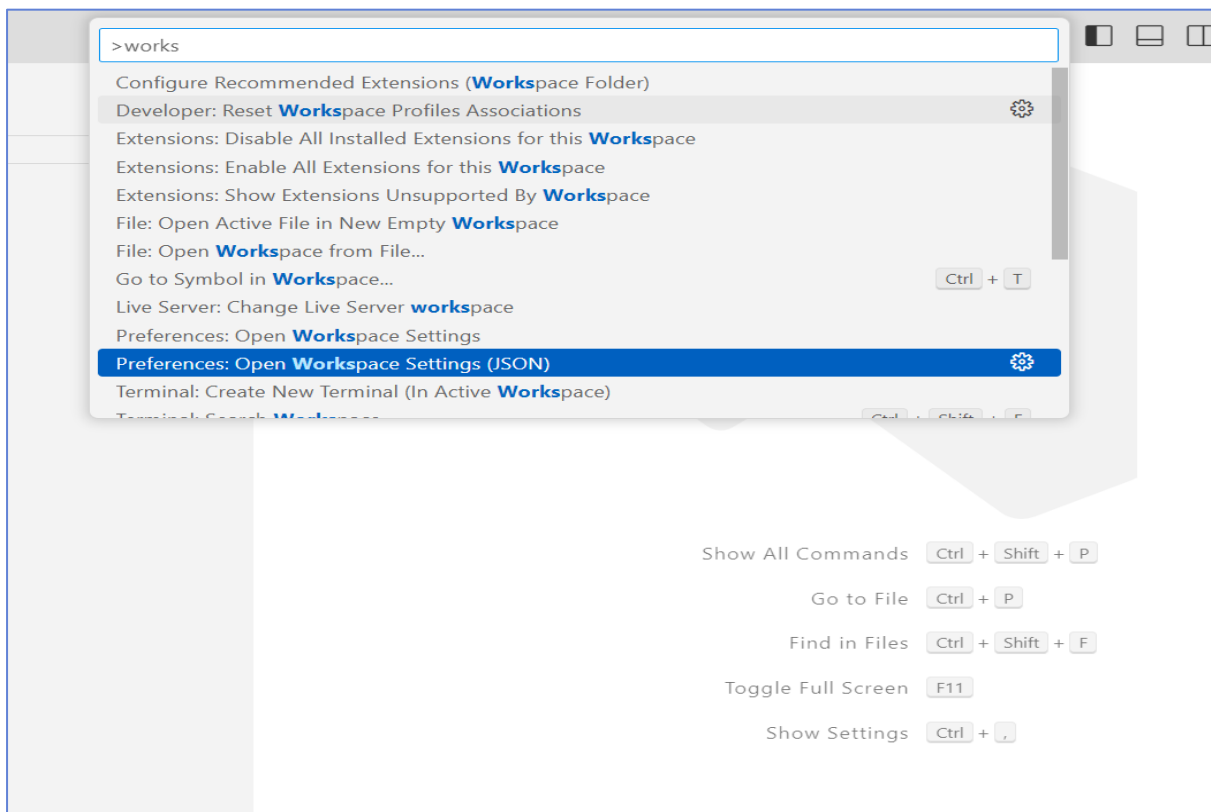
그러나, React에 관련 import를 자동완성을 쓰기 위해서는 jsx파일로 저장해야 한다.

만약,

js파일에서도 자동완성이 가능하도록 하기 위해서는 설정이 필요하다.

1. **Ctrl + shift + p**를 누른다.

2. **Works**를 검색한다.



3. **WorkSpace Settings(JSON)**을 클릭한다.

아래의 내용을 추가한다.

```
{  
  "emmet.syntaxProfiles": {  
    "javascript": "jsx"  
  },  
  "emmet.includeLanguages": {  
    "javascript": "html"  
  }  
}
```