

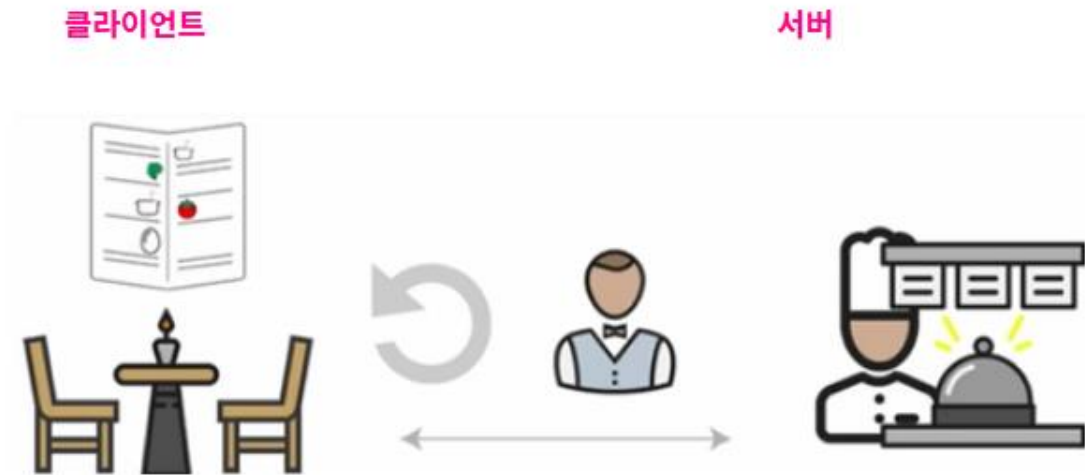
# 1강 개발환경 설정





## ➤ Node.js 란

- Node.js는 브라우저 밖에서도 JavaScript를 실행할 수 있게 해주는 런타임 환경이다.
- Chrome V8 엔진을 기반으로 만들어져 매우 빠르다.
- 원래는 서버 개발용으로 만들어졌지만, 지금은 React 등 프론트엔드 개발환경 구성에도 필수적으로 사용된다.



- Node.js를 통해 명령어로 패키지를 설치하고, 개발 서버를 실행하고, 빌드 작업을 자동화할 수 있다.



## ➤ Node.js 란

- Node.js는 단순한 "서버용 도구"가 아니라,
  - React 개발을 위한 도구 실행 환경으로서의 역할이 중요하다.
1. JavaScript 실행 환경 제공  
→ 브라우저 외부에서도 JS 실행 가능 (예: 서버, 터미널)
  2. 개발 도구 실행 환경 제공  
→ Babel, Webpack, Vite 등 최신 프론트엔드 빌드 도구는 Node.js 위에서 동작
    - CRA = 옛날 세대의 빌드 도구 (webpack 기반)
    - Vite = 최신 세대의 빌드 도구 (ESM + 번개처럼 빠른 HMR)
  3. 패키지 관리 도구 제공 (npm)  
→ 수많은 라이브러리 설치 및 관리

React 프로젝트는 Node.js 위에서 npm(또는 yarn, pnpm)을 이용해 패키지를 설치하고 빌드한다.

Node.js = 실행엔진

npm = 패키지 관리자

React = 코드



## ➤ React는 브라우저에서 동작하는 라이브러리

- React는 브라우저에서 동작하는 JavaScript UI 라이브러리다.
- 단순히 `<script>` 태그로도 사용할 수 있다.
- 하지만 JSX 문법, import/export, 빌드 기능을 사용하려면 Node.js 환경이 필요하다.

예시 ) Node.js 없이 React 사용

```
<script src="https://unpkg.com/react@18/umd/react.development.js"> </script>  
<script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"> </script>
```



➤ React 개발 환경을 만들 때 **Node.js**가 필요한 이유

- React 프로젝트를 만들고 실행하기 위해서는 다음의 과정이 모두 Node.js 환경에서 수행된다.

1. JSX 변환 (Babel) — JSX → JS 코드 변환
2. 번들링 (Webpack, Vite 등) — 여러 JS 파일을 하나로 묶기
3. 개발 서버 실행 — npm run dev, npm start
4. 패키지 설치 및 관리 — npm(Node Package Manager)
5. 프로젝트 생성 도구 사용 — npx create-react-app, **npm create vite@latest**
6. 빌드(배포용 코드 압축 및 최적화)
7. Node.js가 없으면 위 명령어를 실행할 수 없다.

# node.js 설치





- node.js( JavaScript runtime ) : 자바스크립트로 네트워크 애플리케이션을 개발 할 수 있도록 하는 환경
- npm( node package manager ) : node.js를 위한 패키지 매니저 이다. Node.js 설치 시 자동으로 설치된다.

<https://nodejs.org/ko/download>

## Node.js® 다운로드

Node.js® v22.16.0 (LTS) 를 Windows 환경에서 Docker 방식으로 npm

를(을) 사용해 설치하세요.

```
1 # Docker는 각 운영 체제별로 설치 지침을 제공합니다.
2 # 공식 문서는 https://docker.com/get-started/에서 확인하세요.
3
4 # Node.js Docker 이미지를 풀(Pull)하세요:
5 docker pull node:22-alpine
6
7 # Node.js 컨테이너를 생성하고 쉘 세션을 시작하세요:
8 docker run -it --rm --entrypoint sh node:22-alpine
9
10 # Node.js 버전 확인:
11 node -v # "v22.16.0"가 출력되어야 합니다.
12
13 npm 버전 확인:
14 npm -v # 10.9.2가 출력되어야 합니다.
```

PowerShell

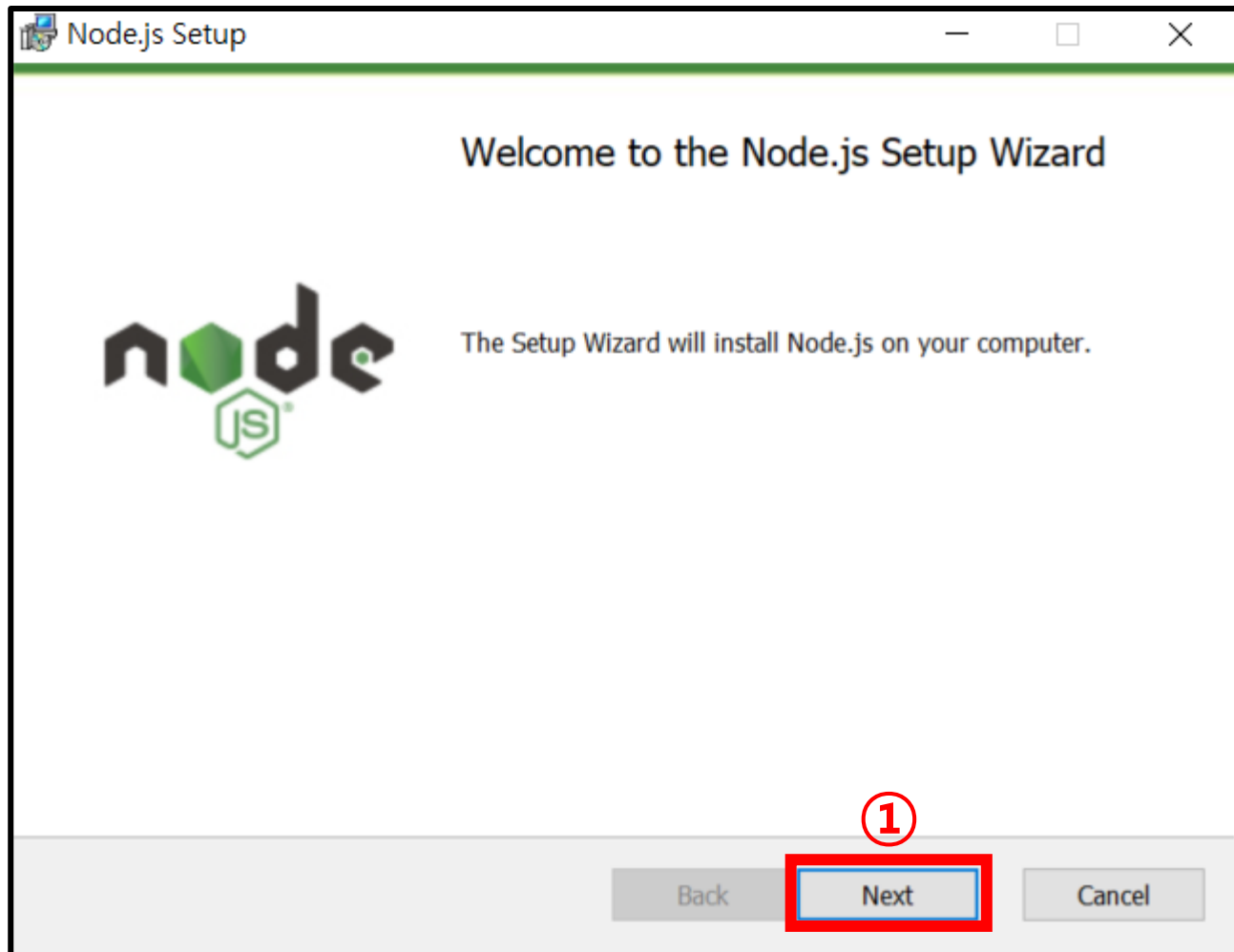
클립보드에 복사

Docker는 컨테이너화 플랫폼입니다. 문제가 발생하면 Docker's 웹사이트를 방문하세요.

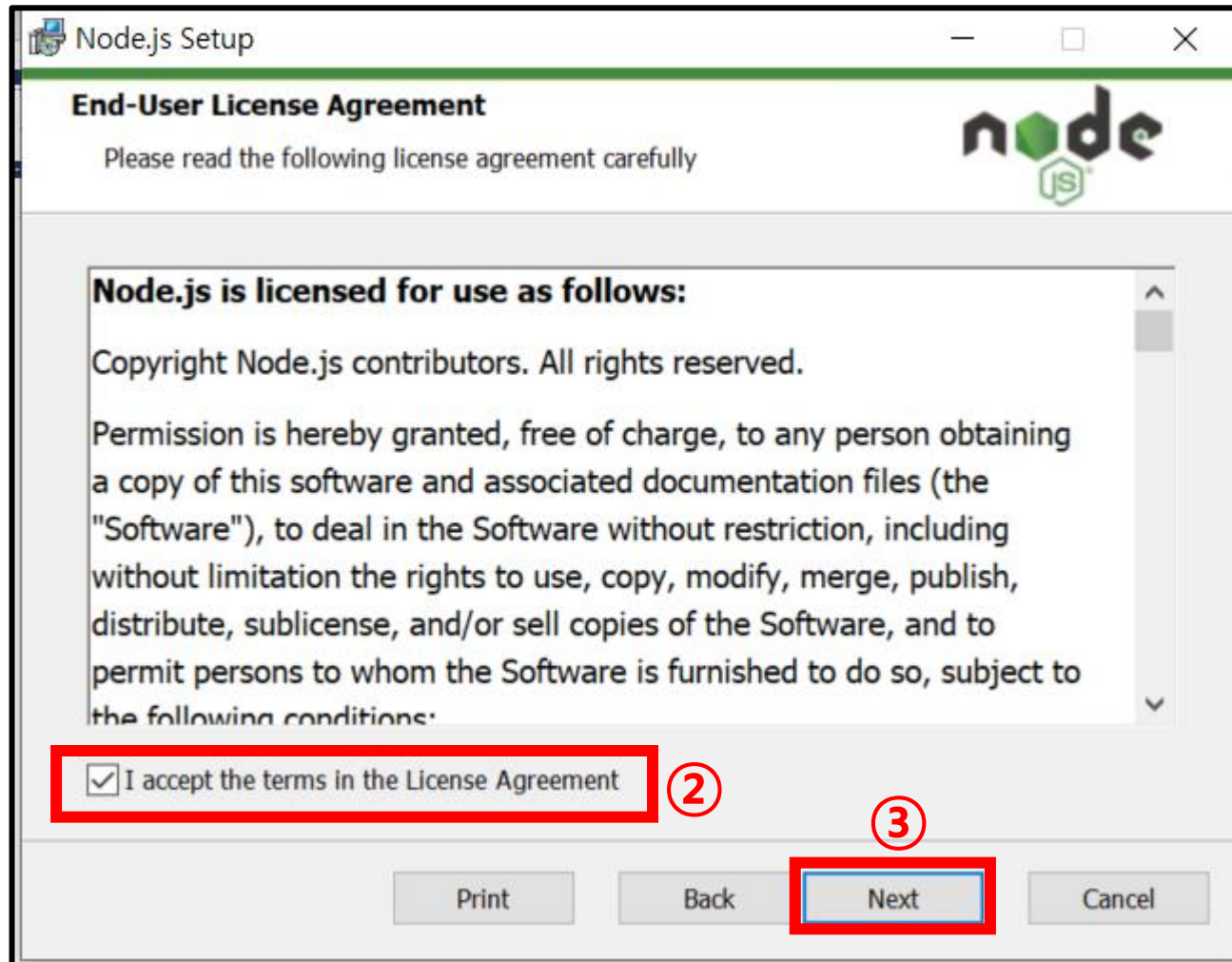
또는 x64 아키텍처가 실행 중인 Windows 환경에서 미리 빌드된 Node.js®를 다운로드하세요.

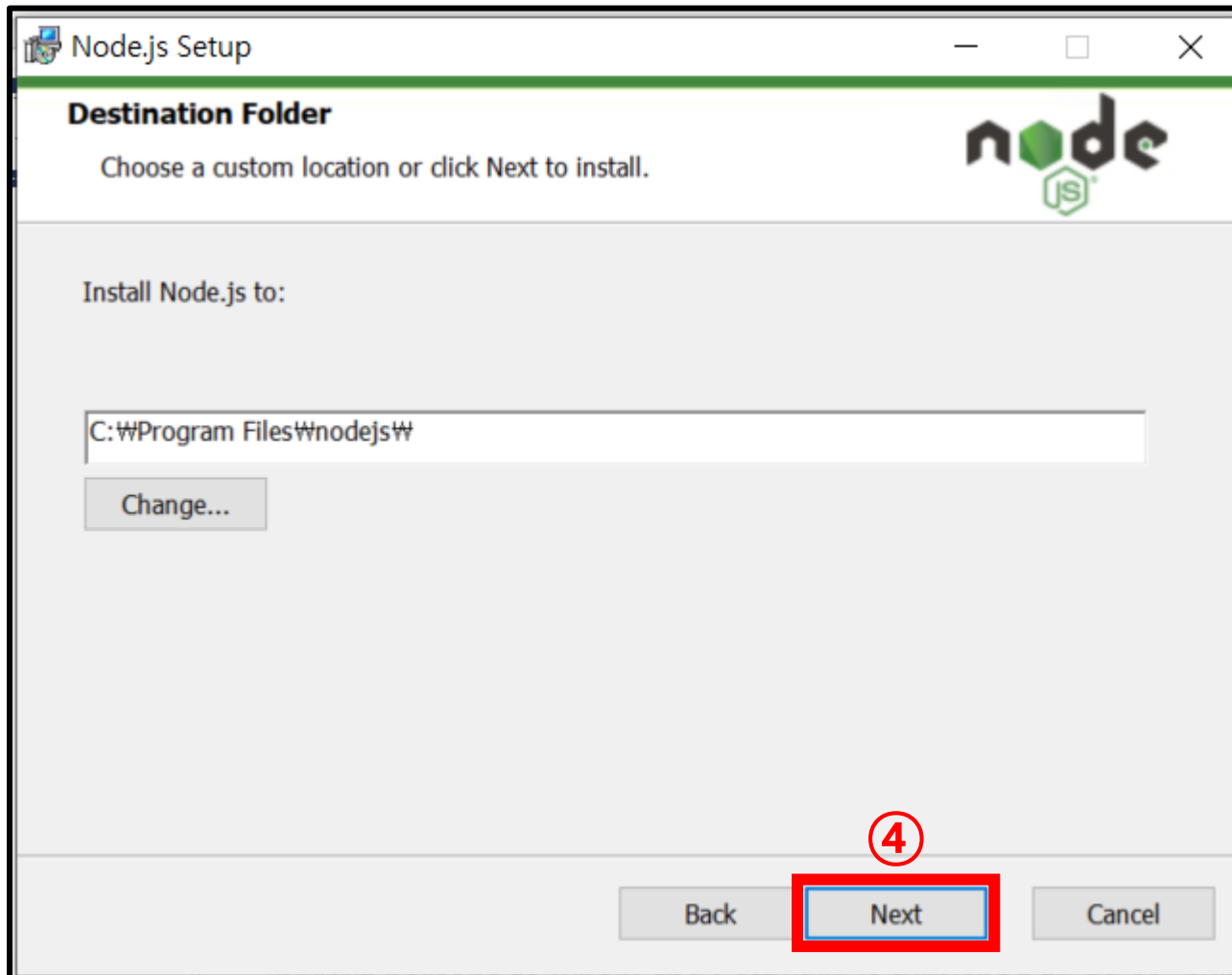
Windows 설치 프로그램 (.msi)

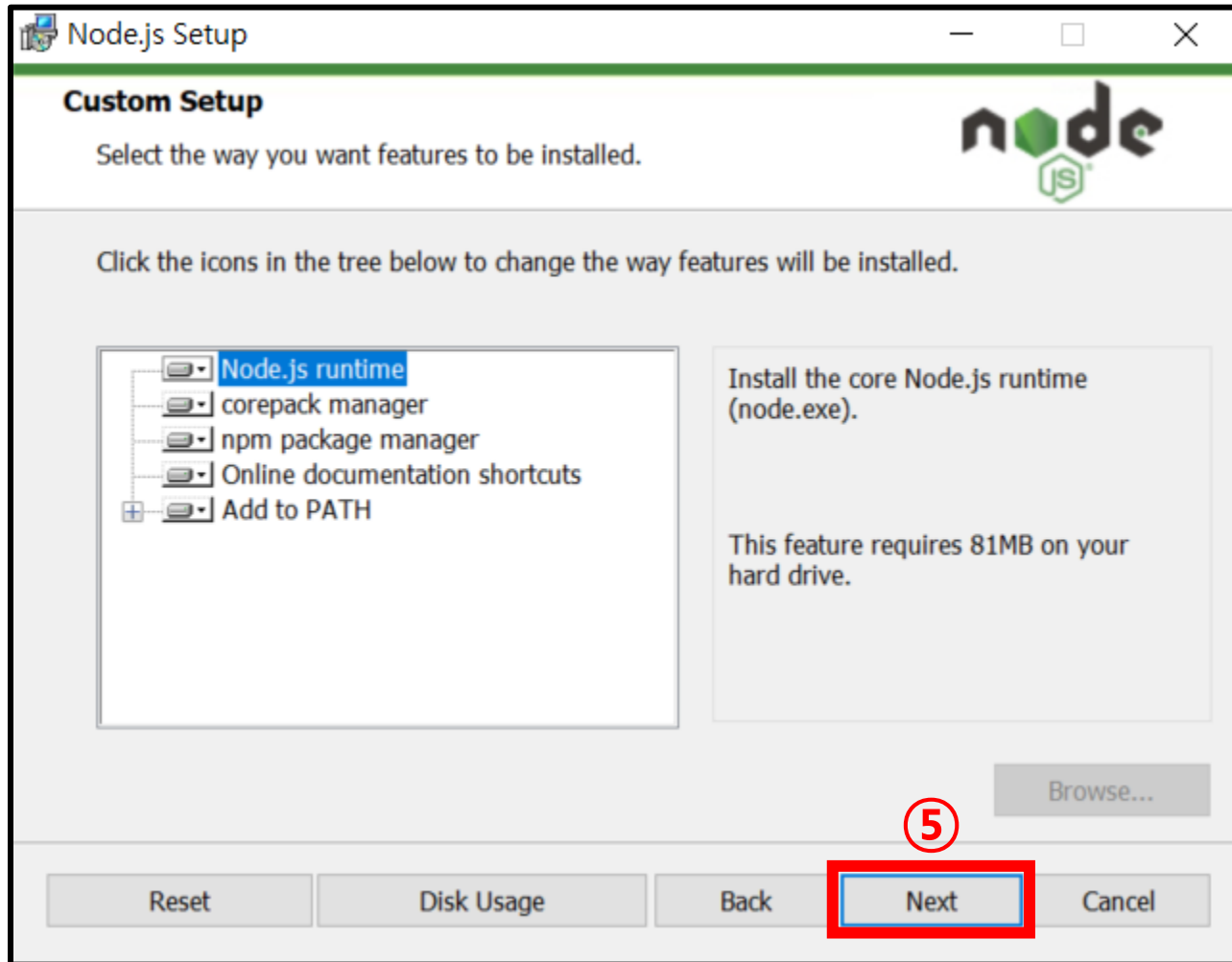
Standalone Binary (.zip)

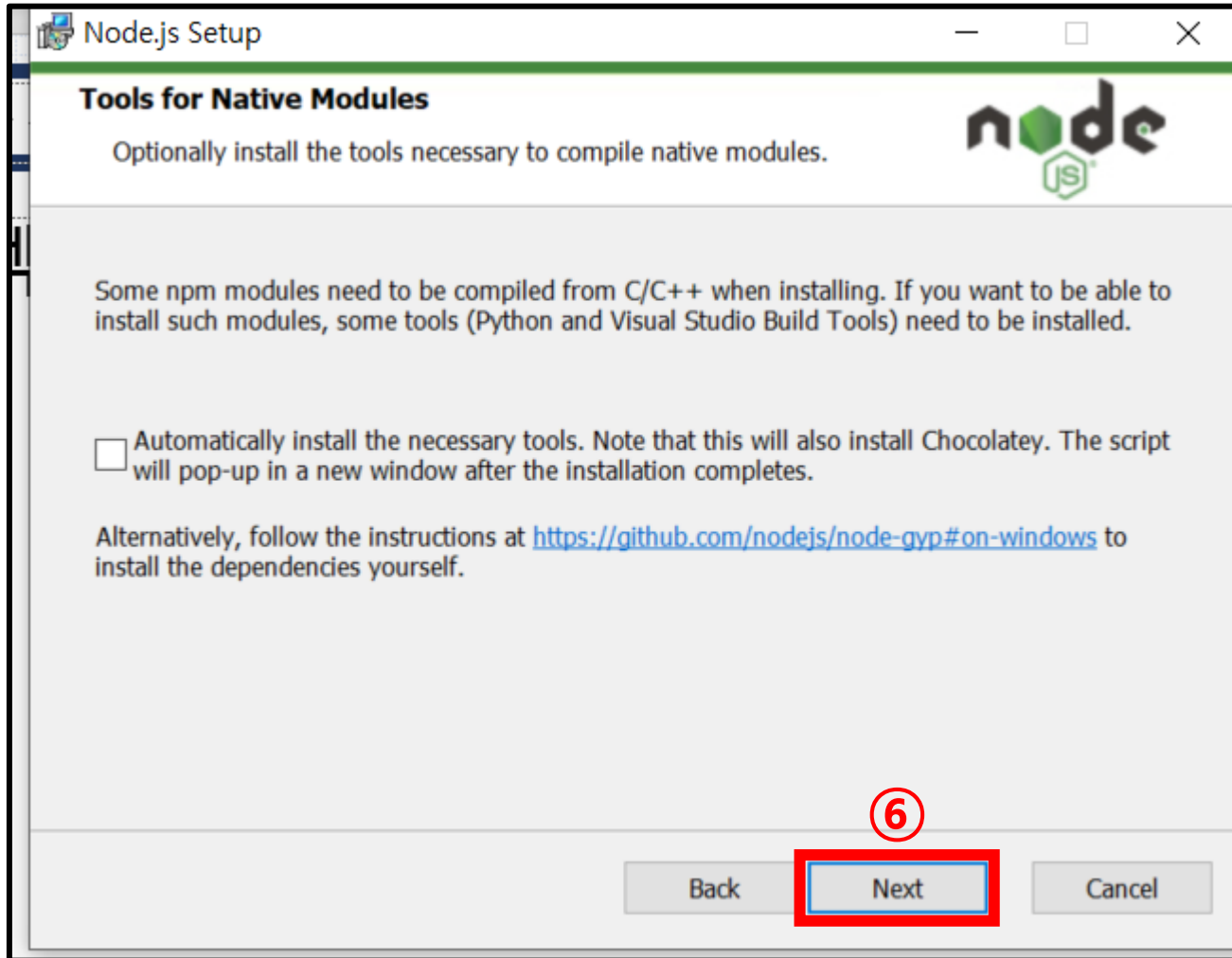


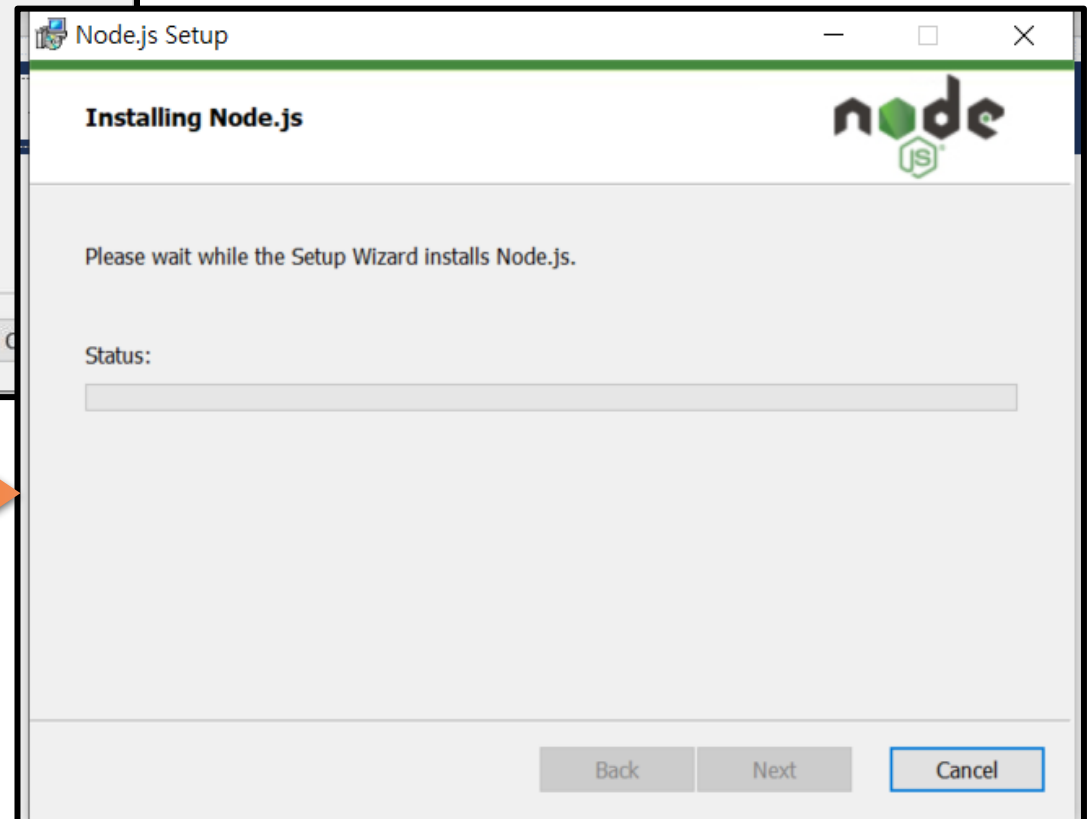
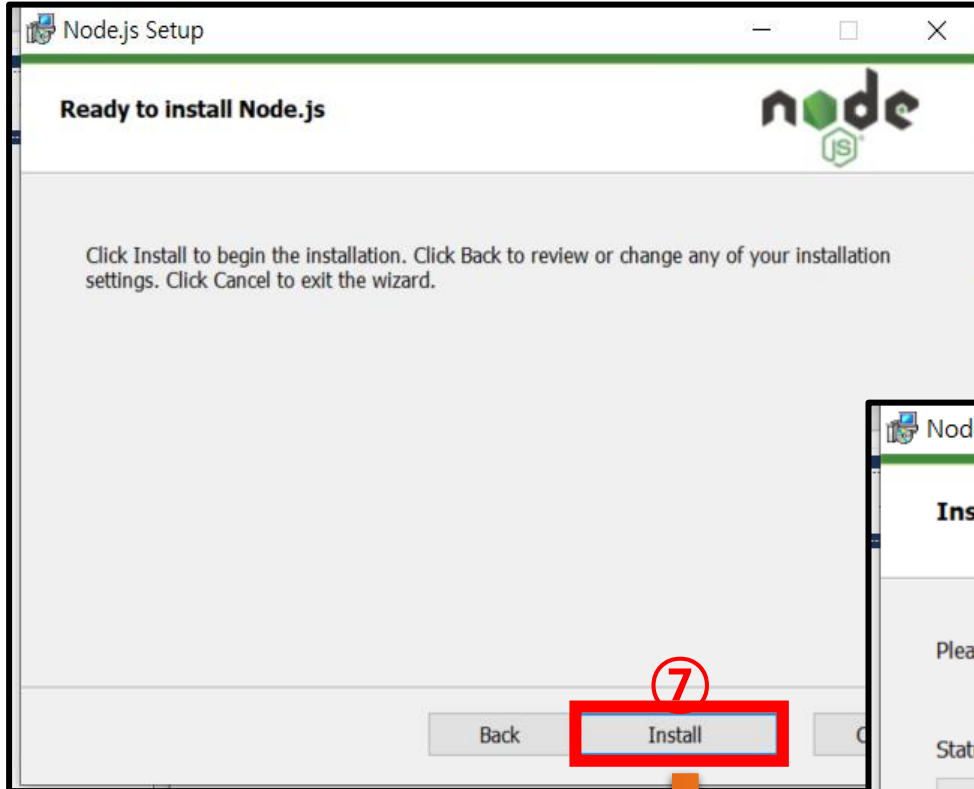


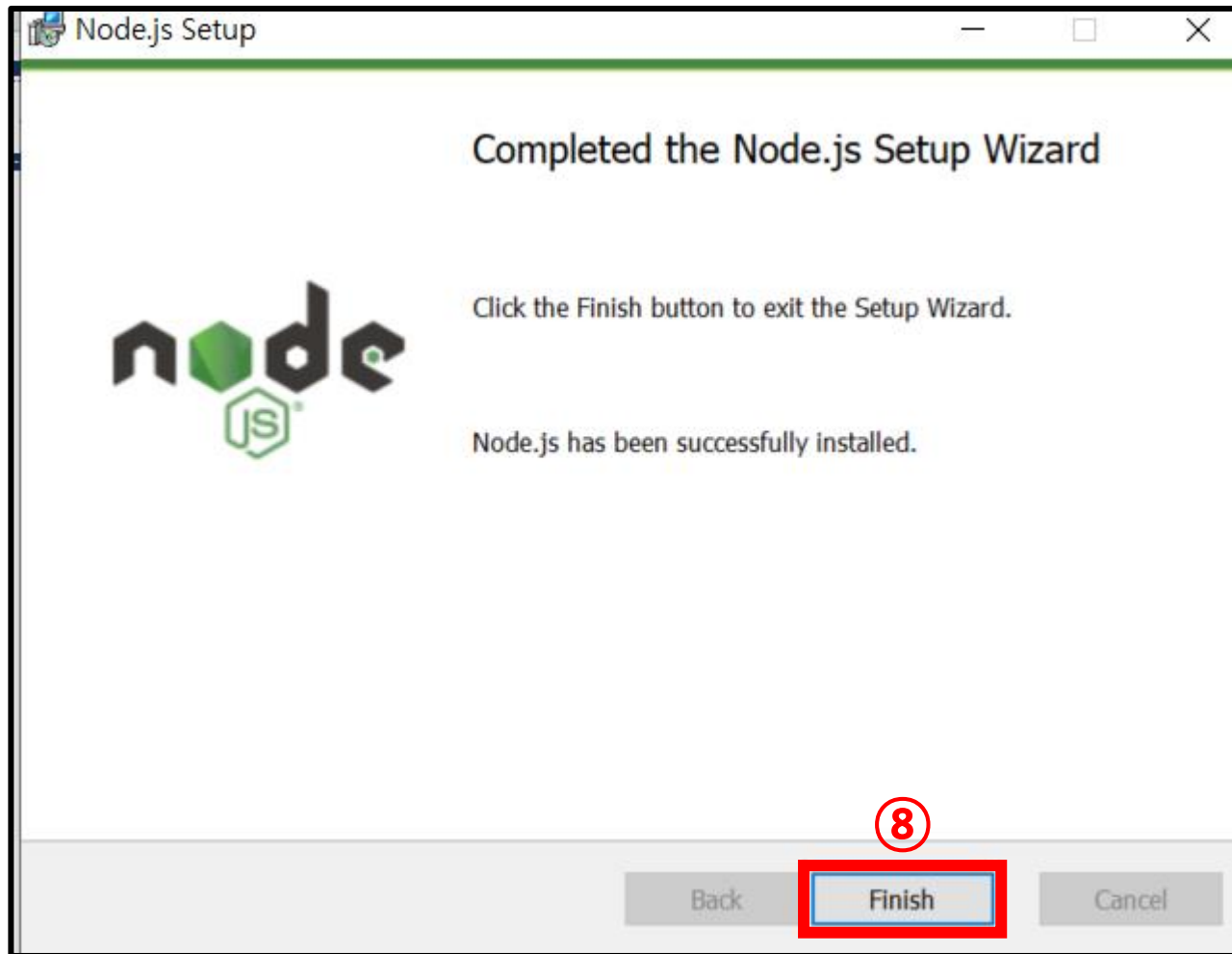














- node.js가 정상적으로 설치되었는지 확인 한다.
- cmd 명령 프롬프트에서 확인

node -v, npm -v

```
관리자: 명령 프롬프트
Microsoft Windows [Version 10.0.19045.6332]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>node -v
v22.20.0

C:\Users\Administrator>npm -v
10.9.3

C:\Users\Administrator>
```

# React – vite 설치하기







- VSC 에디터를 실행한다.
- VSC 에서 생성한 react 폴더를 선택한다.
- 터미널 -> 새터미널 아래 명령어를 입력한다.
- 단, 현재 폴더 . 대신 -> my\_app를 입력한다.

(단, 한글로 된 폴더 이름 사용 하지 않기 => react가 한글로 구동을 잘 못함)

- Node.js 패키지 매니저인 npm을 통해, vite 도구의 최신 템플릿을 사용하여 현재 폴더에 프로젝트를 생성하겠다는 의미

npm이 제공하는 script 명령어

npm create vite@latest .

Node.js 패키지 매니저

Vite 도구의 최신 버전    현재 폴더



- 아래와 같이 대화형 프롬프트가 생성된다.
- 프롬프트란 : 사용자의 입력을 기다리기 위해 표시되는 텍스트, 기호, 질문형태의 메시지를 말한다.  
( ※ 단, powerShell의 실행 정책 때문에 npm 스크립트 실행이 막혀 있어서 Git Bash로만 설치 가능 )

```
Admin@DESKTOP-6ITD093 MINGW64 /d/★2025년 - 그린신촌(KDT)/source/react
$ npm create vite@latest my_app

> npx
> create-vite my_app

Select a framework:
  ● Vanilla
  ○ Vue
  ○ React
  ○ Preact
  ○ Lit
  ○ Svelte
  ○ Solid
  ○ Qwik
  ○ Angular
  ○ Marko
  ○ Others
```

```
◆ Select a framework:
  ○ Vanilla
  ○ Vue
  ● React
  ○ Preact
  ○ Lit
  ○ Svelte
  ○ Solid
  ○ Qwik
  ○ Angular
  ○ Marko
  ○ Others
```



Select a variant:

- TypeScript
- TypeScript + React Compiler
- TypeScript + SWC
- JavaScript
- JavaScript + React Compiler
- JavaScript + SWC
- React Router v7 ↗
- TanStack Router ↗
- RedwoodSDK ↗
- RSC ↗

Use rolldown-vite (Experimental)?:

○ Yes

● No

기존 Vite는 내부적으로 Rollup을 사용한다.

- Rolldown은 Rust로 새로 만든, 훨씬 빠른 실험 버전의 Rollup이다.  
(아직 완전히 안정화되지 않아서 ...)



```
◆ Install with npm and start now?  
● Yes / ○ No
```

npm으로 필요한 패키지를 바로 설치하고, 개발 서버를 지금 실행할까요?”

npm install

npm run dev

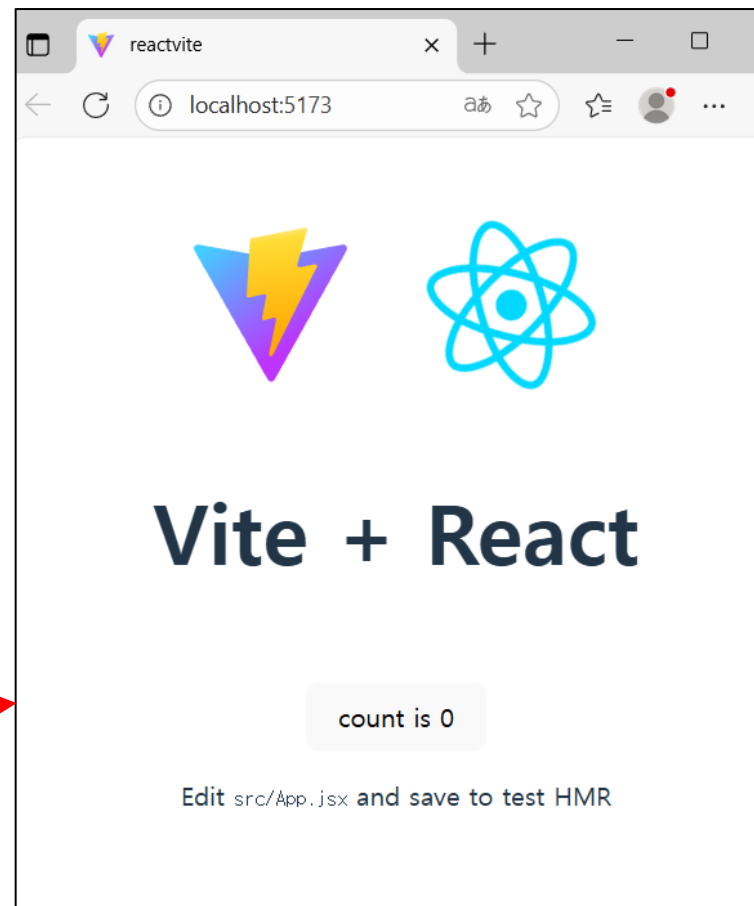
이 두 명령을 자동으로 한 번에 실행해 주겠다는 뜻



```
VITE v7.1.10 ready in 289 ms
```

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

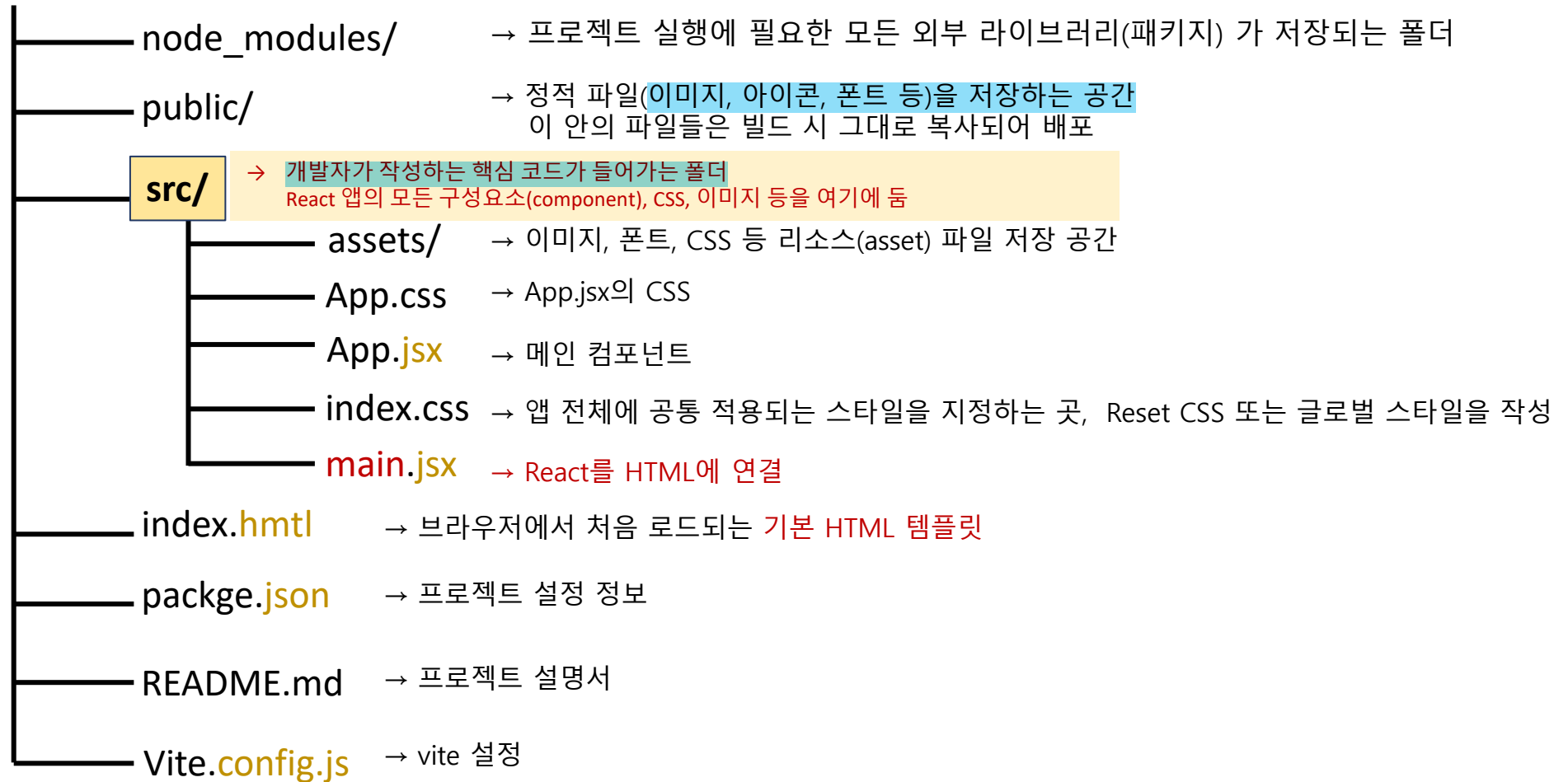
- Ctrl 클릭 하면서 마우스 왼쪽 클릭하면 웹 브라우저에 실행화면이 출력된다.





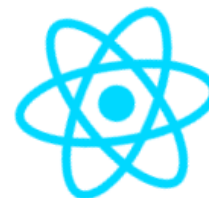
## ➤ React Vite 폴더 구조

Reactvite/



reactvite > src >  App.jsx > ...

```
1  import { useState } from 'react'
2  import reactLogo from './assets/react.svg'
3  import viteLogo from '/vite.svg'
4  import './App.css'
5
6  function App() {
7    const [count, setCount] = useState(0)
8
9    return (
10     <>
11       <div>
12         <a href="https://vite.dev" target="_blank">
13           <img src={viteLogo} className="logo" alt="Vite logo" />
14         </a>
15         <a href="https://react.dev" target="_blank">
16           <img src={reactLogo} className="logo react" alt="React logo" />
17         </a>
18       </div>
19       <h1>Vite + React</h1>
20       <div className="card">
21         <button onClick={() => setCount((count) => count + 1)}>
22           count is {count}
23         </button>
24         <p>
25           Edit <code>src/App.jsx</code> and save to test HMR
26         </p>
27       </div>
28       <p className="read-the-docs">
29         안녕하세요!! 수정
30       </p>
31     </>
32   )
33 }
34
35 export default App
```



# Vite + React

count is 0

Edit src/App.jsx and save to test HMR

안녕하세요!!



## ➤ React Vite 서버 종료 하기

Ctrl + C

## ➤ React Vite 서버 다시 실행 하기

Npm run dev