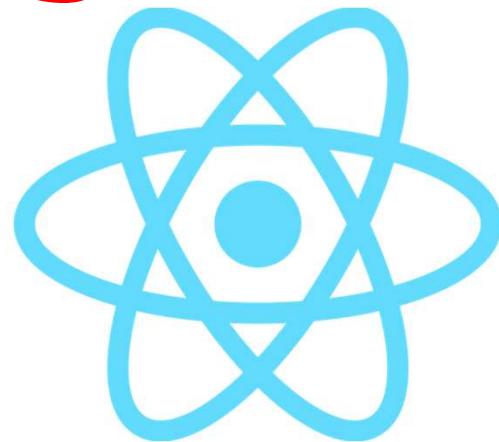


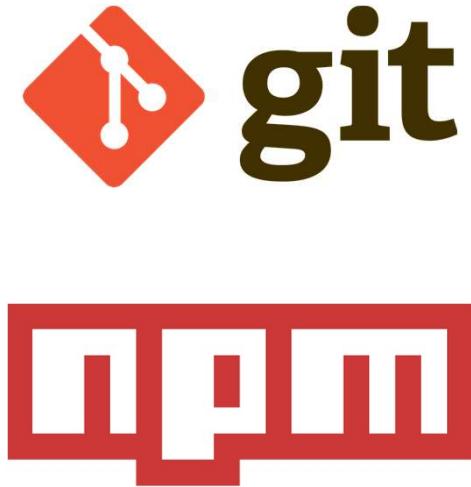
Configure for Lab



Bok, Jong Soon
javaexpert@nate.com
<https://github.com/swacademy/React>

Contents

- Web Browser
 - Google Chrome
 - Mozilla Firefox
- Editor or IDE
 - Microsoft Visual Studio Code
- Node.js
- Git
- npm
- Vite
- create-react-app
- Online Editors



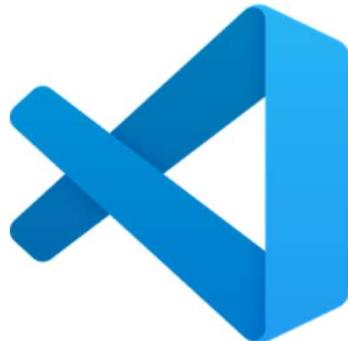
Web Browsers

- Google Chrome Browser
 - <https://www.google.com/chrome/>
- Mozilla Firefox
 - <https://www.mozilla.org/ko/>
- Microsoft Edge
 - <https://www.microsoft.com/ko-kr/edge>



Editors or IDEs

- Visual Studio Code
 - <https://code.visualstudio.com/>
- Atom
 - <https://atom.io/>
- SublimeText3
 - <https://www.sublimetext.com/3>

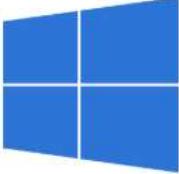


Visual Studio Code

- <https://code.visualstudio.com/Download>

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows
Windows 7, 8, 10

User Installer 64 bit 32 bit
System Installer 64 bit 32 bit
.zip 64 bit 32 bit

↓ .deb
Debian, Ubuntu

.deb 64 bit
.rpm 64 bit
.tar.gz 64 bit

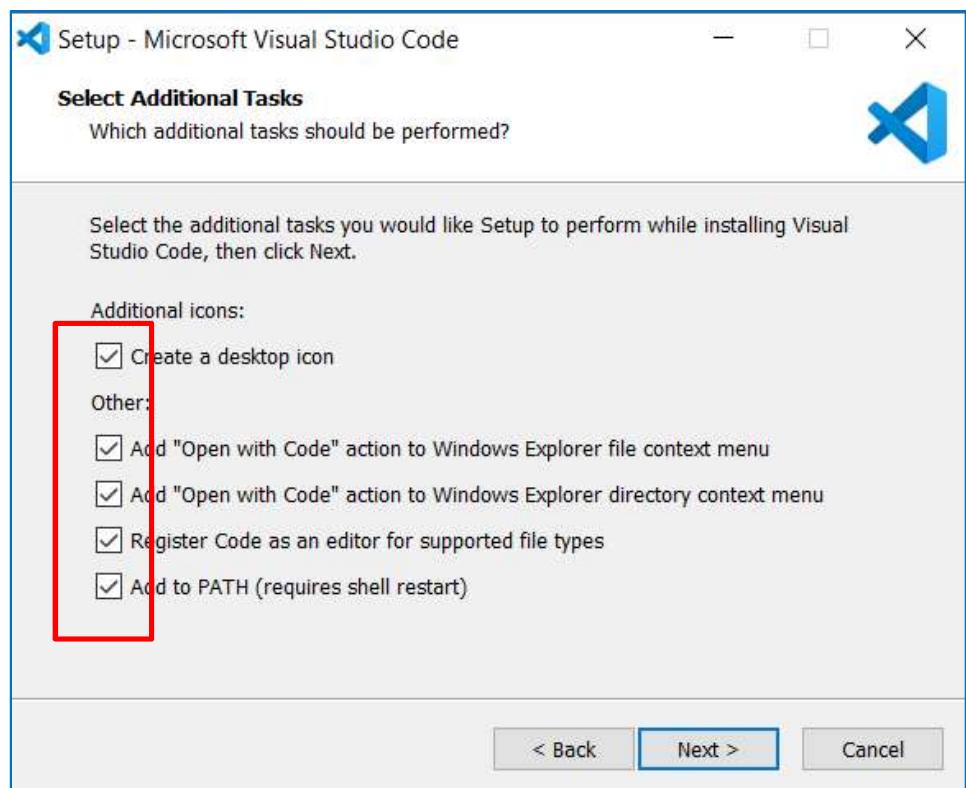
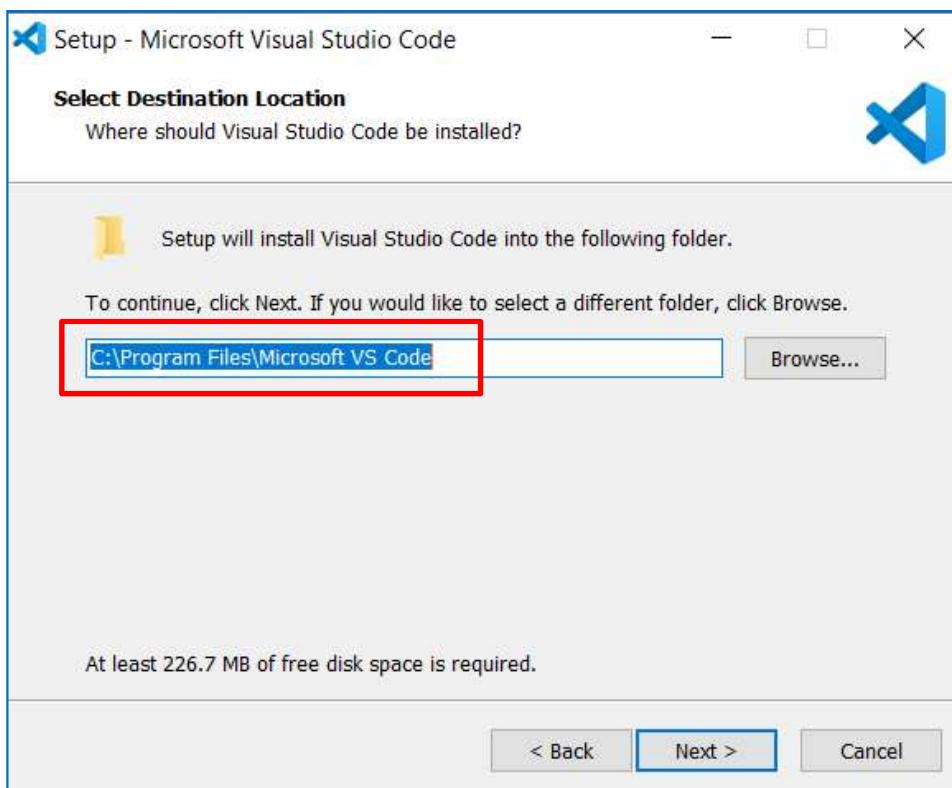
↓ .rpm
Red Hat, Fedora, SUSE

.deb 64 bit
.rpm 64 bit
.tar.gz 64 bit

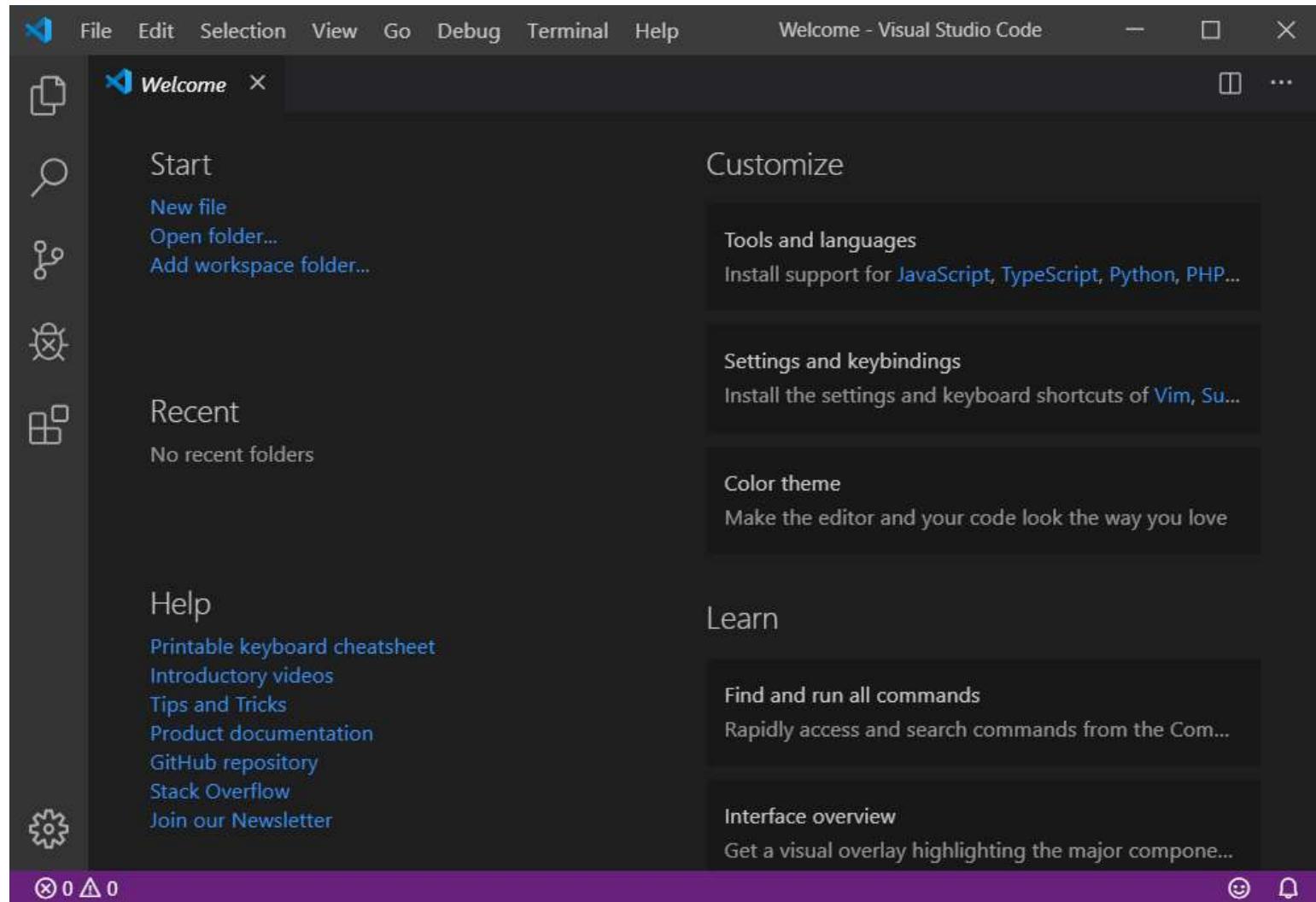
↓ Mac
macOS 10.10+

Snap Store

VSCode (Cont.)



VSCode (Cont.)



VSCode Configuration

- Default Terminal Prompt → Command Prompt(cmd)
- Preferences
 - File : Auto Save → onFocusChange
 - Editor : Tab Size → 2
 - Editor : Default Formatter → TypeScript and JavaScript Language Features
 - Editor : Format On Save → check
 - Explorer : Compact Folders : Uncheck

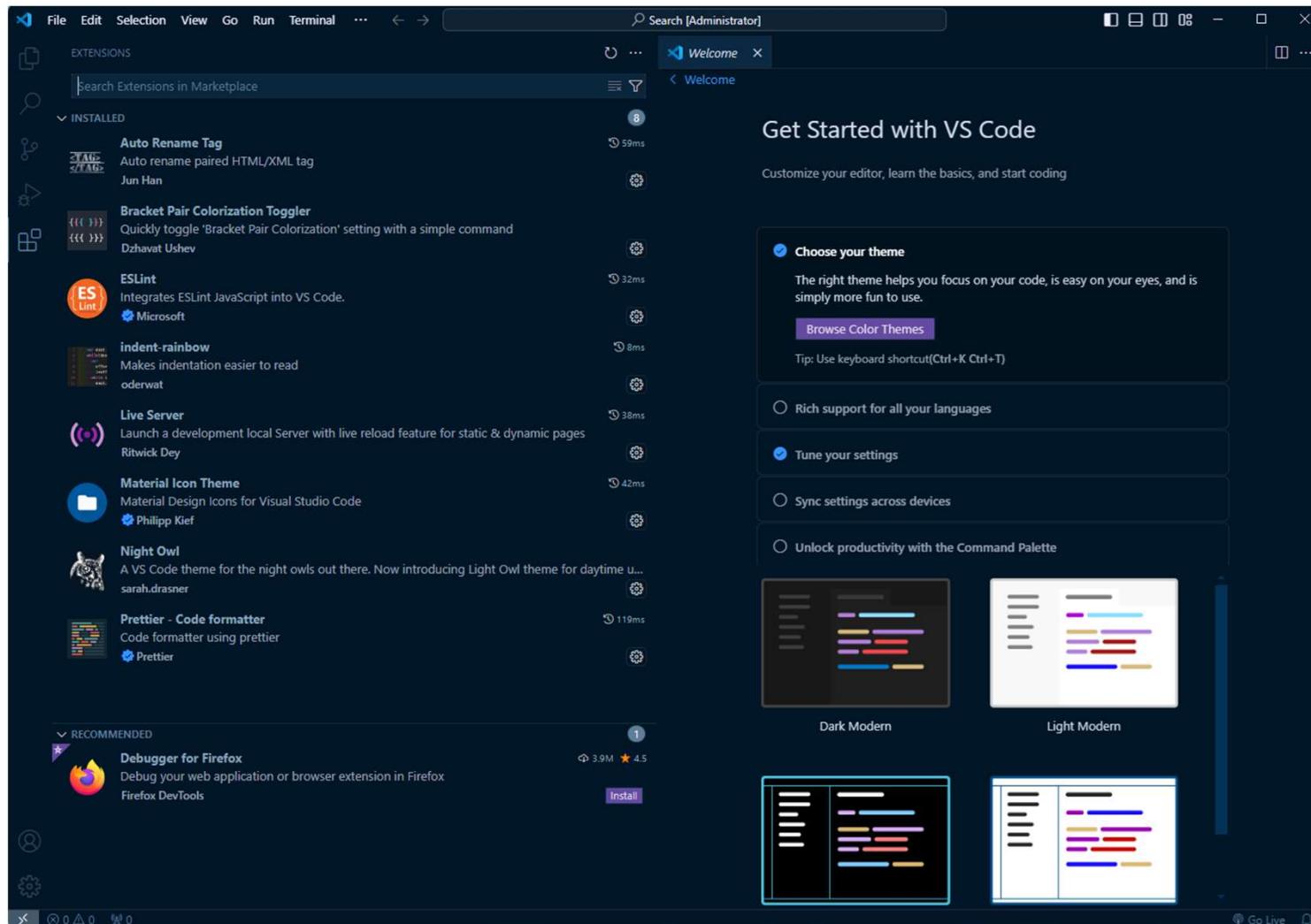
VSCode Extensions

- Night Owl
 - VSCode color theme.
- Material Icon Theme
 - VSCode icon theme
- Live Server
 - Launch a development local Server with live reload feature for static & dynamic pages.(Warning. Oracle 5500 port conflicting)
- ESLint
 - JavaScript code style, 문법 check 가능 제공
- Prettier – Code formatter
 - Code formatter

VSCode Extensions (Cont.)

- Auto Rename Tag
 - Paired HTML/XML tag
- Bracket Pair Colorization Toggler
- Indent-rainbow
 - Makes indentation easier to read
- Error Lens
 - Improve highlighting of errors, warnings and other language diagnostics.
- Auto Close Tag
- Github Theme
- JavaScript (ES6) code snippets
- ES7 React/Redux/GraphQL/React-Native snippets

VSCode Extensions (Cont.)



VSCode Extensions (Cont.)

- [VSCode Theme for 2020](#)
- [8 Awesome Themes for Visual Studio Code](#)
- [6 Hottest VS Code Themes To Use in 2019](#)
- [VSCodethemes](#)

Node.js

■ <https://nodejs.org/en/>

The screenshot shows the official Node.js website. On the left, there's a large heading "Run JavaScript Everywhere" with a subtext about Node.js being a free, open-source runtime environment. Below this is a green button labeled "Download Node.js (LTS)". A red box highlights this button. To the right, there's a code editor window titled "Create an HTTP Server" with some sample Node.js code. The code sets up a simple HTTP server that responds with "Hello World!". At the bottom of the code editor, it says "JavaScript". There's also a "Copy to clipboard" button. The top navigation bar includes links for Learn, About, Download, Blog, Docs, and Certification, along with a search bar and other site icons.

Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

Download Node.js (LTS) ↗

Downloads Node.js **v20.15.1¹** with long-term support.
Node.js can also be installed via package managers.

Want new features sooner? Get **Node.js v22.4.1¹** instead.

```
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`
```

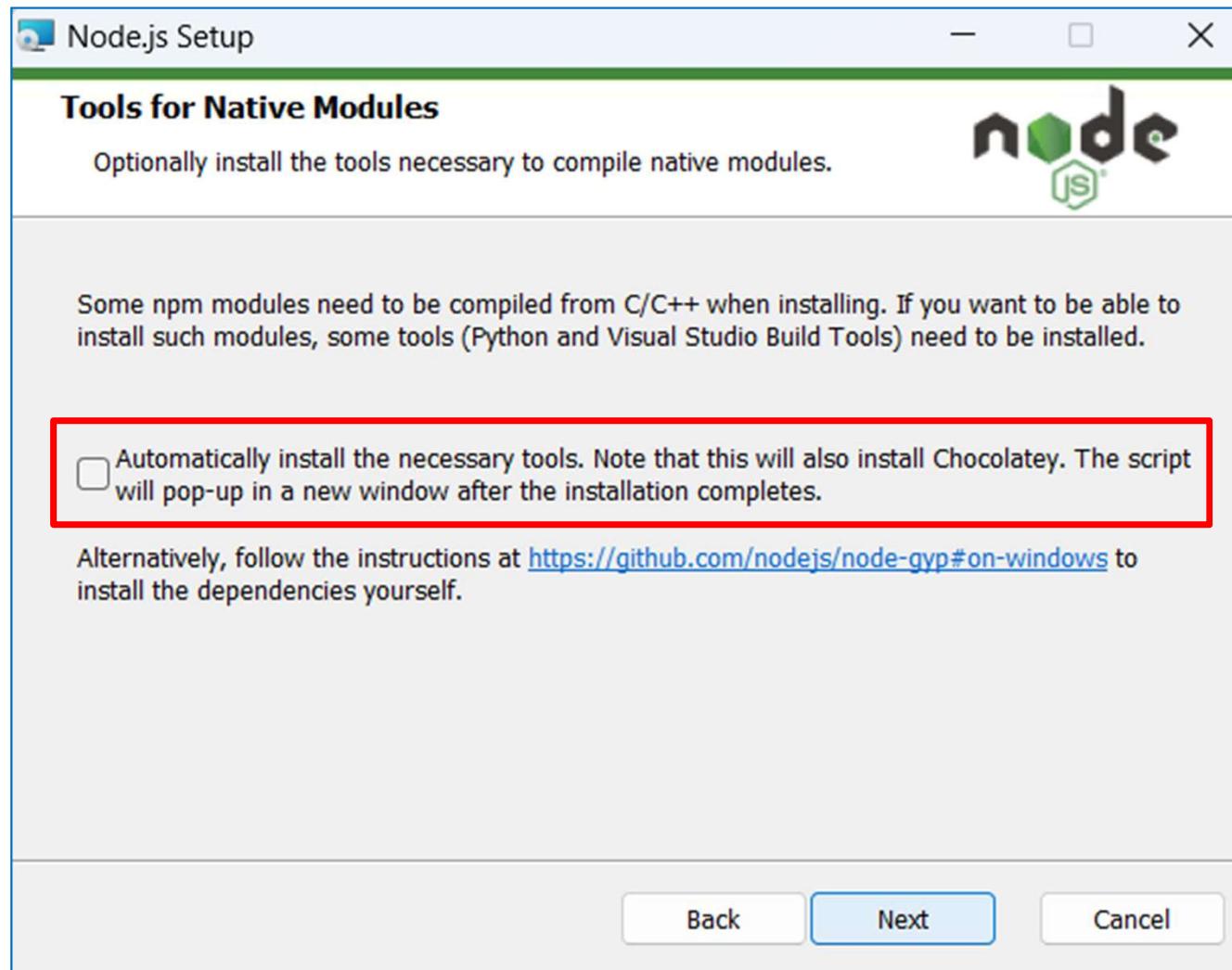
JavaScript

Copy to clipboard

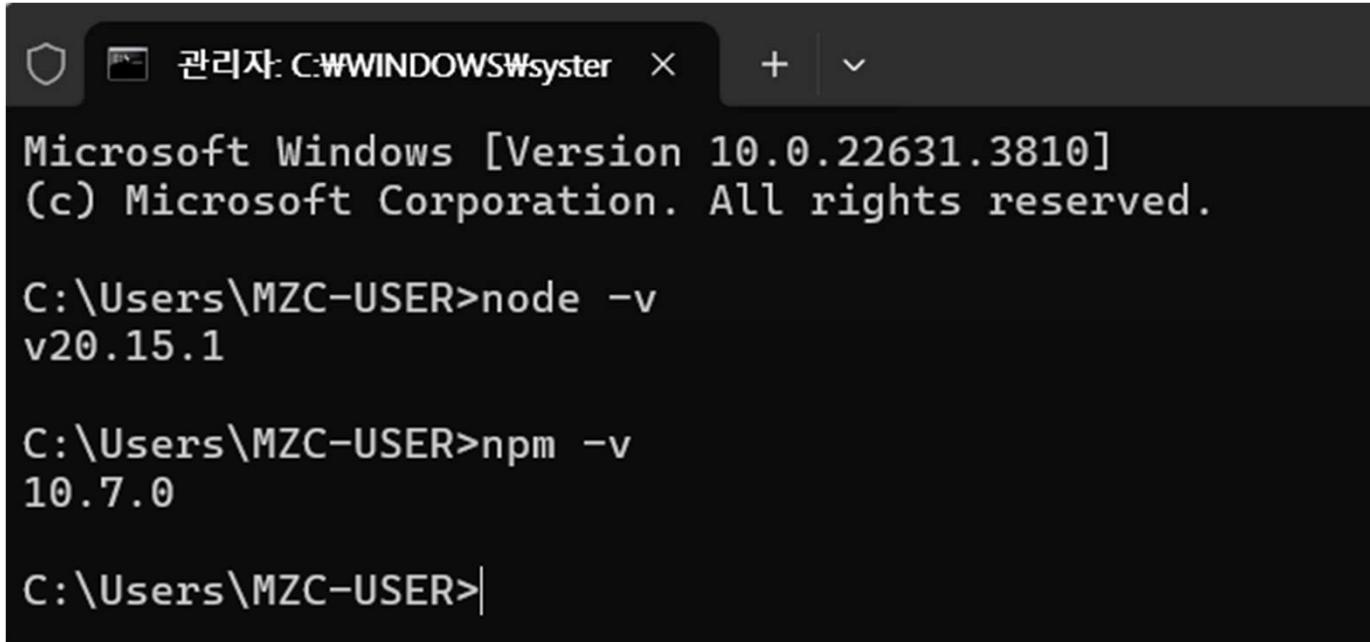
Create an HTTP Server Write Tests Read and Hash a File Streams Pipeline Work with Thre

Start typing... Ctrl K

Node.js (Cont.)



Node.js (Cont.)



관리자: C:\WINDOWS\system X + ▾

```
Microsoft Windows [Version 10.0.22631.3810]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MZC-USER>node -v
v20.15.1

C:\Users\MZC-USER>npm -v
10.7.0

C:\Users\MZC-USER>
```

```
C:\Users\MZC-USER>echo %PATH%
C:\Python312\Scripts\;C:\Python312\;C:\Program Files\Java\jdk-17\bin;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files\Bandizip\;C:\Program Files\Microsoft VS Code\bin;C:\Program Files\PuTTY\;C:\ProgramData\chocolatey\bin;C:\Program Files\nodejs\;C:\Users\MZC-USER\AppData\Local\Microsoft\WindowsApps\;C:\Program Files\JetBrains\IntelliJ IDEA 2024.1.4\bin\;;C:\users\MZC-USER\AppData\Roaming\npm

C:\Users\MZC-USER>
```

Git

■ <https://git-scm.com/>

The screenshot shows the official Git website at <https://git-scm.com/>. The page features a large header with the Git logo and the tagline "git --local-branching-on-the-cheap". A search bar is located in the top right corner. The main content area contains two sections of text describing Git's benefits: "free and open source" and "easy to learn". To the right of the text is a diagram illustrating Git's distributed architecture, showing multiple repositories connected by red lines on a grid background. Below the text are four navigation links: "About", "Documentation", "Downloads", and "Community", each with a corresponding icon. On the right side, there is a graphic of a computer monitor displaying the latest release information: "Latest source Release 2.45.2" with a link to "Release Notes (2024-05-31)" and a "Download for Windows" button.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

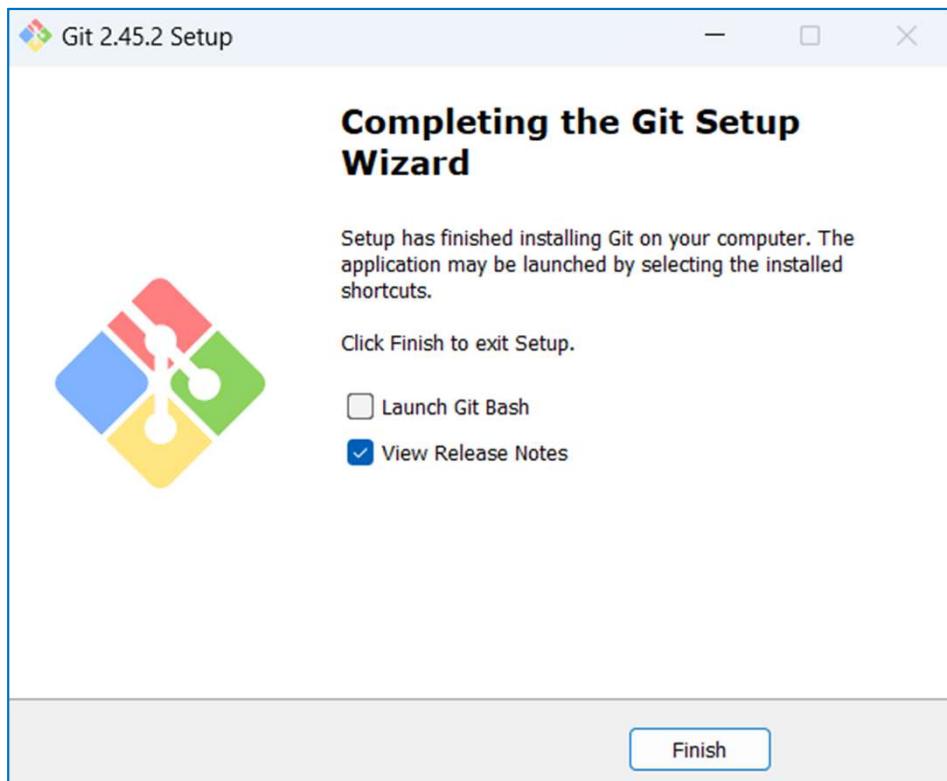
Latest source Release
2.45.2
[Release Notes \(2024-05-31\)](#)

[Download for Windows](#)

Git (Cont.)

```
C:\Users\MZC-USER>git --version  
'git'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는  
배치 파일이 아닙니다.
```

```
C:\Users\MZC-USER>
```



```
관리자: C:\WINDOWS\system >  
Microsoft Windows [Version 10.0.22631.3810]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\MZC-USER>git --version  
git version 2.45.2.windows.1  
  
C:\Users\MZC-USER>
```

npm

- Node Package Manager
- JavaScript Programming을 위한 패키지 관리자
- JavaScript Runtime Environment Node.js의 기본 패키지 관리자.
- 검색
 - \$ **npm search moduleName**
 - npmjs.com에서 검색하는 것이 보기 좋다.
- 상세 정보
 - \$ **npm info (view) moduleName**
- version 정보
 - \$ **npm info moduleName version**
 - \$ **npm info moduleName versions** #모든버전

npm (Cont.)

■ 설치 - 로컬 설치

- 설치된 폴더와 하위 폴더에서만 사용 가능
- 설치한 모듈은 설치 경로 내부에 *node_modules*라는 폴더에 설치되며 해당 경로 하위에서만 호출할 수 있다.
- \$ **npm install moduleName**
- \$ **npm i moduleName**
- \$ **npm moduleName --save** //구 버전
- 버전별 설치 방식
 - npm i bootstrap@4 => 4의 마지막 버전
 - npm i bootstrap@4.5 => 4.5의 마지막 버전
 - npm i bootstrap@5.2.0-beta1 => 매칭되는 버전 설치

npm (Cont.)

■ 설치 - 프로젝트 진행과 상관 없는 모듈 설치

- \$ `npm install moduleName --save-dev` // 구 버전
- \$ `npm i -D moduleName` // 프로젝트 작성과는 상관없는 모듈을 설치하는 경우
// (프로젝트 작성과는 상관 없는 모듈)

- i.e. text에 관련된 모듈.

- \$ `npm i -D eslint`

```
{  
  "dependencies": {  
    "bootstrap": "^4.6.2",  
    "jquery": "^3.6.4",  
    "live-server": "^1.2.2"  
  },  
  "devDependencies": {  
    "eslint": "^8.36.0"  
  }  
}
```

npm (Cont.)

■ 설치 - Global 설치

- 컴퓨터 어느 위치에서도 사용 가능한 형태가 된다.
- \$ **npm install -g moduleName**
- \$ **npm i -g moduleName**
- 설치한 모듈을 어느 경로에서든지 호출할 수 있다.
- 로컬 머신에 흔적이 남지 않는다.
- \$ **npm list -g moduleName** 으로 검색해야 함.
- C:\Users\{username}\AppData\Roaming\npm에서 확인할 것

npm (Cont.)

■ 삭제

- \$ npm un moduleName
- \$ npm uninstall moduleName
- \$ npm un -g moduleName
- \$ npm uninstall -g moduleName

■ 설치 모듈 정보

- \$ npm list react-dom
- \$ npm list jquery

npm (Cont.)

- 작업하기 전에 설치할 모듈을 관리하는 파일부터 작성한다.

- \$ **npm init**
- \$ **npm init -y**

- 아래 기본값으로 자동 작성

package name: (jsx-demo)	//프로젝트 명 (대문자는 사용 불가)
version: (1.0.0)	//프로젝트 버전
description:	// 프로젝트 설명
entry point: (index.js)	//어느 파일 기준으로 실행 할 것인가
test command:	//Local에 설치된 모듈을 실행할 명령을 기술
git repository:	//git address
keywords:	//npm search로 검색될 단어
author:	//작성자
license: (ISC)	//저작권 관련

npm (Cont.)

- 필요한 모듈을 **i** 또는 **i -D** 형태로 설치
- 설치 후 node_modules가 잘못된 경우, moduleName와 package-lock.json 삭제
- 재 설치
 - package.json의 dependencies와 devDependencies를 재 설치해 준다.
 - \$ **npm i** or
 - \$ **npm install** //node_modules 폴더가 삭제가 되어 다시 설치할 때
 - \$ **npm i --production** //dependencies 항목만 재 설치한다.

npm (Cont.)

■ 실행 명령 등록

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "build": "npx babel src -d dist"  
},
```

■ 명령의 실행은

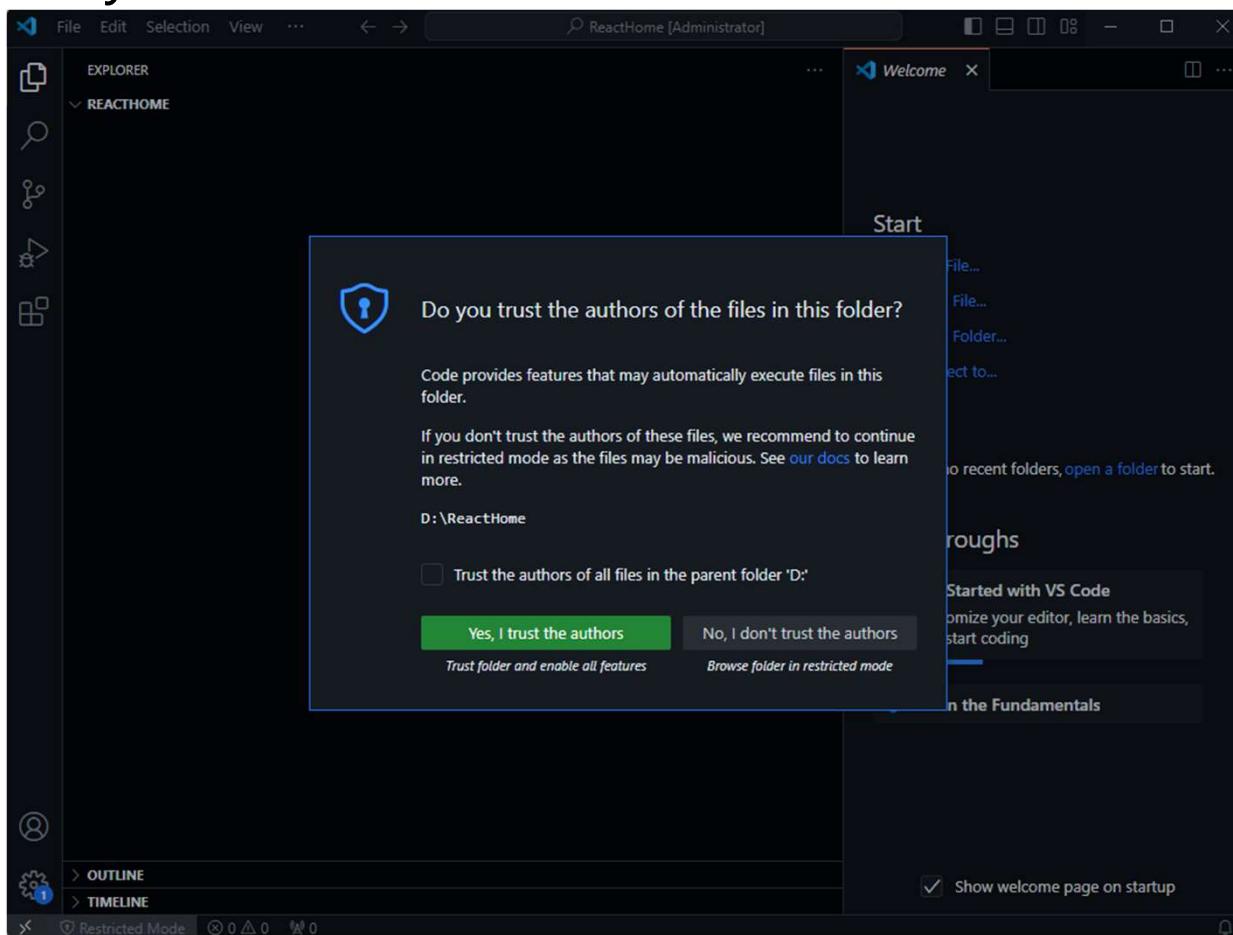
- \$ **npm run test**
- \$ **npm run build**

■ start, test 명령은 run을 생략하고 다음 명령으로 실행 가능

- \$ **npm start**

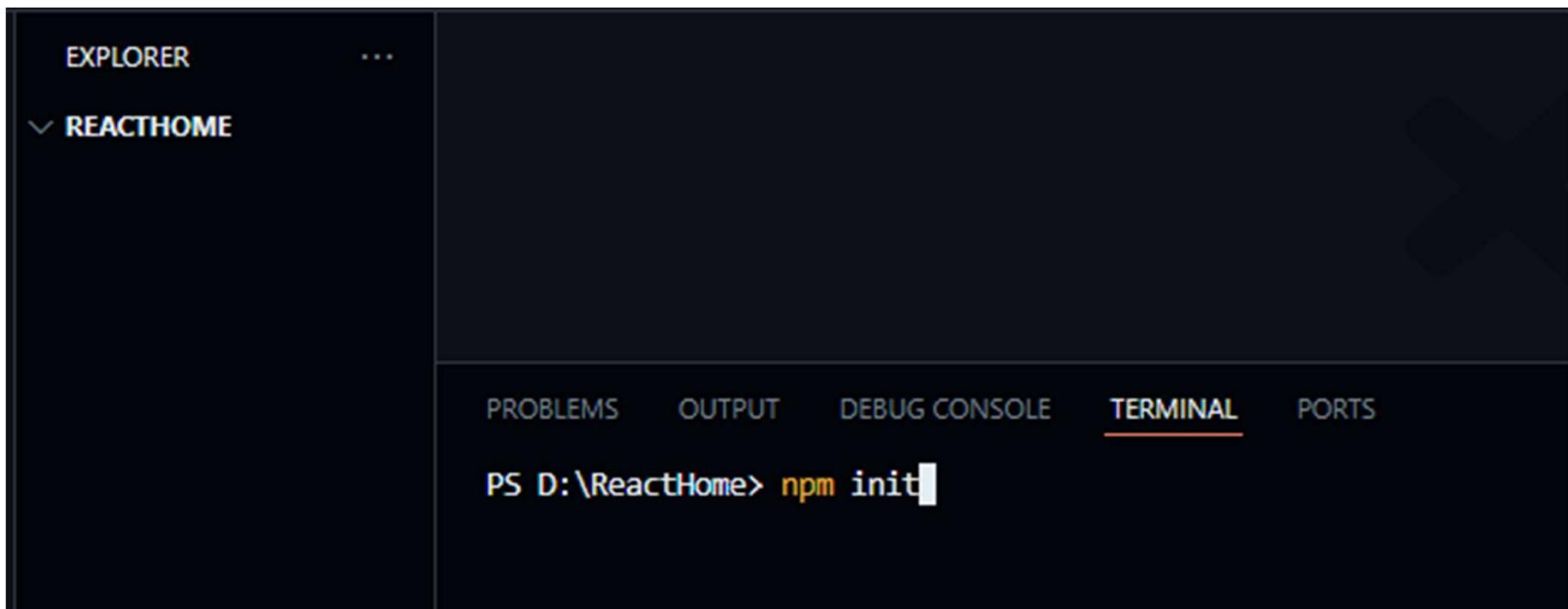
npm (Cont.)

■ Working Directory → ReactHome



npm (Cont.)

■ npm init



The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with a single item named 'REACTHOME'. The main area is dark-themed. At the bottom, there's a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined in orange), and PORTS. The terminal window contains the command 'PS D:\ReactHome> npm init'.

npm (Cont.)

■ npm init

```
PS D:\ReactHome> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (reacthome) Demo
Sorry, name can no longer contain capital letters.
package name: (reacthome) demo
version: (1.0.0)
description: node package test
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
```

```
About to write to D:\ReactHome\package.json:

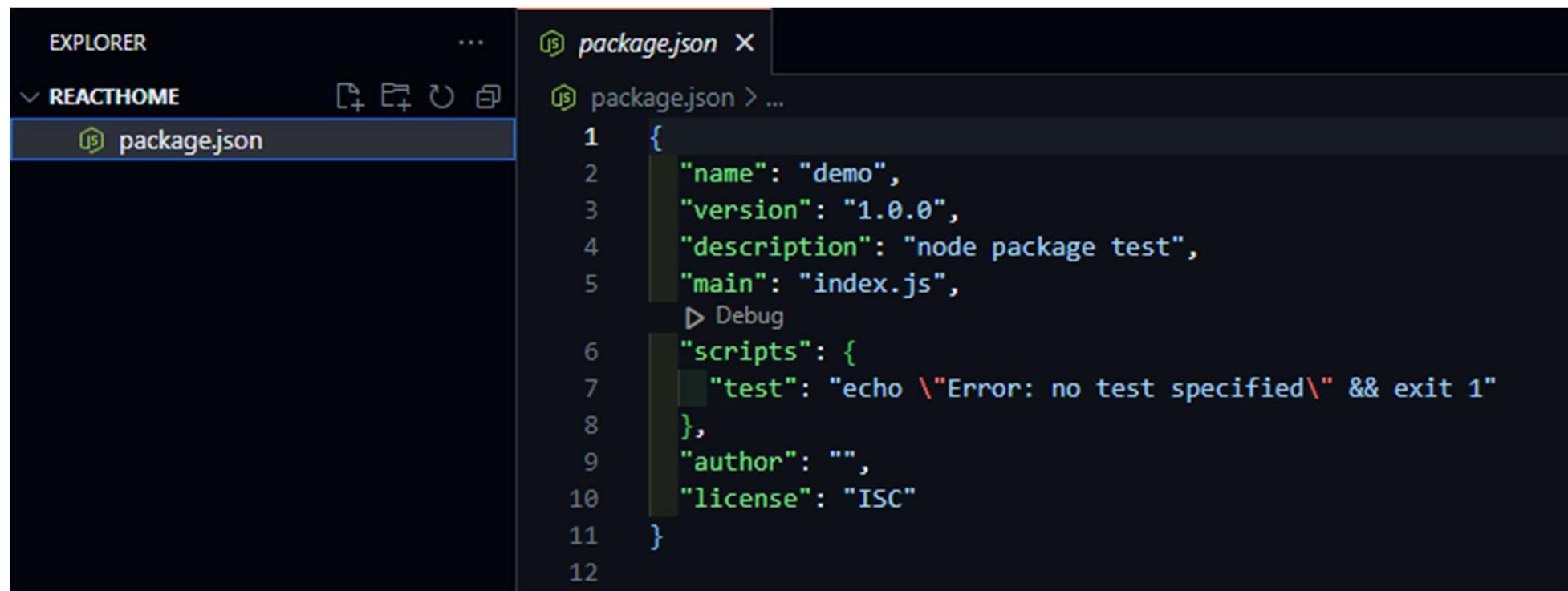
{
  "name": "demo",
  "version": "1.0.0",
  "description": "node package test",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes) yes

npm notice
npm notice New minor version of npm available! 10.7.0 -> 10.8.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.2
npm notice To update run: npm install -g npm@10.8.2
npm notice
PS D:\ReactHome> []
```

npm (Cont.)

■ npm init



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a project structure with a single item: 'REACTHOME' which contains a 'package.json' file. The 'package.json' file is selected and highlighted with a blue border. On the right, the main editor pane shows the contents of the 'package.json' file:

```
1  {
2    "name": "demo",
3    "version": "1.0.0",
4    "description": "node package test",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \\\"Error: no test specified\\\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC"
11 }
12
```

npm (Cont.)

■ index.js 작성

The screenshot shows the Visual Studio Code interface with a dark theme. On the left, the Explorer sidebar displays a folder named 'REACTHOME' containing 'index.js' and 'package.json'. In the center, the code editor has two tabs: 'package.json' (selected) and 'index.js'. The 'index.js' tab contains the following code:

```
1 console.log('Hello, World');
```

Below the code editor is a terminal window showing the output of running the script:

```
PS D:\ReactHome> node index.js
Hello, World
PS D:\ReactHome>
```

npm (Cont.)

■ package.json 설정

```
package.json X index.js
package.json > ...
1  {
2    "name": "demo",
3    "version": "1.0.0",
4    "description": "node package test",
5    "main": "index.js",
6    "scripts": {
7      "start": "node index.js",
8      "test": "echo \\$error: no test specified\\$ && exit 1"
9    },
10   "author": "",
11   "license": "ISC"
12 }
13
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\ReactHome> npm run start
> demo@1.0.0 start
> node index.js

Hello, World
PS D:\ReactHome>
```

npm – 외부 Library 사용하기

■ lodash from npmjs.com

The screenshot shows the npmjs.com package page for the lodash library. At the top, there's a navigation bar with links for 'Pro', 'Teams', 'Pricing', and 'Documentation'. Below that is the npm logo and a search bar with the placeholder 'Search packages'. A 'Sign Up' button is also visible. The main content area features the package name 'lodash' in bold, followed by its version 'v4.17.21'. It indicates that it's a 'Public' package published 3 years ago. There are tabs for 'Readme' (which is selected), 'Code' (Beta), '0 Dependencies', '201,337 Dependents', and '114 Versions'. The 'Readme' tab contains the text: 'The Lodash library exported as Node.js modules.' Below this is an 'Installation' section with instructions for using npm:

```
$ npm i -g npm  
$ npm i --save lodash
```

On the right side, there's an 'Install' section with the command 'npm i lodash' and a copy icon. Below it are links for 'Repository' (github.com/lodash/lodash) and 'Homepage' (lodash.com/). There's also a link for 'Weekly Downloads'.

npm – 외부 Library 사용하기 (Cont.)

■ lodash 설치

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a project structure under 'REACTHOME' with files: node_modules, index.js, package-lock.json, and package.json. The package.json file is currently selected. The main editor area shows the contents of package.json:

```
1  {
2    "name": "demo",
3    "version": "1.0.0",
4    "description": "node package test",
5    "main": "index.js",
6    "scripts": {
7      "start": "node index.js",
8      "test": "echo \\\"Error: no test specified\\\" && exit 1"
9    },
10   "author": "",
11   "license": "ISC",
12   "dependencies": {
13     "lodash": "^4.17.21"
14   }
15 }
```

At the bottom, a terminal window shows the command 'npm i lodash' being run, with the output indicating 1 package added and 0 vulnerabilities found.

```
PS D:\ReactHome> npm i lodash
added 1 package, and audited 2 packages in 3s
found 0 vulnerabilities
PS D:\ReactHome>
```

npm – 외부 Library 사용하기 (Cont.)

■ lodash 이용하기

The screenshot shows a terminal window with two panes. The left pane contains the code for `index.js`:

```
JS index.js  X
JS index.js > ...
1 import lodash from 'lodash';
2
3 const array = [1,1,1,2,2,1,1,4,4,3,2];
4 const uniqueArray = lodash.uniqBy(array);
5
6 console.log(uniqueArray);
```

The right pane shows the command being run and the resulting error:

```
PS D:\ReactHome> npm run start
> demo@1.0.0 start
> node index.js

(node:716) Warning: To load an ES module, set "type": "module" in the package.json or use the .mjs extension.
(Use `node --trace-warnings ...` to show where the warning was created)
D:\ReactHome\index.js:1
import lodash from 'lodash';
^^^^^

SyntaxError: Cannot use import statement outside a module
    at wrapSafe (node:internal/modules/cjs/loader:1281:20)
    at Module._compile (node:internal/modules/cjs/loader:1321:27)
    at Module._extensions..js (node:internal/modules/cjs/loader:1416:10)
    at Module.load (node:internal/modules/cjs/loader:1208:32)
    at Module._load (node:internal/modules/cjs/loader:1024:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:174:12)
    at node:internal/main/run_main_module:28:49

Node.js v20.15.1
PS D:\ReactHome> []
```

A red box highlights the error message: `SyntaxError: Cannot use import statement outside a module`.

npm – 외부 Library 사용하기 (Cont.)

■ lodash 이용하기

The screenshot shows a code editor interface with two tabs: 'index.js' and 'package.json'. The 'package.json' tab is active, displaying the following JSON content:

```
  "scripts": {  
    "start": "node index.js",  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "author": "",  
  "license": "ISC",  
  "type": "module",  
  "dependencies": {  
    "lodash": "^4.17.21"  
  }  
}
```

The line `"type": "module"` is highlighted with a red rectangular box.

npm – 외부 Library 사용하기 (Cont.)

■ lodash 이용하기

```
JS index.js X
JS index.js > ...
1 import lodash from 'lodash';
2
3 const array = [1,1,1,2,2,1,1,4,4,3,2];
4 const uniqueArray = lodash.uniqBy(array);
5
6 console.log(uniqueArray);
```

PS D:\ReactHome> npm run start

```
> demo@1.0.0 start
> node index.js
```

[1, 2, 4, 3]

PS D:\ReactHome> []

Vite

- Is a modern build tool that significantly improves the frontend development experience.
- Is particularly popular for its speed and simplicity, offering a better alternative to traditional bundlers like Webpack.
- Was created by Evan You, the creator of Vue.js, but it works with any frontend framework, including React, Svelte, and others.
- <https://vitejs.dev/>



Vite (Cont.)

■ Installation

```
$ npm create vite@latest
```

```
C:\react-home>npm create vite@latest
Need to install the following packages:
```

```
create-vite@5.4.0
```

```
Ok to proceed? (y) y
```

```
> npx
> create-vite
```

```
? Project name: » vite-project
```

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

Vite (Cont.)

■ Installation

```
$ npm create vite@latest
```

```
✓ Project name: ... vite-project
✓ Select a framework: » React
? Select a variant: » - Use arrow-keys. I
  TypeScript
  TypeScript + SWC
> JavaScript
  JavaScript + SWC
  Remix ↗
```

```
✓ Project name: ... vite-project
✓ Select a framework: » React
✓ Select a variant: » JavaScript
```

```
Scaffolding project in C:\react-home\vite-project...
```

```
Done. Now run:
```

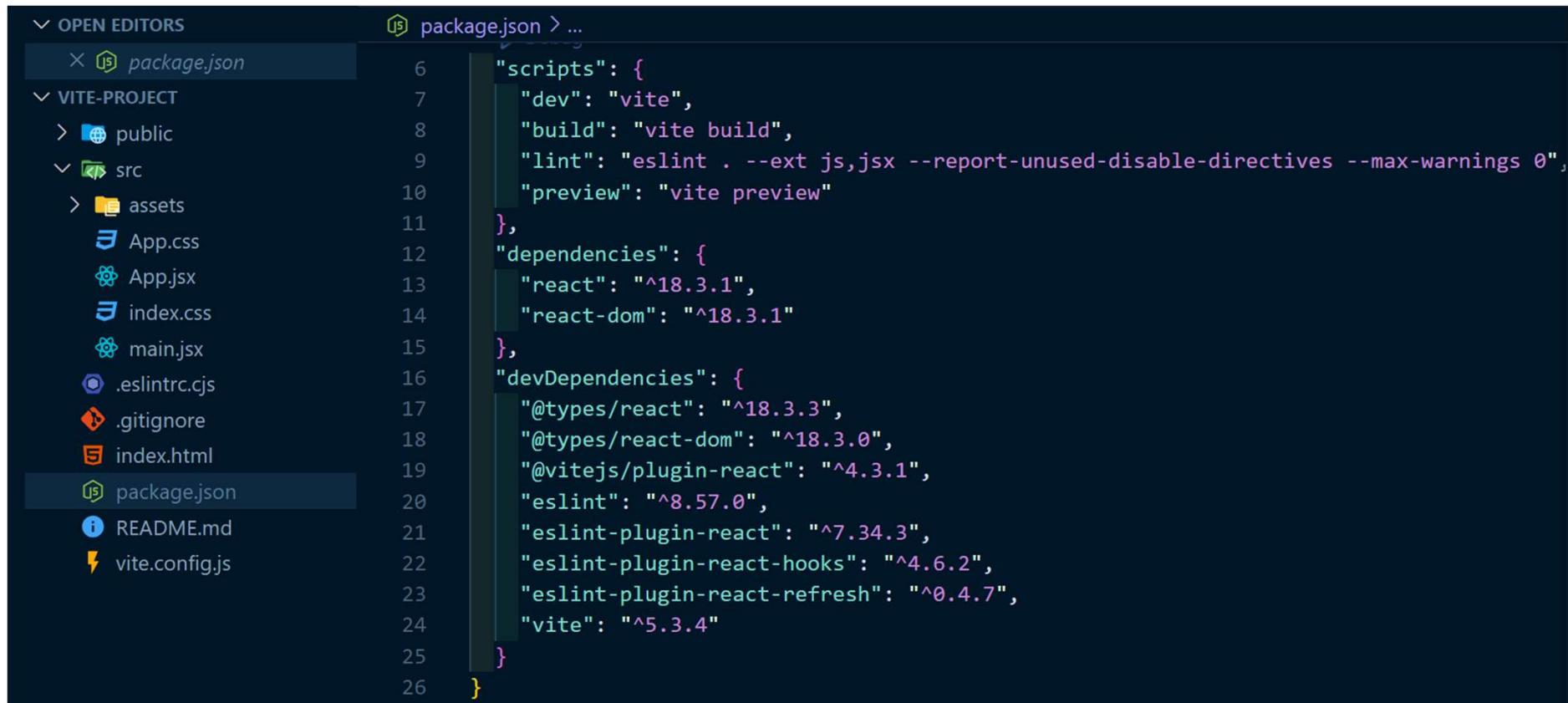
```
cd vite-project
npm install
npm run dev
```

```
C:\react-home>[]
```

Vite (Cont.)

■ Installation

```
$ npm create vite@latest
```



The screenshot shows a code editor interface with the following details:

- OPEN EDITORS:** package.json
- VITE-PROJECT:** A tree view showing the project structure:
 - public
 - src
 - assets
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package.json
 - README.md
 - vite.config.js
- package.json > ...** (highlighted)
- Code Content:** The package.json file content is displayed, showing scripts, dependencies, and devDependencies.

```
6   "scripts": {
7     "dev": "vite",
8     "build": "vite build",
9     "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings 0",
10    "preview": "vite preview"
11  },
12  "dependencies": {
13    "react": "^18.3.1",
14    "react-dom": "^18.3.1"
15  },
16  "devDependencies": {
17    "@types/react": "^18.3.3",
18    "@types/react-dom": "^18.3.0",
19    "@vitejs/plugin-react": "^4.3.1",
20    "eslint": "^8.57.0",
21    "eslint-plugin-react": "^7.34.3",
22    "eslint-plugin-react-hooks": "^4.6.2",
23    "eslint-plugin-react-refresh": "^0.4.7",
24    "vite": "^5.3.4"
25  }
26 }
```

Vite (Cont.)

■ Installation

\$ **npm i**

The screenshot shows the VS Code interface with the 'TERMINAL' tab selected. The terminal window displays the output of an 'npm i' command. The output includes several npm warnings about deprecated packages like 'inflight', 'rimraf', 'glob', and 'object-schema'. It also shows that 273 packages were added and audited, and 102 packages are looking for funding. There were 0 vulnerabilities found. The left side of the screen shows the project file tree under 'VITE-PROJECT'. Two files are highlighted with red boxes: 'node_modules' and 'package-lock.json'. Other visible files include 'public', 'src' (with 'assets' folder), 'App.css', 'App.jsx', 'index.css', 'main.jsx', '.eslintrc.cjs', '.gitignore', 'index.html', 'package.json', 'README.md', and 'vite.config.js'. The terminal path is 'C:\react-home\vite-project>'.

```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not
use it. Check out lru-cache if you want a good and tested way to coalesce async requests
by a key value, which is much more comprehensive and powerful.
npm warn deprecated @humanwhocodes/config-array@0.11.14: Use @eslint/config-array instead
npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated @humanwhocodes/object-schema@2.0.3: Use @eslint/object-schema instead

added 273 packages, and audited 274 packages in 51s

102 packages are looking for funding
  run `npm fund` for details

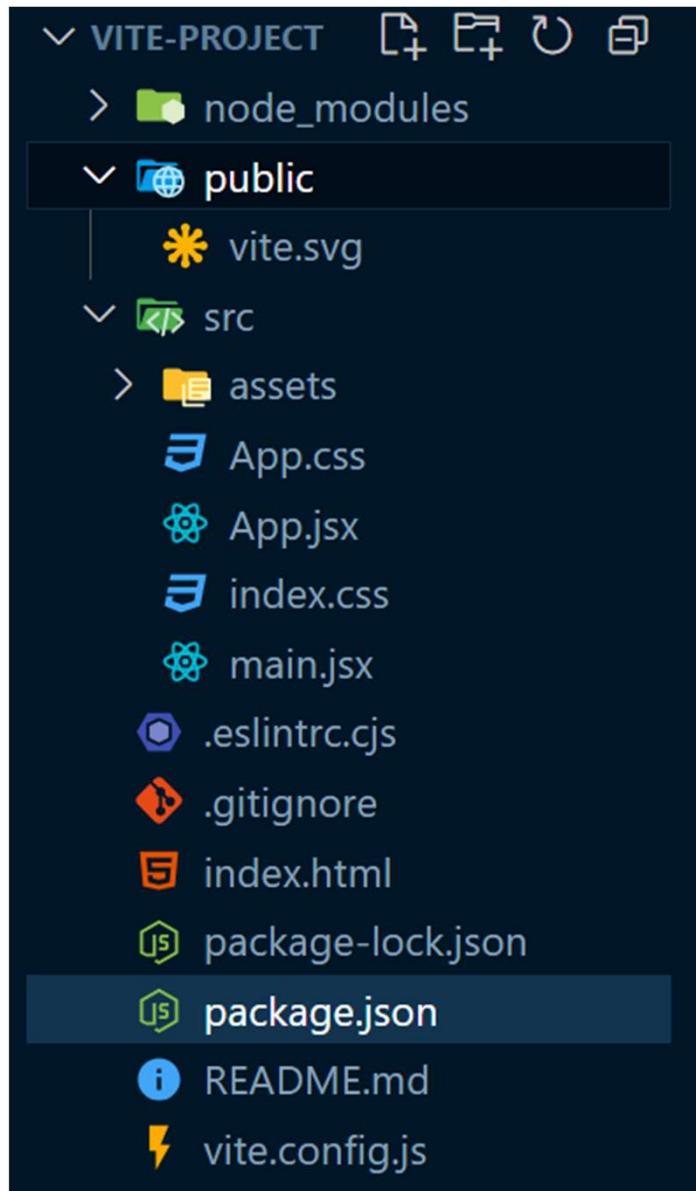
found 0 vulnerabilities

C:\react-home\vite-project>
```

Vite (Cont.)

■ Project Structure

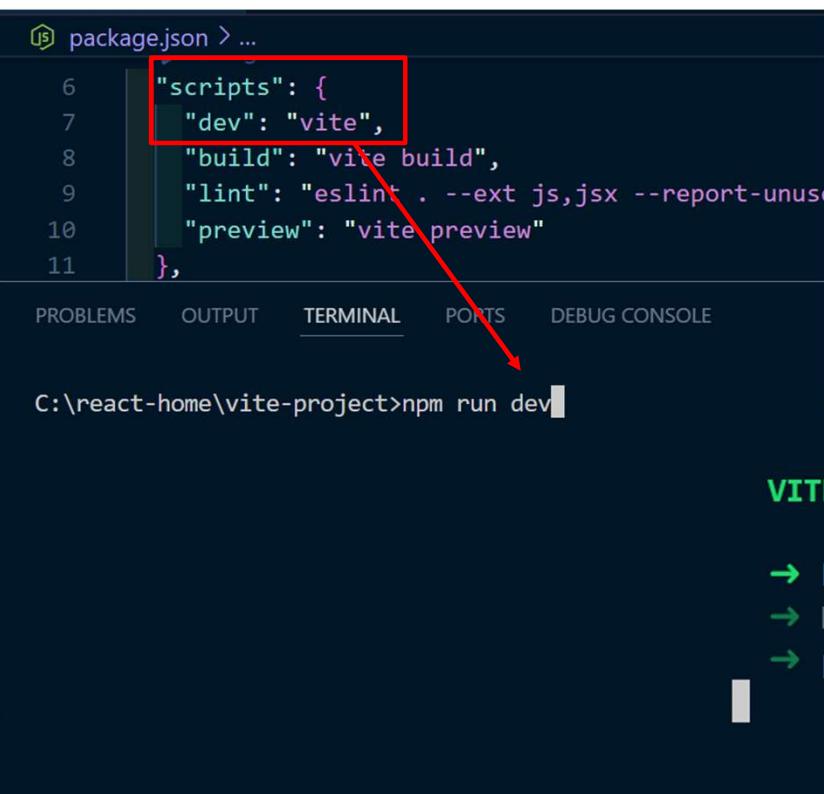
```
└── index.html  
└── src  
    ├── main.js  
    └── App.vue  
└── package.json  
└── vite.config.js
```



Vite (Cont.)

■ Start Project

\$ **npm run dev**



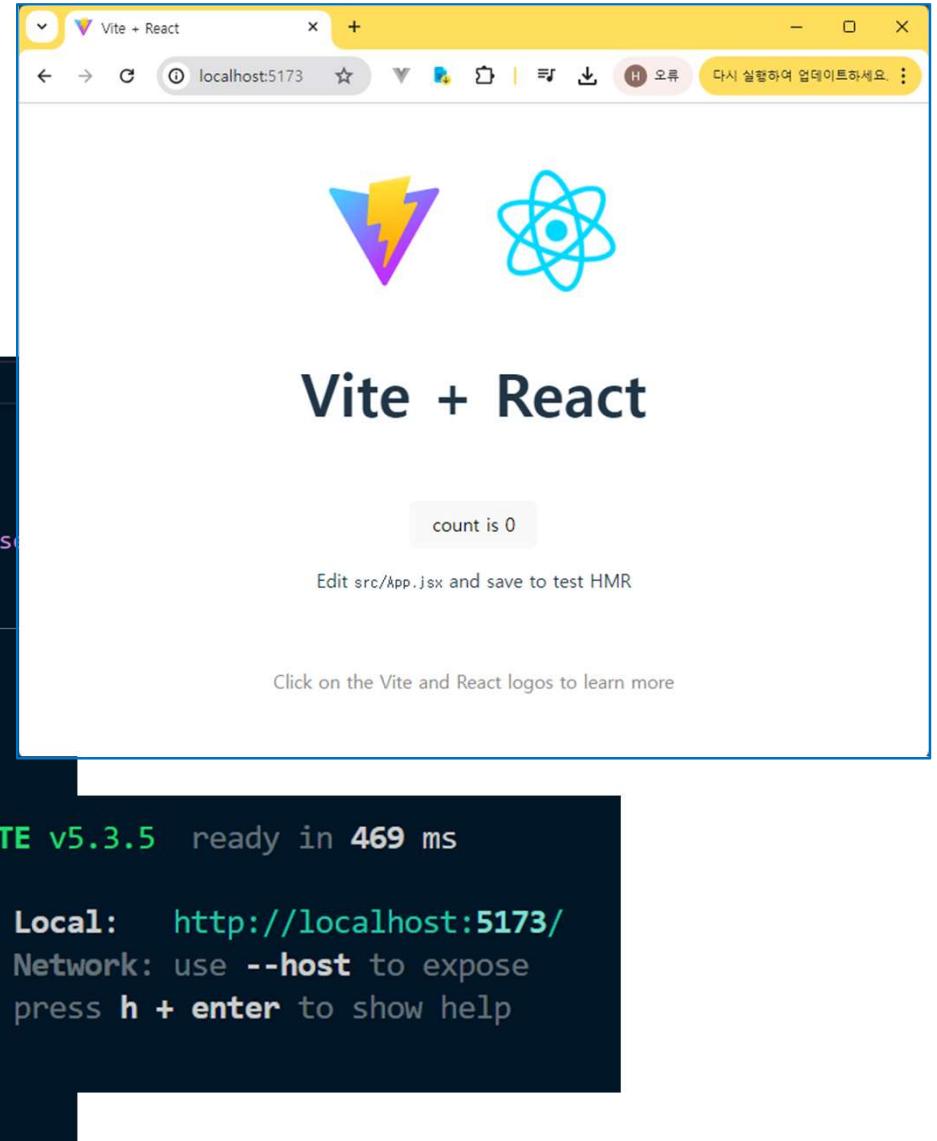
```
package.json > ...
6   "scripts": {
7     "dev": "vite",
8     "build": "vite build",
9     "lint": "eslint . --ext js,jsx --report-unus
10    "preview": "vite preview"
11  },
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

C:\react-home\vite-project>npm run dev

OPEN EDITORS

- package.json
- VITE-PROJECT
 - node_modules
 - public
 - vite.svg
 - src
 - assets
 - App.css
 - App.jsx
 - index.css
 - main.jsx
 - .eslintrc.cjs
 - .gitignore
 - index.html
 - package-lock.json
 - package.json
 - README.md
 - vite.config.js



Vite (Cont.)

■ Start Project

```
$ npm run dev
```

```
VITE v5.3.5  ready in 469 ms

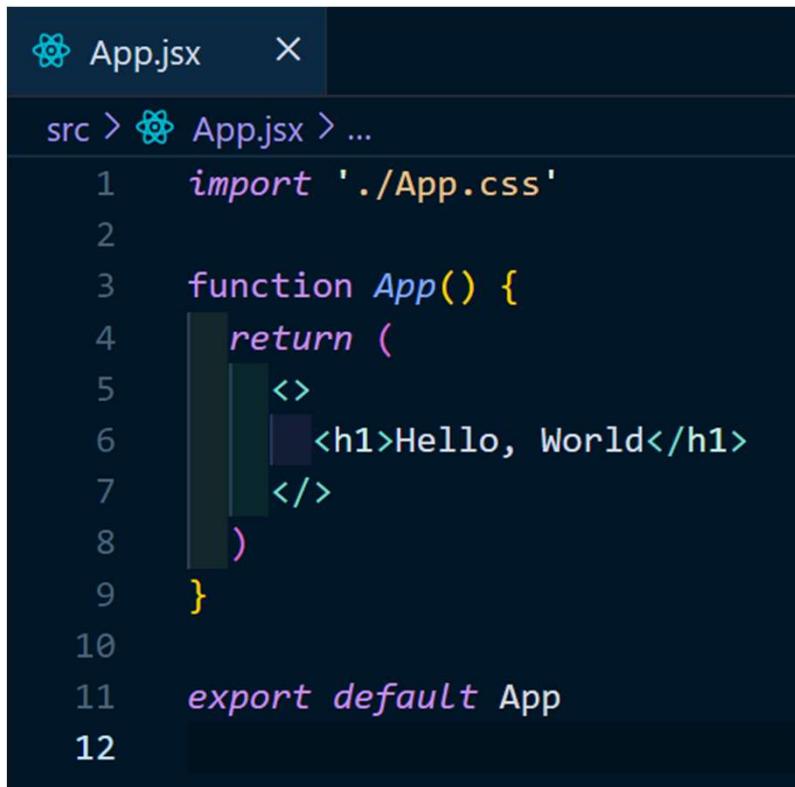
→ Local:  http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
h

Shortcuts
press r + enter to restart the server
press u + enter to show server url
press o + enter to open in browser
press c + enter to clear console
press q + enter to quit
```

Vite (Cont.)

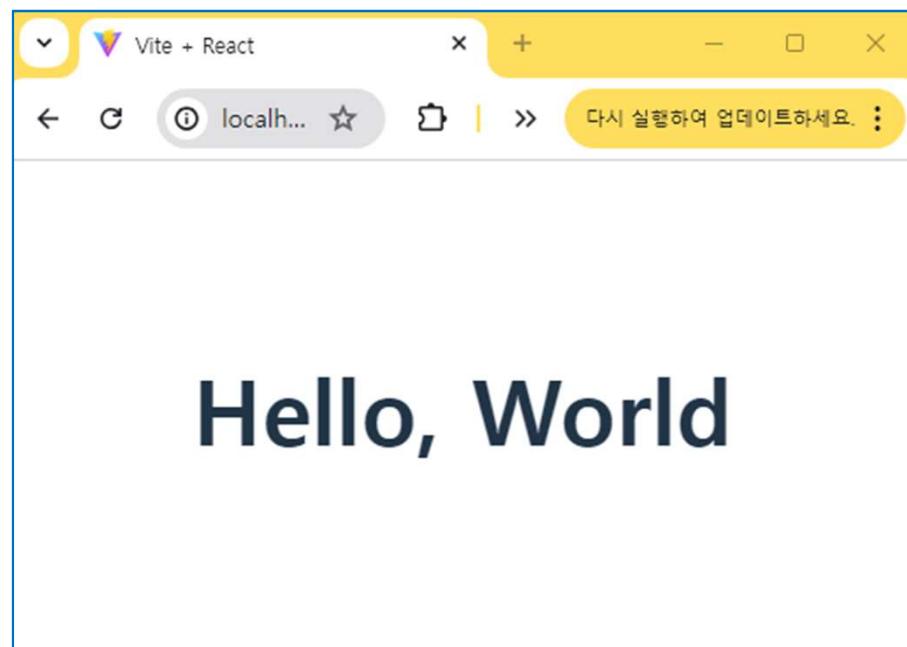
■ Start Project

```
$ npm run dev
```



A screenshot of a code editor showing the contents of the `App.jsx` file. The code is as follows:

```
App.jsx
src > App.jsx > ...
1 import './App.css'
2
3 function App() {
4   return (
5     <>
6       <h1>Hello, World</h1>
7     </>
8   )
9 }
10
11 export default App
12
```



create-react-app

■ 확인

- \$ `npm info create-react-app`

■ 설치

- React Project를 생성하는 Boilerplate 설치
- 어느 경로에서든지 프로젝트를 생성할 수 있도록 global에 설치한다.
- \$ `npm i -g create-react-app`

■ 설치 버전 확인

- \$ `create-react-app --version`

■ 프로젝트 생성

- \$ `create-react-app 프로젝트이름`
- \$ `create-react-app 프로젝트이름 --template typescript`

create-react-app (Cont.)

- 프로젝트로 진입
 - \$ cd **프로젝트이름**
- 프로젝트 시작
 - \$ **npm start**
- 프로젝트 종료
 - \$ **ctrl + c**
- 수업 예제는 bootstrap으로 작성되어 있어 bootstrap을 기본 template5에 먼저 설치
 - \$ **npm i bootstrap@5** => 4의 마지막 버전
 - \$ **npm i bootstrap@5.1** => 4.5의 마지막 버전
 - \$ **npm i bootstrap@5.2.0-beta1** => 매칭되는 버전 설치
 - \$ **npm I bootstrap@5.3**

create-react-app (Cont.)

■ `npx create-react-app .`

```
PS D:\ReactHome> npx create-react-app .
npm notice
npm notice New minor version of npm available! 10.7.0 -> 10.8.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.2
npm notice To update run: npm install -g npm@10.8.2
npm notice
npm error code ENOENT
npm error syscall lstat
npm error path C:\Users\MZC-USER\AppData\Roaming\npm
npm error errno -4058
npm error enoent ENOENT: no such file or directory, lstat 'C:\Users\MZC-USER\AppData\Roaming\npm'
npm error enoent This is related to npm not being able to find a file.
npm error enoent

npm error A complete log of this run can be found in: C:\Users\MZC-USER\AppData\Local\npm-cache\_logs\2024-07-11
T19_53_53_534Z-debug-0.log
PS D:\ReactHome>
```

create-react-app (Cont.)

- `npx create-react-app .`

```
PS D:\ReactHome> npm uninstall -g create-react-app
```

```
up to date in 772ms
```

```
PS D:\ReactHome>
```

create-react-app (Cont.)

■ `npx create-react-app .`

```
Creating a new React app in D:\reacthome.
```

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

```
added 1483 packages in 3m
```

```
262 packages are looking for funding  
  run `npm fund` for details
```

```
Initialized a git repository.
```

```
Installing template dependencies using npm...
```

```
added 63 packages, and changed 1 package in 16s
```

```
262 packages are looking for funding  
  run `npm fund` for details
```

```
Removing template package using npm...
```

Working Directory 이름은 반드시 영문소문자로...

```
Success! Created reacthome at D:\reacthome  
Inside that directory, you can run several commands:
```

```
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 D:\reacthome  
npm start
```

```
Happy hacking!  
PS D:\reacthome> █
```

create-react-app (Cont.)

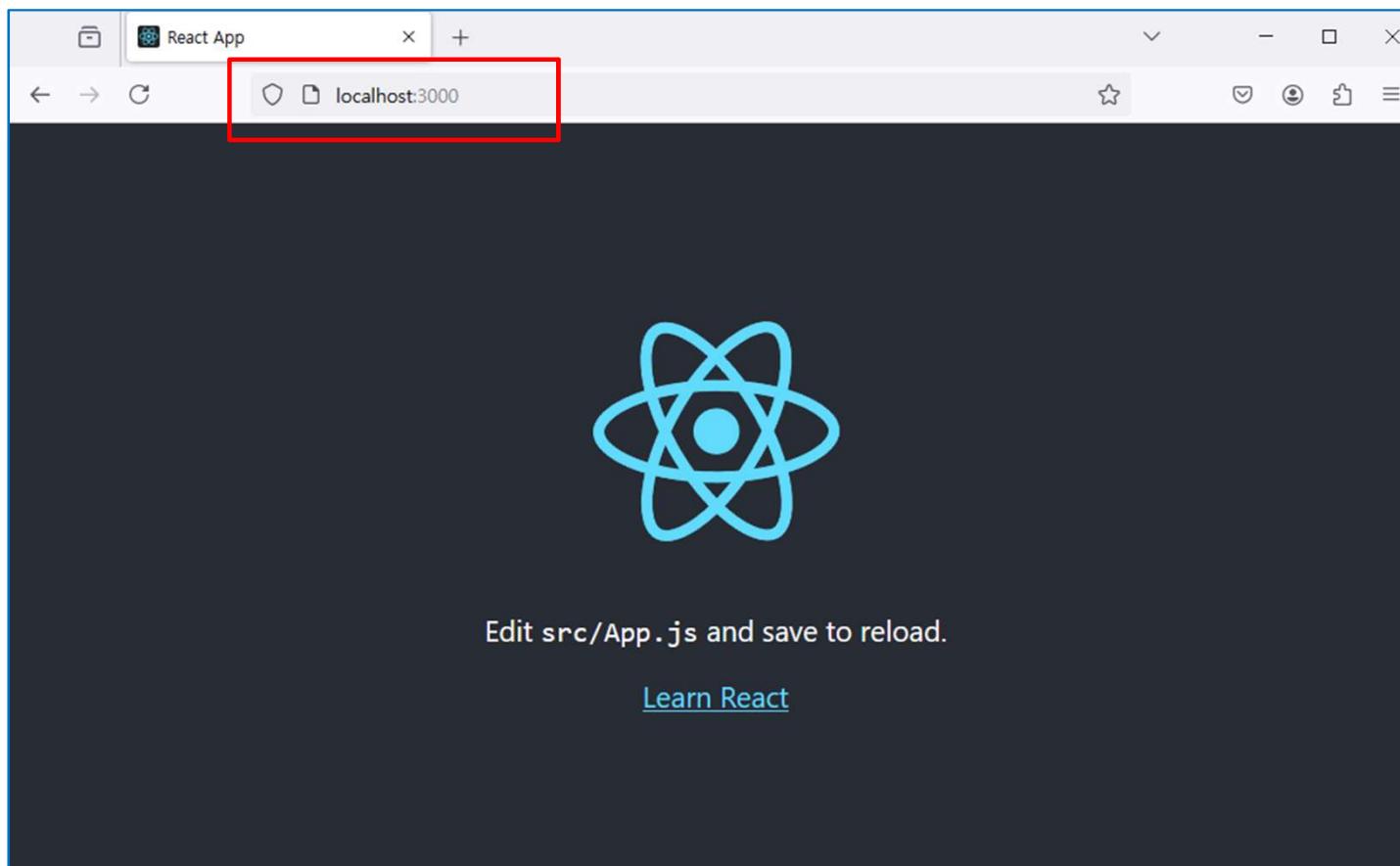
■ npm start

The screenshot shows two windows side-by-side. On the left is a Windows Defender Firewall configuration window titled 'Windows 보안' (Windows Security). It displays a message about network access and lists 'Node.js JavaScript Runtime' under '제작자' (Creator). A red box highlights the '허용' (Allow) button at the bottom. On the right is the '고급 보안이 포함된 Windows Defender 방화벽' (Advanced Firewall Settings) window. It shows a list of inbound rules. Two entries for 'Node.js JavaScript Runtime' are highlighted with a red box. The table below shows the details of these rules:

이름	그룹	프로필	사용
Node.js JavaScript Runtime		공용	예
Node.js JavaScript Runtime		공용	예
Firefox (C:\Program Files\Mozilla Firefox)		개인	예
Firefox (C:\Program Files\Mozilla Firefox)		개인	예
IntelliJ IDEA 2024.1.4		공용	예
IntelliJ IDEA 2024.1.4		공용	예
Microsoft Office Outlook		공용	예
Node.js JavaScript Runtime		공용	예
Node.js JavaScript Runtime		공용	예
Visual Studio Code		공용	예
Visual Studio Code		공용	예
@{Microsoft.Windows.Search_1.14.15.1904...}	@{Microsoft.Windows.Searc...	도메인...	예
@{Microsoft.Windows.SecHealthUI_10.0.1...}	@{Microsoft.Windows.SecHe...	도메인...	예
Skype	(78E1CD88-49E3-476E-B926-...)	모두	예
Skype	(78E1CD88-49E3-476E-B926-...)	모두	예
AllJoyn 라우터(TCP-In)	AllJoyn 라우터	도메인...	예
AllJoyn 라우터(UDP-In)	AllJoyn 라우터	도메인...	예
BranchCache 콘텐츠 검색(HTTP-In)	BranchCache - 콘텐츠 검색(...)	모두	아니요
BranchCache 피어 검색(WSD-In)	BranchCache - 피어 검색(W...	모두	아니요
BranchCache 호스트 캐시 서버(HTTP-In)	BranchCache - 호스트 캐시 ...	모두	아니요

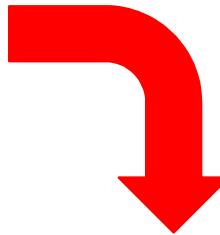
create-react-app (Cont.)

■ npm start



추가 옵션)Reacjs code snippets – VSCode Extension

```
1 ncc Expected an assignment or function call and instead sa
  └─ rcc reactClassCompoment
    └─ rcc reactClassComponent
    └─ rccp reactClassComponentPropTypes
    └─ rccp reactClassComponentPropTypes
    └─ rfcfc reactClassComponentWithMethods
    └─ rcjc reactJustClassComponent
  └─ RTCCertificate
  └─ RTCIceCandidate
  └─ RTCDataChannel
  └─ RTCDataChannelEvent
  └─ RTCPeerConnection
  └─ RTCPeerConnectionIceErrorEvent
```



```
1 import React, { Component } from 'react'
2
3 export default class rcc extends Component {
4   render() {
5     return (
6       <div>
7         </div>
8     )
9   }
10 }
11 }
```

추가 옵션)Reacjs code snippets – VSCode Extension (Cont.)

Keyword	Description
RCC	기본 React Component Code 생성
RCCP	React Component를 Property Type과 함께 생성
RCFC	React Component를 생명주기 함수와 함께 생성
RPC	React Pure Component 생성
RSC	함수형 Component 생성
RSCP	함수형 Component를 Property Type과 함께 생성

create-react-app (Cont.)

■ `npx create-react-app react-first`

```
C:\react-home>npm create-react-app react-first
Unknown command: "create-react-app"
```

To see a list of supported npm commands, run:
`npm help`

```
C:\react-home>npx create-react-app react-first
```

Creating a new React app in `C:\react-home\react-first`.

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

.8

create-react-app (Cont.)

■ npm start

We suggest that you begin by typing:

```
cd react-first  
npm start
```

Happy hacking!

Compiled successfully!

You can now view react-first in the browser.

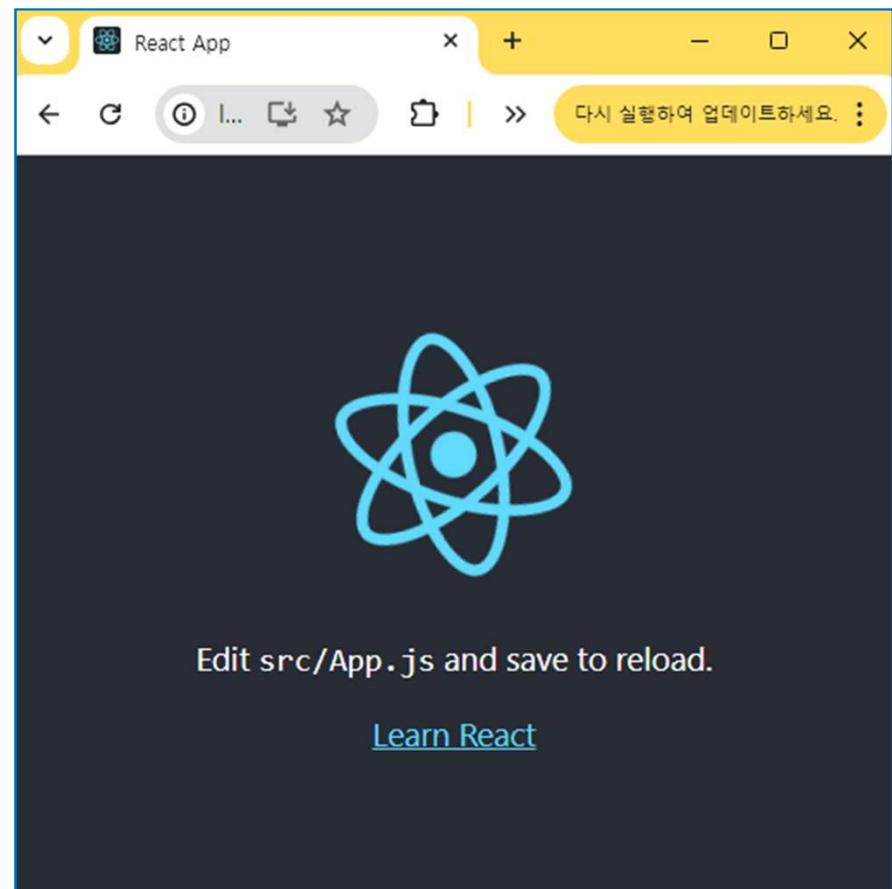
Local: <http://localhost:3000>

On Your Network: <http://192.168.0.16:3000>

Note that the development build is not optimized.

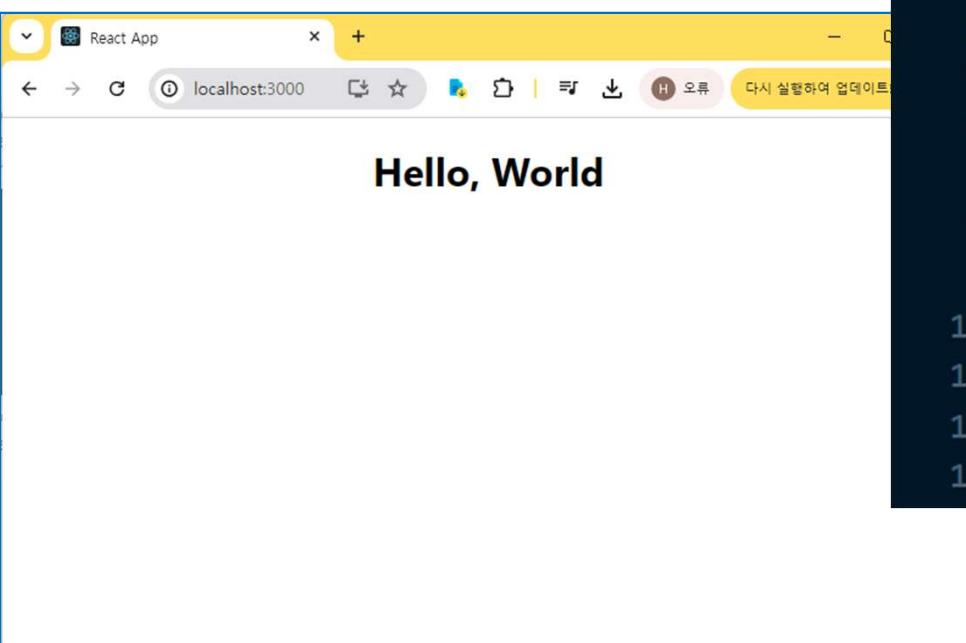
To create a production build, use `npm run build`.

webpack compiled **successfully**

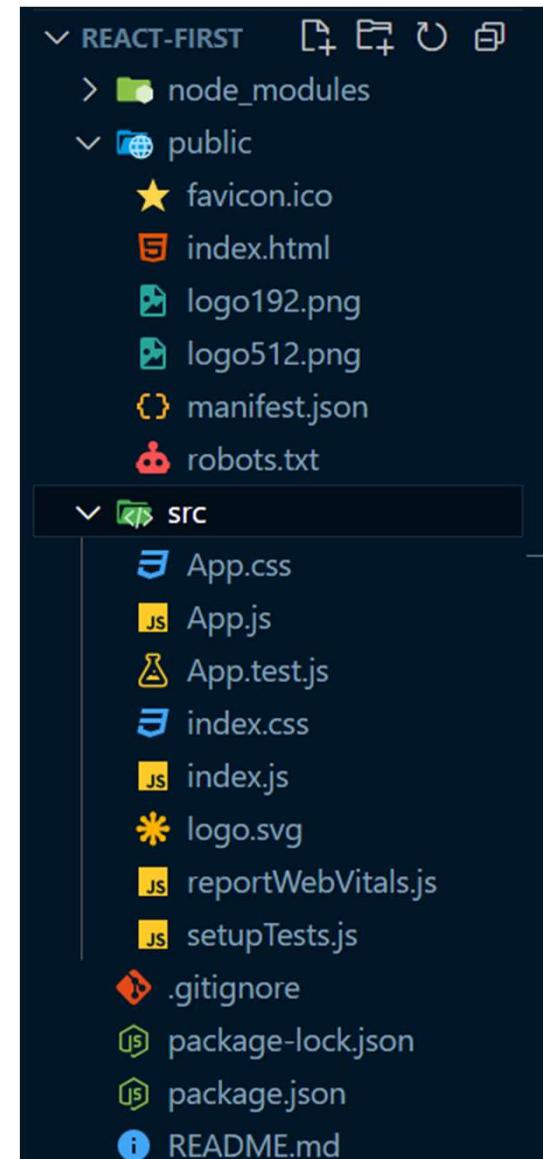


create-react-app (Cont.)

■ npm start



```
JS App.js 1 X
src > JS App.js > App
  2 import './App.css';
  3
  4 function App() {
  5   return (
  6     <div className="App">
  7       <h1>Hello, World</h1>
  8     </div>
  9   );
 10 }
 11
 12 export default App;
 13
```



ESLint Configuration

- File > Preferences > Settings > extensions > ESLint

```
C: > Users > MZC01-HENRY > AppData > Roaming > Code > User > {} settings.json > ...

13 "git.enableSmartCommit": true,
14 "git.autofetch": true,
15 "redhat.telemetry.enabled": true,
16 "cfnLint.path": "cfn-lint --ignore-checks E3012",
17 "files.autoSave": "onFocusChange",
18 "editor.tabSize": 2,
19 "editor.defaultFormatter": "vscode.typescript-language-features",
20 "notebook.formatOnSave.enabled": true,
21 "explorer.compactFolders": false,
22 "eslint.options": {
23     "no-unused-vars": "off",
24     "react/prop-types": "off",
25 }
26 }
```

Online Editor – JSFiddle(<https://jsfiddle.net>)

The screenshot shows the JSFiddle web application. At the top, there's a blue header bar with various boilerplate options like jQuery, Vue, React, etc., and links to roadmap, bug tracker, and service status. Below the header is a toolbar with Run, Save, Tidy, Collaborate, and Update buttons. The main area is divided into three panes: Fiddle meta (containing title, description, resources, and async requests), HTML/CSS code editors, and a Result pane.

Start with a boilerplate:

- jQuery
- Vue
- React
- React + JSX
- Preact
- TypeScript
- CoffeeScript
- SCSS
- CSS Grid (responsive)
- Bootstrap
- PostCSS New

Links:

- Roadmap (vote for features)
- Bug tracker
- About
- Docs
- Service status

JSFiddle is for:

- Demos for docs
- Bug reporting (test-case) for Github Issues
- Presenting code answers on Stack Overflow
- Live code collaboration
- Code snippets hosting

... or just your humble code playground 🙌

Fiddle meta

Untitled fiddle
No description

Add title to make the fiddle visible on your profile page

Resources URL cdnjs

Async requests

Other (links, license)

HTML

```
<div>1</div>
```

CSS

```
.div { 1 }
```

JavaScript + No-Library (pure JS)

```
1 |
```

Result

Online Editor – JSFiddle(<https://jsfiddle.net>) (Cont.)

The screenshot shows the JSFiddle interface with the following elements:

- Header:** "Create a new fiddle - JSFiddle" and a "+" button.
- Address Bar:** "jsfiddle.net/9wuma8by/"
- Toolbar:** Run, Save, Tidy, Collaborate, Update (green), Settings, Sign in.
- Fiddle meta:** Untitled fiddle, No description, Add title to make the fiddle visible on your profile page.
- HTML Tab:** Contains the following code:

```
1 <script src="https://unpkg.com/vue"></script>
2
3 <div id="app">
4
5 </div>
```
- CSS Tab:** Contains a single line of CSS: "1".
- JavaScript + No-Library (pure JS) Tab:** Contains the following Vue.js code:

```
1 new Vue({
2   template : '<p>{{msg}}</p>',
3   data : {
4     msg : 'Hello, World!'
5   }
6 }).$mount('#app')
```
- Output Area:** Displays the output "Hello, World!".
- Resources:** URL (selected), cdnjs.
- Async requests:**
- Other (links, license):**
- Advertisement:** Headway logo with the text "Easiest way to keep your customers".

Online Editor – CodePen(<https://codepen.io>)

The screenshot shows the homepage of CodePen. At the top, there's a navigation bar with a search bar containing "Search CodePen...", a "Sign Up" button, and a "Log In" button. On the left, a sidebar lists various sections: TRY OUR ONLINE EDITOR (with a "Start Coding" button), EXPLORE (Picks, Popular, Topics), GROW (Jobs, CodePen PRO Challenges), COMPANIES (Hire Talent, Create a Team, Advertise), Newsletter, Blog, Docs & Support, and an Adobe Stock promotion. The main content area features a large heading "The best place to build, test, and discover front-end code." followed by a subtext about CodePen being a social development environment for front-end designers and developers. Below this is a "Sign Up for Free" button. To the right, there's a large preview window showing snippets of HTML, SCSS, and JS code. At the bottom, there are three buttons: "Build & Test" (with a build icon), "Learn & Discover" (with a map icon), and "Share Your Work" (with a circular icon). A promotional message for Windows activation is also visible.

TRY OUR ONLINE EDITOR

Start Coding

EXPLORE

Picks

Popular

Topics

GROW

Jobs

CodePen PRO Challenges

COMPANIES

Hire Talent

Create a Team

Advertise

Newsletter

Blog

Docs & Support

MAKE IT WITH **ADOBESTOCK**.
Get 10 free images.

Limited time offer: Get 10 free Adobe Stock images.

CodePen: Build, Test, and Discover

Sign Up Log In

Search CodePen...

The best place to build, test, and discover front-end code.

CodePen is a **social development environment** for front-end designers and developers. Build and deploy a website, show off your work, build test cases to learn and debug, and find inspiration.

Sign Up for Free

HTML

```
<div class="rect"></div>
```

SCSS

```
.rect { background: linear-gradient(-119deg, $gray 0%, $dark-gray 100%); }
```

JS

```
var colors = ["#74B087", "#DE7300", "#74B087"]; function animate() {};
```

Activate Windows
Go to Settings to activate Windows.

Build & Test

Learn & Discover

Share Your Work

Online Editor – CodeSandbox(<https://codesandbox.io>)

