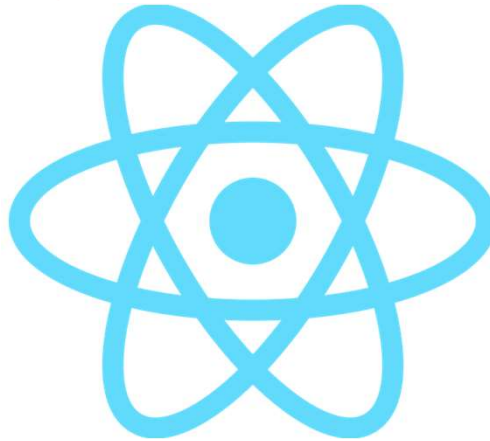


# Configure for Lab



**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
- create-react-app
- Online Editors

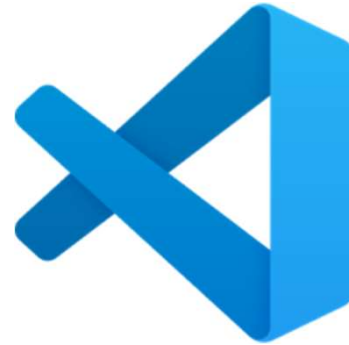
# Web Browsers

- 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>

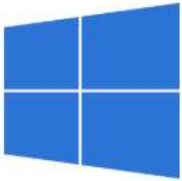


# Visual Studio Code

- <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

↓ **.rpm**

Red Hat, Fedora, SUSE

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

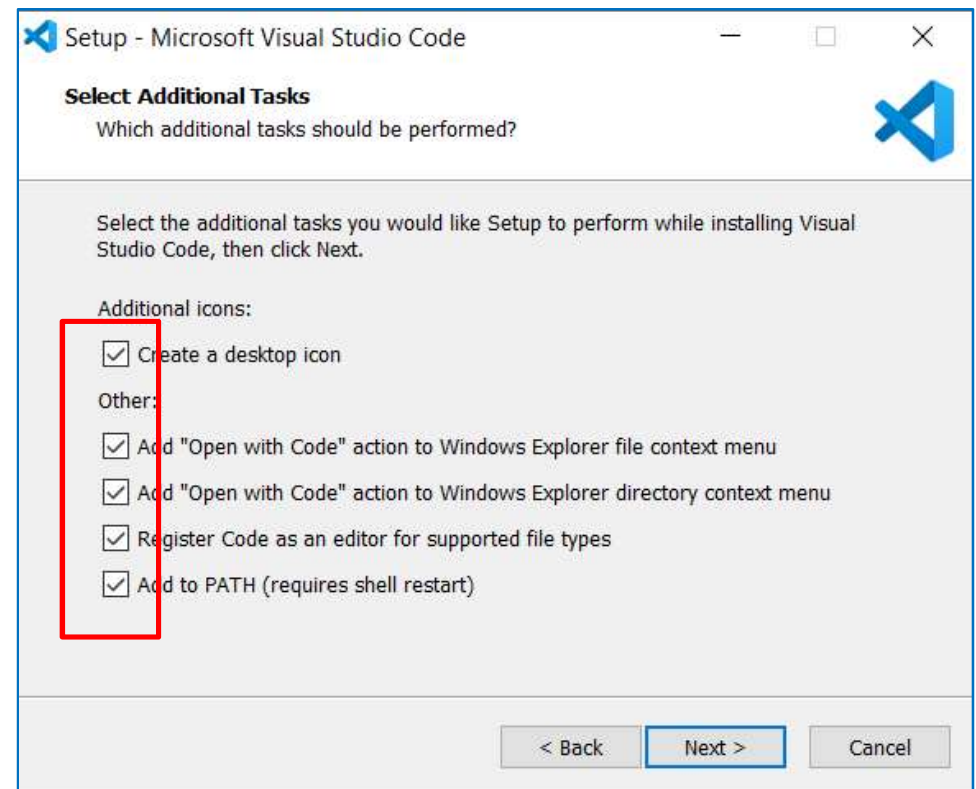
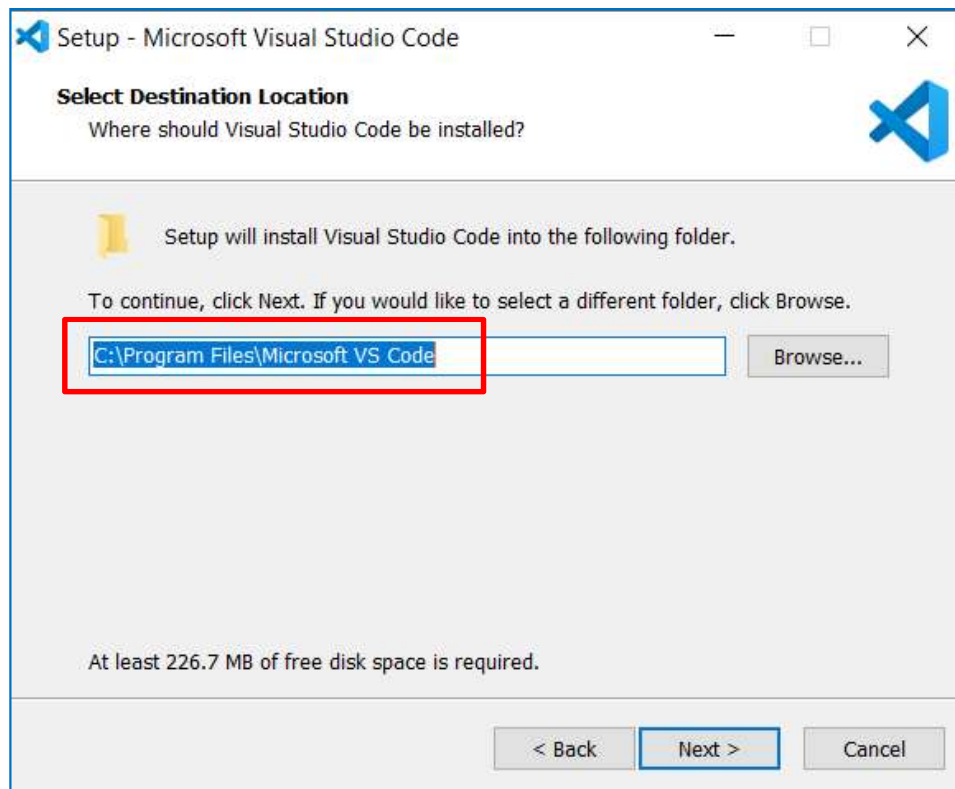
[Snap Store](#)



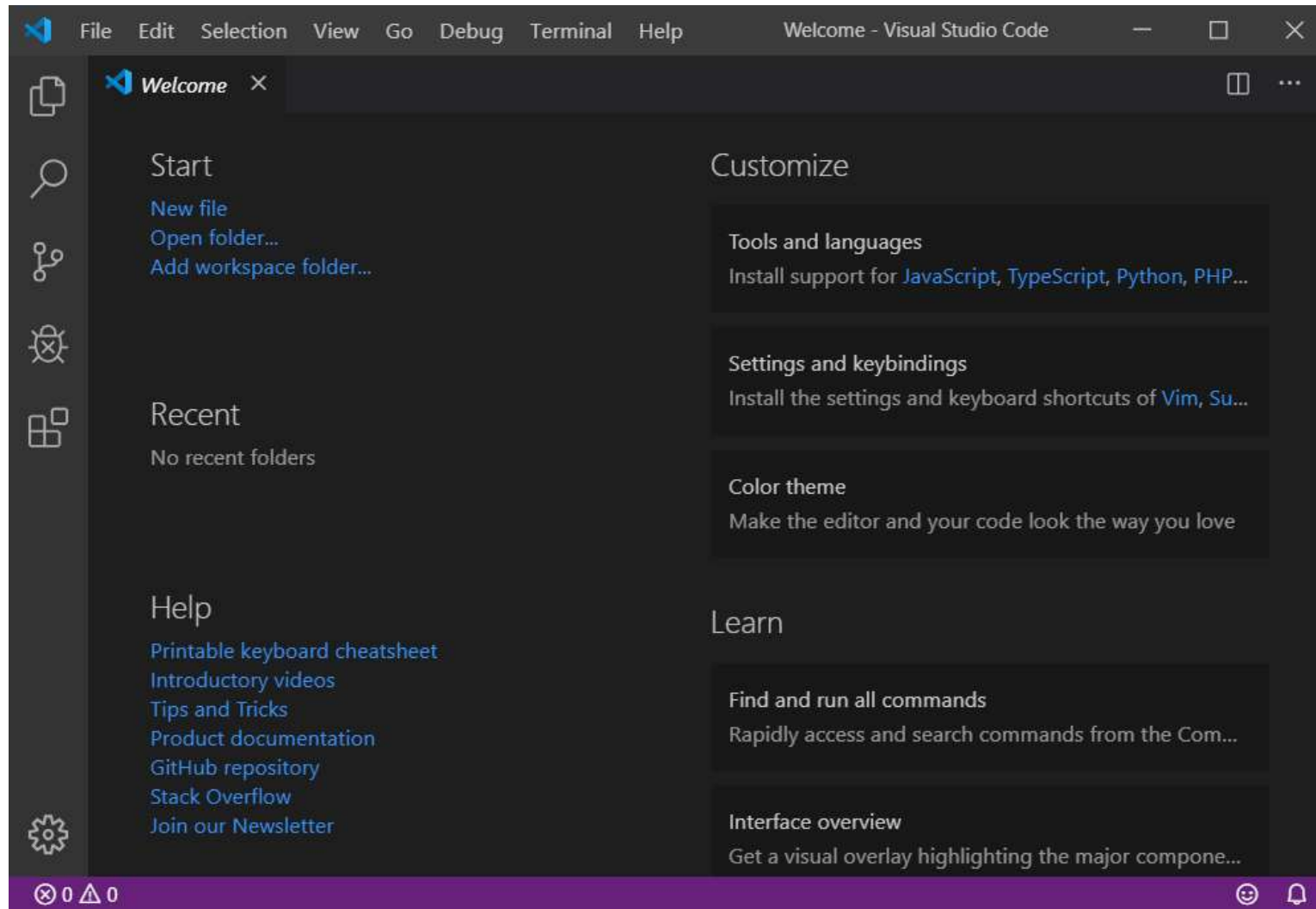
↓ **Mac**

macOS 10.10+

# 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



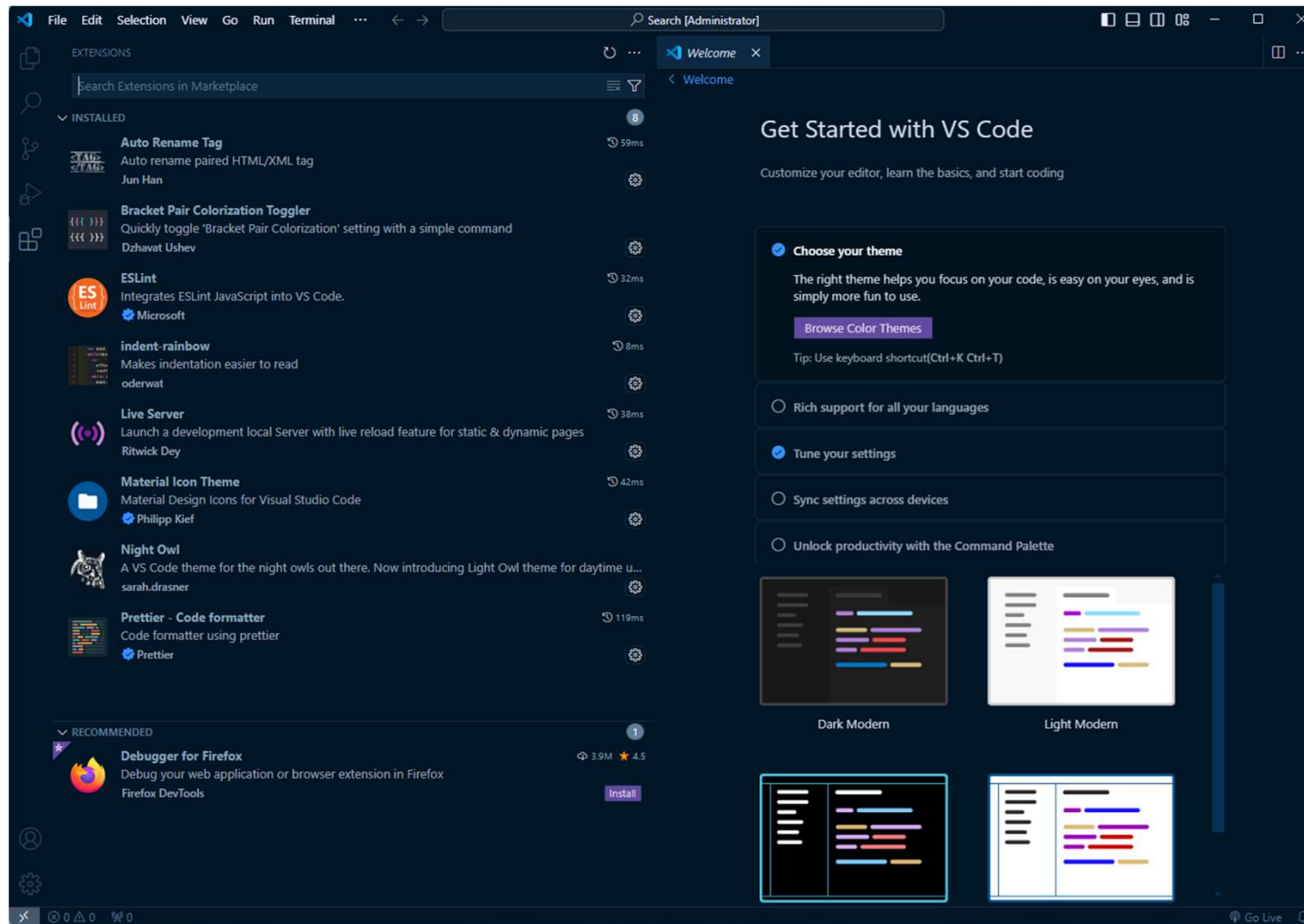
# VSCode Extensions

- 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 기능 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/>

 Learn About Download Blog Docs Certification

Start typing... Ctrl K

## 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 **v20.15.1**<sup>1</sup> with long-term support. Node.js can also be installed via [package managers](#).

Want new features sooner? Get **Node.js v22.4.1**<sup>1</sup> instead.

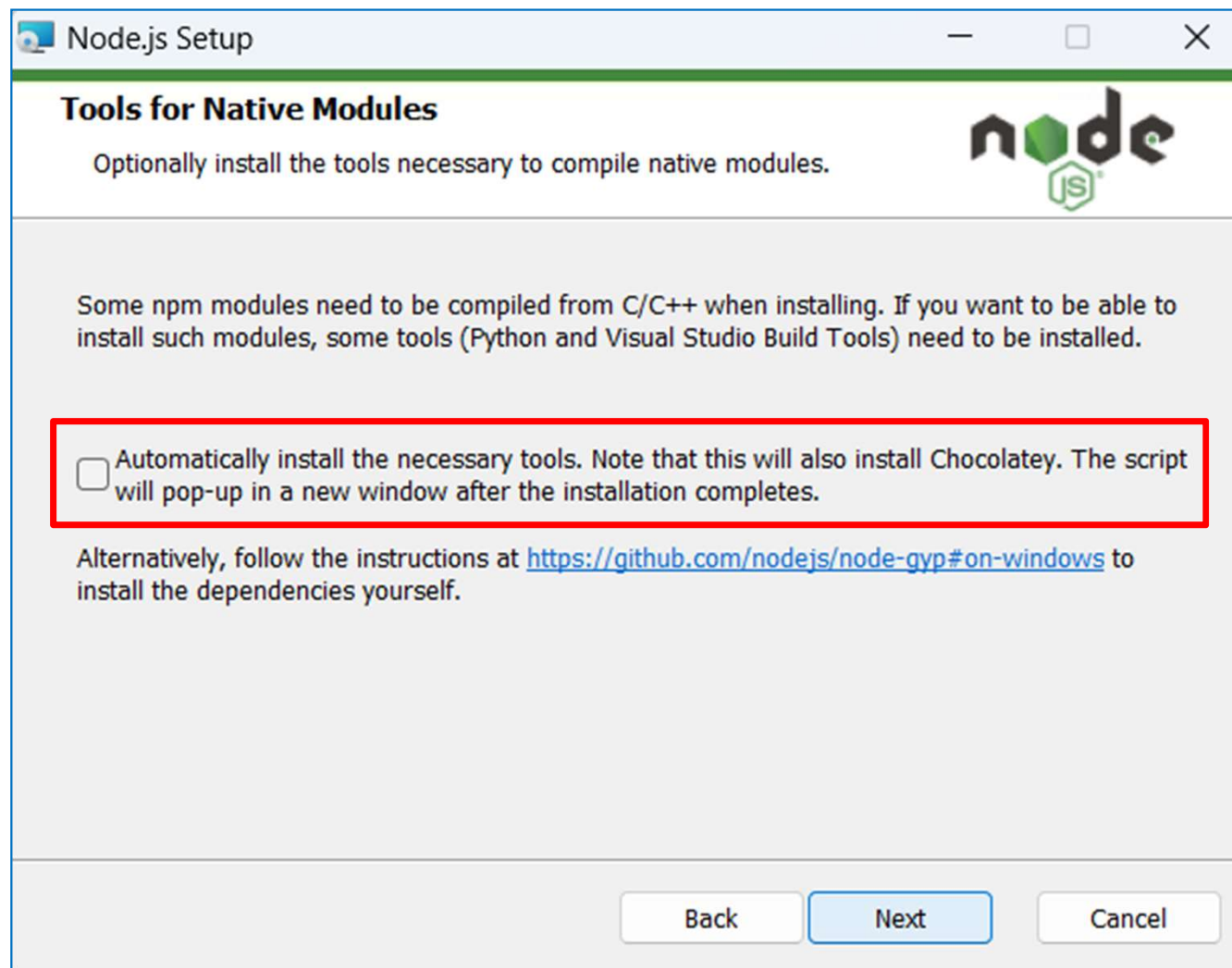
Create an HTTP Server Write Tests Read and Hash a File Streams Pipeline Work with Threads

```
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

Learn more what Node.js is able to offer with our [Learning materials](#).

## Node.js (Cont.)



## Node.js (Cont.)

```
관리자: C:\WINDOWS\system32\cmd.exe
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/>

 **git** --local-branching-on-the-cheap

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient staging areas, and **multiple workflows**.





### 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.

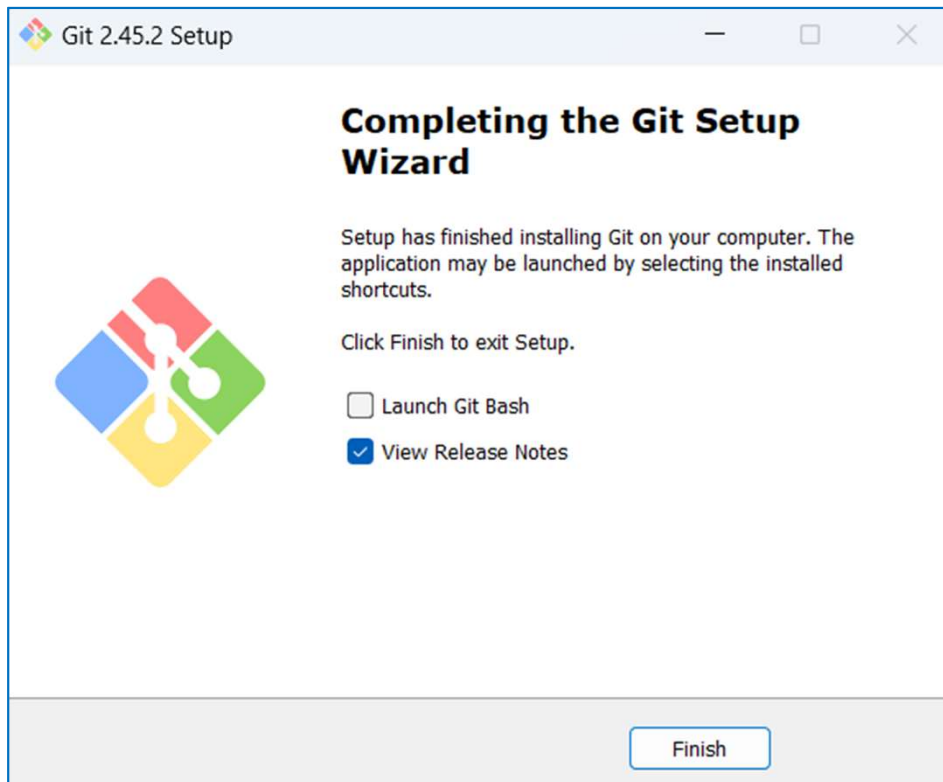




## Git (Cont.)

```
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\system32\cmd.exe
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 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

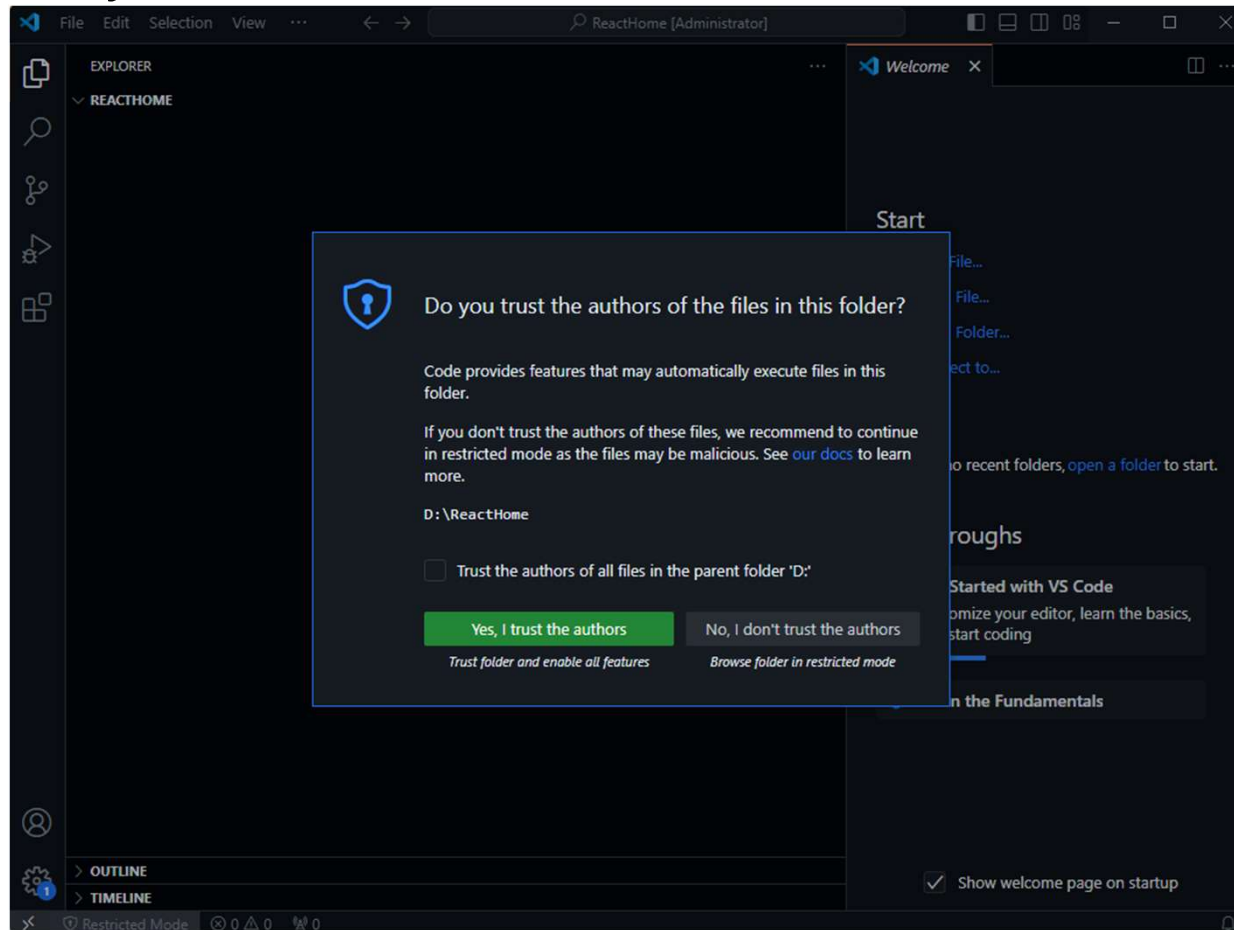
- \$ **npm run test**
- \$ **npm run build**

- start, test 명령은 run을 생략하고 다음 명령으로 실행 가능

- \$ **npm start**

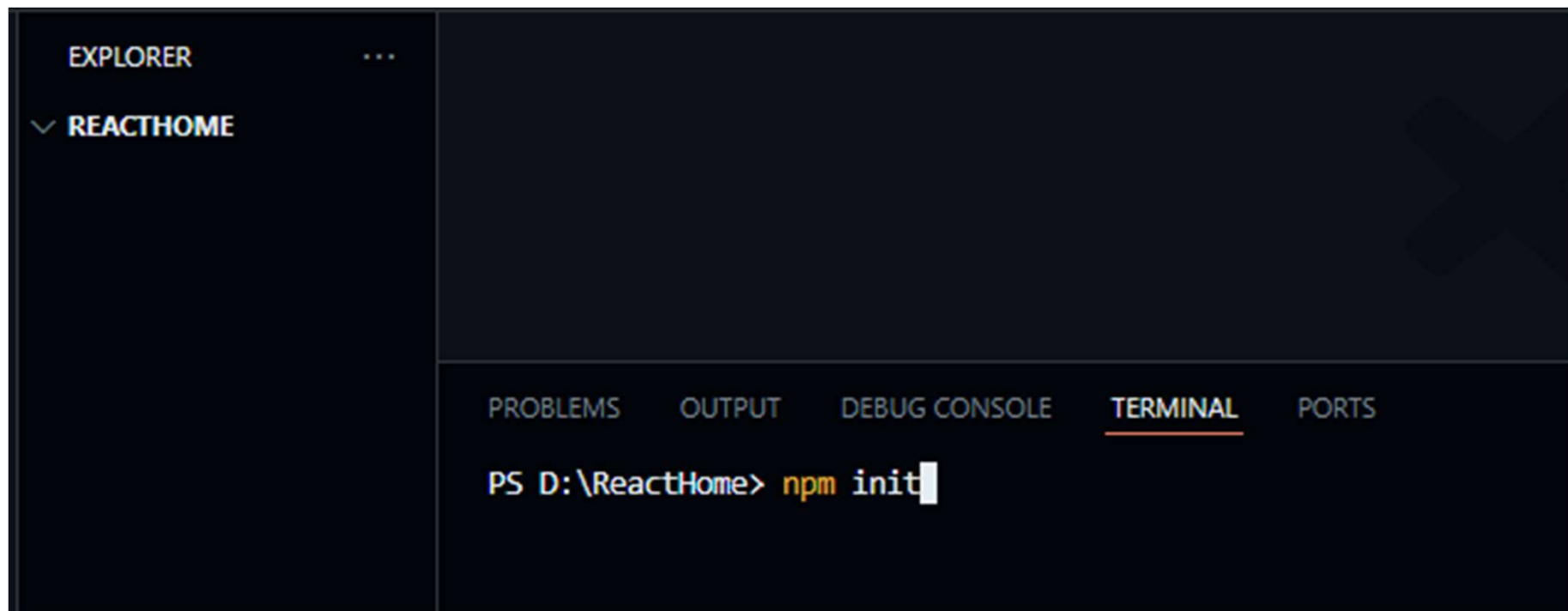
# npm (Cont.)

## ■ Working Directory → ReactHome



## npm (Cont.)

### ■ npm init



# 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>

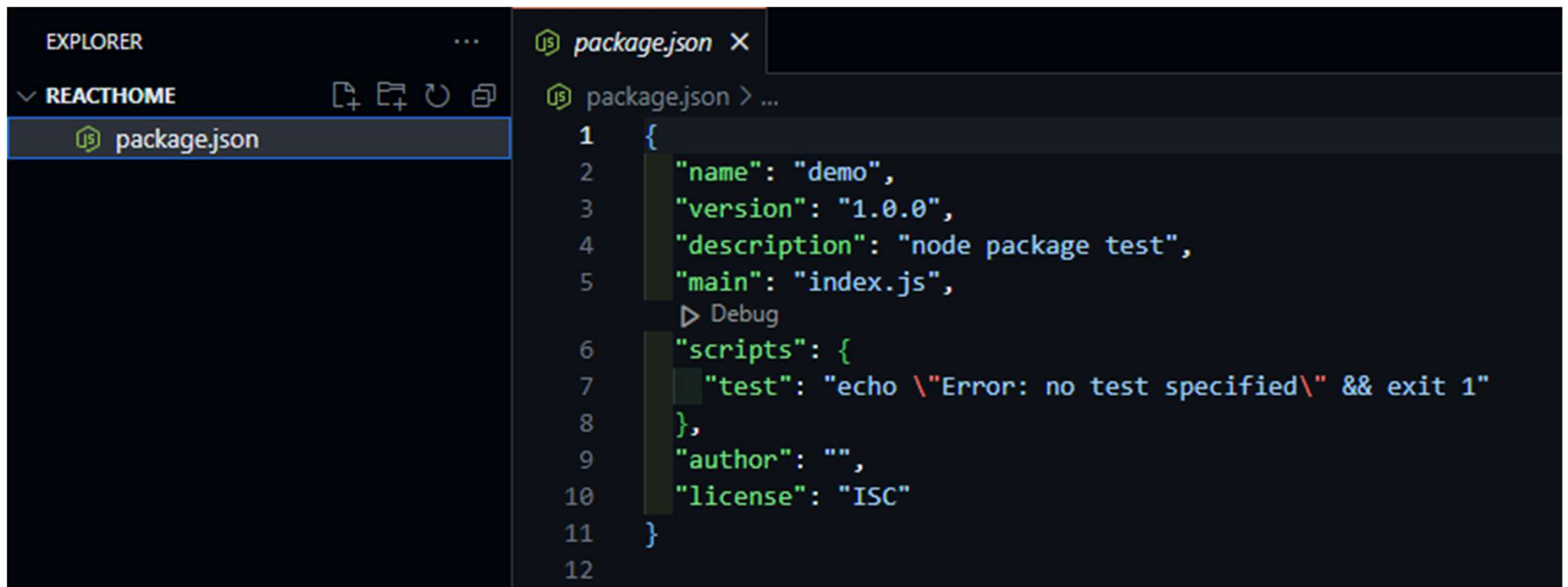
npm notice To update run: `npm install -g npm@10.8.2`

npm notice

PS D:\ReactHome>

## npm (Cont.)

### ■ npm init

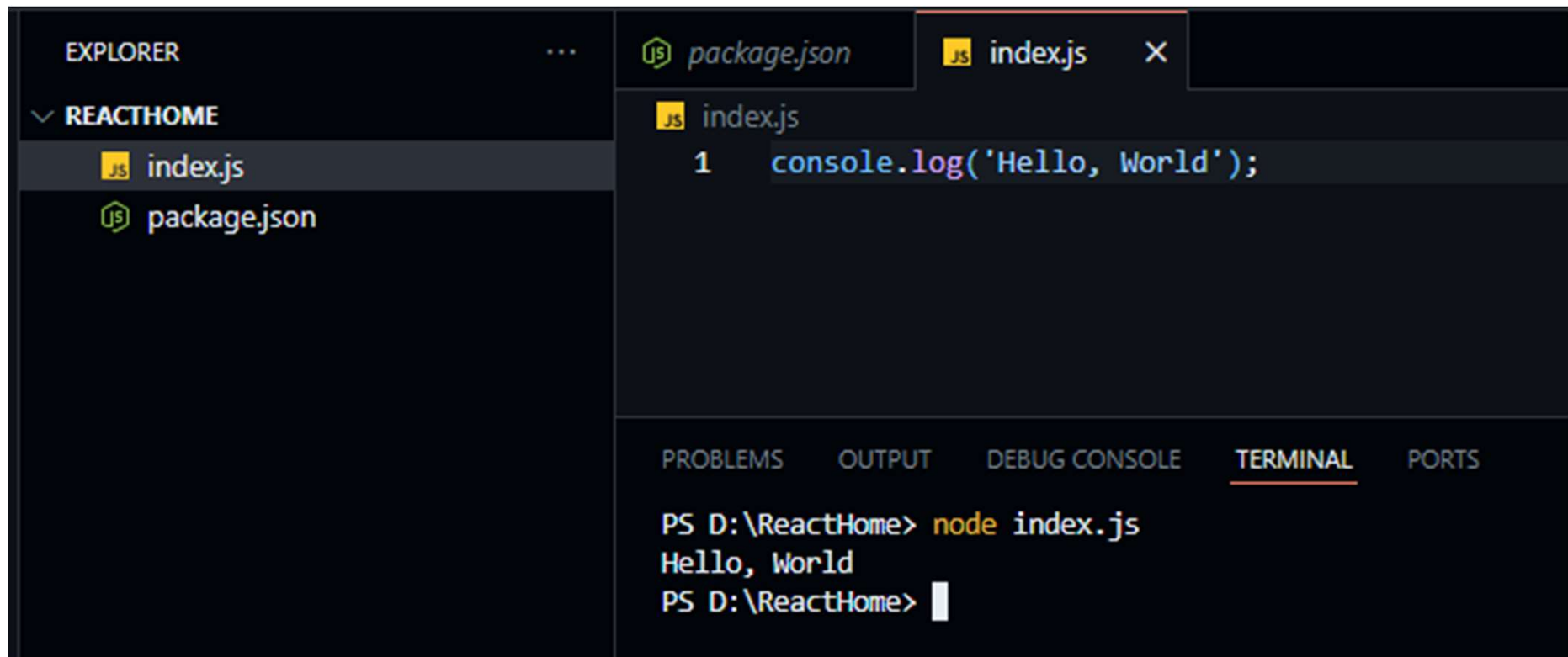


The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a folder named 'REACTHOME' with a file 'package.json' selected. The main editor area displays the content of 'package.json' with line numbers 1 through 12. The file contains the following JSON structure:

```
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 shows a folder named 'REACTHOME' containing two files: 'index.js' and 'package.json'. The 'index.js' file is selected. The main editor area has two tabs: 'package.json' and 'index.js'. The 'index.js' tab is active, showing a single line of code: `1 console.log('Hello, World');`. At the bottom, the TERMINAL panel is open, showing the command `PS D:\ReactHome> node index.js` and its output `Hello, World`. The prompt `PS D:\ReactHome>` is followed by a cursor.

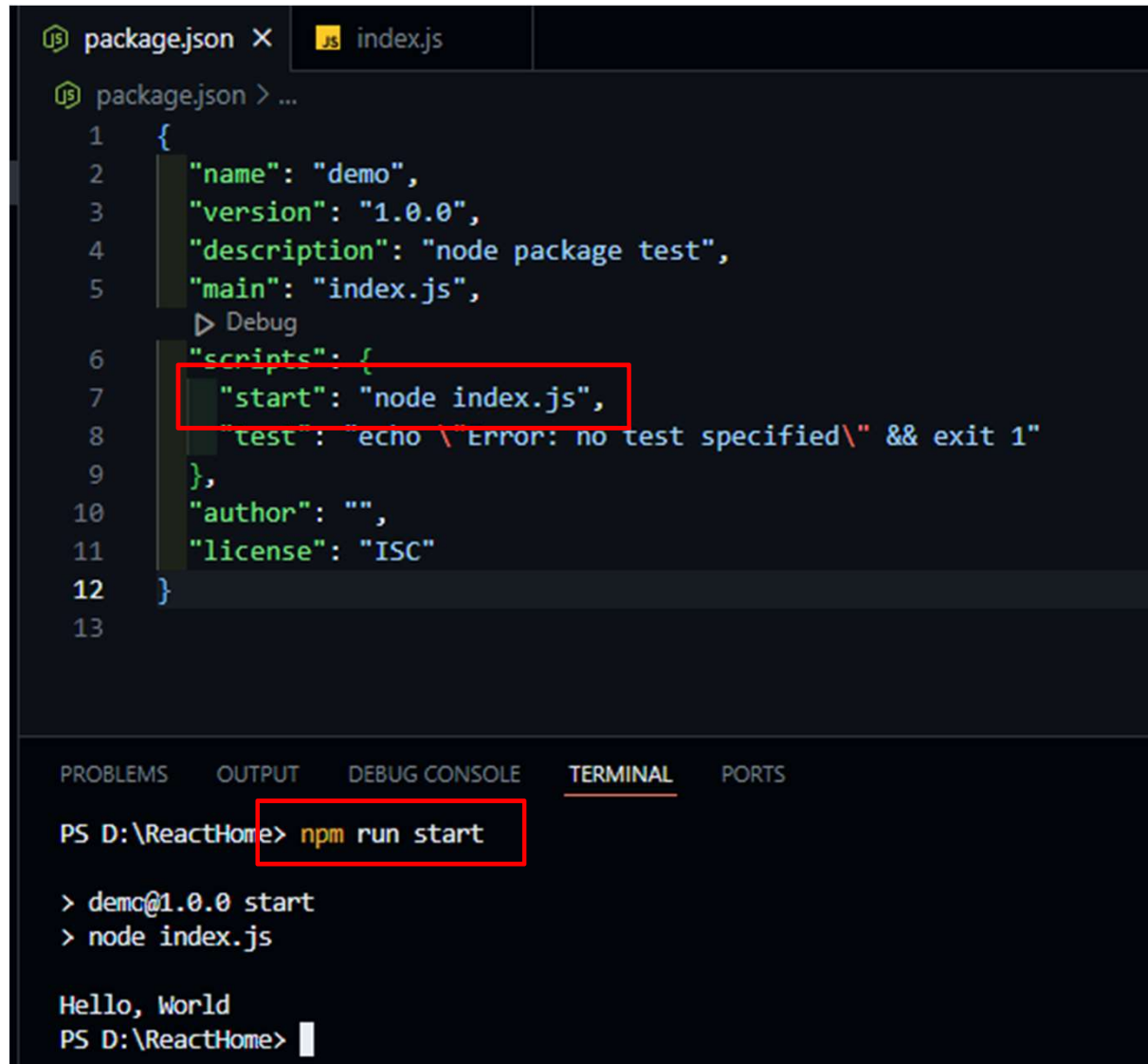
```
EXPLORER
  REACTHOME
    index.js
    package.json

package.json
index.js
1 console.log('Hello, World');

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\ReactHome> node index.js
Hello, World
PS D:\ReactHome> 
```

## npm (Cont.)

### ■ package.json 수정



The screenshot shows a Visual Studio Code editor with two tabs: 'package.json' and 'index.js'. The 'package.json' file is open, displaying a JSON configuration. A red rectangle highlights the 'start' script within the 'scripts' object. Below the editor, the 'TERMINAL' panel is active, showing the command 'npm run start' being executed. Another red rectangle highlights this command. The terminal output shows the command's execution, including the project name and version, the command to run 'index.js', and the output 'Hello, World'.

```
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
```

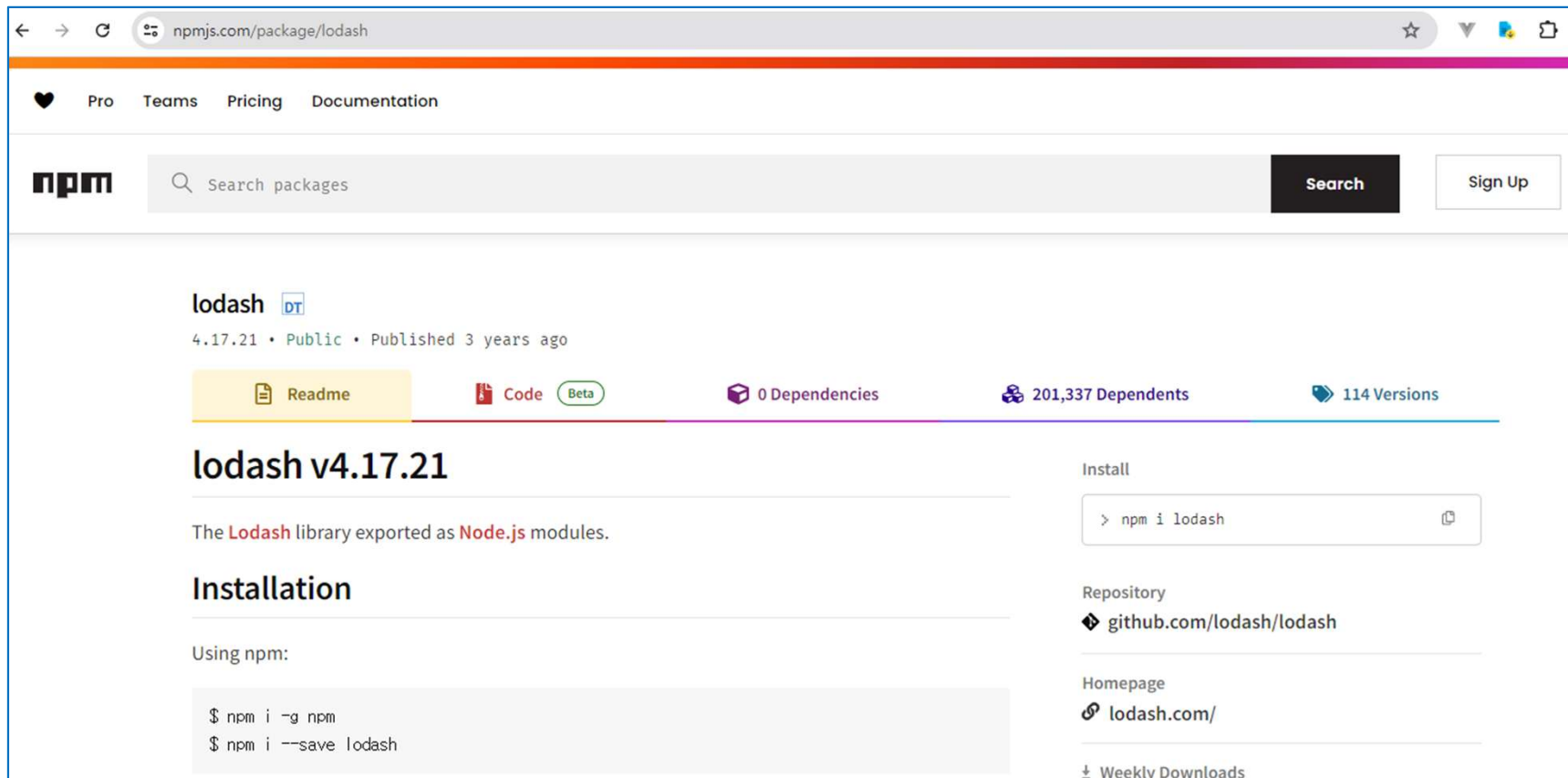
```
PS D:\ReactHome> npm run start

> 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 npm package page for 'lodash'. The browser address bar displays 'npmjs.com/package/lodash'. The page features a navigation bar with links for 'Pro', 'Teams', 'Pricing', and 'Documentation'. Below this is a search bar with the 'npm' logo and a 'Search' button, alongside a 'Sign Up' button. The main content area displays the package name 'lodash' with a 'DT' icon, its version '4.17.21', and its status as 'Public' published '3 years ago'. A horizontal bar contains links for 'Readme', 'Code' (with a 'Beta' badge), '0 Dependencies', '201,337 Dependents', and '114 Versions'. The section 'lodash v4.17.21' describes it as 'The Lodash library exported as Node.js modules.' and includes an 'Installation' section with the command '\$ npm i -g npm' and '\$ npm i --save lodash'. On the right, an 'Install' section shows the command '> npm i lodash' with a copy icon, and a 'Repository' section links to 'github.com/lodash/lodash'. A 'Homepage' section links to 'lodash.com/'. At the bottom right, there is a link for 'Weekly Downloads'.

npmjs.com/package/lodash

Pro Teams Pricing Documentation

npm Search packages Search Sign Up

lodash   
4.17.21 • Public • Published 3 years ago

[Readme](#) [Code](#) [Beta](#) [0 Dependencies](#) [201,337 Dependents](#) [114 Versions](#)

### lodash v4.17.21

The **Lodash** library exported as **Node.js** modules.

#### Installation

Using npm:

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

Install

```
> npm i lodash
```

Repository

[github.com/lodash/lodash](https://github.com/lodash/lodash)

Homepage

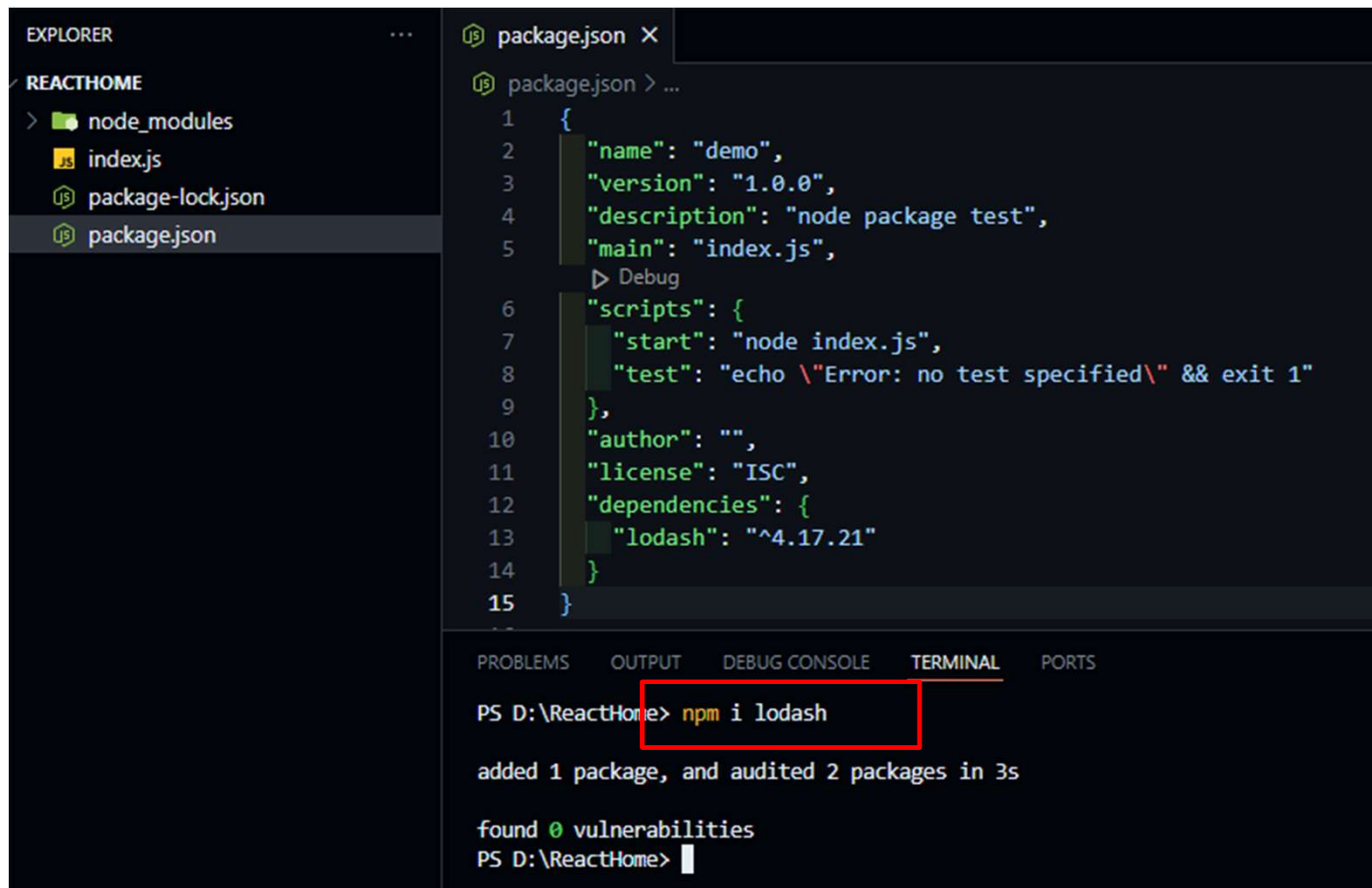
[lodash.com/](https://lodash.com/)

[Weekly Downloads](#)



# npm – 외부 Library 사용하기 (Cont.)

## ■ lodash 설치



The screenshot shows the Visual Studio Code interface with a project named 'REACTHOME'. The Explorer sidebar on the left shows the file structure: 'node\_modules', 'index.js', 'package-lock.json', and 'package.json'. The 'package.json' file is open in the editor, displaying the following JSON configuration:

```
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 }
```

Below the editor, the TERMINAL panel is active, showing the command `npm i lodash` being executed. The output of the command is as follows:

```
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

```
index.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
```

```
(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)
```

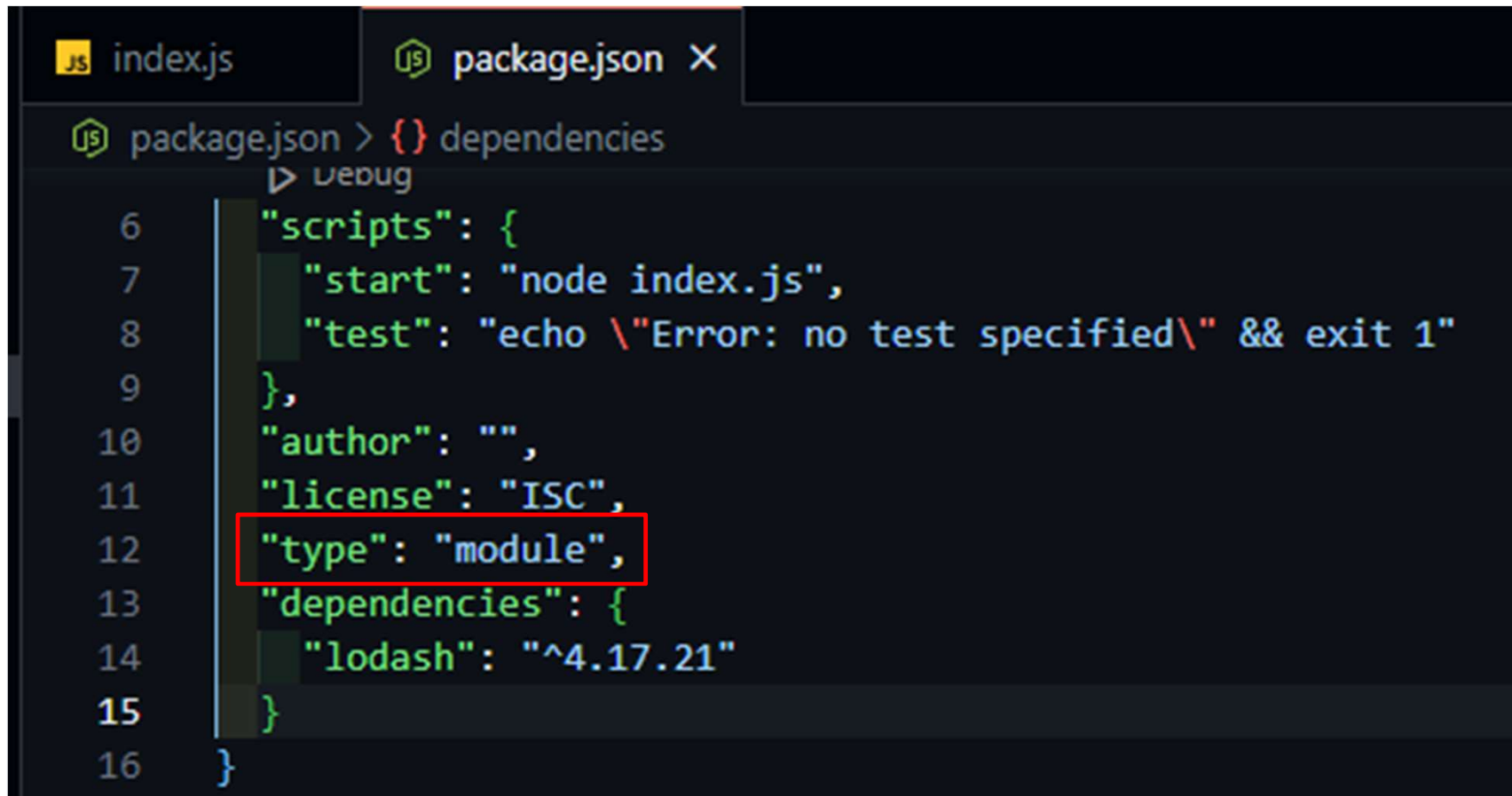
```
at node:internal/main/run_main_module:28:49
```

```
Node.js v20.15.1
```

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

## npm – 외부 Library 사용하기 (Cont.)

### ■ lodash 이용하기



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

```
package.json > {} dependencies
  > Debug
6   "scripts": {
7     "start": "node index.js",
8     "test": "echo \"Error: no test specified\" && exit 1"
9   },
10  "author": "",
11  "license": "ISC",
12  "type": "module",
13  "dependencies": {
14    "lodash": "^4.17.21"
15  }
16 }
```

The line `"type": "module",` on line 12 is highlighted with a red rectangular box.

## npm – 외부 Library 사용하기 (Cont.)

### ■ lodash 이용하기

```
index.js  X
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> 
```

# 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

## 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

Windows 보안

공용 및 프라이빗 네트워크에서 이 앱에 액세스하도록 허용하시겠습니까?

모든 공용 네트워크와 개인 네트워크에서 Node.js JavaScript Runtime의 일부 기능이 차단되었습니다.

 **Node.js JavaScript Runtime**

게시자 Node.js

[더 많이 표시](#)

고급 보안이 포함된 Windows Defender 방화벽

파일(F) 동작(A) 보기(V) 도움말(H)

로컬 컴퓨터의 고급 보안이 포함된 Windows Defender 방화벽

인바운드 규칙

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

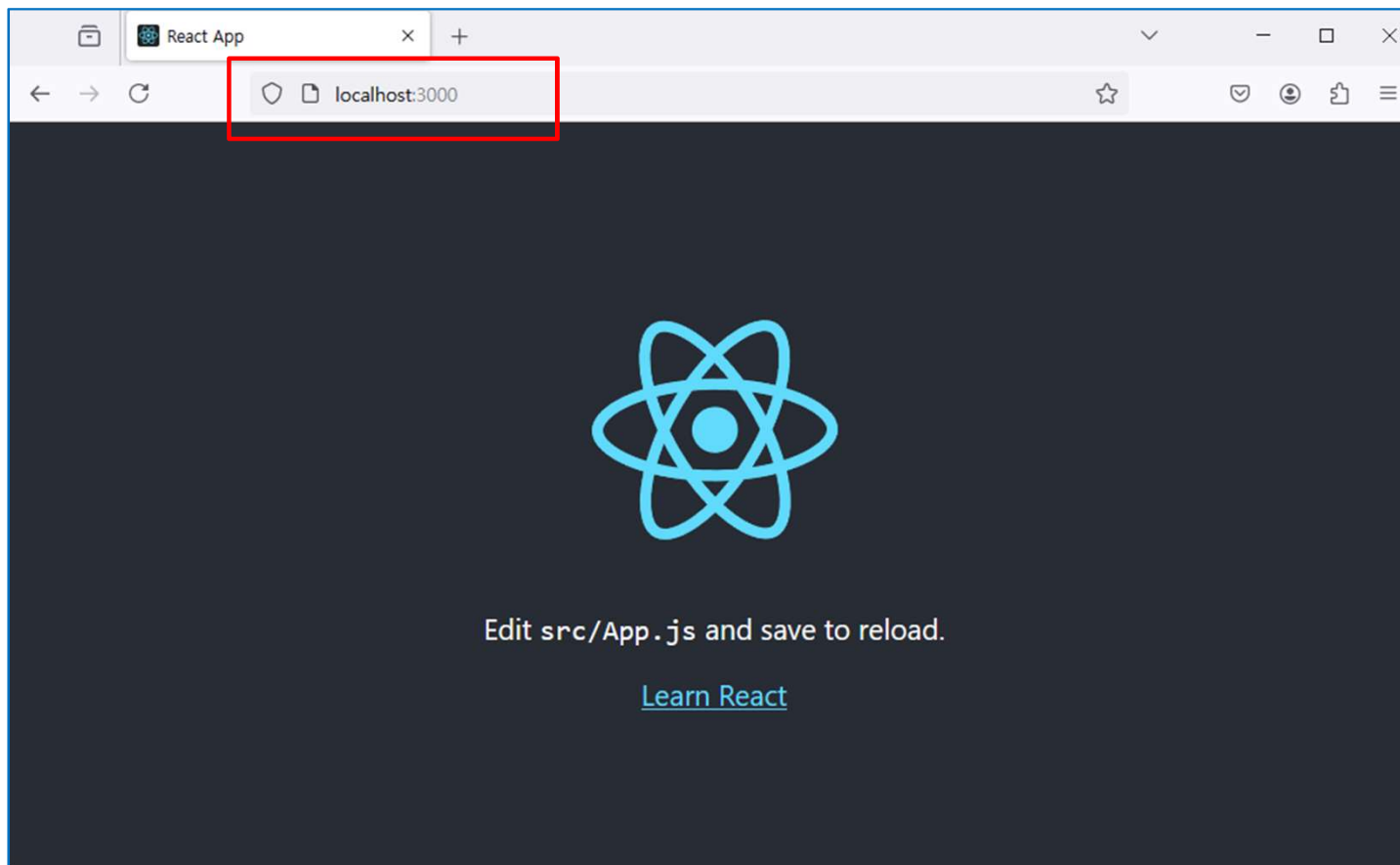
작업

인바운드 규칙

- 새 규칙...
- 프로필별 필터링
- 상태로 필터링
- 그룹으로 필터링
- 보기
- 새로 고침
- 목록 내보내기...
- 도움말

## create-react-app (Cont.)

■ `npm start`



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

The screenshot shows the JSFiddle online editor interface. At the top, there's a blue header bar with the JSFiddle logo and a 'Close' button. Below the logo, there's a section titled 'Start with a boilerplate:' with buttons for jQuery, Vue, React, React + JSX, Preact, TypeScript, CoffeeScript, SCSS, CSS Grid (responsive), Bootstrap, and PostCSS (with a 'New' button). To the right of the boilerplate section, there's a 'Links:' section with links to Roadmap (vote for features), Bug tracker, About, Docs, and Service status. Further right, there's a 'JSFiddle is for:' section with a list of features: Demos for docs, Bug reporting (test-case) for Github Issues, Presenting code answers on Stack Overflow, Live code collaboration, and Code snippets hosting. At the bottom of the blue header, there's a link to '... or just your humble code playground' with a hand icon.

Below the blue header, there's a dark grey bar with navigation buttons: Run, Save, Tidy, Collaborate, Update, and New and updated JS/CSS linters. On the right side of this bar, there are links for Settings and Sign in.

The main workspace is divided into three panels. On the left is the 'Fiddle meta' panel, which contains fields for 'Untitled fiddle' and 'No description', a note to 'Add title to make the fiddle visible on your profile page', and sections for 'Resources' (with URL and cdnjs buttons) and 'Async requests'. Below these is a section for 'Other (links, license)'. The middle panel is the 'HTML' editor, which is currently empty. The right panel is the 'CSS' editor, which is also empty. At the bottom of the workspace, there's a 'JavaScript + No-Library (pure JS)' editor, which is currently empty, and a 'Result' panel on the far right.

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

The screenshot displays the JSFiddle online editor interface. The browser address bar shows the URL `jsfiddle.net/9wuma8by/`. The interface includes a top navigation bar with icons for Run, Save, Tidy, Collaborate, and an Update button. The main workspace is divided into three panels: Fiddle meta (left), HTML (top middle), and CSS (top right). The Fiddle meta panel shows an 'Untitled fiddle' with a 'No description' and a prompt to 'Add title to make the fiddle visible on your profile page'. The HTML panel contains the following code:

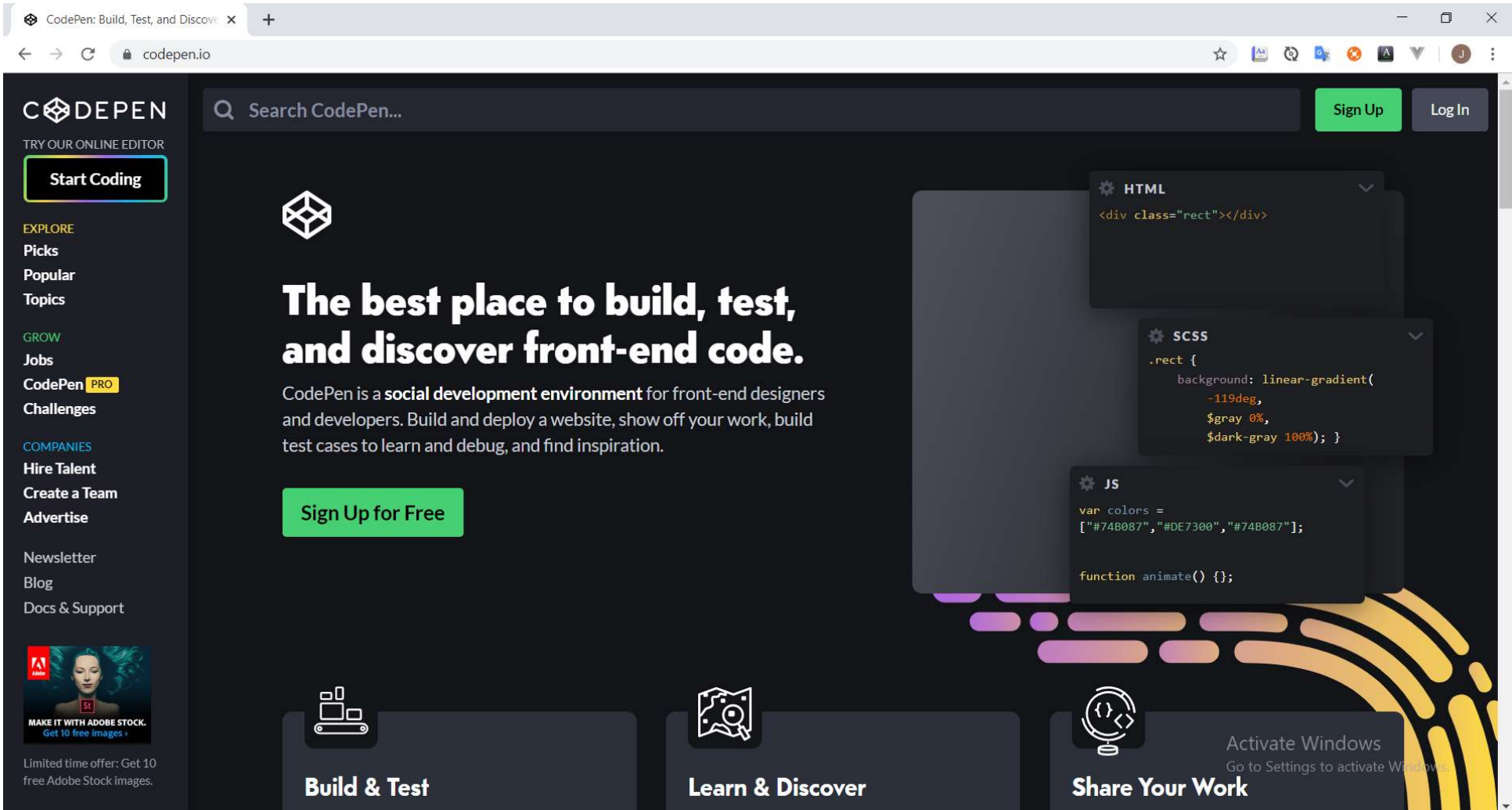
```
1 <script src="https://unpkg.com/vue"></script>
2
3 <div id="app">
4
5 </div>
```

The CSS panel is empty. The JavaScript panel (bottom middle) contains the following code:

```
1 new Vue({
2   template : '<p>{{msg}}</p>',
3   data : {
4     msg : 'Hello, World!'
5   }
6 }).$mount('#app')
```

The preview panel (bottom right) displays the output: 'Hello, World!'. The bottom of the interface features a Headway logo and the text 'Easiest way to keep your customers'.

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



The screenshot shows the CodePen website homepage. The browser's address bar displays "codepen.io". The page features a dark theme with a prominent white CodePen logo and the text "The best place to build, test, and discover front-end code." Below this, a description states: "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." A green "Sign Up for Free" button is visible. On the right, there are "Sign Up" and "Log In" buttons. The left sidebar contains navigation links under "EXPLORE", "GROW", and "COMPANIES". The main content area on the right shows a preview of a web application with three panels: HTML, SCSS, and JS. The HTML panel contains a simple div with the class "rect". The SCSS panel contains a linear gradient rule for ".rect". The JS panel contains a variable "colors" and an "animate" function. At the bottom, there are three sections: "Build & Test", "Learn & Discover", and "Share Your Work". A Windows activation watermark is visible in the bottom right corner.

CodePen: Build, Test, and Discover x +

codepen.io

CODEPEN

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 ADOBE STOCK. Get 10 free Adobe Stock Images.

Limited time offer: Get 10 free Adobe Stock Images.

Search CodePen...

Sign Up

Log In

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() {};
```

Build & Test

Learn & Discover

Share Your Work

Activate Windows  
Go to Settings to activate Windows.



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

