

## 리액트 입문

# 1. 리엑트는 어쩌다 만들어졌을까?

# 1 리엑트는 어쩌다 만들어졌을까?

## ❖ DOM 을 변형시키기 위하여 우리가 어떤 작업을 해야 할까?

- JavaScript를 사용하여 HTML 로 구성된 UI 를 제어해봤다면 익숙할 것
- 브라우저의 DOM Selector API 를 사용해서 특정 DOM 을 선택한 뒤, 특정 이벤트가 발생하면 변화를 주도록 설정

## ❖ HTML/JS 로 만들어진 카운터 예시

```
<h2 id="number">0</h2>
```

```
<div>
```

```
  <button id="increase">+1</button>
```

```
  <button id="decrease">-1</button>
```

```
</div>
```

- 위와 같이 HTML이 구성되어 있고, id 를 사용하여 각 DOM을 선택한 뒤, 원하는 이벤트가 발생하면 DOM 의 특정 속성을 바꾸어주어야 ...

# 1 리엑트는 어쩌다 만들어졌을까?

```
const number = document.getElementById('number');
```

```
const increase = document.getElementById('increase');
```

```
const decrease = document.getElementById('decrease');
```

```
increase.onclick = () => {
```

```
  const current = parseInt(number.innerText, 10);
```

```
  number.innerText = current + 1;
```

```
};
```

```
decrease.onclick = () => {
```

```
  const current = parseInt(number.innerText, 10);
```

```
  number.innerText = current - 1;
```

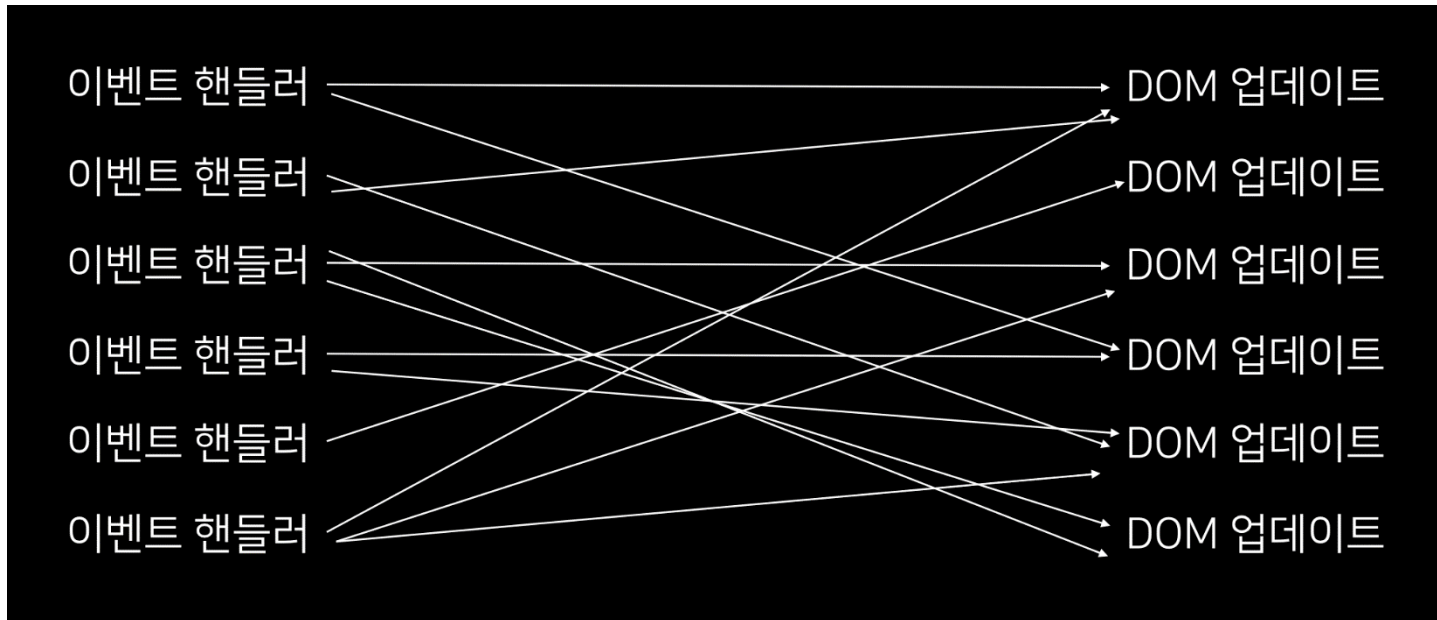
```
};
```

# 1 리엑트는 어쩌다 만들어졌을까?

- ❖ 현재 위 코드를 보면 "+1 버튼이 눌리면, id 가 number인 DOM 을 선택해서 innerText 속성을 1씩 더해줘라" 라는 규칙이 있다.
- ❖ 사용자와의 인터랙션이 별로 없는 웹페이지라면 상관없음
- ❖ 만약에 인터랙션이 자주 발생하고, 이에 따라 동적으로 UI 를 표현해야 된다면, 이러한 규칙이 정말 다양해질 것이고, 그러면 관리하기도 힘들어질 것!!!
- ❖ 숙련된 JavaScript 개발자라면, 코드를 최대한 깔끔하게 정리하여 쉽게 유지보수를 할 수도 있겠지만, 대부분의 경우 웹 애플리케이션의 규모가 커지면, DOM 을 직접 건드리면서 작업을 하면 코드가 난잡해지기 쉽다.

# 1 리액트는 어쩌다 만들어졌을까?

- ❖ 처리해야 할 이벤트도 다양해지고, 관리해야 할 상태 값도 다양해지고, DOM 도 다양해지게 된다면, 이에 따라 업데이트를 하는 규칙도 많이 복잡해지기 때문에, 조금 과장을 많이 하자면 코드가 다음과 같은 형태가 된다.

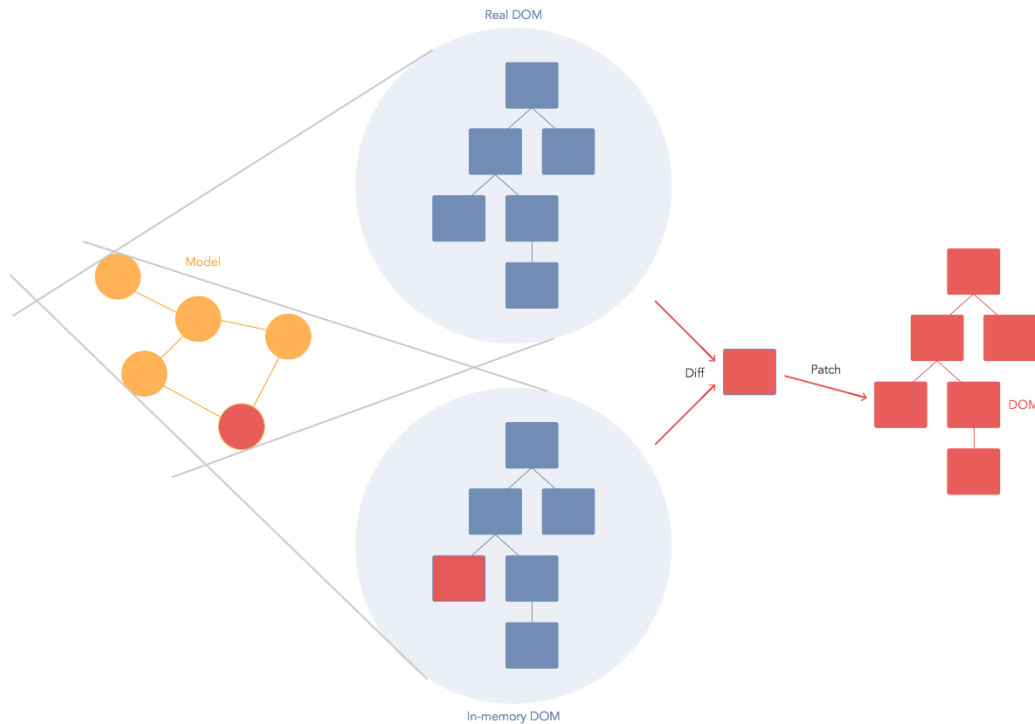


# 1 리엑트는 어쩌다 만들어졌을까?

- ❖ 그래서, Ember, Backbone, AngularJS 등의 프레임워크가 탄생!
- ❖ 이 프레임워크들은 작동방식이 각각 다르지만, 쉽게 설명하자면 자바스크립트의 특정 값이 바뀌면 특정 DOM의 속성이 바뀌도록 연결을 해주어서, 업데이트 하는 작업을 간소화해주는 방식으로 웹개발의 어려움을 해결
- ❖ 하지만 리엑트의 경우에는 조금 다른 발상에서 탄생
  - 리엑트는 어떠한 상태가 바뀌었을 때, 그 상태에 따라 DOM 을 어떻게 업데이트 할 지 규칙을 정하는 것이 아니라, **아예 다 날려버리고 처음부터 모든 걸 새로 만들어서 보여준다면 어떨까?** 라는 아이디어에서 개발이 시작
- ❖ 그러면 "업데이트를 어떻게 해야 할 지" 에 대한 고민을 전혀 안해도 되기 때문에 개발이 정말 쉬워질 것!!
- ❖ 하지만, 정말로 동적인 UI 를 보여주기 위해서 모든 걸 다 날려버리고 모든 걸 새로 만들게 된다면, 속도가 굉장히 느릴 것
- ❖ 작은 웹 애플리케이션이라면 상관없겠지만 규모가 큰 웹 애플리케이션이라면 문제!!

# 1 리엑트는 어쩌다 만들어졌을까?

❖ 하지만, 리엑트에서는 **Virtual DOM** 이라는 것을 사용해서 이를 해결!!





# 1 리엑트는 어쩌다 만들어졌을까?

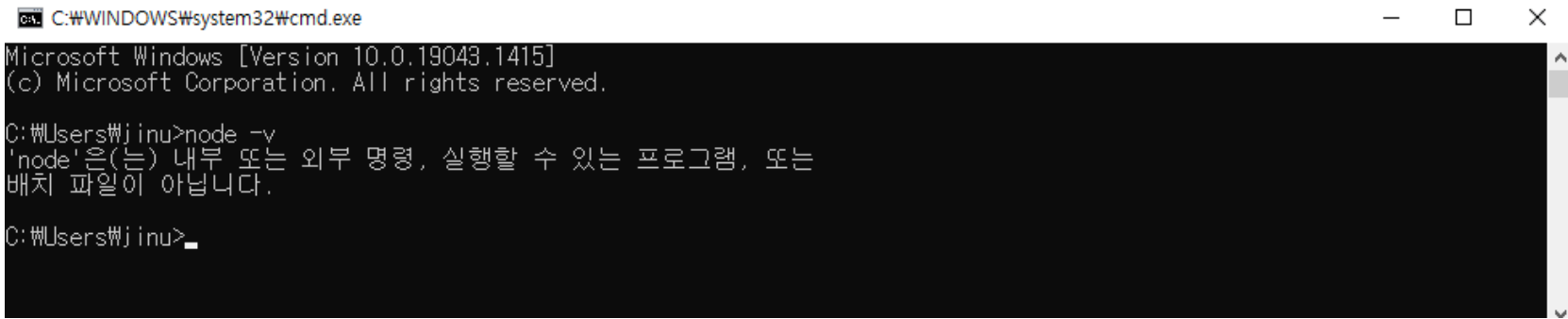
- ❖ Virtual DOM 은 브라우저에 실제로 보여지는 DOM 이 아니라 그냥 메모리에 가상으로 존재하는 DOM 으로서 그냥 JavaScript 객체이기 때문에 작동 성능이 실제로 브라우저에서 DOM 을 보여주는 것 보다 속도가 훨씬 빠르다.
- ❖ 리엑트는 상태가 업데이트 되면, 업데이트가 필요한 곳의 UI 를 Virtual DOM 을 통해서 렌더링!
- ❖ 그리고 나서 리엑트 개발팀이 만든 매우 효율적인 비교 알고리즘을 통하여 실제 브라우저에 보여지고 있는 DOM 과 비교를 한 후, 차이가 있는 곳을 감지하여 이를 실제 DOM 에 패치.
- ❖ 이를 통하여, "업데이트를 어떻게 할 지" 에 대한 고민을 하지 않으면서, 빠른 성능도 유지.

## 2. 작업환경 준비

# 1 작업환경 준비

## ❖ Node.js 설치 확인

- 윈도우 명령 프롬프트(macOS에서는 터미널)를 실행하고 다음 명령어를 입력해 Node.js가 설치되어 있는지 확인



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jinu>node -v
'node'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는
배치 파일이 아닙니다.

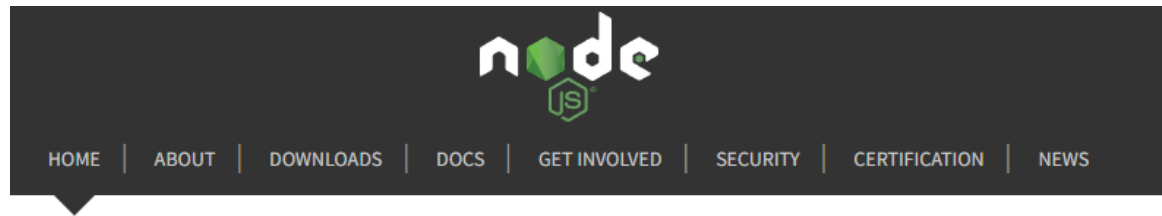
C:\Users\jinu>_
```

# 1 작업환경 준비

## ❖ Node.js 설치

- Node.js 공식 사이트에 접속한 다음 <LTS 버전>이 적힌 버튼을 눌러 Node.js 설치 파일을 내려받아 설치.
  - 설치 과정은 기본 설정을 그대로 두고 <Next>버튼을 눌러 진행하면 되므로 생략

Node.js 공식 사이트: [nodejs.org](https://nodejs.org)



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for Windows (x64)



[Other Downloads](#) | [Changelog](#) | [API Docs](#)   [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#)

# 1 작업환경 준비

## ❖ Node.js 설치 다시 확인

- Node.js를 설치했으니 오류 메시지가 출력된 명령 프롬프트를 종료
- 그런 다음 명령 프롬프트를 다시 열어 다음 명령어를 입력

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jinu>node -v
v16.13.1


C:\Users\jinu>_
```

- Node.js를 설치하면 노드 패키지 매니저(npm)도 함께 설치.
- 앞으로 노드 패키지 매니저(node package manager)는 줄여서 npm이라고 명명

# 1 작업환경 준비

## ❖ npm 설치 확인하기

- 다음 명령어를 입력해 npm 버전을 확인

 C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.19043.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jinu>npm -v
8.1.2

C:\Users\jinu>
```

# 1 작업환경 준비

## ❖ npx 설치

- npx는 1회성으로 최신 버전의 노드 패키지를 내려받아 설치시켜주는 노드 패키지
- 그래서 npx는 npm으로 설치해야...
- 다음 명령어를 입력하여 npx 패키지를 설치하고 확인

```
CA C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jinu>npm install npx -g

added 3 packages, and audited 362 packages in 7s

37 vulnerabilities (1 low, 18 moderate, 11 high, 7 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 8.1.2 -> 8.3.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.3.0
npm notice Run npm install -g npm@8.3.0 to update!
npm notice

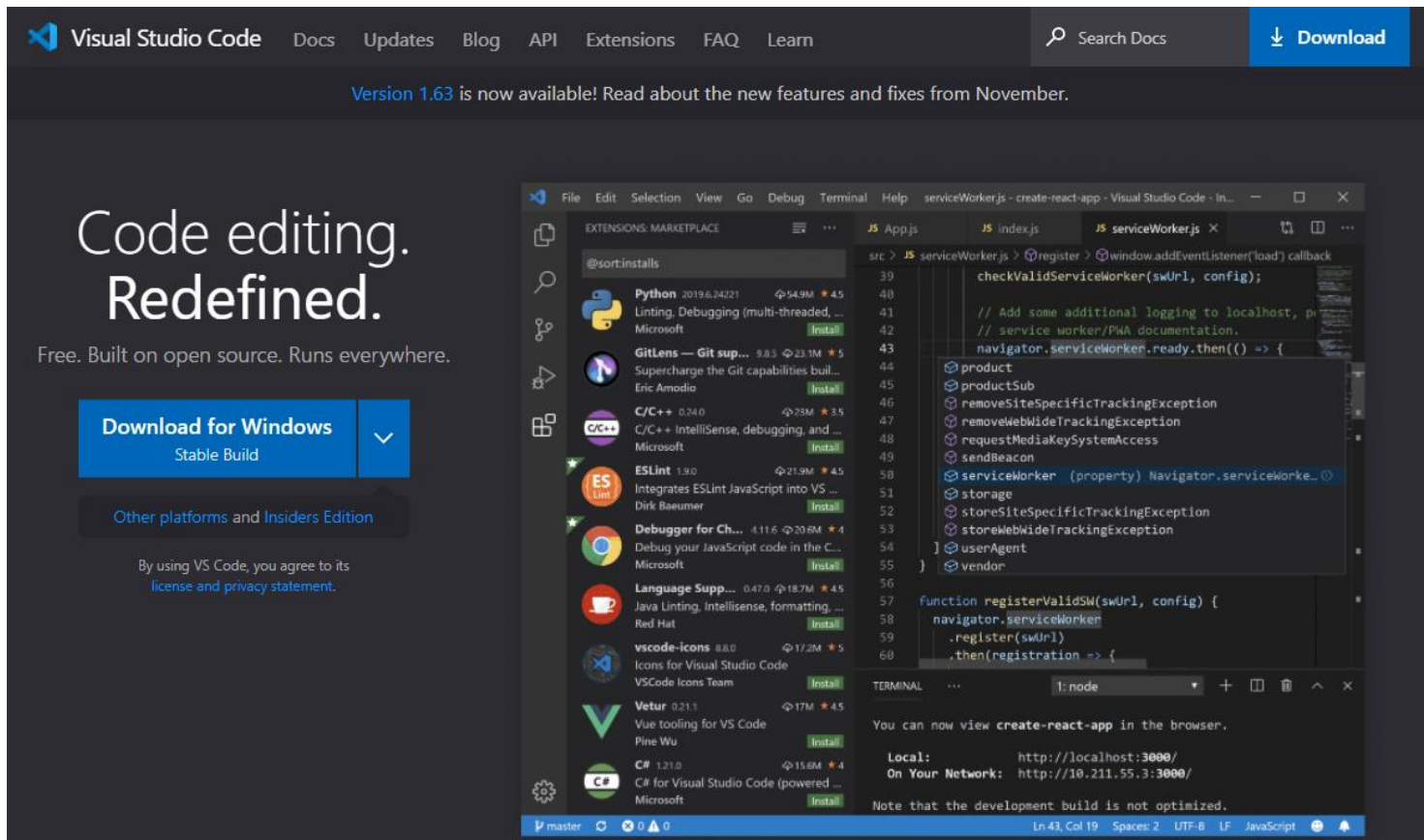
C:\Users\jinu>npx -v
8.1.2

C:\Users\jinu>_
```

# 1 작업환경 준비

## ❖ VSCode 설치

- VSCode 공식 사이트에서 설치 파일을 내려받아 설치  
**code.visualstudio.com**



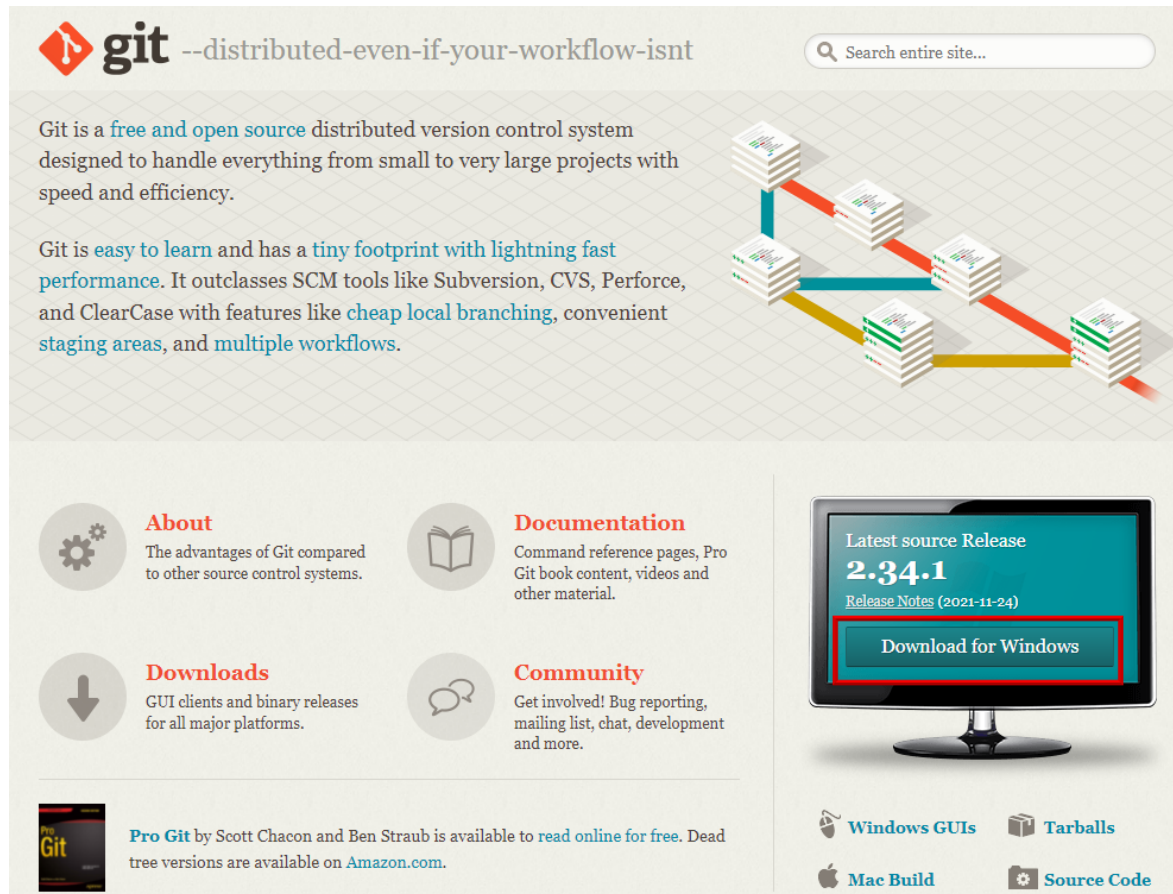


# 1 작업환경 준비

## ❖ git 설치

- 공식 사이트에 접속해 깃 설치 파일을 내려받아 설치.

**git-scm.com**



The screenshot shows the Git website homepage. At the top, the Git logo is followed by the tagline "--distributed-even-if-your-workflow-isnt". A search bar is on the right. The main content area describes Git as a free and open source distributed version control system. Below this, it mentions Git is easy to learn and has a tiny footprint with lightning fast performance. To the right of the text is a diagram showing a branching model with multiple stacks of code blocks connected by lines. The bottom section features four icons with corresponding text: 'About' (gears), 'Documentation' (book), 'Downloads' (down arrow), and 'Community' (speech bubbles). On the right, a monitor displays the 'Latest source Release 2.34.1' with a red box around the 'Download for Windows' button. At the bottom, there are links for 'Windows GUIs', 'Tarballs', 'Mac Build', and 'Source Code'.

**git** --distributed-even-if-your-workflow-isnt

Search entire site...

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.

Latest source Release  
**2.34.1**  
[Release Notes \(2021-11-24\)](#)  
**Download for Windows**

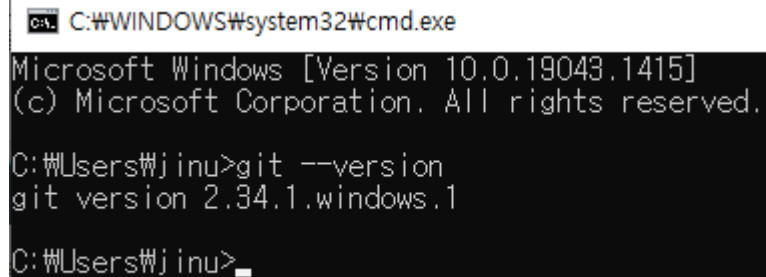
**Pro Git** by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

[Windows GUIs](#) [Tarballs](#)  
[Mac Build](#) [Source Code](#)

# 1 작업환경 준비

## ❖ git 설치확인

- 설치를 완료한 후 깃 설치 여부를 확인



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jinu>git --version
git version 2.34.1.windows.1

C:\Users\jinu>
```

# 1 작업환경 준비

## ❖ 새 프로젝트 생성

- 명령 프롬프트를 실행한 다음 리액트 앱을 만들고 싶은 곳으로 이동
- 다음 명령어를 입력

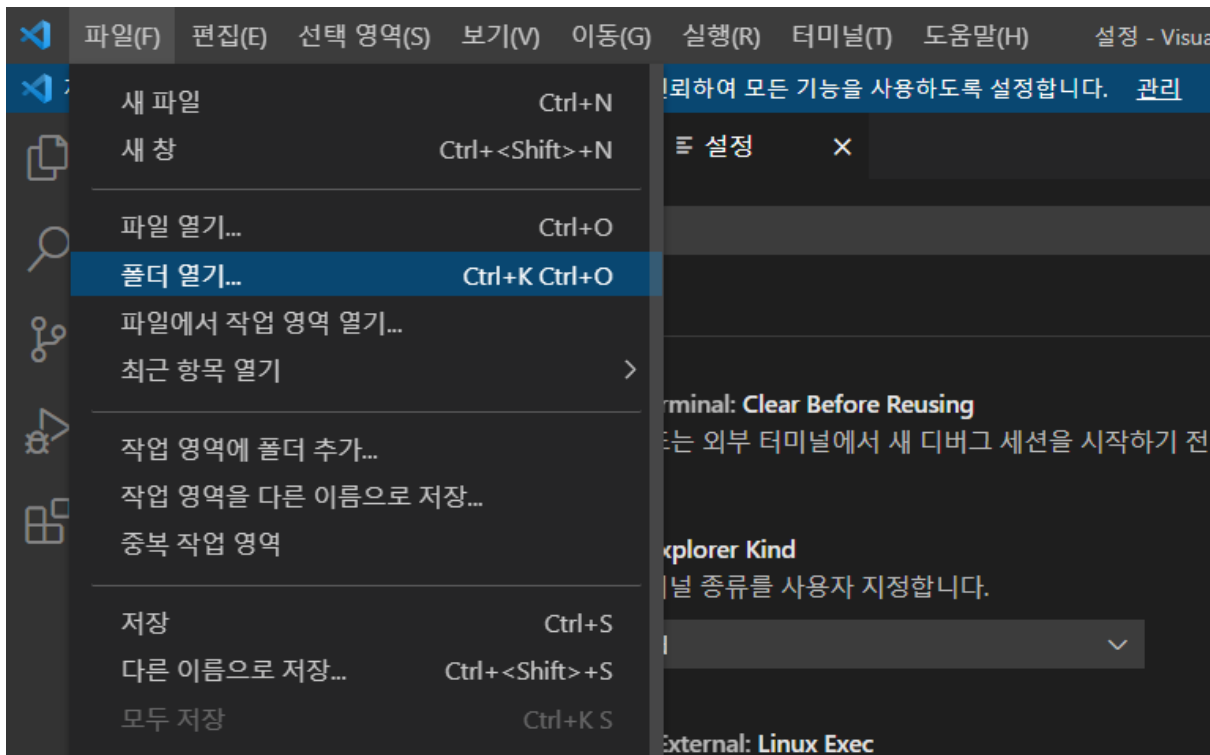
**\$ npx create-react-app movie\_app\_2021**

- 그러면 **movie\_app\_2021** 라는 디렉터리가 생기고 그 안에 리액트 프로젝트가 생성.

# 1 작업환경 준비

## ❖ 프로젝트 폴더 만들기

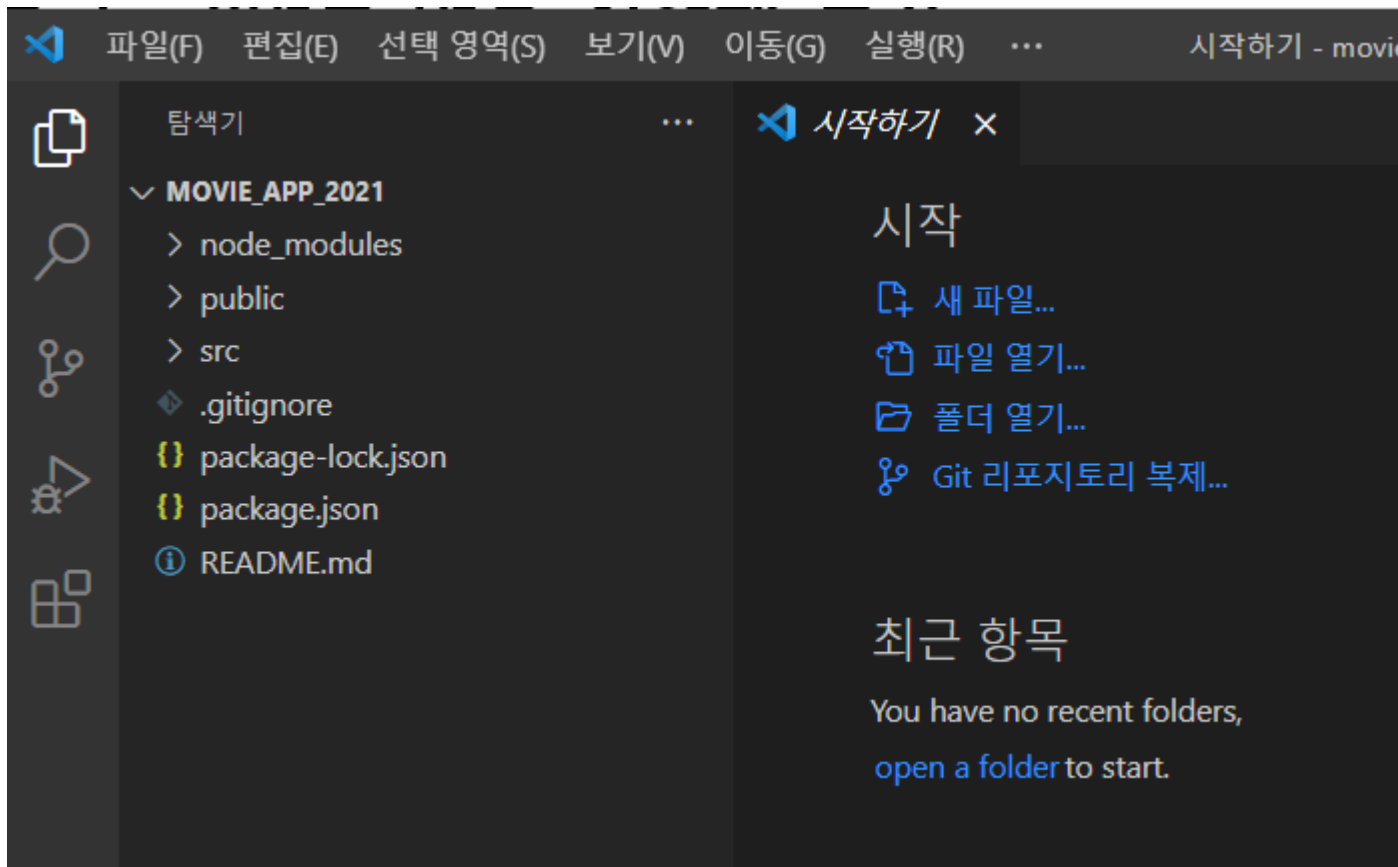
- VS Code를 실행하여 [파일 — 폴더 열기]를 누른 다음 C:\movie\_app\_2021 폴더를 선택



# 1 작업환경 준비

## ❖ 프로젝트 폴더 만들기

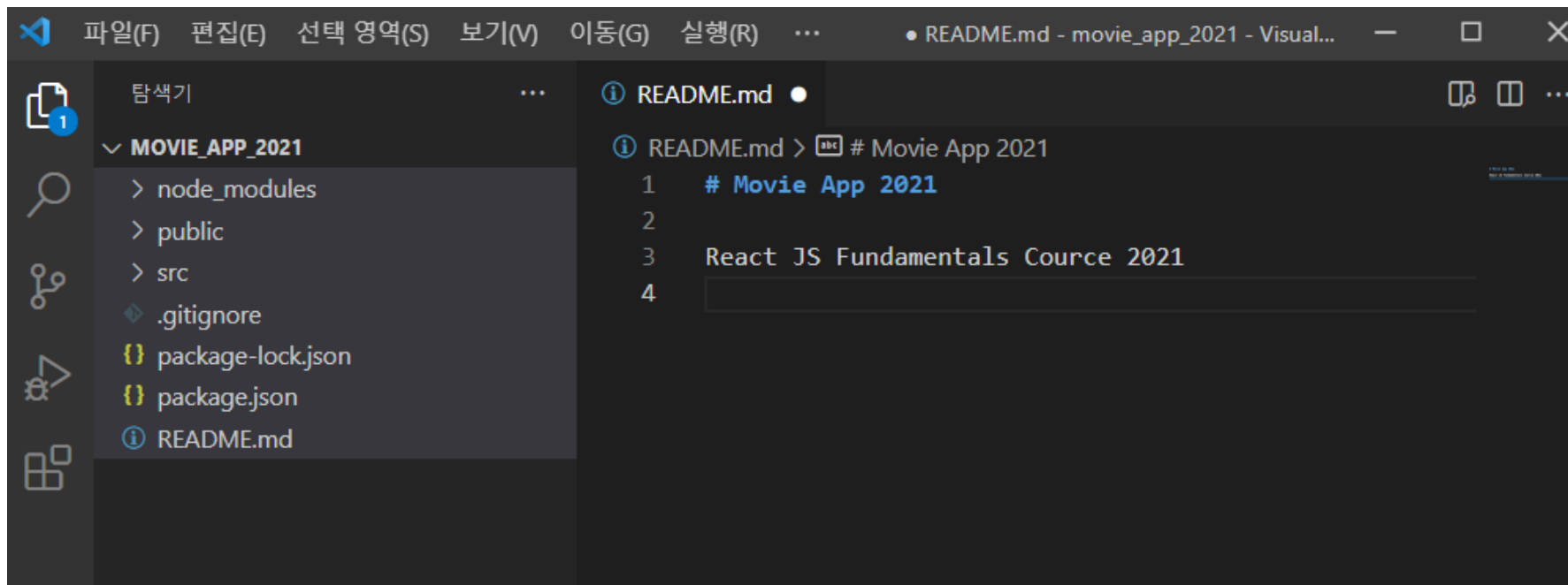
- 폴더를 선택하면 왼쪽에 프로젝트 구조를 볼 수 있는 창이 출력.
- 몇몇 파일을 수정해 보고, 리액트 앱을 실행해 보자



# 1 작업환경 준비

## ❖ README.md 파일 수정하기

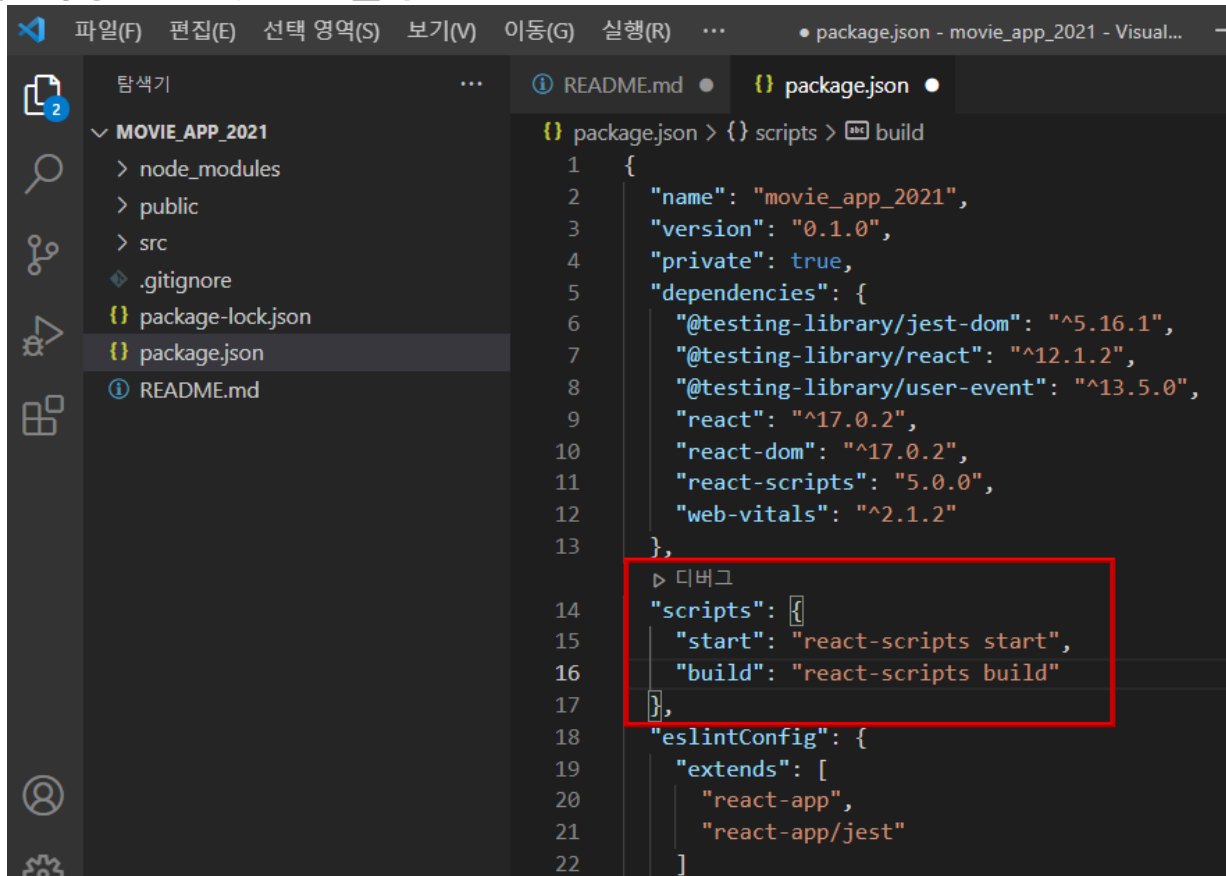
- 루트 폴더에 있는 README.md 파일을 연 다음, 그 안에 작성되어 있던 내용을 모두 삭제
- 그 후, 다음과 같이 입력하고 저장



# 1 작업환경 준비

## ❖ package.json 파일 수정하기

- 이어서 package.json 파일을 열어서 scripts 키 값을 수정.
- test, eject 명령어는 사용하지 않을 테니까 삭제하자.
- 파일 저장하는 것은 필수!



```
package.json > {} scripts > build
1  {
2    "name": "movie_app_2021",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^5.16.1",
7      "@testing-library/react": "^12.1.2",
8      "@testing-library/user-event": "^13.5.0",
9      "react": "^17.0.2",
10     "react-dom": "^17.0.2",
11     "react-scripts": "5.0.0",
12     "web-vitals": "^2.1.2"
13   },
14   "scripts": {
15     "start": "react-scripts start",
16     "build": "react-scripts build"
17   },
18   "eslintConfig": {
19     "extends": [
20       "react-app",
21       "react-app/jest"
22     ]
23   }
24 }
```

# 1 작업환경 준비

## ❖ 리액트 앱 실행하기

- 명령 프롬프트에서 `cd` 명령어를 사용하여 해당 디렉터리에 들어간 다음 `npm start` 명령어를 입력

```
$ cd movie_app_2021
```

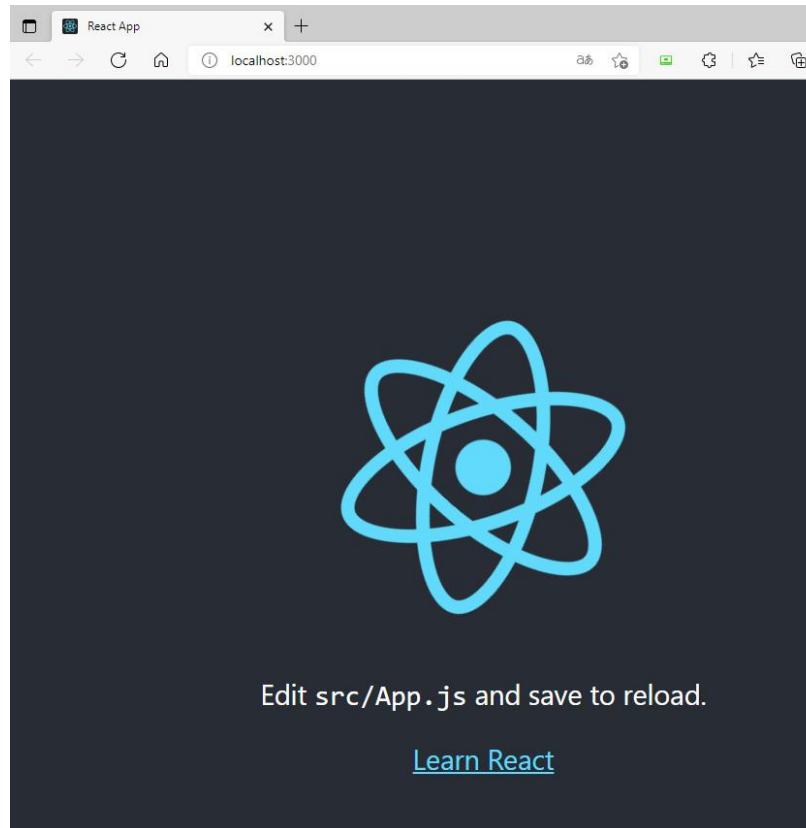
```
$ npm start
```



# 1 작업환경 준비

## ❖ 리액트 앱 실행하기

- 이 명령어를 실행하고 나면 다음과 같이 브라우저에 `http://localhost:3000/` 이 열리고, 돌아가는 리액트 아이콘이 보일 것
- 자동으로 페이지가 열리지 않는다면 브라우저에 주소를 직접 입력



# 1 작업환경 준비

## ❖ 리액트 앱 종료하기

- 명령 프롬프트로 돌아가서 [Ctrl + C] 를 누르면 리액트 앱이 종료

Windows PowerShell

Note that the development build is not optimized.  
To create a production build, use `npm run build`.

assets by path `static/` 1.48 MiB

asset `static/js/bundle.js` 1.47 MiB [emitted] (name: main) 1 related asset

asset `static/js/node_modules_web-vitals_dist_web-vitals_js.chunk.js` 6.89 KiB [emitted] 1 related asset

asset `static/media/logo.6ce24c58023cc2f8fd88fe9d219db6c6.svg` 2.57 KiB [emitted] (auxiliary name: main)

asset `index.html` 1.67 KiB [emitted]

asset `asset-manifest.json` 546 bytes [emitted]

runtime modules 31.3 KiB 15 modules

modules by path `./node_modules/` 1.35 MiB 89 modules

modules by path `./src/` 18 KiB

modules by path `./src/*.css` 8.82 KiB

`./src/index.css` 2.72 KiB [built] [code generated]

`./node_modules/css-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[5].use[1]!./node_modules/postcss-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[5].use[2]!./node_modules/source-map-loader/dist/cjs.js!./src/index.css` 1.37 KiB [built] [code generated]

`./src/App.css` 2.72 KiB [built] [code generated]

`./node_modules/css-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[5].use[1]!./node_modules/postcss-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[5].use[2]!./node_modules/source-map-loader/dist/cjs.js!./src/App.css` 2 KiB [built] [code generated]

modules by path `./src/*.js` 5.59 KiB

`./src/index.js` 1.76 KiB [built] [code generated]

`./src/App.js` 2.46 KiB [built] [code generated]

`./src/reportWebVitals.js` 1.37 KiB [built] [code generated]

`./src/logo.svg` 3.61 KiB [built] [code generated]

webpack 5.65.0 compiled **successfully** in 4563 ms

일괄 작업을 끝내시겠습니까 (Y/N)? y

## 2 깃허브에 리액트 앱 업로드하기

### ❖ 로컬 저장소 초기화하기

>git init

```
C:\> 명령 프롬프트
Microsoft Windows [Version 10.0.19043.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jinu>cd #

C:\>cd movie_app_2021

C:\movie_app_2021>git init
Initialized empty Git repository in C:/movie_app_2021/.git/

C:\movie_app_2021>
```

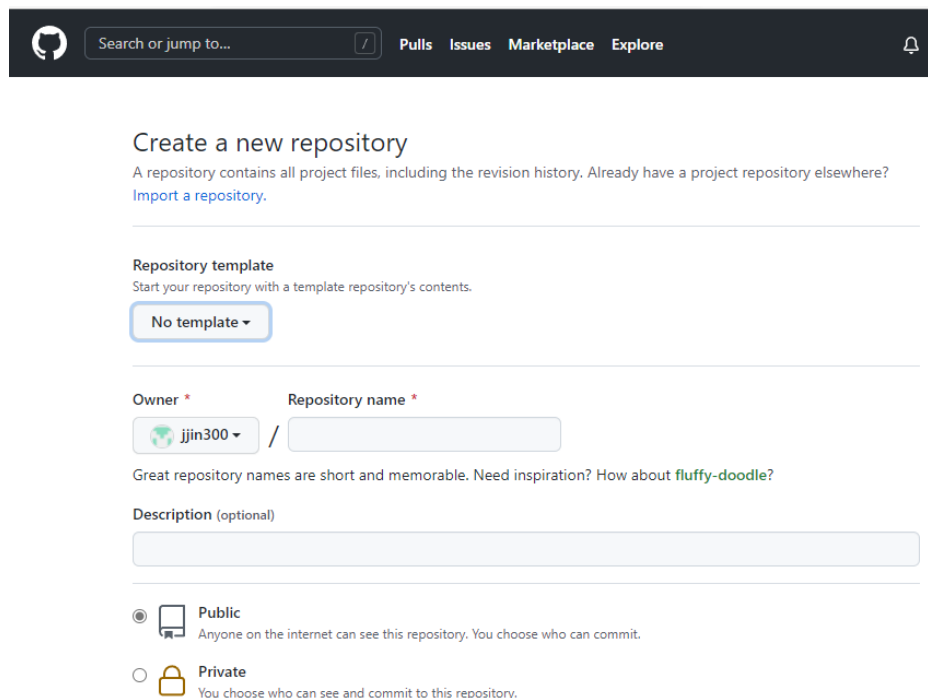
- **git init** 명령어를 실행하면 현재 폴더에 저장소(repository)를 새롭게 초기화

## 2 깃허브에 리액트 앱 업로드하기

### ❖ 깃허브에 저장소 만들기

- 깃허브 저장소 만들기 페이지에 접속.
- 만약 깃허브에 가입하지 않았다면 가입해야 함.
- 가입 및 로그인 후 나오는 화면에서 <start a project>를 눌러도 저장소 만들기 페이지로 갈 수 있음.

[www.github.com/new](https://www.github.com/new)



The screenshot shows the GitHub 'Create a new repository' page. At the top is a dark navigation bar with the GitHub logo, a search bar, and links for Pulls, Issues, Marketplace, and Explore. Below the navigation bar, the main heading is 'Create a new repository'. A subtext explains that a repository contains all project files and provides a link to 'Import a repository'. The 'Repository template' section offers a dropdown menu currently set to 'No template'. The 'Owner' field is populated with 'jjin300'. The 'Repository name' field is empty. A note suggests great repository names are short and memorable, with an example 'fluffy-doodle?'. The 'Description (optional)' field is empty. At the bottom, there are two radio button options: 'Public' (selected) and 'Private'. The 'Public' option includes a lock icon and states 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option includes an open lock icon and states 'You choose who can see and commit to this repository.'

## 2 깃허브에 리액트 앱 업로드하기

### ❖ 깃허브에 저장소 만들기

- Repository name에 'movie\_app\_2021'을 입력.
- Description에는 'React JS Fundamentals Course 2021'이라고 입력.
- Public / Private 옵션에서는 [Public]을 선택.
- 나머지는 기본설정 그대로.
- 그런 다음 <Create repository>를 누르면 저장소가 생성

The screenshot shows the GitHub 'Create repository' form. The following elements are highlighted with red boxes:

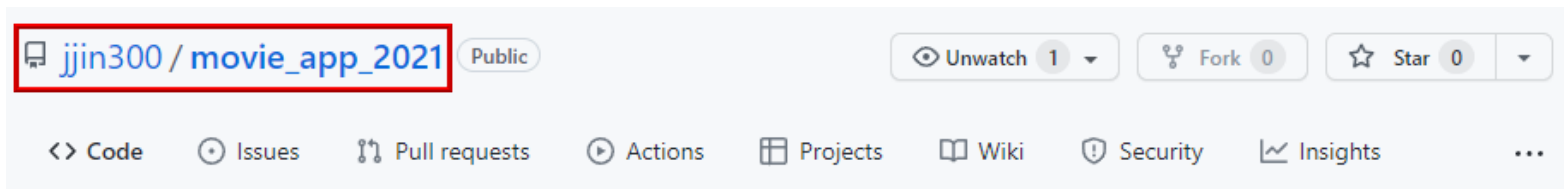
- Repository name:** A text input field containing 'movie\_app\_2021' with a green checkmark icon to its right.
- Description:** A text input field containing 'React JS Fundamentals Course 2021'.
- Visibility:** The 'Public' radio button is selected, with the text 'Anyone on the internet can see this repository. You choose who can commit.'
- Create repository:** A green button at the bottom of the form.

Other visible text includes: 'Owner \* jjin300', 'Repository name \*', 'Great repository names are short and memorable. Need inspiration? How about fluffy-doodle?', 'Description (optional)', 'Initialize this repository with: Skip this step if you're importing an existing repository.', 'Add a README file', 'Add .gitignore', and 'Choose a license'.


## 2 깃허브에 리액트 앱 업로드하기

### ❖ 깃허브에 저장소 만들기

- 그러면 다음과 같은 화면과 함께 깃허브 계정에 **movie\_app\_2021**이라는 이름의 깃허브 저장소가 생성



#### Quick setup — if you've done this kind of thing before

 Set up in Desktop

or

HTTPS

SSH

`https://github.com/jjin300/movie_app_2021.git`



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

#### ...or create a new repository on the command line

```
echo "# movie_app_2021" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/jjin300/movie_app_2021.git
git push -u origin main
```



## 2 깃허브에 리액트 앱 업로드하기

### ❖ 깃허브 저장소에 리액트 앱 업로드하기

- 깃허브 저장소의 URL을 사용해 터미널에 다음과 같이 명령어를 입력.
- 이때 명령어를 손으로 직접 입력하지 말고 저장소에 있는 명령어 예를 복사해서 사용하도록!

**git remote add origin github.com/[계정이름]/movie\_app\_2021.git**

...or create a new repository on the command line

```
echo "# movie_app_2021" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/jjin300/movie_app_2021.git
git push -u origin main
```

## 2 깃허브에 리액트 앱 업로드하기

### ❖ 깃허브 저장소에 리액트 앱 업로드하기

- 다음 명령어를 입력하고 조금 기다리면 깃허브 저장소에 리액트 앱이 업로드!
- `git add .`
- `git commit -m "02-2 깃허브에 리액트 앱 업로드하기 "`
- `git push origin master`



## 2 깃허브에 리액트 앱 업로드하기

### ❖ 깃허브 저장소 확인하기

- 업로드가 완료되면 깃허브 저장소 URL에 접속!

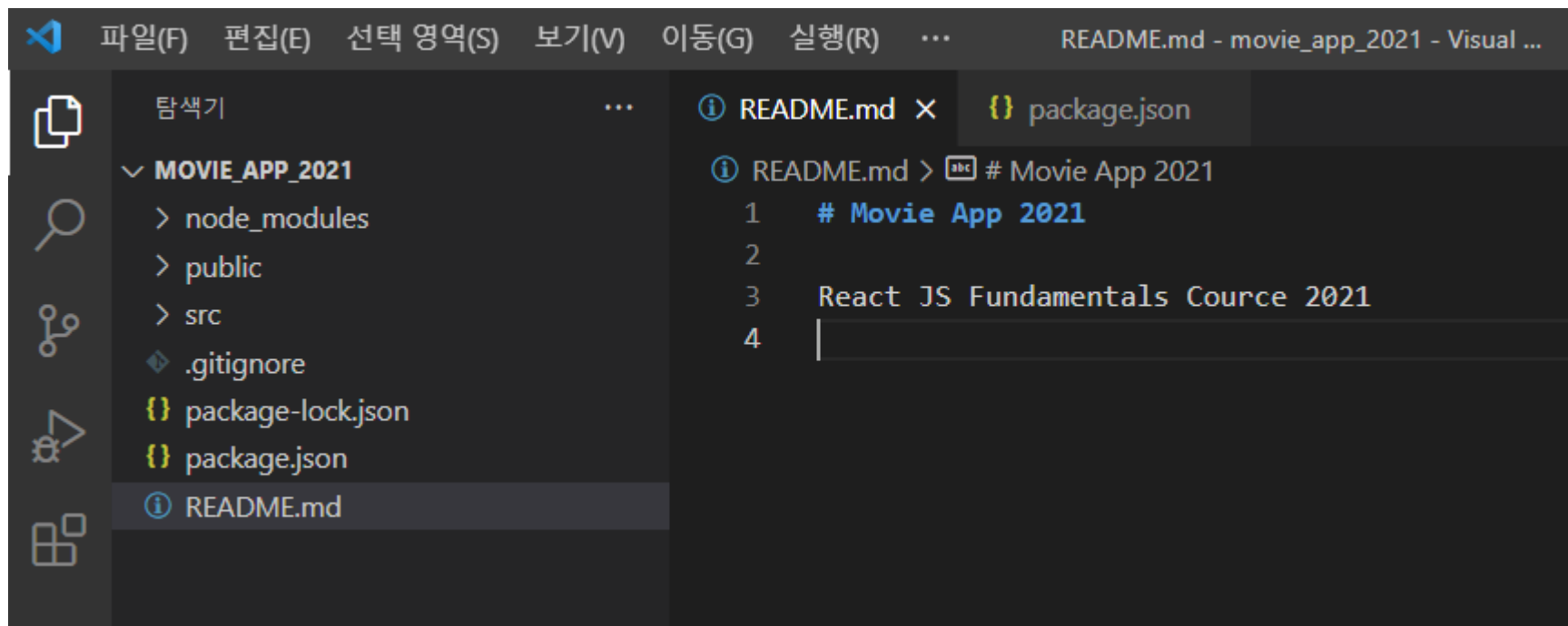
The screenshot shows a GitHub repository page for 'jjin300 / movie\_app\_2021'. The repository is public and has 1 unwatch, 0 forks, and 0 stars. The main content area displays a list of files and folders uploaded 1 minute ago. The files include 'public', 'src', '.gitignore', 'README.md', 'package-lock.json', and 'package.json'. The right sidebar shows the repository's 'About' section, which includes the title 'React JS Fundamentals Course 2021', a 'Readme' link, 0 stars, 1 watching, and 0 forks. The 'Releases' section indicates that no releases have been published and provides a link to 'Create a new release'.

File/Folder	Upload Time
public	02-2 깃허브에 리액트 앱 업로드하기 1 minute ago
src	02-2 깃허브에 리액트 앱 업로드하기 1 minute ago
.gitignore	02-2 깃허브에 리액트 앱 업로드하기 1 minute ago
README.md	02-2 깃허브에 리액트 앱 업로드하기 1 minute ago
package-lock.json	02-2 깃허브에 리액트 앱 업로드하기 1 minute ago
package.json	02-2 깃허브에 리액트 앱 업로드하기 1 minute ago

# 3 리액트 앱살펴보고 수정하기

## ❖ public 폴더 살펴보기

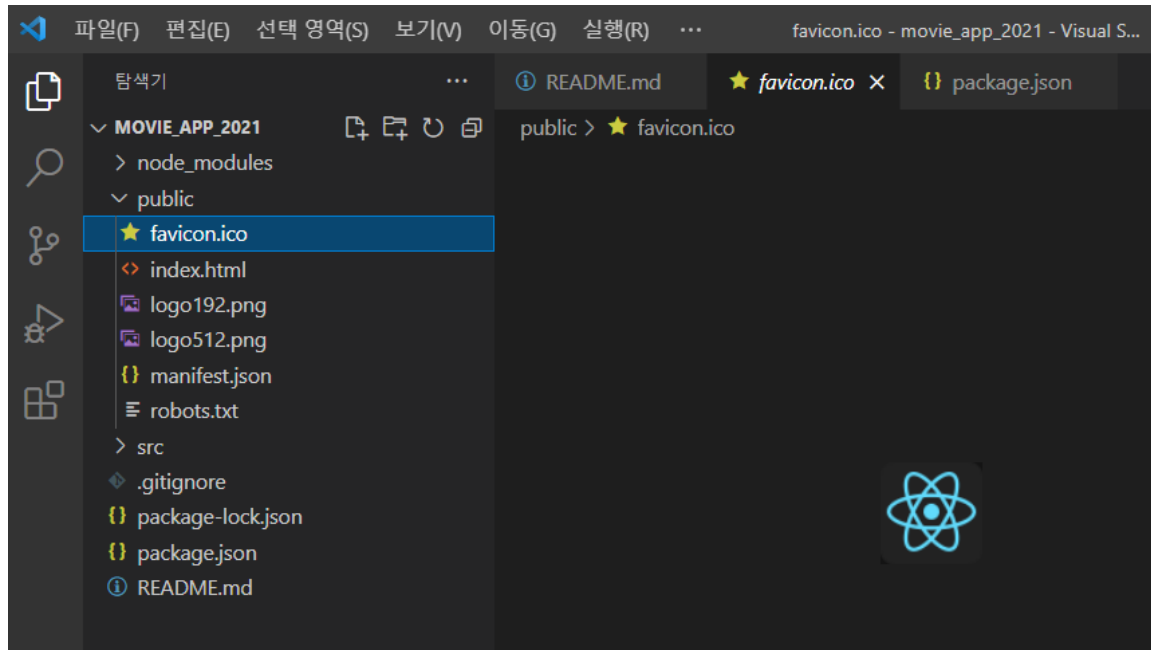
- VSCode의 왼쪽 화면을 이용해서 `movie_app_2021` 폴더를 열어보면, `node_modules`, `public`, `src` 폴더가 보임.
- `node_module` 폴더는 앞으로 수정하지 않을 예정.
- `Public`, `src` 폴더는 앞으로 자주 수정.



### 3 리액트 앱살펴보고 수정하기

#### ❖ 리액트 앱 프로젝트 폴더 살펴보기

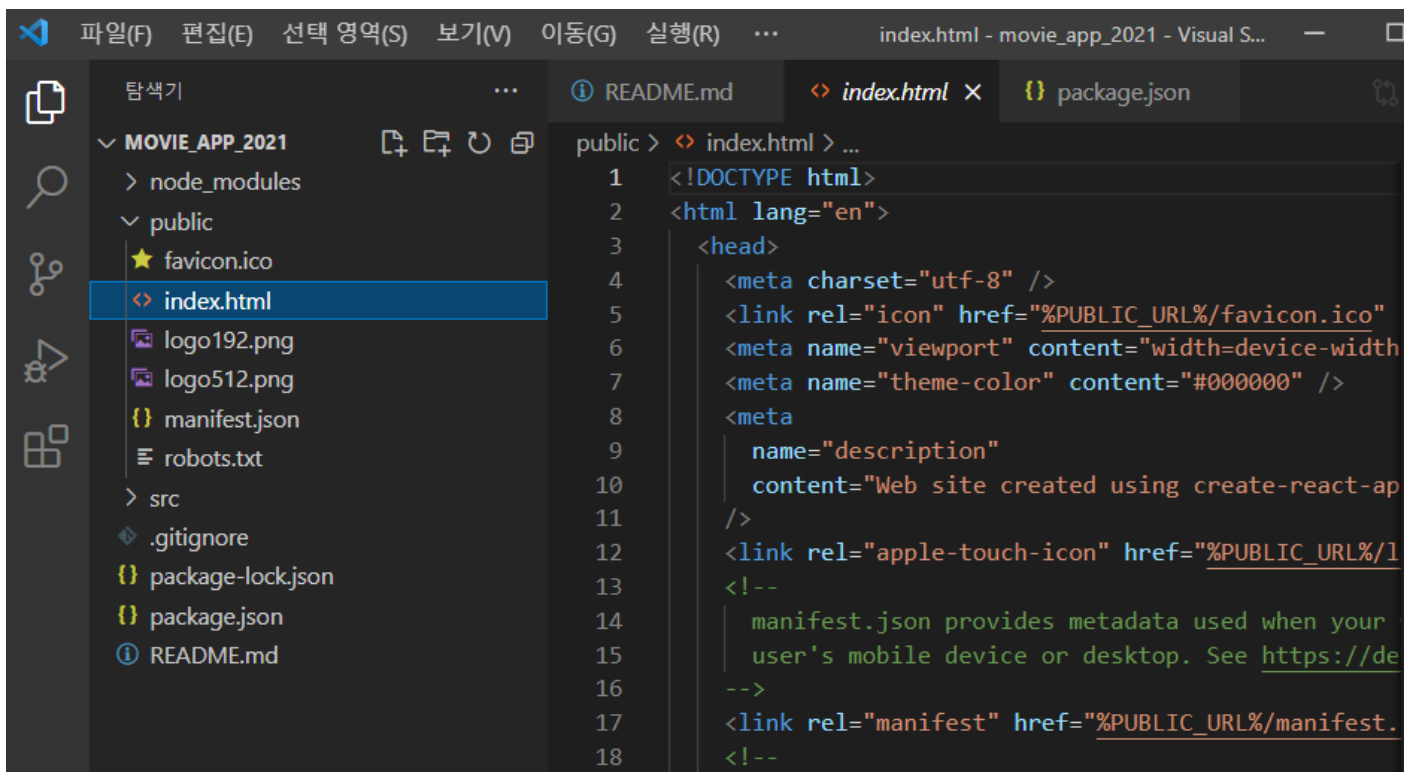
- **public** 폴더에는 **favicon.ico** 파일이 있는데, 이걸 브라우저 제목과 함께 표시되는 아이콘.



### 3 리액트 앱살펴보고 수정하기

#### ❖ index.html 파일 살펴보기

- index.html 파일에는 기본 내용만 작성.
- 여기는 앞으로 거의 사용하지 않음.
- public 폴더에 있는 나머지 파일(manifest.json, logo.png, ...)은 신경 쓰지 않아도 문제없음.

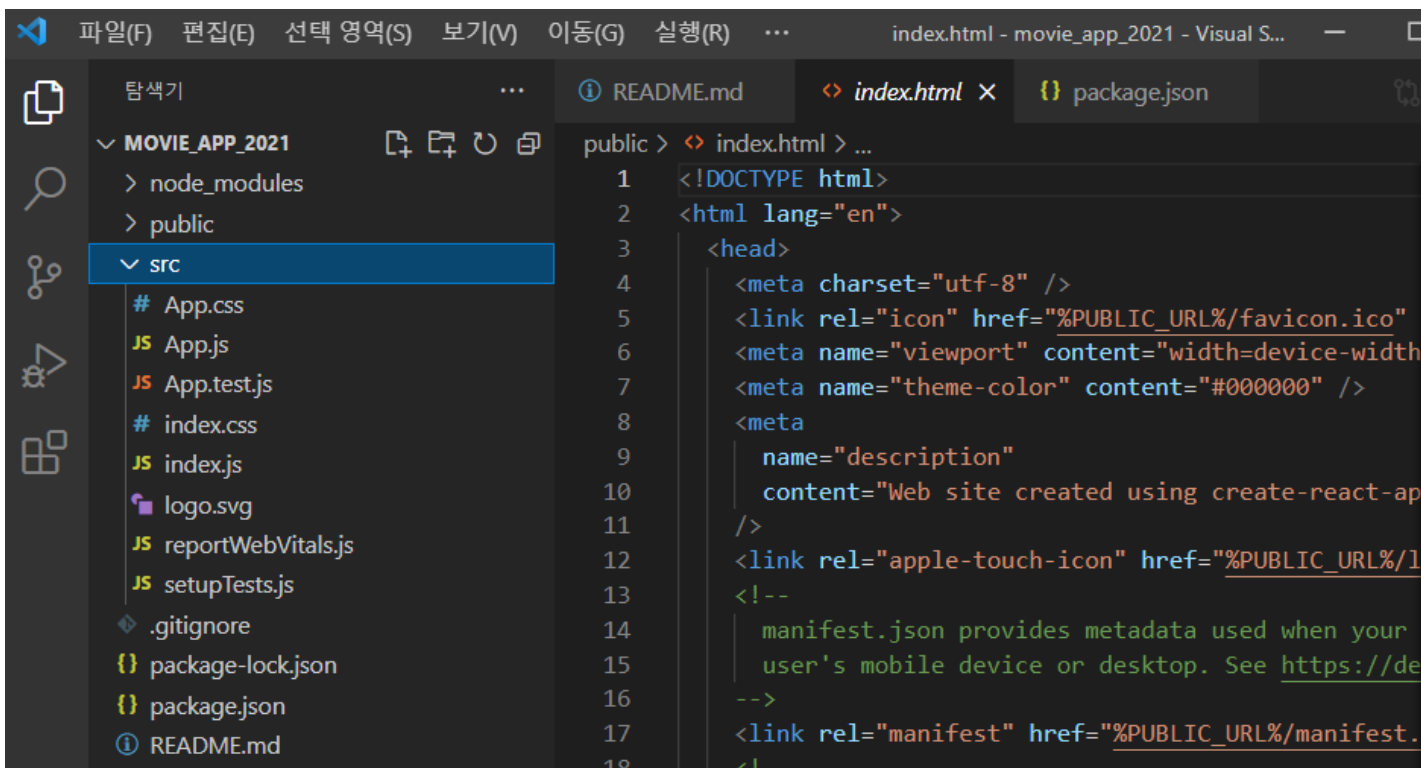


```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <meta name="theme-color" content="#000000" />
8     <meta name="description" content="Web site created using create-react-app" />
9   </head>
10  <body>
11    <noscript>
12      <div>
13        <strong>Your browser does not support JavaScript.</strong>
14        <p>manifest.json provides metadata used when your user's mobile device or desktop. See https://developer.mozilla.org/en-US/docs/Web/Manifest</p>
15      </div>
16    </noscript>
17    <div id="root"></div>
18  </body>
```

# 3 리액트 앱살펴보고 수정하기

## ❖ src 폴더 살펴보기

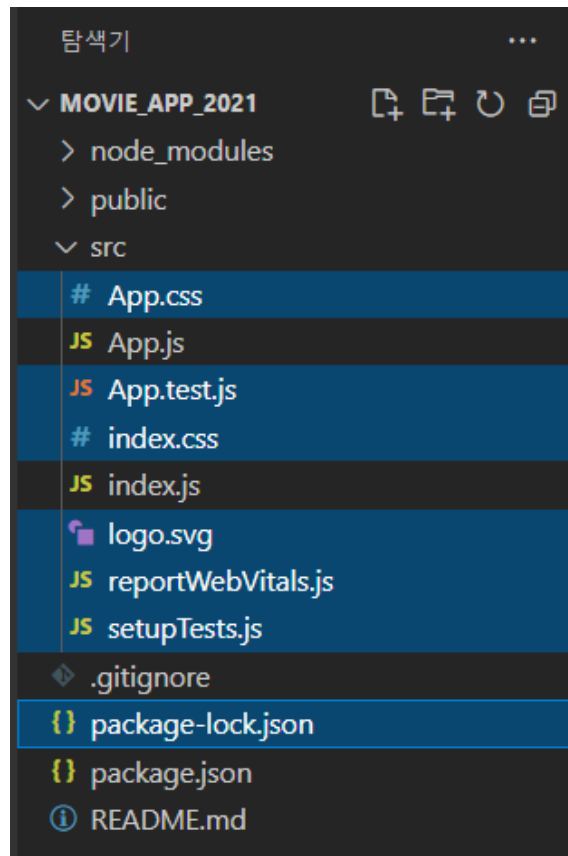
- src 폴더에는 `setupTests.js`, `serviceWorker.js`, `logo.svg`, `index.js`, `index.css`, `App.test.js`, `App.js`, `App.css` 등 많은 파일이 들어 있지만, 우리는 리액트를 기초부터 공부하기 때문에 필요 없는 파일은 삭제.



# 3 리액트 앱살펴보고 수정하기

## ❖ src 폴더 정리하기

- src 폴더에 있는 **App.css**, **App.test.js**, **index.css**, **logo.svg**, **reportWebVitals.js**, **setupTests.js**를 지우고 루트 폴더에 있는 **package-lock.json**을 삭제



# 3 리액트 앱살펴보고 수정하기

## ❖ index.js 파일 수정하기

- src 폴더의 index.js 파일을 열고 다음과 같이 표시한 코드를 삭제.
- 앞으로 필요한 코드가 아니니까 걱정하지 말고 과감히 삭제

1	import React from 'react';	
2	import ReactDOM from 'react-dom';	
3	import './index.css';	
4	import App from './App';	
5	import reportWebVitals from './reportWebVitals';	
6		
7	ReactDOM.render(	
8	<React.StrictMode>	선표는 지우지 말 것
9	<App />	
10	</React.StrictMode>,	
11	document.getElementById('root')	
12	);	

# 3 리액트 앱살펴보고 수정하기

## ❖ index.js 파일 수정하기

- src 폴더의 index.js 파일을 열고 다음과 같이 표시한 코드를 삭제.
- 앞으로 필요한 코드가 아니니까 걱정하지 말고 과감히 삭제

13	
14	// If you want to start measuring performance in your app, pass a function
15	// to log results (for example: reportWebVitals(console.log))
16	// or send to an analytics endpoint. Learn more: <a href="https://bit.ly/CRA-vitals">https://bit.ly/CRA-vitals</a>
17	reportWebVitals();
18	



# 3 리액트 앱살펴보고 수정하기

## ❖ index.js 파일 수정하기

- 수정이 끝나면 다음과 같은 코드만 남음.

1	<code>import React from 'react';</code>
2	<code>import ReactDOM from 'react-dom';</code>
4	<code>import App from './App';</code>
5	
6	<code>ReactDOM.render(</code>
7	<code>  &lt;App /&gt;,</code>
8	<code>  document.getElementById('root')</code>
9	<code>);</code>

# 3 리액트 앱살펴보고 수정하기

## ❖ App.js 파일 수정하기

- src 폴더의 App.js 파일을 수정.
- 노란색으로 표시된 코드는 삭제하고, 자주색으로 표시된 코드는 새로 작성.
- 소괄호나 꺾쇠 등의 기호를 실수로 지우거나 남기지 않도록 주의!

1	<code>import logo from './logo.svg';</code>
2	<code>import './App.css';</code>
3	
4	<code>function App() {</code>
5	<code>  return (</code>
6	<code>    &lt;div className="App" /&gt;</code>
7	<code>      &lt;header className="App-header"&gt;</code>
8	<code>        &lt;img src={logo} className="App-logo" alt="logo" /&gt;</code>
9	<code>        &lt;p&gt;</code>
10	<code>          Edit &lt;code&gt;src/App.js&lt;/code&gt; and save to reload.</code>
11	<code>        &lt;/p&gt;</code>
12	<code>        &lt;a</code>
13	<code>          className="App-link"</code>
14	<code>          href="https://reactjs.org"</code>
15	<code>          target="_blank"</code>

# 3 리액트 앱살펴보고 수정하기

## ❖ App.js 파일 수정하기

- src 폴더의 App.js 파일을 수정.
- 노란색으로 표시된 코드는 삭제하고, 주황색으로 표시된 코드는 새로 작성.
- 소괄호나 꺾쇠 등의 기호를 실수로 지우거나 남기지 않도록 주의!

16	<b>rel="noopener noreferrer"</b>
17	<b>&gt;</b>
18	<b>Learn React</b>
19	<b>&lt;/a&gt;</b>
20	<b>&lt;/header&gt;</b>
21	<b>&lt;/div&gt;</b>
22	<b>);</b>
23	<b>}</b>
24	
25	export default App;

# 3 리액트 앱살펴보고 수정하기

## ❖ App.js 파일 수정하기

- 수정이 끝나면 다음과 같은 코드만 남음.

1	
2	function App() {
3	return <div className="App" />;
4	}
5	
6	export default App;

# 3 리액트 앱살펴보고 수정하기

## ❖ 리액트 앱 다시 실행하기

- > `npm start`



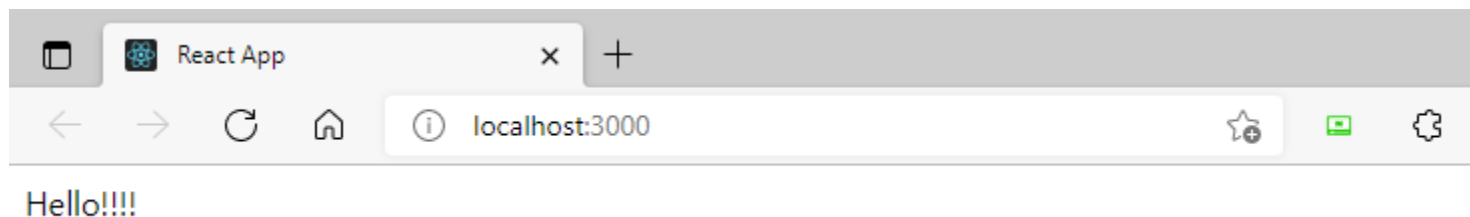
- 오류도 없고, 다 좋다.
- 하지만 화면에 아무것도 표시되지 않았는데 왜 그럴까?
- 그 이유는 아까 `App.js` 파일을 열어 화면에 표시될 코드를 모두 삭제했기 때문.
- 삭제된 내용이 곧바로 리액트 앱 실행 결과에 반영된 것!!

### 3 리액트 앱살펴보고 수정하기

#### ❖ App.js 파일 수정하기

- 빈 화면은 좀 그러니까 인사말을 표시해 보자.
- 다시 **App.js** 파일을 열어서 코드를 변경하고 저장해보자.

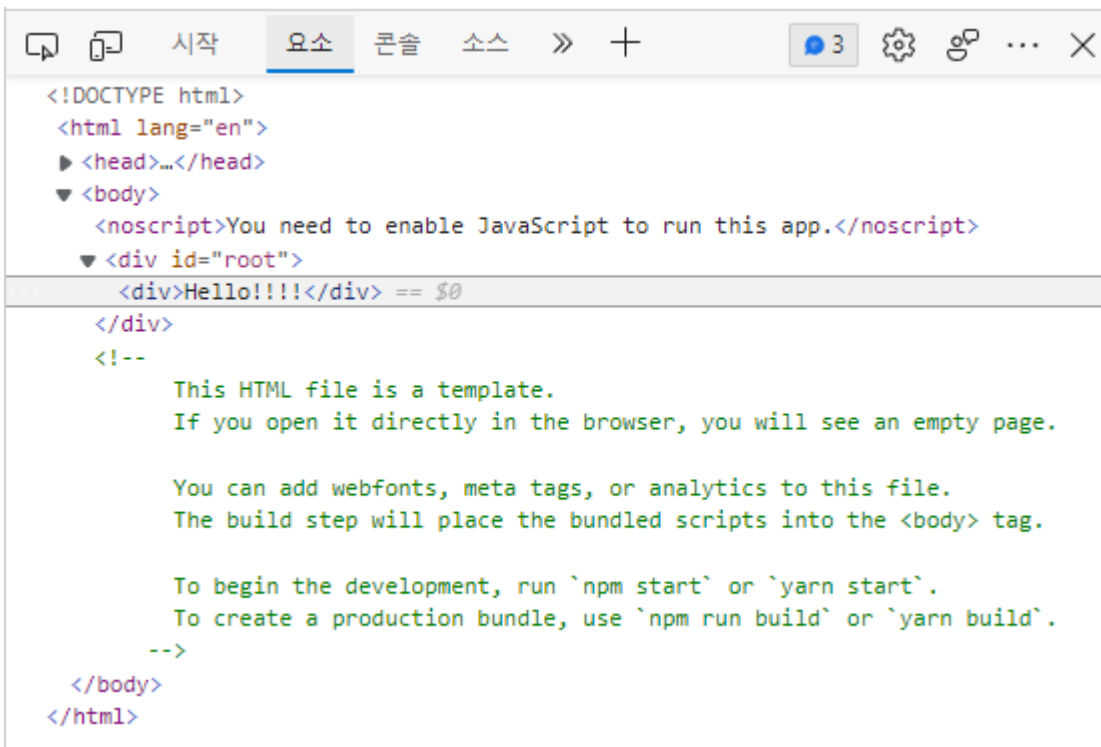
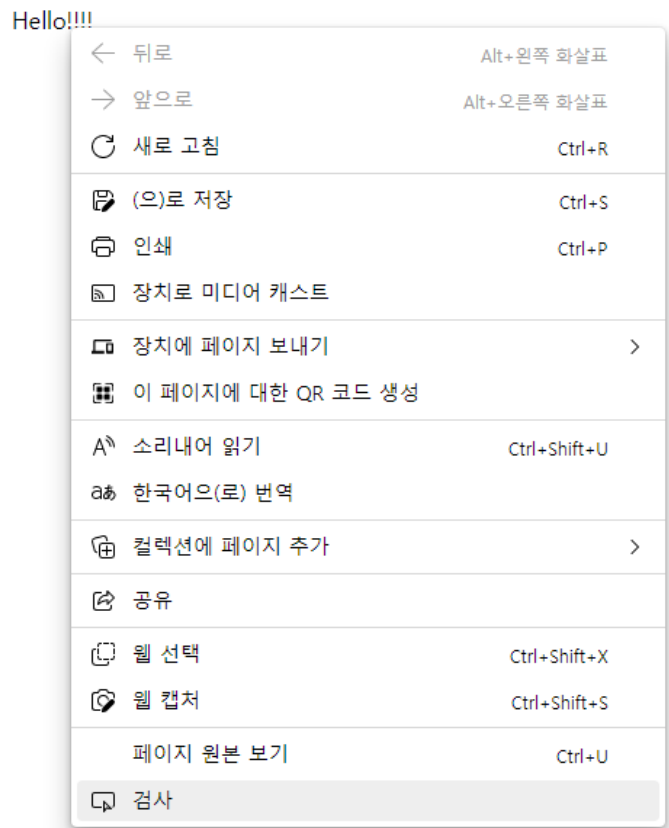
1	
2	function App() {
3	return <div className="App" />;
4	return <div>Hello!!!!</div>;
5	}
6	
7	export default App;



# 3 리액트 앱살펴보고 수정하기

## ❖ 개발자 도구로 리액트 앱 살펴보기

- 브라우저 창에서 Hello!!!!에 마우스 커서를 놓고 오른쪽 버튼을 누른 다음 [검사]를 눌러서 브라우저 개발자 도구를 실행.
- 그런 다음 [요소] 탭을 선택.



# 3 리액트 앱살펴보고 수정하기

## ❖ 개발자 도구로 리액트 앱 살펴보기

- 그러면 앞에서 App.js 파일에 입력한 코드 (`<div>Hello!!!!</div>`) 볼 수 있다.
- 전체 코드를 보면 index.html 파일과 App.js 파일이 합쳐진 것 같다.
- 정말 그럴까?
- index.html 파일을 열어 보자.



# 3 리액트 앱살펴보고 수정하기

## ❖ Hello!!!가 화면에 어떻게 표시되는지 살펴보기

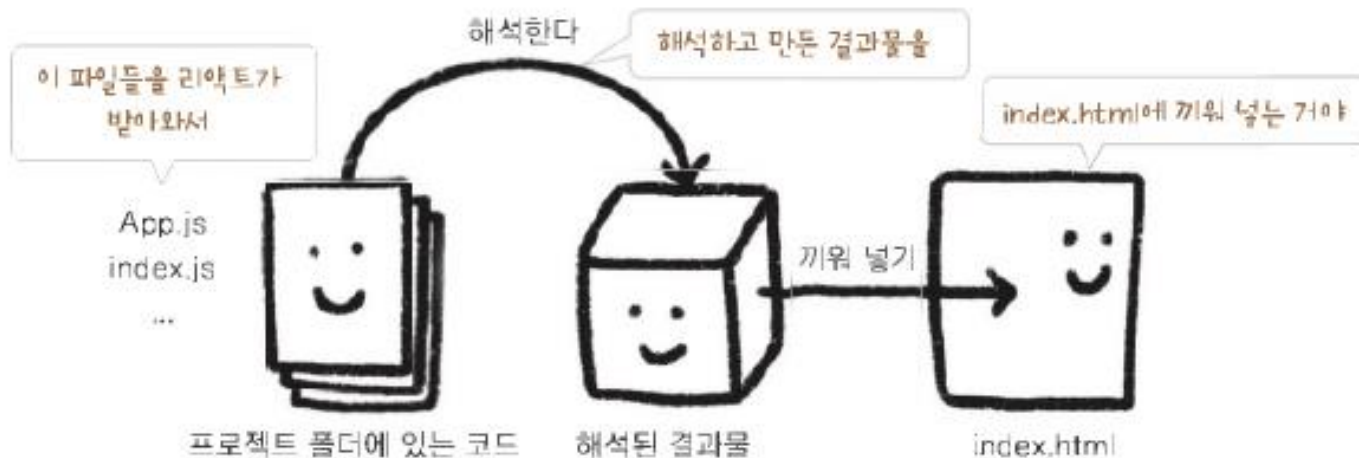
- public 폴더의 index.html 파일을 열어서 body 엘리먼트 주변을 살펴보자.

```
27     <title>React App</title>
28   </head>
29   <body>
30     <noscript>You need to enable JavaScript to run th
31     <div id="root"></div>
32     <!--
33       This HTML file is a template.
34       If you open it directly in the browser, you wil
35
```

- <div id="root">와 </div> 사이에 아무것도 없다!
- 그런데 앞장에서는 이 공간에 코드가 들어 있었다.
- 어떻게 된 걸까?
- 4절에서 그 이유를 알아볼 것이다.

## 4 리액트 동작 원리 알아보기

- ❖ 리액트가 어떻게 동작하는지 원리를 알아보자.
- ❖ 여기를 공부하면 왜 `<div id="root">`와 `</div>` 사이가 비어 있었는지 알 수 있다.
- ❖ 리액트는 우리가 작성한(또는 수정한) 프로젝트 폴더에 있는 코드를 자바스크립트를 이용하여 해석한다.
- ❖ 그리고 해석한 결과물을 `index.html`로 끼워 넣는다.



## 4 리액트 동작 원리 알아보기

- ❖ 그래서 index.html 파일에 없던 `<div>Hello!!!!</div>`가 리액트 앱을 실행하면 생성된다.
- ❖ 조금 더 자세히 이야기해 보면, 리액트는 index.html의 `<div id="root"></div>` 중간에 넣을 결과물을 프로젝트 폴더에 있는 파일 (App.js,...)을 해석하여 만들어 넣는 역할을 담당한다.
- ❖ 정말 그럴까?
- ❖ 실습을 통해 확인해 보자.

# 4 리액트 동작 원리 알아보기

## ❖ index.js 살펴보기

- index.js를 다시 열어 보자.
- 그리고 ReactDOM.render(...) 라고 표시된 부분을 보자.

1	import React from 'react';
2	import ReactDOM from 'react-dom';
4	import App from './App';
5	
6	ReactDOM.render( 7     <App />, 8     document.getElementById('root') 9   );

그린다.

App 컴포넌트를

아이디가 'root' 인 엘리먼트에

- 여기서 주목해야 할 코드는 ReactDOM.render(<App />, document.getElementById('root'))이다.
- 이 코드가 바로 App.js 파일에 작성한 코드를 index.html의 아이디가 'root' 인 엘리먼트에 넣어주는 역할을 한다.
- 정말 그럴까?

## 4 리액트 동작 원리 알아보기

### ❖ index.html 수정해 보기

- index.html 파일을 열어서 `<div id="root"></div>`을 `<div id="potato"></div>`로 바꿔 보자.

	...
29	<code>&lt;body&gt;</code>
30	<code>&lt;noscript&gt;You need to enable JavaScript to run this app.&lt;/noscript&gt;</code>
31	<code>&lt;div id="potato"&gt;&lt;/div&gt;</code>
	...
42	<code>&lt;/body&gt;</code>
43	<code>&lt;/html&gt;</code>

- 이렇게 하면 아이디가 "potato"로 바뀌니까 앱이 실행되지 않을 것!!!

## 4 리액트 동작 원리 알아보기

### ❖ index.html 수정해 보기

- 그러면 리액트 앱을 다시 실행해 보자.
- index.js의 ReactDOM.render(...)에서 document.getElementById('root')는 아이디가 'root'인 엘리먼트를 찾을 텐데, index.html에는 더 이상 아이디가 'root'인 엘리먼트가 없으니 아무 것도 출력하지 않을 것이다.

## 4 리액트 동작 원리 알아보기

### ❖ index.js 수정하여 오류 해결하기

- index.html에서 수정한 코드에 맞춰 index.js 파일의 `document.getElementById('root')`를 `document.getElementById('potato')`로 수정해 보자.

1	<code>import React from 'react';</code>
2	<code>import ReactDOM from 'react-dom';</code>
4	<code>import App from './App';</code>
5	
6	<code>ReactDOM.render(</code>
7	<code>  &lt;App /&gt;,</code>
8	<code>  document.getElementById('potato')</code>
9	<code>);</code>

- 그러면 다시 리액트 앱이 정상으로 작동할 것이다.
- 리액트는 index.js에 있는 `ReactDOM.render()`를 통해 App.js에 있는 `<div>Hello!!!!</div>`를 index.html에 삽입한다.

## 4 리액트 동작 원리 알아보기

### ❖ index.js 수정하여 오류 해결하기



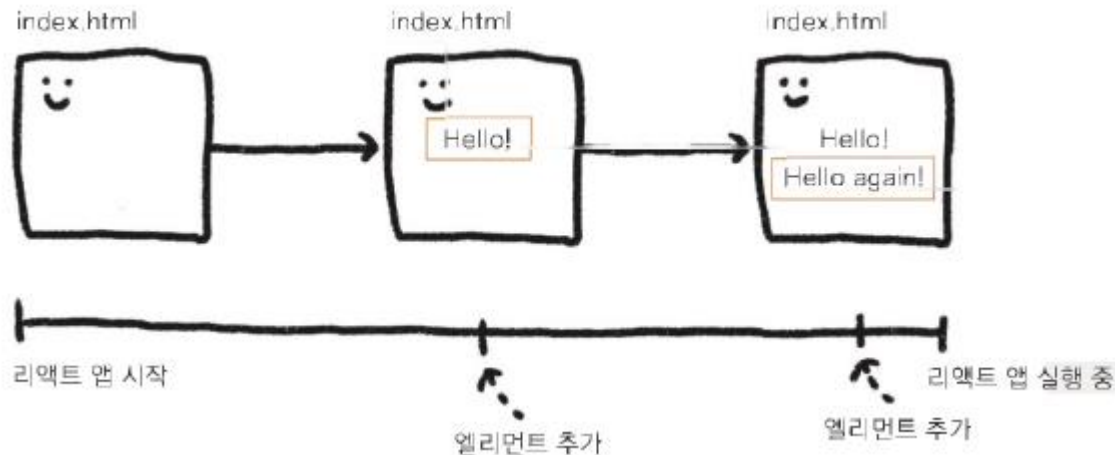
- 이 방식은 리액트가 화면을 빠르게 그릴 수 있게 해주는 장점도 있다.



## 4 리액트 동작 원리 알아보기

### ❖ index.js 수정하여 오류 해결하기

- 가장 왼쪽은 빈 index.html이고 오른쪽은 App.js를 해석하며 점점 채워지는 index.html이다.
- 그림에서 보듯 리액트는 처음부터 모든 HTML을 그려 넣지 않는다.
- 일부 HTML 만 그리고(지금의 경우는 그냥 비어있는 HTML) 이 후 엘리먼트를 추가하거나 제거하는 방식으로 화면을 그린다.
- 리액트는 화면에 표시될 모든 HTML을 처음부터 그리지 않으니까 빠른 거야.



## 4 리액트 동작 원리 알아보기

### ❖ 파일 원래대로 돌려놓기

- 이제 `index.html`과 `index.js`에서 수정했던 'potato'를 'root'로 다시 바꾸자.
- 확장자 `.html`과 `.js`를 헛갈리면 안 된다.
- 리액트의 화면을 수정해 보면서 리액트의 기본 동작 원리를 알아봤다.



**Thank You !**