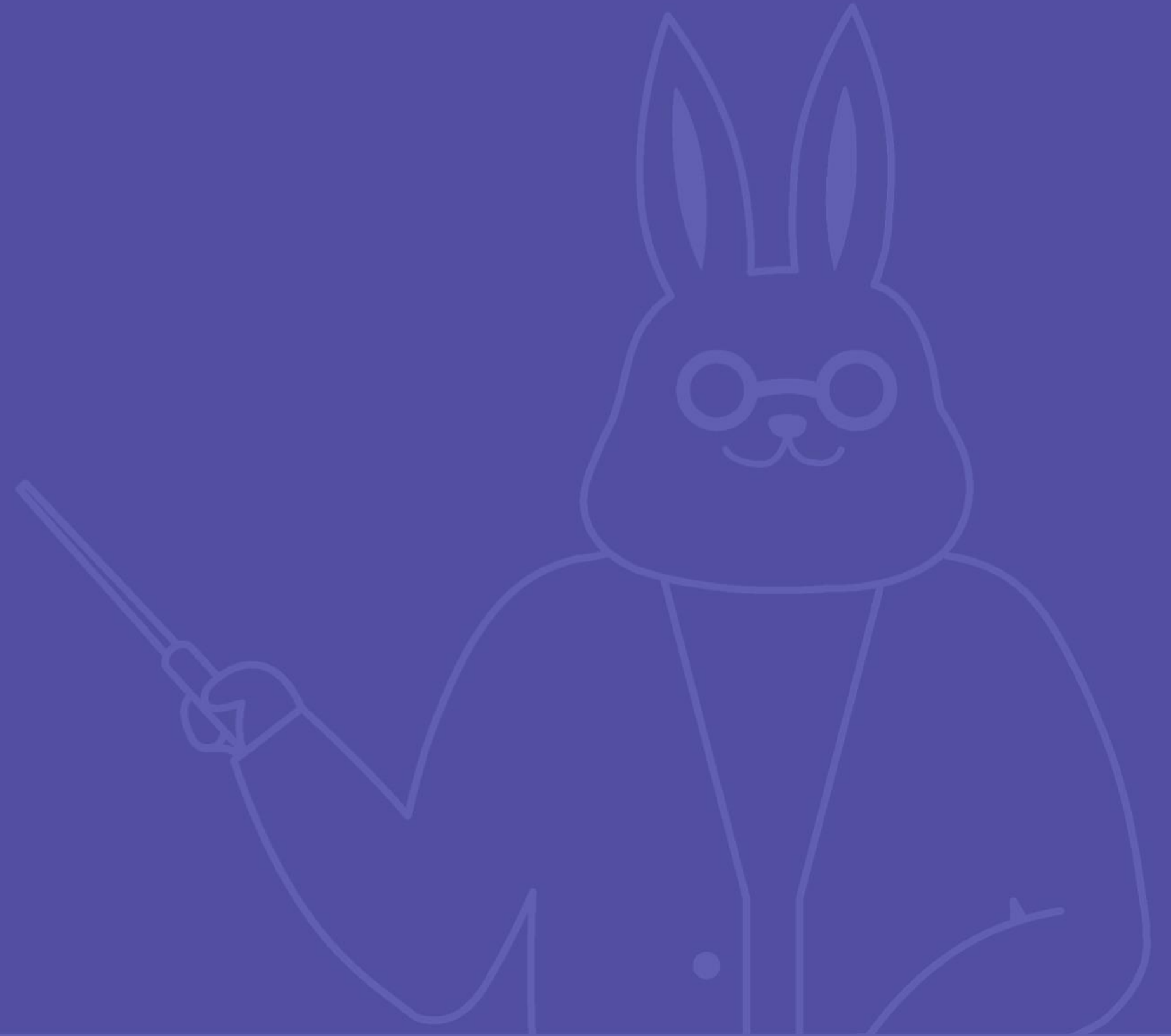


React 기초 I

03 React 프로젝트 생성



목차

01. Create React App 소개
02. Node.js / NPM 소개
03. React 프로젝트 생성해보기
04. 라이브러리 설치와 불러오기

01

Create React App 소개



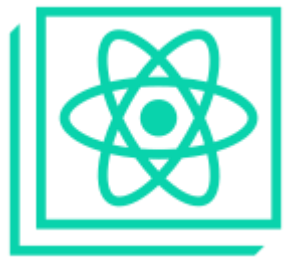
✓ 시작하기 앞서: Create React App을 꼭 써야하는 것은 아닙니다.

jQuery 라이브러리를 사용하는 것 처럼 직접 script를 추가해서 사용해도 돼요.

코드

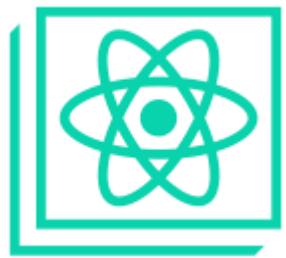
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>React</title>
  </head>

  <body>
    <div id='root'></div>
    <script crossorigin src="https://unpkg.com/react@16/umd/react.development.js"></script>
    <script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
  </body>
</html>
```



Create React App(CRA)란

- React 프로젝트를 손쉽게 생성할 수 있도록 도와주는 ‘보일러플레이트(Boilerplate)’
- 수많은 React용 보일러플레이트가 있지만 Facebook에서 직접 만들어서 관리하는 Create React App이 가장 많이 쓰임
- 프로젝트 생성에 필요한 다양한 기능을 Command로 제공함



Create React App(CRA)의 장점

1. 개발자가 온전히 React App 개발에 집중할 수 있도록 함
 - 상대적으로 덜 중요한 코드는 노출되지 않음
 - 강력한 Command 지원
2. (대부분의) 모든 브라우저에서 해석될 수 있도록 transcompile 지원
 - Babel
 - 배포 시 코드 번들링
 - Webpack

02

Node.js / NPM 소개



✓ Node.js / NPM



- Create React App으로 React 프로젝트 시작 시 Node.js 개발 환경을 제공함
- Node.js와 NPM에 대해 알고 있다면 이해에 큰 도움이 됨

✓ Node.js

```
const http = require('http');

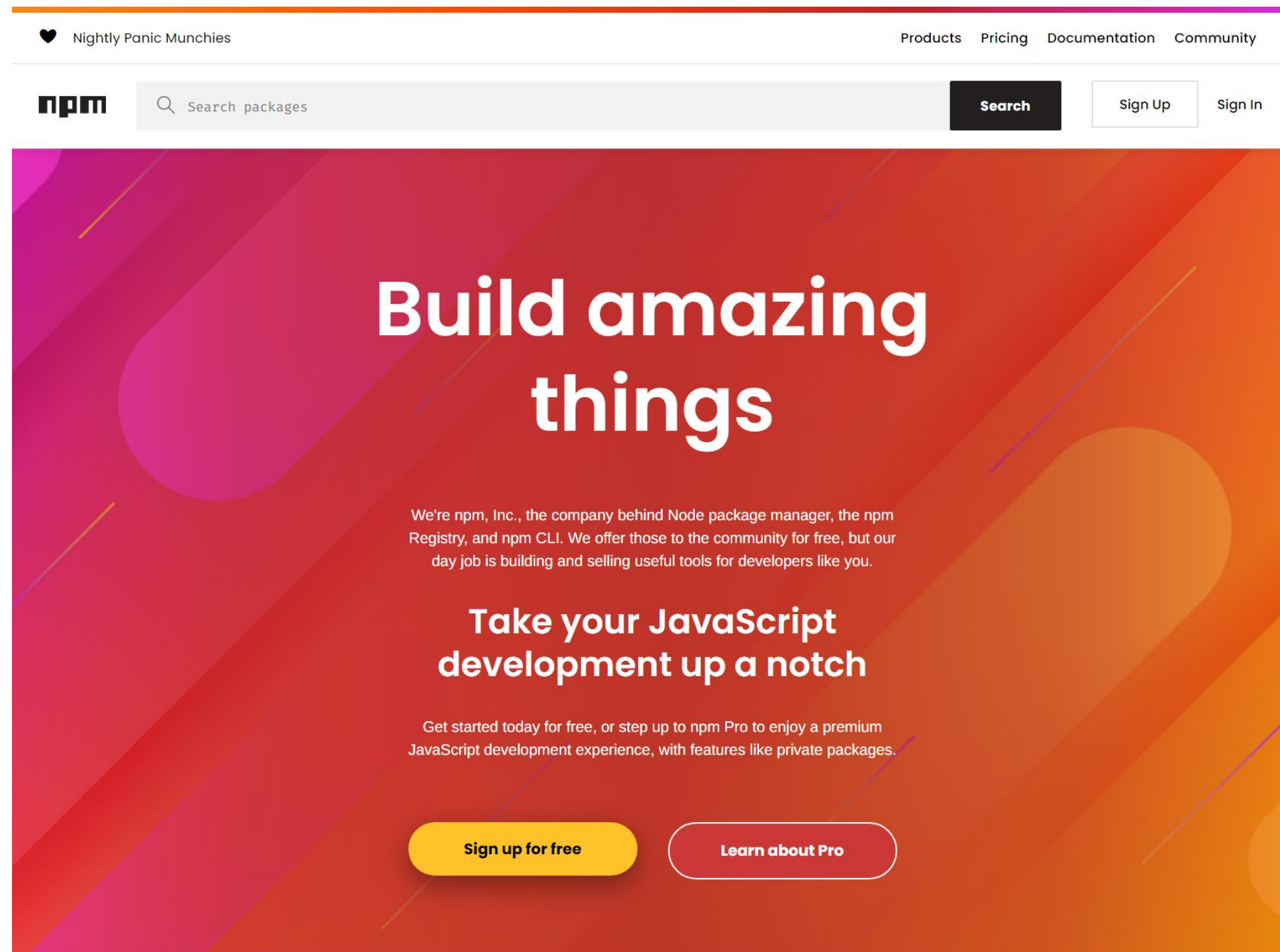
const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

- 주로 서버 프로그래밍에 사용되는 JavaScript 기반의 소프트웨어 플랫폼
- 2009년 발표, 2021년 기준 v16까지 업데이트
- 프론트엔드 개발자의 접근성 ↑
- HTTP 통신 관련 라이브러리 내장
- NPM을 통한 방대한 라이브러리 제공
- **Create React App으로 프로젝트 생성 시 개발 환경 및 테스트 서버로 이용됨**

✓ NPM



- Node.js 환경에서 사용하는 각종 패키지들을 관리하는 저장소
- Node.js 설치 시 함께 설치됨
- 패키지 관리 뿐만 아니라 서버 실행 및 관리에 필요한 다양하고 편리한 명령어를 제공
- 우리가 사용할 **React**와 관련된 **모듈들이 NPM**에서 배포되고 있음

✓ Node.js, NPM 설치하기

<https://nodejs.org> 접속 또는 Node.js 검색



LTS 버전을 다운로드 후 설치해주세요.

03

React 프로젝트 생성해보기



✓ 함께 명령어를 입력하며 React 프로젝트를 만들어보아요

bash

```
npx create-react-app <디렉토리명>  
cd <디렉토리명>  
npm start
```

npx

npm 패키지를 1회성으로 내려 받아 실행할 때 사용하는 명령어입니다.

왼쪽의 예시는 ‘create-react-app’ 이라는 패키지를 한 번 내려 받아 실행한 것입니다.

cd

<디렉토리명>으로 이동합니다.

npm start

현재 디렉토리의 프로젝트를 실행합니다.

✓ 많이 사용하는 명령어를 알아보아요

bash

```
npm install
npm install <패키지명>
npm start
npm build
(Ctrl + C)
```

더 많은 명령어 확인: <https://docs.npmjs.com/cli/v7/commands>

npm install

package.json에 정의된 dependency(의존성 패키지)들을 설치합니다.

npm install <패키지명>

npm 서버로부터 원하는 패키지를 내려 받습니다.

npm start

프로젝트를 실행합니다(Node.js 서버 이용).

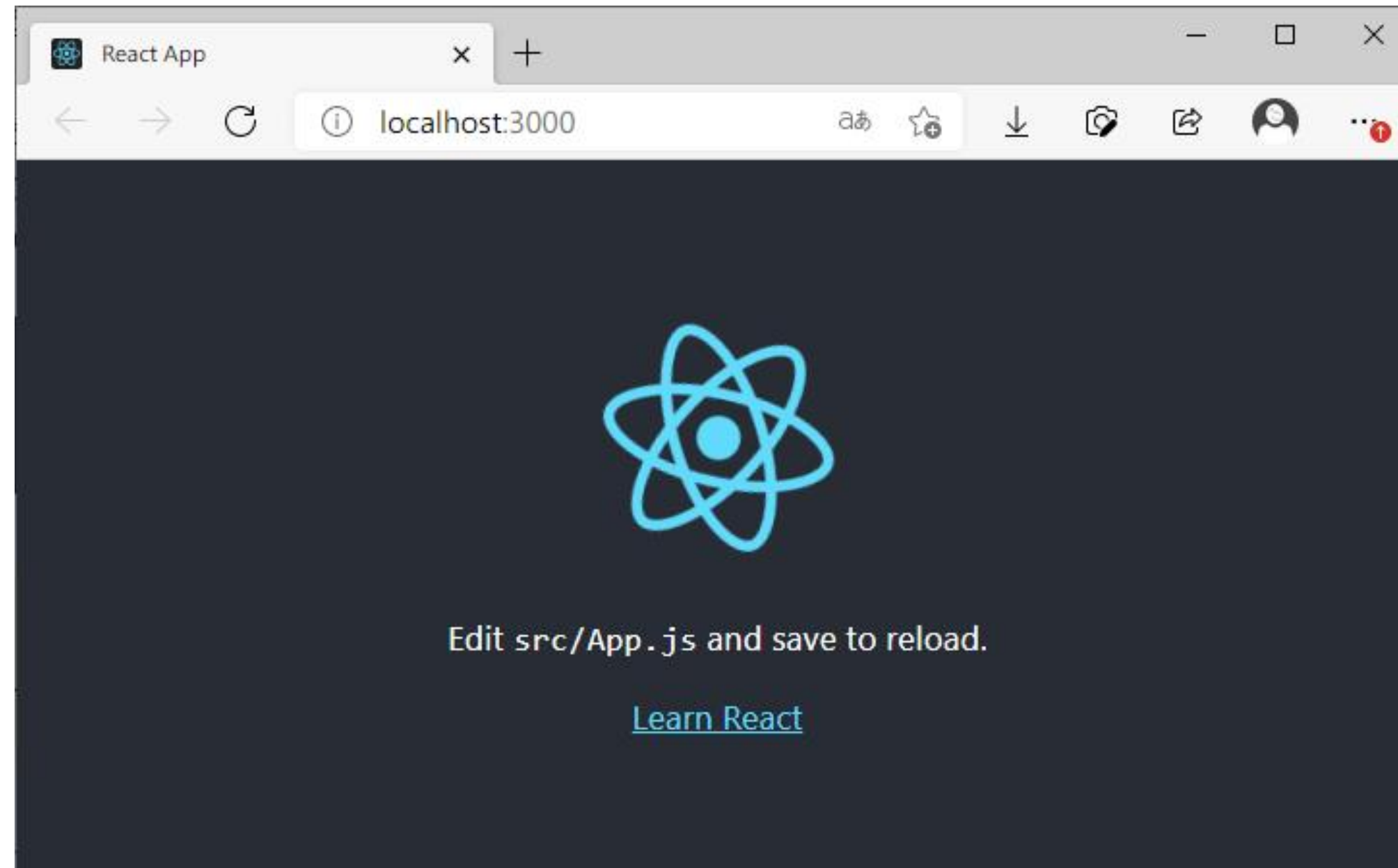
npm build

프로젝트를 빌드합니다.

(Ctrl + C)

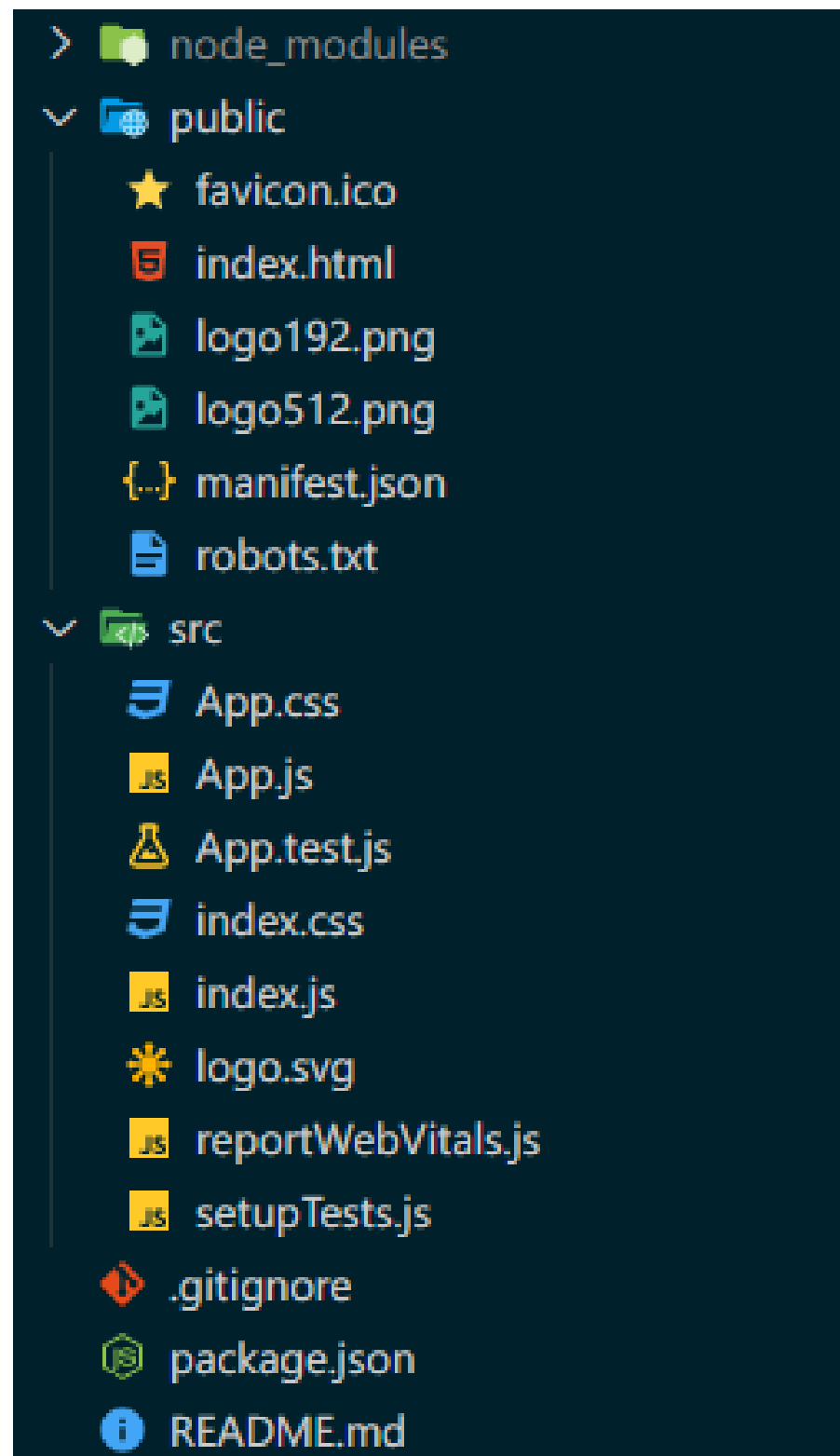
명령을 중지합니다(npm start 후 종료하기).

✓ 점검



이 화면이 뜨면 성공입니다.

✓ 디렉토리 구조 살펴보기



`./node_modules/`

npm을 이용해 설치한 패키지들 모음

`./public/`

정적인 파일들을 모아 놓는 곳

`./src/`

리액트 개발을 위한 파일들을 모아 놓는 곳

`./.gitignore`

Git을 이용할 경우 불필요한 파일을 무시할 수 있도록 하는 설정파일

`./package.json`

프로젝트에 관한 정보와 사용하는 패키지들을 명세하는 파일

`./README.md`

내 프로젝트에 관한 설명을 작성하는 파일

04

라이브러리 설치와 불러오기



✓ 라이브러리 설치

bash

```
npm install
```

package.json 파일 내에 정의된 패키지 모두 설치

```
npm install <패키지명>
```

npm 서버로부터 패키지 내려받기

```
npm install <패키지명>@<version>
```

특정 버전의 패키지 내려받기

```
npm install <Git 레포지토리 주소>
```

npm이 아닌 Git 레포지토리로부터 패키지 내려받기

npm install을 통해 npm 으로부터 패키지를 내려받습니다.

✓ 설치 후 package.json을 열어보면...

json

```
{
  ...
  "dependencies": {
    "@testing-library/jest-dom": "^5.11.4",
    "@testing-library/react": "^11.1.0",
    "@testing-library/user-event": "^12.1.10",
    "react": "^17.0.2",
    "react-dom": "^17.0.2",
    "react-scripts": "4.0.3",
    "web-vitals": "^1.0.1"
  },
  ...
}
```

dependencies(의존성) 라는 키 아래에 설치한 패키지 목록이 나열됩니다.

참고: 버전명에는 ^(캐럿), <, <=, >, >= 등의 기호를 이용해 범위를 나타낼 수 있습니다.

^(캐럿) 예시: ^1.0.2 : >=1.0.2 <2.0
 ^1.0 : >=1.0.0 <2.0
 ^1 : >=1.0.0 <2.0

✓ 설치한 라이브러리를 프로젝트 내에서 불러오기

JS

```
import "패키지명"
```

```
import something from "패키지명"
```

```
import { a, b } from "패키지명"
```

```
import * as something from "패키지명"
```

1. CSS나 import하는 것만으로 역할을 하는 라이브러리인 경우 패키지명을 바로 import 합니다.
2. 기본적으로 패키지를 불러와 활용할 때에는 할당할 이름을 작성합니다.
3. 패키지 내의 일부 메소드나 변수만 가져올 때는 구조분해를 하여 가져오는 것이 효율적입니다.
4. 패키지에 default로 export되는 객체가 존재하지 않을 경우 `* as 이름` 으로 불러올 수 있습니다.

✓ CSS 불러오기

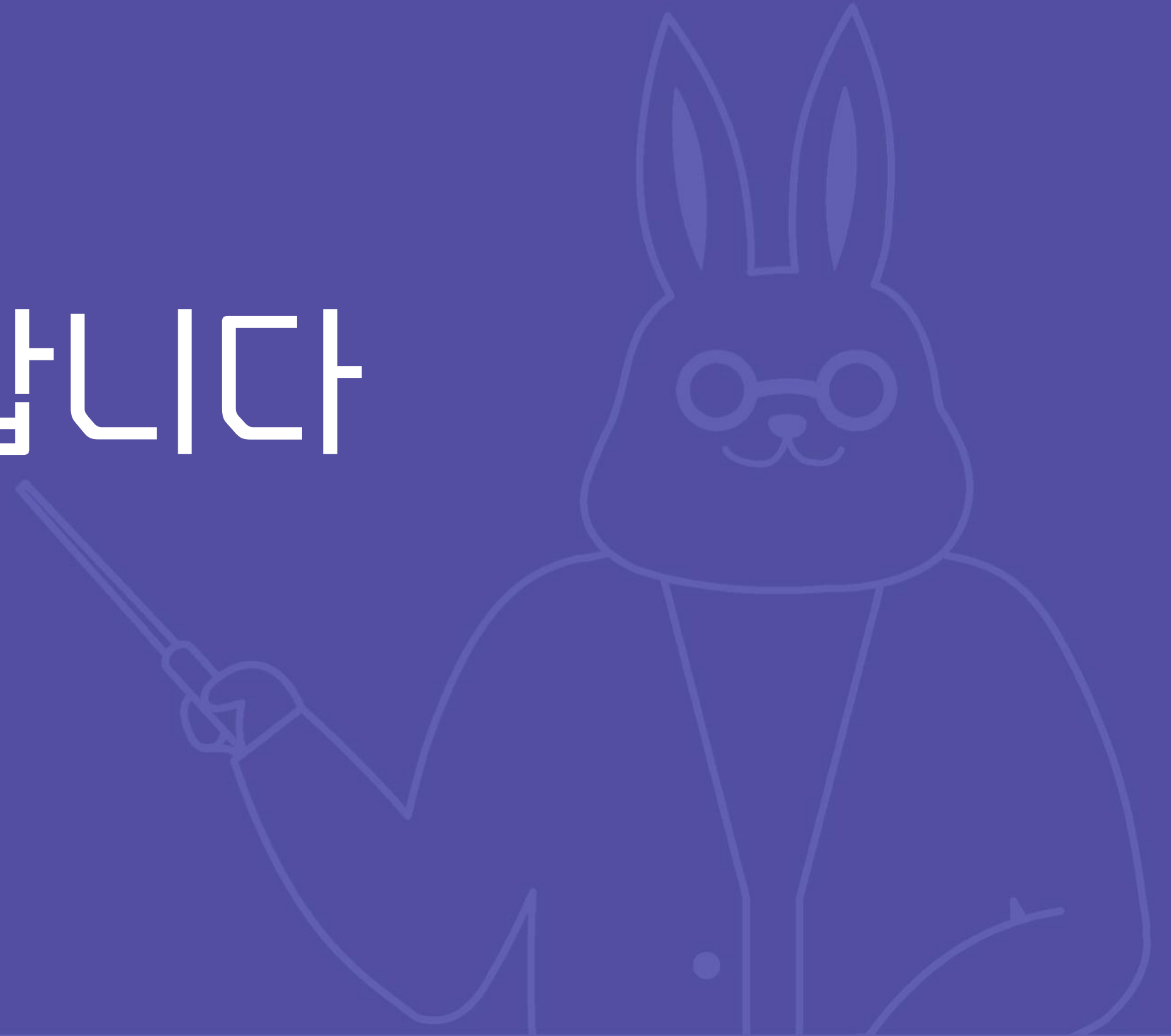
코드

```
import "../App.css";
```

별도의 CSS 파일을 작성 후 프로젝트에 적용하고 싶은 경우,

import 할 경우 style이 적용됩니다.

감사합니다



크레딧

/* elice */

코스 매니저

콘텐츠 제작자

마로

강사

마로

감수자