

Front-End Clone Coding

2학기 프론트엔드 정기 세션 기본 세팅 자료

들어가기 전,,

1. cmd를 실행합니다.
2. node -v를 입력하여 node.js 설치 여부를 확인합니다.
(설치되어 있지 않다면 설치를 진행합니다. 설치가 되어있다면 버전이 출력됩니다.)
3. npm -v를 입력하여 npm 버전을 확인합니다.
4. npm install npx -g를 입력하여 npx를 npm으로 설치합니다.
(npx는 1회성으로 최신 버전의 노드 패키지를 내려받아 설치시켜주는 노드 패키지입니다.)
5. npx -v를 입력하여 npx 설치가 완료되었는지 확인합니다.

“

create-react-app

create-react-app을 사용하면 명령어 1줄만 입력하여 리액트 개발을 바로 시작할 수 있습니다.

create-react-app은 리액트 개발을 바로 시작할 수 있도록
프로젝트 구조 작업, 설정 작업 등을 자동으로 진행해 주는 도구입니다.

우리는 이 도구를 이용할 것입니다.

1. cmd창을 실행시킵니다.
2. 리액트 앱을 만들고 싶은 곳으로 이동합니다. (cd 명령어 사용) (필수 과정 X)
- 3.. `npx create-react-app movie_app_2022`를 입력합니다.

01 create-react-app

 명령 프롬프트

```
Microsoft Windows [Version 10.0.19042.1889]  
(c) Microsoft Corporation. All rights reserved.
```

```
C:\#Users#Yunsol Shin>npx create-react-app movie_app_2022
```

왼쪽과 같은 화면으로 진행하면 됩니다.

이 과정은 시간이 걸리기 때문에 창을 내리거나 끄지 않고 기다리도록 합니다.

과정이 끝나면 아래와 같은 화면이 됩니다.

```
We suggest that you begin by typing:
```

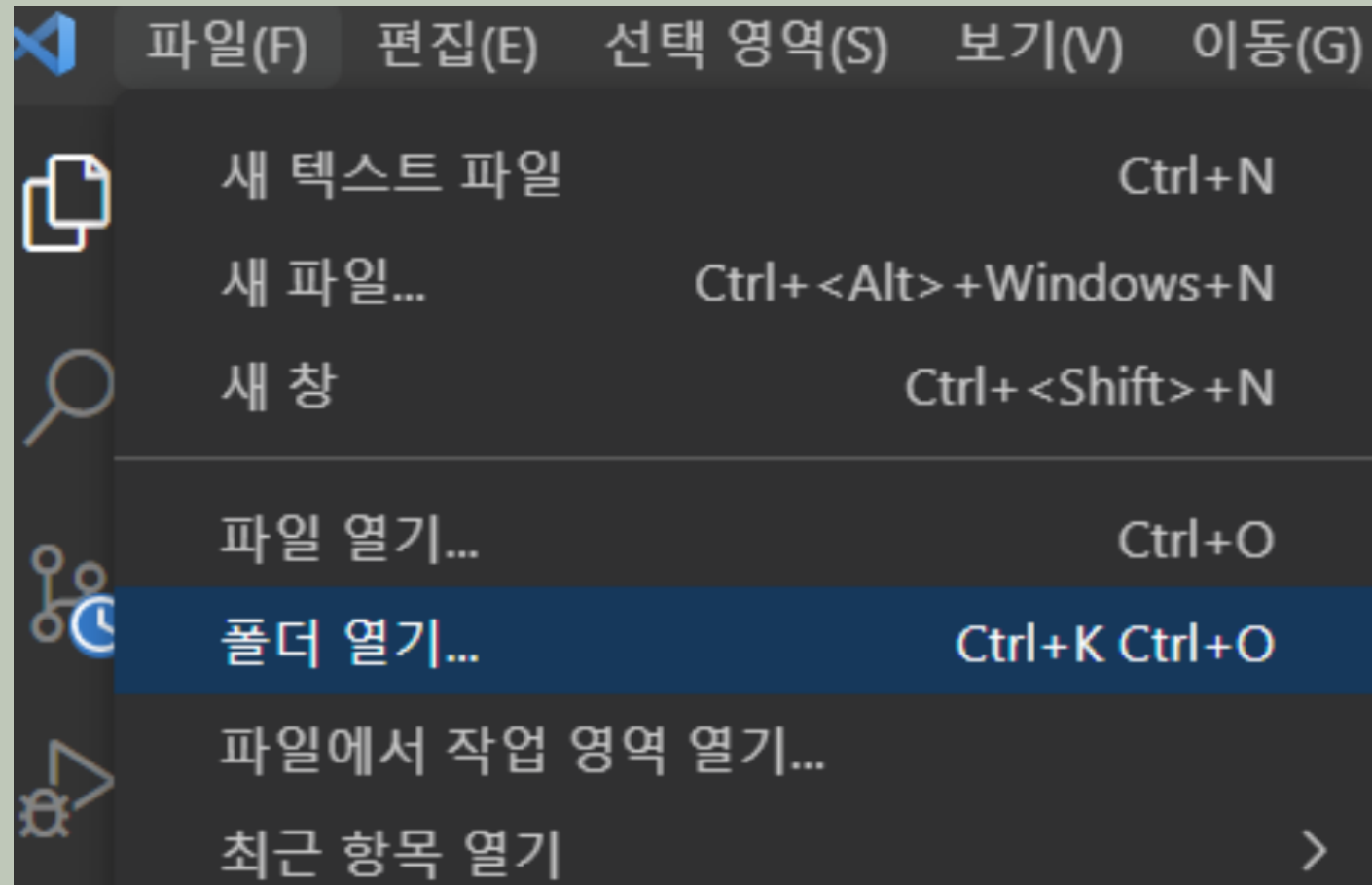
```
cd movie_app_2022  
npm start
```

```
Happy hacking!
```

```
C:\#Users#Yunsol Shin>
```

02 프로젝트 폴더 선택하기

VSCode를 실행하여 [파일 → 폴더 열기]를 누른 후 1번 과정에서 만든 movie_app_2022 폴더를 선택합니다.



03 README.md 파일 수정하기

1. 폴더 안에 있는 README.md 파일을 연다.

2. 그 안에 작성되어 있는 내용을 모두 지운다.


3.

Movie App 2022

React JS Fundamentals Course 2022

라고 입력 후 Ctrl + S로 저장한다.

① README.md X

① README.md >  # Movie App 2022

1 # Movie App 2022

2


3 React JS Fundamentals Course 2022

04 package.json 파일 수정하기

1. 폴더 안에 있는 package.json 파일을 연다.

2. test와 eject 명령어 부분을 삭제 후 저장한다.

```
{ } package.json ×  
{ } package.json > { } scripts  
1 {  
2   "name": "movie_app_2022",  
3   "version": "0.1.0",  
4   "private": true,  
5   "dependencies": {  
6     "@testing-library/jest-dom": "^5.16.5",  
7     "@testing-library/react": "^13.3.0",  
8     "@testing-library/user-event": "^13.5.0",  
9     "react": "^18.2.0",  
10    "react-dom": "^18.2.0",  
11    "react-scripts": "5.0.1",  
12    "web-vitals": "^2.1.4"  
13  },  
14  "scripts": {  
15    "start": "react-scripts start",  
16    "build": "react-scripts build",  
17    "test": "react-scripts test",  
18    "eject": "react-scripts eject"  
19  },
```



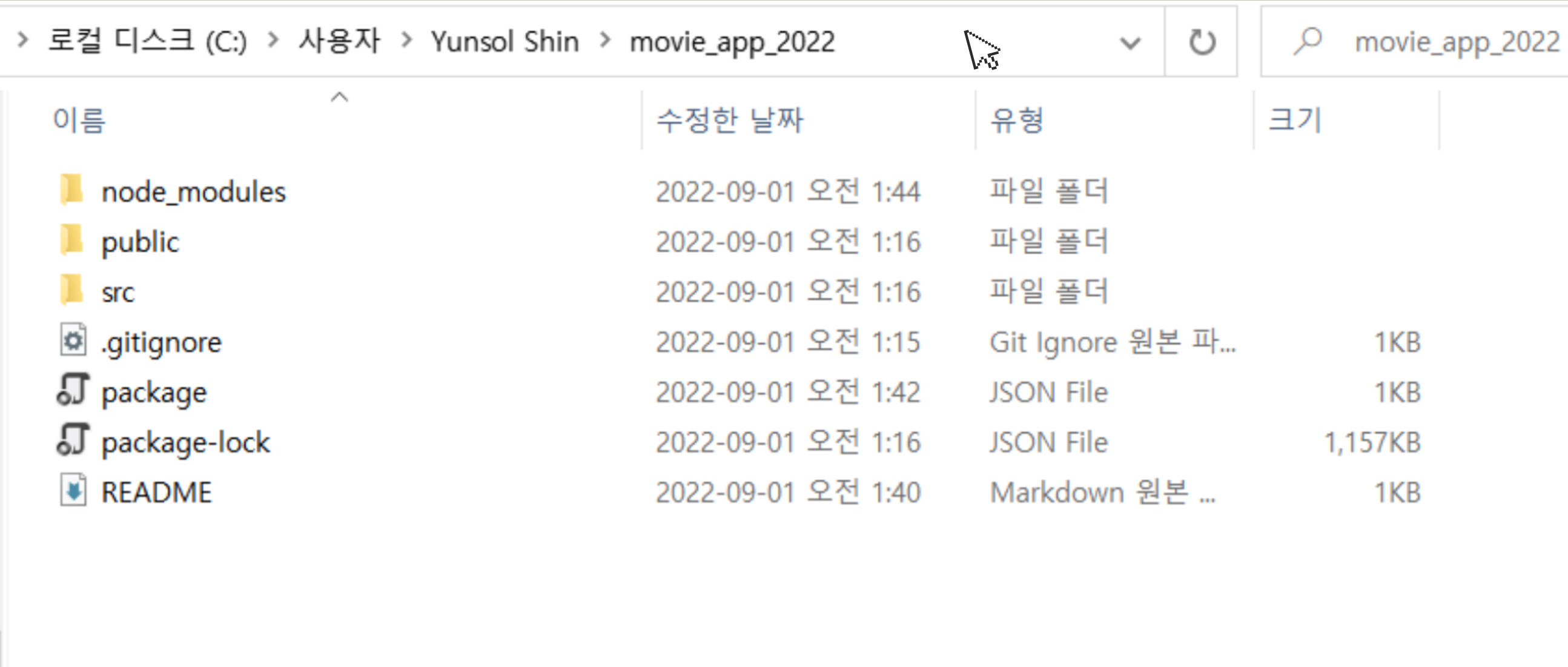
```
    "scripts": {  
      "start": "react-scripts start",  
      "build": "react-scripts build"  
    },
```


05 리액트 앱 실행하기

1. 1번 과정에서 만든 폴더로 들어와 준다.
(아래 사진 참고)

로컬 디스크 (C:) > 사용자 > Yunsol Shin > movie_app_2022				movie_app_	
이름	수정한 날짜	유형	크기		
node_modules	2022-09-01 오전 1:44	파일 폴더			
public	2022-09-01 오전 1:16	파일 폴더			
src	2022-09-01 오전 1:16	파일 폴더			
.gitignore	2022-09-01 오전 1:15	Git Ignore 원본 파...	1KB		
package	2022-09-01 오전 1:42	JSON File	1KB		
package-lock	2022-09-01 오전 1:16	JSON File	1,157KB		
README	2022-09-01 오전 1:40	Markdown 원본 ...	1KB		

2. 아래 사진과 같은 위치에 커서를 올려 놓는다.



3. 클릭을 한다.
클릭을 하면 아래와 같은 화면이 될 것이다. (파랑게)

클립보드	구성	새로 만들기
C:\Windows\System32\cmd.exe		
^	이름 ^	수정한 날짜
	node_modules	2022-09-01 오전 1:44

05 리액트 앱 실행하기

4. 그 상태로

cmd

라고 입력을 한다.

그러면 자신의 폴더 위치에 대한 cmd창이 열린다. 

(아래와 같이 movie_app_2022 파일명으로 나온다.)

```
Microsoft Windows [Version 10.0.19042  
(c) Microsoft Corporation. All rights  
C:\Users\Yunsol Shin\movie_app_2022>n
```

05 리액트 앱 실행하기

5. npm start

라고 입력한다.

```
C:\#Users#Yunsol Shin#movie_app_2022>npm start
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.

> movie_app_2022@0.1.0 start
> react-scripts start

(node:2008) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:2008) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view movie_app_2022 in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.25.240.1:3000

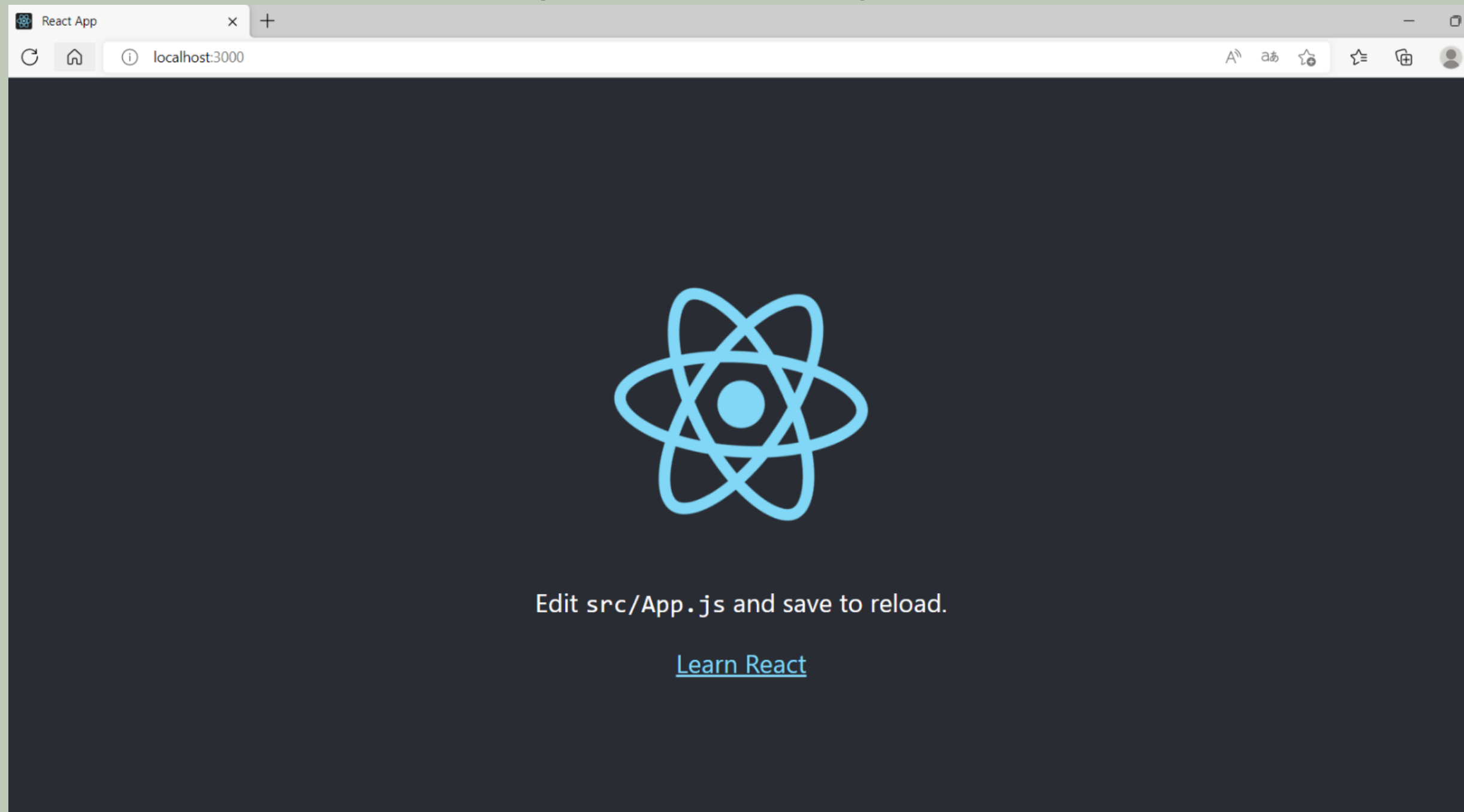
Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

05 리액트 앱 실행하기

5. npm start
라고 입력한다.

앞 페이지의 사진과 같이 'Compiled Successfully!'라는 문장이 보인 후 리액트 앱이 실행된다.



06 src 폴더 정리하기

src 폴더 내에 있는 파일들을 정리해줍니다.

앞으로 사용하지 않을 파일들입니다.

(App.css, App.test.js, index.css, logo.svg, setupTests.js, package-lock.json, reportWebVitals.js)



다음 6개 파일을 삭제하시겠습니까?

App.css

App.test.js

index.css

logo.svg

setupTests.js

package-lock.json

Visual Studio Code



'reportWebVitals.js'을(를) 삭제하시겠습니까?

휴지통에서 이 파일을 복원할 수 있습니다.

07 index.js 파일 수정하기

src 폴더의 index.js 파일을 열고 아래와 같이 노란색으로 표시한 코드들은 삭제한다.

JS index.js X

src > JS index.js > ...

```
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import App from './App';
5  import reportWebVitals from './reportWebVitals';
6
7  const root = ReactDOM.createRoot(document.getElementById('root'));
8  root.render(
9    <React.StrictMode>
10     <App />
11   </React.StrictMode>
12 );
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: https://bit.ly/CRA-vitals
17 reportWebVitals();
18
```


08. App.js 파일 수정하기

```
JS App.js X
src > JS App.js > App
1 import React from 'react';
2 import logo from './logo.svg';
3 import './App.css';
4
5 function App() {
6   return (
7     <div className="App">
8       <header className="App-header">
9         <img src={logo} className="App-logo" alt="logo" />
10        <p>
11          Edit <code>src/App.js</code> and save to reload.
12        </p>
13        <a
14          className="App-link"
15          href="https://reactjs.org"
16          target="_blank"
17          rel="noopener noreferrer"
18        >
19          Learn React
20        </a>
21      </header>
22    </div>
23  );
24 }
25
26 export default App;
```

1. src 폴더의 App.js 파일을 열고 왼쪽과 같이 노란색으로 표시한 코드들은 삭제한다.

2. 7번째 줄에

<div className="App">를

<div className="App" />로 바꿔준다.

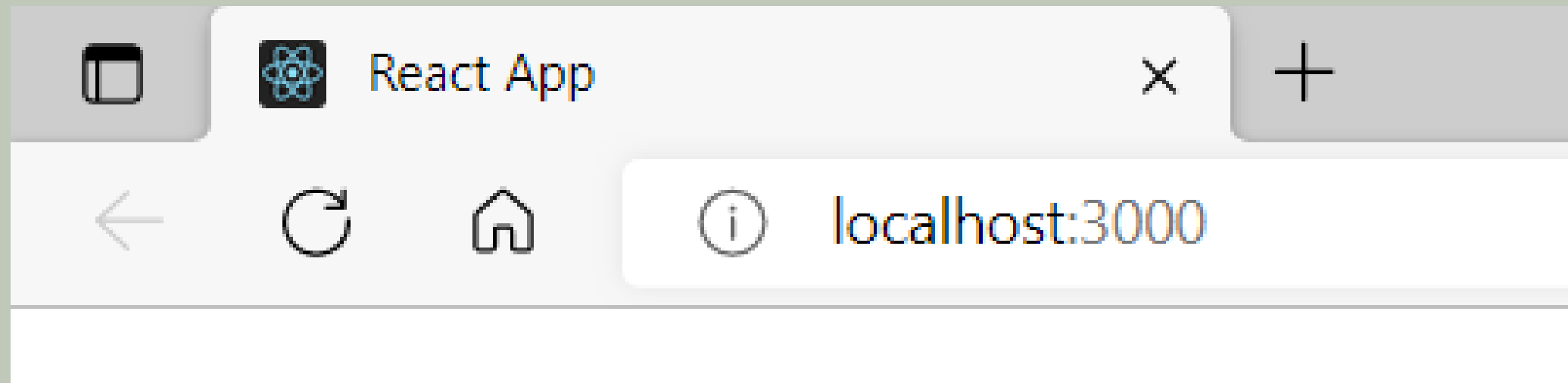
아래와 같이 수정해준다.

3. 저장한다.

```
JS App.js
src > JS App.js > default
1 import React from 'react';
2
3
4 function App() {
5   return <div className="App"/>
6 }
7
8 export default App;
```

08. App.js 파일 수정하기

앞 페이지와 같이 수정 후
npm start를 입력해보면 오류는 없지만
아래 사진과 같이 화면에 아무것도 표시되지 않는다.



08. App.js 파일 수정하기

따라서, 아래와 같이 코드 수정을 다시 해준다.

```
JS App.js ×
src > JS App.js > [🔗] default
  1  import React from 'react';
  2
  3
  4  function App() {
  5    | return <div>Hello Likelion</div>
  6  }
  7
  8  export default App;
```

수정 후 저장을 한다.

이후 다시 npm start를 해보면 아래와 같이 Hello Likelion이 화면에 출력되는 걸 볼 수 있다.

