



## 페이스북에서 만든 웹 개발



### 프레임워크

- **angular, vue**와 함께 가장 많이 사용되고 있는 웹 개발 프레임워크
- **html, jquery**만 이용할 때 보다 좀 더 간편하게 사용자와 상호작용하는 기능을 개발 할 수 있음
- 보여주고 싶은 값이 변했을 때 화면에 반영하는 것을 쉽게 도와줌

0

# 리액트 시작하기

---

WHATEVER YOU WANT, MAKE IT REAL.

강사  
방기진

1. React란

- a. node.js 설치
- b. React 프로젝트 셋업

2. 컴포넌트 & prop

- a. Counter 컴포넌트 만들기
- b. prop이란?

3. State

- a. useState
- b. Counter와 button

4. UseEffect

node

홈 | ABOUT | 다운로드 | 문서 | 참여하기 | 보안 | 뉴스 | CERTIFICATION

## 다운로드

최신 LTS 버전: **16.13.1** (includes npm 8.1.2)

플랫폼에 맞게 미리 빌드된 Node.js 인스톨러나 소스코드를 다운받아서 바로 개발을 시작하세요.

**LTS**  
 대다수 사용자에게 추천

**현재 버전**  
 최신 기능

Windows Installer

node-v16.13.1-x86.msi

macOS Installer

node-v16.13.1.pkg

Source Code

node-v16.13.1.tar.gz

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v16.13.1.tar.gz	

시작하기에 앞서 사용하는 컴퓨터에 **node.js** 를 설치해야 합니다.

<https://nodejs.org/ko/download> 에 접속해서 자신의 컴퓨터에 맞는 인스톨러를 설치하고 **node.js** 를 설치해주세요.

```
genie@genieui-MacBookPro: ~/bang/lecture/react-basic
> genie ~/bang/lecture/react-basic npx create-react-app my-app

Creating a new React app in /Users/genie/bang/lecture/react-basic/my-app.

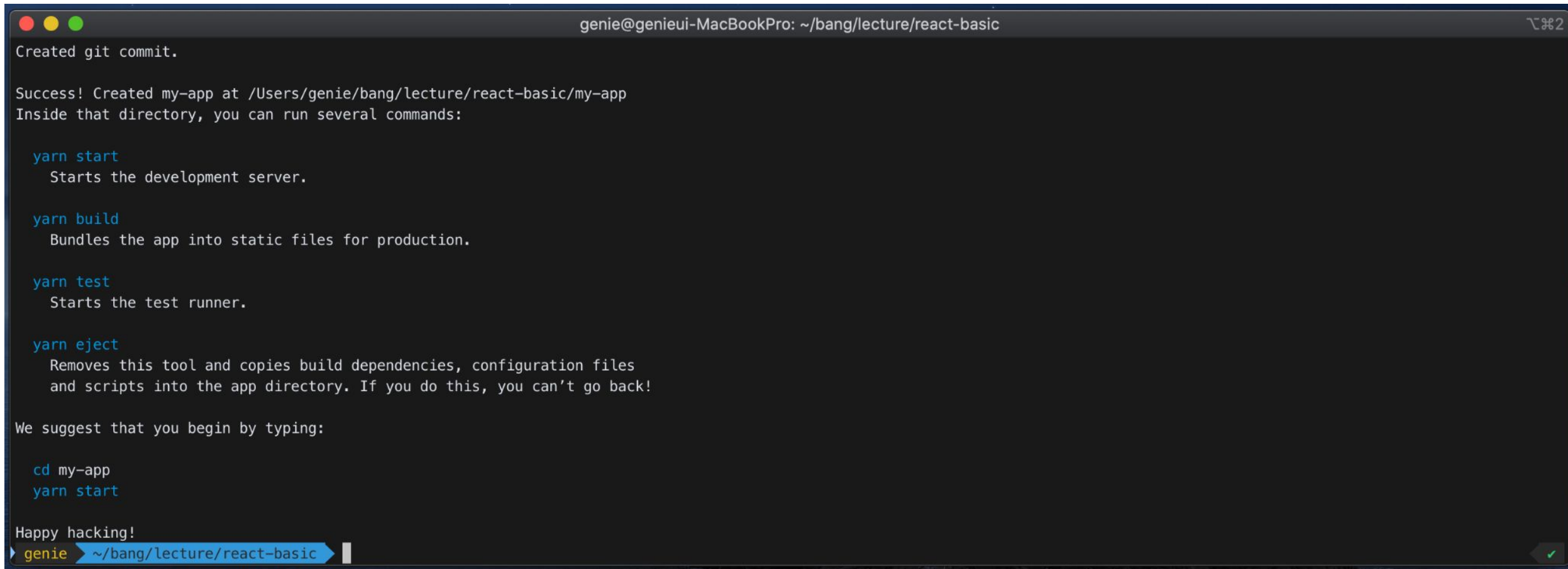
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

yarn add v1.22.4
[1/4] 🔍 Resolving packages...
[2/4] 🚚 Fetching packages...
[3/4] 🔗 Linking dependencies...
warning "react-scripts > @typescript-eslint/eslint-plugin > tsutils@3.20.0" has unmet peer dependency "typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta".
[4/4] ⚡ Building fresh packages...
success Saved lockfile.
success Saved 7 new dependencies.
info Direct dependencies
├─ cra-template@1.1.2
├─ react-dom@17.0.2
├─ react-scripts@4.0.3
└─ react@17.0.2
info All dependencies
├─ cra-template@1.1.2
├─ immer@8.0.1
└─ react-dev-utils@11.0.4
```

node.js를 설치하면 npx가 같이 설치됩니다.  
터미널 또는 쉘을 켜서 작업할 폴더에 들어가세요.

\$ npx create-react-app my-app

리액트 기본 템플릿이 my-app이라는 디렉토리에 설치됩니다.

A terminal window with a dark background and light text. The title bar shows 'genie@genieui-MacBookPro: ~/bang/lecture/react-basic'. The output of the 'create-react-app my-app' command is displayed, including success messages and a list of available yarn commands. The prompt at the bottom shows the user 'genie' at the directory '~/bang/lecture/react-basic'.

```
genie@genieui-MacBookPro: ~/bang/lecture/react-basic
Created git commit.

Success! Created my-app at /Users/genie/bang/lecture/react-basic/my-app
Inside that directory, you can run several commands:

  yarn start
    Starts the development server.

  yarn build
    Bundles the app into static files for production.

  yarn test
    Starts the test runner.

  yarn 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!

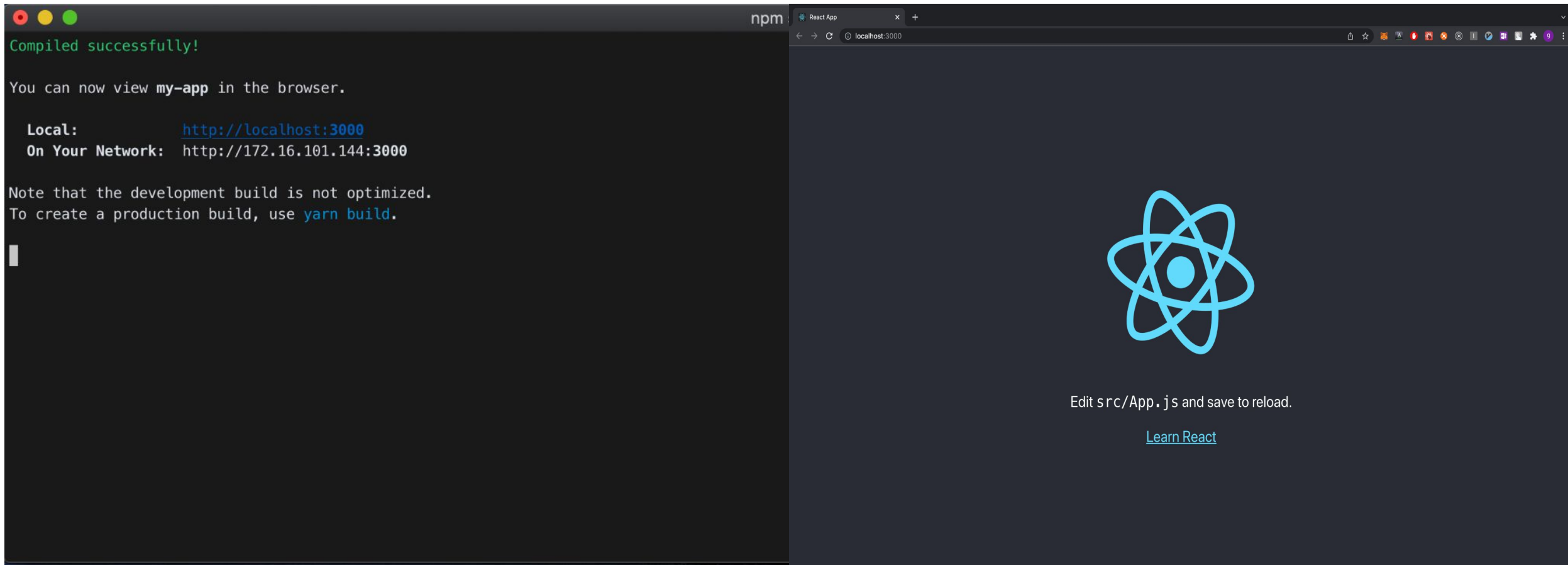
We suggest that you begin by typing:

  cd my-app
  yarn start

Happy hacking!
genie ~/bang/lecture/react-basic
```

\$ npx create-react-app my-app

리액트 기본 템플릿이 **my-app**이라는 디렉토리에 설치됩니다.



```
$ cd my-app  
$ npm start
```

리액트 프로젝트가 시작되고 보통 브라우저가 열리면서 **localhost:3000**으로 연결됩니다.  
브라우저가 켜지지 않으면 크롬을 키고 주소창에 **localhost:3000**을 입력해주세요.

```
1 import logo from './logo.svg';    You, 42 minutes ago • Initialize p
2 import './App.css';
3
4 function App() {
5   return (
6     <div className="App">
7       <header className="App-header">
8         <img src={logo} className="App-logo" alt="logo" />
9         <p>
10           Edit <code>src/App.js</code> and save to reload.
11         </p>
12         <a
13           className="App-link"
14           href="https://reactjs.org"
15           target="_blank"
16           rel="noopener noreferrer"
17         >
18           Learn React
19         </a>
20       </header>
21     </div>
22   );
23 }
24 export default App;
25
```

리액트를 시작하면 왼쪽과 같은 코드를 볼 수 있습니다.

HTML 기본 코드에서 봤던 비슷한 단어들을 많이 볼 수 있습니다.

div, a 태그 같은 것들은 HTML에서도 봤던 것이죠.

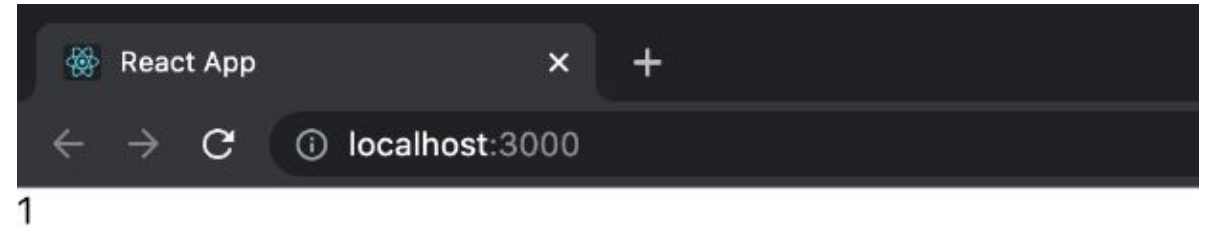
React는 이러한 기존 기능들 뿐 아니라 사용자가 원하는 태그/기능들을 쉽게 원하는 형태로 만들고 웹페이지에 사용할 수 있게 도와줍니다.

우리가 보는 화면은 바로 함수(여기서는 App)가 return 하는 것이고, 보통 리액트에서는 이러한 문법을 JSX라 부릅니다.

리액트에서는 이런 화면에 무언가를 보여주는 함수 또는 변수들을 컴포넌트라고 부릅니다.



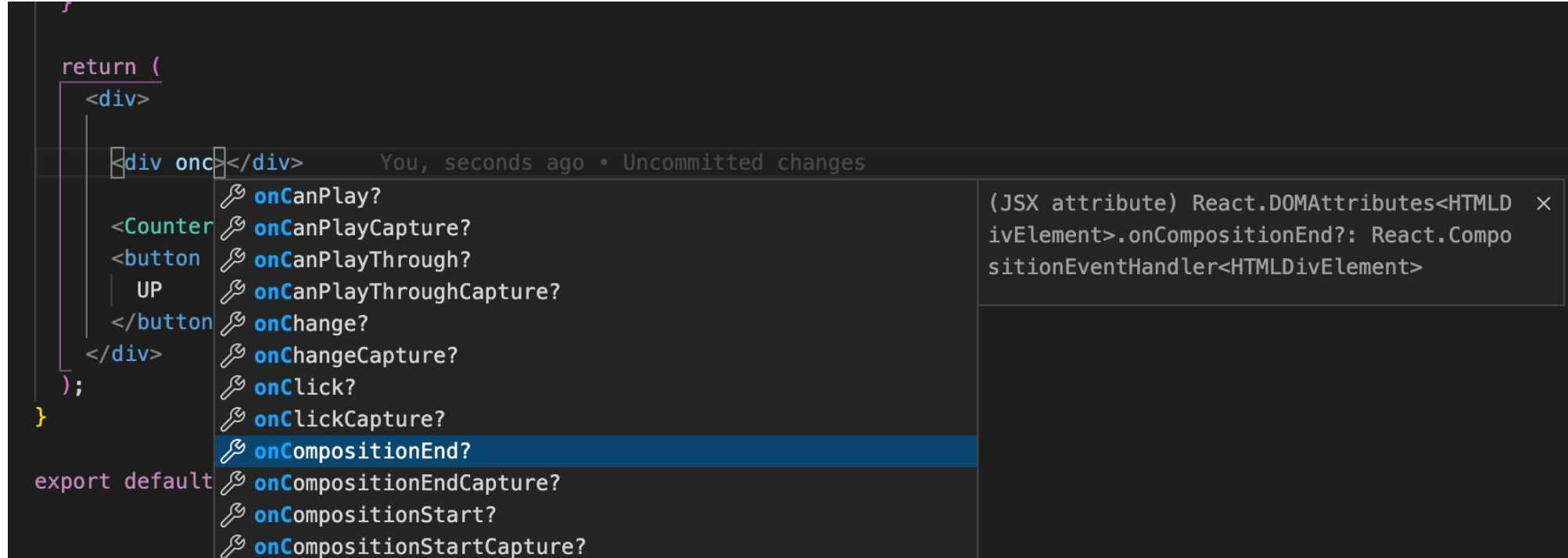
```
src > JS App.js > ...
You, seconds ago | 1 author (You)
15 import logo from './logo.svg';
14 import './App.css';
13
12 function Counter() {
11   return (
10     <div>1</div>
9   )
8 }
7 function App() {
6   return (
5     <div>
4       <Counter />
3     </div>
2   );
1 }
16 You, 12 minutes ago • Initialize project using Create R
1 export default App;
2
```



새로운 컴포넌트를 직접 만들어 봅시다.

원래 있던 코드들을 지우고, 왼쪽의 코드를 입력하고 확인해봅시다.

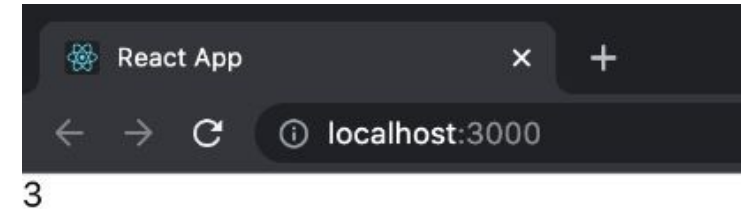
Counter 컴포넌트는 단순히 `<div>1</div>`를 우리가 편하게 사용할 수 있도록 이름을 바꿨을 뿐입니다.



**Prop**은 우리가 기존 **HTML** 태그에서의 속성 또는 리액트에서는 컴포넌트에 줄 수 있는 속성 값입니다.

**img** 태그에서 **src** 속성, **div**의 **background color** 처럼 **prop**을 통해서 우리가 원하는 대로 컴포넌트의 성질을 지정해줄 수 있습니다.

```
src > JS App.js > ...
You, seconds ago | 1 author (You)
15 import logo from './logo.svg';
14 import './App.css';
13
12 function Counter({number}) {
11   return (
10     <div>{number}</div>
9   )
8 }
7 function App() {
6   return (
5     <div>
4       <Counter number={3}/>
3     </div>
2   );
1 }
16 You, 13 minutes ago • Initialize project using
1 export default App;
2
```



직접 Prop을 하나 만들어 봅시다.

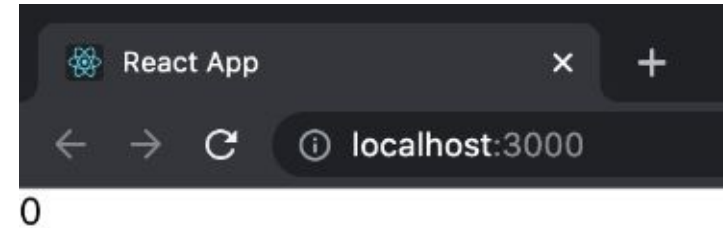
Counter 컴포넌트에 number라는 prop을 만듭니다.

코드를 작성하면 Counter에 number라는 prop을 만들 수 있습니다.

3이라는 값을 주었더니 화면에 3이 나타나는 것을 알 수 있습니다.

Counter(**{number}**)의 중괄호에 주의해주세요.

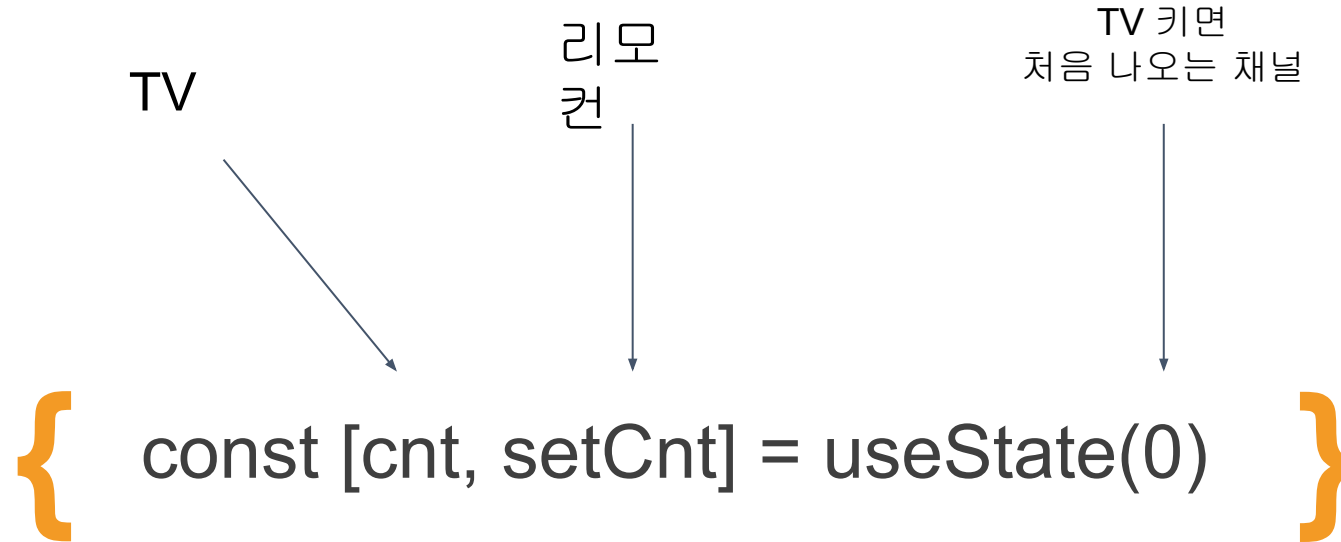
```
You, seconds ago | 1 author (You)
11 import logo from './logo.svg';
10 import './App.css';
9 import { useState } from 'react';
8
7 function Counter({number}) {
6   return (
5     <div>{number}</div>
4   )
3 }
2 function App() {
1   const [cnt, setCnt] = useState(0)
2   You, seconds ago • Uncommitted changes
1   return (
2     <div>
3       <Counter number={cnt}/>
4     </div>
5   );
6 }
7
8 export default App;
9
```



이제 변하는 값을 화면에 표시해봅시다.

React에서는 변하는 값을 쉽게 조작하고 화면에 표시할 수 있게 State라는 기능을 제공 하고 있습니다.

useState를 사용하면 값 변수와, 값을 조작할 수 있는 함수를 얻을 수 있습니다.



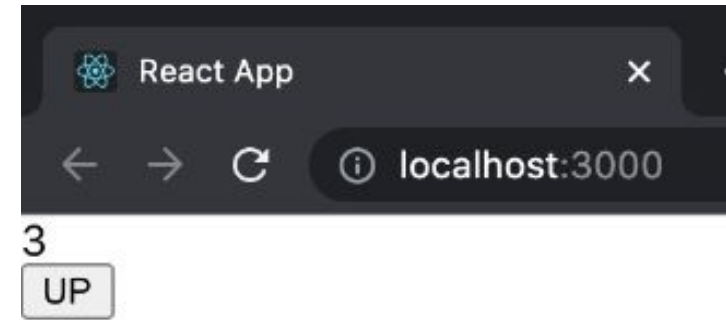
백화점에서 TV를 구매하면 리모컨과 TV를 같이 받게 됩니다.

우리는 TV를 통해서 화면을 보지만, TV 채널은 리모컨을 통해 변경합니다.

`useState`를 사용하면 왼쪽에 TV, 오른쪽에 리모컨을 받게 됩니다.

그리고 리엑트는 이런 변수(채널)이 변하면 자동으로 유저에게 최신 화면을 보여줍니다.

```
15 import logo from './logo.svg';
14 import './App.css';
13 import { useState } from 'react';
12
11 function Counter({number}) {
10   return (
9     <div>{number}</div>
8   )
7 }
6 function App() {
5   const [cnt, setCnt] = useState(0)
4
3   function onPressButton() {
2     setCnt(cnt+1)
1   }
16 | You, seconds ago • Uncommitted changes
1   return (
2     <div>
3       <Counter number={cnt}/>
4       <button onClick={onPressButton}>
5         UP
6       </button>
7     </div>
8   );
9 }
10
11 export default App;
12
```



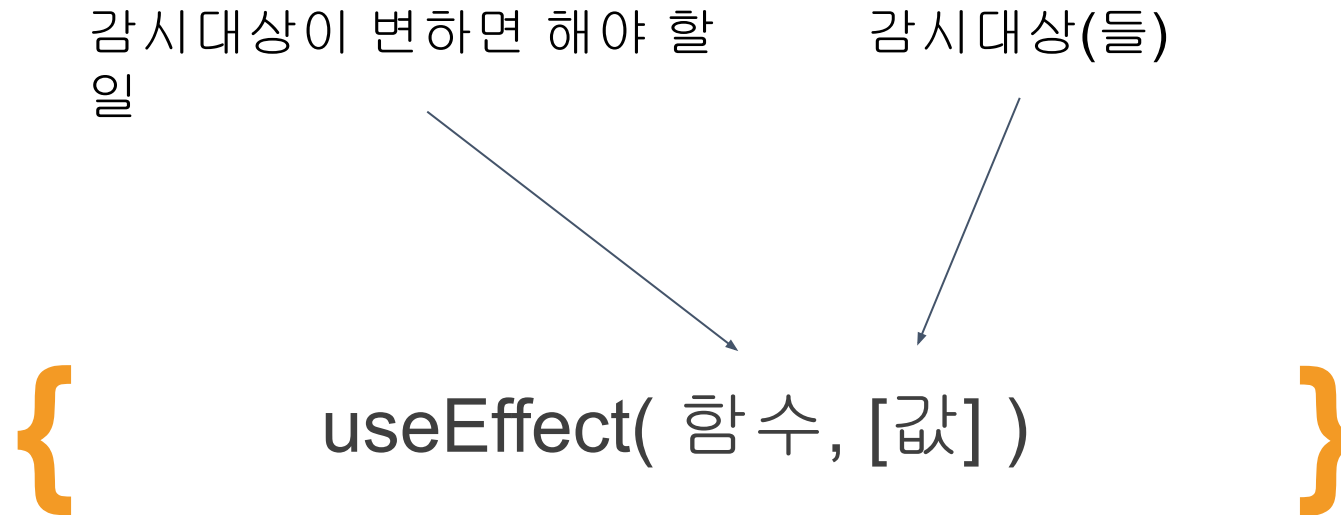
이제 버튼을 눌렀을 때 숫자가 1씩 올라가도록 해봅시다.

코드를 작성하고 버튼을 눌러 보세요.

button onPress에 준 onPressButton 함수를 보세요.

setCnt(cnt+1)을 이용해서 cnt값을 변경시켜 주는 것을 확인할 수 있습니다.

리모컨을 이용해서 tv 채널을 바꿨더니 tv 화면이 바뀌었네요.



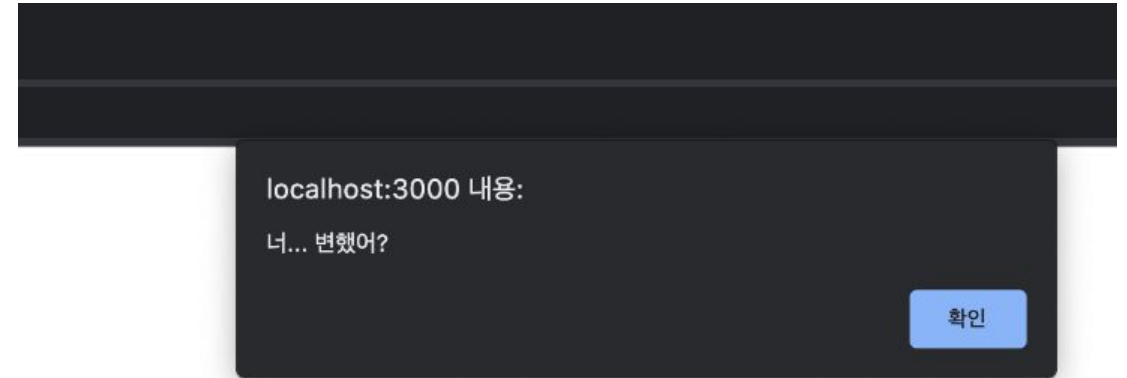
리액트에서 자주 사용하는 함수 중 하나인 **useEffect**를 살펴 봅시다.

**useEffect**를 이용하면 내가 원하는 변수의 값이 바뀌었을 때 원하는 함수를 실행할 수 있습니다.

마치 스파이처럼, 감시대상의 상태(값)이 변하면

**useEffect**는 배정 받은 일을 시작합니다.

```
You, a minute ago | 1 author (You)
19  ✓ import logo from './logo.svg';
18  import './App.css';
17  | import { useState, useEffect } from 'react';
16
15  ✓ function Counter({number}) {
14  |   return (
13  |     <div>{number}</div>
12  |   )
11  | }
10  ✓ function App() {
9   |   const [cnt, setCnt] = useState(0)
8
7   ✓ function onPressButton() {
6   |   setCnt(cnt+1)
5   | }
4
3   ✓ useEffect(() => {
2   |   alert('너... 변했어?')
1   | }, [cnt])
20  You, seconds ago • Uncommitted changes
```



코드를 작성하고 버튼을 눌러봅시다.

cnt의 값이 변경될 때마다 **useEffect**에 들어 있는 함수가 실행되는 것을 알 수 있습니다.

감시대상이 없다면 어떻게 될까요?

그러면 **useEffect**는 최초에 한 번만 함수를 실행하게 됩니다. 그래서 앱의 상태를 초기화할 때 많이 이용됩니다.