

이벤트

클릭 이벤트

src/App.css 수정

```
button { font-size: 12pt; padding: 4px 20px; margin: 8px; }
```

src/App.jsx 수정

```
import React from 'react';
import './App.css';

function App() {
  const onClickHandler = () => alert('버튼1 클릭');
  return (
    <div>
      <button onClick={onClickHandler}>버튼1</button>
      <button onClick={() => alert('버튼2 클릭')}>버튼2</button>
    </div>
  );
}

export default App;
```

이벤트 이름

HTML DOM 객체 이벤트 이름과 Virtual DOM 이벤트 이름이 약간 다르다.

예를 들어 클릭 이벤트의 경우 HTML에서는 onclick이지만,

Virtual DOM에서는 onClick이다.

Virtual DOM 이벤트 이름은 카멜 케이스(camel case)로 표기해야 한다.

이벤트 구현 방법

onClick 이벤트의 값은 자바스크립트 함수이어야 한다.

자바스크립트 함수는 당연히 자바스크립트 코드이므로 { } 괄호로 감싸야 한다.

```
const onClickHandler = () ⇒ alert('버튼1 클릭');
```

위와 같이 함수를 미리 구현한 후, 다음과 같이 지정해줄 수 있다

```
<button onClick={onClickHandler}>버튼1</button>
```

아니면 다음과 같이 바로 함수를 구현할 수도 있다.

```
<button onClick={() ⇒ alert('버튼2 클릭')}>버튼2</button>
```

변수값 변경

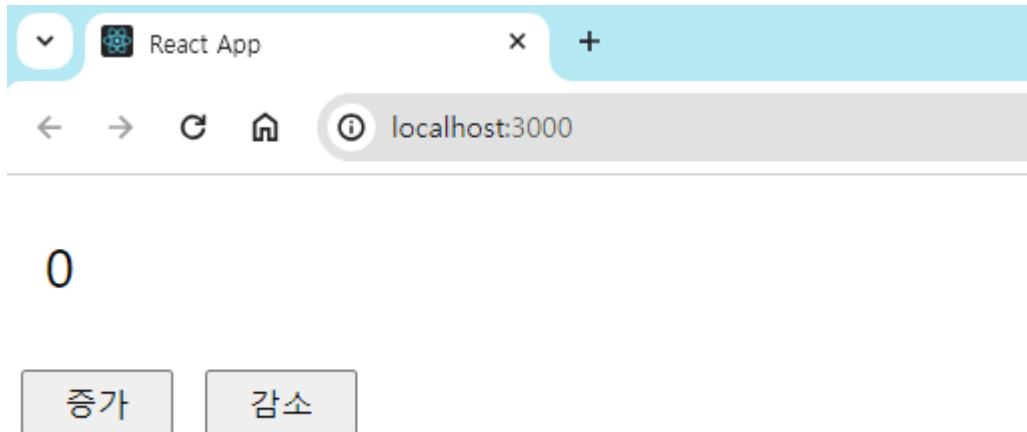
src/App.css 수정

```
button { font-size: 12pt; padding: 4px 20px; margin: 8px; }  
p { font-size: 20pt; padding: 0px 20px; }
```

src/App.jsx 수정

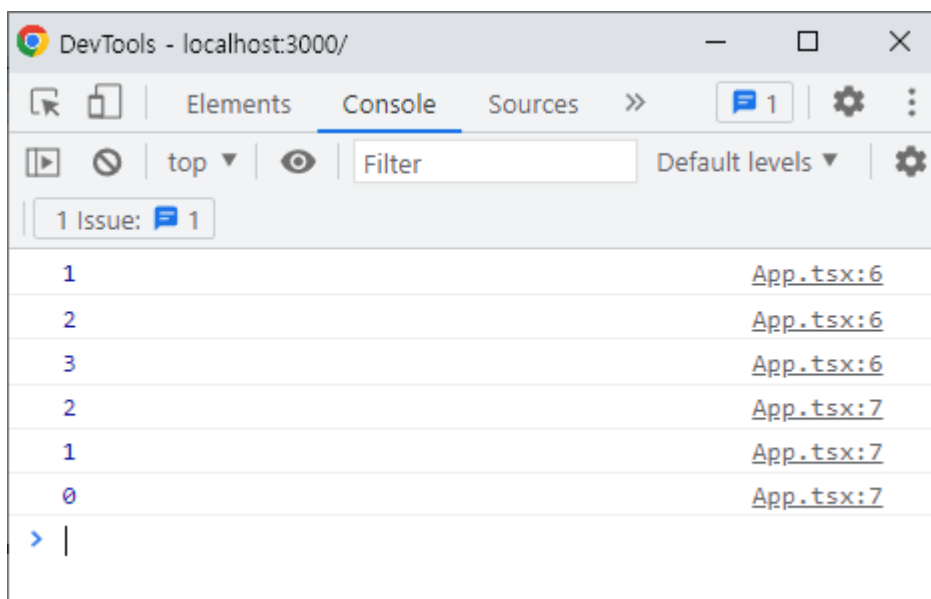
```
import React from 'react';  
import './App.css';  
  
function App() {  
  let count = 0;  
  const increase = () => { ++count; console.log(count); };  
  const decrease = () => { --count; console.log(count); };  
  return (  
    <div>  
      <p>{ count }</p>  
      <button onClick={increase}>증가</button>  
      <button onClick={decrease}>감소</button>  
    </div>  
  );  
}
```

```
export default App;
```



증가 버튼을 클릭하면 increase 함수가 호출되어 ++count 된다.
감소 버튼을 클릭하면 decrease 함수가 호출되어 --count 된다.

그리고 `<p>{ count }</p>` 부분에 count 변수값이 출력되어야 한다.
그런데 버튼을 클릭해되 count 변수값 출력에 변화가 없다.



개발자창의 console 탭을 열어보면, 버튼이 클릭될 때 마다 count 변수값이 변경되고 있음을 알 수 있다.

count 변수 값은 변경되었지만, 화면이 다시 그려지지 않은 것이다. (렌더링 되지 않는 것)
변수값이 변경되었다고 해서, 리액트 앱 화면이 다시 렌더링 되지는 않는다.
