

01

이벤트(Event)

01. 이벤트 핸들러 추가

```
function App(){
  const handleClick = () => {
    alert("버튼을 클릭했습니다!");
  };

  return <button onClick={handleClick}>클릭하세요!</button>;
};

export default App;
```

- HTML에서는 큰따옴표("")를 사용하여 문자열 형태로 이벤트 핸들러를 전달하지만, React에서는 문자열이 아닌 함수 형태의 객체를 prop으로 전달해야 한다.

01. 이벤트 핸들러 추가

■ 인라인 방식으로 익명함수나 화살표 함수의 정의로 전달

```
<button onClick={function handleClick(){  
  alert("버튼을 클릭했습니다!");  
}}>  
// 화살표 함수  
<button onClick={() => {  alert("버튼을 클릭했습니다!"); }}>
```

■ 잘못 사용한 예

```
<button onClick={handleClick()}> 확인 </button>
```

- 함수를 호출했기 때문에 Click이 일어날 때 실행되는 것이 아니라 해당 페이지가 렌더링 될 때마다 실행된다.

01. 이벤트 핸들러 추가

```
function ControlButton({ message, children }){
  return <button onClick={() =>
alert(message)}>{children}</button>;
};

function ControlPanel(){
  return (
    <>
      <ControlButton message="음악을 재생합니다.">▶ Play
    </ControlButton>
      <ControlButton message="재생을 중지합니다.">■ Stop
    </ControlButton>
    </>
  );
};

export default ControlPanel;
```

01. 이벤트 핸들러 추가

```
import ControlPanel from "./ControlPanel";

function App(){
  return <ControlPanel />;
};

export default App;
```

02

상태값(State)

01. 상태 값(State)

- 가상 돔(Virtual DOM)을 다시 렌더링하게 하는 값
- 상태 값은 컴포넌트에 소유된다.
- 상태 값이 변경되면 그 상태 값을 소유한 컴포넌트는 다시 렌더링된다.
- `useState()`라는 HOOK으로 생성한다.
- 상태 값과 상태 값을 변경하는 함수로 구성된다.
- `const [state, setState]=useState("State초기값");`
 - `State` : 상태 값이 저장되는 변수
 - `setState` : 상태 값을 변경하는 함수
 - 상태 값은 `setState()`함수에 의해서만 변경된다.
- 상태 값의 변경은 비동기 통합처리된다.

02. 상태 값(State) 사용하기

```
import { useState } from "react";

function Counter(){
  // 0을 초기값으로 하는 state와 setState() 함수 생성
  const [state, setState] = useState(0);

  return (
    <div>
      <h1>State 값 : {state}</h1>
      /* setState() 함수를 사용하여 state의 값을 1씩 증가시킴 */
      <button onClick={() => setState(state + 1)}>1씩 증가
      </button>
    </div>
  );
};

export default Counter;
```


02. 상태 값(State) 사용하기

```
const [count, setCount] = useState(0);  
  
<button onClick={() => setState(state + 1)}>Click me  
</button>
```

```
const [area, setArea] = useState(  
  { left: 0, top: 0, width: 200, height: 100 });  
  
setArea({ left: 50, top: 0, width: 200, height: 100 });
```

```
const [position, setPosition] = useState(  
  { left: 0, top: 0 });  
const [size, setSize] = useState(  
  { width: 200, height: 100 });
```