

7. State와 Props

≡ 다중 선택

react 입문

발생할 수 있는 문제

→ 관련 없는 부모 컴포넌트의 리렌더링으로 자식 컴포넌트도 리렌더링 되면서 성능이 저하될 수 있음

```
import './App.css';
import { useState } from 'react';

const Bulb = ({ light }) => {
  console.log(light); // ON/OFF 버튼, count 버튼 중 하나를 눌렀을
  return (
    <div>
      {light === "ON" ? (
        <h1 style={{ backgroundColor: "orange" }}>ON</h1>
      ) : (
        <h1 style={{ backgroundColor: "gray" }}>OFF</h1>
      )}
    </div>
  );
};

function App() {
  const [count, setCount] = useState(0);
  const [light, setLight] = useState("OFF");

  return (
    <>
      <div>
        {/* ON/OFF 버튼 클릭 시 조건에 맞게 ON/OFF 변경 */}
        <Bulb light={light} /> -> 자식 컴포넌트
        <button
          onClick={() => {
            setLight(light === "ON" ? "OFF" : "ON");
          }}
        />
      </div>
    </>
  );
}
```

```

    }}
  >
    {light === "ON" ? "끄기" : "켜기"}
  </button>
</div>
<div>
  { /* 버튼을 누를 때마다 h1 안의 숫자 +1 */ }
  <h1>{count}</h1>
  <button
    onClick={() => {
      setCount(count + 1);
    }}
  >
    버튼
  </button>
</div>
</>
);
}

export default App;

```

리렌더링이 되는 세 가지 상황

1. 자신이 관리하는 state의 값이 변경될 때
2. 제공받는 props의 값이 변경될 때
3. 부모 컴포넌트가 리렌더링 될 때 (자식 컴포넌트도 리렌더링 됨)
 - 성능 저하 요인이 될 수 있음(불필요한 렌더링 많아질 수 있기 때문)
 - 해결방법 : 컴포넌트 분리

컴포넌트 분리하기

```
// Bulb.jsx
```

```

import { useState } from "react";

const Bulb = () => {
  const [light, setLight] = useState("OFF");

  return (
    <div>
      {light === "ON" ? (
        <h1 style={{ backgroundColor: "orange" }}>ON</h1>
      ) : (
        <h1 style={{ backgroundColor: "gray" }}>OFF</h1>
      )}
      <button
        onClick={() => {
          setLight(light === "ON" ? "OFF" : "ON");
        }}
      >
        {light === "ON" ? "끄기" : "켜기"}
      </button>
    </div>
  );
};

export default Bulb;

```

```

// Count.jsx

import { useState } from "react";

const Count = () => {
  const [count, setCount] = useState(0);

  return (
    <div>
      {/* 버튼을 누를 때마다 h1 안의 숫자 +1 */}
      <h1>{count}</h1>
      <button
        onClick={() => {

```

```

        setCount(count + 1);
      }}
    >
    버튼
  </button>
</div>
);
};

export default Count;

```

```

// App.jsx

import './App.css';
import Bulb from './components/Bulb';
import Count from './components/Count';

function App() {
  return (
    <>
      <Bulb />
      <Count />
    </>
  );
}

export default App;

```