

# useEffect 훅

## 설명

## 호출 형태 #A: 상태 변수

### Counter.js 생성

```
import React, { useEffect, useState } from "react";
import "./Counter.css";

function Counter() {
  const [count1, setCount1] = useState(0);
  const [count2, setCount2] = useState(0);
  useEffect(() => console.log('useEffect #1', count1), [count1]);
  useEffect(() => console.log('useEffect #2', count2), [count2]);
  return (
    <div className="Counter">
      <h1>Counter</h1>
      <div>
        <span>{ count1 }</span>
        <button onClick={e=>setCount1(count1+1)}>++</button>
      </div>
      <div>
        <span>{ count2 }</span>
        <button onClick={e=>setCount2(count2+1)}>++</button>
      </div>
    </div>
  );
}
```

```
export default Counter;
```

```
useEffect(() => console.log('useEffect #1', count1), [count1])
```

`count1` 값이 직전과 달라졌을 때에만, 노란색 함수가 호출된다.

```
useEffect(() => console.log('useEffect #2', count2), [count2])
```

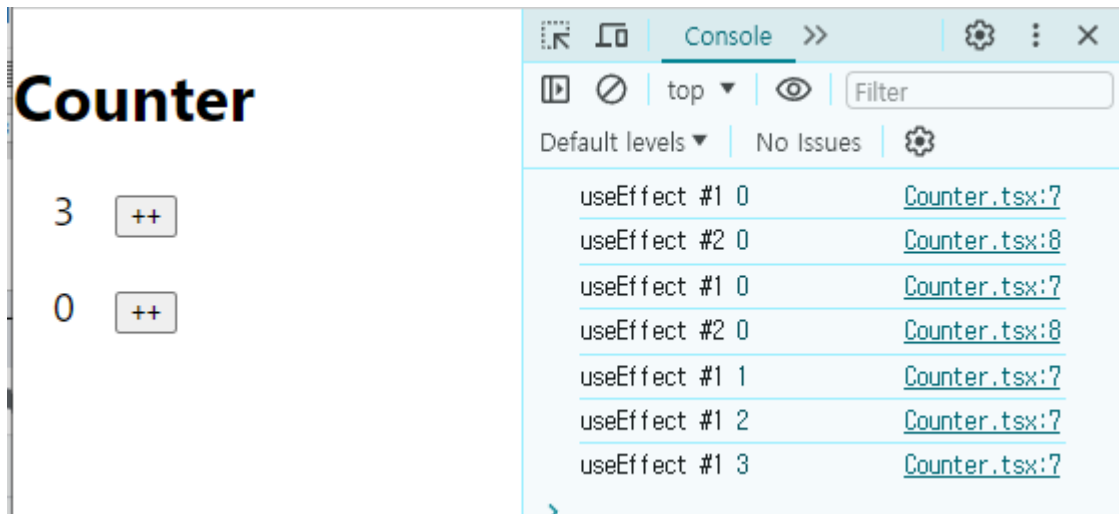
`count2` 값이 직전과 달라졌을 때에만, 노란색 함수가 호출된다.

## Counter.css

```
.Counter span { font-size: 15pt; margin-right: 20px; }  
.Counter div { margin: 20px; }
```

## App.tsx

```
import React from "react";  
import Counter from "./Counter";  
  
function App() {  
  return <Counter />  
}  
  
export default App;
```



처음에는 두 콜백함수가 모두 호출되었지만,  
그 다음부터 count1 상태 값만 변경되니, 첫 번째 콜백함수만 호출되고 있다.

## 호출 형태 #B: mount

### Counter.js 수정

```
import React, { useEffect, useState } from "react";
import "../Counter.css";

function Counter() {
  const [count1, setCount1] = useState(0);
  const [count2, setCount2] = useState(0);
  useEffect(() => console.log('useEffect #1', count1), [count1]);
  useEffect(() => console.log('useEffect #2', count2), [count2]);
  useEffect(() => console.log('mount'), [])
  return (
    <div className="Counter">
      <h1>Counter</h1>
      <div>
        <span>{ count1 }</span>
        <button onClick={e=>setCount1(count1+1)}>++</button>
      </div>
    </div>
  )
}
```

```

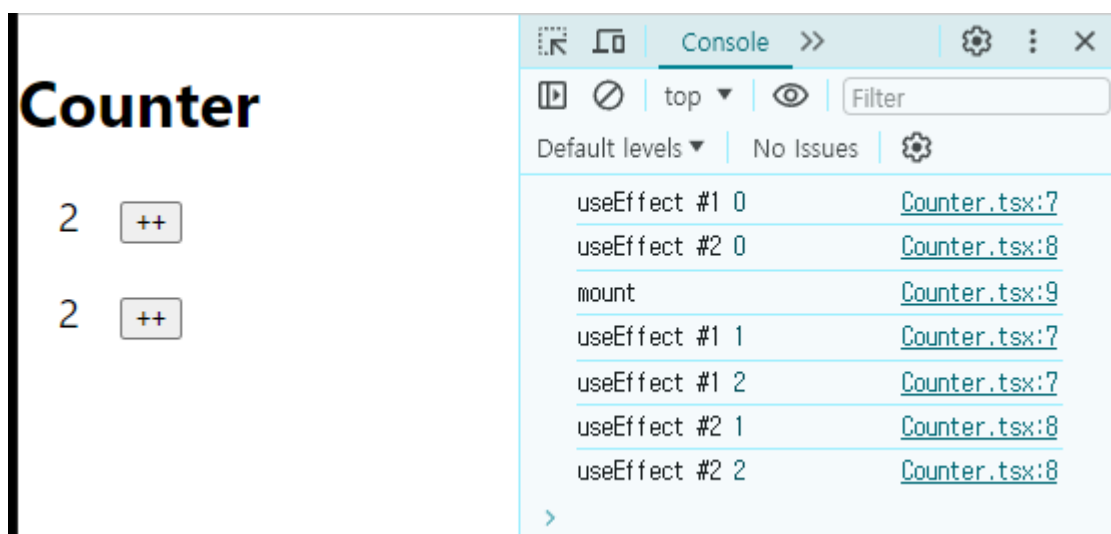
    </div>
    <div>
      <span>{ count2 }</span>
      <button onClick={e=>setCount2(count2+1)}>++</button>
    >
  </div>
</div>
);
}

export default Counter;

```

**useEffect(() => console.log('mount'), [])**

컴포넌트가 마운트될 때, 노란색 함수가 호출된다.



컴포넌트가 마운트될 때, 한 번 콜백함수가 호출되고 있다.

## 호출 형태 #B: mount, unmount

### Counter.js 수정

```

import React, { useEffect, useState } from "react";
import "./Counter.css";

```

```

function Counter() {
  const [count1, setCount1] = useState(0);
  const [count2, setCount2] = useState(0);
  useEffect(() => console.log('useEffect #1', count1), [count1]);
  useEffect(() => console.log('useEffect #2', count2), [count2]);
  useEffect(() => {
    console.log('mount')
    return () => console.log('unmount')
  }, [])
  return (
    <div className="Counter">
      <h1>Counter</h1>
      <div>
        <span>{ count1 }</span>
        <button onClick={e=>setCount1(count1+1)}>++</button>
      </div>
      <div>
        <span>{ count2 }</span>
        <button onClick={e=>setCount2(count2+1)}>++</button>
      </div>
    </div>
  );
}

export default Counter;

```

```

useEffect(() => {
  console.log('mount')
  return
  () => console.log('unmount')
}, [])

```

컴포넌트가 마운트될 때, 노란색 함수가 호출된다.  
이 함수가 리턴하는 것은  
`() => console.log('unmount')` 함수이다.  
리턴된 함수는, 컴포넌트가 언마운트될 때 호출된다.

## App.js 수정

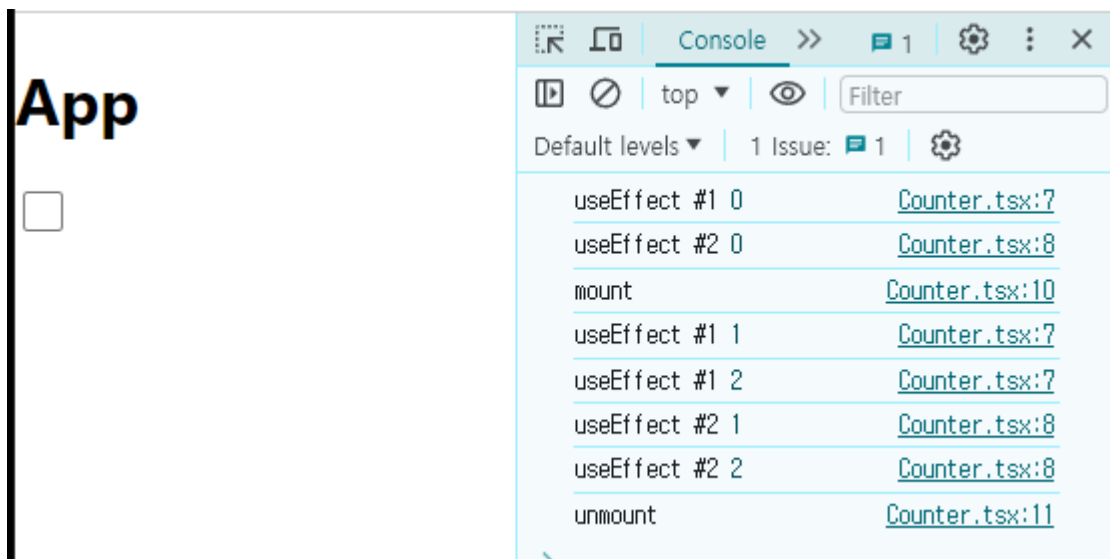
```
import React, { useState } from "react";
import Counter from "./Counter";

function App() {
  const [visible, setVisible] = useState(true);
  return (
    <div>
      <h1>App</h1>
      <input type='checkbox' checked={visible} style=
{{height: 20, width: 20}}
        onChange={e=>setVisible(e.target.c
hecked)} />
      { visible ? <Counter /> : null }
    </div>
  );
}

export default App;
```

컴포넌트가 처음 만들어진 때 => mount  
컴포넌트가 화면에서 사라질 때 => unmount

App 컴포넌트의 visible 상태 값이 true일 때, Counter 컴포넌트가 마운트되고  
false일 때, Counter 컴포넌트가 언마운트된다



## 호출 형태 #C: update

### Counter.js

```
import React, { useEffect, useState } from "react";
import "../Counter.css";

function Counter() {
  const [count1, setCount1] = useState(0);
  const [count2, setCount2] = useState(0);
  useEffect(() => console.log('useEffect #1', count1));
  useEffect(() => console.log('useEffect #2', count2));
  return (
    <div className="Counter">
      <h1>Counter</h1>
      <div>
        <span>{ count1 }</span>
        <button onClick={e=>setCount1(count1+1)}>++</button>
      </div>
      <div>
        <span>{ count2 }</span>
        <button onClick={e=>setCount2(count2+1)}>++</button>
      </div>
    </div>
  );
}
```

```

>
    </div>
  </div>
);
}

export default Counter;

```

```

useEffect(() ⇒ console.log('useEffect #1', count1));
useEffect(
() ⇒ console.log('useEffect #2', count2));

```

Counter 컴포넌트가 렌더링 될 때 마다, 노란색 함수들이 호출된다.  
호출형태 #1

useEffect로 등록한 두 콜백 함수들이,  
Counter 컴포넌트가 렌더링 될 때 마다 호출되고 있다.