

DOM 과 React 프로그래밍 비교 분석하기 : Remove-Mount

제출 파일 : **mount.html**, **Mount.jsx**

참조 파일 : **removeRestore.html**, **Conditional.jsx**

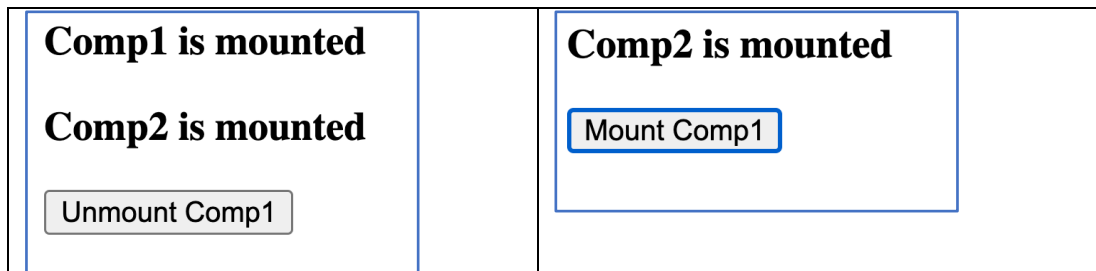
1. 과제-1 : Mount.jsx 에는 다음과 같은 세 개의 컴포넌트가 포함되어 있다.

```
const Mount = () => {  
  const [toggle, setToggle] = useState(true)  
  // 이곳을 채우시오  
}  
  
const Comp1 = () => <h3>Comp1 is mounted</h3>  
const Comp2 = () => <h3>Comp2 is mounted</h3>
```

이 결과 display 되는 모습은 다음과 같이 두 가지이다.

(a) 상태 변수 toggle 이 true 인 경우

(b) 상태 변수 toggle 이 false 인 경우



<button> 을 클릭할 때마다 Comp1 컴포넌트가 Unmount/Mount 되며, <button>의 값도 “Umount Comp1” / “Mount Comp1” 으로 변화한다. 이렇게 동작하는 **Mount.jsx** 파일을 완성하라.

2. 과제 - 2 :

위의 React 코드와 동일하게 동작하는 html 파일 **mount.html** 을 작성하라.

3. 주목할 사항들

- JSX 내에서 if-then, if-then-else 형태의 표현 방법 : Conditional.jsx 참조
- DOM 에서 subtree 를 제거하고/복원하는 방법 : removeRestore.html 참조
- 명령형 프로그래밍 : DOM 트리의 특정 주소를 찾은 다음 removeChild 를 이용하여 특정 부분을 remove 시킴
- 함수형 프로그래밍 : 상태 변수 toggle 의 값이 바뀌면 이 변수가 소속된 Mount 컴포넌트는 re-rendering 을 위해서 다시 실행된다. 이 과정에서 toggle

의 값에 따라서 Comp1 컴포넌트를 호출하거나 호출하지 않는다. 호출하지 않는 경우, 원래 Mount 되어 있는 Comp1 컴포넌트는 unmount 되게 된다.

- DOM 프로그래밍에서는 서브 트리(DOM element) 를 remove 한다고 하며, React 프로그래밍에서는 컴포넌트 (React element)를 unmount 한다는 표현을 사용한다. DOM 에서는 append/insert 라는 표현을, React 에서는 mount 라는 표현을 사용한다.