

2. useEffect 사용하기

☰ 다중 선택

라이프사이클

useEffect

- 리액트 컴포넌트의 사이드 이펙트를 제어하는 새로운 React Hook
- 사이드 이펙트
 - 우리말로 부작용이라는 뜻
 - 리액트에서는 **부수적인 효과** 또는 **파생되는 효과**로 해석할 수 있음

1. 기본 사용법

```
useEffect(effectFunction, dependencyArray);  
// 예시 : count state 값이 바뀔 때마다 첫번째 인수로 전달한 콜백함수 실행  
useEffect(() => {  
  console.log(count);  
}, [count]);
```

- **effectFunction** : 실행할 코드(함수)를 작성
- **dependencyArray** : 어떤 상태/값이 변경될 때 이펙트를 실행할지 명시하는 배열 → 의존성 배열 → **deps**
 - 빈 배열 (**[]**)
 - 이펙트는 한 번만 실행됩니다. (마운트 시)
 - **componentDidMount** 와 유사한 동작
 - 배열 생략
 - 상태/props가 변경될 때마다 실행됩니다.
 - **componentDidUpdate** 와 유사
 - 특정 값 포함 (**[value]**)

- `value`가 변경될 때만 이펙트가 실행됨
 - 의존성 관리가 중요.
- 여러 값 넣어도 됨 `[count, input]` 이런 식으로 → 둘 중 하나가 바뀌게 되면 콜백함 수 실행
- 왜 굳이 `useEffect`를 쓰는 걸까? 이벤트 핸들러 사용하면 안돼?
→ 비동기로 업데이트가 되기 때문

사용 시 주의사항

1. 의존성 배열 정확히 관리

- 의존성 배열에 포함하지 않으면 최신 상태/`props`를 참조하지 못할 수 있음.
- ESLint의 `react-hooks/exhaustive-deps` 규칙을 활성화해 관리.

2. 클린업 함수 중요

- 리소스 낭비를 방지하려면 이벤트 리스너, 타이머 등을 정리해야 함.
- 예: WebSocket 연결 해제, 구독 취소 등.