Hooks

💡 Hooks 등장 배경

React에서 컴포넌트를 구성하는 방식은 Class형, 함수형 2가지 입니다. Class형이 먼저 나왔으며, Class형에서는 LifeCycle API와 state를 사용할 수 있었으나 함수형에서는 이를 사용할 수 없었습니다.

지금 잠시만 생각해봐도 이 2개 없이는 개발을 못할 정도이니 함수형은 거의 사용하지 않는 추세였으나, 함수형에서도 이를 사용하기 위해 React 16.8 버전에서 Hook이라는 개념이 등장하게 되었습니다.

대표적으로 useState, useCallback, useEffect 등이 있으며, Hook의 등장으로 최근엔 대부분 함수형 컴포넌트를 사용하고, 정말 과거에 이미 Class형으로 만들어진 컴포넌트들만 이 리팩토링을 거치지 않는다면 그대로 쓰이고 있습니다.

💡 Hooks 종류

- **useState**
  - **함수형 컴포넌트에서도 가변적인 상태를 지니고 있을 수 있게 해줍니다.**
  - 기본적으로 **비동기**로 동작합니다.

- **useEffect**
  - **컴포넌트가 렌더링 될 때마다 특정 작업을 수행하도록 설정할 수 있는 Hook** 입니다.
  - 이때, **컴포넌트가 mount 될 때, 특정 값이 업데이트 될 때, 컴포넌트가 unmount 될 때 특정 작업**을 할 수 있습니다.

- **mount 될 때**, 실행할 내용은 useEffect의 cb 함수 안에 그냥 넣으면 됩니다.
- **특정 값이 업데이트 될 때** 특정 작업을 하려면, cb 함수 뒤에 배열 형식으로 **dependancyArray를 작성하면 배열 내 값이 변경될 때마다 cb 함수가 실행**됩니다.
- **unmount 될 때**, 실행될 내용은 cb 함수 안에 return 값으로 함수를 넘겨주면 해당 함수가 unmount될 때 실행되며, 이 함수를 **clean up 함수**라고 합니다.

- **useReducer**
  - **useReducer는 useState보다 컴포넌트에서 더 다양한 상황에 따라 다양하게 상태를 업데이트 해주고 싶을 때 사용**합니다. Reducer라는 개념을 사용하며, **Redux에서의 Reducer와 동일하게 state와 action에 따라 동작**합니다.
  - Redux와 다른 점은 **useReducer에서의 action에는 type을 꼭 가지고 있을 필요는 없습니다.**
  - **reducer 함수를 만들고 이를 useReducer의 첫번째 인자, 상태의 초기값을 두 번째 인자로 넣어주면 됩니다.**

- **useMemo**
  - **컴포넌트 내부에서 발생하는 연산을 최적화 할 때 사용**합니다. 기본적으로 **memoization 기법을 사용하는 함수**이기 때문에 **메모리에 값을 저장해두고 특정 경우에만 메모리의 값을 업데이트** 하고, 그 이외에는 메모리에 저장된 값을 그대로 사용합니다.
  - 기본적으로 useEffect와 사용법은 같아 useMemo안에 cb함수와 dependancyArray가 들어갑니다.
  - dependancyArray안의 값이 변경되었을 때만 cb 함수의 return 을 실행시켜 메모리를 업데이트 해주며, dependancyArray안에 아무것도 없다면 처음 mount 시에만 메모리를 업데이트 해주고, 이후에는 메모리에 저장된 값을 그대로 사용합니다.

- **useCallback**
  - useMemo와 상당히 비슷합니다. 주로 **렌더링 성능을 최적화해야 하는 상황에서 사용**하는데, useMemo와 다른 점은 주로 useMemo는 변수를 재사용하기 위해 주로 사용되고, **useCallback은 함수를 재사용하기 위해 사용**된다는 것입니다.

    - 기본적으로 **useCallback을 사용하지 않으면 컴포넌트가 리렌더링 될 때마다 해당 함수를 재생성** 합니다. 하지만 useCallback을 이용하면 **특정 경우에만 해당 함수를 생성**해줍니다.
    - 사용방법은 useEffect, useMemo와 완벽히 동일하며, **dependancyArray에 값이 없으면 최초 렌더링 시에만 함수를 생성**하고, **값이 있으면 해당 변수가 변경되었을 때 함수를 재생성**합니다.

- **useRef**
    - **함수형 컴포넌트에서 ref attribute를 쉽게 사용할 수 있게 해줍니다.** useRef를 사용해 ref를 설정해주면 **useRef를 사용해 만든 객체 안의 current 값이 실제 엘리먼트를 가리키게 됩니다.**