생명 주기 함수 (각 컴포넌트 기준)

3단계의 생명 주기에 따라 동작할 내용을 정의함

- Mount : 화면에 첫 렌더링 (componentDidMount)
- Update : re-rendering (componentDidUpdate)
- Unmount: 화면에서 사라질 때 (componentWillUnmount)

Class 컴포넌트의 생명주기 함수의 기능에 해당함

- componentDidMount : 처음 1회 실행
- componentDidUpdate : 반복 실행 가능
- componentWillUnmount : rendering에서 해당 컴포넌트가 제외될 때, cleanup

useEffect 함수 (side-effect function)

- `useEffect(<function>, <dependency>)`
  - `<dependency>` 는 optional

- side-effect 함수로서 동작 (no return values)
  - return 이 있더라도 그 의미는 일반 함수의 return과는 다름
  - 외부로부터 데이터 가져오기, DOM을 직접적으로 update, 타이머 실행 등

useEffect로 처리되는 생명주기 기능

1. Dependency가 정의되지 않을 경우 callback 함수 실행
   1. 컴포넌트가 맨 처음 rendering 될 때 (componentDidMount)
   2. 컴포넌트가 re-rendering 될 때 (componentDidUpdate, 반복적으로 실행될 수 있음)
   3. Dependency 변수를 표시하지 않는 경우는 모든 상태 변수를 dependency에 명시한 것과 동일(?)

```
useEffect(() => {
  // 작업 ...
})
```

2. Dependency가 정의되는 경우 callback 함수 실행
   1. 컴포넌트가 맨 처음 rendering될 때 (componentDidMount)
   2. 지정된 상태 변수가 변경될 때마다 callback 실행 (componentDidUpdate, 반복 실행 가능)

   (배열이 empty 인 경우 callback은 오직 1번만 실행, componentDidMount)

```
useEffect(() => {
  // 작업 ...
}, [var1, var2, ... varn] ) // dependency array
```

- Unmount 될 때 (componentWillUnmount) : return으로 표현됨
  - 해당 컴포넌트가 디스플레이에서 사라질 때 (즉, 선택되지 못함)
  - 이 return은 일반 함수의 return과는 의미가 다름
  - `let res = useEffect(()=>{ … return ()=>{…}},[vars])` 형태로 사용되지 않음

```
useEffect(() => {
  // subscribe ...

  return() => {
     // cleanup subscription
  }

}, [var1, var2, ... varn] )
```