

K-Digital Training KDT 풀스택 웹 개발자 양성 부트캠프 3기

React

WITH 팀 리처드



Hooks



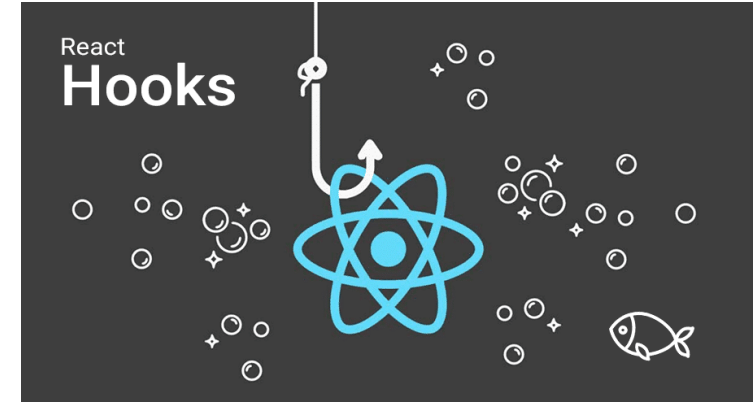
Hook란?

근데 Hook이 뭔가요?

Hook은 함수 컴포넌트에서 React state와 생명주기 기능(lifecycle features)을 “연동(hook into)”할 수 있게 해주는 함수입니다. Hook은 class 안에서는 동작하지 않습니다. 대신 class 없이 React를 사용할 수 있게 해주는 것입니다. (하지만 이미 짜놓은 컴포넌트를 모조리 재작성하는 것은 권장하지 않습니다. 대신 새로 작성하는 컴포넌트부터는 Hook을 이용하시면 됩니다.)

<https://ko.reactjs.org/docs/hooks-overview.html>

Hook 란?



- React 의 새로운 기능
- 클래스 컴포넌트에서만 가능했던

state(상태관리) 와 lifecycle(라이프사이클) 이 가능하도록 돕는 기능

클래스형 컴포넌트에서 사용할 수 있는 기능을
함수형 컴포넌트에서도 사용할 수 있도록 “낙아채는 기능”

Hook 사용 규칙

- 최상위 단계에서만 호출 가능
 - 최상위 컴포넌트 X
 - 반복문, 조건문, 중첩된 함수 내부에서 호출하면 안 되는 것!
- Hook 은 오로지 **React 함수형 컴포넌트** 안에서만 호출 가능하다.

Hooks 종류

- `useState()`
- `useEffect()`
- `useRef()`

Hook – useState()

- 함수형 컴포넌트에서 **상태 관리**를 위해 사용하는 것
- 동적상태관리
- 가장 기본적인 Hook으로 함수형 컴포넌트에서 가변적인 상태를 지니게 해준다.
- 클래스형 컴포넌트에서의 State

Hook – useEffect()

- sideEffect 를 수행한다.
 - mount / unmount / update
- 클래스형 컴포넌트의 componentDidMount 와 componentDidUpdate 가 합쳐진 형태

Hook – useEffect()

```
useEffect(() => {  
  console.log( '렌더링이 완료되었습니다.' );  
  console.log( "name : ", name, " & email : ", email );  
})
```

- 컴포넌트가 **rendering** 될 때마다 특정 작업을 수행

Hook – useEffect()

```
useEffect (() => {  
    console.log("처음 렌더링될 때만 실행됨!!");  
}, []);
```

- **Mount 될 때만 실행된다.**
 - Mount : 최초 rendering

Hook – useEffect()

- 특정 값이 바뀔 때만 실행하게 하고 싶다면?

```
componentDidUpdate( prevProps, prevState ) {  
  if( prevProps.value !== this.props.value ) {  
    doSomething();  
  }  
}
```

```
useEffect ( () => {  
  console.log(name);  
}, [name]);
```

Hook – useEffect()

- 컴포넌트가 update 되기 직전에 작업을 수행하고 싶다면?

=> **cleanup(뒷정리 함수)** 이용하기

```
useEffect (() => {  
  console.log("effect");  
  console.log(name);  
  
  return () => {  
    console.log("cleanup");  
    console.log(name);  
  };  
});
```

Hook – useRef()

- 함수형 컴포넌트에서 ref를 사용하기 쉽게 만들어주는 Hook
- useRef() 는 인자로 받은 값으로 초기화된 변경 가능한 ref 객체를 반환한다.
- **ref.current** 로 현재 가리키는 객체에 접근할 수 있다.

```
const ref = useRef(initialValue);
```

```
const numberInput = useRef(null);
```