

# 리액트 훅 테스트 (feat. act 함수)

실무에 바로 적용하는 프론트엔드 테스트

이재성

# 리액트 훅

- ✓ React 16.8부터 추가됨
- ✓ 컴포넌트 사이의 로직을 재사용, 비즈니스 로직을 분리
  - ✓ class 없이 리액트 기능을 활용할 수 있게 함
- ✓ 이를 통해 컴포넌트와 비즈니스의 결합도를 낮출 수 있음

# act 함수

<https://legacy.reactjs.org/docs/test-utils.html#act>

# act 함수의 역할

- ✓ act 함수는 상호 작용(렌더링, 이펙트 등..)을 함께 그룹화하고 실행해 렌더링과 업데이트가 실제 앱이 동작하는 것과 유사한 방식으로 동작함
  - ✓ 즉, 테스트 환경에서 act를 사용하면 가상의 돔(jsdom)에 제대로 반영되었다는 가정하에 테스트가 가능해짐
- ✓ 컴포넌트를 렌더링한 뒤 업데이트 하는 코드의 결과를 검증하고 싶을 때 사용

# act 함수의 역할

- ✓ act 함수는 상호 작용(렌더링, 이펙트 등..)을 함께 그룹화하고 실행해 렌더링과 업데이트가 실제 앱이 동작하는 것과 유사한 방식으로 동작함
  - ✓ 즉, 테스트 환경에서 act를 사용하면  
가상의 돔(jsdom)에 제대로 반영되었다는 가정하에 테스트가 가능해짐
- ✓ 컴포넌트를 렌더링한 뒤 업데이트 하는 코드의 결과를 검증하고 싶을 때 사용

테스트 환경에서 컴포넌트 렌더링 결과를 jsdom에 반영하기 위해

**act 함수를 반드시 호출해야 함**

# React Testing Library와 act 함수

@testing-library/react, @testing-library/user-event

컴포넌트 렌더링, 이벤트 핸들러 실행



내부적으로 **act** 함수를 호출하여 상태를 반영한다

# 정리

## 리액트 훅

- 리액트 렌더링 메커니즘을 따른 단순 함수이기 때문에 독립적인 단위 테스트를 작성하기 적합함
- React Testing Library의 **renderHook API**를 사용해 편하게 검증할 수 있음

## act()

- 상호 작용(렌더링, 이펙트 등..)을 함께 그룹화하고 실행하여 실제 앱에서 동작하는 것처럼 렌더링과 업데이트 상태를 반영하도록 도와줌
- 테스트 환경에서 컴포넌트의 렌더링, 업데이트 결과를 jsdom에 반영할 때 사용해야 함
- React Testing Library의 render함수와 user-event는 내부적으로 act 함수를 호출하기 때문에 편리하게 테스트 코드 작성이 가능
- 이외에 별도로 리액트 state를 업데이트하여 변경 사항을 검증해야 한다면 act 함수를 사용하여 state를 반영해야 함