

React의 규칙

각 프로그래밍 언어마다 개념을 표현하는 고유한 방식이 있듯이, **React**에도 패턴 이해를 쉽게 하며 고품질의 애플리케이션을 만들 수 있게 하는 일종의 규칙 혹은 모범적인 방식이 있습니다.

- [컴포넌트와 Hook은 순수해야 합니다](#)
- [React가 컴포넌트와 Hook을 호출하는 방식](#)
- [Hook의 규칙](#)

📌 중요합니다!

React를 사용하여 UI를 표현하는 방법에 대해 더 알고 싶다면 [React로 사고하기](#)를 읽어 보는 것을 추천합니다.

이 섹션에서는 모범적인 React 코드를 작성하기 위한 규칙을 설명합니다. 모범적인 React 코드를 작성하면 애플리케이션을 체계적으로 조직하고 안전하며 쉽게 구성할 수 있습니다. 이러한 특성은 애플리케이션이 변화에 더 잘 대처할 수 있게 하고 다른 개발자나 라이브러리, 도구와의 협업을 더 원활하게 합니다.

이러한 규칙을 **React의 규칙**이라고 합니다. 이는 단순한 지침이 아니라 규칙으로, 이를 어길 경우 애플리케이션에 버그가 생길 가능성이 높으며 코드가 일반적이지 않게 변해 이해하기 어렵고 논리적으로 설명하기 힘들어집니다.

코드베이스가 React의 규칙을 따르도록 하기 위해 React의 [ESLint](#) 플러그인과 함께 [Strict Mode](#)를 사용하는 것을 강력히 권장합니다. React의 규칙을 따르면 버그를 찾아 해결할 수 있으며 애플리케이션의 유지 보수성을 높일 수 있습니다.

컴포넌트와 Hook은 순수해야 합니다

컴포넌트와 Hook의 순수성은 React의 주요 규칙으로, 이를 통해 앱이 예측 가능하고 디버깅이 쉬워지며 React가 코드를 자동으로 최적화할 수 있습니다.

- **컴포넌트는 멍등해야 합니다** - React 컴포넌트는 항상 입력 값(Props, State, Context)에 따라 동일한 출력을 반환한다고 가정합니다.
- **사이드 이펙트는 렌더링 외부에서 실행되어야 합니다** - **사이드 이펙트**는 React가 최상의 사용자 경험을 제공하기 위해 컴포넌트를 여러 번 렌더링할 수 있기 때문에 렌더링 중에 실행되어서는 안 됩니다.
- **Props와 State는 불변입니다** - 컴포넌트의 Props와 State는 단일 렌더링에 대한 불변의 **스냅샷**입니다. 절대 이를 직접 변경하지 마세요.
- **Hook의 반환값과 인수는 불변입니다** - 값이 Hook에 전달되면 이를 수정해서는 안 됩니다. JSX의 Props와 마찬가지로 Hook에 전달된 값도 불변입니다.
- **JSX로 전달된 값은 불변입니다** - JSX에 사용된 후에는 값을 변경하지 마세요. JSX가 생성되기 전에 변경을 수행하세요.

React가 컴포넌트와 Hook을 호출하는 방식

React는 사용자 경험을 최적화하기 위해 필요할 때마다 컴포넌트와 Hook을 렌더링합니다.

React는 선언적입니다. 즉 컴포넌트 로직에서 무엇을 렌더링할지 React에게 지시하면, React는 이를 사용자가 최적으로 볼 수 있도록 알아서 처리합니다.

- **컴포넌트 함수를 직접 호출하지 마세요** - 컴포넌트는 JSX에서만 사용해야 합니다. 일반 함수처럼 호출하지 마세요.
- **Hook을 일반 값으로 전달하지 마세요** - Hook은 반드시 컴포넌트 내부에서만 호출되어야 합니다. Hook을 일반 값처럼 전달하지 마세요.

Hook의 규칙

Hook은 자바스크립트 함수로 정의하지만, 호출 위치에 제약이 있는 특별한 유형의 재사용 가능한 UI 로직입니다. Hook을 사용할 때는 **Hook의 규칙**을 따라야 합니다.

- **Hook을 최상위 레벨에서만 호출하세요** - Hook을 반복문, 조건문, 또는 중첩된 함수 내부에서 호출하지 마세요. 대신 Hook을 항상 React 함수 최상위 레벨에서 호출하고, early return 이전

에 사용해야 합니다.

- [Hook을 React 함수에서만 호출하세요](#) - 일반 자바스크립트 함수에서 Hook을 호출하지 마세요.

다음

[컴포넌트와 Hook은 순수해야 합니다](#)



 Meta Open Source

Copyright © Meta Platforms, Inc

uwu?

React 학습하기

[빠르게 시작하기](#)

[설치하기](#)

[UI 표현하기](#)

[상호작용성 더하기](#)

[State 관리하기](#)

[탈출구](#)

API 참고서

[React APIs](#)

[React DOM APIs](#)

커뮤니티

[행동 강령](#)

[팀 소개](#)

[문서 기여자](#)

[감사의 말](#)

더 보기

[블로그](#)

[React Native](#)

[개인 정보 보호](#)

[약관](#)

