

[API 참고서 >](#)

Legacy React API

아래 API는 react 패키지에서 내보냈지만 Exported 새로 작성할 코드에서는 권장하지 않습니다. 링크를 통해 각각의 API 페이지에서 제시한 대안을 확인해주세요.

Legacy APIs

- `Children`은 `children` Prop으로 받은 JSX를 조작하고 변형할 수 있습니다. [대안 확인하기](#).
 - `cloneElement`를 통해 다른 엘리먼트를 시작점으로 사용하여 React 엘리먼트를 생성할 수 있습니다. [대안 확인하기](#).
 - `Component`는 자바스크립트 클래스로써 React 컴포넌트를 정의합니다. [대안 확인하기](#).
 - `createElement`로 React 엘리먼트를 생성합니다. 일반적으로 JSX를 대신 사용합니다.
 - `createRef`는 임의의 값을 포함할 수 있는 참조 객체를 생성합니다. [대안 확인하기](#).
 - `forwardRef`는 컴포넌트가 `ref`로 DOM 노드를 부모 컴포넌트에 노출시킵니다.
 - `isValidElement`는 값의 React 엘리먼트 여부를 확인합니다. 일반적으로 `cloneElement`와 함께 사용합니다.
 - `PureComponent`는 `Component`와 유사하지만, 동일한 Prop의 재렌더링은 생략합니다. [대안 확인하기](#).
-

Removed APIs

아래 API들은 React 19에서 제거되었습니다.

- `createFactory`: 대신 JSX를 사용하세요.
- 클래스 컴포넌트: `static contextTypes`: 대신 `static contextType`를 사용하세요.
- 클래스 컴포넌트: `static childContextTypes`: 대신 `static contextType`를 사용하세요.
- 클래스 컴포넌트: `static getChildContext`: 대신 `Context`를 사용하세요.
- 클래스 컴포넌트: `static propTypes`: 대신 `TypeScript`같은 타입 시스템을 사용하세요.
- 클래스 컴포넌트: `this.refs`: 대신 `createRef`를 사용하세요.

다음

Children



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

