



createRef

⚠️ 주의하세요!

`createRef` 는 주로 [클래스 컴포넌트](#)에 사용됩니다. 일반적으로 함수 컴포넌트는 [`useRef`](#) 를 대신 사용합니다.

`createRef` 는 임의의 값을 포함할 수 있는 [`ref`](#) 객체를 생성합니다.

```
class MyInput extends Component {  
  inputRef = createRef();  
  // ...  
}
```

- 레퍼런스
 - [`createRef\(\)`](#)
- 사용법
 - [클래스 컴포넌트에서 `ref` 선언하기](#)
- 대안
 - [`createRef` 를 사용하는 클래스에서 `useRef` 를 사용하는 함수로 마이그레이션하기](#)

레퍼런스

`createRef()`

`ref`를 클래스 컴포넌트 안에 선언하려면 `createRef` 를 호출합니다.

```
import { useRef, Component } from 'react';

class MyComponent extends Component {
  intervalRef = useRef();
  inputRef = useRef();
  // ...
}
```

아래에서 더 많은 예시를 볼 수 있습니다.

매개변수

`createRef` 는 매개변수를 받지 않습니다.

반환값

`createRef``는 단일 속성을 가진 객체를 반환합니다.

- `current`: 처음에는 `null`로 설정됩니다. 이를 나중에 다른 것으로 설정할 수 있습니다. `ref` 객체를 JSX 노드의 `ref` 어트리뷰트로 React에 전달하면 React는 이를 `current` 프로퍼티로 설정합니다.

주의 사항

- `createRef` 는 항상 *다른* 객체를 반환합니다. 이는 `{ current: null }` 을 직접 작성하는 것과 같습니다.
- 함수 컴포넌트에서는 항상 동일한 객체를 반환하는 `useRef` 를 대신 사용할 수 있습니다.
- `const ref = useRef()` 는 `const [ref, _] = useState(() => useRef(null))` 와 동일합니다.

사용법

클래스 컴포넌트에서 ref 선언하기

클래스 컴포넌트 내에서 참조를 선언하려면 `createRef` 를 호출하고 그 결과를 클래스 필드에 할당합니다.

```
import { Component, createRef } from 'react';

class Form extends Component {
  inputRef = createRef();

  // ...
}
```

이제 `ref={this.inputRef}` 를 JSX에 있는 `<input>` 에 전달하면, React는 `this.inputRef.current` 를 input DOM 노드가 차지하게 합니다. 예를 들어 input에 포커싱하는 버튼을 만드는 방법은 다음과 같습니다.

App.js

다운로드 새로그침 Clear 포크

```
import { Component, createRef } from 'react';

export default class Form extends Component {
  inputRef = createRef();

  handleClick = () => {
    this.inputRef.current.focus();
  }

  render() {
    return (
      <>
```

자세히 보기

⚠️ 주의하세요!

`createRef` 는 주로 **클래스 컴포넌트**에 사용됩니다. 일반적으로 함수 컴포넌트는 `useRef` 를 대신 사용합니다.

대안

`createRef` 를 사용하는 클래스에서 `useRef` 를 사용하는 함수로 마이그레이션하기

새로운 코드를 작성한다면 **클래스 컴포넌트** 대신 함수 컴포넌트를 사용하는 것을 추천합니다. `createRef` 를 사용하는 기존 클래스 컴포넌트가 있는 경우, 이를 변환하는 방법은 다음과 같습니다. 다음은 원본 코드입니다.

App.js

다운로드 새로그침 Clear 포크

```
import { Component, createRef } from 'react';

export default class Form extends Component {
  inputRef = createRef();

  handleClick = () => {
    this.inputRef.current.focus();
  }

  render() {
    return (
      <>
```

▼ 자세히 보기

이 컴포넌트를 클래스에서 함수로 변환하려면, `createRef` 호출을 `useRef` 호출로 바꿔줍니다.

App.js

다운로드 새로그침 Clear 포크

```
import { useRef } from 'react';

export default function Form() {
  const inputRef = useRef(null);

  function handleClick() {
    inputRef.current.focus();
  }

  return (
    <>
      <input ref={inputRef} />
    </>
  );
}
```

▼ 자세히 보기

< [이전](#)
[createElement](#)

[다음](#) >
[forwardRef](#)

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](#)

더 보기

[블로그](#)

[React Native](#)

[개인 정보 보호](#)

감사의 말

약관

