



use no memo

"use no memo" 는 React 컴파일러가 특정 함수를 최적화하지 않도록 합니다.

- 레퍼런스
 - "use no memo"
 - "use no memo" 가 최적화를 제외하는 방법
 - "use no memo" 를 사용해야 하는 경우
- 사용법
- 문제 해결
 - 지시어가 컴파일을 방지하지 않는 경우
 - 모범 사례
 - 참고

레퍼런스

"use no memo"

React 컴파일러 최적화를 방지하려면 함수의 시작 부분에 "use no memo" 를 추가하세요.

```
function MyComponent() {  
  "use no memo";  
  // ...  
}
```

함수에 "use no memo" 가 포함되어 있으면 React 컴파일러는 최적화 중에 해당 함수를 완전히 건너뜁니다. 이것은 디버깅할 때나, 컴파일러와 제대로 동작하지 않는 코드를 다룰 때 임시 탈출구로 유용합니다.

주의 사항

- "use no memo" 는 import나 다른 코드보다 먼저 함수 본문의 맨 처음에 있어야 합니다. (주석은 괜찮습니다.)
- 지시어는 백틱이 아닌 큰따옴표나 작은따옴표로 작성해야 합니다.
- 지시어는 "use no memo" 또는 별칭인 "use no forget" 과 정확히 일치해야 합니다.
- 이 지시어는 모든 컴파일 모드와 다른 지시어보다 우선합니다.
- 이것은 영구적인 해결책이 아닌 임시 디버깅 도구로 사용하기 위한 것입니다.

"use no memo" 가 최적화를 제외하는 방법

React 컴파일러는 빌드 시간에 코드를 분석하여 최적화를 적용합니다. "use no memo" 는 컴파일러에게 함수를 완전히 건너뛰도록 알려주는 명시적인 경계를 만듭니다.

이 지시어는 다른 모든 설정보다 우선합니다.

- all 모드에서: 전역 설정에도 불구하고 함수를 건너뛵니다.
- infer 모드에서: 휴리스틱이 최적화할 경우에도 함수를 건너뛵니다.

컴파일러는 React 컴파일러가 활성화되지 않은 것처럼 이러한 함수를 처리하여 작성된 그대로 유지합니다.

"use no memo" 를 사용해야 하는 경우

"use no memo" 는 드물게 그리고 임시로 사용해야 합니다. 일반적인 시나리오는 다음과 같습니다.

컴파일러 문제 디버깅

컴파일러가 문제를 일으키는 것으로 의심되면 문제를 분리하기 위해 일시적으로 최적화를 비활성화하세요.

```
function ProblematicComponent({ data }) {  
  "use no memo"; // TODO: 이슈 #123 수정 후 제거  
  
  // 정적으로 감지되지 않은 React 규칙 위반  
  // ...  
}
```

서드 파티 라이브러리 통합

컴파일러와 호환되지 않을 수 있는 라이브러리와 통합할 때 사용합니다.

```
function ThirdPartyWrapper() {  
  "use no memo";  
  
  useThirdPartyHook(); // 컴파일러가 잘못 최적화할 수 있는 사이드 이펙트가 있음  
  // ...  
}
```

사용법

"use no memo" 지시어는 React 컴파일러가 해당 함수를 최적화하지 않도록 함수 본문의 시작 부분에 배치합니다.

```
function MyComponent() {  
  "use no memo";  
  // 함수 본문  
}
```

지시어는 해당 모듈의 모든 함수에 영향을 미치도록 파일 상단에 배치할 수도 있습니다.

```
"use no memo";  
  
// 이 파일의 모든 함수는 컴파일러에 의해 건너뛰어집니다
```

함수 수준의 "use no memo" 는 모듈 수준의 지시어를 재정의합니다.

문제 해결

지시어가 컴파일을 방지하지 않는 경우

"use no memo" 가 작동하지 않는 경우입니다.

```
// ❌ 잘못된 예 - 코드 뒤에 지시어
```

```
function Component() {  
  const data = getData();  
  "use no memo"; // 너무 늦음!  
}
```

```
// ✅ 올바른 예 - 지시어가 먼저
```

```
function Component() {  
  "use no memo";  
  const data = getData();  
}
```

다음도 확인하세요.

- 철자 - "use no memo" 와 정확히 일치해야 합니다.
- 따옴표 - 백틱이 아닌 작은따옴표나 큰따옴표를 사용해야 합니다.

모범 사례

최적화를 비활성화하는 이유를 항상 문서화하세요.

```
// ✅ 좋은 예 - 명확한 설명과 추적
```

```
function DataProcessor() {  
  "use no memo"; // TODO: React 규칙 위반 수정 후 제거  
  // ...  
}
```

```
// ❌ 나쁜 예 - 설명 없음
```

```
function Mystery() {  
  "use no memo";  
  // ...  
}
```

참고

- ["use memo" - 컴파일에 포함하기](#)
- [React 컴파일러 - 시작 가이드](#)

< [이전](#)
["use memo"](#)

[라이브러리 컴파일](#) > [다음](#)

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](#)

