



API 참고서 > 지시어 >

# use memo

"use memo"는 특정 함수를 React 컴파일러의 최적화 대상으로 표시합니다.

## ▣ 중요합니다!

대부분의 경우 "use memo"는 필요하지 않습니다. 이 지시어는 최적화 대상을 명시적으로 표시해야 하는 annotation 모드에서 주로 사용합니다. infer 모드에서는 컴파일러가 이름 규칙(컴포넌트는 PascalCase, Hook은 use 접두사)을 기반으로 컴포넌트와 Hook을 자동으로 감지합니다. infer 모드에서 컴포넌트나 Hook이 컴파일되지 않는다면, "use memo"로 강제로 컴파일하기 보다는 이름 규칙을 올바르게 수정해야 합니다.

- [레퍼런스](#)

- ["use memo"](#)
- ["use memo" 가 함수를 최적화 대상으로 표시하는 방법](#)
- ["use memo" 를 사용해야 하는 경우](#)

- [사용법](#)

- [다양한 컴파일 모드에서 사용하기](#)

- [문제 해결](#)

- [최적화 여부 확인하기](#)
  - [참고](#)

## 레퍼런스

### "use memo"

특정 함수를 React 컴파일러 최적화 대상으로 표시하려면 함수의 시작 부분에 "use memo" 를 추가하세요.

```
function MyComponent() {  
  "use memo";  
  // ...  
}
```

함수에 "use memo" 가 포함되어 있으면, React 컴파일러는 빌드 시간에 이를 분석하고 최적화합니다. 컴파일러는 불필요한 재계산과 리렌더링을 방지하기 위해 값과 컴포넌트를 자동으로 메모이제이션합니다.

## 주의 사항

- "use memo" 는 함수 본문의 최상단에 있어야 하며, import나 다른 코드보다 앞에 있어야 합니다. (주석은 괜찮습니다.)
- 지시어는 백틱이 아니라 큰따옴표 또는 작은따옴표로 작성해야 합니다.
- 지시어는 "use memo" 와 정확히 일치해야 합니다.
- 함수의 첫 번째 지시어만 처리되며, 그 이후의 지시어는 무시됩니다.
- 지시어의 동작 방식은 `compilationMode` 설정에 따라 달라집니다.

## "use memo" 가 함수를 최적화 대상으로 표시하는 방법

React 컴파일러를 사용하는 React 앱에서는, 빌드 시점에 함수를 분석하여 최적화가 가능한지 판단합니다. 기본적으로 컴파일러는 어떤 컴포넌트를 메모이제이션할지 자동으로 추론하지만, 이는 설정한 `compilationMode` 에 따라 달라질 수 있습니다.

"use memo" 는 기본 동작을 재정의하여 함수를 명시적으로 최적화 대상으로 표시합니다.

- annotation 모드: "use memo" 를 선언한 함수만 최적화합니다.
- infer 모드: 컴파일러가 휴리스틱을 사용해 판단하지만, "use memo" 를 사용하면 최적화를 강제합니다.
- all 모드: 기본적으로 모든 코드가 최적화되므로, "use memo" 는 불필요합니다.

이 지시어는 코드베이스에서 최적화된 코드와 최적화되지 않은 코드 사이에 명확한 경계를 만들어, 컴파일 과정을 세밀하게 제어할 수 있게 합니다.

# "use memo" 를 사용해야 하는 경우

다음과 같은 경우에 "use memo" 사용을 고려할 수 있습니다.

## annotation 모드를 사용하는 경우

compilationMode: 'annotation' 에서는, 최적화하려는 모든 함수에 이 지시어를 반드시 선언해야 합니다.

```
// ✓ 이 컴포넌트는 최적화됩니다
function OptimizedList() {
  "use memo";
  // ...
}

// ✗ 이 컴포넌트는 최적화되지 않습니다
function SimpleWrapper() {
  // ...
}
```

## React 컴파일러를 점진적으로 도입하는 경우

먼저 annotation 모드로 시작한 뒤, 안정적인 컴포넌트부터 선택적으로 최적화하세요.

```
// 리프(Leaf) 컴포넌트부터 최적화 시작
function Button({ onClick, children }) {
  "use memo";
  // ...
}

// 동작을 검증하면서 점차 상위 컴포넌트로 확장
function ButtonGroup({ buttons }) {
  "use memo";
  // ...
}
```

# 사용법

## 다양한 컴파일 모드에서 사용하기

"use memo" 의 동작은 컴파일러 설정에 따라 달라집니다.

```
// babel.config.js
module.exports = {
  plugins: [
    ['babel-plugin-react-compiler', {
      compilationMode: 'annotation' // 또는 'infer', 'all'
    }]
  ];
};
```

## Annotation 모드

```
// ✅ "use memo"로 최적화됨
function ProductCard({ product }) {
  "use memo";
  // ...
}
```

```
// ❌ 최적화되지 않음 (지시어 없음)
function ProductList({ products }) {
  // ...
}
```

## Infer 모드 (기본값)

```
// 컴포넌트 이름 규칙을 따르므로 자동으로 메모이제이션됨
function ComplexDashboard({ data }) {
  // ...
}
```

```
// 건너뜀: 컴포넌트 이름 규칙을 따르지 않음
```

```
function simpleDisplay({ text }) {  
  // ...  
}
```

`infer` 모드에서는 컴파일러가 이름 규칙(컴포넌트는 PascalCase, Hook은 `use` 접두사)을 기반으로 컴포넌트와 Hook을 자동으로 감지합니다. `infer` 모드에서 컴포넌트나 Hook이 컴파일되지 않는다면, "`use memo`" 를 사용해 강제로 컴파일하기 보다는 이름 규칙을 수정하는 것을 권장합니다.

## 문제 해결

### 최적화 여부 확인하기

컴포넌트가 최적화되었는지 확인하려면

1. 빌드 결과물에서 컴파일된 코드를 확인하세요.
2. React DevTools에서 Memo ⚡ 배지를 확인하세요.

## 참고

- `"use no memo"` - 컴파일 대상에서 제외
- `compilationMode` - 컴파일 동작 설정
- `React Compiler` - 시작 가이드

이전

지시어

다음

`"use no memo"`

## 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](#)

