

# Optimizing **React** Performance with **useMemo**



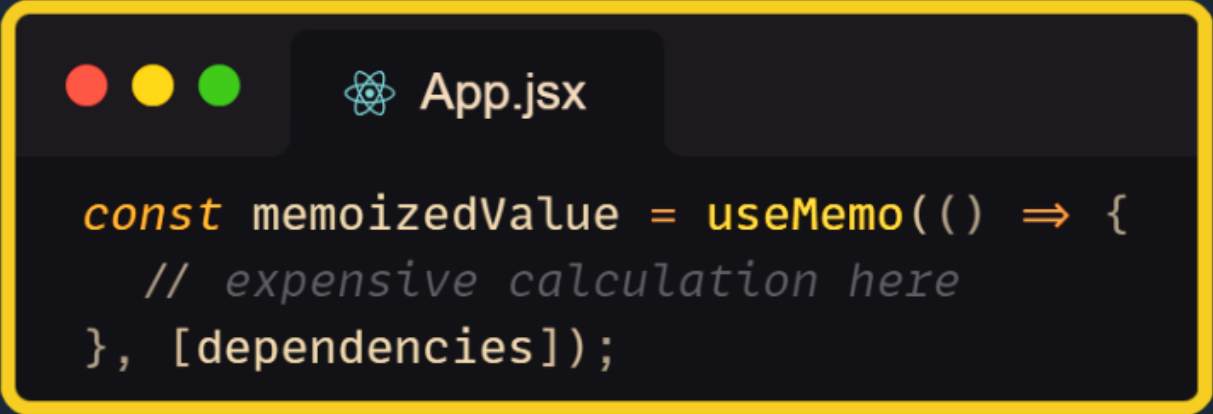
## What is **useMemo** ?

useMemo is a React hook that helps optimize performance by memoizing (remembering) the result of a calculation so it's not repeated unnecessarily.

## Why Use **useMemo** ?

- To avoid recalculating expensive operations (like loops or API calls) on every render.
- It ensures that your app runs faster, especially when dealing with large or complex data.

## Basic Syntax:



```
const memoizedValue = useMemo(() => {  
  // expensive calculation here  
}, [dependencies]);
```

- **memoizedValue**: The result of your calculation.
- **expensive calculation**: The function or operation you want to optimize.
- **[dependencies]**: The values that, when changed, will trigger a recalculation.



## When to Use **useMemo** ?

When you have:

- Expensive functions (like loops or data manipulation).
- Components that re-render often but the data doesn't change much.

## Example Without **useMemo** ?

```
App.jsx

import React, { useState } from 'react';

function App() {
  const [count, setCount] = useState(0);

  // Expensive calculation that runs every render
  const result = count * 100;

  return (
    <div>
      <h1>Result: {result}</h1>
      <button onClick={() => setCount(count + 1)}>Increase</button>
    </div>
  );
}

export default App;
```

**Problem:** The calculation of result happens on every render, even when count doesn't change much.



## Optimized Example Using **useMemo** ?

```
App.jsx

import React, { useState, useMemo } from 'react';

function App() {
  const [count, setCount] = useState(0);

  // Memoize the result to prevent recalculation on every render
  const result = useMemo(() => count * 100, [count]);

  return (
    <div>
      <h1>Result: {result}</h1>
      <button onClick={() => setCount(count + 1)}>Increase</button>
    </div>
  );
}

export default App;
```

### Explanation:

- **useMemo** only recalculates result when **count** changes, avoiding unnecessary recalculations on each render.
- This improves performance, especially when the calculation is more complex.





# Hopefully You Found It Usefull!

“Be sure to save this post so you  
can come back to it later”

[like](#)[Comment](#)[Share](#)