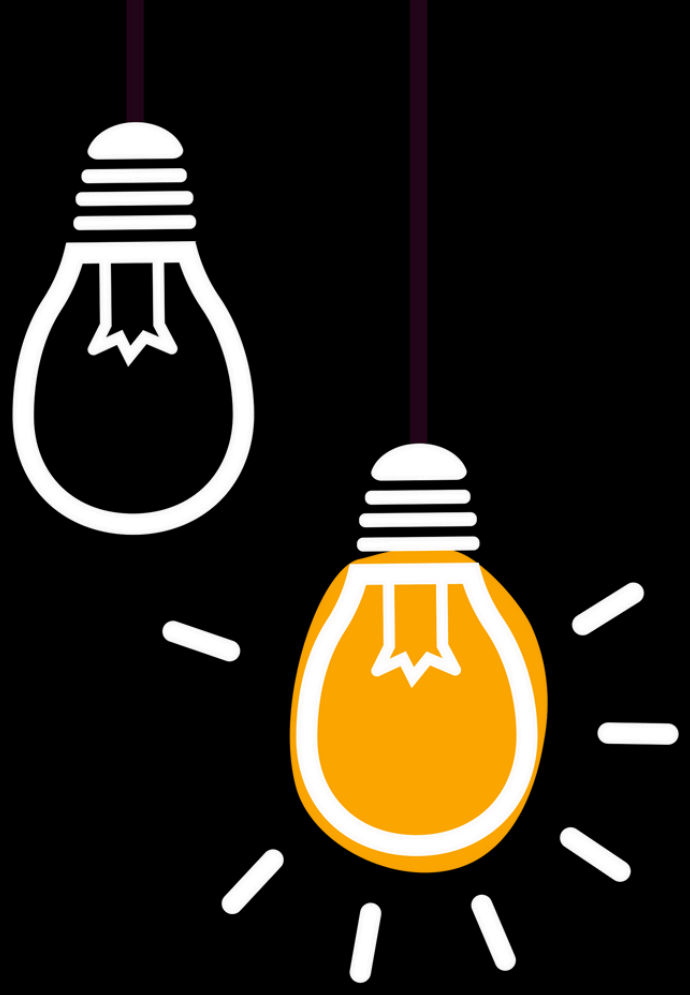


React Hooks

useMemo() Explained

🚀 Everything about useMemo() Hook in just **8 pages**

Arpitha Rajeev
arpitha.rajeev37@gmail.com

useMemo() Purpose

- Performance Optimization
- Avoiding unneccessary re-renders by memoizing a value
- Used in heavy computations such as filtering a large dataset or complex mathematical calculation

Arpitha Rajeev
arpitha.rajeev37@gmail.com

useMemo() Syntax

- Like other hooks, **useMemo**() also takes a callback function and a dependency array

useMemo()

```
const memoizedValue = useMemo(() =>
  expensiveComputation(a,b), [a, b]);
```

Arpitha Rajeev
arpitha.rajeev37@gmail.com

useMemo() Syntax Explained

- The function passed as a callback is **expensiveComputation(a, b)**
- This calculates the result but takes a lot of time and if the parameters **a** and **b** remain same, there is no need to calculate the result again
- Instead, we want to calculate the result only if either of **a** or **b** changes

Arpitha Rajeev
arpitha.rajeev37@gmail.com

useMemo() Syntax Explained

- Hence, as a second parameter in the **useMemo**() **Hook**, we are passing a dependency array
- That means the callback function is called only when either of the parameters in dependency array changes
- If they remain same, **useMemo**() **Hook** returns the memoized value

Arpitha Rajeev
arpitha.rajeev37@gmail.com

useMemo() Example

useMemo()

```
const [searchTerm, setSearchTerm] = useState("");

const filteredItems = useMemo(() => {
  return items.filter((item) => item.includes(searchTerm))
)}, [searchTerm]);
```

- If user enters same value in the search bar, useMemo() will return the stored value without filtering again in the huge dataset

Arpitha Rajeev
arpitha.rajeev37@gmail.com

useMemo() Example

useMemo()

```
const [searchTerm, setSearchTerm] = useState("");

const filteredItems = useMemo(() => {
  return items.filter((item) => item.includes(searchTerm))
)}, [searchTerm]);
```

- The filter function is called only if the **searchTerm** changes
- This increases the performance by avoiding unnecessary recalculations

Arpitha Rajeev
arpitha.rajeev37@gmail.com

useMemo(): When to Avoid?

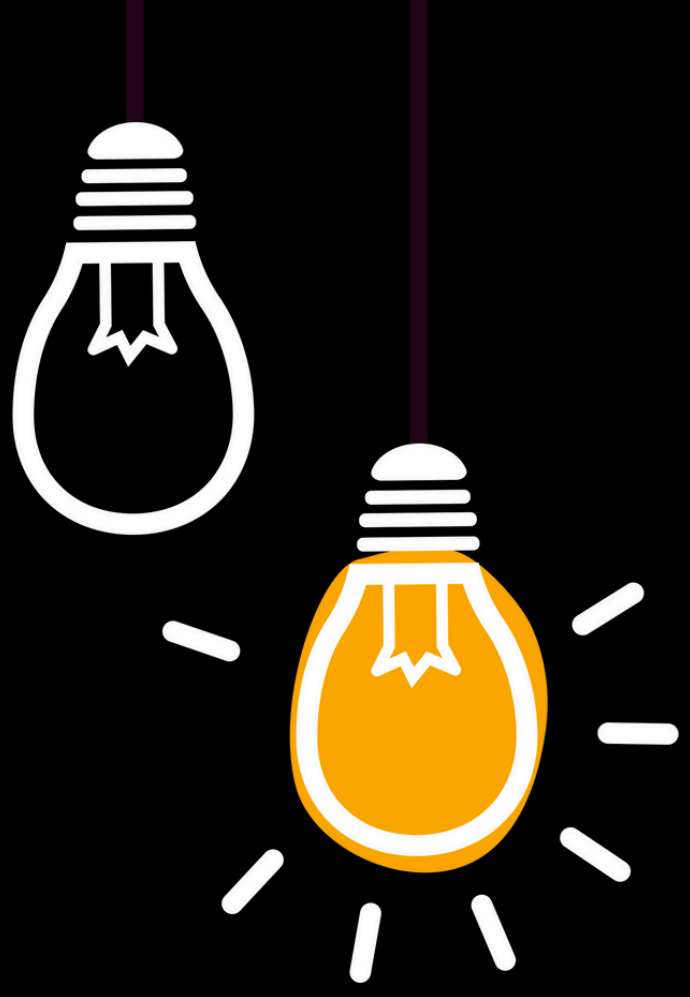
- If the **calculation is not expensive** and complex, avoid using **useMemo**() because of **Over-Optimization**
- Memoized values are stored in the **memory**, hence if the operation is not expensive it costs us unnecessary extra memory

Arpitha Rajeev
arpitha.rajeev37@gmail.com

useMemo(): How it works?

- The result of the function executed for the first time is stored internally by **React**
- This internal storage is not accessible to the user directly
- **useMemo**() returns the previously computed value from its internal cache

Arpitha Rajeev
arpitha.rajeev37@gmail.com



Follow Me

For more such content on Software Development

Arpitha Rajeev
arpitha.rajeev37@gmail.com