

# REACTJS

*Hà Nội, ngày 29 tháng 12 năm 2023*



# I. React Hook

## I.4. useMemo

useMemo là một trong những hooks của React được sử dụng để memoize (lưu trữ cache) kết quả của một hàm và chỉ tính toán lại giá trị khi các dependencies thay đổi. Điều này giúp tối ưu hóa hiệu suất bằng cách tránh tính toán lại giá trị nếu những thứ quan trọng không thay đổi.

### Khi nào sử dụng useMemo:

- Tính toán Tính Đồng Nhất (Expensive Computations)
- Tránh Re-calculations Trong Render.



# I. React Hook

## I.4. useMemo

**Cần chú ý khi sử dụng useMemo:**

- Chỉ Memoize Khi Cần Thiết
- Dependency Array.
- Không Tạo Memoization Quá Mức.



# I. React Hook

## I.5. useContext

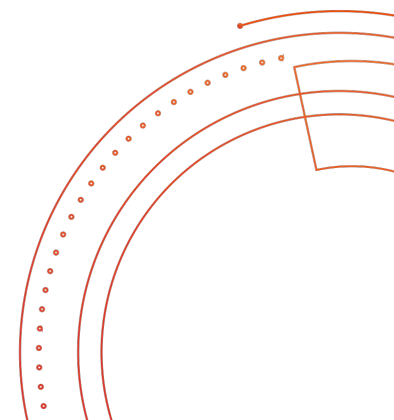
useContext là một trong những hooks của React được sử dụng để truy cập giá trị của một Context.

Context là một cách để chia sẻ dữ liệu giữa các component mà không cần truyền giá trị qua nhiều cấp component.

useContext giúp bạn dễ dàng truy cập giá trị từ Context trong một component.

### Sử Dụng useContext:

- Chia sẻ dữ liệu global
- Tránh truyền props qua quá nhiều cấp



# I. React Hook

## I.5. useContext

### Tạo Context (Có 2 cách tạo)

- Trước hết, bạn cần tạo một Context bằng cách sử dụng `React.createContext`.

```
import React from 'react';  
  
const ExampleContext = React.createContext();  
export default ExampleContext;
```

# I. React Hook

## I.5. useContext

### Sử Dụng Context Provider:

- Sử dụng `Context.Provider` để cung cấp giá trị cho các component con.

```
function App() {  
  const exampleValue = "Hello from Context";  
  return (  
    <ExampleContext.Provider value={exampleValue}>  
      <MyComponent />  
    </ExampleContext.Provider>  
  );  
}
```

# I. React Hook

## I.5. useContext

### Sử Dụng useContext Trong Component:

- Trong component mà bạn muốn sử dụng giá trị từ Context, sử dụng **useContext**.

```
import React, { useContext } from 'react';  
  
function MyComponent() {  
  const valueFromContext = useContext(ExampleContext);  
  
  return (  
    <div>{valueFromContext}</div>  
  
  );  
}
```

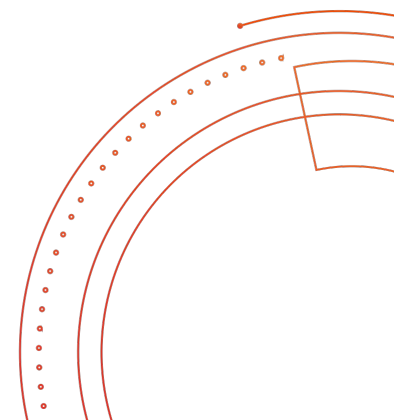


# I. React Hook

## I.5. useContext

### Cần Chú Ý Khi Sử Dụng **useContext**:

- **useContext** sẽ cập nhật component mỗi khi giá trị của Context thay đổi. Điều này có thể gây re-render không mong muốn nếu không cần thiết.
- Để tối ưu hiệu suất, hãy đặt **Context.Provider** càng gần component mà cần truy cập Context. Điều này giúp tránh việc cập nhật không cần thiết cho các component không cần truy cập giá trị từ Context.
- **useContext** thường được sử dụng để chia sẻ dữ liệu global. Nếu bạn chỉ cần truy cập một số dữ liệu cục bộ trong component, có thể làm mất đi sự rõ ràng của code. Hãy sử dụng props cho các trường hợp này.





---

# THANK YOU

---

FOR  
WATCHING