### TỔNG QUÁT NỘI DUNG

**- Class Component** là cách truyền thống, dùng lifecycle methods để quản lý vòng đời và side effects.

**- Function Component** hiện đại, dùng Hook useEffect thay thế lifecycle methods, giúp code ngắn gọn, dễ hiểu và tái sử dụng logic hiệu quả.

- useEffect thực hiện side effects tương tự lifecycle trong class component và có hàm cleanup trả về.

- Cleanup function trong useEffect giúp tránh rò rỉ bộ nhớ và lỗi khi component unmount hoặc effect chạy lại.

### KHÁC NHAU

| **Tiêu chí** | **Function Component** | **Class Component** |
| --- | --- | --- |
| **Cấu trúc** | Một hook (useEffect) xử lý tất cả giai đoạn , | Nhiều method riêng biệt |
| **Cleanup** | Trả về hàm cleanup trong useEffect | Dọn dẹp trong componentWillUnmount |
| **Truy cập state/props** | Qua closures và hook | Qua this.state và this.props |
| **Tái sử dụng logic** | Cao, dễ đóng gói thành custom hooks để tái sử dụng | Thấp, thường dùng HOC hoặc render props |
| **Độ rõ ràng giai đoạn** | Ít tách biệt, dùng nhiều useEffect với dependency khác nhau để phân chia logic | Rõ ràng, mỗi lifecycle method xử lý riêng từng giai đoạn |
| **Dễ học và test** | Dễ hiểu, logic hook dễ test và tái sử dụng | Khó với người mới do this và lifecycle phức tạp |

### Tóm tắt

\*useEffect là cách hiện đại, gọn nhẹ, quản lý side effects hiệu quả trong function component.

\*Lifecycle methods rõ ràng nhưng kém linh hoạt và khó tái sử dụng.

\*React ưu tiên dùng Function Component với Hooks cho dự án mới vì code sạch và dễ bảo trì hơn