Tổng Hợp Kiến Thức

Kỹ thuật phân chia Component hợp lý

- **Tại sao cần phân chia Component?**
  - **Tái sử dụng**: Giảm code trùng lặp, tăng tốc độ phát triển.
  - **Bảo trì dễ dàng**: Code dễ đọc, dễ quản lý.
  - **Tối ưu hiệu suất**: Hạn chế re-render không cần thiết.
  - **Tổ chức folder hợp lý**: Dễ tìm kiếm và điều hướng.
- **Khi nào nên tách Component?**
  - **Có thể tái sử dụng** (Button, Input, Card, v.v.).
  - **Component quá lớn/phức tạp**: Chia nhỏ để dễ quản lý.
  - **KHÔNG nên tách nếu**:
    - Chỉ dùng một nơi.
    - Quá nhỏ, không làm code dễ đọc hơn.
    - Tăng độ phức tạp không cần thiết.
- **Cấu trúc thư mục Component**

```
src/
└── components/
    ├── Button.jsx
    ├── Button.css
    ├── Card.jsx
    ├── Card.css
    ├── Input.jsx
    ├── Input.css
    └── ...
```

Render Có Điều Kiện trong React

- **if/else** (Dùng khi điều kiện phức tạp):

```jsx
if (isLoggedIn) {
  content = <p>Chào mừng!</p>;
} else {
  content = <p>Vui lòng đăng nhập.</p>;
}
```

- **Toán tử ba ngôi** `?:` (Dùng cho điều kiện đơn giản)

```jsx
{isLoggedIn ? <p>Chào mừng!</p> : <p>Vui lòng đăng nhập.</p>}
```

- **Toán tử logic AND** `&&` (Dùng khi chỉ có điều kiện `true`)

```jsx
{isLoggedIn && <p>Chào mừng!</p>}
```

- **Toán tử logic OR** `||` (Dùng để đặt giá trị mặc định)

```jsx
<h1>Xin chào, {userName || "khách!"}</h1>
```

Đặt State Để Tối Ưu Re-render

- **Quy tắc quan trọng**
  - **State thay đổi → Component re-render** (và có thể ảnh hưởng đến component con).
  - **Colocate State**: Đặt state gần nơi sử dụng nhất.
  - **Lifting State Up**: Khi nhiều component cần dùng chung state.

---

useEffect – Quản Lý Vòng Đời Component

- **Cú pháp cơ bản:**

```
useEffect(() => {
  // Code chạy sau mỗi render
  return () => {
    // Cleanup khi component bị unmount
  };
}, [/* dependencies */]);
```

- **Các trường hợp dùng useEffect:**
  - **Chạy một lần khi component mount** (componentDidMount)

```
useEffect(() => {
  console.log("Component mounted!");
}, []);
```

  - **Chạy lại khi state hoặc props thay đổi** (componentDidUpdate)

```
useEffect(() => {
  console.log(`State count thay đổi: ${count}`);
}, [count]);
```

  - **Cleanup khi component unmount** (componentWillUnmount)

```
useEffect(() => {
  const timer = setInterval(() => {
    console.log("Timer chạy!");
  }, 1000);

  return () => {
    clearInterval(timer);
    console.log("Timer dừng!");
  };
}, []);
```