

# Tổng Hợp Kiến Thức

## Mục đích Context API

- Chia sẻ dữ liệu "toàn cục" trong một phần của cây component.
- Giải quyết vấn đề **Props Drilling** (không cần truyền props lòng vòng).

## Thành phần chính của Context API

- **createContext(defaultValue)**

- Tạo ra một **đối tượng Context**.
- `defaultValue` : Giá trị mặc định (sử dụng khi không có Provider phù hợp).
- Ví dụ:

```
const MyContext = React.createContext('Giá trị mặc định');
```

- **Provider**

- Component **cung cấp** dữ liệu Context.
- `value` : Prop **bắt buộc**, chứa dữ liệu Context cần chia sẻ (có thể là bất kỳ giá trị JS nào: state, object, function...).
- Bọc các component con cháu cần truy cập Context bằng `Provider`.
- Ví dụ:

```
<MyContext.Provider value={/* Giá trị Context */}>
  /* Các component con cháu */
</MyContext.Provider>
```

- **Consumer**

- Component **tiêu thụ** dữ liệu Context (cách cũ hơn).
- Sử dụng Render Props pattern: `<MyContext.Consumer>{value => /* render UI dựa trên value */}</MyContext.Consumer>`.
- `value` : Giá trị Context được cung cấp bởi `Provider` **gần nhất**.
- Ví dụ:

```
<MyContext.Consumer>
  {value => (
    <div>Giá trị Context: {value}</div>
  )}
</MyContext.Consumer>
```

- **useContext(MyContext) Hook**

- Hook **tiêu thụ** dữ liệu Context (cách hiện đại).
- `useContext(MyContext)` trả về **giá trị Context** được cung cấp bởi `Provider` **gần nhất**.
- Ví dụ:

```
const value = useContext(MyContext);
return <div>Giá trị Context: {value}</div>;
```

## Cập nhật Context Value

- `Provider` re-render khi prop `value` thay đổi.
- Khi `Provider` re-render, tất cả component `Consumer` hoặc `useContext` **bên dưới nó re-render theo**.
- Thường dùng `useState` trong `Provider` để quản lý và cập nhật `value`.

# Các bước thiết lập Context API

- 1. **Tạo Context:** `const MyContext = createContext(defaultValue);`
- 2. **Cung cấp Context:** `<MyContext.Provider value={...}> ... </MyContext.Provider>`
- 3. **Tiêu thụ Context:**
  - **Consumer :** `<MyContext.Consumer> {value ⇒ ...} </MyContext.Consumer>`
  - **useContext :** `const value = useContext(MyContext);`