**[Bài đọc] Props**

1. **Khái niệm**

* Trong ReactJS, props là một cách để truyền dữ liệu từ một component cha xuống một component con.
* Trong class components, props là một đối tượng chứa các thuộc tính (properties) được truyền từ component cha.
* Ví dụ:

**import React, { Component } from "react";**

**class ChildComponent extends Component {**

**render() {**

**// Sử dụng props để lấy dữ liệu từ component cha**

**const { message } = this.props;**

**return (**

**<div>**

**<p>Message from parent: {message}</p>**

**</div>**

**);**

**}**

**}**

**class ParentComponent extends Component {**

**render() {**

**// Truyền dữ liệu xuống component con thông qua props**

**return (**

**<div>**

**<ChildComponent message="Hello from parent!" />**

**</div>**

**);**

**}**

**}**

**export default ParentComponent;**

* Trong ví dụ trên:
  + Trong **ParentComponent**, chúng ta truyền dữ liệu xuống **ChildComponent** thông qua thuộc tính message.
  + Khi render **ChildComponent** và gán giá trị cho **message**, React sẽ tự động chuyển nó xuống component con dưới dạng props.
  + Trong **ChildComponent**, chúng ta sử dụng **this.props.message** để truy cập giá trị của message.
  + Các giá trị của props là không thể thay đổi (immutable), nghĩa là một khi chúng đã được đặt, bạn không thể thay đổi chúng trực tiếp từ bên trong component.

1. **Một số đặc điểm của props**
   * Props là không thể thay đổi (immutable): Bạn không nên thay đổi giá trị của props từ bên trong component. Điều này giúp React duy trì tính nguyên vẹn của luồng dữ liệu và làm cho quá trình quản lý trạng thái trở nên dễ dàng hơn.
   * Destructuring props: Đôi khi, để làm mã nguồn ngắn gọn hơn, bạn có thể sử dụng destructuring để trích xuất các giá trị props trực tiếp. Trong ví dụ trên, const { message } = this.props; là một ví dụ về việc này.
   * Truyền xuống functions qua props: Bạn cũng có thể truyền các hàm xuống các component con thông qua props. Điều này cho phép các component con gọi các hàm được định nghĩa ở component cha.