BÀI THU HOẠCH III
(Phần II)

TÌM HIỂU VỀ JSX, PROPS, STATE VÀ COMPONENT

Thực tập sinh: Lê Vương Khánh

Người hướng dẫn: Nguyễn Trần Xuân Lộc, Trương Tấn Sang

MỤC LỤC

I,Tìm hiểu về JSX

1, Định nghĩa

JSX là một phần của JavaScript và là một phần quan trọng của React, một thư viện JavaScript phổ biến được sử dụng để xây dựng giao diện người dùng. JSX là một cú pháp mở rộng của JavaScript, cho phép bạn viết mã HTML-like (có thể gọi là "markup") bên trong mã JavaScript. JSX giúp bạn dễ dàng tạo và quản lý các thành phần giao diện người dùng trong ứng dụng React.

Dưới đây là một ví dụ:

Trong ví dụ này, <h1>Hello, world!</h1> được coi là một biểu thức JSX và nó sẽ được biên dịch thành JavaScript tương ứng. Trong trường hợp này, nó sẽ tạo một đối tượng React Element để hiển thị tiêu đề "Hello, world!" trong giao diện người dùng. JSX cung cấp một cách tiện lợi để xây dựng cây thành phần trong React, với khả năng kết hợp các thành phần con vào thành phần cha và truyền dữ liệu thông qua các thuộc tính (props). Nó giúp tạo ra mã JavaScript dễ đọc hơn và dễ bảo trì hơn cho ứng dụng React.

2, Đặc điểm và chức năng chính của JSX

Đặc Điểm và Chức Năng Chính của JSX	Mô Tả
Kết hợp JavaScript và HTML-like	Cho phép viết mã JavaScript và HTML-like trong cùng một file.
Biểu thức JavaScript	Có thể sử dụng biểu thức JavaScript bên trong JSX để tính toán giá trị.
Tạo các thành phần React	Giúp tạo các thành phần React dễ dàng bằng cách viết mã HTML-like.

Đặc Điểm và Chức Năng Chính của JSX	Mô Tả
Kết hợp các thành phần	Cho phép kết hợp và nhúng các thành phần React vào nhau.
Truyền dữ liệu qua props	Thông qua props, bạn có thể truyền dữ liệu từ thành phần cha sang con.
Tích hợp với Babel	JSX được Babel hỗ trợ để biên dịch thành JavaScript chuẩn.

3, Các hạn chế cần phải lưu ý

Không phải là HTML thuần: JSX giống HTML, nhưng có một số khác biệt. Ví dụ, bạn cần sử dụng className thay vì class để đặt lớp CSS, và htmlFor thay vì for để liên kết với nhãn <label>. JSX cũng có một số từ khóa riêng biệt như class và for.

Phải biên dịch: JSX không phải là JavaScript chuẩn và cần được biên dịch thành JavaScript thông qua trình biên dịch như Babel. Việc này tạo ra bước biên dịch bổ sung trong quá trình phát triển ứng dụng.

Tính dễ hiểu cho người mới: Đối với những người mới học React, việc sử dụng JSX có thể tạo ra sự ngạc nhiên và khó hiểu ban đầu. Việc học cú pháp mới và cách kết hợp mã JavaScript và HTML có thể đòi hỏi thời gian.

Khả năng lỗi phát sinh: Do JSX được biên dịch thành JavaScript, nên có thể phát sinh lỗi trong quá trình biên dịch hoặc lúc runtime. Điều này có thể làm tăng thời gian debug và khó khăn trong việc tìm kiếm lỗi.

Phải quan tâm đến vấn đề an toàn: JSX có thể dễ dàng dẫn đến các vấn đề bảo mật nếu không được sử dụng cẩn thận. Chẳng hạn, việc truyền dữ liệu người dùng trực tiếp vào JSX mà không được xử lý cẩn thận có thể dẫn đến các lỗ hổng bảo mật như Cross-Site Scripting (XSS).

Khả năng làm phức tạp mã: Đôi khi, việc sử dụng JSX có thể làm phức tạp mã và làm cho nó khó đọc hơn, đặc biệt là khi bạn có các biểu thức phức tạp hoặc

cây thành phần lồng nhau sâu.

II) Prop, State và Component

1, Định nghĩa

-Props: Props là viết tắt của "properties" và là một cách để truyền dữ liệu từ thành phần cha sang thành phần con trong React. Props là không thay đổi (immutable), có nghĩa rằng chúng không thể thay đổi giá trị bên trong một thành phần con. Props được truyền dưới dạng các thuộc tính của thành phần và có thể được truy cập bằng this.props trong thành phần con.

Dưới đây là ví dụ:

-State: State là một đối tượng chứa các thông tin liên quan đến thành phần và có thể thay đổi theo thời gian. Khi state thay đổi, thành phần sẽ render lại để hiển thị dữ liệu mới. State thường được sử dụng để lưu trữ và quản lý trạng thái của thành phần.

Ví dụ:

- Component: Component là một phần của ứng dụng React và được sử dụng để phân chia giao diện người dùng thành các phần nhỏ và có thể tái sử dụng. Mỗi thành phần có thể bao gồm các props và state của riêng nó và có thể tự định nghĩa cách hiển thị và hoạt động của nó thông qua phương thức render(). Trong React, có hai loại thành phần chính: thành phần lớp (Class Component) và thành phần hàm (Functional Component). Thành phần lớp kế thừa từ React.Component và có trạng thái (state), trong khi thành phần hàm là một hàm JavaScript không có trạng thái riêng (stateless) và thường được sử dụng để định nghĩa thành phần đơn giản và không quản lý trạng thái.
  Ví dụ:

II) Tính chất

1) Props

- Không thay đổi (Immutable): Props là không thay đổi, có nghĩa rằng sau khi được gán giá trị, chúng không thể thay đổi bên trong thành phần con. Điều này đảm bảo tính ổn định và dự đoán khi truyền dữ liệu từ cha đến con.

- Truyền dữ liệu từ cha đến con: Props được sử dụng để truyền dữ liệu từ thành phần cha sang thành phần con. Điều này cho phép tái sử dụng thành phần con và làm cho ứng dụng phản ánh dữ liệu thực tế.

- Được truyền dưới dạng thuộc tính: Props là các thuộc tính của thành phần và có thể truy cập bằng cú pháp this.props (trong thành phần lớp) hoặc thông qua tham số props (trong thành phần hàm).

2) State

- Thay đổi theo thời gian: State là một đối tượng chứa thông tin có thể thay đổi theo thời gian. Khi state thay đổi, thành phần sẽ được render lại để hiển thị trạng thái mới.

- Quản lý trạng thái của thành phần: State thường được sử dụng để lưu trữ và quản lý trạng thái nội bộ của một thành phần. Nó giúp thành phần React biết được dữ liệu nào cần hiển thị và cách xử lý các sự kiện.

- Không nên thay đổi state trực tiếp: State không nên được thay đổi trực tiếp bằng cách gán lại giá trị. Thay vào đó, bạn nên sử dụng phương thức setState để cập nhật state một cách an toàn.

3) Component:

- Phân chia giao diện người dùng: Component là cách để phân chia giao diện người dùng thành các phần nhỏ và đơn giản hơn. Nó giúp tạo cấu trúc và tổ chức ứng dụng một cách có trật tự.

- Tái sử dụng: Component cho phép tái sử dụng mã và logic. Bạn có thể sử dụng các thành phần đã tồn tại trong nhiều nơi trong ứng dụng của bạn, giúp tiết kiệm thời gian và tạo ra mã dễ bảo trì.

- Có thể có trạng thái hoặc không trạng thái: React có hai loại thành phần chính: thành phần có trạng thái (Class Component) và thành phần không trạng thái (Functional Component). Thành phần có trạng thái có khả năng lưu trữ và quản lý state, trong khi thành phần không trạng thái là các hàm JavaScript đơn giản không có state.

III) Sharing state trong Component

1, Định Nghĩa

Sharing state trong component đề cập đến việc chia sẻ dữ liệu trạng thái (state) giữa các thành phần (components) trong ứng dụng React. Khi bạn cần truyền dữ liệu từ một thành phần này sang một thành phần khác hoặc muốn các thành phần chia sẻ dữ liệu chung, bạn cần thực hiện việc "chia sẻ trạng thái."

2, Các phương pháp tiến hành

- Truyền props từ cha đến con: Đây là cách phổ biến nhất để chia sẻ dữ liệu giữa các thành phần. Thành phần cha có thể truyền dữ liệu cho thành phần con bằng cách sử dụng props. Thành phần con sau đó sử dụng dữ liệu này để hiển thị hoặc thực hiện các hành động cụ thể.

- Sử dụng Context API: Context API là một cách mạnh mẽ để chia sẻ dữ liệu trạng thái giữa các thành phần mà không cần truyền props qua nhiều cấp cha. Bạn có thể tạo ra một Context và sử dụng Provider để cung cấp dữ liệu và Consumer để tiêu thụ nó. Điều này giúp bạn tránh truyền props qua nhiều cấp cha và làm cho việc chia sẻ dữ liệu dễ dàng hơn.

- Sử dụng Redux hoặc MobX: Redux và MobX là các thư viện quản lý trạng thái của ứng dụng React. Chúng cho phép bạn lưu trữ trạng thái ứng dụng trong một store toàn cục và chia sẻ dữ liệu giữa các thành phần thông qua việc kết nối (connect) với store. Sử dụng React Hooks:

- React Hooks như useState và useEffect cho phép bạn quản lý trạng thái trong các thành phần hàm. Bạn có thể sử dụng hooks để tạo và chia sẻ dữ liệu trạng thái giữa các thành phần hàm.

Lựa chọn cách nào để chia sẻ trạng thái trong component phụ thuộc vào tình huống cụ thể và cách bạn tổ chức ứng dụng của mình. Prop drilling (truyền props qua nhiều cấp cha) có thể thích hợp cho các ứng dụng nhỏ, trong khi Context API, Redux hoặc MobX thích hợp cho các ứng dụng lớn

và phức tạp hơn. React Hooks cung cấp một giải pháp linh hoạt cho các thành phần hàm.