

Làm Chủ State trong React: Từ Lý Thuyết Đến Thực Hành

Chào mừng các bạn đến với buổi trình bày về State trong React! Buổi hôm nay, chúng ta sẽ cùng nhau khám phá khái niệm State, một trong những yếu tố cốt lõi để xây dựng các ứng dụng React động và tương tác. Chúng ta sẽ đi từ lý thuyết cơ bản, so sánh State với Props, tìm hiểu về Hook **useState**, và cuối cùng là áp dụng vào một project thực tế để các bạn có thể nắm vững kiến thức một cách tốt nhất.



by **Hồng Thái Phạm**

State Là Gì? Phân Biệt State và Props

State (Trạng Thái)

State là một đối tượng JavaScript chứa dữ liệu của một component. State có thể thay đổi theo thời gian, và khi State thay đổi, component sẽ tự động re-render (vẽ lại) để cập nhật giao diện.

Tóm lại, State là dữ liệu "nội bộ" của component, có thể thay đổi và ảnh hưởng đến giao diện. Props là dữ liệu "bên ngoài", được truyền từ component cha và không thể thay đổi bởi component con.

Props (Thuộc Tính)

Props là cách một component "nhận" dữ liệu từ component cha của nó. Props là bất biến, tức là component con không thể thay đổi giá trị của props mà nó nhận được.



Hook **useState**: Quản Lý State Trong Function Component

1

Import **useState**

Nhập hook **useState** từ thư viện React:

```
import React, { useState } from 'react';
```

2

Khai Báo State

Sử dụng **useState** để khai báo một biến state và một hàm để cập nhật nó:

```
const [count, setCount] = useState(0);
```

3

Cập Nhật State

Gọi hàm **setCount** để thay đổi giá trị của **count**. Khi state thay đổi, component sẽ re-render.

Hook **useState** giúp chúng ta quản lý state một cách dễ dàng trong function component. Thay vì sử dụng class component phức tạp, **useState** cho phép chúng ta khai báo và cập nhật state trực tiếp trong hàm.



Ví Dụ: Bộ Đếm (Counter) Đơn Giản

Khởi Tạo State

Khởi tạo state **count** với giá trị ban đầu là 0.

Hiển Thị State

Hiển thị giá trị của state **count** trên giao diện.

Cập Nhật State

Tạo một hàm để tăng giá trị của **count** khi người dùng bấm nút.

Chúng ta sẽ xây dựng một bộ đếm đơn giản để minh họa cách state thay đổi và cập nhật giao diện. Khi người dùng bấm nút, giá trị của bộ đếm sẽ tăng lên và hiển thị ngay lập tức.



Code Demo: Xây Dựng Bộ Đếm (Counter)

```
import React, { useState } from 'react';

function Counter() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}

export default Counter;
```

Đây là đoạn code hoàn chỉnh cho bộ đếm của chúng ta. Hãy cùng nhau phân tích từng dòng để hiểu rõ hơn về cách **useState** hoạt động.

Giải Thích Code: `useState` và `setCount`



`useState(0)`

Khởi tạo state **count** với giá trị ban đầu là 0. Hàm này trả về một mảng gồm hai phần tử: giá trị state hiện tại và hàm để cập nhật state.



`setCount(count + 1)`

Hàm **setCount** được sử dụng để cập nhật giá trị của state **count**. Khi hàm này được gọi, React sẽ re-render component để hiển thị giá trị mới.



Re-render

Khi state thay đổi, component sẽ tự động re-render để cập nhật giao diện. Đây là cơ chế hoạt động chính của React.

Hàm **useState(0)** sẽ khởi tạo giá trị state ban đầu là 0. **setCount** là một function giúp cập nhật giá trị đó. Mỗi khi state được cập nhật, component sẽ re-render để hiển thị dữ liệu mới.

Demo: Thay Đổi State và Cập Nhật Giao Diện

1

Bấm Nút

Người dùng bấm vào nút "Click me".

2

Gọi `setCount`

Hàm `setCount` được gọi để tăng giá trị của state `count`.

3

Re-render

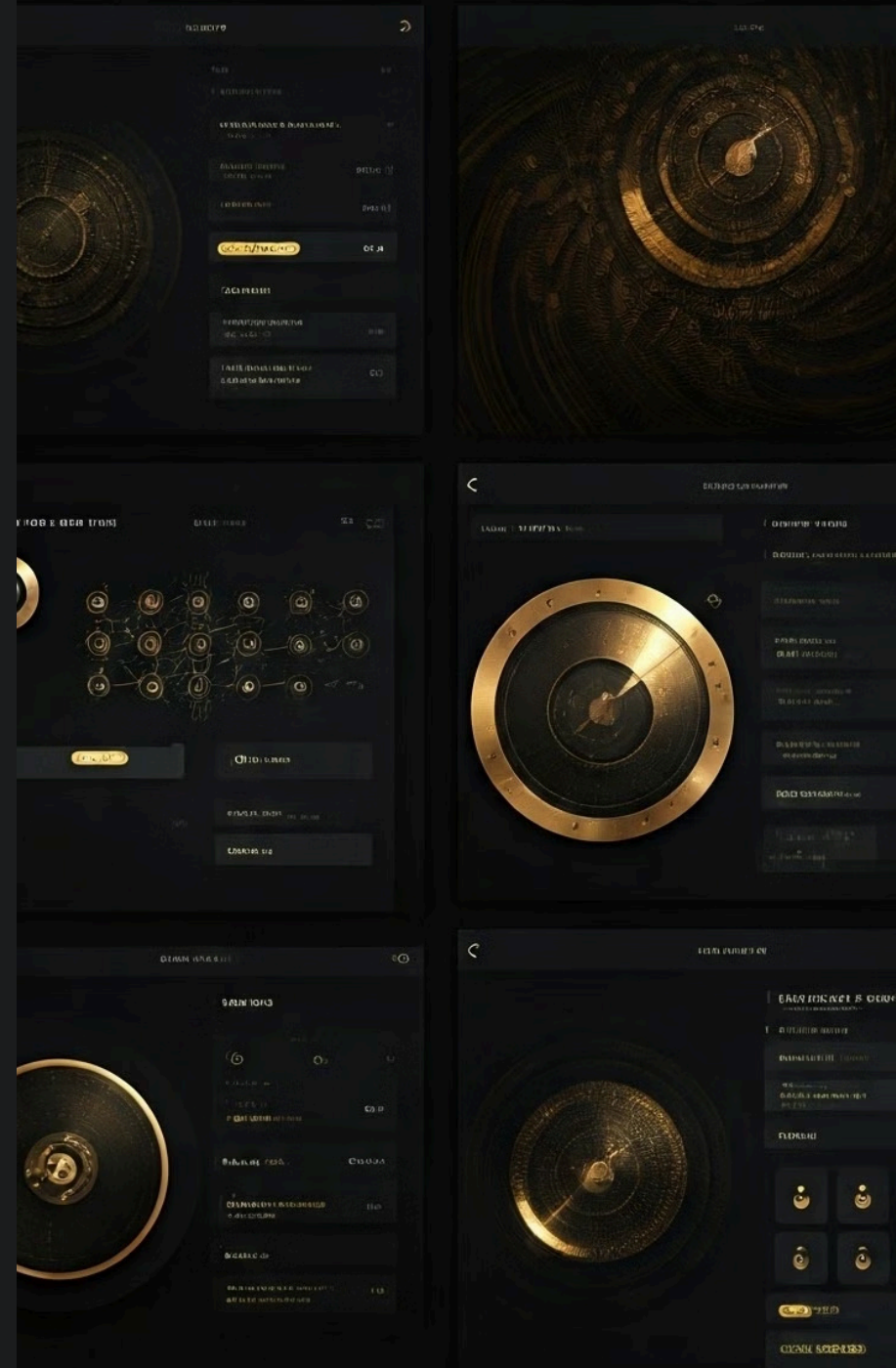
React re-render component `Counter`.

4

Hiển Thị

Giao diện hiển thị giá trị mới của state `count`.

Bây giờ, chúng ta sẽ cùng nhau xem một demo trực tiếp để thấy rõ cách state thay đổi và giao diện được cập nhật ngay lập tức.



Tổng Kết và Các Bước Tiếp Theo

1 State

State là dữ liệu nội bộ của component, có thể thay đổi và ảnh hưởng đến giao diện.

2 useState

Hook **useState** giúp quản lý state một cách dễ dàng trong function component.

3 Thực Hành

Hãy thực hành xây dựng các ứng dụng nhỏ để nắm vững kiến thức về state.

Vậy là chúng ta đã cùng nhau tìm hiểu về State trong React. Hy vọng rằng buổi trình bày này đã giúp các bạn hiểu rõ hơn về khái niệm này. Hãy tiếp tục thực hành và khám phá thêm để trở thành những React developer chuyên nghiệp!

REACT STATE MANAGEMENT

