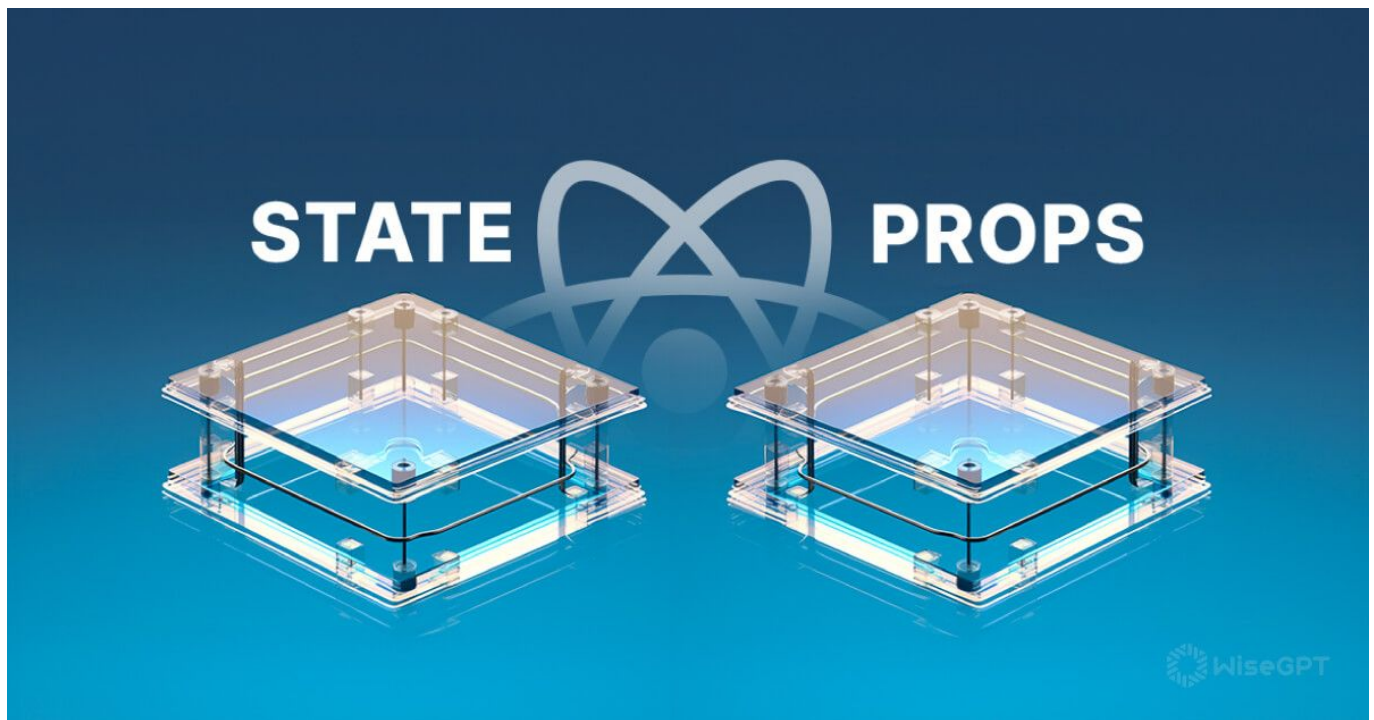


# Các concept cần nắm trong ReactJS

---

## Tổng quan 🎯

1. Conditional Rendering
2. Event Handlers
3. useState hook



# 1. Conditional Rendering

Conditional Rendering là gì? Tùy vào **điều kiện (input)** khác nhau sẽ **render ra UI (output)** khác nhau

## 1.1. Dùng if

```
const StudentList = ({ studentList }) => {  
  if (!Array.isArray(studentList) || studentList.length === 0) return  
  null  
  
  return <div>...</div>  
}
```

```
const App = () => {  
  const [loading, setLoading] = useState(false)  
  
  if (loading) return <p>loading...</p>  
  
  return <div>...</div>  
}
```

## 1.2. Dùng ternary operator (condition ? x : y) (toán tử 3 ngôi)

```
// good to render two values depending on condition  
function Switch({ isOn }) {  
  return <div>{isOn ? 'ON' : 'OFF'}</div>  
}
```

```
// ok to use in case the rendered element is short and can be inline  
function Switch({ isOn }) {  
  return (  
    <div>  
      {isOn ? <p>ON</p> : <div><p>OFF</p></div>}  
    </div>  
  )  
}
```

```
// use null in case you want to render nothing  
function Switch({ isOn }) {  
  return (  
    <div>
```

```

        {isOn ? <p>ON</p> : null}
      </div>
    )
  }

```

### 1.3. Dùng AND operator (&&)

Chúng ta cùng xem qua nhược điểm của cách dưới đây

```

function Switch({ isOn }: SwitchProps) {
  return (
    <div>
      {isOn
        ? (
            <div>
              <p>ON</p>
              <p>ON</p>
              <p>ON</p>
              <p>ON</p>
            </div>
          )
        : (
            <div>
              <p>OFF</p>
              <p>OFF</p>
              <p>OFF</p>
              <p>OFF</p>
              <p>OFF</p>
            </div>
          )
      }
    </div>
  )
}

```

Cách khắc phục

```

function Switch({ isOn }) {
  return (
    <div>
      {isOn && <p>ON</p>}
      {!isOn && <div><p>OFF</p></div>}
    </div>
  )
}

```

Lưu ý

```
function App() {  
  return (  
    <ul>  
      <li>>false: {false}</li>  
      <li>undefined: {undefined}</li>  
      <li>null: {null}</li>  
      <li>Empty string: {''}</li>  
      <li>Zero: {0}</li>  
      <li>NaN: {NaN}</li>  
    </ul>  
  );  
}  
  
export default App;
```

```
// don't  
{arr.length && <div>...</div>}  
// DO  
{arr.length > 0 && <div>...</div>}
```

## Bài tập

- <https://codesandbox.io/p/sandbox/friendly-sara-38vvww>
- <https://codesandbox.io/p/sandbox/autumn-meadow-zcg8z7>

## 2. Event Handlers

Cách thực hiện 1 action trong Javascript thuần

```
const button = document.querySelector('.btn');

function doSomething() {
  // Stuff here
}

button.addEventListener('click', doSomething);
```

Chúng ta có thể nhúng `function doSomething` vào thẻ `button` ở HTML

```
<button onclick="doSomething()">
  Click me!
</button>
```

Cách thực hiện 1 event trong JSX

```
function AlertButton() {
  function handleClick() {
    alert("HELLO!!")
  }

  return (
    <button onClick={handleClick}>
      Click me!
    </button>
  );
}
```

Cách đặt tên

```
const handleClick = () => {}

const handleFocus = () => {}

const handleChange = () => {}

return (
  <form>
    <input placeholder="Name" onChange={handleChange} onFocus=
{handleFocus} />
  </form>
)
```

```
        <button
          onClick={handleClick}
        >
          Submit!
        </button>
      </form>
    );
```

Chú ý:

```
// DO ✅
<button onClick={handleClick}>
  Passing a Reference
</button>

// don't ❌ (call lập tức)
<button onClick={handleClick()}>
  Passing an Invocation
</button>
```

```
// ❌ Invalid: calls the function right away
<button onClick={setTheme('dark')}>
  Toggle theme
</button>

// ✅ Valid:
<button onClick={() => setTheme('dark')}>
  Toggle theme
</button>
```

## Bài tập

- <https://codesandbox.io/p/sandbox/focused-chaum-wp7rhx>
- <https://codesandbox.io/p/sandbox/charming-oskar-58x5xr>

### 3. useState hook

```
function Counter() {  
  let count = 0  
  
  const handleIncreaseCount = () => {  
    count = count + 1  
  }  
  
  return (  
    <button onClick={handleIncreaseCount}>  
      Value: {count}  
    </button>  
  );  
}  
  
export default Counter;
```

Dùng useState hook

```
import React from 'react';  
  
function Counter() {  
  const countArray = React.useState(0);  
  
  const count = countArray[0];  
  const setCount = countArray[1];  
  
  const handleIncreaseCount = () => {  
    setCount(count + 1)  
  }  
  
  return (  
    <button onClick={handleIncreaseCount}>  
      Value: {count}  
    </button>  
  );  
}  
  
export default Counter;
```

Dùng **array destructuring** để tách countArray

```
import React from 'react';  
  
function Counter() {  
  const [count, setCount] = React.useState(0);
```

```
const handleIncreaseCount = () => {
  setCount(count + 1)
}

return (
  <button onClick={handleIncreaseCount}>
    Value: {count}
  </button>
);
}

export default Counter;
```

### 3.1. Cách đặt tên

Chúng ta có thể đặt tên ntn

```
const [hello, world] = React.useState(0);
```

Theo phổ biến, mọi người thường tuân theo quy ước "x, setX":

```
const [user, setUser] = React.useState();
const [errorMessage, setErrorMessage] = React.useState();
const [flowerBouquet, setFlowerBouquet] = React.useState();
```

### 3.2. Initial value (Giá trị khởi tạo ban đầu)

```
const [count, setCount] = React.useState(1);
console.log(count); // 1
```

Chúng tôi cũng có thể cung cấp một chức năng. React sẽ gọi hàm này trong lần kết xuất đầu tiên để tính giá trị ban đầu: Ta có thể đặt vào `useState` 1 `function`.

React sẽ gọi hàm này trong lần kết xuất đầu tiên để tính giá trị ban đầu, và sẽ không gọi lại ở những lần re-render tiếp theo

```
const [count, setCount] = React.useState(() => {
  return 1 + 1;
});

console.log(count); // 2
```



Điều này được gọi là **initializer function**. Đôi khi nó có thể hữu ích nếu chúng ta cần thực hiện một thao tác tốn kém để tính giá trị ban đầu. Ví dụ: đọc từ LocalStorage:

```
const [count, setCount] = React.useState(() => {
  return window.localStorage.getItem('saved-count');
});
```

### 3.3. setState is async

```
function App() {
  const [count, setCount] = React.useState(0);

  return (
    <>
      <p>
        You've clicked {count} times.
      </p>
      <button
        onClick={() => {
          setCount(count + 1);

          // Kết quả dòng này là bao nhiêu
          // A. 0
          // B. 1
          // C. undefined
          // D. None of them
          console.log(count)
        }}
      >
        Click me!
      </button>
    </>
  );
}
```

Cách fix case trên

```
function App() {
  const [count, setCount] = React.useState(0);

  return (
    <>
      <p>
        You've clicked {count} times.
      </p>
      <button
        onClick={() => {
          const nextCount = count + 1;
```

```
        setCount(nextCount);  
        console.log(nextCount)  
      }}  
    >  
    Click me!  
  </button>  
</>  
);  
}
```

## Bài tập

- <https://codesandbox.io/p/sandbox/reverent-khayyam-2sy7vj>
- <https://codesandbox.io/p/sandbox/magical-golick-sj7y93>