**Render với điều kiện trong ReactJS**

Trong bài viết này chúng ta sẽ cùng nhau đi tìm hiểu về cách để render với điều kiện trong ReacJS. Trong React, chúng ta có thể tạo ra các component riêng biệt, và chỉ định thành phần nào được render bằng cách sử dụng biểu thức điều kiện.

Render kèm điều kiện trong React, hoạt động tương tự như Javascript thông thường, bạn chỉ cần sử dụng biểu thức điều kiện if, hay [conditional operator](https://freetuts.net/render-voi-dieu-kien-trong-reactjs-2417.html) trong Javascript, sau đó React sẽ thực hiện nhiệm vụ kiểm tra điều kiện và render theo trường hợp yêu cầu. Ở đây mình có ví dụ về kiểm tra đăng nhập của người dùng:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | import React, { Component } from 'react';    export default class App extends Component {      render() {      var isLogin = true      if(isLogin) {        return (          <div>            <h3>Freetuts.net</h3>          </div>        );      }else{        return(          <div>             <h3>Vui lòng đăng nhập</h3>          </div>        )      }    }  } |

Dưới đây, là một vài cách mà bạn có thể thêm điều kiện render trong các component.

**Mục lục**

* [1. Gán element vào biến](https://freetuts.net/render-voi-dieu-kien-trong-reactjs-2417.html#goto-h2-0)
* [2. Biểu thức điều kiện trong JSX](https://freetuts.net/render-voi-dieu-kien-trong-reactjs-2417.html#goto-h2-1)
* [3. Ngăn chặn component render](https://freetuts.net/render-voi-dieu-kien-trong-reactjs-2417.html#goto-h2-2)

**1. Gán element vào biến**

Chúng ta có thể lưu các element vào một biển. Đây là cách mà bạn có thể tùy chọn các element được hiển thị bằng cách thêm điều kiện vào cho nó. Giả sử, mình sẽ viết lại ví dụ ở trên bằng cách gán element vào biến.

*Bài viết này được đăng tại [free tuts .net]*

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | import React, { Component } from "react";    export default class App extends Component {    render() {      const isLogin = true;        if (isLogin) {        //Gán element vào một biến        var notification = <h3>Freetuts.net</h3>;      } else {        //Gán element vào một biến        var notification = <h3>Vui lòng đăng nhập</h3>;      }        return <div>{notification}</div>;    }  } |

Chúng ta sẽ nhận được kết quả giống như ví dụ ở đầu bài, ngoài ra bạn còn có thể sử dụng cú pháp conditional operator, ở đây mình có ví dụ :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | import React, { Component } from "react";    export default class App extends Component {    render() {      const isLogin = true;      //Conditional operator      const notification = (isLogin) ? <h3>Freetuts.net</h3> : <h3>Vui lòng đăng nhập</h3>        return <div>{notification}</div>;    }  } |

**2. Biểu thức điều kiện trong JSX**

Ngoài cách sử dụng các biểu thức điều kiện bên ngoài hàm return() như ví dụ bên trên, chúng ta còn có thể sử dụng nó trong JSX bằng cách viết nó trong dấu ngoặc {}, chúng ta vẫn sẽ sử dụng ví dụ ở đầu bài:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | import React, { Component } from "react";    export default class App extends Component {    render() {      const isLogin = true;      return (        <div>          {isLogin ? (<h3>Freetuts.net</h3>) : (<h3>Vui lòng đăng nhập</h3>)}        </div>      );    }  } |

Trong JSX bạn chỉ có thể sử dụng cú pháp conditional operator, kết quả hiển thị của ví dụ vừa rồi cũng có kết quả tương tự với ví dụ đầu tiên.

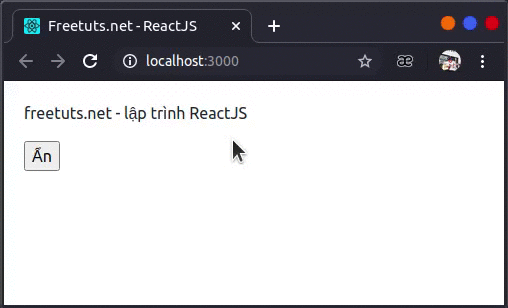
**3. Ngăn chặn component render**

Trong một số trường hợp nào đó, bạn sẽ muốn một component tự ẩn đi dù nó được render trong một component khác. Để làm được điều đó, ta sẽ trả về null thay vì trả về JSX.

|  |  |
| --- | --- |
| 1  2  3  4 | const Demo = (props) => {    //Return về null để không hiển thị    return null  } |

Giả sử mình có một component hiển thị các thông báo:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43 | import React, { Component } from "react";  const Notification = (props) => {    //Kiểm tra giá trị của props    if(props.isShow) {      //Trả về JSX để hiển thị      return (          <ul>            <li>Thông báo 1</li>            <li>Thông báo 2</li>          </ul>      )    }else{      //Trả về null để ẩn      return null    }  }  export default class App extends Component {    constructor(props) {      super(props);      this.state = {        isShowNotification: false      };    }    render() {        const {isShowNotification} = this.state        return (        <div style={{margin: 20}}>          <p>freetuts.net - lập trình ReactJS</p>          <button onClick = {() => {            //Cập nhật lại state            this.setState({              isShowNotification: !isShowNotification            })          }}>{isShowNotification ? 'Hiển thị' : 'Ẩn'}</button>            {/\* Gọi component Notification \*/}          <Notification isShow = {isShowNotification}/>        </div>      );    }  } |



Trên đây chúng ta đã cùng nhau đi tìm hiểu về Render với điều kiện trong ReactJS. Đây là kiến thức rất cơ bản về nó nhưng cũng hết sức quan trọng trong quá trình làm việc với ReactJS sau này. Mong rằng bài viết sẽ giúp ích cho bạn.