* Tạo ứng dụng

npx create-react-app my-app

cd my-app

npm start

* JSX : JS mở rộng

Const user = {

Name: “Loc”,

Email: “nnloc123@gmail.com”

};

//Const element = <h1>My name is {user.Name}, and email: {user.Email}</h1>

Const element = (

<div className=”hello”>

<h1>Hello World</h1>

<h2>My name is {user.Name} and my email: {user.Email}</h1>

</div>

);

ReactDom.render(element, document.getElementById(‘root’));

Var Numbers = [1, 2, 3, 4];

C1: Var doubleNumbers = Numbers.map(function(eachNumber)

{

Return eachNumber \* 2 + “, ”;

});

C2: Var doubleNumbers = Numbers.map((eachNumber) => eachNumber \* 2 + “, “);

ReactDom.render(doubleNumbers, document.getElementById(‘root’));

//Render

C1:

Var counter = 0;

setInterval(function(){

counter +=1;

console.log(“updating…”, counter)

},1000);

C2:

Var count = 0;

Function updateTime(){

Count +=1;

Console.log(“updating …”,count);

}

setInterval(updateTime, 1000);

C3:

Function updateTime(){

Const element = (

<div>

<h1> The timer update example </h1>

<h2> Current time is: {new Date().toLocaleTimeString()}</h2>

</div>

);

ReactDom.render(element, document.getElementById(‘root’));

}

//Prop

Function UserInfo(props){

Return(

<div>

<h1>Name: {prop.name}</h1>

<h1>Email: {prop.email}</h1>

</div>

);

}

Const element = <UserInfo name=”loc” email =”[nnloc123@gmail.com](mailto:nnloc123@gmail.com)”/>

ReactDom.render(element,documentgetElementByIt(‘root’));

Or

ReactDom.render(<UserInfo name=”loc” email=”[nnloc123@gmail.com](mailto:nnloc123@gmail.com)”/>, documentgetElementById(‘root’));

//Components

Class UserInfo extends React.Component{

Render(){

Return(

<div>

<div>

<h1>Name: {this.props.name}</h1>

<h1>Email: {this.props.email}</h1>

</div>

<div>

<h1>Information: {this.props.info}</h1>

</div>

</div>

);

}

}

//Separate div

Class User1 extends React.Component{

Render(){

Return(

<div class=”User1”>

<h1>Name: {this.props.name}</h1>

<h1>Email: {this.props.email}</h1>

</div>

);

}

}

Class User2 extends React.Component{

Render(){

Return(

<div class=”User2>

<h1>Information: {this.props.info}</h1>

</div>

);

}

}

Class UserInfo extends React.Component{

Render(){

Return(

<div>

<User1 name={this.props.name} email={this.props.email}/>

<User2 info={this.props.info}/>

</div>

);

}

}

//State va Cycle

class Counter extends React.Component {

constructor(props){

super(props);

this.state = {seconds: 0};

}

incrementCounter(){

this.setState(

(prevState, props) => ({

Seconds: prevState.seconds + 1

})

);

}

componentDidMount(){

this.timeID = setInterval(() => this.incrementCounter(), 1000);

}

componentWillMount(){

clearInterval(this.timeID);

}

Render(){

Return(

<div>

<h1>This is a counting machine!</h1>

<h2>Second: {this.state.seconds}</h2>

</div>

);

}

ReactDOM.render(<Counter/>,document.getElementById(‘root’);

//Handling Events

Class ToggleButton extends React.Component{

Constructor(props){

Super(props);

This.state = {isOn: true};

C1: This.buttonOnClick = this.buttonOnClick.bind(this);

}

C1:

buttonOnClick(){

this.setState(prevState => ({

isOn: !prevState.isOn

}

));

}

C2:

buttonOnClick = () => {

this.setState(prevState => ({

isOn: !prevState.isOn

}

));

}

Render(){

Return(

<button className=”toggleButton” onClick={this.buttonOnClick}>

This is toggle button

{this.state.isOn ? “turning On” : “turning Off”}

<button/>

);

}

}

ReactDom.render(<ToggleButton/>,document.getElementById(‘root’));