**MỤC LỤC**

[Components 2](#_Toc155275625)

[Funciton 2](#_Toc155275626)

[render() 2](#_Toc155275627)

[console.log(a,b) 2](#_Toc155275628)

[filter() 2](#_Toc155275629)

[setTimeout : 2](#_Toc155275630)

[ComponentDidMount(){} 2](#_Toc155275631)

[Các định nghĩa 2](#_Toc155275632)

[export 2](#_Toc155275633)

[return 3](#_Toc155275634)

[State 3](#_Toc155275635)

[button 3](#_Toc155275636)

[input 4](#_Toc155275637)

[Form 5](#_Toc155275638)

[Prop 5](#_Toc155275639)

[Biến 6](#_Toc155275640)

[Mảng : 6](#_Toc155275641)

[Truyền function từ hàm cha sang hàm con 7](#_Toc155275642)

[Khoảng trắng : 8](#_Toc155275643)

[npm toastify 8](#_Toc155275644)

[Cú pháp : 8](#_Toc155275645)

[<br/> 8](#_Toc155275646)

# Components

Có 2 loại :

* class component
* function component : (function , array)

Tên component viết hoa chữ đầu . Vd :class Mycomponent extends React.Component{}.

# Funciton

Khai báo hàm : Ví dụ : funtion App() {} = const App = () => {}

## render()

Là component mà muốn render ra dữ liệu ( dữ liệu trả ra )

## console.log(a,b)

Là hàm dùng để hiển thị thông điệp hoặc giá trị trên console của trình duyệt

## filter()

Lọc có điều kiện

Ví dụ : lọc các phần tử có length > 6 và gán vào result

const word = ['sparay', 'limit', 'elite', 'exubearant', 'destruction', 'present']

const result = word.filter(word => word.length > 6)

## setTimeout :

Hàm này sẽ set delay

Ví dụ: set consolog sau 3 giây sẽ hiện

    setTimeout(() => {

            console.log('check time out')

        }, 3000);

## ComponentDidMount(){}

Trong quá trình lifecycle , khi một file .js được gọi , nó sẽ chạy hàm render(){} trước sau đó mới tới ComponentDidMount(){} , sau đó nó sẽ chạy luôn phiên render(){} => ComponentDidMount(){}

# Các định nghĩa

## export

* export default : chỉ trả về một class hoặc component export default mycomponent .
* export {} : trả về nhiều class/ component . Vd : export { Mycomponent1 , Mycomponent2} .

## return

chỉ trả về 1 div . Nếu muốn dùng 2 div thì chèn <> </> vào code . Ví dụ :

    render() {

        return (

            <>

                <div className='first'>

                    Hello , my component of {this.state.name}

                </div>

                <div className='second'>

                    My page is : {this.state.page}

                </div>

            </>

        );

    }

## State

Là trạng thái , là dữ liệu có thể thay đổi được

Là object ( có key : value) Ví dụ :

    state = {

        name: 'Tuan',

        page: 'NAT'

    }

Mỗi lần thay đổi thì giao diện được update lại ( không phải refresh lại page)

đ

Để lấy giá trị trong state dùng câu lệnh . Ví dụ :

{this.state.name}

## button

Ví dụ :

<button>Click me</button>

Ví dụ :

Khi ấn Click me thì sẽ có bảng alert hiện ra :

handleClickButton = () => {

        alert('Click me');

    }

    render() {

        return (

            <>

                <div className=first>

                    <button onClick={() => this.handleClickButton()}>click me</button>

                </div>

            </>

        );

    }

## input

Các loại input :

A screenshot of a computer

Description automatically generated

Ví dụ :

<input value={this.state.name} type=”text”

Ví dụ :

class MyComponent1 extends React.Component {

    state = {

        name: 'Tuan',

        page: 'NAT'

    }

    handleOnChangeName = (event) => {

        this.setState({ name: event.target.value });

    } //Cú pháp thay đổi state

    render() {

        return (

            <input value={this.state.name} type="text"

                onChange={(event) => this.handleOnChangeName(event)}

            />

        );

    }

}

## Form

<form action="/action\_page.php">

                        <label htmlFor="fname">First name:</label><br />

                        <input type="text"

                            value={this.state.firstName}

                            onChange={(event) => this.handleChangeFirstName(event)}

                        />

                        <br />

                        <label htmlFor="lname">Last name:</label><br />

                        <input type="text"

                            value={this.state.lastName}

                            onChange={(event) => this.handleChangeLastName(event)}

                        />

                        <br />

                        <input type="submit"

                            value="Submit"

                            onClick={(event) => this.handleSummit(event)}

                        />

</form>

## Prop

Bản chất là biến

Truyền data từ cha xuống con

Ví dụ :

MyComponent.js :

class MyComponent extends React.Component {

    render() {

        return (

            <>

                <ChildComponent

                    name={'child one'}

                    age={'22'}

                />

            </>

        );

    }

}

ChildComponent.js :

class ChildComponent extends React.Component {

    render() {

        console.log('check props: ', this.props)

        let name = this.props.name;

        let age = this.props.age; /\* = let {name,age} = this.props; (tên biến bằng tên key của props) \*/

        return (

            <>

                <div className='first'>

                    Child component name : {name} - {age}

                </div>

            </>

        );

    }

}

## Biến

let name = ‘Tuan’ : tự định nghĩa name là string => dùng biến : <div>My name is {name}</div>

## Mảng :

Khai báo :

 arrJob: [

            { id: 'abc001', title: 'Dev' },

            { id: 'abc002', title: 'Tester' },

            { id: 'abc003', title: 'Tester' }

        ]

In mảng :

<div className='arrJob'>

                    {

                        arrJob.map((item, index) => {

                            return (

                                <div key={item.id}>

                                    {item.id} - {item.title}

                                </div>

                            );

                        })

                    }

</div>

## Truyền function từ hàm cha sang hàm con

Ví dụ :

Hàm cha :

addNewJob = (job) => {

        console.log('check job from parent: ', job)

        this.setState({

            arrJob: [...this.state.arrJob, job]

        })

    }

render() {

        return (

            <>

                <AddComponents

                    addNewJob={this.addNewJob}

                />

            </>

        );

    }

Hàm con :

 handleSubmit = (event) => {

        event.preventDefault();

        console.log('>>> check data : ', this.state)

        if (!this.state.title || !this.state.salary) {

            alert('Please input data')

            return;

        }

        this.props.addNewJob({

            title: this.state.title,

            salary: this.state.salary

        })

        this.setState({

            title: '',

            salary: ''

        })

    }

## Khoảng trắng :

&nbsp hoặc <></>

## npm toastify

Hiện bảng thông báo bên phải

# Cú pháp :

## <br/>

Là hiển thị xuống dòng