constructor(props){

super(props)

// Hàm này Thực hiện việc thiết lập state cho component

// Việc sử dụng super(props) là để có thể sử dụng this.props trong phạm vi hàm constructor này

}

\* componentWillMount() :thực hiện duy nhất một lần

\* componentDidMount() : thực hiện duy nhất một lần

\* componentWillUnmount() :

+thực hiện duy nhất một lần

+hữu dụng khi xoá các timer không sử dụng

\*componentWillReceiveProps(nextProps):

+thực hiện liên tục khi props thay đổi

+ Sử dụng để thay đổi trạng thái (state) của component phụ thuộc props

\*shouldComponentUpdate(nextProps,nextState):

+Hàm này thực hiện khi state và props thay đổi

+ Hàm này sẽ trả về kết quả true/false, bạn sẽ cần sử dụng đến hàm này để xử lý xem có cần update component không

\*componentWillUpdate(nextProps,nextState):

+Hàm này thực hiện khi shouldComponentUpdate true,nếu fale thì react không trả vể hàm này

\*componentDidUpdate(nextProps,nextState):

+Hàm này thực hiện khi componentWillUpdate được render lại từ kết quả componentWillUpdate

Khởi tạo component:

Lần lượt các hành động sau để khởi tạo component:

+khỏi tạo class (đã kế thừa từ component của react)

+ khởi tạo các giá trị mặc định cho props (default props)

+khởi tạo các giá trị mặc định cho state (constructor)

+Gọi componentWillMount()

+Gọi render()

+Gọi componentDidMount()

State thay đổi:

+Gọi hàm shouldComponentUpdate()->default:true

+Gọi componentWillUpdate()-với điều kiện là componentShouldUpdate()->true.

+Gọi reder()

+Gọi componentDidMount()

Props thay đổi:

+Gọi hàm componentWillReceiveProps()

+Gọi hàm componentShouldUpdate()

+Gọi hàm componentWillUpdate()

+Gọi render()

+Gọi componentDidMout()

Khi unmount Component:

+Goi componentWillUnmount()

\*Chú ý:

-this.state khởi tạo một lần duy nhất muốn thay dổi phải dùng setState(). Khi xài setState() ta đã thực hiện một vòng đời của component: (shouldComponentUpdate, componentWillUpdate, render, componentDidUpdate). Và cũng luôn nhớ rằng, this.setState là hàm async, nên truy cập this.state ngay sau khi setState sẽ không nhận được giá trị mới của this.state.

+setState là hàm async

+setState luôn reder trừ khi shouldComponentUpdate() ->false

+ để tránh render thì ta dùng componentWillReceiveProps(nextProps), nếu nextprops giống với props hiện tại thì không render.