Công cụ: Visual Studio Code, git, Nodejs

FakerJs, Semantic-ui.com

Plug-in: reactjs code snippet

Npm install –g create-react-app

Create-react-app *tenApp*

Cd tenApp

Npm start

Các Chú ý:

\_ Dùng Function để tạo Component khi có nội dung đơn giản, không giải quyết vấn đề logic, cập nhật nội dung. Ngược lại thì dùng Class để tạo Component.

\_ Benefits of Class component:

+ can use State -> easier to handle user input.

+ understanding lifecycle events -> easier to do things when the app starts .

+ Easier code organization.

\_ Các quy luật của Class Component:

+ Must be a Javascript class.

+ Must extend (subclass) React.Component.

+ Must define a ‘render’ method that return some amount JSX.

\_ LifeCycle:

+ Constructor: Good place to do one-time setup.

+ Render: Avoid doing anything besides returning JSX.

+ ComponentDidMount: Good place to do data-loading.

+ ComponentDidUpdate: Good place to do more data-loading when state/props change.

+ ComponentWillUnmount: Good place to do cleanup .

\_ 2 cách khởi tạo state:

+ Cách 1:

state = {lat: null, errorMessage: ''};

+ Cách 2: như cũ, khởi tạo trong Constructor method.

SEMANTIC UI:

\_ thêm CDN vào html: https://cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.4.1/semantic.min.js

COMPONENT:

\_ Là 1 khối được đóng gói, gồm các html, css, state, props.

\_Do mình tạo ra.

\_ Dùng để đóng gói thành 1 gói độc lập, nó là 1 thế giới nhỏ .

\_Cách tạo, 4 cách(chỉ được return về 1 div, ko trả về các div ngang hàng nhau):

+ Kiểu **bình thường**: function One();

+ Kiểu **anonymous function**: function();

+ Kiểu **arrow function**: () =>{};

+ Kiểu **Class.**

**(**

**Cách 1:**

function One() {

  return (

    <div>

      <h1>Dòng số 1</h1>

      <h2>Dòng số 2</h2>

    </div>

  )

}

**Cách 2: Anonymous function**

var Two = function(){

  return (

    <div>

      <h3>CaCH1 SỐ 2</h3>

      <h3>CaCH1 SỐ 2</h3>

    </div>

  )

}

**Cách 3: Arrow function**

var Three = () => (<div><div>Cách 3</div><div>Cách 3</div></div>)

**Cách 4: Class (**phím tắt: rcjs -> tab**)**

class App extends Component {

  render() {

    return (

      <div>

        cách 4

      </div>

    );

  }

}

**)**

PROPS:

(what)

\_ là thuộc tính của COMPONENT do mình truyền vào.

(how) Cách thao tác với props:

CÁCH 1 : Dùng function:

var Two = function(props){

  return (

<div>

    <img src = {props.anh}></img>

    <div>{props.tieude}</div>

  </div>

  )

}

 <Two tieude= "tieu đề 1" anh = "https://znews-photo.zadn.vn/w210/Uploaded/ngotgs/2019\_09\_17/truc\_1.jpg"></Two>

CÁCH 2 : Dùng Class (thường hơn )

class Four extends Component {

  render() {

    return (

      <div>

         <img src = {this.props.anh}></img>

         <div>{this.props.tieude}</div>

      </div>

    );

  }

}

 <Four tieude= "tieu đề 1" anh = "https://znews-photo.zadn.vn/w210/Uploaded/ngotgs/2019\_09\_17/truc\_1.jpg"></Four>

JSX:

\_ Cho phép mình viết HTML trong javascript.

\_ Gần giống với HTML.

\_ Viết file HTML ra rồi chuyển đổi qua JSX thông qua phần mềm trên google.

Hàm MAP():

\_ Duyệt từng phần tử (FOREACH) của mảng.

  const so = [1,2,3,4,5];

  const so1 = so.map((x) => (

    <div>

      <ul>

        <li>{x \* 2}</li>

      </ul>

    </div>

 <div>{so1}</div>

THỰC HÀNH “Component hóa ” GIAO DIỆN HTML bằng ReactJs:

Bước 1: Viết file HTML hoàn chỉnh.

Bước 2: Chuyển qua ReactJs.

TƯƠNG TÁC TRONG REACTJS:

\_ 4 lưu ý khi viết hàm:

+ 1. onClick => C viết hoa.

+ 2. Dùng dấu {} thay vì “”.

+ 3. Dùng từ khóa This.

+ 4. Hàm ko có tham số thì ko cần dùng dấu ().Có tham số thì viết bình thường.

 thongbao = () => {alert("thong baos na")};

<button onClick={this.thongbao}>CLICK</button>

\_ Viết tương tác có tham số:

+ Dùng hàm binding(): this.ham.bind(this, thamso);

 Thongbao2 = (x) => {alert(x)};

<button onClick={this.thongbao2.bind(this, "hoho")}>CLICK3</button>

+ Dùng arrow function(){}.

 Thongbao2 = (x) => {alert(x)};

 <button onClick={() => this.thongbao2("sad")}>CLICK2</button>

STATE:

(WHAT)

\_ là từ khóa, 1 object.

\_ là biến trung gian giữ các biến, tham số cần thiết trong quá trình thao tác với COMPONENT.

\_ chỉ giới hạn trong 1 COMPONENT (private).

(HOW)

\_ TẠO STATE:

 this.state{

       trangthai: 0

     }

\_ ĐỔI STATE:

this.setState({

     trangthai : 1

   });

REF:

\_ Cho phép không cần render lại tham số như State.

REACT ROUTER:

(WHAT)

\_ Cho phép chuyển hướng website

(WHY)

\_ Thao tác chức năng mà không cần load lại trang.

(HOW) 5 bước:

\_ Bước 1: Tạo giao diện.

\_ Bước 2: Điều hướng kiểu SPA.

\_ Bước 3: Truyền tham số.

\_ Bước 4: Dựng Slug cho url bằng React Router.

\_ Bước 5: Router Redirect.