**React**

**Bài: 0:** Thư viện jq dùng để xây dựng giao diện người dùng. Web app những công cụ phức tạp

React làm ui thôi.

**Bài 1:single page app**

Render sử lí ở phía server rồi

Single age app:

* Được facebook phát triển .
* Request đầu tiên nhận về code HTML . những cái tiếp theo thì nó request data lên server

Server gửi về json.

* Instagram gửi request lên đầu tiên: chuyển sang trang khác url nó thay đổi nhưng nó không readload lại toàn bộ trang, request ban đầu không thay đổi , trừ khi enter link mới request lại.

**Bài 2: Cài đặt**

CDN links.

**Bài 3: React.createElement**

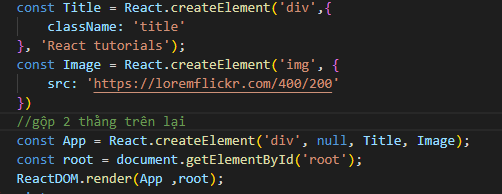
Nhận vào một 2 hoặc nhiều tham số:

tham số 1 kiểu của element mà bạn muốn tạo có thể lấy trong html element.

Tham số 2 thuộc tính.

Tham số 3 nội dung.

Atributte trong html là props trong Reacjs.



**Bài 4: JSX**

Viết component viết jsx dễ dàng hơn.

Chuyển đổi cú pháp jsx

Tool babeljs.io: dịch chuẩn js mới về cái cũ hơn.

**Bài 5: Create React App**

Công cụ để tạo React App <https://create-react-app.dev/docs/getting-started>

**Bài 6: Component**

Sinh ra để viết ra ui có nhiều cái thẻ khác nhau

Cách khai báo class để tạo component.

* Component cái hay là có thể nhóm được nhiều thứ vào bên trong.
* Có thể tái sử dụng lại được ui trong app của chúng ta.

**Tạo thư mục componet trong src: TodoItem.js**

import React from 'react';

function TodoItem (){

    return(

        <div className="TodoItem">

            <p>Đi chợ</p>

        </div>

    );

}

export default TodoItem;

**App.js**

import TodoItem from './component/TodoItem';

* **Gọi ra trong trang App.js: <TodoItem />.**

**Bài 7: Props ( Props của 1 Component cũng giống như các tham số đầu vào của 1 hàm, render method là đầu ra )**

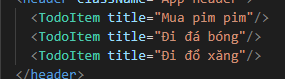
**Deploy: hành động đưa code lên server**

Thay đổi thông số trong component cho nó linh hoạt hơn.

Các component đều có thể nhận giá trị props trong đó.

* Truyền biến thì phải **{object hoặc biến}.**

**App.js**

* 

**Todolist.js**

function TodoItem (props){

    return(

        <div className="TodoItem">

            <p>{props.title}</p>

        </div>

    );

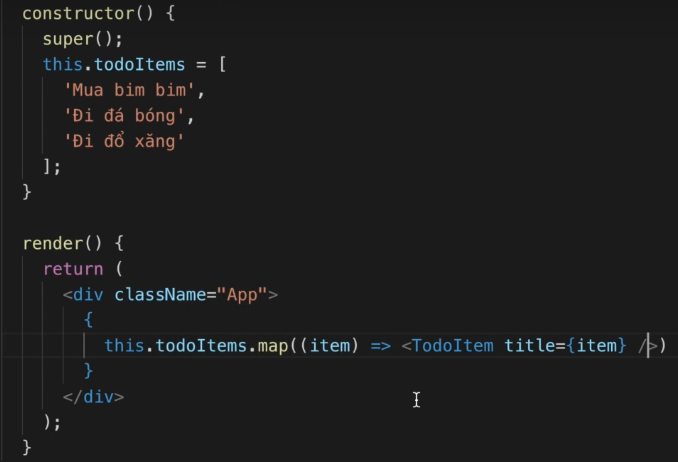
}

**Bài 8: Render a list**

**Bất kì chỗ nào dùng máp để biến từ 1 Array của Object => Array các component phải thêm key vào cho nó.**

**App.js**

**Triển khai bằng Components:**



* Lưu ở constructor: **supper():** của thằng này kế thừa là **Component**

**Function:**

function App() {

  const TodoItems = ['Mua bim bim', 'Đi đá bóng', 'Đi đổ săng'];

  return (

    <div className="App">

      <header className="App-header">

       {

         TodoItems.map((item, index) => <TodoItem key={index} title = {item} />)

       }

      </header>

    </div>

  );

}

**Bài 9: React Developer Tools ( debug React App )**

Cài đặt tool này vô để xem code react.

**Bài 10: Dynamic class names( thêm nhiều class một cách linh hoạt cho components ).**

Thêm nhiều class cho componets của chúng ta.

Ví dụ:thêm class vào element: App.js // còn 1 cách nữa là module: classnames.

function App() {

  const TodoItems = [

    { title: 'Mua bim bim', isComplete: true},

    { title: 'Đi đá bóng', isComplete:true},

    { title:'Đi đổ săng'}

  ];

  return (

    <div className="App">

      <header className="App-header">

       {

         TodoItems.map((item, index) =>

         <TodoItem key={index} item= {item} />

         )

       }

      </header>

    </div>

  );

}

Todo-list.js:

function TodoItem (props){

    const { item } = props;

    let className = 'TodoItem';

    if(item.isComplete){

        className += ' TodoItem-complete'

    }

    return(

        <div className={className}>

            <p>{props.item.title}</p>

        </div>

    );

}

**Bài 11: Conditional rendering.( Nhiều cách render theo điều kiện )**

Ví dụ: không có danh sách thì sẽ hiển thị thông báo không có sản phẩm.

Cách 1 if else.

const TodoItems = [

    { title: 'Mua bim bim', isComplete: true},

    { title: 'Đi đá bóng', isComplete:true},

    { title:'Đi đổ săng'}

  ];

  if(TodoItems.length > 0 )

  {

    return (

      <div className="App">

        <header className="App-header">

         {

           TodoItems.map((item, index) =>

           <TodoItem key={index} item= {item} />

           )

         }

        </header>

      </div>

    );

  }

  else{

    return(

      <div className="App">

        Nothing here!

      </div>

    );

  }

Cách 2: ví dụ:

   {todoItems.length > 0 && todoItems.map((item, index) =>

         <TodoItem key={index} item= {item} />

         )}

        {todoItems.length === 0 && 'Nothing here!'}

* true && 6 // kết quả ra là 6.
* False && 6 // kết quả sẽ ra là false.

**Bài 12: State( Là trạng thái bên trong một component )**

* S