Este ejemplo fue tomado integro de <https://www.youtube.com/watch?v=Dorf8i6lCuk&t=771s> un video de la empresa

[Academind](https://www.youtube.com/channel/UCSJbGtTlrDami-tDGPUV9-w) . Solo se tradujo al español y se redactó con el humilde entendimiento de quien escribe este pequeño artículo. Ojala le sirva de ayuda a alguien.

Muchas gracias a Maximilian por su trabajo.

# React

React es un frameword para aplicaciones web From-End. Este frameword cambia la perspectiva de lenguaje java-script de imperativo a declarativo. Esto lo logra precompilando todo el proyecto y generando la estructura html css y javascript que correrá en el navegador.

## ReactDOM.render

Este es el punto de entrada de toda aplicación react.

Ejemplo:

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

ReactDOM.**render**(<**App** />,document.**getElementById**('root'));

El primer argumento trata del componente React que se mostrara, exactamente en el bloque root de la página public/index.html

<body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div *id*="root"></div>

</body>

## Componente

Cada componente React comienza con letra mayúscula y se le llama en el lenguaje JSX como a una etiqueta html, <**App** />, <**App** ></**App** >.Puede llamrcele de ambas formas como etiqueta simple o doble.

### Definición de un componente

Un componente es una función de javascript que comienza con la letra mayúscula y devuelve código JSX, que a su vez puede estar formado por otros componentes.

import Todo from './component/Todo';

function **App**() {

  return(

<div>

    <h1>My Todos</h1>

    <**Todo** *text*='Learning react'/>

    <**Todo** *text*='CUrse master react'/>

    <**Todo** *text*='Other Curse Doctor react'/>

   </div>) ;

}

export default **App**;

### Props

Este es otro concepto importante de react, los props son parámetros que se pasan entre componentes.

Ejemplo 1:

<**Todo** *text*='Learning react'/>

Cuando se invoca a un componente en el lenguaje JSX todo lo que se pase dentro de las etiquetas de apertura y cierre de html se pasa como parámetros en formato JSON. De esta forma se comunican los componentes pasándose referencias a funciones que usaran en sus eventos u otros parámetros incluso otros componentes.

function **Todo**(props){

    return (

        <div *className*='card'>

            <h2>{props.text}</h2>

             <div *className*='actions'>

                 <button *className*='btn' >DELETE</button>

            </div>

        </div>

        );

}

En la línea <h2>{props.text}</h2>una vez compilado el código JSX se obtendrá la cadena 'Learning react'

Ejemplo 2:

En este ejemplo pasamos en los parámetros props.onCancel y props.onConfirm referencias a la función closeModal.

function **closeModal**(){**//Cerramos la ventana modal**}

<**Modal** *onCancel*={**closeModal**} *onConfirm*={**closeModal**}/>

En la definición del componente Modal nos encargamos de asignar esas referencias a los eventos onClick de los botones Cancel y Confirm.

function **Modal**(props){

     return (

     <div *className* ='modal'>

      <p>Are you sure?</p>

      <button *className*='btn btn--alt' *onClick*={ props.**onConfirm**}>Cancel</button>

      <button *className*='btn' *onClick*={ props.**onCancel** }>Confirm</button>

   </div>);

}

export default **Modal**;

### Estados

React solo visualiza o renderiza o ejecuta el código relacionado con un componente dado si se le informa que su estado ha cambiado, en otro caso solo lo visualiza una única vez al inicio de la aplicación. Para conseguirlo nos provee

import {useState} from 'react'

En la definición del componente lo declararíamos como una matriz

function **Todo**(props){

    const [modalIsOpen,**setModalIsOpen**]=**useState**(false);

Estos 2 parámetros, el primero de ellos el estado actual y el segundo la función a través de la cual le cambiaremos el estado al componente. El parámetro pasado **useState**(false) significa que false será el valor inicial de modalIsOpen. El tipo de valor con el cual se actualizará el estado del componente puede ser cualquiera, cadena, objeto, en este caso es un valor booleano.

## Explicando el ejemplo Completo

Solo será una aplicación react sencilla, en la que se muestran tres tarjetas, cada una de ellas con un botón delete, como se muestra en la siguiente imagen.



Cuando se le da click a uno de los botones delete de cualquiera de las tarjetas nos muestra la siguiente imagen.



Una vez demos clic en los botones cancel, Confirm o fuera de la ventana modal nos llevaría a la vista anterior.

### Backdrop

function **Backdrop**(props)

{

    return (<div *className*='backdrop' *onClick*={props.onCancel}/>);

}

export default **Backdrop**;

La función de este componente es una superposición que no permita acceder a las tarjetas de manera que se cree la sensación de una ventana modal. Se le pasa la referencia de la función que debe ejecutar cuando se de click en este componente a través de los props. A Continuación la clase CSS asociada a este componente.

.backdrop{

  position:fixed;

  z-index: index 1;

  background-color: **rgba**(0,0,0,0.75);

  width: 100%;

  height: 100vh;

  top:0;

  left:0;

}

### Modal

La ventana modal que surge al dar click en los botones delete de las tarjetas, se usan referencias de funciones pasadas por los props igual al componente anterior para los eventos. Fuera de esto no tiene mayor interés.

function **Modal**(props){

    function **onConfirm**()

    {

        props.**onConfirm**();

    }

    function **onCancel**(){

        props.**onCancel**();

    }

     return (

     <div *className* ='modal'>

         <p>Are you sure?</p>

         <button *className*='btn btn--alt' *onClick*={**onConfirm**}>Cancel</button>

         <button *className*='btn' *onClick*={**onCancel**}>Confirm</button>

     </div>);

}

export default **Modal**;

.btn{

font: inherit;

padding: 0.5rem 1.5rem;

cursor:pointer;

border-radius: 4px;

background-color: #800040;

color: white;

border: 1px solid #800040;

margin: 0 1rem;

}

.btn:hover{

  background-color: #9c1458;

  border-color: #9c1458;

}

.btn--alt{

  background-color:transparent;

  color: #800040;

}

.btn--alt:hover{

  background-color: #f8dae9;

}

### Todo

import {useState} from 'react'

import Modal from '../component/Modal';

import Backdrop from '../component/Backdrop';

function **Todo**(props){

    const [modalIsOpen,**setModalIsOpen**]=**useState**(false);

    function **deleteHandler**(){

**setModalIsOpen**(true);

    }

    function **closeModal**(){

**setModalIsOpen**(false);

    }

    return (

        <div *className*='card'>

            <h2>{props.text}</h2>

             <div *className*='actions'>

                    <button *className*='btn' *onClick*={**deleteHandler**}>DELETE</button>

            </div>

            {modalIsOpen&& <**Modal** *onCancel*={**closeModal**} *onConfirm*={**closeModal**}/>}

            {modalIsOpen&& <**Backdrop** *onCancel*={**closeModal**}/>}

        </div>

        );

}

export default **Todo**;

### App

import Todo from './component/Todo';

function **App**() {

  return(<div>

    <h1>My Todos</h1>

    <**Todo** *text*='Learning react'/>

    <**Todo** *text*='CUrse master react'/>

    <**Todo** *text*='Other Curse Doctor react'/>

  </div>) ;

}

export default **App**;

### Index.js

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

ReactDOM.**render**(<**App** />,document.**getElementById**('root'));

\* {

  box-sizing: border-box;

}

body{

  font-family: sans-serif;

  margin: 3rem;

  background-color: #dfdfdf;

}

h1 , h2{

  color:#333333;

}

### Index.html

 <title>React App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div *id*="root"></div>

  </body>