

JSX ReactJS

Introducción

JSX es una extensión de `JavaScript` creada para la librería React que nos permite combinar `JS` y `XML` convirtiendo etiquetas en elementos reactivos, de ahí el nombre JavaScript XML.

Esta extensión nos permite escribir un código más limpio, con pocos factores y sin tantas repeticiones.

En estos dos ejemplos se puede ver la simplificación del uso de JSX a la hora de programar aplicaciones en React:

```
//Sin JSX

const myName = React.createElement('h2', {}, 'Luke Skywalker');

ReactDOM.render(myName, document.getElementById('root'));

//Con JSX

const myName = <h2>Luke Skywalker</h2>

ReactDOM.render(myName, document.getElementById('root'));
```

¿Cómo funcionan las expresiones JSX?

En el siguiente ejemplo vamos a declarar una variable llamada `hero` que utilizaremos mediante el uso de llaves.

```
const hero = 'Spider-Man';
const greetings = <h1>Yo soy {hero}</h1>
```

Dentro de las `llaves` de JSX se puede escribir cualquier expresión de JavaScript.

```
const carWheels = <p>Cars have {2 + 2} wheels</p>
```

Hasta ahora hemos visto ejemplos sencillos que contienen una sola etiqueta de HTML, pero JSX nos permite insertar un bloque de mayor tamaño de HTML.

Para esto hay que tener en cuenta que todo el código HTML que sobrepase un único elemento deberá ir anidado entre paréntesis que incluyan un elemento padre, como el `div` en el siguiente ejemplo:

```
const myHeader = (  
  <div>  
    <h1>This is the title</h1>  
    <h2>This is the subtitle</h2>  
    <p>This is some text</p>  
  </div>  
)
```

Al ser HTML una parte fundamental de esta sintaxis también podremos añadirle `clases` a las etiquetas que contenga nuestro código.

La única diferencia en JSX es que utiliza el término `className` en lugar de `class`:

```
const myName = <h2 className="heroname">Luke Skywalker</h2>
```

Los condicionales también tienen cabida en React, pero no se pueden declarar dentro de nuestro elemento JSX.

Para poder hacer uso de condicionales tendremos que colocarlos fuera del elemento y llamarlos con las llaves anteriormente mencionadas:

```
const x = 10;  
  
const time = <p>{x < 12 ? "Good morning" : "Good afternoon"}</p>
```

Mini-ejercicios

Ha llegado el momento de ponerse a trabajar con ReactJS, para ello os proponemos una pequeña práctica que os ayude afianzar el funcionamiento de `JSX`.

1. Crea una aplicación de ReactJS con vite → name: jsx-basics.
2. Renderiza “Buenos días” [6-12] , “Buenas tardes” [13-19] o “Buenas noches”[20-5] según el valor numérico asignado.
3. Recorrer los elementos de un array y renderizarlos ⇒ Si os da un error de keys en la consola podéis usar el index como `key={index}` .
4. Mapea un array de objetos en los que transformes alguna de sus propiedades y renderiza los elementos.
5. Crea un botón que modifique un valor de false a true y renderice un contenido cuando este valor se modifique.