Key props y .map()

DigitalHouse >
Coding School

Índice

1 | .map()

.map()

Observemos cómo el array es pasado dentro del atributo con nombre **items** y que adicionalmente va entre llaves. Son estas llaves las que nos permiten escribir tipos de datos de JavaScript que no sean simplemente cadenas de texto. Así es como se define el valor de una props a través de un **Array**.

{}

```
const usuarios = ["Maru","Uriel","Daniel"];
<MiLista
     items = { usuarios }
     />
```

En este ejemplo podemos observar que estamos pasando al componente **MiLista**, los usuarios que se encuentran en el array **usuarios** y los mismos son asignados a la variable **items**.

.map()

Dentro del **componente**, lo primero que tenemos que hacer es recibir las **props** como parámetro de la función. Luego, dentro de la estructura de JSX que hayamos definido, vamos a tener que iterar sobre el array recibido para poder imprimir los usuarios.

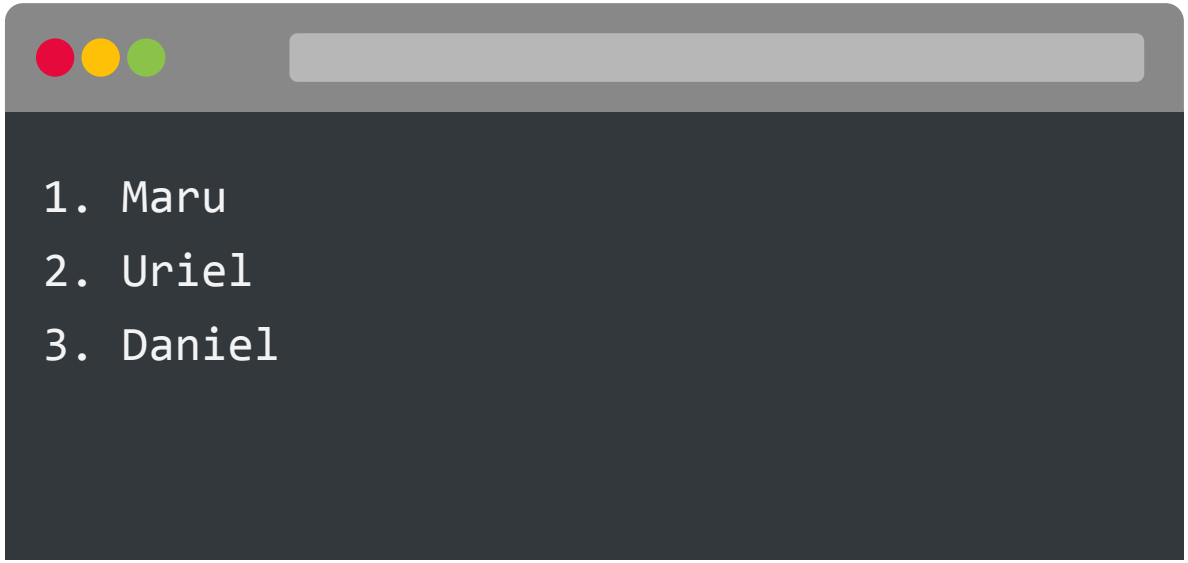
{}

```
function MiLista(props) {
 return (
   <div>
     <ol>
       {props.items.map(item => <li> {item} </li>)}
     </ol>
   </div>
 );
}
```

El **método** que conviene implementar para iterar sobre el array e imprimir su contenido en React es **map()**.

.map()

Así se vería en el navegador. Debemos estar atentos al error que nos aparece en las Dev Tools. ¡Vamos a solucionarlo!



▶Warning: Each child in an array or iterator should have a unique "key" prop.

2 | Key props

Las **keys** ayudan a React a identificar qué elementos se han modificado, agregado o eliminado.

Key props

React, por medio de las keys, determina si es el mismo elemento o no. Es la manera que se **identifican** nuestros componentes en el proyecto.

A la hora de usar keys, tengamos lo siguiente en cuenta:

- Solo es necesario agregar keys cuando **devolvemos un array de elementos iguales**.
- La key debe ser **única entre elementos hermanos**.
- Las keys **no se muestran en el HTML** final (si quisiéramos que se muestren, también deberíamos utilizar id).

Key props

Los componentes que renderizan varios elementos del mismo tipo necesitan de una **key** con valor único, porque la misma ayuda a React a identificar qué **items** cambiaron, cuáles se agregaron o cuáles se eliminaron.

{}

```
function MiLista(props) {
 return (
  <div>
   <ol>
    {props.items.map((item, i)=><li key={i+item}> {item}</li>)}
   </ol>
  </div>
 );
}
```

Las **keys** deben ser dadas a los elementos dentro del mapeo del **array** para darle a los elementos una identidad única y estable.

Documentación

Para saber más podemos acceder a la documentación oficial de React haciendo clic en el siguiente link.


DigitalHouse>
Coding School