



Certified Tech Developer

The Ultimate Degree

Front End III

Tips de ayuda

Tips para CSS

- Con módulos de CSS, cada componente puede tener su propio archivo CSS (o SCSS). Normalmente el archivo CSS reside en el mismo directorio donde reside el componente para un acceso rápido y fácil.
- La importación CSS como módulos es posible gracias a webpack.
- La forma en que se compondrán los nombres modificados de las clases se debe hacer en la configuración de webpack.
- Cada componente importa su archivo de estilos bajo un nombre. Esto crea un objeto JavaScript con el que nos referiremos a las distintas clases y selectores que hayamos creado en nuestros archivos de clase.
- La palabra clave `className` es propia de JavaScript, no de React.



Tips para map y keys

- La función map devuelve un nuevo array con los resultados de una función aplicada sobre cada elemento del array original.
- La función map no modifica el array original.
- La función map no reemplaza a los bucles for, simplemente es una alternativa más conveniente.
- React utiliza el atributo key para identificar cuáles elementos han cambiado, se han agregado o se han eliminado.
- Si no usamos key en una lista, podríamos tener un muy bajo rendimiento de la aplicación al hacer cambios en ella, o podríamos tener resultados inesperados (bugs) dependiendo del tipo de operaciones que realicemos sobre la lista.



- Los atributos key pueden ser una propiedad ID del elemento de la lista, o alguna composición hecha con algunas partes del contenido del elemento. Por ejemplo, si el elemento no tiene ID, pero tiene una descripción única, podríamos usar alguna función hash sobre el texto para evitar espacios y caracteres no alfanuméricos. Pero esto solo tiene sentido si la descripción es única.
- La key solo tiene que ser única entre hermanos, no es necesario que sea única globalmente.
- En casos en que no tengamos nada para usar como una key, podemos usar el parámetro index del elemento dentro del array. Esto no es óptimo, pero se utiliza cuando la lista no será modificada por el usuario.

Tips para estructura de componentes de clase

- La clase más básica que podemos escribir en React solamente necesita derivarse de `React.Component` e implementar `render` con un `return` que podría ser de tipo `null`.



- Las clases de React son las mismas que las clases de JavaScript.
- En JavaScript los constructores se escriben con la primera letra en mayúsculas únicamente por convención. En React, es necesario para la compatibilidad con XML de modo que no se confunda con una etiqueta HTML.
- Antes de usar `this` dentro del constructor debemos haber llamado a `super` que es el constructor de la clase base.
- Las funciones que están pensadas para que sean ejecutadas por otros componentes o elementos necesitan ser vinculadas con `bind` dentro del constructor.
- Las funciones que no están pensadas para que sean ejecutadas por otros componentes o elementos no necesitan ser vinculadas con `bind`.
- Debemos exportar nuestra clase para que sea accesible como un módulo por otros módulos de la aplicación. Hay dos formas de exportar, una con nombre y otra sin nombre. Dependiendo de la que usemos, debemos importar usando el nombre preciso o podemos usar cualquier nombre que queramos.



- En React, y por lo tanto en JavaScript, hay dos formas de herencia: la herencia clásica y la herencia basada en prototipo.
- La herencia basada en clases es simplemente un mecanismo para crear nuevas clases a partir de clases existentes. En JavaScript y React se utiliza la palabra clave `extends`.

Tips para state y setState

- Para manipular el estado en un componente de clase en React se utiliza la función `setState` provista por la clase base de React.
- Hay momentos clave en los que se debe manipular el estado. Uno es justo después de montar el componente, otro es justo después de que cambia alguna de sus propiedades.



- Cuando React crea un componente de clase sucede lo siguiente: invoca al constructor de la clase base; enseguida invoca al constructor de la clase del componente; el componente se hace accesible en el DOM (se monta); se ejecuta `componentDidMount`; cada vez que las propiedades del componente se actualizan se ejecuta `componentDidUpdate`; y justo antes de destruirse el componente se ejecuta `componentWillUnmount`.
- No se debe llamar a `setState` en `componentWillUnmount` porque el componente está próximo a ser destruido y ya no se volverá a renderizar.
- El operador `spread` (puntos suspensivos) se usa para expandir arrays u objetos JavaScript. Al usarse para pasar props, lo que hacemos es expandir ese objeto y —al mismo tiempo— meter esos objetos dentro de un objeto literal, es decir las dos llaves. De esta manera, cada prop es asignada a cada uno de los objetos expandidos.
- Trabajar con funciones en lugar de clases es lo recomendado no solo por la comunidad de React, sino también por la comunidad de JavaScript en general.



- El término ciudadano de primera clase se usa en programación para indicar que lo que sea denominado con ese término es tratado como un tipo de dato. En JavaScript —y en la programación funcional en general— las funciones son ciudadanas de primera clase. Esto en la práctica lo que significa es que una función se puede almacenar con un nombre (referencia), se puede devolver desde adentro de otra función, o incluso puede pasarse como parámetro a otras funciones.

¡Hasta la próxima!