



Certified Tech Developer

The Ultimate Degree

Front End III

createElement



¿Por qué es importante?

Quizá lo más importante de conocer *createElement* radica en que permite prescindir de configurar la compilación de JSX en nuestro ambiente de desarrollo. El otro factor importante es que, en realidad, el navegador no entiende JSX, sino que tras bastidores, en un proceso conocido como transpilación, el código JSX es transformado en llamadas a la función *React.createElement*, por lo que es importante entender qué es lo que en realidad sucede.



Datos importantes

Podemos crear una referencia a la función *React.createElement* y usar esa referencia como atajo para escribir menos:

```
const create = React.createElement;

const myP = create('p', null, 'Hello World');
const myDiv = create('div', null, myP);
```

En el código de arriba, en lugar de escribir *React.createElement* cada vez que lo necesitemos, creamos una constante que apunta a la función y luego usamos esa constante para crear nuestros elementos. El primer elemento es un tag `<p>`, sin *props* (pasamos *null*), y como hijo le pasamos el archiconocido string “Hello World”; el segundo elemento es un `<div>`, sin *props* (nuevamente pasamos *null*), y como hijo le pasamos el *paragraph* recién creado.

Recordemos que la función *getElementById* no es parte de la especificación ECMAScript, sino que está disponible dentro de *document* que es una propiedad del objeto global *window* que ofrece el DOM. El DOM (Document Object Model) es una API para trabajar con documentos HTML y XML.

Esto es todo por ahora. ¡Hasta la próxima!