

Volvamos de el método `for each` y vamos hacia muchos más porque los arreglos tienen muchos métodos, podemos utilizar muchos funciones por construcción para trabajar, para iterar sobre los datos de un arreglo, de veces hay muchos, vamos a ver los principales, además de `for each` con `map` y `filter` pero hay muchos otros, algunos de los cuales vamos a ver también aquí.

Antes de continuar con más métodos de los arreglos voy a volverme a chequear que efectivamente hay un entendible esta línea, esto que llamamos `template string` o `string interpolación básicamente` consiste en utilizar `back tick` los que son esas comillas invertidas, está del principio y está del final para poder mostrar texto con variables, esto es exactamente lo mismo que escribir esto que van a ver ustedes así, voy a comentar esta línea y ahora esto vamos a escribirlo como si fuera, digamos, la manera más tradicional o antigua, comillas, aquí estamos creando un elemento de tipo `"Y"`, luego `+`, el nombre de una variable, otra vez `"` y otra vez estas comillas un elemento `"y"`, si ahora guardamos esto como `var` a la izquierda que no cambia nada porque esta función es exactamente igual, aquí tenemos un texto que es interpolado por el método `string` del `document`, como un elemento de tipo `"Y"`, `ok`, y luego tenemos una referencia a una variable que es esta variable que estamos haciendo en cada pasada, en cada iteración del `for each`. Escribirlo de esta manera para nos puede resultar un tanto más sencillo o no, depende, ustedes pueden utilizar lo que quieran, aquí me va a lo tanto que están considerando ciertos espacios en blanco hay, con los `+` entre comillas, entonces, pero estamos haciendo exactamente lo mismo, quiero que sea quede claro, ahora sí, vamos a continuar con los métodos de los arreglos, vamos con `count`, hay muchos, vamos a ver bueno para esos `count` otros. Comencemos por el método `find` de un `array` que va a encontrar si un valor efectivamente existe en un arreglo o no, podríamos probarlo aquí `debug`, podríamos escribir por ejemplo `top songs find`, bueno el primero es el método incluido en realidad, aquí, esto que ya lo utilizamos para un ejercicio pero no está de más revisarlo, vamos a ver si de `top songs include` por ejemplo aquí un `string`, `inspeccionar` y lo escribimos exactamente igual, vamos a hacer de esto un `console log`, `oops`, vamos a escribirlo y esto nos va a devolver el elemento si es que lo encuentra, vamos a abrir aquí la consola de desarrollo para ver los resultados, si nos devuelve `true`, `ok`, vamos a limpiar aquí, podría vamos a limpiar ocupando nuestra página, ahora sí, comencemos nuevamente y nos dice `true`, porque efectivamente está encontrando este valor, aquí se está corriendo la línea 16, lo pedimos lo tenemos hecho arriba, bien mismo, lo escribimos dos veces, vamos a borrarlo de aquí arriba y esta también vamos a comentarla por ahora y ahora aquí ustedes pueden ver que nos devuelve `undefined` porque este valor no encuentra dentro de nuestro arreglo, de hecho aquí está, pero qué ocurre si lo escribimos de otra manera, lo escribimos con `undefined` nos va a retornar `false` puesto que no lo encuentra, lo mismo que si escribiéramos `undefined` con, un valor que efectivamente no se encuentra, `ok`, bien, también podríamos correr el método `find` y veremos que si el método `find` no encuentra un valor, vamos aquí a `index`, tenemos que pasar una función al método `find`, el método `find` va a recibir un, recordes un el concepto de los `callbacks`, una función que se le pasa como parámetro a otra función, `ok`, entonces aquí por ejemplo, `count` y podríamos aquí preguntar si `count` es igual a `undefined` no importa, aquí, y nos va a decir `only find`, `ok`, porque no encuentra precisamente este valor ya que la `"y"` está con `undefined`, entonces cuando no encuentra el valor el método `find` retorna `only find` a diferencia de `include` que nos retornaba `false`, vamos aquí ocurre si lo escribimos