

Trabajamos mucho hasta ahora con los arreglos y eso está muy bien todo el tiempo en la vida real si trabajas en el sector IT, en el sector de la tecnología, vas a trabajar con arreglos, pero hay otras estructuras que son tal vez más complejas, no porque sean más difíciles digo complejas en cuanto a cómo están construidas, por ejemplo, el arreglo funciona en una dimensión, podemos tener muchos datos de distintos tipos, uno detrás de otro, pero también existen los objetos literales que me permiten almacenar más datos en una estructura que viene dada por un par, una pareja, un par, que se llama “key value” o “Clave y Valor”, separados por comas, se escribe key o clave dos puntos el valor una coma y otro par debajo, entonces podemos tener un objetos muy grandes inclusive muy completos y además el objeto puede tener arreglos dentro y después podríamos tener arreglos que contengan objetos y entonces tendríamos estructuras de datos realmente muy ricas, eso es lo que ocurre en la vida real pero para llegar ahí primero veamos desde cero, que es, cómo se trabaja con objetos literales en javascript. Veremos aquí en el código la manera de declarar un objeto literal, simplemente indicaremos la creación de una variable, podríamos llamarle persona o podríamos llamarle, mejor aún, bueno si está bien, persona, vamos a llamarle persona, luego veremos aquí las propiedades de esa persona, tenemos aquí adentro propiedades y sus correspondientes valores, una propiedad podría ser nombre y luego de los dos puntos podríamos indicar look, tendríamos una siguiente propiedad con el nombre, apellido, podría ser... Tendríamos una edad, quizá, pongamos aquí puede ser de tipo numérico, tipo string, ok, no, eso es indistinto nosotros podemos indicar aquí cualquier tipo de valor que sea válidamente reconocido en javascript, pero además también podríamos tener estructuras de datos dentro de esta estructura de datos, por ejemplo, podríamos indicar aquí adentro habilidades y esto podría ser un arreglo de objeto, un arreglo perdón, si podría ser un arreglo de objetos pero no lo vamos a complicar tanto mientras aprendemos los objetos literales, podríamos poner aquí habilidades, esto y podríamos indicar y pongamos por ahora esas habilidades, con eso estaría bien y aquí tenemos un objeto literal, es una estructura de datos más flexible que un arreglo, que un array, porque nos permite trabajar con otras dimensiones, de hecho podemos tener propiedades anidadas aquí adentro como, por ejemplo, habilidades que no tiene un valor sino un arreglo que contiene valores, bien, este es un objeto, como accedemos a las propiedades de los objetos, de la siguiente manera, utilizamos la notación de puntos o dot notation aquí si nosotros corremos un console-log, vamos a abrir la consola aquí en el navegador para que puedan observar la salida de nuestro programa, veremos el objeto completo, que tiene su nombre, su apellido, su edad y sus habilidades y aquí me indica que las habilidades en realidad no son un valor en sí mismo, sino, que son un objeto, es decir, un objeto de tipo array que contiene valores, ahora cómo hacemos para acceder a cada una de estas propiedades de manera individual si es que así lo necesitáramos, bueno podríamos indicar, por ejemplo, aquí vamos a mostrarles el objeto de una manera tal vez un poco más ordenada con la el objeto console-table, el método table del objeto console, esta es una manera tal vez un poco más ordenada porque no lo muestra como una tabla de doble entrada, aquí tenemos las propiedades y aquí tenemos los valores, muy bien, ahora vamos a lo prometido, accedamos a la propiedad edad, ponemos el punto luego del nombre del objeto y simplemente escribimos edad y obtendremos 30, podemos indicar asimismo el resto de las propiedades y accederemos a todas, noten que si accedemos a habilidades la salida será diferente porque nos va a mostrar, aquí vamos a hacerlo con un console-log, en realidad ahora nos va a mostrar el arreglo que

contiene las habilidades, si deseábamos acceder a una de ellas puntualmente, bueno, recuerden que habilidades es un array o vector o arreglo dentro del objeto persona, por lo tanto para acceder a la propiedad en sí utilizamos el nombre del objeto, la notación de punto y luego el nombre de la propiedad, aquí he cambiado algo sin querer, ahí está, pero para acceder a una de las a uno de los valores de esta propiedad que es un arreglo seguiremos utilizando como aprendimos hasta ahora el índice del arreglo, si aquí ponemos por ejemplo 0 la respuesta será control de sable de luz, si ponemos 1 será pilotaje de naves exwing, bueno, vamos a mostrarlo para que no se queden solamente con mi palabra y que ocurre cuando intentamos acceder a un índice de un array que no ha sido definido, pues el resultado es tal y como habíamos visto antes cuando vimos los arreglos, los arrays o vectores, el resultado es andy find, ok, aquí tenemos entonces una manera de acceder a nuestras propiedades, ahora podríamos agregar aquí, por ejemplo, más datos, más propiedades, bueno si, podríamos por ejemplo indicando persona, la anotación de punto y vamos a dar nombre a una propiedad que aún no existe, allí podríamos poner origen y aquí podríamos decir que es igual a bueno si no recuerdo mal creo que el Look nació en Tatooine, perfecto, bueno no hemos visto ningún cambio por supuesto porque no estamos preguntando por el objeto persona, pero si volvemos a preguntar por él aquí veremos o con consulte-table también, persona, veremos que se ha agregado el lugar de nacimiento, ok, así es como podemos agregar nuevas propiedades a un objeto, también podemos cambiar un valor por supuesto, en el caso de escribir persona, punto, nombre igual Carlos Alberto y luego pedir nuevamente aquí el objeto persona veremos que su nombre ha cambiado ahora es Carlos Alberto Skywalker, podríamos hacer una adaptación al español o latinoamericano o algo por el estilo o hispanoamericano de la guerra de las galaxias, dejémoslo como Look, porque así le queda mejor, bien perfecto, por supuesto cómo agregamos también aquí una propiedad con un nuevo valor o sobre escribimos una propiedad con un valor nuevo también podemos agregar un nuevo vector o agregar un valor a este vector, a este arreglo, como podríamos hacerlo, bueno, recuerden que el método para agregar un elemento a un arreglo es el método push, podríamos entonces indicar persona, punto, habilidades, push y bueno qué otra habilidad podría haber aprendido Look Skywalker, tejer al crochet, que no todo es andar combatiendo al imperio en la vida no, y si ahora pidiéramos nuevamente esta línea pero sin indicar una de las uno de los elementos sino todos veríamos que las habilidades de Look son control de sable de luz, pilotaje de naves exwing y además tejer al crochet, ok, estas son las maneras en las que podemos acceder a un objeto, ahora imaginen que tenemos un objeto aquí similar a este, vamos a ponerle persona 2 y a este persona 1 y aquí que hagamos algunos cambios, bueno, no recuerdo exactamente cuáles pueden ser las habilidades de Leia que las tenía y muchas, pero inventemos algo aquí, pongamos, me ha cambiado el juego de caracteres ahora sí podré poner el acento, organización de grupos, bueno por ahora pongamos esto, bien, también podríamos poner diplomacia y aquí ahora el error que está apareciendo a la derecha es porque estamos preguntando por un objeto que ya no existe, ya no existe persona, existe persona 1 y persona 2, cuál es el objeto de haber creado un nuevo, valga el juego de palabras objeto, aquí tenemos 2 es mostrar como por ejemplo podemos expandir las capacidades de nuestras estructuras de datos ahora podríamos tener un arreglo const, que se llame hermanos o de alguna otra manera, no importa, y aquí dentro podríamos tener a persona 1 y persona 2 y aquí notarán ustedes que expandimos los arreglos porque es verdad que los arreglos pueden almacenar

datos de a uno a la vez, pero también pueden almacenar objetos, que a su vez son estructuras de datos complejos, entonces aquí si ahora hiciéramos un console-log de hermanos veríamos un arreglo que contiene dos objetos, ajá mucho más completo que antes cierto, tenemos los corchetes por fuera esto es un vector, un arreglo y dentro dos pares de llaves, esto indica que aquí dentro hay dos objetos, bien, como podríamos iterar por ejemplo este arreglo para finalizar con algunos de los muchos métodos de arreglos que vimos recientemente, aquí por ejemplo podríamos simplemente correr un, vamos a mostrarlo por pantalla con un console-log, un console-log de qué cosa, bueno, de un hermano, punto o hagamos los siguientes hermanos, vamos a escribirlo bien por favor, hermanos., por ejemplo for each no estaría mal, for each recuerden aquí recibimos un callback, este callback va a iterar nuestro arreglo hermanos, es decir, va a pasar una vez por cada elemento y aquí adentro podríamos llamarle de cualquier manera a cada uno de los elementos, item, element, elemento, personaje, personas, como sea, ok, y aquí podríamos hacer entonces, esto retornaría, esto significa esta flecha return, podríamos hacer aquí un console-log, console-log de persona, si solamente queremos el nombre podríamos poner personas.nombre y el resultado sería Look y Leia, ok, y con esto puedo hacer más cosas, por supuesto, puedo pedir el nombre y el apellido, persona.apellido y obtendremos y obtendremos este, bueno, precisamente el nombre y el apellido de cada objeto dentro del arreglo hermanos para cada una de las pasadas, para cada una de las iteraciones, ok, hasta aquí entonces, bueno, un repaso de cómo construir nuestros objetos literales, de la complejidad que éstos pueden contener y además como a su vez pueden ser contenidos dentro de un arreglo y así vamos viendo cómo podemos tener estructuras de datos cada vez más complejas y más completas.