

Un pilar en el manejo de las estructuras de datos son las estructuras o las instrucciones también son estructuras de repetición y control, estructuras de repetición y control como vimos el if, vimos el switch, hay sentencias que nos permiten recibir un objeto que tiene datos dentro, muchos datos inclusive y hacer algo una vez por cada uno de los datos, ese es el concepto detrás de una estructura de repetición y control, repetición porque vamos a repetir una operación N veces eso lo vamos a determinar nosotros sobre la base de un número fijo o un número variable como podría ser el tamaño, la longitud de un arreglo, la cantidad de datos que contiene, una vez que estamos iterando se llama iterar a cada una de las veces que hacemos algo por cada elemento, por ejemplo, en el array bueno, ahí vamos a hacer la estructura de control, nuestro algoritmo, haremos algo, devolveremos algo, una operación, mostraremos algo por pantalla, en fin, pueden ser muchas cosas, comenzaremos con la raíz de todo esto y es el famoso, digo famoso porque lo es, es el bucle for. La manera de declarar un bucle for es la siguiente, la palabra reservada for y luego entre paréntesis vamos a definir el contador, en este caso lo vamos a inicializar es una variable que se llama Y podría llamarse de cualquier otro modo, por convención le ponemos Y va a comenzar en cero, luego vendrá la condición, la condición es aquella que debe cumplirse para que las instrucciones dentro del ámbito del bucle for se sigan ejecutando, ahora lo veremos con más claridad, pongamos que mientras Y sea menor a 10 y a continuación indicaremos el acumulador, el crecimiento digamos, el incremento de nuestro contador, aquí entre llaves haremos algo mientras el bucle esté corriendo, ok,, vamos a explicar esto primero vamos a indicar qué es lo que debe hacer nuestro bucle, aquí vamos a escribir, por ejemplo, vamos a hacerlo entre back-tics que son estas comillas invertidas que me permiten mezclar texto con variables para no tener que usar coma o más, vamos aquí a concatenar esta es la iteración o la pasada, la corrida del bucle, número y aquí vamos a indicar el nombre de una variable, cual es la variable que sabe cuántas veces se va ejecutando el bucle, la variable Y, el contador, ok, entonces aquí vamos a indicar Y, qué va a ocurrir aquí bien, se va a ejecutar nuestro bucle for, se inicializa nuestro contador y esto ya no volverá a correr, ahora nuestro programa vendrá aquí y preguntará si Y es menor que 10, en efecto Y vale 0, por lo tanto, se ejecutará esta línea, dirá esta es la iteración número, bueno y aquí dirá 0, ok, porque Y comienza en cero, si queremos que diga 1 es tan sencillo como poner Y más 1, bien, luego corra esta última línea, se actualizará el valor de Y, decir que Y digamos, decir Y ++ es lo mismo que decir que Y es igual a Y +1, bien, es exactamente igual pero la verdad es que es mucho más corto y preferible escribir Y ++, Y ++ recuerden es lo mismo que decir Y es igual a Y+1, por lo tanto Y ahora vale 1, se ejecuta nuevamente la pregunta, la condición, Y es menor que 10, bueno, sí porque ahora vale 1 valía 0 y ahora vale 1, por lo tanto se ejecuta de esta línea, vuelve a sumarse 1 ahí que ahora valdrá 2, se volverá a preguntar si Y es menor que 10, en efecto porque ahora vale 2, en fin, y así es como se repite este bucle, por eso se llaman las estructuras de repetición y control suelen llamarse bucles, bien, porque, porque se repiten en bucle mientras una condición se cumple o hasta que una condición deja de cumplirse, si guardamos este archivo vamos a ver aquí el resultado, primera pasada, segunda, tercera cuarta, quinta, etcétera, muy bien, y esto podría correr de menor a mayor, bueno pues sí, aquí podría ser igual a 10 y aquí podríamos

invertir la pregunta, podríamos decir que mientras Y sea mayor a 10 o mayor a 0, mejor dicho no, mientras sea mayor a 0 y aquí podríamos decir y menos menos, es decir, vamos a arrancar en 10, vamos a preguntar si Y es mayor que 0 lo cual en principio va a ser verdad, ejecutaremos esta línea y luego disminuirémos el contador que irá bajando 10, 9, 8, 7, hasta que en un momento cuando sea cero esto no se va a cumplir, la condición, porque la condición dice que esta línea ha de correr está y todas las que estuvieran aquí verdad, mientras Y sea mayor a 0, cuando Y sea igual a 0 la condición ya no se va a cumplir, por lo tanto, bueno aquí como estamos poniendo $Y + 1$, tendremos que poner directamente Y por una cuestión de visualización y arranca en el número 10, 9, 8, etcétera, cuando llega a 1 se ejecuta por última vez, el contador vale 1, llega a esta línea y menos menos recuerden es lo mismo que Y es igual a $Y - 1$ y si valía 1 y le quitó 1 vale 0, entonces cuando aquí preguntan Y es mayor a 0, eso ya no es verdad, porque Y no es mayor a 0, es igual a cero, por lo tanto ahí se termina mi bucle, nunca entrará aquí y qué utilidad tiene esto, bueno puede tener muchas utilidades la verdad, podríamos tener aquí arriba, por ejemplo, si lo recuperó, un arreglo, un arreglo con datos, cualquier dato, nombres, números, etcétera, nombres de canciones, lo que sea, supongan que queremos recorrer este arreglo e imprimir por consola o mostrar en pantalla cada uno de los valores del arreglo, el arreglo se llama top songs, cómo podemos hacerlo, bueno, recuerden que para acceder al valor de un arreglo, a cualquiera de los valores, tenemos que recurrir al índice, aquí tendríamos que poner, por ejemplo, `document.wright NN` y aquí `top songs`, bueno y empezamos por la canción cero, el índice cero nos dirá `master of puppets`, ok, si decimos uno obtendremos `karma police`, etcétera, etcétera, bien no queremos esto, queremos que un bucle recorra todo el arreglo y nos pinte por pantalla o en la consola o donde sea toda la información del arreglo, para esto son especialmente útiles los bucles, entonces, `for` `led` y `Y` igual 0, vamos a preguntar, vamos a decir, bien, aquí viene un dato importante, cuanto tiempo queremos, cuantas veces queremos que corra este bucle, no sabemos cuál es la longitud del arreglo pero sabemos que si preguntamos, por ejemplo, `top songs.length`, vamos a obtener, esto no, esto lo vamos a comentar porque como no está terminado nos da un error, aquí vamos a obtener un 5 porque esa es la longitud del arreglo, recuerden la longitud de un arreglo está dada por la cantidad de elementos que contiene 1, 2, 3, 4, 5, por lo tanto, sabemos que deberíamos recorrer este arreglo 5 veces para obtener todos los valores, bien, entonces indiquemos aquí que la condición es que Y nuestro contador sea, bueno, si tiene 5 puedo decir menor a 6 o para ser más exacto menor o igual a 5, mientras el valor del contador sea menor o igual a 5, bien, aquí dudé un segundo saben por qué, porque el contador aquí el 0 en realidad va a hacer referencia al índice y si bien es cierto que tengo 5 elementos nuestro índice comienza en 0 o sea que es 0, 1, 2, 3, y 4 entonces, aquí tendríamos que preguntar efectivamente así, mientras Y sea menor a 5, entonces la primera vez será 0 la segunda 1, etcétera. La última vez será 4 y 4 es el quinto elemento 0, 1, 2, 3, 4, perfecto y entonces queremos que finalmente después de cada pasada, después de cada iteración, nuestro contador aumente en 1 y que queremos que haga aquí adentro, bueno, lo mismo que hacíamos hace instantes un `document.wright.line top songs`, y cómo accedemos a los arreglos, por su índice, pero aquí ya no pondremos 0 ni uno, ni dos, ni tres, ese dato está aquí, en el contador, la primera vez es 0 se imprimirá la primera canción, la segunda vez va a ser 1, se imprimirá la segunda y así sucesivamente. Si corremos este bucle ahí tenemos nuestro resultado, estamos recorriendo nuestro arreglo e imprimiendo

cada uno de los valores, para esto son particularmente útiles los bucles, más adelante veremos algunos métodos directamente, inclusive más sencillos de escribir, tiene que ver con métodos de los arreglos, pero primero queríamos enseñarles cómo funciona por dentro un bucle for.