



Quiz Investigativo

Ariel Gutiérrez Calvo

Universidad Cenfotec

Fundamentos de Programación Web

Francisco Jimenez Bonilla

2 Noviembre 2023

Arreglos	3
Arreglos Unidimensionales	4
Arreglos Multidimensionales	6
Conclusión	7
Referencias	8

Arreglos

En muchos casos, al intentar resolver un problema mediante programación, se necesita la capacidad de almacenar un conjunto de datos. En ocasiones los datos tienen características comunes, como ser del mismo tipo y representar lo mismo.

La forma de almacenar estos datos es en una estructura llamada **Arreglo**.

Los arreglos pueden ser unidimensionales o multidimensionales; de igual manera, se utilizan para hacer más eficiente el código de un programa.

Arreglos Unidimensionales

Un arreglo unidimensional es un tipo de dato que permite almacenar un conjunto de datos homogéneos, es decir, que son el mismo tipo de datos. El límite de los arreglos es finito.

Los arrays en Javascript funcionan de manera diferente que en otros lenguajes de programación, esto se debe a que los arreglos son completamente dinámicos ya que estos pueden cambiar su longitud de manera constante como por ejemplo: un arreglo de longitud 4 puede pasar a ser de longitud 12.

De igual manera cuando se trabaja con lenguajes de tipado débil como lo sería JS se puede trabajar en un mismo arreglo con múltiples tipos de datos (tipo numérico, cadena de texto, booleano, etc...).

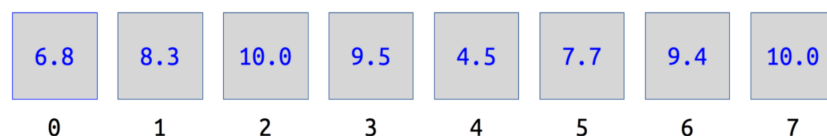
Se podría decir que los arrays en JS son multipropósito, pueden servir como listas o arreglos.

Algunas de las **características** que poseen los arreglos:

La numeración de las localidades del arreglo comienzan en 0, por lo general se pensaría que lo más natural es empezar desde el 1 sin embargo no funciona así debido a la manera como están numeradas las localidades de memoria en las computadoras, este tipo de numeración se aplica para la gran mayoría de lenguajes de programación.

Gracias a esto cada una de las localidades actúa como una variable independiente de las demás, por esa razón puede ser leída, modificada sin afectar a las demás. Esto se debe de tomar en cuenta por si se intentara hacer referencia a una localidad que no exista en el arreglo.

calificaciones[]



La gran **ventaja** que tienen los arreglos, es que ayudan a facilitar procesos, brinda soluciones prácticas a problemas que pueden requerir mucho trabajo.

Para ejemplificar de una mejor manera se podría exponer el caso de una lista de calificaciones de un grupo, este grupo cuenta con 8 alumnos y todos poseen una calificación diferente. La manera más práctica de resolver el problema es generando un array con todos los datos necesarios.

Para **declarar arreglos** en JavaScript se puede hacer de tres maneras;

A continuación se muestran las tres formas para declarar un **arreglo con Javascript**, utilizamos la palabra clave **var** seguido del nombre que llevará el **array** e inicializamos el array asignando los datos directamente, observamos que podemos utilizar en un mismo **array** datos de tipo cadena y numérico, todo esto será agregado dentro de corchetes(**[]**). La segunda forma para declarar **arrays** es haciendo uso de la palabra clave **new** seguido del objeto **Array** y asignamos los respectivos datos en este caso datos de tipo cadena(**String**) los dos primeros **arrays son de longitud 3**. La tercera forma para declarar un **array** en este caso cuando pasamos un solo argumento de tipo numérico al **objeto Array**, se asume que inicializamos un **array** vacío de 4 posiciones.

```
3 var arrayInicial = ["Juan", "Luis", 2];
4
5 var arraySecundario = new Array("Juan", "Luis", "Jose");
6
7 var arrayTerciario = new Array(4);
```

IMPRIMIR UN ARREGLO O ARRAY

Para poder acceder a las respectivas posiciones de un **array** lo hacemos a través de **índices** los cuales son identificadores numéricos, inicia desde la posición **0** hasta **N-1**(donde N-1 es igual a la longitud del array - 1), en este caso imprimimos todos los datos que tenemos en dicho **arreglo** el cual tiene tres posiciones de memoria reservados.

```
3 var arrayInicial = ["Juan", "Luis", 2];  
4  
5 document.write("Dato 1: "+arrayInicial[0] +  
6 |         |         |         | "  
7 |         |         |         | "  
8
```

Incrementar La longitud de un Array Unidimensional Indirectamente

Hay que recordar que los arreglos en Javascript son completamente dinámicos esto quiere decir que la longitud puede cambiar.

En la siguiente imagen podemos ver que nuestro **array** es de longitud **3** hemos asignado un nuevo dato en la posición **5**(`arrayInicial[5] = "Maria"`) la cual no existía pero al realizar este proceso estamos redimensionando nuestro **array** y ahora sería un **array de longitud 6**, en otros lenguajes de programación como por ejemplo **JAVA** esto generaría un error de desbordamiento pero en el caso de **Javascript** es totalmente válido.

```
3 var arrayInicial = ["Juan", "Luis", 2];
4
5 arrayInicial[5] = "Maria";
6
```

información e imágenes extraídas del sitio: <https://thebigwebdeveloper.blogspot.com>

Arreglos Multidimensionales

Los **arreglos multidimensionales** son una estructura de datos que almacena los valores en más de una dimensión. Los arreglos unidimensionales almacenan los valores en una dimensión por eso para poder acceder a las posiciones se utiliza solo un índice como se vió anteriormente. Los arreglos de **2** dimensiones guardan sus valores en filas y columnas por esto mismo se necesitan dos índices para acceder a cada una de sus posiciones. El arreglo multidimensional es como un contenedor que guardará más valores para cada posición.

¿Cuándo utilizarlos?

Los arreglos multidimensionales se utilizan cuando se deben almacenar múltiples valores del mismo tipo. A su vez, permite agrupar muchos datos usando un mismo identificador.

Para ejemplificar mejor se va a crear un arreglo de dos dimensiones donde por un lado se tienen ciudades y por el otro la temperatura media que hace en cada lugar durante invierno.

```
var temperaturas_medias_ciudad0 = new Array(3)
temperaturas_medias_ciudad0[0] = 12
temperaturas_medias_ciudad0[1] = 10
temperaturas_medias_ciudad0[2] = 11

var temperaturas_medias_ciudad1 = new Array (3)
temperaturas_medias_ciudad1[0] = 5
temperaturas_medias_ciudad1[1] = 0
temperaturas_medias_ciudad1[2] = 2

var temperaturas_medias_ciudad2 = new Array (3)
temperaturas_medias_ciudad2[0] = 10
temperaturas_medias_ciudad2[1] = 8
temperaturas_medias_ciudad2[2] = 10
```

En este ejemplo se crearon tres arreglos de 1 dimensión y tres elementos como los ya conocidos. Ahora se crea un nuevo arreglo de tres elementos y se introduce dentro de cada una de sus casillas los arreglos creados anteriormente, esto crea un arreglo de arreglos es decir, un arreglo de dos dimensiones

```
var temperaturas_ciudades = new Array (3)
temperaturas_ciudades[0] = temperaturas_medias_ciudad0
temperaturas_ciudades[1] = temperaturas_medias_ciudad1
temperaturas_ciudades[2] = temperaturas_medias_ciudad2
```

Se puede observar que para introducir el arreglo entero se hace referencia al mismo sin paréntesis ni corchetes, solo con su nombre. Siendo el arreglo de temperaturas_ciudades el arreglo bidimensional

Información e imágenes extraídas del sitio: <https://desarrolloweb.com/articulos/632.php>, visitar el sitio para mayor información y ejemplos.

En conclusión, se puede observar la importancia y utilidad de los arreglos como un método clave en el área de la programación sirviendo tanto para su desarrollo como mejora. Al ser un tema tan extenso no se logra abarcar el completo de su información sin embargo se mantienen muy claras las bases para entender el tema principalmente.

Esta información es sólo una pequeña muestra de lo que se puede realizar con arreglos en la programación, aún queda mucho más por investigar y poner en práctica.

- ❖ <https://thebigwebdeveloper.blogspot.com>
- ❖ <https://desarrolloweb.com/articulos/632.php>