



4.A. Estructuras definidas por el usuario en JavaScript

1.- Estructuras de datos.

En los lenguajes de programación existen estructuras de datos especiales, que nos sirven para guardar información más compleja que una variable simple. En la segunda unidad vimos como crear variables y cómo almacenar valores (simples), dentro de esas variables.

Hay un montón de tipos de estructuras de datos (listas, pilas, colas, árboles, conjuntos,...), que se pueden utilizar para almacenar datos, pero una estructura de las más utilizadas en todos los lenguajes es el array.

El **array**, es como una variable o zona de almacenamiento continuo, donde podemos introducir varios valores en lugar de solamente uno, como ocurre con las variables normales.

Los arrays también se suelen denominar **matrices** o **vectores**. Desde el punto de vista lógico, una matriz se puede ver como un conjunto de elementos ordenados en fila (o filas y columnas si tuviera dos o más dimensiones).

Se puede considerar que todos los arrays son de una dimensión: la dimensión principal, pero los elementos de dicha fila pueden a su vez contener otros arrays o matrices, lo que nos permitiría hablar de **arrays multidimensionales** (los más fáciles de imaginar son los de una, dos y tres dimensiones).

Los arrays son una estructura de datos, adecuada para situaciones en las que el acceso a los datos se realiza de forma aleatoria e impredecible. Por el contrario, si los elementos pueden estar ordenados y se va a utilizar acceso secuencial sería más adecuado usar una lista, ya que esta estructura puede cambiar de tamaño fácilmente durante la ejecución de un programa.

Los arrays nos permiten guardar un montón de elementos y acceder a ellos de manera independiente.

Cada elemento es referenciado por la posición que ocupa dentro del array. Dichas posiciones se llaman **índices** y siempre son correlativos.

Existen tres formas de indexar los elementos de un array:

- **indexación base-cero(0)**: en este modo, el primer elemento del array será la componente 0, es decir tendrá el índice 0.
- **indexación base-uno(1)**: en este modo, el primer elemento tiene el índice 1.
- **indexación base-n(n)**: este modo, es un modo versátil de indexación, en el que el índice del primer elemento puede ser elegido libremente.

En JavaScript cuando trabajamos con índices numéricos utilizaremos la *indexación base-cero(0)*.

El objeto **Array** descrito se denomina genéricamente array escalar debido a que sus elementos se referencian utilizando un número. Existe otro tipo de array en JavaScript (y en otros lenguajes de programación): el **array asociativo** (también llamado mapa, diccionario, hash, etc). En un array asociativo cada elemento es referenciado utilizando **claves** que son cadenas de caracteres. De esta forma un array asociativo se puede considerar como un conjunto de pares clave-valor y así se pueden referenciar valores utilizando cualquier tipo de clave ya que las claves

Una diferencia importante entre ambos tipos de arrays es que los escalares se pueden ordenar por el índice (ya que este es numérico) en cambio los asociativos no tienen orden.

Este tipo de array se corresponde con el objeto **Object** que, como veremos, se utilizar para crear objetos definidos por el usuario. Así, el nombre de la propiedad o método de un objeto definido por el usuario se corresponde con clave del elemento y el valor de esa propiedad o el código de ese método se corresponde con el valor del elemento. Veremos esto en el apartado sobre creación de objetos definidos por el usuario.

Para saber más

[Más información y tipos de estructuras de datos.](#)

Citas para pensar

"Lo que el estilo es a la persona, la estructura es a la obra."

Goytisolo, Luis.

