

ARRAYS EN JAVASCRIPT

En JavaScript, array es una única variable que se utiliza para almacenar diferentes elementos. A menudo se utiliza cuando queremos almacenar una lista de elementos y acceder a ellos mediante una sola variable.

A diferencia de la mayoría de los lenguajes de programación, donde el array es una referencia a la variable múltiple, en JavaScript es una única variable que almacena múltiples elementos.

1. DECLARACIÓN DE UN ARRAY

Existen básicamente dos formas de declarar un array.

Ejemplo:

```
var house = [];           // Method 1
var house = new Array();  // Method 2
```

Es preferible emplear siempre el primer método. Veamos el porqué con un ejemplo:

Inicialización de un array con el método 1

```
// Initialization while declaring
var house = ["1BHK", "2BHK", "3BHK", "4BHK"];

// Initialization after declaring
house[0] = "1BHK";
house[1] = "2BHK";
house[2] = "3BHK";
house[3] = "4BHK";
```

Inicialización de un array con el método 2:

```
// Initialization while declaring
// Creates an array having elements 10, 20, 30, 40, 50
var house = new Array(10, 20, 30, 40, 50);

// Creates an array of 5 undefined elements
var house1 = new Array(5);

// Creates an array with element 1BHK.
var home = new Array("1BHK");
```

Como se muestra en el ejemplo anterior, el array 'house' contiene 5 elementos, es decir (10, 20, 30, 40, 50), mientras que 'house1' contiene 5 elementos no definidos en lugar de tener un solo elemento 5. Por lo tanto, aunque se trabaja con números, este método generalmente no es el más adecuado, pero funciona bien con Strings y Boolean, como se muestra en el ejemplo anterior, el array 'home' sí contiene exclusivamente el elemento 1BHK.

Arrays en JavaScript pueden almacenar diferentes elementos. Se podrán albergar valores numéricos, Strings o Booleanos en un array unidimensional.

```
// Storing Numbers, Boolean, Strings in an Array;
var house = ["1BHK", 25000, "2BHK", 50000, "Rent", true];
```

Longitud de un Array (Length Property).

La propiedad Length nos devolverá la longitud del array. Este valor será siempre uno más al índice más alto de array (recordemos que el primer índice del array es el cero).

```
var house = ["1BHK", 25000, "2BHK", 50000, "Rent", true];

// len contains the length of the array.
var len = house.length;
for (var i = 0; i < len; i++)
    alert(house[i]);
```

2. OPERACIONES CON ARRAYS

Inicialización, Lectura y Escritura de todos los elementos de un Array

```
// Número de elementos totales del Array
const nMax = 4;

// Inicialización del Array reservando espacio para nMax elementos
var lista = new Array( nMax );

// Lectura de los elementos del Array
for ( var i = 0; i < nMax; i++ )
    lista[ i ] = prompt( "Introduce el elemento " + i + ": ", "" );

// Escritura/Muestra por pantalla de todos los elementos del Array
for ( var i = 0; i < nMax; i++ )
    document.write( "<br />Elemento Array[" + i + "] -> ", lista[ i ] );
```

Búsqueda en Arrays: Búsqueda Lineal

```
// Búsqueda Lineal de un elemento en el array

// Pedimos el valor a localizar
var elemBusq = prompt( "Introduce el elemento a buscar: ", "Escribe" );

//Iniciamos la búsqueda hasta encontrar el elemento o llegar a fin de array
i = 0;

while ( ( i < nMax - 1 ) && ( lista[ i ] != elemBusq ) )
    i++;

if ( lista[ i ] == elemBusq )
    alert ( "Elemento encontrado en la posición " + i + " del array" );
else
    alert ( "El elemento " + elemBusq + " no se encuentra en el array" );
```

3. FUNCIONES DE MANIPULACIÓN DE ARRAYS

FUNCIONES PARA AÑADIR ELEMENTOS O CONCATENAR ARRAYS

FUNCIÓN	UTILIDAD	EJEMPLOS aprenderaprogramar.com
concat(it1, it2, ... , itN)	Devuelve la concatenación de it1, it2, ..., itN con el array sobre el que se invoca. It1, it2, ..., itN pueden ser tipos primitivos u objetos.	<pre>var A3 = A1.concat(5, 9); var A4 = A1.concat(A2); var A5 = A1.concat([-4, 22, 11]);</pre>
push(x)	Añade x al final del array como nuevo (o nuevos) elemento, y devuelve la nueva longitud del array.	<pre>A1.push(55, 66); //Añade 55 y 66 al final</pre>
unshift(x)	Añade x al principio del array como nuevo (o nuevos) elementos.	<pre>A1.unshift([77, 88, 99]); //Ahora A1 tiene 3 elementos más, al principio</pre>
splice (ind, 0, it1, it2, ... , itN)	Modifica el array añadiendo los elementos it1, it2, ..., itN, que son insertados en la posición ind (desplazando a los existentes).	<pre>A1.splice(3, 0, 'xxx', 'yyy'); //Inserta xxx en posición 3, yyy en posición 4 y desplaza a los elementos existentes antes.</pre>
splice (ind, cuant, it1, it2, ... , itN)	Modifica el array eliminando cuant elementos e insertando it1, it2, ..., itN, desde el índice ind.	<pre>ejemplo.splice(3, 2, 'es', 'un', 'en'); //Se borran dos elementos y se insertan tres, con lo que la longitud del array es 1 más de la anterior.</pre>

FUNCIONES QUE PERMITEN ORDENAR O REORDENAR ARRAYS

FUNCIÓN	UTILIDAD	EJEMPLOS aprenderaprogramar.com
reverse()	Invierte el orden de los elementos en el array (el final pasa a ser el principio).	<pre>A1.reverse(); //Los elementos quedan ordenados al revés</pre>
sort()	Si no recibe parámetros, ordena los elementos del array por orden alfabético (que no coincide con el numérico), quedando el array modificado. Comentaremos esta función con más detenimiento.	<pre>var result = [2, 11, 111, 7].sort(); //result vale [11, 111, 2, 7] porque el orden es alfabético, no numérico.</pre>

FUNCIÓN	UTILIDAD	EJEMPLOS aprenderaprogramar.com
pop()	Elimina el último elemento del array y lo devuelve.	<pre>var ultimoElemento = A1.pop(); //Ahora A1 tiene 1 elemento menos</pre>
shift()	Elimina el primer elemento del array y lo devuelve.	<pre>var primerElemento = A1.shift(); //Ahora A1 tiene 1 elemento menos</pre>
splice (ind, cuant)	Modifica el array borrando cuant elementos a partir del índice ind.	<pre>A1.splice(3, 2);</pre>
splice (ind, cuant, it1, it2, ... , itN)	Modifica el array eliminando cuant elementos e insertando it1, it2, ..., itN, desde el índice ind.	<pre>ejemplo.splice(3, 2, 'es', 'un', 'en'); //Se borran dos elementos y se insertan tres, con lo que la longitud del array es 1 más de la anterior.</pre>
delete A[ind]	Elimina el elemento con índice ind del array A. El contenido a[ind] pasa a ser undefined.	<pre>delete A[7]; //Ahora A[7] contiene undefined</pre>

FUNCIONES PARA EXTRAER PARTES O ELEMENTOS DE UN ARRAY SIN ALTERARLO

FUNCIÓN	UTILIDAD	EJEMPLOS aprenderaprogramar.com
slice (firstIn, lastOut)	Devuelve un array con los elementos extraídos entre los índices firstIn y lastOut-1. Es decir, el elemento en la posición firstIn se incluye y el elemento en la posición lastOut se excluye.	<pre>var result = [1, 2, 3, 4, 5].slice(1,4); //result contiene [2, 3, 4]</pre>
slice (firstIn)	Devuelve un array con los elementos extraídos entre el índice firstIn y el último elemento. Si se indica un valor negativo, se extrae un array con los firstIn últimos elementos.	<pre>var result = [1, 2, 3, 4, 5].slice(-2) //result contiene [4, 5]</pre>
slice ()	Si se usa esta función sin argumentos devuelve un array con todos los elementos.	<pre>var result = [1, 2, 3, 4, 5].slice(); //result contiene [1, 2, 3, 4, 5]</pre>

FUNCIONES PARA RECUPERAR ÍNDICES DE POSICIONES

FUNCIÓN	UTILIDAD	EJEMPLOS aprenderaprogramar.com
indexOf(x)	Busca x dentro del array y devuelve la posición de la primera ocurrencia.	<code>var result = A1.indexOf(14);</code>
lastIndexOf()	Busca x dentro del array empezando por el final y devuelve la posición de primera ocurrencia.	<code>var result = A1.lastIndexOf(14);</code>