# Introducción a los arrays en JAVASCRIPT

El objeto global de JavaScript **Array** es un constructor para matrices unidimensionales (también llamadas vectores o arreglos), que son objetos tipo lista de alto nivel.

Lo primero será instanciar el array...

1. **var** miarray = **new** Array(4);

Una vez instanciado el array vemos que tenemos varias formas de inicializar un array en [Javascript](http://www.manualweb.net/tutorial-javascript/" \o "Manual de Javascript). La primera es indicando los valores del array en la propia instanciación.

1. **var** ciudades = **new** Array("Avila","Salamanca","Zamora","León");

La segunda es asignando los valores a cada una de las posiciones del array.

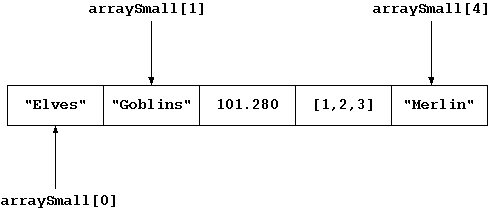
1. **var** ciudades = **new** Array(4);
2. ciudades[0] = "Avila";
3. ciudades[1] = "Salamanca";
4. ciudades[2] = "Zamora";
5. ciudades[3] = "León";

En nuestro caso hemos declarado un array de 4 posiciones y a la hora de asignar los valores tenemos que recordar que en [Javascript](http://www.manualweb.net/tutorial-javascript/" \o "Manual de Javascript) el primero de los elementos es el 0 y el último será el tamaño del array menos 1.

Si pensamos en la optimización de los arrays se puede evitar el uso de la sentencia new. Ya que el Array es un elemento primitivo del lenguaje. En este sentido, la forma más óptima sería la siguiente:

1. **var** ciudades = ["Avila","Salamanca","Zamora","León"];

Representación gráfica de un array de 5 elementos



**Crear un array**

var fruits = ["Manzana", "Plátano"];

console.log(fruits.length);

// 2

**Acceder (en índice) a un elemento de un Array**

var first = fruits[0];

// Manzana

var last = fruits[fruits.length - 1];

// Plátano

**Bucle sobre un Array**

fruits.forEach(function (item, index, array) {

console.log(item, index);

});

// Manzana 0

// Plátano 1

**Añadir al final de un Array**

var newLength = fruits.push("Naranja");

// ["Manzana", "Plátano", "Naranja"];

**Eliminar elemento del final de un Array**

var last = fruits.pop(); // elimina Naranja (del final)

// ["Manzana", "Plátano"];

**Eliminar elemento del inicio de un Array**

var first = fruits.shift(); // elimina Manzana del inicio

// ["Plátano"];

**Añadir elemento al inicio de un Array**

var newLength = fruits.unshift("Fresa"); // añade al inicio

// ["Fresa", "Plátano"];

**Encuentra el índice de un elemento en el Array**

fruits.push("Mango");

// ["Fresa", "Plátano", "Mango"];

var pos = fuits.indexOf("Plátano");

// 1

**Elimina un elemento por la posición del índice**

var removedItem = fruits.splice(pos, 1); // esta es la forma de eliminar un elemento

// ["Fresa", "Mango"];

**Copiar un Array**

var shallowCopy = fruits.slice(); // esta es la forma de crear una copia

// ["Fresa", "Mango"];

**Sintaxis**

Usando el constructor del objeto  Array:

new Array(arrayLength)

new Array(elemento0, elemento1, ..., elementoN)

Usando la sintaxis:

[elemento0, elemento1, ..., elementoN]

Parámetros

**arrayLength**

Es la longitud inicial de la matriz unidimensional. Puede acceder a este valor utilizando la propiedad length. Si el valor especificado no es un número, se crea una matriz unidimensional de longitud 1, con el primer elemento teniendo el valor especificado. La longitud máxima permitida para una matriz unidimensional es de 4.294.967.295.

**element*N***

Una lista de valores para los elementos de la matriz unidimensional. Cuando es especificada de esta manera, la matriz unidimensional se inicializa con los valores especificados como sus elementos y la propiedad length de la matriz unidimensional toma el valor del número de argumentos.

**Descripción**

Una matriz unidimensional (o vector) es un conjunto ordenado de valores asociados con un único nombre de variable. Nótese que usted [no debería utilizarlo como una matriz unidimensional asociativa](http://www.andrewdupont.net/2006/05/18/javascript-associative-arrays-considered-harmful/), utilice en su lugar un [object](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object" \o "El constructor Object crea un contenedor de objeto).

El siguiente ejemplo crea un objeto Array con una matriz unidimensional de textos, la matriz unidimensional bebidas contiene tres elementos y una longitud de tres:

bebidas = ["Café", "Té", "Chocolate"];

Puede contruir una matriz unidimensional de dos o más elementos comenzando con el índice 0 si define valores iniciales para todos los elementos. El siguiente código crea una matriz unidimensional con tres elementos:

miMatriz = new Array("Hola", miVariable, 3.14159);

Indexando una matriz unidimensional

Indexe una matriz unidimensional por sus números ordinales. Por ejemplo, se asume que define la siguiente matriz unidimensional:

miMatriz = new Array("Viento", "Lluvia", "Fuego");

Puede referirse a los elementos de esta forma:

* miMatriz[0] es el primer elemento
* miMatriz[1] es el segundo elemento
* miMatriz[2] es el tercer elemento

Especificando un sólo parámetro

Cuando especifica un único parámetro numérico con el constructor de un Array, especifica la longitud inicial de la matriz unidimensional. El siguiente código crea una matriz unidimensional de cinco elementos:

Métodoconstructor = new Array(5);

El comportamiento del constructor de un  Array depende de si el único parámetro es un número.

* Si el valor especificado es un número, el constructor convierte el número a un tipo entero de 32-bits sin signo  unsigned y genera una matriz unidimensional con la propiedad longitud (tamaño de la matriz unidimensional) convirtiéndolo en entero  integer. La matriz unidimensional inicialmente no contiene elementos, aún cuando se piense que se puede tener un valor de longitud distinto de cero.
* Si el valor especificado no es un número, una matriz unidimensional de longitud es creado, con el primer elemento teniendo el valor especificado.

El siguiente código crea una matriz unidimensional de longitud 25, luego asigna los valores para los tres primeros elementos:

Tipomúsica = new Array(25);

Tipomúsica[0] = "R&B";

Tipomúsica[1] = "Blues";

Tipomúsica[2] = "Jazz";

Incrementar la longitud de una matriz unidimensional indirectamente

La longitud de una matriz unidimensional se incrementa si se asigna un valor a un elemento mayor que la longitud actual de una matriz unidimensional. El siguiente código crea una matriz unidimensional de longitud cero 0, luego asigna un valor al elemento 99. Esto cambia la longitudo de una matriz unidimensional a 100.

colores = new Array();

colores[99] = "amarillopatito";

Creación de una matriz unidimensional utilizando el resultado de una coincidencia encontrada

El resultado de una coincidencia [String.match](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/String/match" \o "El método match() se usa para obtener todas las ocurrencias de una expresión regular dentro de una cadena.) entre una expresión regular y una cadena puede crear una matriz unidimensional. Esta matriz unidimensional posee propiedades y elementos que proveen de información acerca de la coincidencia. Una matriz unidimensional es el valor retornado por la ejecución de una expresión regular [String.exec](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/String/exec" \o "La documentación acerca de este tema no ha sido escrita todavía . ¡Por favor  considera contribuir !), cadena coincidente [String.match](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/String/match" \o "El método match() se usa para obtener todas las ocurrencias de una expresión regular dentro de una cadena.) y cadena remplazante [String.replace](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/String/replace" \o "El método replace() halla un emparejamiento entre una expresión regular y una cadena, y reemplaza la subcadena emparejada con una nueva subcadena.). Para explicar como ayudan estas propiedades y elementos, véase el siguiente ejemplo y refiérase a la table a continuación:

// Coincidencia de una d seguida de uno o más b seguida de una d

// Recordar las coincidencias de b y la siguiente d

// Ignorar el resto del caso

miexpresiónRegular = /d(b+)(d)/i;

miMatriz = miexpresiónRegular.exec("cdbBdbsbz");

Las propiedades y elementos retornados desde una coincidencia son las siguientes:

|  |  |  |
| --- | --- | --- |
| Propiedad/Elemento | Descripción | Ejemplo |
| input | Una propiedad sólo de lectura que refleja la cadena original con la expresión regular que fue coincidente. | cdbBdbsbz |
| index | Una propiedad sólo de lectura que está basada en cero su índice de las cadenas coincidentes. | 1 |
| [0] | Un elemento sólo de lectura que especifica los últimos caracteres coincidentes. | dbBd |
| [1], ...[n] | Elemento sólo de lectura que especifica las cadenas coincidentes parentizadas, si son incluidas en la expresión regular. El número de las posibles cadenas parentizadas es ilimitado. | [1]: bB [2]: d |

**Propiedades**

[**Array.constructor**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/constructor)

Especifica la función que crea un prototipo de Objeto.

[**Array.length**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/length)

Refleja el número de elementos en una matriz unidimensional.

[**Array.prototype**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/prototype)

Permite la adición de propiedades a todos los objetos.

**Métodos**

[**Array.from()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/from)

Crea un nuevo Array de un objeto iterable o parecido a un array.

[**Array.isArray()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/isArray)

Devuelve true si una variable es un Array, de lo contrario devuelve false.

[**Array.observe()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/observe)

Observa los cambios del Array asincronicamente, similar a [Object.observe()](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Object/observe" \o "El método Object.observe() es usado para observar de forma asíncrona cambios sobre un objeto.  Este método transmite información sobre cambios en el objeto, en el orden en que estos ocurren.) para los objetos. Provee un flujo de cambio en orden.

[**Array.of()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/of)

Crea un nuevo Array con un numero de argumentos variable, sin importar numero o tipo de argumentos.

Métodos transformadores [Mutator]

Estos métodos modifican la matriz unidimensional:

[**Array.pop()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/pop)

Elimina el último elemento de una matriz unidimensional y retorna este elemento.

[**Array.push()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/push)

Añade uno o más elementos al final de una matriz unidimensional y retorna la nueva longitud de una matriz unidimensional.

[**Array.reverse()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/reverse)

Invierte el orden de los elementos de una matriz unidimensional - el primero llega a ser el último y el último llega aser el primero.

[**Array.shift()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/shift)

Elimina el primer elemento de una matriz unidimensional y retorna este elemento.

[**Array.sort()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/sort)

Ordena los elementos de una matriz unidimensional.

[**Array.splice()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/splice)

Añade y/o elimina elementos de una matriz unidimensional.

[**Array.unshift()**](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array/unshift)

Añade uno o más elementos al comienzo de una matriz unidimensional y retorna la nueva longitud de la matriz unidimensional.

**Ejemplos**

Creación de una matriz unidimensional

El siguiente ejemplo crea una matriz,  msgArray, con una longitud de 0, luego asigna valores para  msgArray[0] y  msgArray[99], cambiando la longitud de la matriz unidimensional a 100.

var msgMatriz = new Array();

msgMatriz[0] = "Hola";

msgMatriz[99] = "mundo";

// La sentencia siguiene es verdadera,

// debido al elemento msgMatriz[99] definido.

if (msgMatriz.length == 100)

myVar = "La longitud es 100.";

Creación de una matriz de dos dimensiones

El siguiente ejemplo crea una matriz bidimensional y asigna el resultado a  myVar.

var myVar = "Prueba de una matriz multidimensional;";

var nb = 4;

var a = new Array(nb); // crea una matriz de longitud 4

for (var i = 0; i < nb; i++) {

a[i] = new Array(nb); // define cada elemento como una matriz de longitud 4

for (var j = 0; j < nb; j++) {

a[i][j] = "[" + i + "," + j + "]"; // asigna a cada elemento de la matriz bidimensional

// los valores de i y j

}

}

for (var i = 0; i < nb; i++) {

var str = "Fila " + i + ":";

for (var j = 0; j < nb; j++) {

str += a[i][j]; // añade a la cadena el contenido de la matriz bidimensional

}

myVar += str + "; "; // añade a la cadena original el resultado obtenido por cada iteración

}

Más información en: <https://developer.mozilla.org/es/>