



APRENDERAPROGRAMAR.COM

FUNCIONES ARRAYS
JAVASCRIPT. PUSH, SORT
(ORDENAR NÚMEROS),
CONCAT, JOIN, POP,
SHIFT, SLICE, SPLICE.
(CU01153E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº53 del Tutorial básico "JavaScript desde cero".

Autor: César Krall

FUNCIONES PREDEFINIDAS PARA ARRAYS (ARREGLOS)

JavaScript proporciona una serie de funciones predefinidas para el manejo de arrays. Entre ellas podemos citar concat, join, split, pop, push, shift, unshift, reverse. Vamos a estudiarlas. Recordar también que para conocer el número de elementos en un array basta con invocar la propiedad length del array escribiendo nombreArray.length



En las siguientes tablas se resumen funciones que pueden ser útiles para trabajar con arrays en JavaScript. Algunas de ellas ya fueron nombradas cuando hablamos de funciones para cadenas de texto, ya que JavaScript asemeja las cadenas de texto como arrays de caracteres.

Supondremos que A1, A2, A3, A4 y A5 son arrays JavaScript.

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 PARA EXTRAER ELEMENTOS O PARTES DE UN ARRAY CON ELIMINACIÓN

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>

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

FUNCIONES PARA TRANSFORMAR ARRAYS EN STRINGS O TIPOS PRIMITIVOS

FUNCIÓN	UTILIDAD	EJEMPLOS aprenderaprogramar.com
<code>join(separador)</code>	Une los elementos de un array en una cadena de texto donde cada elemento está separado por 'separador'.	<pre>var frase = ['quiero', 'aprender'].join(' '); //frase es tipo String y contiene 'quiero aprender'</pre>
<code>toString()</code>	Une los elementos de un array en una cadena de texto donde cada elemento está separado por una coma.	<pre>var frase = ['quiero', 'aprender'].toString(); //frase es tipo String y contiene 'quiero,aprender'</pre>
<code>valueOf()</code>	Este método devuelve la representación como tipo primitivo de un objeto. En el caso de un array, hace lo mismo que el método <code>toString()</code> .	<pre>alert (A1.valueOf()); //Mismo resultado que alert(A1);</pre>

FUNCIONES QUE PERMITEN ORDENAR O REORDENAR ARRAYS

FUNCIÓN	UTILIDAD	EJEMPLOS aprenderaprogramar.com
<code>reverse()</code>	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>
<code>sort()</code>	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>

LA FUNCIÓN SORT JAVASCRIPT. CÓMO ORDENAR NUMÉRICAMENTE

Hemos visto que `[2, 11, 111, 7].sort()` devuelve `[11, 111, 2, 7]` porque el array se ordena por orden alfabético y alfabéticamente cualquier número que contenga un 1 inicial se coloca antes que cualquier número que contenga un 2 inicial, independientemente de cuál de los dos números es mayor o menor.

Si el array contiene elementos `undefined`, estos se colocan al final del array.

Nos planteamos ahora, ¿cómo ordenar números de menor a mayor (o de mayor a menor)?

Para que la función `sort` ordene de forma distinta a la predeterminada se le debe pasar como parámetro una función de comparación. Por ejemplo:

[3, 5, 1, 7, 4].sort (funcionDeComparacion);

Donde funcionDeComparacion debe ser una función que reciba como parámetros dos elementos del tipo con el que trabaja el array y devuelva un número negativo si el primer elemento debe ordenarse antes que el segundo, cero si ambos elementos tienen igual orden, o un número positivo si el segundo elemento debe ordenarse antes que el primero.

Definir una función de comparación que permita ordenar un array numérico de menor a mayor es sencillo:

```
function funcionDeComparacion (elem1, elem2) {  
    return elem1 - elem2;  
}
```

Esta función devuelve un número positivo si elem1 es mayor que elem2, con lo cual elem1 se colocará detrás. Si escribimos return elem2 - elem1 se ordenará de mayor a menor.

Escribe el siguiente código y comprueba los resultados.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">  
<script type="text/javascript">  
function ejemplo() {  
var miArrayOriginal = [1, 4, 2, 9, 11, 44, 111, 7, 4];  
var miArray = [1, 4, 2, 9, 11, 44, 111, 7, 4];  
var msg = 'El array original es: ' + miArrayOriginal + '\n';  
msg = msg + 'Ordenado de mayor a menor es ' + miArray.sort(deMayorAMenor) + '\n';  
msg = msg + 'Ordenado de menor a mayor es ' + miArray.sort(deMenorAMayor) + '\n';  
alert(msg);  
}  
function deMenorAMayor(elem1, elem2) {return elem1-elem2;}  
function deMayorAMenor(elem1, elem2) {return elem2-elem1;}  
</script>  
</head>  
<body><div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>  
<div style="color:blue;" id="pulsador" onclick="ejemplo()"> Probar </div>  
</body>  
</html>
```

El resultado esperado es:

El array original es: 1,4,2,9,11,44,111,7,4

Ordenado de mayor a menor es 111,44,11,9,7,4,4,2,1

Ordenado de menor a mayor es 1,2,4,4,7,9,11,44,111

EJERCICIO 1

Crea un script donde a partir del array [33, 2, 36, 55, 4, 1] se realicen los siguientes procesos:

- a) Mostrar el array ordenado de menor a mayor y de mayor a menor usando la función sort y definiendo la función de ordenación de forma anónima, es decir, deberás invocar `X.sort(function (...){ ... })` definiendo la función de forma anónima en vez de cómo función con nombre.
- b) Crea una variable de nombre `deMenorAMayor` que contenga el resultado de ordenar de menor a mayor los elementos del array.
- c) Crea una variable de nombre `deMayorAMenor` que contenga el resultado de ordenar de mayor a menor los elementos del array.
- d) Muestra por pantalla el array original, la variable `deMenorAMayor` y la variable `deMayorAMenor`.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

EJERCICIO 2

Crea un script donde sea posible ordenar palabras por orden alfabético sin tener en cuenta la existencia de mayúsculas o minúsculas. Por ejemplo, declarar `var miArray=['Moto', 'soto', 'Abaco', 'abeja', 'Sapo', 'nieve', 'Zumba', 'barco']` y tras ordenar obtener `['Abaco', 'abeja', 'barco', 'Moto', 'nieve', 'Sapo', 'soto', 'Zumba']`.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01154E

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206