

Secciones

- Manuales
- Programas
- FAQs
- Directorio
- Vídeos
- En directo
- Monotemáticos
- Desde 0
- Diseño web
- HTML
- CSS
- Javascript
- ASP
- PHP

Longitud de los arrays

Por [Miguel Angel Alvarez](#) [Seguir a @midesweb](#)

📅 22 de diciembre de 2001 📁 [Manuales de Javascript](#)

Aprendemos más cosas sobre el funcionamiento de los arrays y en concreto vemos como utilizar su propiedad length para acceder al número de casillas que tiene.

En el artículo anterior del [Manual de Javascript](#) empezamos a explicar el

funcionamiento de los arrays, en este artículo veremos como utilizar su propiedad length para acceder al número de casillas que tiene.

Este sitio web usa cookies para garantizar que obtengas la mejor experiencia en nuestro sitio web.

Más información

Permitir cookies

Configuración de cookies

utilizan una propiedad del objeto array, la propiedad length. Ya veremos en objetos qué es una propiedad, pero para nuestro caso podemos imaginarnos que es como una variable, adicional a las posiciones, que almacena un número igual al número de casillas que tiene el array.

Para acceder a una propiedad de un objeto se ha de utilizar el operador punto. Se escribe el nombre del array que queremos acceder al número de posiciones que tiene, sin corchetes ni paréntesis, seguido de un punto y la palabra length.

```
var miArray = new Array()

miArray[0] = 155
miArray[1] = 499
miArray[2] = 65

document.write("Longitud del array: " + miArray.length)
```

Este código imprimiría en pantalla el número de posiciones del array, que en este caso es 3. Recordamos que un array con 3 posiciones abarca desde la posición 0 a la 2.

Es muy habitual que se utilice la propiedad length para poder recorrer un array por todas sus posiciones. Para ilustrarlo vamos a ver un ejemplo de recorrido por este array para mostrar sus valores.

```
for (i=0;i<miArray.length;i++){
    document.write(miArray[i])
}
```

Hay que fijarse que el bucle for se ejecuta siempre que i valga menos que la longitud del array, extraída de su propiedad length.

El siguiente ejemplo nos servirá para conocer mejor los recorridos por los arrays, el funcionamiento de la propiedad length y la creación dinámica de nuevas posiciones. Vamos a crear un array con 2 posiciones y rellenar su valor. Posteriormente introduciremos un valor en la posición 5 del array. Finalmente imprimiremos todas las posiciones del array para ver lo que pasa.

```
var miArray = new Array(2)

miArray[0] = "Colombia"
miArray[1] = "Estados Unidos"

miArray[5] = "Brasil"

for (i=0;i<miArray.length;i++){
    document.write("Posición " + i + " del array: " + miArray[i])
    document.write("<br>")
}
```

El ejemplo es sencillo. Se puede apreciar que hacemos un recorrido por el array desde 0 hasta el número de posiciones del array (indicado por la propiedad length). En el recorrido vamos imprimiendo el número de la posición seguido del contenido del array en esa posición. Pero podemos tener una duda al preguntarnos cuál será el número de elementos de este array, ya que lo habíamos declarado con 2 y luego le hemos introducido un tercero en la posición 5. Al ver la salida del programa podremos contestar nuestras preguntas. Será algo parecido a esto:

- Posición 0 del array: Colombia
- Posición 1 del array: Estados Unidos
- Posición 2 del array: null
- Posición 3 del array: null
- Posición 4 del array: null
- Posición 5 del array: Brasil

Se puede ver claramente que el número de posiciones es 6, de la 0 a la 5. Lo que ha ocurrido es que al introducir un dato en la posición 5, todas las casillas que no estaban creadas hasta la quinta se crean también.

Las posiciones de la 2 a la 4 están sin inicializar. En este caso nuestro navegador ha escrito la palabra *null* para expresar esto, pero otros navegadores podrán utilizar la palabra undefined. Ya veremos más adelante qué es este null y dónde lo podemos utilizar, lo importante ahora es que comprendas cómo trabajan los arrays y los utilices correctamente.

Podemos [ver el efecto de este script en tu navegador en una página a parte](#).

Continuaremos el tema de arrays en la siguiente entrega de este manual: [Arrays multidimensionales](#).

Autor

Miguel Angel Alvarez

Seguir a @midesweb

Miguel es fundador de DesarrolloWeb.com y la plataforma de formación online EscuelaIT. Comenzó en el mundo del desarrollo web en el año 1997, transformando su hobby en su trabajo.



Subir ↑

Manual

[Manual de JavaScript](#)

- ← Arrays en Javascript
- Propiedades con get y set en Javascript →

Compartir

Compartir

3

Recomendar

Tweet

Comentarios

📝 [Enviar un comentario al artículo](#)

cho Juan Perenal

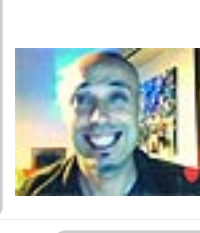
Longitud Arrays

22/10/2009

Chacho, los arrays comienzan en index 1. no cero

[Marcar como spam](#)

midesweb



Confirmado! Arrays comienzan con el índice 0

23/8/2010

El último comentario está equivocado. Te aseguro que los arrays comienzan con el índice (index, o como lo quieras llamar) cero. En algunos lenguajes empiezan con el índice 1, pero en Javascript y la mayoría de ellos, empiezan siempre en 0.

[Marcar como spam](#)

kronos

arrays dimensionales

06/9/2010

hola, queria saber como se crea un array de dos dimensiones

para saber el valor medio entre ambos, lo que hice fue esto:

```
public class dosarray{

public static void main(String[]args)
{

double array[][] = {{1, 2, 3}, {3, 2, 1}};

int suma=0, total=0;

for(int outerindex= 0; outerindex <= array.length; outerindex++){
{
for(int innerindex=0; innerindex < array[outerindex].length; innerindex++){

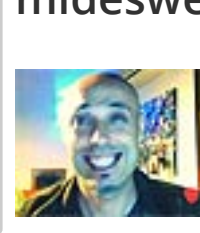
suma+= array[outerindex][innerindex];
total++;
}
}

System.out.println("valor medio del array=" + (suma/total));
}
}

lo hice basandome en un ejemplo de un libro pero no me dio, que esta mal?
```

[Marcar como spam](#)

midesweb



arrays multidimensionales

28/9/2010

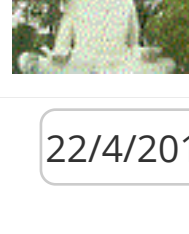
Accede al próximo capítulo de este manual, donde tratamos los

arrays multidimensionales:

<http://www.desarrolloweb.com/articulos/632.php>

[Marcar como spam](#)

Erick



dimencion Arrays Multidimensionales

22/2/2011

kronos:

En tu codigo lo que esta mal es el ultimo innerindex debe decir outerindex y al darle Valor a la Array te faltan valores me imagino que lo sacaste del libro de La Biblia de Java 2 que se bajo de Internet a mi tambien me causo dolor de cabeza porque no esta vien laimprecision jajajaja este es el codigo correcto

```
public class BLOQUE FOR {

/**
 * @param args
 */
public static void main(String[] args) {
double array[][]={{1,2,3},{3,2,1},{1,2,3}};
int sum=0,total=0;
for(int outerindex = 0; outerindex < array.length; outerindex++)
{
for(int innerindex=0;innerindex < array[outerindex].length;innerindex++;
sum +=array[outerindex][outerindex];
total++;
}
System.out.println("Valor Promedio del array="+sum/total));
}

}
```

[Marcar como spam](#)

Luis Monroy Ramirez

Pero cual es el limite de campos por array?

23/1/2015

Quisiera cual es el limite de campos en el array ya que tengo una matriz que me almacena mas de 2mil variables en un array y al recibirlos solo me muestra 1000.

Agradesco su colaboracion.

[Marcar como spam](#)

Rafael

en js los arreglos se modifican dinamicamente?

12/2/2016

Es decir que puedo declarar un arreglo de 2 posiciones y en tiempo de ejecución ponerle 10 valores y no me dará problema de desbordamiento o algo así?

[Marcar como spam](#)

📝 [Enviar un comentario al artículo](#)

Principales

- Manuales
- FAQs
- En directo
- Vídeos

Monotemáticos

- Desde cero
- HTML, CSS
- Javascript, Ajax
- Diseño, ASP

