



APRENDERAPROGRAMAR.COM

ARRAYS JAVASCRIPT
(ARREGLOS).
DECLARACIÓN,
INICIALIZACIÓN. ARRAY
VACÍO O CON ELEMENTOS
UNDEFINED. EJEMPLOS.
(CU01121E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

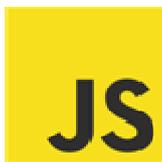
Fecha revisión: 2029

Resumen: Entrega nº21 del Tutorial básico “JavaScript desde cero”.

Autor: César Krall

ARRAYS JAVASCRIPT (ARREGLOS).

Al igual que en la mayor parte de los lenguajes de programación, en JavaScript podemos trabajar con arrays (también llamados arreglos, vectores o matrices). Los arrays son de gran importancia ya que permiten organizar series de datos que comparten el mismo nombre pero se diferencian por un índice.



DECLARACIÓN E INICIALIZACIÓN DE ARRAYS

La declaración de un array se hace de misma forma que se declara cualquier variable: `var nombreDelArray;`

El array adquiere condición de tal cuando la variable se inicializa con forma de array, bien como array con un contenido inicial o bien como array vacío:

```
var pais;  
pais = ['Mexico', 'España', 'Argentina', 'Chile', 'Colombia', 'Venezuela', 'Perú', 'Costa Rica'];
```

O podemos hacerlo todo en una sola línea:

```
var pais = ['Mexico', 'España', 'Argentina', 'Chile', 'Colombia', 'Venezuela', 'Perú', 'Costa Rica'];
```

En este ejemplo decimos que hemos declarado un array de 8 elementos. Cada elemento tiene un índice, comenzando los índices por el número 0. Por tanto los 8 elementos del array anterior son: `pais[0]`, `pais[1]`, `pais[2]`, `pais[3]`, `pais[4]`, `pais[5]`, `pais[6]` y `pais[7]`.

También podemos inicializar un array vacío de dos formas distintas:

```
var fruta = [];  
var fruta = new Array();
```

Ambas expresiones tienen el mismo efecto: crean un array vacío. En este caso se entiende que se añadirán contenidos a posteriori. Por ejemplo `fruta[3] = 'manzana';`

ARRAYS CON ELEMENTOS SIN DEFINIR

Podemos dar valor a un elemento de un array sin que los anteriores elementos estén definidos como hemos hecho en el ejemplo, declarando el elemento de índice 3 sin haber definido los índices 0, 1 y 2. Los elementos no definidos toman valor "undefined". En este ejemplo `fruta[0]` no ha sido definido por lo que si intentamos invocarlo su valor es `undefined`.

Para que JavaScript comprenda que una variable es un array hay que indicárselo explícitamente. Es válido: `var pais = []; pais[0] = 'Mexico';`

Sin embargo no es válido: `var pais; pais[0] = 'Mexico';` ¿Por qué? Porque en este código no hemos declarado explícitamente que `pais` sea un array. Si no lo hemos declarado, no podemos empezar a usar índices asociados a la variable como si se tratara de un array. Si lo hacemos se entiende como un error en el código, lo que dará lugar a que JavaScript no se ejecute.

Un array puede inicializarse dejando ciertos índices sin un contenido definido. Para ello, se deja un espacio separado por comas como en este ejemplo:

```
var ciudad = ["Buenos Aires", , "Madrid"];
```

Esta declaración supone que el array tiene 3 elementos. `Ciudad[0]` que tiene valor Buenos Aires, `ciudad[1]` que tiene valor `undefined`, y `ciudad [2]` que tiene valor Madrid.

```
var capital = [ , 'Mexico D.F.' , 'Santiago'];
```

Esta declaración supone que el array `capital` tiene 4 elementos que son `ciudad[0]` con valor `undefined`, `ciudad[1]` con valor México D.F., `ciudad[2]` con valor `undefined` y `ciudad[3]` con valor Santiago.

Una coma final no genera un nuevo elemento. Por ejemplo:

```
var capital = [ , 'Mexico D.F.' , 'Santiago', ,];
```

En este caso la coma final es ignorada y el array sigue teniendo 4 elementos. Si queremos definir el array con un quinto elemento vacío se recomienda hacerlo así:

```
var capital = [ , 'Mexico D.F.' , 'Santiago', undefined];
```

También sería posible hacerlo dejando una coma final libre pero esto es menos recomendable y en algunos navegadores puede dar lugar a errores:

```
var capital = [ , 'Mexico D.F.' , , 'Santiago', ,];
```

ACCESO A ÍNDICES NO EXISTENTES

En otros lenguajes de programación, intentar acceder a un índice de array inexistente devuelve un error, pero en JavaScript no es así. Si escribimos `pais[40]` cuando sólo hemos definido hasta el índice 7, el resultado es que `pais[40]` devuelve `undefined`. A diferencia de en otros lenguajes, los arrays en JavaScript no tienen un número fijo de elementos, sino que el número de elementos del array se ajusta dinámicamente según las necesidades.

USO DE LOS ARRAYS

Los arrays son de gran utilidad para automatizar los cálculos y procesos, por lo que siempre algo pueda expresarse con un nombre seguido de un índice, será preferible usar un array a usar variables independientes.

Los arrays por defecto siempre empiezan por el índice 0, pero en determinadas ocasiones algunos programadores prescinden de ese índice. Por ejemplo `var mes = []; mes[0] = undefined; mes[1] = 'enero'; mes[2]='febrero'; mes[3]='marzo'; mes[4] = 'abril'; mes[5]='mayo'; mes[6] = 'junio';`

```
mes[7]='julio'; mes[8]='agosto'; mes[9]='septiembre'; mes[10]='octubre'; mes[11]='noviembre';  
mes[12]='diciembre';
```

Hemos definido `mes[0]` como `undefined`. ¿Por qué? Porque en general es preferible dejar constancia de que si `mes[0]` tiene valor `undefined` es porque el programador ha decidido que sea así, de esta manera no hay duda respecto a que pueda ser un error o un olvido.

También sería posible definir doce variables como `mes1`, `mes2`, `mes3`, `mes4`, `mes5`, `mes6`, `mes7`, `mes8`, `mes9`, `mes10`, `mes11`, `mes12`. Sin embargo esto es algo que desde el punto de vista de la programación es en general indeseable, ya que estas doce variables funcionan como variables que no tienen relación entre sí. Por tanto cuando tengamos que recorrer los meses no podremos hacerlo de forma automatizada usando índices, ya que aquí no existen índices (aunque el nombre de la variable lleve un número eso no significa que ese número sea un índice).

TIPADO DE LOS ARRAYS

En otros lenguajes de programación un array contiene un tipo de datos y se dice que el array es de ese tipo. Por ejemplo un array puede ser un array de enteros, o un array de cadenas de texto. Pero no pueden existir arrays que contengan indistintamente elementos de distinto tipo. Esto sí es posible en JavaScript, y por ello se dice que los arrays en JavaScript no tienen tipo. Por ejemplo se admite algo como esto: `var datos = ['Frío', 33, false, 'nube', -11.22, true, 3.33, 'variado'];`

En este array el elemento de índice cero es de tipo texto, el de índice 1 es un valor numérico, el elemento de índice tres es un valor booleano, etc.

Normalmente los arrays contendrán elementos de un tipo, por ejemplo valores numéricos, pero en ocasiones nos interesará que contengan elementos de distintos tipos.

JavaScript admite que los elementos de un array sean elementos de naturaleza compleja (objetos), o incluso que un elemento de un array sea otro array.

PROPIEDAD LENGTH DE LOS ARRAYS

La propiedad `length` de un array indica el número máximo de elementos en el array de acuerdo con el índice máximo existente (independientemente de que los elementos del array tengan contenido o no). Por ejemplo si definimos `var oficina = []; oficina[25] = 'Oficial José Vargas Corononado';` la propiedad `oficina.length` devuelve 26, número de elementos para el array (de 1 a 25 más el correspondiente al 0).

MÁS SOBRE LOS ARRAYS

Los arrays son elementos de gran importancia dentro de JavaScript y aún queda mucho que estudiar sobre ellos. De momento el conocimiento adquirido nos sirve para seguir avanzando, pero más adelante volveremos a explicar más cosas relacionadas con los arrays.

EJEMPLO

Escribe el siguiente código y guárdalo en un archivo de extensión html (puedes cambiar la ruta de la imagen si lo deseas):

```
<html> <head> <title>Curso JavaScript aprenderaprogramar.com</title> <meta charset="utf-8">
<script type="text/javascript">
function mostrarMensaje1() {
var mes; mes = []; mes[0] = undefined; mes[1] = 'enero'; mes[2]='febrero'; mes[3]='marzo'; mes[4] = 'abril'; mes[5]='mayo';
mes[6] = 'junio'; mes[7]='julio'; mes[8]='agosto'; mes[9]='septiembre'; mes[10]='octubre'; mes[11]='noviembre';
mes[12]='diciembre';
var pais = ['Mexico', 'España', 'Argentina', 'Chile', 'Colombia', 'Venezuela', 'Perú', 'Costa Rica'];
var msg, msg2; msg = 'El país de índice 2 es: ' + pais[2] + '\n\n';
var datos = ['Frío', 33, false, 'nube', -11.22, true, 3.33, 'variado'];
msg = msg + 'En el índice 1 de datos tenemos: ' + datos[1] + ' (numérico), p.ej. multiplica por 2: ' + (datos[1]*2) + '\n\n';
msg = msg + 'En el índice 2 de datos tenemos: ' + datos[2] + ' y es booleano\n\n';
msg = msg + 'En el índice 3 de datos tenemos: ' + datos[3] + ' y es un texto \n\n';
msg = msg + 'En el índice 40 de datos tenemos: ' + datos[40] + '\n\n';
var fruta = [];
msg = msg + 'En el índice 0 de fruta tenemos: ' + fruta[0] + ' y en el índice 30 ' + fruta[30] + '\n\n';
fruta[1] = 'pera'; fruta[2] = undefined; fruta[30] = 'manzana';
msg = msg + 'En el índice 0 de fruta tenemos: ' + fruta[0] + ' y en el índice 30 ' + fruta[30] + '\n\n';
alert (msg);
msg2 = 'Mostramos el array país: ' + pais + '\n\n';
msg2 = msg2 + 'Mostramos el array fruta: ' + fruta + '\n\n';
msg2 = msg2 + 'Mostramos el array datos: ' + datos + '\n\n';
msg2 = msg2 + 'Mostramos el array mes: ' + mes + '\n\n';
msg2 = msg2 + 'Intentamos sumar o concatenar arrays: ' + (pais + fruta) + '\n\n';
msg2 = msg2 + 'Valor length en el array pais es: ' + (pais.length) + ' y en el array fruta es ' + fruta.length + '\n\n';
alert (msg2);
var ejemplo = new Array(); alert('Contenido de ejemplo: ' + ejemplo); ejemplo [0]= 1; ejemplo [2]= 44;
alert('Contenido de ejemplo: ' + ejemplo);
}
</script>
</head>
<body> <div> <p>Aquí un párrafo de texto situado antes de la imagen, dentro de un div contenedor</p>

<p onclick ="alert('Alerta JavaScript')" style="color: #D2691E;">Aquí otro párrafo de texto. JavaScript es un lenguaje
utilizado para dotar de efectos dinámicos a las páginas web. </p> </div> </body> </html>
```

Visualiza el resultado y comprueba que la página web se muestra con normalidad y que JavaScript se ejecuta con normalidad cuando haces click sobre la imagen.

El resultado esperado es que se muestre lo siguiente (compruébalo):

El país de índice 2 es: Argentina

En el índice 1 de datos tenemos: 33 (numérico), p.ej. multiplica por 2: 66

En el índice 2 de datos tenemos: false y es booleano

En el índice 3 de datos tenemos: nube y es un texto

En el índice 40 de datos tenemos: undefined

En el índice 0 de fruta tenemos: undefined y en el índice 30 undefined

En el índice 0 de fruta tenemos: undefined y en el índice 30 manzana

Mostramos el array país: Mexico,España,Argentina,Chile,Colombia,Venezuela,Perú,Costa Rica
Mostramos el array fruta: ,pera,,,,,,,,,,,,,,,,,,,,,,,,,,,,manzana
Mostramos el array datos: Frío,33,false,nube,-11.22,true,3.33,variado
Mostramos el array mes:
,enero,febrero,marzo,abril,mayo,junio,julio,agosto,septiembre,octubre,noviembre,diciembre
Intentamos sumar o concatenar arrays: Mexico,España,Argentina,Chile,Colombia,Venezuela,Perú,Costa Rica,pera,,,,,,,,,,,,,,,,,,,,,,,,,,,,manzana
Valor length en el array pais es: 8 y en el array fruta es 31
Contenido de ejemplo:
Contenido de ejemplo: 1,,44

Fíjate en las siguientes cuestiones: un array en JavaScript puede contener elementos de distintos tipos. El contenido de un elemento no definido es undefined. Un array puede tener elementos intermedios no definidos. Cuando tratamos de mostrar por pantalla un array, se produce una conversión automática a texto. Esto no significa que el array se un texto ni un tipo String, sino simplemente que el intérprete hace una conversión automática para tratar de ofrecer un resultado. En el caso de elementos no definidos, al mostrarse el array se muestran espacios separados por comas.

EJERCICIO

Crea un script donde declares un array vacío denominado nombres. Pide al usuario tres nombres usando la sentencia prompt de JavaScript y almacena esos nombres como elementos 0, 1 y 2 del array. A continuación muestra el contenido del array por pantalla.

Ejecuta el código y comprueba sus resultados. Para comprobar si es correcta tu solución puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01122E

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