



APRENDERAPROGRAMAR.COM

## FOR IN JAVASCRIPT: RECORRER ARRAYS O PROPIEDADES DE OBJETOS. DIFERENCIA ENTRE FOR NORMAL Y FOR IN. EJEMPLOS (CU01135E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

**Resumen:** Entrega nº35 del Tutorial básico “JavaScript desde cero”.

Autor: César Krall

## RECORRER COLECCIONES CON FOR IN

El recorrido de arrays es posible usando un bucle for normal o un bucle while, donde existe un control explícito de los índices que se van recorriendo (0, 1, 2, 3...). JavaScript permite recorrer arrays y colecciones de objetos usando un tipo especial de bucle: el bucle for – in (que a su vez tiene ciertas similitudes con el for each).



La sintaxis a emplear para recorrer las propiedades de un objeto es la siguiente:

```
for (nombreIndice in nombreObjeto) {  
  ... ejecución de sentencias ...  
}
```

En el caso concreto de los arrays podemos escribir típicamente lo siguiente:

```
for (indice in nombreDelArray) {  
  ...  
  realizar operaciones sobre nombreDelArray[indice]...  
  ...  
}
```

En este código vemos cómo podemos usar recorrer un array con un for normal o, de forma equivalente, con un for in:

```
function ejemploForIn() {  
  var dato = [2, 6, 5, 1, 18, 44];  
  var msgForNormal = "";  
  var msgForIn = "";  
  //For normal  
  for (var i=0; i<dato.length; i++) { msgForNormal = msgForNormal + dato[i] + ' - '; }  
  //For in  
  for (i in dato) { msgForIn = msgForIn + dato[i] + ' - '; }  
  alert ('msgForNormal contiene ' + msgForNormal + ' y msgForIn contiene ' + msgForIn);  
}
```

Incorpora el código dentro de una página web y comprueba cómo el resultado que obtienes es el siguiente: msgForNormal contiene 2 - 6 - 5 - 1 - 18 - 44 - y msgForIn contiene 2 - 6 - 5 - 1 - 18 - 44 -

Es decir, hemos recorrido el array con un for normal y con un for in. ¿Es mejor usar un for tradicional o un for in? Ambas formas resultan relativamente equivalentes y útiles, aunque en algunos casos específicos puede interesar más usar una forma u otra, y en algunos casos específicos no se obtienen los mismos resultados usando ambas formas. Nosotros nos inclinamos preferentemente por el uso del for tradicional, pero en determinadas circunstancias y para algunos usos especiales más avanzados (de los que hablaremos cuando estudiemos los objetos en JavaScript y el recorrido de las propiedades de un objeto) puede ser interesante usar for - in.

Una cuestión a tener en cuenta cuando se hacen usos más avanzados de for – in es que el orden en que se obtienen los elementos de la colección no tiene por qué coincidir con el orden de los índices. En el ejemplo anterior teníamos un array `var dato = [2, 6, 5, 1, 18, 44]`; y el orden esperado de recorrido es 2 - 6 - 5 - 1 - 18 - 44. Pero el for in no necesariamente respeta este orden, podría hacer un recorrido en un orden aparentemente aleatorio como 18 - 6 - 44 - 1 - 5. Con los arrays probablemente no ocurra esto, pero con otro tipo de colecciones sí puede ocurrir. Por tanto al usar un for – in podemos estar seguros de que se van a recorrer todos los elementos en la colección, pero no podemos asegurar que ese recorrido vaya a seguir un orden determinado. Si el orden es importante, será mejor usar un for tradicional u otro tipo de control.

Otra consideración a tener en cuenta es el rendimiento o velocidad de ejecución esperable cuando utilizamos un for – in respecto al que obtenemos cuando usamos un for tradicional. En general, si se trata de un array pequeño las diferencias pueden resultar inapreciables. Sin embargo, con arrays de muchos datos la ejecución es mucho más rápida utilizando un for tradicional.

## EJERCICIO

1) Genera un script que pida cinco números al usuario usando un bucle for normal (usa prompt para pedir los datos y conviértelos a valor numérico posteriormente). Almacena los números introducidos por el usuario en un array. A continuación usando un for in que recorra el array, muestra un mensaje informando del resultado de multiplicar cada uno de los números por 3. Ejemplo:

Se pedirán al usuario cinco números, supongamos que introduce 1, 3, 9, 10 y 7

A continuación se mostrará el mensaje:

Multiplicamos por 3 los números introducidos:  $1*3 = 3$ ,  $3*3 = 9$ ,  $9*3 = 27$ ,  $10*3 = 30$  y  $7*3 = 21$ .

Para comprobar si tus respuestas son correctas puedes consultar en los foros [aprenderaprogramar.com](http://aprenderaprogramar.com).

**Próxima entrega:** CU01136E

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