Esta página fue traducida del inglés por la comunidad, pero no se mantiene activamente, por lo que puede estar desactualizada. Si desea ayudar a mantenerlo, descubra cómo activar las configuraciones regionales inactivas.

for...in

La instrucción `for-in` itera sobre todas las `propiedades enumerables` de un objeto que está codificado por cadenas (ignorando los codificados por `Símbolos`, incluidas las propiedades enumerables heredadas.

JavaScript Demo: Statement - For...In

```
const object = { a: 1, b: 2, c: 3 };

for (const property in object) {
  console.log(`${property}: ${object[property]}`);
}

// expected output:
// "a: 1"
// "b: 2"
// "c: 3"

```

Run ›

Reset

Sintaxis

```
for (variable in objeto)
  instrucción
```

`variable`

Asigna un nombre de propiedad diferente a la *variable* en cada iteración.

`objeto`

Objeto cuyas propiedades enumerables que no son símbolos se iteran.

Descripción

Un bucle `for...in` solo itera sobre propiedades enumerables que no son símbolo. Los objetos creados a partir de constructores integrados como `Array` y `Object` han heredado propiedades no enumerables de `Object.prototype` y `String.prototype`, como el método `indexOf()` de `String` o el método `toString()` de `Object`. El bucle iterará sobre todas las propiedades enumerables del objeto en sí y aquellas que el objeto hereda de su cadena de prototipos (las propiedades de los prototipos más cercanos tienen prioridad sobre las de los prototipos más alejados del objeto en su cadena de prototipos).

Propiedades `deleted`, `added` o `modified`

Un bucle `for...in` itera sobre las propiedades de un objeto en un orden arbitrario (consulta el operador `delete` para obtener más información sobre por qué no puede depender del aparente orden de la iteración, al menos en una configuración entre navegadores).

Si una propiedad se modifica en una iteración y luego se visita en un momento posterior, su valor en el bucle es su valor en ese momento posterior. Una propiedad que se elimina antes de haber sido visitada no se visitará más tarde. Las propiedades agregadas al objeto sobre el que se está produciendo la iteración se pueden visitar u omitir de la iteración.

En general, es mejor no agregar, modificar o eliminar propiedades del objeto durante la iteración, aparte de la propiedad que se está visitando actualmente. No hay garantía de si se visitará una propiedad agregada, si se visitará una propiedad modificada (distinta de la actual) antes o después de que se modifique, o si se visitará una propiedad eliminada antes de eliminarla.

Iteración en arreglos y `for…in`

> **Nota**: `for...in` no se debe usar para iterar sobre un `Array` donde el orden del índice es importante.

Los índices del arreglo son solo propiedades enumerables con nombres enteros y, por lo demás, son idénticos a las propiedades generales del objeto. No hay garantía de que `for...in` devuelva los índices en un orden en particular. La instrucción de bucle `for...in` devolverá todas las propiedades enumerables, incluidas aquellas con nombres no enteros y aquellas que se heredan.

Debido a que el orden de iteración depende de la implementación, es posible que la iteración sobre un arreglo no visite los elementos en un orden coherente. Por lo tanto, es mejor usar un bucle `for` con un índice numérico (o `Array.prototype.forEach()` o el bucle `for...of`) cuando se itera sobre arreglos donde el orden de acceso es importante.

Iterar solo sobre propiedades directas

Si solo deseas considerar las propiedades adjuntas al objeto en sí mismo, y no sus prototipos, usa `getOwnPropertyNames()` o realiza una `hasOwnProperty()` verificación (`propertyIsEnumerable()` también se puede utilizar). Alternativamente, si sabes que no habrá ninguna interferencia de código externo, puedes extender los prototipos incorporados con un método de verificación.

¿Por qué usar `for...in`?

Dado que `for...in` está construido para iterar propiedades de objeto, no se recomienda su uso con arreglos y opciones como `Array.prototype.forEach()` y existe `for...of`, ¿cuál podría ser el uso de `for...in`?

Es posible que se utilice de forma más práctica con fines de depuración, ya que es una forma fácil de comprobar las propiedades de un objeto (mediante la salida a la consola o de otro modo). Aunque los arreglos suelen ser más prácticos para almacenar datos, en situaciones en las que se prefiere un par clave-valor para trabajar con datos (con propiedades que actúan como la "clave"), puede haber casos en los que desees comprobar si alguna de esas claves cumple un valor particular.

Ejemplos

Utilizar `for...in`

El siguiente bucle `for...in` itera sobre todas las propiedades enumerables que no son símbolos del objeto y registra una cadena de los nombres de propiedad y sus valores.

```
var obj = {a: 1, b: 2, c: 3};

for (const prop in obj) {
  console.log(`obj.${prop} = ${obj[prop]}`);
}

// Produce:
// "obj.a = 1"
// "obj.b = 2"
// "obj.c = 3"
```

Iterar propiedades directas

La siguiente función ilustra el uso de `hasOwnProperty()` — las propiedades heredadas no se muestran.

```
var triangle = {a: 1, b: 2, c: 3};

function ColoredTriangle() {
  this.color = 'red';
}

ColoredTriangle.prototype = triangle;
```

```
var obj = new ColoredTriangle();

for (const prop in obj) {
  if (obj.hasOwnProperty(prop)) {
    console.log(`obj.${prop} = ${obj[prop]}`);
  }
}

// Produce:
// "obj.color = red"
```

Especificaciones

Especificación
ECMAScript (ECMA-262) La definición de 'declaración for...in' en esta especificación.

Compatibilidad del navegador

Report problems with this compatibility data on GitHub

for...in	
Chrome	1
Edge	12

Firefox	1
Internet Explorer	6
Opera	2
Safari	1
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Node.js	0.10.0

 Full support

Compatibilidad: expresiones iniciadoras en modo estricto

Antes de Firefox 40, era posible utilizar una expresión iniciadora (`i=0`) en un bucle `for...in` :

```
var obj = {a: 1, b: 2, c: 3};
for (var i = 0 in obj) {
  console.log(obj[i]);
}
// 1
// 2
// 3
```

Este comportamiento no estándar ahora se ignora en la versión 40 y posteriores, y presentará un `SyntaxError` ("`iniciador for...in no válido` (en-US) en `modo estricto` (error 748550 y error 1164741 ").

Otros motores como v8 (Chrome), Chakra (IE/Edge) y JSC (WebKit/Safari) están investigando si eliminar también el comportamiento no estándar.

Ve también

- `for...of` — una declaración similar que itera sobre la propiedad `values`
- `for each...in` — una declaración similar pero obsoleta que itera sobre los valores de las propiedades de un objeto, en lugar de los nombres de las propiedades en sí
- `for`
- `Expresiones generadoras` (usa la sintaxis `for...in`)
- `Enumerabilidad y posesión de propiedades`
- `Object.getOwnPropertyNames()`

- `Object.prototype.hasOwnProperty()`
- `Array.prototype.forEach()`

Last modified: 8 ago 2021, by MDN contributors