Curso de Javascript

Unidad Didáctica 10: Arrays

http://cursosdedesarrollo.com/

Ayuntamiento de Vitoria-Gasteiz
Vitoria-Gasteizko Udala

Índice de contenidos

- Introducción
- Representación de un array
- Propiedad length
- Borrado
- Conclusiones

http://cursosdedesarrollo.com/

Introducción

Un array es una asignación lineal de memoria donde los elementos son accedidos a través de índices numéricos, siendo además una estructura de datos muy rápida

http://cursosdedesarrollo.com/

Introducción

Desafortunadamente, JavaScript no utiliza este tipo de arrays

En su lugar, JavaScript ofrece un objeto que dispone de características que le hacen parecer un array

http://cursosdedesarrollo.com/

Introducción

Internamente, convierte los índices del array en strings que son utilizados como nombres de propiedades, haciéndolo sensiblemente más lento que un array

Representación de un Array

JavaScript ofrecen una manera muy cómoda para crear y representar arrays

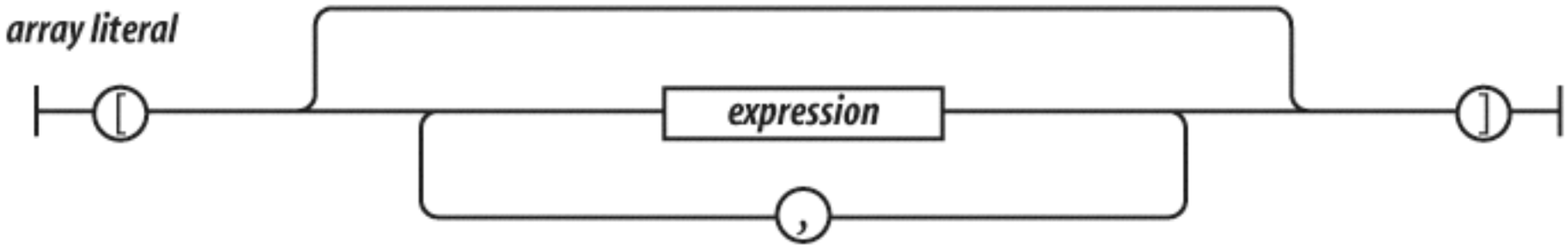
Una representación de un array consiste en una pareja de corchetes ([ ]) que contienen cero o más expresiones

http://cursosdedesarrollo.com/

Representación de un Array

El primer valor recibe la propiedad de nombre '0', la segunda la propiedad de nombre '1', y así sucesivamente

Representación de un Array



http://cursosdedesarrollo.com/

Representación de un Array

Algunos ejemplos de declaración de arrays:

```
var empty = [];

var numbers = [

'zero', 'one', 'two', 'three', 'four',

'five', 'six', 'seven', 'eight', 'nine'

];

empty[1]        // undefined

numbers[1]      // 'one'

empty.length    // 0

numbers.length  // 10
```

http://cursosdedesarrollo.com/

Representación de un Array

Si representasámos el array como un objeto

var numbers_object = {

'0': 'zero', '1': 'one', '2': 'two',

'3': 'three', '4': 'four', '5': 'five',

'6': 'six', '7': 'seven', '8': 'eight',

'9': 'nine'

];

http://cursosdedesarrollo.com/

Representación de un Array

Ambas representaciones contienen 10 propiedades, y ests propiedades tienen exactamente los mismos nombres y valores

http://cursosdedesarrollo.com/

Representación de un Array

La diferencia radica en que numbers hereda de Array.prototype, mientras que numbers_object lo hace de Object.prototype, por lo que numbers hereda una serie de métodos que convierten a numbers en un array

http://cursosdedesarrollo.com/

Representación de un Array

En la mayoría de los lenguajes de programación, se requiere que todos los elementos de un array sean del mismo tipo, pero en JavaScript eso no ocurre. Un array puede contener una mezcla valores:

```
var misc = [

'string', 98.6, true, false, null, undefined,

['nested', 'array'], {object: true}, NaN,

Infinity

];

misc.length // 10
```

http://cursosdedesarrollo.com/

Propiedad Length

Todo array tiene una propiedad length

A diferencia de otros lenguajes, la longitud del array no es fija, y podemos añadir elementos de manera dinámica

http://cursosdedesarrollo.com/

Propiedad Length

Esto hace que la propiedad length varíe, y tenga en cuenta los nuevos elementos. La propiedad length hace referencia al mayor índice presente en el array, más uno. Esto es:

```
var myArray = [];
myArray.length              // 0
myArray[1000000] = true;
myArray.length              // 1000001
// myArray contiene un elemento!
```

http://cursosdedesarrollo.com/

Propiedad Length

La propiedad length puede indicarse de manera explícita

Aumentando su valor, no vamos a reservar más espacio para el array, pero si disminuimos su valor, haciendo que sea menor que el número de elementos del array, eliminará los elementos cuyo índice sea mayor que el nuevo length:

numbers.length = 3; // numbers es ['zero', 'one', 'two']

Borrado

Como los arrays de JavaScript son realmente objetos, podemos utilizar el operador delete para eliminar elementos de un array:

```
delete numbers[2];

// numbers es ['zero', 'one', undefined, 'shi', ‘go']
```

Desafortunadamente, esto deja un espacio en el array. Esto es porque los elementos a la derecha del elemento eliminado conservan sus nombres

Borrado

Para este caso, JavaScript incorpora una función splice, que permite eliminar y reemplazar elementos de un array. El primer argumento indica el por qué elemento comenzar a reemplazar, y el segundo argumento el número de elementos a eliminar.

```
numbers.splice(2, 1);

// numbers es ['zero', 'one', 'shi', 'go']
```

Conclusiones

Hemos visto cómo manejar
los datos en un array

http://cursosdedesarrollo.com/

Datos de Contacto

http://www.cursosdedesarrollo.com
info@cursosdedesarrollo.com

http://cursosdedesarrollo.com/

Licencia

David Vaquero Santiago