

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

MF0491_3



Realizado por Manuel Macías

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Arrays

Los objetos te permiten almacenar colecciones de datos a través de nombres.

Pero a menudo necesitamos una colección ordenada, donde tenemos un 1ro, un 2do, un 3er elemento y así sucesivamente. Por ejemplo, necesitamos almacenar una lista de algo: usuarios, bienes, elementos HTML, etc.

No es conveniente usar objetos aquí, porque no proveen métodos para manejar el orden de los elementos. No podemos insertar una nueva propiedad “entre” los existentes. Los objetos no están hechos para eso.

Existe una estructura llamada Array (llamada en español arreglo o matriz/vector) para almacenar colecciones ordenadas.

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Declaración

Hay dos sintaxis para crear un array vacío:

```
let arr = new Array();
```

```
let arr = [];
```

Casi siempre se usa la segunda. Podemos suministrar elementos iniciales entre los corchetes:

```
let fruits = ["Apple", "Orange", "Plum"];
```

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Los elementos del array están numerados comenzando desde cero.

Podemos obtener un elemento por su número entre corchetes:

```
let fruits = ["Apple", "Orange", "Plum"];
```

```
alert( fruits[0] ); // Apple  
alert( fruits[1] ); // Orange  
alert( fruits[2] ); // Plum
```

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Podemos reemplazar un elemento:

```
fruits[2] = 'Pear'; // ahora ["Apple", "Orange", "Pear"]
```

...o agregar uno nuevo al array:

```
fruits[3] = 'Lemon'; // ahora ["Apple", "Orange", "Pear", "Lemon"]
```

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

La cuenta total de elementos en el array es su longitud `length`:

```
let fruits = ["Apple", "Orange", "Plum"];
```

```
alert( fruits.length ); // 3
```

También podemos usar `alert` para mostrar el array completo.

```
let fruits = ["Apple", "Orange", "Plum"];
```

```
alert( fruits ); // Apple,Orange,Plum
```

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Un array puede almacenar elementos de cualquier tipo.

Por ejemplo:

// mezcla de valores

let arr = ['Apple', { name: 'John' }, true, function() { alert('hello'); }];

// obtener el objeto del índice 1 y mostrar su nombre

alert(arr[1].name); // John

// obtener la función del índice 3 y ejecutarla

arr[3](); // hello

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Coma residual

Un array, al igual que un objeto, puede tener una coma final:

```
let fruits = [  
  "Apple",  
  "Orange",  
  "Plum",  
];
```

La “coma final” hace más simple insertar y remover items, porque todas la líneas se vuelven similares.

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Obtener los últimos elementos con “at”

Digamos que queremos el último elemento de un array.

Podemos calcular explícitamente el último índice y luego acceder al elemento:

```
let fruits = ["Apple", "Orange", "Plum"];
```

```
alert( fruits[fruits.length-1] ); // Plum
```

----- ***** -----

```
let fruits = ["Apple", "Orange", "Plum"];
```

```
// es lo mismo que fruits[fruits.length-1]
```

```
alert( fruits.at(-1) ); // Plum
```

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Métodos pop/push, shift/unshift

Una **cola** es uno de los usos más comunes de un array. En ciencias de la computación, significa una colección ordenada de elementos que soportan dos operaciones:

unshift: inserta un elemento al principio.

shift: obtiene el elemento del principio, avanzando la cola, y así el segundo elemento se vuelve primero.

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Métodos pop/push.

Hay otro caso de uso para los arrays, la estructura de datos llamada **pila**. Esta soporta dos operaciones:

push: agrega un elemento al final.

pop: toma un elemento desde el final.

Entonces los elementos nuevos son agregados o tomados siempre desde el “final”.

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Métodos pop/push

Para las pilas, la última introducida es la primera en ser recibida, en inglés esto es llamado principio LIFO (Last-In-First-Out, última en entrar primera en salir).

Para las colas, tenemos FIFO (First-In-First-Out primera en entrar, primera en salir).

Los arrays en JavaScript pueden trabajar como colas o pilas. Ellos permiten agregar o quitar elementos al o del principio o al o del final.

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Los métodos `push/pop` son rápidos, mientras que `shift/unshift` son lentos.



La operación **shift** debe hacer 3 cosas:

1. Remover el elemento con índice 0.
2. Mover todos los elementos hacia la izquierda y reenumerarlos: desde el índice 1 a 0, de 2 a 1 y así sucesivamente.
3. Actualizar la longitud: la propiedad `length`.

push/pop: no necesitan mover nada. Para extraer un elemento del final, el método `pop` limpia el índice y acorta `length`.

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Métodos shift/unshift

Unshift

Agrega el elemento al principio del array:

```
let fruits = ["Orange", "Pear"];
```

```
fruits.unshift('Apple');
```

```
alert( fruits ); // Apple, Orange, Pear
```

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

Métodos shift/unshift

Unshift

Agrega el elemento al principio del array:

```
let fruits = ["Orange", "Pear"];
```

```
fruits.unshift('Apple');
```

```
alert( fruits ); // Apple, Orange, Pear
```

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

El método `split()`

El método `split()` divide (fragmenta) un string en dos o más sub cadenas usando un separador (divisor). El separador puede ser un solo carácter, otra cadena, o una expresión regular.

Una vez que se ha dividido la cadena en múltiples sub cadenas, el método `split()` sitúa las subdivisiones resultantes en una cadena que posteriormente es retornada como respuesta. Lo anterior es realizado sin modificar la cadena original.

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

El método `split()`

```
let mensaje = 'Soy un tipo feliz y afortunado';  
// Dividiendo la cadena "mensaje" usando el carácter espacio  
let arr = mensaje.split(' ');
```

```
// El array  
console.log(arr); // ["Soy", "un", "tipo", "feliz", "y", "afortunado"]
```

```
// Acceso a cada elemento del arreglo resultante  
console.log(arr[0]); // "Soy"  
console.log(arr[1]); // "un"  
console.log(arr[2]); // "tipo"  
console.log(arr[3]); // "feliz"  
console.log(arr[4]); // "y",  
console.log(arr[5]); // "afortunado"
```

PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

ARRAYS

El método join()

El método join () se utiliza para convertir todos los elementos de un array en una cadena.

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
var energy = fruits.join(" y ");
```

El resultado de la salida de energy:

Plátano y naranja y manzana y mango



Programación web en el entorno cliente

GRACIAS

MANUEL MACÍAS

TUTORIAS@MANUELMACIAS.ES



PROGRAMACIÓN JAVASCRIPT