Métodos de Array

El objeto `Array` tiene los siguientes métodos:

- **`concat`** une dos arreglos y retorna un nuevo arreglo.

```
   miArreglo = new Array("1","2","3")
   miArreglo = miArreglo.concat("a", "b", "c"); // miArreglo ahora tiene ["1", "2",
"3", "a", "b", "c"]
```

- **`join`**`(deliminator = ",")` une todos los elementos de un arreglo en una cadena separada por un carácter delimitador opcional

```
miArreglo = new Array("Viento","Lluvia","Fuego")
lista = miArreglo.join(" - "); // la lista tiene "Viento - Lluvia - Fuego"
```

- **`pop`** remueve el último elemento de un arreglo y retorna este elemento.

```
miArreglo = new Array("1", "2", "3");
último=miArreglo.pop(); // miArreglo ahora tiene ["1", "2"], último = "3"
```

- **`push`** añade uno o más elementos al final de un arreglo y retorna la nueva longitud del array

```
miArreglo = new Array("1", "2");
miArreglo.push("3"); // miArreglo ahora tiene ["1", "2", "3"]
```

- **`reverse`** invierte el orden de los elementos de un arreglo: el primer elemento del arreglo se convierte en el último y el último se convierte en el primero.

```
miArreglo = new Array ("1", "2", "3");
miArreglo.reverse(); // invierte el arreglo tal que miArreglo = [ "3", "2", "1" ]
```

- **`shift`** remueve el primer elemento de un arreglo y retorna este elemento.

```
miArreglo = new Array ("1", "2", "3");
primero=miArreglo.shift(); // miArreglo ahora tiene ["2", "3"], primero es "1"
```

- **`unshift`** añade un nuevo elemento a un arreglo al principio del mismo y retorna la nueva longitud del array

```
miArreglo = new Array ("1", "2", "3");
primero=miArreglo.unshift("4"); // miArreglo ahora tiene ["4","1","2", "3"]
```

- **slice** (índice_inicial, índice_hasta_sin_incluir) extrae una seccción de un arreglo y retorna un nuevo arreglo.

```
   miArreglo = new Array ("a", "b", "c", "d", "e");
   miArreglo = miArreglo.slice(1,4); //comienza en el índice 1 y extrae todos los
elementos hasta el índice 4, retornando [ "b", "c", "d" ]
```

- **splice**(índice, contador_para_remover, añade_elemento1, añade_elemento2, ...) añade y/o elimina elementos de un arreglo.

```
   miArreglo = new Array ("1", "2", "3", "4", "5");
   miArreglo.splice(1,3,"a","b","c", "d"); // miArreglo ahora tiene ["1", "a", "b",
"c", "d", "5"]
   // este método comienza en el índice uno (o donde estaba el "2"), elimina 3
elementos de allí y luego inserta todos los elementos consecutivamente en su lugar.
```

- **sort** ordena los elementos de un array.

```
   miArreglo = new Array("Viento","Lluvia","Fuego")
   miArreglo.sort(); // ordena el arreglo tal que miArreglo = [ "Fuego", "Lluvia",
"Viento" ]
```

La ordenación se hace en orden alfabético, para especificar otro orden que no sea el alfabético, se debe pasar como argumento una función de comparación (recibe dos argumentos), la cual tiene la tarea de decidir cuál de los dos argumentos debería aparecer primero, esto es, si la función regresa un número menor a cero, entonces el primer argumento aparece antes que el segundo, pero si la función regresa un número mayor a cero, entonces el primer argumento aparece después que el segundo argumento, y la función regresara cero si los dos argumentos son iguales.
Analice el siguiente ejemplo y vea con atención cuál es su salida, compárelo con el ejemplo posterior.

```
var aNumeros = [1111,4,222,33];
aNumeros = aNumeros.sort();
document.write(aNumeros);
```

Salida:

1111,222,33,4

```
aNumeros = aNumeros.sort(function(a,b){return a-b;});
document.write(aNumeros);
```

Salida:

4,33,222,1111

foreach(): llama a una función para cada elemento del array

every() : Chequea si todos los elementos del array superan un test

filter() :Crea un nuevo array con todos los elementos del array superan un test

find() : Devuelve el valor del primer elemento del array que supera un test

findIndex() : Devuelve el índice del primer elemento del array que supera un test

some() : Chequea si algún elemento del array supera un test

Métodos Arrays (w3schools.com)