

Array

## Constructor

### Array()

Crea un nuevo objeto Array.

## Propiedades estáticas

### get Array[@@species]

La función del constructor se utiliza para crear objetos derivados.

## Métodos estáticos

### Array.from()

Crea una nueva instancia de Array a partir de `similarAUnArray`, un objeto iterable o parecido a un *array*.

### Array.isArray()

Devuelve `true` si valor es un *array*, y `false` en caso contrario.

### Array.of()

Crea una nueva instancia de Array con un número variable de parámetros, independientemente del número y del tipo de dichos parámetros.

## Propiedades de instancia

### Array.prototype.length

Indica el número de elementos de un *array*.

### Array.prototype[@@unscopables]

Símbolo que contiene todos los nombres de las propiedades que se excluyen de un ámbito de enlace `with`.

## Métodos de instancia

### Array.prototype.concat()

Devuelve un nuevo *array* que es la concatenación de aquél sobre el que se invoca, seguido de otros *array(s)* o valor(es).

[Array.prototype.copyWithin\(\)](#)

Copia una secuencia de elementos de un *array* dentro del propio *array*.

[Array.prototype.entries\(\)](#)

Devuelve un nuevo objeto `Array Iterator` que contiene los pares clave/valor para cada índice del *array*.

[Array.prototype.every\(\)](#)

Devuelve `true` si todos los elementos del *array* cumplen el predicado que recibe como parámetro.

[Array.prototype.fill\(\)](#)

Asigna un `valor` estático a todos los elementos del *array* entre las posiciones `inicio` y `fin`.

[Array.prototype.filter\(\)](#)

Devuelve un nuevo *array* que contiene todos los elementos de aquél para el cual se llama que cumplan el predicado que se le pasa como parámetro.

[Array.prototype.find\(\)](#)

Devuelve el primer `elemento` del *array* que cumpla el predicado que se pasa como parámetro, o `undefined` si ninguno lo cumple.

[Array.prototype.findIndex\(\)](#)

Devuelve el índice del primer elemento del *array* que cumpla el predicado que se pasa como parámetro, o `-1` si ninguno lo cumple.

[Array.prototype.forEach\(\)](#)

Llama a la función pasada como parámetro para todos los elementos del *array*.

[Array.prototype.includes\(\)](#)

Determina si el *array* contiene el `valorBuscado` y devuelve `true` o `false` según sea el caso.

[Array.prototype.indexOf\(\)](#)

Devuelve el índice del primer elemento del *array* que sea igual a `elementoBuscado`, o `-1` si no existe.

[Array.prototype.join\(\)](#)

Concatena en un string todos los elementos de un *array*.

[Array.prototype.keys\(\)](#)

Devuelve un nuevo `Array Iterator` que contiene las claves de cada índice del *array*.

[Array.prototype.lastIndexOf\(\)](#)

Devuelve el índice del último elemento del *array* que sea igual a `elementoBuscado`, o -1 si no existe.

[Array.prototype.map\(\)](#)

Devuelve un nuevo *array* que contiene el resultado de llamar a la función pasada como parámetro a todos los elementos del *array* sobre el que se invoca.

[Array.prototype.pop\(\)](#)

Elimina el último elemento de un *array*, y devuelve dicho elemento.

[Array.prototype.push\(\)](#)

Añade uno o más elementos al final de un *array* y devuelve el nuevo valor de su propiedad `length`.

[Array.prototype.reduce\(\)](#)

Aplica la función pasada como parámetro a un `acumulador` y a cada valor del *array*, que se recorre de izquierda a derecha, para reducirlo a un único valor.

[Array.prototype.reduceRight\(\)](#)

Aplica la función pasada como parámetro a un `acumulador` y a cada valor del *array*, que se recorre de derecha a izquierda, para reducirlo a un único valor.

[Array.prototype.reverse\(\)](#)

Invierte el orden de los elementos de un *array* (el primero pasa a ser el último y el último a ser el primero) en el propio *array*. Este método modifica el array.

[Array.prototype.shift\(\)](#)

Elimina el primer elemento de un *array*, y devuelve dicho elemento.

[Array.prototype.slice\(\)](#)

Extrae una porción del *array* sobre el que se llama y devuelve un nuevo *array*.

[Array.prototype.some\(\)](#)

Devuelve `true` si al menos un elemento del *array* cumple con el predicado que se pasa como parámetro.

[Array.prototype.sort\(\)](#)

Ordena los elementos de un *array*, modificando éste, y devuelve el array ordenado.

[Array.prototype.splice\(\)](#)

Añade, borra o modifica elementos de un *array*.

[Array.prototype.toLocaleString\(\)](#)

Devuelve un string adaptado a la configuración regional que representa el *array* y sus elementos. Redefine el método [Object.prototype.toLocaleString\(\)](#).

[Array.prototype.toString\(\)](#)

Devuelve un string que representa el *array* y sus elementos. Redefine el método [Object.prototype.toString\(\)](#).

[Array.prototype.unshift\(\)](#)

Añada uno o más elementos al inicio de un *array* y devuelve el nuevo valor de `length` para el *array* resultante.

[Array.prototype.values\(\)](#)

Devuelve un nuevo objeto `Array Iterator` que contiene los valores para cada índice del *array*.

[Array.prototype\[@@iterator\]\(\)](#)

Devuelve un nuevo objeto `Array Iterator` que contiene los valores para cada índice del *array*.