

GM2

Arrays

GM2

Los Arrays



Para JavaScript los arrays son un tipo especial de objetos.

Por esta razón disponemos de muchos **métodos** muy útiles a la hora de trabajar con la información que hay adentro.



“

Un array es una estructura de datos que puede almacenar una colección de elementos. Los elementos de un array están indexados, lo que significa que cada elemento tiene una posición numérica, conocida como índice, que comienza en 0.



”

Estructura de un array

Utilizamos corchetes `[]` para indicar el **inicio** y el **fin** de un array. Utilizamos comas `,` para **separar** sus elementos.

Dentro de un array, podemos almacenar la cantidad de elementos que queramos, sin importar el tipo de dato de cada uno.

Es decir, podemos tener en un mismo array datos de tipo string, number, boolean y todos los demás.

```
{ } let miArray = ['Star Wars', true, 23];
```

Posiciones dentro de un array

Cada dato de un array ocupa una posición numerada conocida como **índice**. La **primera posición** de un array es **siempre 0**.

```
{ } let pelisFavoritas = ['Batman', 'Spiderman', 'Iron Man'];
```



Para acceder a un elemento puntual de un array, nombramos al array y, **dentro de los corchetes**, escribimos el **índice** al cual queremos acceder.

```
{ } pelisFavoritas[2];  
// accedemos a la película Alien, el índice 2 del array
```

GM2

Posición **UNDEFINED** dentro de un array

Qué pasa si accedemos a una posición del array donde no existe el elemento

```
{ } let pelisFavoritas = ['Batman', 'Spiderman', 'Iron Man'];
```

0 1 2

Y accedemos al índice 3 del array de películas

```
{ } pelisFavoritas[3];  
// Veremos por consola undefined
```

Longitud de un array

Otra propiedad útil de los arrays es su longitud, o cantidad de elementos. Podemos saber el número de elementos usando la propiedad `length`.

```
{ } let pelisFavoritas = ['Star Wars', 'Kill Bill', 'Alien'];
```

$$1 + 1 + 1 = 3$$

Para acceder al total de elementos de **un array**, nombramos al array y, **seguido de un punto** `.`, escribiremos **la palabra** `length`.

```
{ } pelisFavoritas.length;  
// Devuelve 3, el número de elementos del array
```


GM2

Metodos

GM2

push()

.push()

Agrega uno o varios elementos al final del array.

- **Recibe** uno o más elementos como parámetros.
- **Retorna** la nueva longitud del array.

```
{}
```

```
let colores = ['Rojo', 'Naranja', 'Azul'];
colores.push('Violeta'); // retorna 4
console.log(colores); // ['Rojo', 'Naranja', 'Azul', 'Violeta']

colores.push('Gris', 'Oro');
console.log(colores);
// ['Rojo', 'Naranja', 'Azul', 'Violeta', 'Gris', 'Oro']
```

GM2

pop()

.pop()

Elimina el último elemento de un array.

- **No recibe** parámetros.
- **Devuelve** el elemento eliminado.

```
{}  
let series = ['Mad Men', 'Breaking Bad', 'The Sopranos'];  
  
series.pop();  
  
console.log(series); // ['Mad men', 'Breaking Bad']
```

GM2

shift()

.shift()

- El método `shift()` elimina el primer elemento del array y lo retorna. Este método modifica la longitud del array.

```
{}  
let series = ['Mad Men', 'Breaking Bad', 'The Sopranos'];  
  
series.shift();  
  
console.log(series); // ['Breaking Bad', 'The Sopranos']
```

GM2

unshift()

.unshift()

- El método `unshift()` agrega uno o más elementos al inicio del array, y devuelve la nueva longitud del array.

```
let series = ['Mad Men', 'Breaking Bad', 'The Sopranos'];

{} series.unshift('The Simpsons');

console.log(series); // ['The Simpsons', 'Mad Men', 'Breaking Bad', 'The Sopranos']
```

.includes()

También similar a `.indexOf()`, con la salvedad que retorna un booleano.

- **Recibe** un elemento a buscar en el array.
- **Retorna** *true* si encontró lo que buscábamos, *false* en caso contrario.

```
{}  
let frutas = ['Manzana', 'Pera', 'Frutilla'];  
  
frutas.includes('Frutilla');  
// Encontró lo que buscaba. Devuelve true  
  
frutas.includes('Banana');  
// No encontró lo que buscaba. Devuelve false
```

GM2

find()

GM2

MÉTODO FIND

El método **find()** devuelve el valor del primer elemento del Array que satisface la función de comprobación enviada por parámetro. Si ningún valor satisface la función de comprobación, se devuelve undefined.

```
let numeros    = [1, 2, 3, 4, 5];  
//La función parámetro generalmente es una función flecha sin cuerpo.  
let encontrado = numeros.find(function(numeros){  
    return numeros > 3  
}); //Encuentra 4
```

```
let nombres= ["Ana", "Ema", "Juan"];  
let nombreEma = nombres.find(function(nombre){  
    return nombre === "Ema"  
}); //Encuentra "Ema"  
let NombreEma = nombres.find(function(nombre){  
    return nombre === "Luis"  
}); //undefined
```

GM2

filter()

GM2

MÉTODO FILTER

El método **filter()** crea un nuevo Array con todos los elementos que cumplan la función de comprobación enviada por parámetro. Generalmente, se obtiene un Array con menos elementos que la lista a filtrar.

```
const numeros = [1, 2, 3, 4, 5];  
const filtro1 = numeros.filter(function(numero){  
    return numero > 3  
}); //Encuentra [4,5]  
  
const filtro2 = numeros.filter(function(numero){  
    return numero < 4  
}); //Encuentra [1,2,3]
```