



MÉTODOS DE ARREGLOS EN JAVASCRIPT

1. **push()**

Agrega uno o más elementos al final de un arreglo.

```
let array = [1, 2, 3];  
array.push(4); // array ahora es [1, 2, 3, 4]
```

2. **pop()**

Elimina el último elemento de un arreglo y lo retorna.

```
let array = [1, 2, 3, 4];  
let lastElement = array.pop(); // lastElement es 4 y array ahora es [1, 2, 3]
```

3. **shift()**

Elimina el primer elemento de un arreglo y lo retorna.

```
let array = [1, 2, 3];  
let firstElement = array.shift(); // firstElement es 1 y array ahora es [2, 3]
```

4. **unshift()**

Agrega uno o más elementos al principio de un arreglo.

```
let array = [2, 3];  
array.unshift(1); // array ahora es [1, 2, 3]
```



5. splice()

Cambia el contenido de un arreglo eliminando, reemplazando o agregando nuevos elementos.

```
let array = [1, 2, 3, 4];  
array.splice(1, 2, 'a', 'b'); // array ahora es [1, 'a', 'b', 4]
```

Elimina 2 elementos desde el índice 1 y agrega 'a' y 'b'.

6. slice()

Retorna una copia de una parte del arreglo dentro de un nuevo arreglo.

```
let array = [1, 2, 3, 4];  
let newArray = array.slice(1, 3); // newArray es [2, 3]
```

Copia desde el índice 1 hasta el índice 3 (sin incluirlo).

7. forEach()

Ejecuta una función sobre cada elemento del arreglo.

```
let array = [1, 2, 3];  
array.forEach(element => console.log(element * 2)); // Imprime 2, 4, 6
```

8. map()

Crea un nuevo arreglo con los resultados de la llamada a una función aplicada a cada uno de sus elementos.

```
let array = [1, 2, 3];  
let newArray = array.map(element => element * 2); // newArray es [2, 4, 6]
```



9. `filter()`

Crea un nuevo arreglo con todos los elementos que pasen la prueba implementada por la función dada.

```
let array = [1, 2, 3, 4];  
let newArray = array.filter(element => element > 2); // newArray es [3, 4]
```

10. `reduce()`

Aplica una función a un acumulador y a cada elemento del arreglo (de izquierda a derecha) para reducirlo a un único valor.

```
let array = [1, 2, 3, 4];  
let sum = array.reduce((accumulator, currentValue) => accumulator + currentValue, 0); //  
sum es 10
```

11. `find()`

Retorna el primer elemento del arreglo que cumpla con la condición dada en la función.

```
let array = [1, 2, 3, 4];  
let found = array.find(element => element > 2); // found es 3
```

12. `findIndex()`

Retorna el índice del primer elemento que cumpla con la condición dada en la función.

```
let array = [1, 2, 3, 4];  
let index = array.findIndex(element => element > 2); // index es 2
```



13. `includes()`

Determina si un arreglo incluye un determinado elemento.

```
let array = [1, 2, 3];  
let hasTwo = array.includes(2); // hasTwo es true
```

14. `join()`

Une todos los elementos de un arreglo en una cadena y la retorna.

```
let array = [1, 2, 3];  
let str = array.join('-'); // str es '1-2-3'
```

15. `sort()`

Ordena los elementos del arreglo y retorna el arreglo ordenado.

```
let array = [3, 1, 4, 1, 2];  
array.sort(); // array ahora es [1, 1, 2, 3, 4]
```

16. `reverse()`

Invierte el orden de los elementos del arreglo.

```
let array = [1, 2, 3];  
array.reverse(); // array ahora es [3, 2, 1]
```