# Métodos de Arrays en JavaScript: Guía y Ejemplos

## 1. .push()

**Agrega uno o más elementos al final del array.**

const arr = [1, 2];  
arr.push(3); // [1, 2, 3]

## 2. .pop()

**Elimina el último elemento y lo devuelve.**

const arr = [1, 2, 3];  
const ultimo = arr.pop(); // ultimo = 3, arr = [1, 2]

## 3. .shift()

**Elimina el primer elemento y lo devuelve.**

const arr = [1, 2, 3];  
const primero = arr.shift(); // primero = 1, arr = [2, 3]

## 4. .unshift()

**Agrega uno o más elementos al inicio del array.**

const arr = [2, 3];  
arr.unshift(1); // [1, 2, 3]

## 5. .concat()

**Combina dos o más arrays y devuelve uno nuevo.**

const a = [1, 2];  
const b = [3, 4];  
const c = a.concat(b); // [1, 2, 3, 4]

## 6. .join()

**Convierte todos los elementos en una cadena, separados por lo que definas.**

const arr = ["a", "b", "c"];  
arr.join("-"); // "a-b-c"

## 7. .slice()

**Devuelve una copia superficial de una parte del array.**

const arr = [1, 2, 3, 4];  
const sub = arr.slice(1, 3); // [2, 3]

## 8. .splice()

**Agrega, quita o reemplaza elementos en el array.**

const arr = [1, 2, 3, 4];  
arr.splice(1, 2); // Quita 2 elementos desde índice 1: [2, 3], arr ahora es [1, 4]

## 9. .indexOf()

**Devuelve el primer índice donde se encuentra el valor.**

const arr = [1, 2, 3, 2];  
arr.indexOf(2); // 1

## 10. .lastIndexOf()

**Devuelve el último índice donde se encuentra el valor.**

const arr = [1, 2, 3, 2];  
arr.lastIndexOf(2); // 3

## 11. .includes()

**Devuelve true si el array contiene el valor.**

const arr = [1, 2, 3];  
arr.includes(2); // true

## 12. .find()

**Devuelve el primer elemento que cumple la función.**

const arr = [1, 2, 3];  
arr.find(x => x > 1); // 2

## 13. .findIndex()

**Devuelve el índice del primer elemento que cumple la función.**

const arr = [1, 2, 3];  
arr.findIndex(x => x > 1); // 1

## 14. .some()

**Devuelve true si al menos uno cumple la función.**

const arr = [1, 2, 3];  
arr.some(x => x > 2); // true

## 15. .every()

**Devuelve true si todos cumplen la función.**

const arr = [1, 2, 3];  
arr.every(x => x > 0); // true

## 16. .filter()

**Crea un nuevo array con los que cumplen la función.**

const arr = [1, 2, 3];  
arr.filter(x => x > 1); // [2, 3]

## 17. .map()

**Crea un nuevo array aplicando una función a cada elemento.**

const arr = [1, 2, 3];  
arr.map(x => x \* 2); // [2, 4, 6]

## 18. .reduce()

**Reduce el array a un solo valor.**

const arr = [1, 2, 3];  
arr.reduce((acc, val) => acc + val, 0); // 6

## 19. .reduceRight()

**Reduce de derecha a izquierda.**

const arr = [1, 2, 3];  
arr.reduceRight((acc, val) => acc - val); // ((3-2)-1) = 0

## 20. .forEach()

**Ejecuta una función para cada elemento.**

const arr = [1, 2, 3];  
arr.forEach(x => console.log(x));

## 21. .sort()

**Ordena el array (por defecto, alfabéticamente).**

const arr = [3, 1, 2];  
arr.sort(); // [1, 2, 3]

## 22. .reverse()

**Invierte el orden de los elementos.**

const arr = [1, 2, 3];  
arr.reverse(); // [3, 2, 1]

## 23. .flat()

**Aplana arrays anidados.**

const arr = [1, [2, [3]]];  
arr.flat(2); // [1, 2, 3]

## 24. .flatMap()

**Mapea y aplana un nivel.**

const arr = [1, 2];  
arr.flatMap(x => [x, x \* 2]); // [1, 2, 2, 4]

## 25. .fill()

**Rellena el array con un valor.**

const arr = [1, 2, 3];  
arr.fill(0); // [0, 0, 0]

## 26. .copyWithin()

**Copia parte del array dentro del mismo array.**

const arr = [1, 2, 3, 4];  
arr.copyWithin(1, 2); // [1, 3, 4, 4]

## 27. .entries()

**Devuelve un iterador de pares [índice, valor].**

const arr = ["a", "b"];  
for (let [i, v] of arr.entries()) {  
 console.log(i, v);  
}

## 28. .keys()

**Devuelve un iterador de los índices.**

const arr = ["a", "b"];  
for (let i of arr.keys()) {  
 console.log(i);  
}

## 29. .values()

**Devuelve un iterador de los valores.**

const arr = ["a", "b"];  
for (let v of arr.values()) {  
 console.log(v);  
}

## 30. .toString()

**Convierte el array a cadena separada por comas.**

const arr = [1, 2, 3];  
arr.toString(); // "1,2,3"

## 31. .at()

**Devuelve el elemento en el índice dado, permite negativos.**

const arr = [1, 2, 3];  
arr.at(-1); // 3

## 32. Array.from() (estático)

**Crea un array a partir de iterable o array-like.**

Array.from("hola"); // ["h", "o", "l", "a"]

## 33. Array.isArray() (estático)

**Verifica si es un array.**

Array.isArray([1,2,3]); // true

## 34. Array.of() (estático)

**Crea un array con los argumentos pasados.**

Array.of(1, 2, 3); // [1, 2, 3]