**Resumen de Métodos de Arrays en JavaScript**

**Métodos Tradicionales (Viejo JavaScript)**

Estos métodos han estado disponibles en JavaScript desde sus primeras versiones y son fundamentales para la manipulación básica de arrays.

**1. push()**

* **Descripción**: Agrega uno o más elementos al final de un array.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3];

numeros.push(4, 5);

console.log(numeros); // [1, 2, 3, 4, 5]

**2. pop()**

* **Descripción**: Elimina el último elemento de un array y lo devuelve.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3, 4];

let ultimoNumero = numeros.pop();

console.log(ultimoNumero); // 4

console.log(numeros); // [1, 2, 3]

**3. shift()**

* **Descripción**: Elimina el primer elemento de un array y lo devuelve.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3];

let primerNumero = numeros.shift();

console.log(primerNumero); // 1

console.log(numeros); // [2, 3]

**4. unshift()**

* **Descripción**: Agrega uno o más elementos al principio de un array.
* **Ejemplo**:

javascript

Copiar código

let numeros = [2, 3];

numeros.unshift(1);

console.log(numeros); // [1, 2, 3]

**5. concat()**

* **Descripción**: Combina dos o más arrays y devuelve un nuevo array.
* **Ejemplo**:

javascript

Copiar código

let numeros1 = [1, 2];

let numeros2 = [3, 4];

let todosLosNumeros = numeros1.concat(numeros2);

console.log(todosLosNumeros); // [1, 2, 3, 4]

**6. join()**

* **Descripción**: Combina todos los elementos de un array en una cadena, separados por un delimitador.
* **Ejemplo**:

javascript

Copiar código

let palabras = ["Hola", "mundo"];

let frase = palabras.join(" ");

console.log(frase); // "Hola mundo"

**7. reverse()**

* **Descripción**: Invierte el orden de los elementos de un array en su lugar.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3];

numeros.reverse();

console.log(numeros); // [3, 2, 1]

**8. slice()**

* **Descripción**: Devuelve una copia superficial de una parte del array seleccionada desde start hasta end (sin incluir end).
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3, 4, 5];

let algunosNumeros = numeros.slice(1, 3);

console.log(algunosNumeros); // [2, 3]

**9. splice()**

* **Descripción**: Cambia el contenido de un array eliminando, reemplazando o agregando nuevos elementos.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3, 4, 5];

numeros.splice(2, 1); // Elimina 1 elemento desde el índice 2

console.log(numeros); // [1, 2, 4, 5]

**10. sort()**

* **Descripción**: Ordena los elementos de un array en su lugar y devuelve el array.
* **Ejemplo**:

javascript

Copiar código

let letras = ["d", "a", "c", "b"];

letras.sort();

console.log(letras); // ["a", "b", "c", "d"]

**Métodos Introducidos en ES5 y Posteriores**

Estos métodos fueron introducidos en versiones más recientes de ECMAScript (ES5, ES6 y posteriores) y permiten un manejo más funcional y eficiente de los arrays.

**11. forEach() (ES5)**

* **Descripción**: Ejecuta una función proporcionada una vez por cada elemento del array.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3];

numeros.forEach(numero => {

console.log(numero);

});

// Imprime: 1, 2, 3

**12. map() (ES5)**

* **Descripción**: Crea un nuevo array con los resultados de la llamada a una función proporcionada en cada elemento del array.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3];

let cuadrados = numeros.map(numero => numero \* numero);

console.log(cuadrados); // [1, 4, 9]

**13. filter() (ES5)**

* **Descripción**: Crea un nuevo array con todos los elementos que pasen la prueba implementada por la función proporcionada.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3, 4, 5];

let pares = numeros.filter(numero => numero % 2 === 0);

console.log(pares); // [2, 4]

**14. reduce() (ES5)**

* **Descripción**: Aplica una función a un acumulador y a cada valor de un array (de izquierda a derecha) para reducirlo a un único valor.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3, 4];

let suma = numeros.reduce((acumulador, valorActual) => acumulador + valorActual, 0);

console.log(suma); // 10

**15. reduceRight() (ES5)**

* **Descripción**: Similar a reduce(), pero trabaja de derecha a izquierda.
* **Ejemplo**:

javascript

Copiar código

let letras = ["a", "b", "c"];

let combinacion = letras.reduceRight((acumulador, letra) => acumulador + letra);

console.log(combinacion); // "cba"

**16. some() (ES5)**

* **Descripción**: Verifica si al menos un elemento en el array cumple con la función de prueba.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3, 4];

let hayMayoresQueTres = numeros.some(numero => numero > 3);

console.log(hayMayoresQueTres); // true

**17. every() (ES5)**

* **Descripción**: Verifica si todos los elementos en el array cumplen con la función de prueba.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3, 4];

let todosSonPares = numeros.every(numero => numero % 2 === 0);

console.log(todosSonPares); // false

**18. indexOf() (ES5)**

* **Descripción**: Devuelve el primer índice en el que se encuentra un elemento dado en el array, o -1 si no se encuentra.
* **Ejemplo**:

javascript

Copiar código

let letras = ["a", "b", "c", "d"];

let indice = letras.indexOf("c");

console.log(indice); // 2

**19. lastIndexOf() (ES5)**

* **Descripción**: Devuelve el último índice en el que se encuentra un elemento dado en el array, o -1 si no se encuentra.
* **Ejemplo**:

javascript

Copiar código

let letras = ["a", "b", "c", "b"];

let indice = letras.lastIndexOf("b");

console.log(indice); // 3

**20. find() (ES6)**

* **Descripción**: Devuelve el primer elemento en el array que cumple con la función de prueba.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3, 4];

let mayorQueDos = numeros.find(numero => numero > 2);

console.log(mayorQueDos); // 3

**21. findIndex() (ES6)**

* **Descripción**: Devuelve el índice del primer elemento en el array que cumple con la función de prueba, o -1 si no se encuentra.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3, 4];

let indice = numeros.findIndex(numero => numero > 2);

console.log(indice); // 2

**22. fill() (ES6)**

* **Descripción**: Llena todos los elementos de un array desde un índice de inicio hasta un índice de final con un valor estático.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3, 4];

numeros.fill(0, 1, 3);

console.log(numeros); // [1, 0, 0, 4]

**23. includes() (ES6)**

* **Descripción**: Verifica si un array incluye un cierto elemento, devolviendo true o false.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3];

let incluyeDos = numeros.includes(2);

console.log(incluyeDos); // true

**24. flat() (ES2019)**

* **Descripción**: Aplana un array en el nivel especificado.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, [2, [3, 4]], 5];

let aplanado = numeros.flat(2);

console.log(aplanado); // [1, 2, 3, 4, 5]

**25. flatMap() (ES2019)**

* **Descripción**: Mapea cada elemento usando una función de mapeo, luego aplana el resultado en un nuevo array.
* **Ejemplo**:

javascript

Copiar código

let numeros = [1, 2, 3];

let mapeadoYAplanado = numeros.flatMap(numero => [numero, numero \* 2]);

console.log(mapeadoYAplanado); // [1, 2, 2, 4, 3, 6]

**Resumen Final**

Este resumen cubre los métodos más comunes y útiles que se pueden utilizar con arrays en JavaScript. Los métodos tradicionales son esenciales para manipular arrays de manera básica, mientras que los métodos más recientes introducen capacidades avanzadas que permiten trabajar con arrays de forma más funcional y eficiente. Los ejemplos proporcionados están diseñados para ser claros y fáciles de entender, especialmente para aquellos que recién se están introduciendo en el mundo de la programación.