

■ Hoja de repaso rápido – Tema 3: Arrays (JavaScript)

1. Introducción

Un array es un tipo especial de objeto que almacena varios valores en una sola variable.
Usa índices numéricos (empiezan en 0). Puede contener valores de distintos tipos (números, cadenas, objetos, funciones...).
No tiene tamaño fijo.

Declaración

```
let a = ["uno", "dos", "tres"];  
let b = new Array("uno", "dos", "tres");
```

Acceso

```
a[0]; // primer elemento  
a[a.length - 1]; // último elemento
```

1.1. Desestructuración

Permite extraer valores de un array en variables individuales:
`const [x, y, z] = [10, 20, 30];`

1.2. Operador ... (Spread / Rest)

Spread: `const nuevo = [...arr1, ...arr2];`
Rest: `function sumar(...nums) { console.log(nums); }`

2. Añadir y eliminar elementos

`push()` → añade al final
`unshift()` → añade al inicio
`pop()` → elimina el último
`shift()` → elimina el primero

2.1. `slice()` y `splice()`

`slice(inicio, fin)` → devuelve una copia parcial (no modifica).
`splice(pos, cantidad, nuevos...)` → modifica el array (elimina o añade).

3. Métodos de uso habitual

length → número de elementos.

sort() → ordena alfabéticamente (Unicode).

Para números: arr.sort((a,b) => a - b);

Conversión

toString() → convierte en cadena separada por comas.

join(sep) → une con separador personalizado.

split(sep) → cadena a array.

Otros métodos

concat() → une arrays.

reverse() → invierte orden.

indexOf() / lastIndexOf() → buscan posiciones.

4. Funciones Flecha (Arrow Functions)

Sintaxis: const f = (x) => x * 2;

Ventajas: sintaxis corta y retorno implícito.

Limitaciones: sin this, sin arguments, no constructores.

4.2. Métodos funcionales

filter() → filtra elementos → nuevo array.

find() / findIndex() → busca elementos.

every() / some() → todos / alguno cumple.

map() → transforma elementos → nuevo array.

reduce() → reduce a un valor único.

forEach() → ejecuta función por elemento.

includes() → comprueba si contiene valor.

Array.from() → crea array desde iterable.

5. Copia de arrays

Los arrays son objetos por referencia.

let a = [1,2,3];

let b = a; // referencia (mismo array)

```
let c = a.slice(); // copia independiente
```