



IES JAROSO

Desarrollo Web Entorno Cliente

TEMA 4:



PROGRAMACIÓN CON FUNCIONES, ARRAYS Y OBJETOS DEFINIDOS POR EL USUARIO.

Resumen Arrays paralelos y multidimensionales en JavaScript

1. Qué es un Array

Un **array** es una estructura de datos que permite guardar **varios valores** en una sola variable.

```
let frutas = ["Manzana", "Naranja", "Plátano"];
```

Características:

- Los elementos se almacenan en **posiciones numeradas (índices)**.
- El primer elemento está en el índice **0**.
- La propiedad **.length** devuelve el número de elementos.

```
console.log(frutas[0]);      // Manzana  
console.log(frutas.length); // 3
```

2. Arrays paralelos

Definición:

Cuando varios arrays usan **el mismo índice** para guardar información relacionada.

 **Ejemplo:**

```
let profesores = ["Cristina", "Catalina", "Vieites", "Benjamín"];
```

```
let asignaturas = ["Seguridad", "Bases de Datos", "Sistemas",  
"Redes"];  
let alumnos     = [24, 17, 28, 26];
```

Todos los arrays comparten el mismo índice:

- `profesores[0]` → "Cristina"
- `asignaturas[0]` → "Seguridad"
- `alumnos[0]` → 24

Uso típico: recorrerlos en paralelo

```
for (let i = 0; i < profesores.length; i++) {  
  document.write(`${profesores[i]} imparte ${asignaturas[i]} y tiene  
  ${alumnos[i]} alumnos.<br>`);  
}
```

Ventajas:

- Fácil de entender.
- Muy útil cuando cada tipo de dato debe guardarse por separado.

Inconveniente:

Hay que mantener **la misma longitud** en todos los arrays para evitar errores.

3. Arrays multidimensionales

Definición:

Un **array** que contiene otros **arrays** en sus posiciones.

Se usan para almacenar **tablas de datos** (filas y columnas).

Ejemplo:

```
let datos = [  
  ["Cristina", "Seguridad", 24],  
  ["Catalina", "Bases de Datos", 17],  
  ["Vieites", "Sistemas", 28],  
  ["Benjamín", "Redes", 26]
```

```
];
```

Acceso a los datos:

```
console.log(datos[0][0]); // "Cristina"  
console.log(datos[1][1]); // "Bases de Datos"  
console.log(datos[3][2]); // 26
```

Recorrido completo:

```
for (let i = 0; i < datos.length; i++) {  
    document.write(`${datos[i][0]} imparte ${datos[i][1]} y tiene  
    ${datos[i][2]} alumnos.<br>`);  
}
```

Ventajas:

- Agrupa los datos relacionados en una sola estructura.
- Más ordenado que varios arrays paralelos.

4. Visualización en tabla HTML

```
document.write("<table border='1'>");  
for (let i = 0; i < datos.length; i++) {  
    document.write("<tr>");  
    for (let j = 0; j < datos[i].length; j++) {  
        document.write(`<td>${datos[i][j]}</td>`);  
    }  
    document.write("</tr>");  
}  
document.write("</table>");
```

Resultado:

| Nombre | Asignatura | Alumnos |
|----------|------------|---------|
| Cristina | Seguridad | 24 |

| | | |
|----------|----------------|----|
| Catalina | Bases de Datos | 17 |
| Vieites | Sistemas | 28 |
| Benjamín | Redes | 26 |



5. Diferencias clave

| Concepto | Arrays Paralelos | Arrays Multidimensionales |
|------------|------------------------------------|--------------------------------------|
| Estructura | Varios arrays separados | Un solo array con arrays dentro |
| Acceso | <code>array1[i], array2[i]</code> | <code>array[i][j]</code> |
| Ventaja | Simplicidad | Organización y cohesión |
| Uso típico | Datos dispersos o de distinto tipo | Tablas, matrices, datos relacionados |