

CURSO DE JAVASCRIPT

**Corporación Ecuatoriana para el Desarrollo de la
Investigación y la Academia**

Ing. María Fernanda Granda J., PhD

25-11-2025

**“Conectando ideas,
transformando
sociedades”**

Módulo 5: Arreglos en JavaScript

- Creación de un arreglo.
- Acceso a los datos de un arreglo.
- Métodos de arreglos.
- Matrices multidimensionales.

¿Qué son los arreglos (arrays)?

- Son objetos que almacenan varios valores en una sola variable.
- Útiles para trabajar con grupos de elementos de forma organizada y permiten acceder a ellos mediante un índice numérico.



Operaciones con Arreglos

Crear un arreglo

- Se pueden crear utilizando corchetes [] y los elementos se separan con comas.

```
// arreglo de strings
let cars=["bmw", "volvo","honda"];
// arreglo de números
let numeros=[4,8,10,3];
// arreglo de valores de verdad
let verdades=[true, false, true];

//Pueden ser de distintos tipos
let persona=[702323345,'José', 'Pérez', 1981, false];
```

Operaciones con Arreglos

Crear un arreglo

- También puedes usar el constructor de Arreglo para crear un arreglo.
- En este ejemplo se creará un arreglo de longitud 2 y ninguno de los elementos son definidos en él

```
let elementos= new Array(2);  
console.log(elementos);
```

```
[ <2 empty items> ]
```

Operaciones con Arreglos

Crear un arreglo

- En este ejemplo se creará un arreglo de longitud dos con elementos 1 y 2 en él.

```
let elementos= new Array(1,2);  
console.log(elementos);
```

```
[ 1, 2 ]
```

Operaciones con Arreglos

- Acceder a un ítem del arreglo. Ejemplo la posición 0.

```
console.log("Contenido del arreglo: "+ cars);  
console.log("Valor de la posición 0: "+ cars[0]);
```

```
Contenido del arreglo: bmw,volvo,honda  
Valor de la posición 0: bmw
```

Operaciones con Arreglos

- Obtener el número de elementos del arreglo.

```
let len = cars.length;
```

```
console.log("Número de elementos: "+ cars.length);
```

```
Número de elementos: 3
```


Operaciones con Arreglos

- Acceder al último elemento del arreglo.

```
let last = cars[cars.length - 1];
```

```
console.log("Último elemento del arreglo: "+ cars[cars.length-1]);
```

```
Último elemento del arreglo: honda
```

Operaciones con Arreglos

- Recorrer los elementos de un arreglo.

```
cars.forEach((item, index, array) => {console.log(item, index);});
```

Contenido del Arreglo

bmw 0

volvo 1

honda 2

Métodos de Arreglos

- **toString():** Convertir un arreglo en String (texto)

```
let texto=cars.toString();  
console.log("Convertir a String: "+texto);
```

```
Convertir a String:  bmw,volvo,honda
```

Métodos de Arreglos

- **pop()**: eliminar el último elemento del arreglo

```
let ultimo=cars.pop();  
console.log("Obtiene y elimina el último elemento del arreglo: "+ ultimo);  
console.log("Contenido del arreglo: "+ cars);
```

```
Obtiene y elimina el último elemento del arreglo: honda  
Contenido del arreglo: bmw,volvo
```

Métodos de Arreglos

- **push():** agregar un nuevo elemento al final del arreglo

```
// agrega un elemento al final del arreglo  
cars.push("mercedes");  
console.log("Agrega un elemento al final del arreglo: "+ cars);
```

```
Agrega un elemento al final del arreglo: bmw,volvo,mercedes
```

Métodos de Arreglos

- **shift():** eliminar el primer elemento del arreglo

```
let primer=cars.shift();  
console.log("Obtiene y elimina el primer elemento del arreglo: "+ primer);  
console.log("Contenido del arreglo: "+ cars);
```

```
Obtiene y elimina el primer elemento del arreglo: bmw  
Contenido del arreglo: volvo,mercedes
```

Métodos de Arreglos

- **unshift():** adicionar un nuevo elemento al inicio del arreglo

```
// agrega un elemento al inicio del arreglo  
cars.unshift("hyundai");  
console.log("Contenido del arreglo: "+ cars);
```

```
Contenido del arreglo: hyundai,volvo,mercedes
```

Métodos de Arreglos

- **concat():** unir el arreglo con otro arreglo existente.

```
// unir arreglos  
let motos=["yamaha","suzuki","royal enfield"];  
let vehiculos=cars.concat(motos);
```

Contenido de vehículos : hyundai,volvo,mercedes,yamaha,suzuki,royal enfield

- Podemos combinar más de dos arreglos. Podemos combinar cualquier número de arreglos con esta sintaxis:

```
array.concat(arr1, arr2, ..., ..., ..., arrN);
```


Métodos de Arreglos

- **sort():** ordenar un arreglo en orden ascendente.
- Convierte los tipos de elemento en Strings y luego los arregla.
- Modifica el arreglo original.

```
// ordena los elementos del arreglo en orden ascendente  
console.log("Contenido de vehículos : "+ vehiculos.sort());
```

```
Contenido de vehículos : hyundai,mercedes,royal enfield,suzuki,volvo,yamaha
```

Métodos de Arreglos

- **sort()**. Para valores numéricos el resultado no es el esperado.

```
let numeros = [23, 5, 100, 56, 9, 13, 37, 10, 1];  
console.log(numeros.sort()); //[1,10,100, 13, 23, 37, 5, 56, 9]
```

Métodos de Arreglos

- **sort()**. Se puede escribir la función comparadora. Ejem:

```
let numeros = [23, 5, 100, 56, 9, 13, 37, 10, 1];  
function comparaAsc(a,b){  
    return (a-b);  
}  
console.log(numeros.sort(comparaAsc)); //[1,5, 9, 10, 13, 23, 37, 56, 100]
```

```
// otro ejemplo de función  
console.log(numeros.sort(function(a,b) {  
    return a-b;  
})));
```

```
// función flecha  
console.log(numeros.sort((a,b)=>(a-b))));
```

Métodos de Arreglos

- **sort()**. Para el orden descendente sólo hay que cambiar la comparación a (b-a).

```
// función flecha  
console.log(numeros.sort((a,b)=>(b-a)));
```

Métodos de Arreglos

- **reverse():** colocar los elementos del arreglo en orden inverso.
- Modifica el arreglo original.

```
// en un orden invertido  
console.log("Contenido de vehículos : "+ vehiculos.reverse());
```

```
Contenido de vehículos : yamaha,volvo,suzuki,royal enfield,mercedes,hyundai
```

Mapear, reducir y filtrar arreglos

- **map():** crear un nuevo arreglo y ejecutar una función sobre cada elemento del arreglo.

```
// uso de la función map para multiplicar el contenido del arreglo *2
let num1=[2,3,4,5,6,7];
let num2=num1.map(multiplicar);

function multiplicar(valor){
  return valor*2;
}
console.log("Original : "+ num1);
console.log("Resultado de map : "+ num2);
```

```
Original : 2,3,4,5,6,7
Resultado de map : 4,6,8,10,12,14
```

Mapear, reducir y filtrar arreglos

- **filter()**: tomar cada elemento del arreglo y aplicar un condicional para evaluarlo.

```
let num3=num1.filter(comparar);  
function comparar(valor){  
    return valor>4;  
}  
console.log("Resultado de filter : "+ num3);
```

```
Original : 2,3,4,5,6,7  
Resultado de filter : 5,6,7
```

Mapear, reducir y filtrar arreglos

- **reduce()**: reducir un arreglo de valores a un valor usando una función.

```
let num4= num1.reduce(sumar);  
function sumar(total, valor){  
    return total+valor;  
}  
console.log("Resultado de reduce : "+ num4);
```

```
Original : 2,3,4,5,6,7  
Resultado de reduce : 27
```


Mapear, reducir y filtrar arreglos

- **includes():** verificar si un array contiene un elemento específico. Devuelve true si el elemento existe en el array y false si no lo hace. También puedes especificar un índice de inicio para la búsqueda.

```
// Uso de includes  
let num1=[2,3,4,5,6,7];  
console.log("Num1 contiene el 4 : "+ num1.includes(4)); //true  
console.log("Num1 contiene el 8 : "+ num1.includes(8)); //false
```

```
Original : 2,3,4,5,6,7  
Num1 contiene el 4 : true  
Num1 contiene el 8 : false
```

Mapear, reducir y filtrar arreglos

- **indexOf():** encontrar el índice de la primera ocurrencia de un elemento en un array. Si el elemento no se encuentra, devuelve -1.

```
// Uso de indexOf  
let num1=[2,3,4,5,6,7];  
console.log("Posición del 4 : "+ num1.indexOf(4)); //2  
console.log("Posición del 8 : "+ num1.indexOf(8)); //-1
```

```
Original : 2,3,4,5,6,7  
Posición del 4 : 2  
Posición del 8 : -1
```

Mapear, reducir y filtrar arreglos

- **join():** retornar el contenido del arreglo como un string. No cambia el contenido del arreglo original.
- Cualquier separador puede ser utilizado. Por defecto es la coma.

```
// uso de join  
let num1=[2,3,4,5,6,7];  
console.log("Convertido en una lista: "+ num1.join("; "));
```

```
Original : 2,3,4,5,6,7  
Convertido en una lista: 2; 3; 4; 5; 6; 7
```

Matrices multidimensionales

- Crear un arreglo de arreglos

```
//Matriz
let precios=[
  ['bmw',150000],
  ['volvo',100000],
  ['honda',70000],
];
console.log("Matriz de precios: "+precios);
```

```
Matriz de precios: bmw,150000,volvo,100000,honda,70000
```

GRACIAS