



## Practica 04-10 JS Métodos Arrays para Programación funcional con funciones callback. from, of, at, Array[], with, unshift, shift, push, pop, fill, concat

Nombre		Curso	
Apellidos		Fecha	

### a) Metodos Estaticos

Array.from(fcallback) Array.of() constructor Array

### b) Categoria: Acceso

Array.at() Array[] Array.with(índice, valor)  
 Array.unshift(...Args )Array.shift() Array.push(...Args ) Array.pop()

### c) Categoria: Modificar Array

Array.fill(valor, inicio, fin )  
 Array.concat(Array)

## Practica Funcionales 1.

### 1. Array.from()

Convierte un objeto iterable (como un string o un objeto tipo array) en un array, aplicando una función de mapeo (fcallback).

#### Ejemplo:

```
const string = "12345";
const array = Array.from(string, num => Number(num) * 2); // Aplica fcallback: multiplica cada número por 2
console.log(array); // Resultado: [2, 4, 6, 8, 10]
```

### 2. Array.of()

Crea un nuevo array a partir de los elementos proporcionados como argumentos.

#### Ejemplo:

```
const array = Array.of(1, 2, 3, 4, 5);
console.log(array); // Resultado: [1, 2, 3, 4, 5]
```

### 3. Array.at()

Devuelve el elemento en la posición especificada. Los índices negativos cuentan desde el final.

#### Ejemplo:

```
const array = [10, 20, 30, 40, 50];
console.log(array.at(2)); // Resultado: 30 (índice positivo)
console.log(array.at(-1)); // Resultado: 50 (índice negativo)
```

## **4. Array.prototype.with(index, value)**

Crea una copia del array, reemplazando el valor en el índice especificado.

### **Ejemplo:**

```
const array = [10, 20, 30, 40, 50];
const newArray = array.with(2, 99); // Reemplaza el valor en el índice 2 con 99
console.log(newArray); // Resultado: [10, 20, 99, 40, 50]
```

## **5. Array.unshift(...args)**

Añade uno o más elementos al inicio del array y devuelve la nueva longitud.

### **Ejemplo:**

```
const array = [3, 4, 5];
array.unshift(1, 2); // Agrega 1 y 2 al inicio
console.log(array); // Resultado: [1, 2, 3, 4, 5]
```

## **6. Array.shift()**

Elimina el primer elemento del array y lo devuelve.

### **Ejemplo:**

```
const array = [1, 2, 3, 4, 5];
const firstElement = array.shift();
console.log(firstElement); // Resultado: 1
console.log(array); // Resultado: [2, 3, 4, 5]
```

## **7. Array.push(...args)**

Añade uno o más elementos al final del array y devuelve la nueva longitud.

### **Ejemplo:**

```
const array = [1, 2, 3];
array.push(4, 5); // Añade 4 y 5 al final
console.log(array); // Resultado: [1, 2, 3, 4, 5]
```

## **8. Array.pop()**

Elimina el último elemento del array y lo devuelve.

### **Ejemplo:**

```
const array = [1, 2, 3, 4, 5];
const lastElement = array.pop();
console.log(lastElement); // Resultado: 5
console.log(array); // Resultado: [1, 2, 3, 4]
```

## **9. Array.fill(valor, inicio, fin)**

Rellena los elementos del array con un valor estático desde el índice de inicio hasta el de fin (no inclusivo).

### **Ejemplo:**

```
const array = [1, 2, 3, 4, 5];
array.fill(0, 1, 4); // Rellena con 0 desde índice 1 hasta 3
console.log(array); // Resultado: [1, 0, 0, 0, 5]
```

## **10. Array.concat(arra)**

Combina dos o más arrays y devuelve uno nuevo.

### **Ejemplo:**

```
const array1 = [1, 2, 3];
const array2 = [4, 5, 6];
const combinedArray = array1.concat(array2);
console.log(combinedArray); // Resultado: [1, 2, 3, 4, 5, 6]
```

## Ejercicio 1: Inventario de productos en una tienda

**Contexto:** Estás manejando el inventario de una tienda y comienzas con un inventario básico de productos.

1. **Descripción:** Crea un array con los ID de los productos de la tienda [1, 2, 3, 4, 5] usando `Array.of()`. Luego:

- Un cliente devolvió productos dañados, así que usa `Array.fill()` para marcar los productos en las posiciones 2 a 4 con el código 7 (producto defectuoso).
- La tienda recibió nuevos productos, así que usa `Array.unshift()` para agregar dos nuevos IDs (0 y 0) al inventario (nuevos productos).
- Se vendió el último producto, así que usa `Array.pop()` para eliminar el último ID.
- Finalmente, llega un nuevo lote con IDs [10, 20, 30], por lo que debes combinar los arrays con `Array.concat()`.

2. **Resultado esperado:**

[0, 0, 1, 7, 7, 10, 20, 30]

## Ejercicio 2: Lista de empleados

**Contexto:** Estás gestionando una lista de empleados de una empresa.

1. **Descripción:** Crea un array de empleados a partir de las iniciales de sus nombres usando `Array.from()` a partir del string "abcdef" (por ejemplo: Ana, Bruno, Carlos, Daniela, Ernesto, Fernanda). Luego:

- Usa `Array.at()` para obtener el nombre del tercer y último empleado.
- El segundo empleado ha cambiado de departamento, así que usa `Array.with()` para cambiar su inicial por "X".
- El primer empleado se ha jubilado, por lo que debes eliminarlo con `Array.shift()`.
- Finalmente, has contratado a dos nuevos empleados, así que usa `Array.push()` para añadir "Y" y "Z" al final de la lista.

2. **Resultado esperado:**

['b', 'X', 'd', 'e', 'Y', 'Z']

## Ejercicio 3: Gestión de calificaciones

**Contexto:** Eres un profesor que está gestionando las calificaciones de los estudiantes.

1. **Descripción:** Tienes un array que contiene las calificaciones de los estudiantes [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]. Luego:

- Algunas calificaciones deben ser ajustadas, así que usa `Array.fill()` para modificar las calificaciones entre las posiciones 4 y 8 por 99 (ajuste especial).

- Se agregaron estudiantes a la lista, así que usa `Array.unshift()` para agregar las calificaciones de tres nuevos estudiantes (5, 6, 7).
- Un estudiante ha sido transferido y uno más se graduó, por lo que usa `Array.pop()` y `Array.shift()` para eliminar las calificaciones del último y primer estudiante respectivamente.

## 2. Resultado esperado:

[6, 7, 10, 11, 99, 99, 99, 99, 15, 16, 17, 18, 19]

## Ejercicio 4: Precio de productos

**Contexto:** Estás gestionando los precios de productos de un supermercado.

1. **Descripción:** Tienes un array con los precios de algunos productos [100, 200, 300, 400].

Luego:

- Usa `Array.push()` para añadir los precios de nuevos productos (500 y 600).
- El primer producto ya no está en venta, por lo que usa `Array.shift()` para eliminar su precio.
- El tercer producto tiene un nuevo precio, así que usa `Array.with()` para cambiar su precio a 999.
- Finalmente, tienes una nueva lista de precios [700, 800, 900], así que combina los arrays con `Array.concat()`.

## 2. Resultado esperado:

[200, 300, 999, 500, 600, 700, 800, 900]

## Ejercicio 5: Modificación de texto en una base de datos

**Contexto:** Estás procesando un texto que se va a almacenar en una base de datos.

1. **Descripción:** Crea un array de caracteres a partir del texto "hello world" usando `Array.from()`. Luego:

- Usa `Array.at()` para obtener el quinto carácter (debes validar que es una letra válida) y el último.
- Usa `Array.unshift()` para añadir un marcador especial "#" al inicio del array.
- Reemplaza todos los espacios con el carácter "\*" usando `Array.fill()`.
- Combina el texto con una nueva frase exclamativa agregando ['!', '!', '!'] al final con `Array.concat()`.

## 2. Resultado esperado:

['!', 'h', 'e', 'l', 'l', '\*', 'w', 'o', 'r', 'l', 'd', '!', '!', '!']

## Ejercicio 6: Gestión de pedidos

**Contexto:** Estás gestionando los números de pedidos de una tienda en línea.

**1. Descripción:** Crea un array con los números de los pedidos pendientes [1, 2, 3, 4, 5, 6].

Luego:

- Usa `Array.at()` para obtener el número del tercer pedido y del último pedido.
- El primer pedido ha sido procesado, así que usa `Array.with()` para cambiar el número del primer pedido a 0 (pedido procesado).
- Los pedidos del 2 al 4 se han cancelado, así que usa `Array.fill()` para cambiar esos valores por 8 (pedido cancelado).
- El primer y último pedido han sido procesados, así que usa `Array.shift()` y `Array.pop()` para eliminarlos.
- Finalmente, se agregaron tres nuevos pedidos, así que usa `Array.push()` para agregar los pedidos 7, 8, 9.

**2. Resultado esperado:**

[8, 8, 8, 5, 6, 7, 8, 9]

## Conclusión:

Estos ejercicios ahora están contextualizados en situaciones más reales, como la gestión de inventarios, listas de empleados, calificaciones, precios de productos, procesamiento de texto y gestión de pedidos. ¡Espero que te ayuden a entender mejor cómo y cuándo usar estas funciones en situaciones prácticas!