

Ejercicio 1: Declaración y acceso a elementos de un array

Crea un array de 5 elementos con los nombres de tus provincias favoritas. Después:

1. Añade una provincia nueva al final del array.
2. Muestra la longitud del array y todos sus elementos usando `console.table`.
3. Usa `console.log` para acceder al tercer elemento del array utilizando `at()` y la notación de corchetes.

Declaración de arrays, métodos `at`, `push` y propiedad `length`. (Archivo: `appl.js`)

Ejercicio 2: Recorrido de arrays

Declara un array mixto que contenga 5 elementos de diferentes tipos de datos (números, strings, booleanos, etc.). Recorre este array utilizando:

1. Un bucle `for` tradicional.
2. Un bucle `for...in`.
3. Un bucle `for...of`.
4. El método `forEach`.

Bucle `for`, `for...in`, `for...of` y `forEach`. (Archivo: `appl.js`)

Ejercicio 3: Inserción y eliminación de elementos.

Dado el array `const numeros = [10, 20, 30, 40]`; realiza lo siguiente:

1. Inserta el número 5 al inicio del array.
2. Inserta el número 50 al final del array.
3. Elimina el primer elemento.
4. Elimina el último elemento.
5. Utiliza `splice` para eliminar el segundo elemento y añadir en su lugar los números 25 y 35.

Métodos `push`, `unshift`, `shift`, `pop`, y `splice`. (Archivos: `appll.js`, `applv.js`)

Ejercicio 4: Copias de arrays

Dado el array `const frutas = ['manzana', 'pera', 'uva', 'mango']`;

1. Haz una copia del array usando el método `slice`.
2. Haz otra copia utilizando el operador `spread`.
3. Modifica el array original añadiendo una fruta nueva al final.
4. Muestra ambos arrays copiados por consola y verifica que no han cambiado.

Métodos `slice`, `concat`, y operador `spread`. (Archivo: `appv.js`)

Ejercicio 5: Destructuring

Dado el array `const colores = ['rojo', 'verde', 'azul', 'amarillo'];`:

1. Usa destructuring para extraer el primer y segundo color en variables llamadas `primerColor` y `segundoColor`.
2. Usa destructuring para extraer el último color en una variable llamada `ultimoColor`.
3. Intercambia los valores de dos variables utilizando destructuring.

Destructuring en arrays. (Archivo: `appVI.js`)

Ejercicio 6: Método map

Dado el array `const numeros = [2, 4, 6, 8, 10];`:

1. Usa `map` para crear un nuevo array con el triple de cada número.
2. Usa `map` para crear un array que indique el índice junto al valor, por ejemplo: “0: 2”, “1: 4”, etc.
3. Usa `map` para añadir el prefijo “Número: ” a cada valor del array.

Método `map`. (Archivo: `appVII.js`)

Ejercicio 7: Arrays bidimensionales

Crea un array bidimensional de 3x2 con los nombres de frutas y sus respectivos precios, por ejemplo: `[['Manzana', 1], ['Pera', 1.5], ['Uva', 2]]`. Después:

1. Usa un bucle `for` anidado para recorrer y mostrar cada fruta y su precio.
2. Añade una nueva fruta con su precio al final del array.
3. Muestra el array completo en formato de tabla con `console.table`.

Arrays bidimensionales, bucles anidados, `forEach`. (Archivos: `appVIII.js`, `appIX.js`)