

Relación B

Arrays y objetos JSON básicos

Nota

Esta relación trabaja con los tipos de datos complejos más habituales de JavaScript: arrays (similares a listas de Python) y objetos literales (equivalentes a diccionarios en Python o listas con nombre en R). Se estudian también los métodos de arrays más usados y la notación JSON.

Ejercicio 1. Cree un array `notas` que contenga 8 calificaciones numéricas (entre 0 y 10) de un estudiante ficticio. Muestre por consola:

- El número de elementos del array.
- El primer y el último elemento.
- El array completo con `console.log()` y también en el documento con `document.write()`.
- La suma de todas las notas y la media (calcúlelas con un bucle `for` clásico).

Ejercicio 2. Sobre el array del ejercicio anterior, emplee los métodos `push()`, `pop()`, `shift()`, `unshift()` y `splice()` para:

- Añadir dos nuevas notas al final.
- Eliminar la última nota y mostrarla en un `alert()`.
- Eliminar la primera nota y mostrarla por consola.
- Añadir una nota en la posición 3 sin eliminar ningún elemento.
- Reemplazar el elemento de la posición 5 por un nuevo valor.

Muestre el estado del array tras cada operación.

Ejercicio 3. Dado el array `notas` del ejercicio anterior, use los métodos funcionales de arrays para obtener:

- Un nuevo array con todas las notas incrementadas en 1 punto (sin superar 10). Use `map()` junto con `Math.min()`.
- Un nuevo array solo con las notas aprobadas (≥ 5). Use `filter()`.
- La suma total de las notas usando `reduce()`.
- Si alguna nota es exactamente 10. Use `some()`.
- Si todas las notas son positivas. Use `every()`.

Ejercicio 4. Cree un array de cadenas `asignaturas` con al menos 6 nombres de asignaturas de un grado en Matemáticas o Estadística. A continuación:

- Ordene el array alfabéticamente con `sort()` y muéstrelo.
- Ordene el array por longitud del nombre (de menor a mayor), pasando una función comparadora a `sort()`.
- Obtenga el índice de una asignatura dada usando `indexOf()` e `includes()`.
- Invierta el orden del array con `reverse()` y muéstrelo.
- Convierta el array en una sola cadena separada por comas usando `join()`.

Ejercicio 5. Cree un objeto literal `estudiante` que represente a un alumno de Estadística.

ca con los campos: `nombre`, `apellidos`, `edad`, `grado`, `notaMedia` y `aprobado` (booleano). Acceda a cada campo usando notación de punto y notación de corchetes. Añada dinámicamente un nuevo campo `email`. Elimine el campo `aprobado` con `delete`. Muestre el objeto resultante por consola y con `document.write()` usando `JSON.stringify()`.

Ejercicio 6. Cree un array `clase` con al menos 5 objetos estudiante (similares al del ejercicio anterior, cada uno con sus propios valores). Usando `map()`, `filter()` y `reduce()` sobre este array de objetos, calcule y muestre por consola:

- Un array solo con los nombres de los estudiantes.
- Un array con los estudiantes aprobados.
- La nota media de toda la clase.
- El nombre del estudiante con la nota más alta (use `reduce()` o `sort()` más acceso al primer elemento).

Ejercicio 7. Cree una función `aplanar(arr)` que reciba un array que puede contener tanto números como otros arrays de números anidados, y devuelva un único array plano con todos los números. Por ejemplo, `[1, [2, 3], [4, [5, 6]]]` debe devolver `[1, 2, 3, 4, 5, 6]`. Pruébela con el método nativo `flat(Infinity)` y compárela con su implementación manual. Muestre el resultado con `document.write()`.

Ejercicio 8. Dado el siguiente objeto que representa los resultados de un experimento estadístico:

```
const experimento = {
  nombre: "Lanzamiento de dado",
  muestras: 600,
  frecuencias: {"1":95, "2":102, "3":99, "4":101, "5":97, "6":106},
  metadata: { fecha: "2025-01-15", analista: "Garcia" }
};
```

- Acceda e imprima por consola el nombre del analista.
- Recorra las frecuencias con `for...in` e imprima cada par clave-valor.
- Convierta el objeto a JSON con `JSON.stringify()` con sangría de 2 espacios y muéstrelo en el documento.
- Cree una copia profunda del objeto usando `JSON.parse(JSON.stringify(...))` y modifique el analista en la copia. Compruebe que el original no cambia.

Ejercicio 9. Explore la desestructuración de arrays y objetos (ES6+):

- Dado un array `[10, 20, 30, 40, 50]`, use desestructuración para asignar los tres primeros valores a variables `a`, `b`, `c` y el resto a un array `resto` (con el operador `spread ...`).
- Dado el objeto estudiante del ejercicio 5, desestructure en una sola línea los campos `nombre`, `edad` y `notaMedia`.
- Intercambie los valores de dos variables `x` e `y` usando desestructuración de array (sin variable auxiliar).

Muestre los resultados por consola.

Ejercicio 10. Cree una función `estadisticos(datos)` que reciba un array de números

y devuelva un objeto con los campos: `min`, `max`, `media`, `rango` y `varianza`. Pruébela con al menos dos arrays distintos. Muestre el objeto resultado usando `JSON.stringify()` con sangría en el documento y también por consola.

Sugerencia: la varianza puede calcularse como la media de los cuadrados de las diferencias respecto a la media.

Ejercicio 11. Explore la diferencia entre la copia por referencia y por valor en JavaScript. Cree un array original y una copia usando los tres métodos siguientes:

- Asignación directa (`copia = original`).
- Operador spread (`copia = [...original]`).
- `JSON.parse(JSON.stringify(original))` para copia profunda.

Modifique un elemento en la copia y compruebe con `console.log()` si el original se ve afectado en cada caso. Repita el proceso con un objeto anidado para evidenciar la diferencia entre copia superficial y profunda.

Ejercicio 12. Cree un array multidimensional `matriz` de 3×3 que represente una tabla de contingencia sencilla. Acceda a la celda de la segunda fila y tercera columna. Recorra la matriz completa con dos bucles `for` anidados y muestre cada elemento con su posición `[fila][columna]` en el documento usando `document.write()`. Calcule también la suma de cada fila y la suma total.

Ejercicio 13. Convierta la siguiente cadena JSON en un objeto JavaScript, acceda a sus campos y genere una tabla HTML en el documento mostrando los datos:

```
const jsonStr = `[
  {"id":1,"nombre":"Algebra","creditos":6,"nota":7.5},
  {"id":2,"nombre":"Calculo","creditos":6,"nota":8.0},
  {"id":3,"nombre":"Estadistica","creditos":4.5,"nota":6.5},
  {"id":4,"nombre":"Probabilidad","creditos":4.5,"nota":9.0}
]`;
```

La tabla debe incluir una fila de cabecera y una fila por asignatura, más una última fila con la nota media ponderada por créditos.

Ejercicio 14. Cree un programa que simule un histograma de frecuencias. Genere un array de 100 números aleatorios entre 0 y 9 (use `Math.floor` y `Math.random`). Cree un objeto donde las claves sean los valores 0–9 y los valores sean la frecuencia absoluta de aparición de cada número. Muestre el histograma en el documento como una lista HTML, indicando para cada valor su frecuencia y un símbolo `|` por cada vez que aparece.