



## Programación Imperativa

# Arrays y colecciones

Vamos a afianzar nuestro conocimiento de arrays explorando distintas acciones sobre ellos. Veremos si se comportan como esperamos u ocurren resultados inesperados.

En primer lugar, practiquemos las acciones más comunes y necesarias:

- A. Acceder a elementos específicos de un array.
- B. Modificar cada uno de sus elementos e imprimirlos.
- C. Agregar elementos a un array.
- D. Extraer elementos de un array.
- E. Comparar elementos de un array con los elementos de otro.

## ¿Qué devuelven estos códigos?

En este ejercicio deberás pensar qué devuelven estos códigos sin probarlos en la consola.

1. 

```
let numbers =[22, 33, 54, 66, 72]  
console.log(numbers[numbers.length])
```
2. 

```
let grupoDeAmigos = [ "Harry", "Ron", "Hermione", "Spiderman", "Hulk",  
"Ironman","Penélope Glamour", "Pierre Nodoyuna","Patán" ]  
console.log(grupoDeAmigos[5])
```



3. `let str = "un string cualquiera"`  
`let arrayAleatorio= ["Digital", "House", "true", "string", "123", "false", "54", str]`  
`console.log(arrayAleatorio[arrayAleatorio.length - 1])`

## Colecciones de películas (y más...)

El tech leader del equipo necesita conocer la estructura de datos del nuevo proyecto. Para esto, deberás:

1. Crear la estructura adecuada para guardar las siguientes películas:  
*"star wars", "totoro", "rocky", "pulp fiction", "la vida es bella"*  
**Importante:** verificá que todo funciona correctamente accediendo a alguna de las películas una vez creada la estructura correspondiente.
2. Más tarde, de producción dieron el aviso de que las películas deberían estar todas en mayúsculas. Para esto solicitan que crees una función que reciba por parámetro un array y convierta el contenido de cada elemento a mayúsculas.  
**Pista:** revisá qué hace el método de strings `.toUpperCase()`.
3. Mientras trabajabas en el feature anterior, se dio el aviso de que también hay que crear otra estructura similar a la primera, pero con las siguientes películas animadas:  
*"toy story", "finding Nemo", "kung-fu panda", "wally", "fortnite"*  
Por lo tanto, te piden crear una función que reciba dos arrays como parámetros, para poder agregar los elementos contenidos en el segundo array dentro del primero, y así retornar un solo array con todas las películas como sus elementos.  
**Importante:** las películas animadas también deberían convertirse a mayúsculas.
4. Durante el proceso, uno de los desarrolladores advierte que el último elemento del array de películas animadas es un videojuego. Ahora tenés que editar el código y modificarlo de manera que puedas eliminar ese último elemento antes de migrar el contenido al array que contenga todas las películas.  
**PD:** por precaución, guardá el elemento que vas a eliminar en una variable.
5. Finalmente, te envían dos arrays con calificaciones que hacen distintos usuarios del mundo sobre las películas con el siguiente formato:  
`const asiaScores = [8, 10, 6, 9, 10, 6, 6, 8, 4];`



```
const euroScores = [8, 10, 6, 8, 10, 6, 7, 9, 5];
```

Te piden crear una función que compare las calificaciones e indique si son iguales o diferentes. Te confirman que están en el orden adecuado y que solo traen valores numéricos del 1 al 10.

**PD:** los elementos de los scores tanto de Asia como de Europa corresponden en orden al del array resultante de combinar películas con películas animadas. Es decir, el primer elemento del array de películas general corresponde al primer elemento de asiaScores y al primer elemento de euroScores, y así sucesivamente.

Para verificar que hasta acá viene todo bien, te recomendamos probar cada una de las funciones y testear su correcto funcionamiento.

Si llegaste hasta este punto y está todo bien, el tech leader del equipo debe estar extremadamente alegre con tu trabajo y desempeño. ¡Buen trabajo!

## Extra bonus

Si te quedaste con ganas de trabajar un poco más, te dejamos unos ejercicios extras para que puedas practicar más sobre arrays. A partir de acá, los ejercicios pueden empezar a escalar en dificultad tanto estructural como lógica. Recordá que siempre podés revisar el material visto. Si no encontrás algo, podés consultar con tus compañeros, así como también googlear la información que creas necesaria (recomendamos siempre documentaciones oficiales como la de [MDN Web Docs](https://developer.mozilla.org/en-US/docs/MDN/Community/MDN_web_docs), o foros reconocidos como [Stack Overflow](https://stackoverflow.com/)).

**PD:** si bien esos sitios están en español, tené en cuenta que la mayor cantidad de información va a estar en inglés.

## Array inverso

En este ejercicio deberás crear una función que devuelva un array con sus elementos invertidos, sin modificarlo. Luego, deberás hacer una función que lo modifique e invierta el orden de sus elementos.



1. Creá la función *imprimirInverso* que tome un array como argumento y que imprima en la consola cada elemento en orden inverso (no tenés que invertir el array).
2. Creá la función *inversor* que tome un array como argumento y devuelva uno nuevo invertido.

## sumaArray()

En este ejercicio, deberás crear una función *sumaArray()* que acepte un arreglo de números (3 elementos) y devuelva la suma de todos ellos. Ejemplo:

- `sumaArray([1,2,3])` // 6
- `sumaArray([10, 3, 10])` // 23
- `sumaArray([-5,100, 19])` // 114

## Simulación Array.join()

En este ejercicio deberás crear una función llamada *join()* que reciba un arreglo de strings de 4 elementos y simule el comportamiento del método `Array.join()`.

**Importante:** no podés usar el método `Array.join()` original.

Ejemplo:

- `join(["h","o","l","a"])` debe retornar el string "hola".
- `join(["c","h","a","u"])` debe retornar el string "chau".