



## 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 un array 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.



## ¿Que devuelven estos códigos?

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

1. `let numbers =[22, 33, 54, 66, 72]`

`console.log(numbers[numbers.length])`

**RES: undefined**

2. `let grupoDeAmigos = [ ["Harry", "Ron", "Hermione"], ["Spiderman", "Hulk", "Ironman"], ["Penélope Glamour", "Pierre Nodoyuna", "Patán"] ]`

`console.log(grupoDeAmigos[1][0])`

**RES: "Spiderman"**

3. `let str = "un string cualquiera"`

`let grupoDeAmigos = [ [45, 89, 0], ["Digital", "House", true], ["string", "123", "false", 54, true, str] ]`

`console.log(grupoDeAmigos[2][grupoDeAmigos[2].length - 1])`

**RES: "un string cualquiera"**



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

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

1. Nos piden crear la estructura adecuada para guardar las siguientes películas:

*"star wars", "totoro", "rocky", "pulp fiction", "la vida es bella".*

**Importante:** verifica que todo funciona correctamente accediendo a alguna de las películas una vez creada la estructura correspondiente

```
let peliculas = ["star wars", "totoro", "rocky", "pulp fiction", "la vida es bella"]
```

2. Más tarde de producción nos dieron el aviso de que las películas deberían estar todas en mayúsculas, para ellos nos solicitan que creemos una función que reciba por parámetro un array y convierta el contenido de cada elemento a mayúsculas.

Pista: revisa que hace el método de strings `.toUpperCase()`.

```
function convertirAMayusculas(array){  
  array[0] = array[0].toUpperCase()  
  array[1] = array[1].toUpperCase()  
  array[2] = array[2].toUpperCase()  
  array[3] = array[3].toUpperCase()  
  array[4] = array[4].toUpperCase()  
  return array  
}  
convertirAMayusculas(peliculas)
```



3. Mientras trabajabamos en la 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 nos piden crear una función que reciba dos arrays como parámetros, para poder agregar los elementos contenidos en el segundo array, dentro del primero para así retornar un solo array con todas las películas como sus elementos.

**Importante:** las pelis animadas también deberían convertirse a mayúsculas!

```
function pasajeDeElementos(array1, array2) {  
  array1.push(array2.pop().toUpperCase())  
  array1.push(array2.pop().toUpperCase())  
  array1.push(array2.pop().toUpperCase())  
  array1.push(array2.pop().toUpperCase())  
  array1.push(array2.pop().toUpperCase())  
  return array1  
}  
pasajeDeElementos(peliculas,pelisAnimadas);
```

4. Durante el proceso uno de los desarrolladores advierte que el último elemento del array de películas animadas es un videojuego. Ahora debemos editar nuestro código y modificarlo de manera que podamos eliminar ese último elemento antes de migrar el contenido al array que contenga todas las películas.

PD: por precaución guarda el elemento que vas a eliminar en una variable.



```
let pelisAnimadas = ["toy story", "finding Nemo", "kung-fu panda",  
"wally", "fortnite"]  
  
let videojuego = pelisAnimadas.pop()  
  
function pasajeDeElementos(array1, array2) {  
  array1.push(array2.pop().toUpperCase())  
  array1.push(array2.pop().toUpperCase())  
  array1.push(array2.pop().toUpperCase())  
  array1.push(array2.pop().toUpperCase())  
  return array1  
}  
  
console.log(pasajeDeElementos(peliculas, pelisAnimadas));
```

5. Finalmente, nos 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];
```

Nos piden crear una función que compare las calificaciones y nos diga si son iguales o diferentes. Nos 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 com de europa corresponden en orden al del array resultante de combinar películas con películas animadas, es decir, el primer elemento de tu array de películas general corresponde al primer elemento de asiaScores y al primer elemento de euroScores, y así sucesivamente.

```
const asiaScores = [8, 10, 6, 9, 10, 6, 6, 8, 4];
```



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

function compararCalificaciones(asia, europa) {
  let comparacionesAsiaEuropa = []
  comparacionesAsiaEuropa[0] = asia[0] === europa[0]
  comparacionesAsiaEuropa[1] = asia[1] === europa[1]
  comparacionesAsiaEuropa[2] = asia[2] === europa[2]
  comparacionesAsiaEuropa[3] = asia[3] === europa[3]
  comparacionesAsiaEuropa[4] = asia[4] === europa[4]
  comparacionesAsiaEuropa[5] = asia[5] === europa[5]
  comparacionesAsiaEuropa[6] = asia[6] === europa[6]
  comparacionesAsiaEuropa[7] = asia[7] === europa[7]
  comparacionesAsiaEuropa[8] = asia[8] === europa[8]
  return comparacionesAsiaEuropa
}

let comparaciones = compararCalificaciones(asiaScores, euroScores)
console.log(comparaciones);
```

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

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

Si te quedaste con ganas de trabajar un poco más, te dejamos unos ejercicios extras para que puedas practicar más sobre arrays!!.



## Arreglo Inverso

En este ejercicio deberás crear una función que devuelva un arreglo 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 Arreglo de 5 elementos como Argumento y que imprima en la consola cada elemento en orden inverso (no tenés que invertir el Arreglo).

```
function imprimirInverso(array) {  
  console.log(array[4])  
  console.log(array[3])  
  console.log(array[2])  
  console.log(array[1])  
  console.log(array[0])  
}  
imprimirInverso([1,2,3,4,5])
```

2. Creá la Función *inversor* que tome un Arreglo de 5 elementos como Argumento y devuelva uno nuevo invertido.

```
function inversor(array) {  
  let nuevoArray = []  
  nuevoArray.push(array.pop())  
  nuevoArray.push(array.pop())  
  nuevoArray.push(array.pop())  
  nuevoArray.push(array.pop())  
  nuevoArray.push(array.pop())  
  return nuevoArray  
}
```



```
}  
console.log(inversor(["a","b","c","d","e"]));
```

## 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

```
function sumaArray(arrNum) {  
    return arrNum[0] + arrNum[1] + arrNum[2]  
}  
  
console.log(sumaArray([-5,100,19]));
```

## 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.

Por ejemplo:

*`join(["h","o","l","a"])` debe retornar el string "hola".*





*join(["c","h","a","u"]) debe retornar el string "chau".*

```
function join(arr) {  
    return arr[0] + arr[1] + arr[2] + arr[3]  
}  
  
let str = ["h","o","l","a"]  
console.log(join(str));
```