

UF3.NF1.JavaScript Estructures de dades

Crea un projecte al teu IDE que es digui `JSEstructures_de_dades<nom><cognoms>` i realitza els següents exercicis. Treballa amb un repositori al `Git` per pujar cadascun dels exercicis amb un `commit` i adjunta en l'entrega la url.

1.- Crea un array amb 5 números de manera aleatòria. Utilitza les funcions necessàries per, en aquest ordre, ordenar de menor a major i obtenir el primer i últim valor.

```
let array : any[] = []
2 for (let i : number = 0; i < 5 ; i++){
3     array.push(Math.round(Math.random()*100))
4 }
new *
5 array.sort( compareFn: function (a,b){return a - b})
6 console.log(array)
7 console.log(array[0])
8 console.log(array[4])
9
```

```
[ 12, 50, 56, 62, 85 ]
12
85
```

2.- Utilitza l'operador de propagació per crear un altre array usant els valors de l'exercici anterior

```
JS ex1.js JS ex2.js x
1 let array : any[] = []
2 for (let i : number = 0; i < 5 ; i++){
3     array.push(Math.round(Math.random()*1000))
4 }
new *
5 array.sort( compareFn: function (a,b){return a - b})
6 console.log(array)
7 console.log(array[0])
8 console.log(array[4])
9
10 let array2 : any[] = [20,18,...array]
11 console.log(array2)
12
```

```
"C:\Program Files\nodejs\node.exe"
[ 208, 309, 551, 617, 966 ]
208
966
[
  20, 18, 208,
  309, 551, 617,
  966
]
```

3.- Fusiona dos arrays i esborren els valors duplicats.

```
JS ex1.js JS ex2.js JS ex3.js x
1 let array : number[] = [1,2,3,4,5,6]
2 let array2 : number[] = [4,5,6,7,8,9]
3
4 usage new *
5 function ex3 () : T[] {
6     let array3 : number[] = [...array, ...array2]
7     return array3.filter((value : number , index : number ) : boolean => array3.indexOf(value) === index)
8 }
9 console.log(ex3())
```

```
[
  1, 2, 3, 4, 5,
  6, 7, 8, 9
]
```

4.- Modifica l'exercici anterior per utilitzar una estructura que no permeti duplicats.

```
1 let array : number[] = [1,2,3,4,5,6]
2 let array2 : number[] = [4,5,6,7,8,9]
3 let array3 : Set<number> = new Set( values: [...array,...array2])
4 console.log(array3)
```

```
Set(9) { 1, 2, 3, 4, 5, 6, 7, 8, 9 }
```

5.- Codifica la funció map d'una Array perquè en creï una nova amb el valor cúbic de cada número.
Exemple: [1, 3, 6] => [1, 27, 216]

```
1 const mapToCubics = array => array.map(element => element ** 3);
2
3 let array : number[] = [1, 3, 6];
4 let arrayCubics = mapToCubics(array);
5
6 console.log("Array original:", array);
7 console.log("Array amb valors cúbics:", arrayCubics);
```

```
Array original: [ 1, 3, 6 ]
Array amb valors cúbics: [ 1, 27, 216 ]
```

6.- A partir d'una array de strings, es creï una nova amb aquelles que tinguin més d'una lletra 'a'

```
1 let filtre = array => array.filter(word => (word.match(/a/g)).length > 1);
2
3 let array : string[] = ["hola", "manolo", "patata", "mama"]
4 let arraydemesdunaA = filtre (array);
5
6 console.log("Array original:", array);
7 console.log("Nova array amb més d'una lletra 'a':", arraydemesdunaA);
8
```

Array original: ['hola', 'manolo', 'patata', 'mama']
Nova array amb més d'una lletra 'a': ['patata', 'mama']

7.- Crea una estructura Map i insereix 3 claus – valor. Les 3 claus han de ser de tipus de dades diferents. Et deixa?

```
1 let map : Map<any, any> = new Map()
2
3 let valor1 : Map<any, any> = map.set("hola", "String")
4 let valor2 : Map<any, any> = map.set(123, { propietat: 'Sóc un objecte!' })
5 let valor3 : Map<any, any> = map.set(true, ["hola que tal", "em dic manolo"])
6
7 console.log(map)
```

Map(3) {
 'hola' => 'String',
 123 => { propietat: 'Sóc un objecte!' },
 true => ['hola que tal', 'em dic manolo']
}

8.- Donada una paraula, insereix cada caràcter dins d'una cua. Extreu ara de la cua els caràcters un a un per poder decidir si és un palíndrom o no.

9.- Crea una Cua d'Objectes amb dades de 3 pel·lícules. Itera la cua per mostrar per pantalla la pel·lícula més antiga.

