

**Alumno/a: José Manuel González Peña****Listado arrays en JavaScript**

- 1. Indica los tres argumentos del método forEach a un array. Demuestra su uso mediante un ejemplo.**

Argumento de devolución de llamada	Definición
<i>value</i>	Valor del elemento de la matriz.
<i>index</i>	Índice numérico del elemento de la matriz.
<i>array1</i>	Objeto Array que contiene el elemento.

```
let a = [100,200,300];  
a.forEach(function(x) {alert(x);});
```

- 2. Indica la utilidad del operador in con los arrays. Demuestra su uso mediante un ejemplo.**

El operador in retorna true si la propiedad especificada está en el objeto especificado.

```
let a = [100,200,300];  
0 in a //devuelve true  
4 in a //deuelve false
```

- 3. Indica la función que comprueba si un objeto es o no un Array. Demuestra su uso mediante un ejemplo.**

```
let a = [100,200,300];  
Array.isArray(a);
```

- 4. Crea una función que cree un array de la dimensión indicada, cuyo contenido sean los números naturales comenzando desde 0**

```
function crearArray(argument) {  
    let array=[];  
    for (var i = 0 ; i <= argument; i++) {  
        array.push(i);  
    }  
    console.log(array);  
}
```

- 5. Crea una función que devuelva un array con cada uno de los argumentos.**

```
let array = [];  
function devolverArray() {  
    for (var i = 0; i <= arguments.length-1; i++) {  
        array.push(arguments[i]);  
    }  
    console.log(array);  
}
```

- 6. Crea una función que devuelva un array con cada uno de los argumentos. En caso de que alguno de sus argumentos sea un array, que introduzca sus elementos uno a uno.**

```
function devolverArray2() {
    for (var i = 0; i <= arguments.length - 1; i++) {
        if (Array.isArray(arguments[i])) {
            arr = arguments[i];
            for (y in arr) {
                array.push(arr[y]);
            }
        } else {
            array.push(arguments[i]);
        }
    }
    console.log(array);
}
```

- 7. Crea una función que elimine todos los undefined de un array.**

```
function borrarUn(argument) {
    if (Array.isArray(argument)) {
        for (i in argument) {
            if (argument[i]==undefined) {
                argument.splice(i,1);
            }
        }
    }
}
```

- 8. Indica la diferencia entre los siguientes métodos, y demuestra su uso con algunos arrays: Array.prototype.forEach(), Array.prototype.every(), Array.prototype.some() y Array.prototype.filter()**

- **Array.prototype.forEach():** Llama a una función para cada elemento del array.  

```
let matriz = [10, 20, 30,40,50];
matriz.forEach(function(x) { console.log(x); });
```
- **Array.prototype.every():** Devuelve true si cada elemento en este array satisface la función de testeo proporcionada.  

```
function mayorCero(elem, i) {
    return elem > 0;
}
[-10,5,4,-50].every(mayorCero);    // false
[1,2,3,4,5].every(mayorCero);    // true
```
- **Array.prototype.some():** Devuelve true si al menos un elemento en este array satisface la función de testeo proporcionada.  

```
function mayorCero2(elem, i) {
    return elem > 0;
}
[-10,-20].some(mayorCero2);    //false
[-10,5,4,-50].some(mayorCero2);    //true
```

- `Array.prototype.filter()`: Crea un nuevo array con todos los elementos de este array para los cuales la función de filtrado proporcionada
- ```
function mayorCero3(elem, i) {
    return elem > 0;
}
[-10,5,4,7,8,-50].filter(mayorCero3);
```

## 9. Averigua qué método es el más eficiente para manejarse con arrays. Compruébalo mediante `performance.now()` o similares

### 1. Introduce 10 elementos en un array mediante `push()`, `unshift()`, directamente, fijando tamaño en `new Array...`

```
function anadirArray() {
    let array = new Array(10);
    t0 = performance.now();
    array.push(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
    t1 = performance.now();
    console.log("Tiempo: " + (t1 - t0) + " milisegundos")
}
anadirArray();
function anadirArray2() {
    let array = new Array(10);
    t0 = performance.now();
    array.unshift(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
    t1 = performance.now();
    console.log("Tiempo: " + (t1 - t0) + " milisegundos")
}
anadirArray2();
```

### 2. Eliminar 10 elementos en un array mediante `pop()`, `shift()`, directamente, fijando tamaño...

```
function borrarArray() {
    let array = new Array(10);
    t0 = performance.now();
    for (var i = 0; i <= 10; i++) {
        array.pop(i);}
    t1 = performance.now();
    console.log("Tiempo: " + (t1 - t0) + " milisegundos")
}
borrarArray();
function borrarArray2() {
    let array = new Array(10);
    t0 = performance.now();
    for (var i = 0; i <= 10; i++) {
        array.shift(i);
    }
    t1 = performance.now();
    console.log("Tiempo: " + (t1 - t0) + " milisegundos")
    console.log(array);}
borrarArray2();
```