Métodos de arrays con callbacks

1) arr.sort([callback])

Ordena el array de forma ascendente o descendente, según como se lo indique en el callback.

Este método no retorna un nuevo array, simplemente modifica el array original al llamarlo.

Ejemplo

```
const arr = [5,4,8,1,9,2,3,10,6,7];
arr.sort( (a, b) => a-b );        //[1,2,3,4,5,6,7,8,9,10]
arr.sort( (a, b) => b-a );        //[10,9,8,7,6,5,4,3,2,1]
```

a-b => ascendente

b-a => descendente

2) arr.forEach([callback])

recorre cada elemento del array. Tampoco retorna nada, solo recorre los elementos y hay que indicar en el callback que hacer con esos elementos.

Ejemplo

Array de objetos

```
const empleados = [{},{},{},{},{},{}];
empleados.forEach(emp => alert("hola, soy " + emp.nombre + "\n");

empleados.forEach(emp => {
      if(emp.exp > 2){
            emp.sueldo += 20000;
      }
});
```

Supongamos que queremos subir el sueldo a los empleados con más de 2 años de experiencia

Recordemos que forEach significa "por cada" y nos puede servir mucho cuando queremos recorrer arrays de objetos.

3) arr.some([callback])

devuelve true si al menos uno de los elementos cumple con una condición escrita en el cuerpo del callback.

Ejemplo

```
const productos = [{}, {}, {}, {}];
//en este ejemplo se comprueba si al menos un producto está vencido
let hayVencido = productos.some(prod => prod.fechaVen < fechaActual);

if(hayVencidos){
	console.log(“tenes uno o más productos vencidos”);
}else{
	console.log(“nada está vencido”);
}
```

4) arr.every([callback])

Devuelve true si absolutamente TODOS los elementos del array cumplen con la condición.

Ejemplo

```
/*imaginemos un juego MOBA donde hayan equipos de 5 y para empezar a buscar partida, todos los miembros deben tener el atributo preparado en true*/
const equipo = [{}, {}, {}, {}, {}];

let preparados = equipo.every(miembro => miembro.preparado == true);

if(preparados){
	alert(“buscando partida”);
}else{
	Alert(“faltan miembros por prepararse”);
}
```

5) arr.map([callback])

modifica TODOS los elementos del array (modifica el array original). Si queremos modificar solo algunos elementos que cumplan con una condición, hay que usar un if dentro del cuerpo del callback.

Ejemplo

```
const empleados = [{},{},{},{},{},{}];

empleados.map(emp => emp.sueldo += 20000);   /*le subo el sueldo a todos los empleados*/
```

6) arr.filter([callback])

retorna un nuevo array con los elementos que cumplan con una condición. Cabe aclarar que al obtener los elementos, estos siguen estando en el array origial (no se eliminan).

Ejemplo

```
const juegos = [{},{},{},{},{},{}];

//obtengo juegos de distintas plataformas
const juegosPC = juegos.filter(juego => juego.plataforma == “pc”);

const juegosPS4 = juegos.filter(juegos => juegos.plataforma == “ps4”);
```

7) arr.reduce([callback])

Reduce todos los elementos del array a un solo elemento. Esto se entiende mejor con un ejemplo.

Ejemplo

```
const carrito = [{},{},{},{},{},{}];

let precioTotal = carrito.reduce((a,b) => a.precio + b.precio);

alert(`el precio total de la compra es ${precioTotal}`);
```