

# Funciones sobre array

## 1. **forEach()**

El método **forEach()** ejecuta la función indicada una vez por cada elemento del array. Este método puede ayudarte a recorrer los elementos de un *array*.

### EJE:

```
let coches = ["Ford", "Seat", "Kia", "Fiat"];  
coches.forEach(function(coche){  
  console.log(coche);  
});
```

# Funciones sobre array

## 2. map()

El método `map()` crea un nuevo array con los resultados de la llamada a la función indicada aplicados a cada uno de sus elementos.

EJE:

```
let preciosSinIva = [12, 30, 25, 20, 15, 30, 35];  
let preciosConIva = preciosSinIva.map(function(precio){  
  return precio + precio*0.21;  
});  
console.log(preciosConIva);
```

# Funciones sobre array

## 3. includes()

El método includes() determina si un array incluye un determinado elemento, devuelve true o false según corresponda. Utiliza el algoritmo sameValueZero para determinar si se encuentra el elemento dado.

### EJE:

```
let coches = ["Ford", "Seat", "Kia", "Fiat"];  
console.log(coches.includes("Ford"));
```

// El ejemplo devuelve **true**;

# Funciones sobre array

## 4. filter()

El método filter() crea un nuevo array con todos los elementos que cumplan la condición implementada por la función dada.

### EJE:

```
let edades = [15, 17, 19, 22, 35, 30, 35, 24];  
let mayoresDeEdad = edades.filter(function(mayor){  
  return mayor >= 18  
});  
console.log(mayoresDeEdad);  
// Devuelve un array : [19, 22, 35, 30, 35, 24];
```

# Funciones sobre array

## 5. `reduce()`

El método **`reduce()`** aplica una función a un acumulador y a cada valor de un array (de izquierda a derecha) para reducirlo a un único valor.

### EJE:

```
let valores = [1, 2, 3, 4, 5];  
console.log(valores.reduce(function(acumulador, valor){  
  return acumulador + valor;  
})))
```

# Funciones sobre array

## 6. `some()`

El método **`some()`** verifica si algún elemento de un array cumple con el test implementado por la función brindada. Si se aprueba, devuelve “true” de lo contrario “false”.

### EJE:

```
let precios = [10, 20, 30, 40, 50];  
console.log(precios.some(function(precio){  
  return precio > 20;  
}));
```

**// Devuelve true , porque hay algunos elementos del array que cumplen con la condición**

# Funciones sobre array

## 7. **every()**

El método **every()** verifica si todos los elementos en el arreglo pasan la prueba implementada por la función dada.

### EJE:

```
let precios = [10, 20, 30, 40, 50];  
console.log(precios.every(function(precio){  
  return precio > 20;  
}));
```

**// Devuelve false, el primer elemento del array "10" no cumple la condición**

# Funciones sobre array

## 8. `sort()`

El método **`sort()`** ordena los elementos de un array localmente y devuelve el array.

**EJE:**

```
let precios = [2, 4, 3, 1, 5];  
console.log(precios.sort((a,b)=>a-b);
```

```
// Devuelve [1,2,3,4,5];
```