



Métodos funcionales en Arrays (Pt. 2)

Introducción a JavaScript
Mentor: Joshua Eduardo González Ruíz

OBJETIVO





Aprender cómo y cuándo utilizar los métodos funcionales de los Arrays y los objetos en **JS**.

Utilizarlos para realizar procesos de forma adecuada en el código.

ÍNDICE

- **Métodos funcionales pt. 2**
 - Crear un nuevo arreglo con los resultados de llamar a una función proporcionada en cada elemento de este arreglo pág. 5
 - Crear un nuevo arreglo con solo elementos que pasan la condición dentro de la función proporcionada pág. 6
 - Aplicar una función contra un acumulador y cada elemento del arreglo para reducirlo a un solo valor pág. 7
- **Introducción a objetos o diccionarios** pág. 8
 - **Objetos** pág. 9
 - **Array de objetos** pág. 12
- **Práctica** pág. 13
- **Referencias** pág. 14

map()

```
1 let kodersNames = ["Francisco", "Jess", "Héctor", "Mónica", "Gustavo",  
2 "Roberto", "Cristian", "Heidel", "Erick", "Aldahir"];  
3  
4 console.log('Se crea un arreglo nuevo en el cual se eliminan los valores de  
5 los elementos que tienen menos de 6 letras: \n')  
6 console.log(kodersNames.map(koder => {  
7   if (koder.length > 5) {  
8     return koder  
9   }  
10 }));
```

Se crea un arreglo nuevo en el cual se eliminan los valores de los elementos enen menos de 6 letras: 🔍 ✕

```
[  
  'Francisco', undefined,  
  'Héctor',    'Mónica',  
  'Gustavo',   'Roberto',  
  'Cristian',  'Heidel',  
  undefined,   'Aldahir'  
]
```

Hint: hit control+c anytime to enter REPL.

>

filter()

```
1 let kodersNames = ["Francisco", "Jess", "Héctor", "Mónica", "Gustavo",  
"Roberto", "Cristian", "Heidel", "Erick", "Aldahir"];  
2  
3  
4 console.log('Se crea un arreglo nuevo en el cual se eliminan los elementos  
que tienen menos de 6 letras: \n')  
5  
6 console.log(kodersNames.filter(koder => {  
7   if (koder.length > 5) {  
8     return koder  
9   }  
10 }));
```

Práctica Clase 9 - Métodos funcionales en Arrays (pt. 2)

Se crea un arreglo nuevo en el cual se eliminan los elementos que tienen menos de 6 letras:

```
[  
  'Francisco', 'Héctor',  
  'Mónica',    'Gustavo',  
  'Roberto',   'Cristian',  
  'Heidel',    'Aldahir'  
]
```

Hint: hit control+c anytime to enter REPL.

➤

reduce()

```
1 let kodersAverage = [80, 100, 90, 50, 100, 60, 75, 85, 80, 50];
2
3 console.log('Se obtiene el promedio general de la cantidad de koders en el
4 arreglo "kodersAverage": \n');
5 let generalAverage = kodersAverage.reduce((sumTotal, koderAverage, index,
6 kodersAverage) => {
7   sumTotal += koderAverage;
8   if (index === kodersAverage.length - 1) {
9     return sumTotal / kodersAverage.length;
10  } else {
11    return sumTotal;
12  }
13 });
14 console.log(generalAverage);
```

Se obtiene el promedio general de la cantidad de koders en el arreglo "kodersAverage":

77

Hint: hit control+c anytime to enter REPL.

✎

Introducción a objetos

OBJETOS

```
1 //Se declara un objeto de las siguientes formas:
2
3 let persona = {};
4
5 //let persona = new Object();
6
7 //Agregamos elementos al objeto "persona"
8
9 persona = { nombre: 'Joshua González', edad: 24, nacionalidad: 'MEX',
10 puesto: 'Sr. Technical Support Engineer', empresa: 'HERE Technologies',
11 añosExperiencia: 7, lenguajes: ['Dart', 'Flutter', 'Java', 'Kotlin',
12 'Objective-C', 'Swift', 'JavaScript'], idiomas: ['Inglés', 'Español'] };
13
14 //Otra forma de agregar elementos al objeto "persona"
15
16 /*persona.nombre = 'Joshua González';
17 persona.edad = 24;
18 persona.nacionalidad = 'MEX';
19 persona.puesto = 'Sr. Technical Support Engineer';
20 persona.empresa = 'HERE Technologies';
21 persona.añosExperiencia = 7;
22 persona.lenguajes = ['Dart', 'Flutter', 'Java', 'Kotlin', 'Objective-C',
23 'Swift', 'JavaScript'];
24 persona.idiomas = ['Inglés', 'Español'];*/
25
26 console.log(persona);
27
28 //Acceder o cambiar el valor de un elemento del objeto "persona" como si
29 fuera un arreglo
30
31 console.log(persona["nombre"]);
32
33 persona["nombre"] = 'Joshua Eduardo González Ruíz';
34
35 console.log(persona.nombre);
```

```
{
  nombre: 'Joshua González',
  edad: 24,
  nacionalidad: 'MEX',
  puesto: 'Sr. Technical Support Engineer',
  empresa: 'HERE Technologies',
  añosExperiencia: 7,
  lenguajes: [
    'Dart',
    'Flutter',
    'Java',
    'Kotlin',
    'Objective-C',
    'Swift',
    'JavaScript'
  ],
  idiomas: [ 'Inglés', 'Español' ]
}
Joshua González
Joshua Eduardo González Ruíz
Hint: hit control+c anytime to enter REPL.
> []
```

OBJETOS

```
32 //Obtenemos las claves de los elementos del objeto "persona"
33
34 console.log(Object.keys(persona));
35
36 //Obtenemos los valores de los elementos del objeto "persona"
37
38 console.log(Object.values(persona));
39
40 //Recorremos todas las claves de los elementos del objeto "persona" con los
    cuales luego podemos acceder a los valores propios de esos elementos:
41
42 for (dato in persona) {
43     console.log(dato);
44 }
```

```
[
  'nombre',
  'edad',
  'nacionalidad',
  'puesto',
  'empresa',
  'aniosExperiencia',
  'lenguajes',
  'idiomas'
]
[
  'Joshua Eduardo González Ruíz',
  24,
  'MEX',
  'Sr. Technical Support Engineer',
  'HERE Technologies',
  7,
  [
    'Dart',
    'Flutter',
    'Java',
    'Kotlin',
    'Objective-C',
    'Swift',
    'JavaScript'
  ],
  [ 'Inglés', 'Español' ]
]
nombre
edad
nacionalidad
puesto
empresa
aniosExperiencia
lenguajes
idiomas
```

OBJETOS

```
32 //Eliminar algún elemento del objeto "persona"
33
34 delete persona.nacionalidad;
35
36 // delete persona["nacionalidad"];
37
38 console.log(persona);
```

```
{
  nombre: 'Joshua Eduardo González Ruíz',
  edad: 24,
  puesto: 'Sr. Technical Support Engineer',
  empresa: 'HERE Technologies',
  añosExperiencia: 7,
  lenguajes: [
    'Dart',
    'Flutter',
    'Java',
    'Kotlin',
    'Objective-C',
    'Swift',
    'JavaScript'
  ],
  idiomas: [ 'Inglés', 'Español' ]
}
```

Hint: hit control+c anytime to enter REPL.

>

ARRAY DE OBJETOS

```
1 //Se declara un arreglo de objetos de la siguiente forma:
2
3 let koders = [
4   {
5     nombre: "Francisco"
6   },
7   {
8     nombre: "Jess"
9   },
10  {
11    nombre: "Héctor"
12  },
13  {
14    nombre: "Mónica"
15  },
16  {
17    nombre: "Gustavo"
18  }
19 ]
20
21 /*
22 koderUno = { nombre: "Francisco" };
23 koderDos = { nombre: "Jess" };
24 koderTres = { nombre: "Héctor" };
25 koderCuatro = { nombre: "Mónica" };
26 koderCinco = { nombre: "Gustavo" };
27
28 koders = [koderUno, koderDos, koderTres, koderCuatro, koderCinco];
29 */
30
31 console.log(koders);
```

```
[
  { nombre: 'Francisco' },
  { nombre: 'Jess' },
  { nombre: 'Héctor' },
  { nombre: 'Mónica' },
  { nombre: 'Gustavo' }
]
Hint: hit control+c anytime to enter REPL.
> 
```

Tiempo de
practicar



Referencias

- **Arrays – Learn web development | MDN. (2021, November 19). MDN Web Docs. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Arrays**