



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

# **Diplomatura Profesional Front End Developer**

## **Módulo 2: Desarrollo Web con JavaScript**

### **JavaScript**

# Métodos de array: map, filter y find

## Array.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.

```
1 const numeros = [1, 3, 8, 20];
2 const dobles = numeros.map(function(num) {
3     return num * 2;
4 });
5
6 console.log(dobles); // [2, 6, 16, 40];
7
8 // versión simplificada usando arrow functions
9 const dobles = numeros.map(num => num * 2)
```

## Array.filter()

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

```
1 const numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
2 const pares = numeros.filter(num => num % 2 === 0);
3 console.log(pares); // [2, 4, 6, 8, 10]
```

## Array.find()

El método **find()** devuelve el valor del primer elemento del array que cumple la función de prueba proporcionada, de lo contrario devuelve undefined.

```
1 const numeros = [3, 5, 10, 15, 28, 76, 149];  
2 const encontrado = numeros.find(num => num > 12);  
3 console.log(encontrado); // 15
```

## Interpolación de strings

---

Las plantillas de cadena de texto (o template strings) son una característica implementada desde ES6 que nos permite, entre otras cosas, interpolar o intercalar variables o expresiones dentro de nuestras cadenas de texto sin tener que recurrir a la concatenación.

Para hacer uso de esta nueva funcionalidad tenemos que rodear nuestros strings con la tilde invertida (`) y las variables o expresiones que deseemos interpolar deberán estar contenidas dentro de la estructura `\${}`.

```
1 var edad1 = 18;  
2 var edad2 = 32;  
3 var mensaje1 = `La cuota para personas de ${edad1} es de  
  ${edad1 >= 21 ? 500 : 350}`;  
4 // La cuota para las personas de 18 es de 350  
5 var mensaje2 = `La cuota para personas de ${edad2} es de  
  ${edad2 >= 21 ? 500 : 350}`;  
6 // La cuota para las personas de 32 es de 500
```

## JSON

---

JSON o JavaScript Object Notation es un formato de texto utilizado para definir estructuras de objetos y listas complejas. Es ampliamente utilizado para el intercambio de datos entre aplicaciones debido a la simplicidad de su implementación y la facilidad de lectura por parte de las aplicaciones y las personas. Es un subconjunto de la notación de objetos literal de Javascript, por lo cual existen múltiples similitudes entre ambos.

## Estructura

## básica

```
1 {  
2   "nombre": "Valeria",  
3   "edad": 28,  
4   "esEstudiante": true,  
5   "hobbies": [  
6     "fútbol",  
7     "programación",  
8     "cine",  
9   ]  
10 }
```

En el ejemplo de estructura podemos apreciar lo siguiente:

- Todo el objeto está rodeado por llaves {}.
- Los nombres de las propiedades están rodeados por comillas dobles “”.
- Los tipos de valores posibles son: texto, número, booleano, array o incluso otros objetos.
- Las listas o arrays se rodean con corchetes [] y pueden contener cualquiera de los tipos de datos aceptados.

En javascript podemos definir lo que se denomina objeto literal usando una sintaxis similar a JSON. Una de las principales diferencias es que no es necesario utilizar comillas dobles en los nombres de propiedades. El ejemplo anterior podría escribirse en javascript de la siguiente manera.

```
1 const persona = {
2   nombre: "Valeria",
3   edad: 28,
4   esEstudiante: true,
5   hobbies: [
6     "fútbol",
7     "programación",
8     "cine",
9   ]
10 };
11
12 console.log(persona.nombre); // "Valeria"
13 console.log(persona.hobbies.length); // 3
```

