
ICARO
Nuestro compromiso es con el futuro.

Introducción a la programación

Métodos comunes en JavaScript

¿Qué son los métodos map, find y filter?

Map, find y filter son métodos que se utilizan en JavaScript para manipular y transformar los elementos dentro de un array.

¿Por qué son útiles?

Los métodos map, find y filter son útiles porque nos permiten trabajar con arrays de manera más eficiente y simplificada, sin tener que usar bucles complejos y lógica adicional.

Ejemplo de cómo se pueden usar en la vida real.

Por ejemplo, podríamos usar estos métodos para trabajar con grandes conjuntos de datos, como una base de datos de usuarios, o para transformar los datos en un formato que sea más fácil de leer o trabajar.

Método map

Ejemplo práctico: Transformación de datos en un array.
El método map nos permite aplicar una función a cada elemento en un array y crear un nuevo array con los resultados. En este ejemplo, estamos usando el método map para crear un nuevo array que tenga cada elemento duplicado.

Aplicación común: Conversión de monedas.

Un ejemplo de uso común del método map es la conversión de monedas. Podríamos tener un array de precios en diferentes monedas y usar el método map para convertir todos los precios a una moneda específica.

```
1  const numeros = [1, 2, 3, 4, 5];
2
3  const duplicados = numeros.map((numero) => numero *
4  2)
5  console.log(duplicados); //[2, 4, 6, 8, 10]
6
```

Método find

Ejemplo práctico: Búsqueda de elementos en un array.

El método find nos permite encontrar el primer elemento en un array que cumpla una cierta condición. En este ejemplo, estamos usando el método find para encontrar el primer número en el array que sea igual a 3.

Aplicación común: Búsqueda de usuarios en una base de datos.

Un ejemplo de uso común del método find es en la búsqueda de usuarios en una base de datos. Podríamos tener un array de usuarios y usar el método find para encontrar el primer usuario que cumpla ciertas condiciones, como su dirección de correo electrónico o su ID de usuario.

```
1  const numeros = [1, 2, 3, 4, 5];
2
3  const numeroEncontrado = numeros.find((numero) => numero ===
4  3);
5  console.log(numeroEncontrado); // 3
```

Método filter

Ejemplo práctico: Selección de elementos que cumplen ciertas condiciones en un array.

El método filter nos permite crear un nuevo array que contenga solo los elementos que cumplan ciertas condiciones. En este ejemplo, estamos usando el método filter para crear un nuevo array que contenga solo los números pares en el array original.

Aplicación común: Filtrado de resultados de una búsqueda.

Un ejemplo de uso común del método filter es en el filtrado de resultados de una búsqueda.

```
1  const numeros = [1, 2, 3, 4, 5];
2
3  const numerosPares = numeros.filter((numero) => numero % 2 === 0);
4
5  console.log(numerosPares); // [2, 4]
```

Método push

Ejemplo práctico: Agregar elementos al final de un array.

El método push nos permite agregar elementos al final de un array existente. En este ejemplo, estamos agregando el elemento "plátano" al final del array "frutas".

Aplicación común: Agregar nuevos elementos a una lista de compras.

```
1  const frutas = ['manzana', 'pera', 'naranja'];
2
3  frutas.push('plátano');
4
5  console.log(frutas); // [ 'manzana', 'pera', 'naranja', 'plátano' ]
```

Método slice

Ejemplo práctico: Crear un sub-array a partir de un array existente.

El método slice nos permite crear un nuevo array a partir de un subconjunto de elementos en un array existente. En este ejemplo, estamos creando un nuevo array que contiene los elementos "pera" y "naranja".

Aplicación común:

- Un usuario podría usar slice() para crear una lista personalizada de tareas a partir de una lista más grande de tareas disponibles.
- Un usuario podría usar slice() para crear una lista de reproducción personalizada a partir de una lista más grande de canciones disponibles.

```
1  const frutas = ['manzana', 'pera', 'naranja', 'plátano'];
2
3  const subArray = frutas.slice(1, 3);
4
5  console.log(subArray); // [ 'pera', 'naranja' ]
```

Muchas gracias!

ICARO Asociación Civil
CUIT 30716564815
info@icaro.org.ar
www.icaro.org.ar