Investigación sobre métodos de Arrays en JS.

- push() y pop(): Se utilizan para agregar o eliminar elementos al final de un array. push() agrega uno o más elementos al final y devuelve la nueva longitud del array. pop() elimina el último elemento y lo devuelve.
- shift() y unshift(): Permiten agregar o eliminar elementos al inicio de un array. shift() elimina el primer elemento y lo devuelve, mientras que unshift() agrega uno o más elementos al inicio y devuelve la nueva longitud.
- concat(): Se utiliza para unir dos o más arrays y devolver un nuevo array que contiene los elementos de los arrays originales. Es útil cuando se necesita combinar arrays sin modificar los originales.
- slice(): Crea una copia superficial de una porción del array en un nuevo array. Se utiliza para extraer una sección de un array sin modificar el original, especificando el índice de inicio y fin (no inclusivo).
- splice(): Este método modifica el propio array agregando o eliminando elementos. Se utiliza para insertar, eliminar o reemplazar elementos en cualquier posición del array.
- indexOf() y lastIndexOf(): Devuelven el índice de la primera o última ocurrencia de un elemento dado en el array, o -1 si no se encuentra. Son útiles para buscar la posición de un elemento específico.
- forEach(): Ejecuta una función provista por el usuario una vez por cada elemento del array. Se utiliza para realizar operaciones en cada elemento sin modificar el array original.
- map(): Crea un nuevo array aplicando una función a cada elemento del array original. Es útil para transformar los elementos de un array de acuerdo con una regla específica.
- filter(): Crea un nuevo array con todos los elementos que pasan la prueba implementada por la función provista. Se utiliza para filtrar elementos basados en un criterio específico.
- reduce(): Aplica una función contra un acumulador y cada valor del array para reducirlo a un único valor. Es útil para realizar operaciones de reducción como sumar, promediar o concatenar elementos de un array.
- every() y some(): Prueban si todos o algunos elementos del array pasan la prueba implementada por la función provista, respectivamente. Son útiles para verificar si todos o algunos elementos cumplen cierta condición.
- find() y findIndex(): Devuelven el valor o el índice del primer elemento que satisface la función de prueba provista. Se utilizan para encontrar el primer elemento que cumpla un criterio específico.
- includes(): Determina si un array incluye un valor dado, devolviendo true o false. Es útil para verificar si un elemento existe en un array.
- flat() y flatMap(): Crean una nueva matriz con todos los elementos de sub-matriz concatenados recursivamente, y flatMap() primero mapea cada elemento usando una función y luego aplana el resultado en una nueva matriz. Se utilizan para trabajar con estructuras de datos anidadas.

Ejemplos de los métodos de JS

- reduce():

```
const numeros = [5, 10, 15, 20, 25];

const suma = numeros.reduce((acumulador, valorActual) => {
  return acumulador + valorActual;
}, 0);

console.log(suma); // Salida: 75
```

- filter:

Se tiene un array con la información de estudiantes y se quiere filtrar a los estudiantes que solo hayan aprobado con una calificación mayor o igual a 60.

```
const estudiantes = [
  { nombre: 'Juan', calificacion: 75 },
  { nombre: 'María', calificacion: 92 },
  { nombre: 'Pedro', calificacion: 58 },
  { nombre: 'Ana', calificacion: 67 },
  { nombre: 'Luis', calificacion: 49 }
];

const estudiantesAprobados = estudiantes.filter(estudiante => {
  return estudiante.calificacion >= 60;
});

console.log(estudiantesAprobados);
```

El output seria:

```
[
  { nombre: 'Juan', calificacion: 75 },
  { nombre: 'María', calificacion: 92 },
  { nombre: 'Ana', calificacion: 67 }
]
```