ARRAYS

Arrays in javascript

An array is a collection of values stored in a single variable.

JavaScript provides multiple methods to manipulate arrays.

1.Removing Duplicate Values from an Array

Problem: Remove duplicate values from an array.

Solution:

```
let arr = [1, 2, 3, 2, 4, 3, 5];
let uniqueArr = [...new Set(arr)];
console.log(uniqueArr); // [1, 2, 3, 4, 5]
```

Explanation:

- Set stores only unique values.
- The spread operator ... converts the Set back into an array.

ARRAYS

Arrays in javascript

2.Finding the Second Largest Number in an Array

Problem: Find the second largest number in an array.

Solution:

```
function secondLargest(arr) {
    let sorted = [...new Set(arr)].sort((a, b) => b - a);
    return sorted.length > 1 ? sorted[1] : null;
}
console.log(secondLargest([10, 20, 5, 30, 30])); // Output: 20
```

Explanation:

- new Set(arr) removes duplicates.
- .sort((a, b) => b - a) sorts the array in descending order.
- The second element is returned if available.

ARRAYS

Arrays in javascript

3.Sorting an Array in Descending Order

Problem: Sort an array in descending order.

Solution:

```
let numbers = [5, 2, 9, 1, 5, 6];
numbers.sort((a, b) => b - a);
console.log(numbers); // [9, 6, 5, 5, 2, 1]
```

Explanation:

- .sort((a, b) => b - a) sorts the array from largest to smallest.

ARRAYS

Arrays in javascript

4.Reversing an Array Without reverse()

Problem: Reverse an array without using .reverse().

Solution:

```
function reverseArray(arr) {
    let reversed = [];
    for (let i = arr.length - 1; i >= 0; i--) {
        reversed.push(arr[i]);
    }
    return reversed;
}
console.log(reverseArray([1, 2, 3, 4])); // [4, 3, 2, 1]
```

Explanation:

- A new array reversed is created.
- The loop starts from the last index and pushes each element into reversed.

ARRAYS

Arrays in javascript

5.Finding the Most Frequent Element in an Array

Problem: Find the most frequent element in an array.

Solution:

```
function mostFrequent(arr) {
    let freqMap = {};
    let maxFreq = 0, mostFrequentNum = null;

    for (let num of arr) {
        freqMap[num] = (freqMap[num] || 0) + 1;
        if (freqMap[num] > maxFreq) {
            maxFreq = freqMap[num];
            mostFrequentNum = num;
        }
    }
    return mostFrequentNum;
}
console.log(mostFrequent([1, 3, 3, 2, 3, 2, 2, 2, 2]));
// Output: 2
```

Explanation:

- A freqMap object stores the frequency of each number.
- The loop updates the highest frequency found and stores the most frequent number.