```javascript
/*

==================================================================

                Arrays Interview Questions 🔥

==================================================================

*/

console.clear();

/*

1. Remove all falsy values (false, 0, "", null, undefined, NaN) from an array

*/

const booleanArray = [0, 1, false, 2, "", 3, null, undefined, NaN, 4, true];

function removeFalsyValues(arr) {

    // filter(Boolean) removes all falsy values by only keeping truthy ones

    return arr.filter(Boolean);
}

const filteredArray = removeFalsyValues(booleanArray);

console.log("The filtered array is:", filteredArray);

/*

2. Reverse an array without using .reverse()

*/

```

```javascript
const originalArray = [1, 2, 3, 4, 5];

const reversedArray = [];

// Loop from the last index and push into a new array

for (let i = originalArray.length - 1; i >= 0; i--) {

    reversedArray.push(originalArray[i]);
}

console.log("The reversed array is:", reversedArray);

/*

3. Clone an array using slice() and Array.from()

*/

const original = [1, 2, 3];

// slice() creates a shallow copy

const clone1 = original.slice();

// Array.from() creates a shallow copy

const clone2 = Array.from(original);

console.log("Cloned Arrays:", clone1, clone2);

/*

4. Merge multiple arrays using concat()

*/

const arr1 = [1, 2];

const arr2 = [3, 4];

// concat merges arrays and returns a new one

const merged = arr1.concat(arr2, [5, 6]);

console.log("Merged Array:", merged);

/*

5. Convert an array to a string using join()

*/

const words = ["Hello", "World"];

// join(" ") combines elements with space between them

console.log("Joined String:", words.join(" "));

/*

6. Count occurrences using reduce()

*/

const votes = ["yes", "no", "yes", "yes", "no"];

// Tally each element's count in an object

const voteCount = votes.reduce((acc, vote) => {

    acc[vote] = (acc[vote] || 0) + 1;

    return acc;

}, {});
```

```javascript

console.log("Vote Count:", voteCount);

/*

7. Remove duplicates manually using indexOf()

*/

const duplicates = [1, 2, 3, 2, 4, 1];

const unique = [];

// Only push if element is not already present

for (let i = 0; i < duplicates.length; i++) {

    if (unique.indexOf(duplicates[i]) === -1) {

        unique.push(duplicates[i]);
    }
}

console.log("Unique Array:", unique);

/*

8. Convert string to array using split()

*/

const sentence = "Split this sentence";

// split(" ") breaks sentence by spaces

console.log("Array from String:", sentence.split(" "));

/*
```

```

9. Flatten a 1-level nested array using concat()

*/

const nested = [1, [2, 3], 4];

// concat with spread to flatten one level

const flattened = [].concat(nested[0], nested[1], nested[2]);

console.log("Flattened Array:", flattened);

/*

10. Replace elements using map()

*/

const nums = [1, 2, 3, 4];

// map returns a new array with modified values

const doubled = nums.map(function (num) {

    return num * 2;
});

console.log("Doubled:", doubled);

/*

11. Check if all elements are even using every()

*/

const evenCheck = [2, 4, 6].every(function (num) {

    return num % 2 === 0;

});

console.log("All Even:", evenCheck);

/*

12. Check if any element is negative using some()

*/

const mixed = [3, -1, 5];

const hasNegative = mixed.some(function (num) {

    return num < 0;

});

console.log("Contains Negative:", hasNegative);

/*

13. Find the max value using reduce()

*/

const values = [5, 2, 8, 1];

// Compare each value and keep the highest

const max = values.reduce(function (a, b) {

    return a > b ? a : b;

});

```

```
console.log("Max Value:", max);

/*

14. Replace or delete elements using splice()

*/

const tools = ["hammer", "wrench", "screwdriver"];

// Replace 'wrench' at index 1 with 'pliers'

tools.splice(1, 1, "pliers");

console.log("After Splice:", tools);

/*

15. Get a portion of an array using slice()

*/

const colors = ["red", "green", "blue", "yellow"];

// Extract elements from index 1 to 3 (excluding 3)

const sliced = colors.slice(1, 3);

console.log("Sliced Part:", sliced);

/*

16. Check if an element exists using includes()

*/

console.log("Has blue?", colors.includes("blue")); // true or false

```

```
/*

17. Get index using indexOf() and lastIndexOf()

*/

// First occurrence

console.log("First index of 'red':", colors.indexOf("red"));

// Last occurrence

console.log("Last index of 'blue':", colors.lastIndexOf("blue"));

/*

18. Iterate using forEach()

*/

colors.forEach(function (color, index) {

    // Access both index and value

    console.log(index + ":", color);
});

/*

19. Sort array of objects by age using sort()

*/

const users = [

    { name: "Heet", age: 22 },

    { name: "Aman", age: 20 },
];

// Sort by age (ascending)

users.sort(function (a, b) {

    return a.age - b.age;
});

console.log("Sorted Users by Age (Ascending):", users);

/*

20. Sum even numbers using filter() and reduce()

*/

const data = [1, 2, 3, 4, 5, 6];

// Filter evens, then sum them

const evenSum = data

    .filter(function (n) {

        return n % 2 === 0;
    })

    .reduce(function (a, b) {

        return a + b;

    }, 0);

console.log("Sum of Evens:", evenSum);

/*

```

```
21. Convert arguments to array using Array.from()

*/

function argsToArray() {

    // Array.from converts arguments object to real array

    const argsArray = Array.from(arguments);

    console.log("Arguments as Array:", argsArray);
}

argsToArray(10, 20, 30);
```