JAVASCRIPT DAY 21th and 22th

CHEAT SHEET OF 21th and 22th

Key Takeaways from Arrays and String Manipulation Topic

1. Arrays Basics

- **Definition**: Arrays are collections of indexed elements that can store multiple values in a single variable.
- **Indexing**: Arrays in JavaScript are zero-indexed (i.e., the first element is at index 0).
- **Data Types**: Arrays can hold elements of any data type (numbers, strings, objects, etc.).

2. Array Methods

- **map()**: Transforms each element of an array using a callback function and returns a new array.
- **filter()**: Filters elements based on a condition, returning a new array of elements that meet the condition.
- **reduce()**: Reduces an array to a single value by iterating over the array and applying a callback function.
- **includes()**: Checks if an array contains a specific element, returning true or false.
- **find()**: Returns the first element in the array that satisfies a condition.
- **every()**: Tests if all elements in an array satisfy a condition.
- **some()**: Tests if at least one element satisfies a condition.
- **splice()**: Adds, removes, or replaces elements in an array.
- **findIndex()**: Returns the index of the first element that satisfies a condition, or -1 if none is found.
- **reverse()**: Reverses the order of elements in an array.
- **slice()**: Returns a shallow copy of a portion of an array.
- **flat()**: Flattens nested arrays into a single-level array.
- **copyWithin()**: Copies a section of an array to another location without changing the array's length.
- **fill()**: Fills all or part of an array with a specific value.

3. String Manipulation Methods

- **split()**: Splits a string into an array based on a specified delimiter.
- **join()**: Joins elements of an array into a single string, with optional delimiters.
- **toString()**: Converts an array into a string with commas separating the elements.
- **substring()**: Extracts a substring between two specified indices.
- **substr()**: Extracts a substring from a string based on a starting index and length.
- **replace()**: Replaces occurrences of a specified pattern in a string.
- **trim()**: Removes whitespace from both ends of a string.
- **padStart()**: Pads the beginning of a string with specified characters to achieve a target length.

- **padEnd()**: Pads the end of a string with specified characters to achieve a target length.

4. Advanced Concepts

- **flatMap()**: Combines map() and flat() into a single operation.
- **reduceRight()**: Similar to reduce(), but processes elements from right to left.
- **lastIndexOf()**: Finds the last occurrence of an element in an array.
- **forEach()**: Iterates over each element in an array and applies a callback function.

5. Practical Applications

- Arrays and string methods are vital in data storage, manipulation, and transformation in JavaScript.
- String methods help in processing and formatting text data.
- Array methods like map(), filter(), and reduce() simplify data handling and avoid manual loops.

6. Real-World Uses

- **E-commerce**: Manipulating shopping cart items, updating quantities, or calculating totals.
- **Data Processing**: Transforming and filtering datasets for reports, charts, or user interfaces.
- **API/Database**: Formatting and filtering data for APIs or databases.
- **User Input**: Enhancing user input for better formatting and validation.

7. Best Practices

- **Method Chaining**: Use method chaining (e.g., map().filter().reduce()) for cleaner, more concise code.
- **Array Methods**: Prefer higher-order methods like map(), filter(), and reduce() over manual loops for readability and efficiency.
- **Handling Nested Arrays**: Use flat() and flatMap() to handle nested arrays effectively.
- **String Optimization**: Optimize string operations using methods like trim(), padStart(), and padEnd() for better performance.

CODING QUESTIONS OF 21th and 22th Day

Basic Array Questions

1) Declare an array of 5 integers and display its elements using document.write:

```
let arr = [1, 2, 3, 4, 5];
document.write(arr);
```

2) Create an array of 5 strings and print each string on a new line using a loop:

```
let arr = ["apple", "banana", "cherry", "date", "elderberry"];
for (let i = 0; i < arr.length; i++) {
```

```
    document.write(arr[i] + "<br>");
}
```

3) Access the first and last elements of an array using the at() method:

```
let arr = [10, 20, 30, 40, 50];
document.write("First element: " + arr.at(0) + "<br>");
document.write("Last element: " + arr.at(-1));
```

4) an array of integers and find the sum of its elements using a loop:

```
let arr = [1, 2, 3, 4, 5];
let sum = 0;
for (let i = 0; i < arr.length; i++) {
    sum += arr[i];
}
document.write("Sum: " + sum);
```

5) Reverse the elements of an array using the reverse() method:

```
let arr = [1, 2, 3, 4, 5];
arr.reverse();
document.write(arr);
```

Array Manipulation Questions

6) Use the map() method to multiply all elements in an array by 10:

```
let arr = [1, 2, 3, 4, 5];
let result = arr.map(x => x * 10);
document.write(result);
```

7) Filter out all elements less than 50 from an array using the filter() method:

```
let arr = [10, 20, 30, 60, 70];
let result = arr.filter(x => x >= 50);
document.write(result);
```

8) Use the reduce() method to calculate the product of all elements in an array:

```
let arr = [1, 2, 3, 4];
let product = arr.reduce((acc, val) => acc * val, 1);
document.write(product);
```

9) Remove duplicates from an array using filter() and indexOf():

```
let arr = [1, 2, 3, 2, 4, 1];
```

```
let result = arr.filter((value, index, self) => self.indexOf(value) === index);
document.write(result);
```

10) Create an array of strings and check if a particular string exists using the includes() method:

```
let arr = ["apple", "banana", "cherry"];
let exists = arr.includes("banana");
document.write(exists);  // true
```

Advanced Array Manipulations

11) Find the index of a specific element in an array using findIndex():

```
let arr = [10, 20, 30, 40];
let index = arr.findIndex(x => x === 30);
document.write(index);  // 2
```

12) Use the every() method to check if all elements in an array are greater than 10:

```
let arr = [12, 15, 18, 20];
let result = arr.every(x => x > 10);
document.write(result);  // true
```

13) Check if any element in an array is divisible by 5 using the some() method:

```
let arr = [2, 3, 4, 5, 6];
let result = arr.some(x => x % 5 === 0);
document.write(result);  // true
```

14) Use the splice() method to remove elements from index 2 to 4 in an array:

```
let arr = [10, 20, 30, 40, 50];
arr.splice(2, 3);
document.write(arr);  // [10, 20]
```

15) Add new elements at the beginning and end of an array using unshift() and push():

```
let arr = [10, 20, 30];
arr.unshift(5);  // Add to the beginning
arr.push(40);    // Add to the end
document.write(arr);  // [5, 10, 20, 30, 40]
```

String Manipulation Questions

16) Convert a string to an array using Array.from() and display the result:

```
let str = "hello";
```

```
let arr = Array.from(str);
document.write(arr);  // ['h', 'e', 'l', 'l', 'o']
```

17) Reverse a string and convert it back to a string using split(), reverse(), and join():

```
let str = "hello";
let reversedStr = str.split("").reverse().join("");
document.write(reversedStr);  // "olleh"
```

18) Replace all occurrences of a word in a string with another word using the replace() method:

```
let str = "I love apples";
let newStr = str.replace(/apples/g, "oranges");
document.write(newStr);  // "I love oranges"
```

19) Extract a substring from a string using the substring() method:

```
let str = "Hello, world!";
let subStr = str.substring(0, 5);
document.write(subStr);  // "Hello"
```

20) Extract a portion of a string using the substr() method:

```
let str = "Hello, world!";
let subStr = str.substr(7, 5);
document.write(subStr);  // "world"
```

Combined Array and String Manipulations

21) Convert an array of numbers to a comma-separated string using toString():

```
let arr = [1, 2, 3, 4, 5];
let str = arr.toString();
document.write(str);  // "1,2,3,4,5"
```

22) Concatenate all elements of an array into a single string using the join() method:

```
let arr = ["apple", "banana", "cherry"];
let str = arr.join(", ");
document.write(str);  // "apple, banana, cherry"
```

23) Split a sentence into words and store them in an array using the split() method:

```
let str = "Hello world how are you";
let words = str.split(" ");
document.write(words);  // ["Hello", "world", "how", "are", "you"]
```

24) Flatten a multi-dimensional array to a single-dimensional array using the flat() method:

```
let arr = [1, [2, 3], [4, [5, 6]]];
let flattenedArr = arr.flat(2);
document.write(flattenedArr);  // [1, 2, 3, 4, 5, 6]
```

25) Use flatMap() to combine two arrays of related elements:

```
let arr1 = [1, 2, 3];
let arr2 = [4, 5, 6];
let combined = arr1.flatMap((x, index) => [x, arr2[index]]);
document.write(combined);  // [1, 4, 2, 5, 3, 6]
```

Practical Applications

26) Find the largest and smallest elements in an array:

```
let arr = [10, 20, 30, 40, 50];
let largest = Math.max(...arr);
let smallest = Math.min(...arr);
document.write("Largest: " + largest + ", Smallest: " + smallest);
```

27) Count the number of occurrences of a specific element in an array:

```
let arr = [1, 2, 2, 3, 2, 4];
let count = arr.filter(x => x === 2).length;
document.write(count);  // 3
```

28) Use copyWithin() to copy a portion of an array into another position in the same array:

```
let arr = [1, 2, 3, 4, 5];
arr.copyWithin(0, 3, 5);
document.write(arr);  // [4, 5, 3, 4, 5]
```

29) Find the last occurrence of a specific value in an array using lastIndexOf():

```
let arr = [1, 2, 3, 2, 4];
let index = arr.lastIndexOf(2);
document.write(index);  // 3
```

30) Check if an array is sorted in ascending order using the sort() method:

```
let arr = [1, 2, 3, 4, 5];
let sorted = arr.slice().sort((a, b) => a - b);
let isSorted = arr.toString() === sorted.toString();
document.write(isSorted);  // true
```

Real-World Scenarios

31) Create an array of product prices and apply a discount using the map() method:

```
let prices = [100, 200, 300];
let discountedPrices = prices.map(price => price * 0.9);
document.write(discountedPrices);
```

32) Filter products with prices greater than a certain amount using the filter() method:

```
let prices = [100, 200, 300, 50];
let result = prices.filter(price => price > 150);
document.write(result);  // [200, 300]
```

33) Calculate the total cost of items in a shopping cart using the reduce() method:

```
let prices = [100, 200, 150];
let totalCost = prices.reduce((acc, price) => acc + price, 0);
document.write(totalCost);  // 450
```

34) Display the list of unique categories from an array of product objects:

```
let products = [
   { name: "Phone", category: "Electronics" },
   { name: "Shirt", category: "Clothing" },
   { name: "Laptop", category: "Electronics" },
];
let categories = [...new Set(products.map(product => product.category))];
document.write(categories);  // ["Electronics", "Clothing"]
```

35) Check if all students in a class scored above 40 using the every() method:

```
let scores = [45, 50, 60, 30];
let result = scores.every(score => score > 40);
document.write(result);  // false
```

Interactive Challenges

36) Write a program to accept an array from the user and reverse its elements:

```
let arr = prompt("Enter elements of an array (comma separated)").split(",");
arr.reverse();
document.write(arr);
```

37) Find the index of a user-specified value in an array using indexOf():

```
let arr = [10, 20, 30, 40, 50];
```

```
let value = prompt("Enter a value to find");
let index = arr.indexOf(Number(value));
document.write(index);  // Index of the specified value
```

38) Trim a string using trim() after checking if it starts and ends with whitespace:

```
let str = prompt("Enter a string");
let trimmedStr = str.trim();
document.write(trimmedStr);
```

39) Create a string padding program using padStart() and padEnd() methods:

```
let str = "5";
let paddedStr = str.padStart(3, "0").padEnd(5, "0");
document.write(paddedStr);  // "00500"
```

40) Ask the user to enter a sentence and replace all occurrences of a specific word:

```
let str = prompt("Enter a sentence");
let newStr = str.replace(/word/g, "replacement");
document.write(newStr);
```

Algorithmic Questions

41) Merge two arrays into one without duplicates:

```
let arr1 = [1, 2, 3];
let arr2 = [3, 4, 5];
let merged = [...new Set([...arr1, ...arr2])];
document.write(merged);
```

42) Rotate an array to the left by 2 positions using a loop:

```
let arr = [1, 2, 3, 4, 5];
let rotated = [...arr.slice(2), ...arr.slice(0, 2)];
document.write(rotated);  // [3, 4, 5, 1, 2]
```

43) Calculate the average of an array's elements:

```
let arr = [1, 2, 3, 4, 5];
let avg = arr.reduce((acc, val) => acc + val, 0) / arr.length;
document.write(avg);  // 3
```

44) Sort an array of strings alphabetically and display the sorted array:

```
let arr = ["banana", "apple", "cherry"];
arr.sort();
```

```
document.write(arr);  // ["apple", "banana", "cherry"]
```

45) Count the number of vowels in a string:

```
let str = "Hello world";
let count = (str.match(/[aeiou]/gi) || []).length;
document.write(count);  // 3
```

Custom Challenges

46) Flatten a 2D array into a 1D array using flat():

```
let arr = [[1, 2], [3, 4], [5, 6]];
let flattenedArr = arr.flat();
document.write(flattenedArr);  // [1, 2, 3, 4, 5, 6]
```

47) Implement a function to compare two arrays for equality:

```
function arraysEqual(arr1, arr2) {
   return arr1.length === arr2.length && arr1.every((value, index) => value === arr2[index]);
}
let result = arraysEqual([1, 2, 3], [1, 2, 3]);
document.write(result);  // true
```

48) Use reduceRight() to concatenate an array of strings in reverse order:

```
let arr = ["one", "two", "three"];
let result = arr.reduceRight((acc, curr) => acc + " " + curr);
document.write(result);  // "three two one"
```

49) Remove specific elements from an array based on user input:

```
let arr = [1, 2, 3, 4, 5];
let element = prompt("Enter element to remove");
arr = arr.filter(x => x != element);
document.write(arr);
```

50) Implement a custom function using forEach() to display each element of an array along with its index:

```
let arr = ["apple", "banana", "cherry"];
arr.forEach((value, index) => {
   document.write(`Index: ${index}, Value: ${value}<br>`);
});
```

FAQ'S OF DAY 21th AND 22th

General Questions

1. **What is an array in JavaScript?** An array is a data structure that stores multiple values in a single variable, with each value accessible via its index.
2. **How are arrays indexed in JavaScript?** Arrays are zero-indexed, meaning the first element has an index of 0, the second has an index of 1, and so on.
3. **What is the difference between an array and an object?** Arrays are ordered collections of elements, while objects are unordered collections of key-value pairs.
4. **Can arrays hold elements of different data types?** Yes, arrays in JavaScript can hold elements of different data types, such as numbers, strings, and objects.

Array Methods

5. **What is the purpose of the map() method?** The map() method creates a new array by applying a given function to each element of the original array.
6. **How does the filter() method work?** The filter() method creates a new array containing only the elements that satisfy a specified condition.
7. **What does the reduce() method do?** The reduce() method reduces an array to a single value by applying a callback function repeatedly to each element.
8. **What is the difference between splice() and slice()?**
   - **splice():** Modifies the original array by adding, removing, or replacing elements.
   - **slice():** Returns a shallow copy of a portion of an array without modifying the original array.
9. **What does the flat() method do?** The flat() method flattens nested arrays into a single array up to the specified depth.
10. **How is includes() different from some()?**

- **includes():** Checks if a specific value exists in the array.
- **some():** Tests if at least one element passes a provided condition.

String Manipulations

11. **What is the difference between substring() and substr()?**

- **substring():** Extracts a portion of a string using start and end indices.
- **substr():** Extracts a portion of a string using a start index and length.

12. **How does the split() method work?** The split() method splits a string into an array of substrings based on a specified delimiter.

13. **What does the trim() method do?** The trim() method removes whitespace from both ends of a string.
14. **How can you pad a string in JavaScript?** You can use **padStart()** or **padEnd()** to add characters to the beginning or end of a string to reach a specified length.
15. **What is the purpose of the replace() method?** The replace() method replaces a specified substring or pattern with another string.

Practical Scenarios

16. **How do you remove duplicates from an array?** You can use **filter()** with **indexOf()** or convert the array to a **Set** and back to an array using **Array.from()**.
17. **How can you check if all elements in an array satisfy a condition?** Use the **every()** method to test if all elements meet the specified condition.
18. **How do you find the index of the first matching element in an array?** Use the **findIndex()** method with a callback function that specifies the condition.
19. **What method would you use to reverse a string?** Convert the string to an array using **split()**, reverse the array using **reverse()**, and convert it back to a string using **join()**.
20. **How can you combine two arrays into one?** Use the **concat()** method or the spread operator (**...**).

Best Practices

21. **What is the most efficient way to handle nested arrays?** Use the **flat()** or **flatMap()** methods for flattening arrays.
22. **Why is chaining methods useful in JavaScript?** Chaining methods (e.g., **map().filter().reduce()**) makes code more concise and easier to read.
23. **How can you handle large strings efficiently?** Use string methods like **substring()** or **slice()** to process parts of the string instead of manipulating it directly.
24. **What is the difference between pop() and shift()?**

- **pop():** Removes the last element from an array.
- **shift():** Removes the first element from an array.

25. **When should you use reduce() over map()?** Use **reduce()** when you need to aggregate or summarize array elements into a single value, and use **map()** for element-wise transformations.

<u>MCQ'S OF DAY 21th and 22th</u>

1. **What does an array in JavaScript represent?**

   B) A collection of indexed elements

2. **How are arrays indexed in JavaScript?**

   B) Starting at 0

3. **Which symbol is used to define an array in JavaScript?**

   C) [ ]

4. **What is the output of console.log([10, 20, 30][1])?**

   B) 20

5. **How can you find the length of an array arr?**

   C) arr.length

6. **Which method is used to reverse an array in place?**

   A) reverse()

7. **What is the output of console.log(typeof [1, 2, 3])?**

   A) object

8. **Which method would you use to add elements to the end of an array?**

   A) push()

9. **What will arr[5] return if the array has only 3 elements?**

   A) undefined

10. **What is the default separator when using join() on an array?**

    A) Comma

11. **How can you add elements to the beginning of an array?**

    B) unshift()

12. **Which array method removes the last element?**

    B) pop()

13. **How can you check if an array includes a specific value?**

    C) includes()

14. **What is the index of the last element in an array arr of size 5?**

    A) 4

15. **Which method is used to sort elements in an array?**

    A) sort()

16. **Which method creates a shallow copy of an array?**

    A) slice()

17. **What is the output of [1, 2, 3].map(x => x * 2)?**

    A) [2, 4, 6]

18. **What does filter() return if no elements match the condition?**

    C) An empty array

19. **Which method combines multiple arrays into one?**

    A) concat()

20. **What does reduce() do in an array?**

    B) Combines elements into a single value

21. **Which method would you use to find the index of a specific element in an array?**

    B) findIndex()

22. **What is the purpose of the flat() method?**

    B) To flatten nested arrays

23. **What does the every() method return for an empty array?**

    A) true

24. **Which method adds elements to the middle of an array?**

    A) splice()

25. **What will arr.copyWithin(2, 0) do?**

    B) Copy elements from index 0 to the position starting at index 2

26. **What does the split() method do?**

    B) Splits a string into an array of substrings

27. **Which method removes whitespaces from both ends of a string?**

    A) trim()

28. **What will "Hello".padStart(8, "*") return?**

    B) ***Hello

29. **Which method is used to replace all occurrences of a substring in JavaScript?**

    B) replaceAll()

30. **What will "Welcome".substring(3) return?**

    C) come

31. **What is the difference between substr() and substring()?**

    B) substr() takes a start and length; substring() takes start and end indices

32. **How do you reverse a string in JavaScript?**

    B) split('').reverse().join('')

33. **Which method converts an array to a comma-separated string?**

    B) join()

34. **What will "ExcelR".repeat(3) return?**

    A) ExcelExcelExcel

35. **What does the toUpperCase() method do?**

    A) Converts all characters to uppercase

36. **Which array method removes duplicates most effectively?**

    C) Set() with Array.from()

37. **Which method combines mapping and flattening?**

    C) flatMap()

38. **What is the output of [1, 2, 3].reduceRight((a, b) => a + b)?**

    B) 9

39. **How does some() differ from every()?**

    B) some() returns true if at least one element passes a test; every() returns true only if all elements pass the test

40. **Which method would you use to remove a specific element without leaving gaps in the array?**

    A) splice()