

1. Create a function that returns the last element of an array.
2. Find the combination of two arrays.
3. Generate a random integer between 0 to 18.
4. Create a function that takes an array containing both numbers and strings, and returns a new array containing only the string values.
5. Find the maximum number in an array.
6. Write a function that returns the length of a given object (number of keys).
7. In an array of objects filter out those objects which have gender's value male.

8. Given an array of strings, return a new array where all strings are in uppercase.
9. Check if an object is empty (has no keys).
10. Create an array of numbers and double each value using `.map()`.
11. Convert an array of strings into a single comma-separated string.
12. Write a function to flatten a nested array (one level deep).(e.g., `[1, [2, 3], 4] → [1, 2, 3, 4]`)
13. Write a function that checks if all elements in an array are numbers.
14. Build a simple `isPrime()` function to check if a number is prime.
15. Create a function that removes duplicate values from an array.

16. What's the difference between `parseInt` and `Number()`?
17. Why does `0.1 + 0.2 !== 0.3` in JavaScript?
18. Explain floating-point precision issues in JavaScript.
19. How would you handle high-precision decimal math in JS?
20. What is the difference between `slice` and `splice`?
21. Create a function that reverse each word of a given sentence. E.g., Mai hun manas → sanam nuh iam
22. In an array of numbers and strings, only add those numbers which are not strings.
23. How would you check if a number is an integer?

24. Write a JavaScript function that reverse a number.
25. Write a javascript function that returns a passed string with letters in alphabetical order.
26. Write a Javascript function that accepts a string as a parameter and conversts the first letter of each word of the string in upper case.
27. Write a javascript function to get the first element of an array. Passing a parameter 'n' will return the first 'n' elements of the array.
28. Write a javascript function to get the number of occurrences of each letter in specified string.
29. write a javascript program to find the most frequent item of an array.
30. write a javascript program to shuffle an array.

31. How can you update the DOM based on user input in real-time (e.g., live preview of a form)?
32. How would you handle form validation in real time and show error messages dynamically?
33. How do you find the closest ancestor element that matches a selector?
34. How would you toggle a class on an element when another element is clicked (e.g., show/hide sidebar)?
35. Create a Show/Hide Password Toggle
36. Create a Sticky Header on Scroll. The header becomes fixed to the top after scrolling down. (Key concepts: scroll event, window.scrollY, classList.add/remove)

37. What is a pure function, and why is it useful in UI rendering?
38. How would you use `.map()` to transform a list of products into a list of HTML elements?
39. How do you use `.reduce()` to calculate the total price in a shopping cart?
40. Explain immutability and how you would update an object in an array without mutating the original.
41. How would you compose multiple functions to transform data step-by-step (e.g., `sanitize` → `trim` → `capitalize`)? Scenario: You're preparing user input before storing it. (Expected knowledge: Function composition, chaining, pipe or compose logic.)
42. What is the difference between `forEach` and `map`, and when is it wrong to use `map`?
43. How do you implement your own version of `.map()` function on arrays?

44. Guess the Output:

```
let date = new Date(0);  
console.log(date.toString());
```

45. Validate that a user's selected date range is no longer than 30 days.

46. Calculate difference between two dates in the format of “\_ years \_ months \_ days \_ hours \_ minutes \_ sec”.

47. Add or subtract n days from a given date. (E.g., Add 7 days to "2025-05-01")

48. Calculate the user's age from their date of birth.

49. Write a formatDate(dateStr) function that returns a user-friendly format like Jan 1, 2025. Why is it better to centralize this logic in a utility?

50. What are the different options and parameters available in toLocaleString and toLocaleDateString etc to format date.

# Locale & Options

```
const date = new Date();

date.toLocaleString(locale, options) // Full date + time
date.toLocaleDateString(locale, options) // Date only
date.toLocaleTimeString(locale, options) // Time only
// locale --> en-IN, en-GB, fr-Fr, hi-En
```

Option	Possible Values
weekday	"long", "short", "narrow"
year	"numeric", "2-digit"
month	"numeric", "2-digit", "long", "short"
day	"numeric", "2-digit"
hour	"numeric", "2-digit"
minute	"numeric", "2-digit"
second	"numeric", "2-digit"
hour12	true or false
timeZone	"UTC", "Asia/Kolkata", etc.
timeZoneName	"short", "long"