

1. JavaScript
- 2.
3. Prototype pollution
4. Weak Set
5. Proxy object
6. Null vs undefined
7. Memory allocation for undefined
8. Symbol
9. Class: adding methods, child class constructor
10. Remove key from object
11. Loop through an object
12. for...in vs for...of
13. Iterator, enumerator
14. Execution order when timers and callbacks are involved
15. Ternary syntax, ternary with return
16. .reduce(): longest string in array
17. Generator: yield multiples of a given number
18. Currying
19. Closure
20. Temporal Dead Zone
21. Object.seal

22. Debouncing
23. Class syntax
24. Object syntax
25. Generate array containing multiples of 7 between 1 and 100
26. Function to return average of two args
27. Swap values of two variables
28. TDZ (Temporal Dead Zone)
29. Pass by reference
30. Increment loop iterator by any number
31. Difference between  $(e) \Rightarrow \{e\}$  and  $(e) \Rightarrow (e)$
32. `const val = "apple"; const x = {val}` - what is inside x?
33. Destructuring syntax
34. Execution order when timers and callbacks are involved (referenced link)
35. Pass by reference
36. Execution context
37. Primitive Datatype vs Non-Primitive
38. String Coercion
39. Examples of higher-order functions
40. Factory Functions
41. Check a function whether palindrome or

not

42. IIFE (Immediately Invoked Function Expression)
43. Benefits of IIFE
44. Prototypical inheritance
45. DOM manipulation
46. Object empty
47. Array length
48. Remove nth element from array
49. Variadic function to return sum and average of args
50. Switch case, fall-through
51. Generator function to yield even numbers endlessly
52. Primitive vs non-primitive data types - understand better