

Javascript Interview Questions

1. What is the difference between null and undefined ?
2. What is NaN?
3. How do you access an element in the DOM using JavaScript?
4. How would you create a new HTML element using JavaScript?
5. What is the purpose of the `querySelectorAll` method?
6. Explain the concept of event bubbling and event capturing?
7. What is event propagation in the DOM?
8. How do you enable event capturing in JavaScript?
9. What is the default phase of event propagation?
10. How can you stop the propagation of an event?
11. What is the difference between localStorage and sessionStorage?
12. Can you store non-string data types in Web Storage?
13. How can you remove an item from Web Storage?
14. What happens to Web Storage data when the user clears their browser cache?
15. Explain the difference between `==` and `===` in Js.
16. What is the purpose of the `typeof` operator in Js.
17. Explain the differences between `var`, `let`, and `const`.
18. What is the difference between a function declaration and a function expression?
19. Explain the concept of callbacks.
20. What is the difference between `call` and `apply` and `bind` in JavaScript?
21. How does the `for...in` loop differ from the `for...of` loop?
22. How do you create an object in JavaScript?
23. Explain the concept of object destructuring in JavaScript.
24. What is prototypal inheritance in JavaScript?
25. How do you check if a property exists in an object?
26. Explain the difference between `Object.keys()`, `Object.values()`.
27. How do you clone an object in JavaScript?

28. What is the purpose of the `"this"` keyword in the context of an object?
29. How do you create a read-only property in an object in javascript?
30. What is `Object.create` used for in JavaScript?
31. Explain the difference between `slice()` and `splice()` methods.
32. How can you iterate over elements in an array?
33. How can we check whether a given variable is an array?
34. What is the difference between `forEach()` and `map()` methods?
35. What is the purpose of the `find()` method?
36. What is the `"this"` keyword in JavaScript?
37. How does arrow function differ in handling `this` compared to regular functions?
38. What is function borrowing?
39. Explain template literals in ES6.
40. How do you handle default values for function parameters in ES6?
41. What is the difference between `__proto__` and `prototype` in JavaScript?
42. What is a closure in JavaScript?
43. How can closures be used to create private variables in JavaScript?
44. Real time coding example around closure? Example: Create a unique handler function to update the styles of elements in DOM.
45. How can we implement a private counter using closure?
46. How does the `setTimeout` function utilize closures?
47. Explain the concept of "function currying" and how closures are involved.
48. What is a constructor function?
49. What is the purpose of the `prototype` property in a constructor function?
50. What is an IIFE?
51. Why might you use an IIFE in JavaScript?
52. How do you set the prototype of an object directly in JavaScript?
53. How can you create a closure inside a loop when using `setTimeout`?
54. Explain the difference between global execution context and function execution context.
55. How does JavaScript handle asynchronous operations in the execution context?

56. What is a call stack and how does it work?
57. How does the call stack work with asynchronous operations?
58. Explain the concept of a callback hell. How can it be mitigated?
59. What is a Promise in JavaScript? Explain the different states of a Promise.
60. How do you create a Promise in JavaScript?
61. What is the purpose of the `Promise.all` method?
62. Describe the difference between `Promise.resolve` and `Promise.reject`.
63. Explain the purpose of the `finally` method in a Promise.
64. What is the purpose of `async` and `await` in JavaScript?
65. Explain the concept of the event loop and how `async/await` interacts with it.