Beginner Level (0-100 Questions)

1. What is JavaScript?
2. How do you include JavaScript in an HTML file?
3. What are the different data types in JavaScript?
4. How do you declare a variable in JavaScript?
5. What is the difference between `var`, `let`, and `const`?
6. How do you create a string in JavaScript?
7. What is the difference between == and ===?
8. How do you write a comment in JavaScript?
9. How do you create an array in JavaScript?
10. How do you access an element in an array?
11. How do you add an element to an array?
12. How do you remove the last element from an array?
13. What is a function in JavaScript?
14. How do you call a function in JavaScript?
15. What is an object in JavaScript?
16. How do you access a property in an object?
17. How do you add a property to an object?
18. What is the purpose of `return` in a function?
19. What is the difference between `null` and `undefined`?
20. How do you create a conditional statement in JavaScript?
21. How do you create a loop in JavaScript?
22. What is the difference between `for`, `while`, and `do...while` loops?
23. How do you break out of a loop?
24. How do you continue to the next iteration in a loop?
25. What is an event in JavaScript?
26. How do you attach an event handler to an HTML element?
27. What is `this` keyword in JavaScript?
28. How do you use the `Date` object in JavaScript?
29. How do you generate a random number in JavaScript?
30. What is a callback function?
31. How do you handle errors in JavaScript?
32. What are template literals?
33. How do you use the `map` function in JavaScript?
34. What is the difference between `map` and `forEach`?
35. How do you use the `filter` function in JavaScript?
36. What is the difference between `filter` and `map`?
37. How do you use the `reduce` function in JavaScript?
38. What is `typeof` operator?
39. How do you convert a string to a number in JavaScript?
40. How do you convert a number to a string in JavaScript?
41. What is `NaN`?
42. How do you check if a variable is `NaN`?

43. What is `Infinity` in JavaScript?
44. What is a `Promise` in JavaScript?
45. How do you create a `Promise` in JavaScript?
46. What is `async` and `await` in JavaScript?
47. How do you use `JSON` in JavaScript?
48. What is the `JSON.stringify` method?
49. What is the `JSON.parse` method?
50. How do you use `localStorage` in JavaScript?
51. What is the difference between `localStorage` and `sessionStorage`?
52. How do you check the length of a string in JavaScript?
53. How do you split a string in JavaScript?
54. How do you join an array into a string in JavaScript?
55. What is `Math` object in JavaScript?
56. How do you round a number in JavaScript?
57. How do you find the maximum and minimum values in an array?
58. How do you sort an array in JavaScript?
59. How do you reverse an array in JavaScript?
60. How do you find an element in an array?
61. What is `for...in` loop?
62. What is `for...of` loop?
63. What is an anonymous function in JavaScript?
64. How do you create a closure in JavaScript?
65. What is `setTimeout` function?
66. How do you use `setInterval` function?
67. How do you clear a timeout in JavaScript?
68. What is the `typeof` operator?
69. How do you check if a variable is an array?
70. What is the `instanceof` operator?
71. How do you use `try...catch` in JavaScript?
72. What is `strict mode` in JavaScript?
73. How do you use `eval` in JavaScript?
74. What is the `bind` method in JavaScript?
75. How do you use `call` and `apply` methods?
76. What is an arrow function?
77. How do you use `default parameters` in JavaScript?
78. What is destructuring in JavaScript?
79. How do you use `spread` operator in JavaScript?
80. How do you use `rest` operator in JavaScript?
81. What is `document` object in JavaScript?
82. How do you get an element by ID in JavaScript?
83. How do you get elements by class name?
84. How do you get elements by tag name?
85. How do you query an element using a selector?

86. What is `innerHTML`?
87. How do you change the text content of an element?
88. How do you add a class to an element?
89. How do you remove a class from an element?
90. What is event bubbling?
91. What is event capturing?
92. How do you stop event propagation?
93. What is `preventDefault` method?
94. How do you clone an object in JavaScript?
95. What is the difference between deep copy and shallow copy?
96. How do you merge two objects?
97. How do you merge two arrays?
98. What is `RegExp` in JavaScript?
99. How do you create a regular expression in JavaScript?
00. How do you test a regular expression in JavaScript?

Intermediate Level (100 Questions)

1. What is hoisting in JavaScript?
2. What are closures in JavaScript?
3. What is the difference between `var`, `let`, and `const` in terms of scope and hoisting?
4. How does the JavaScript engine execute the code?
5. What are IIFE (Immediately Invoked Function Expressions)?
6. How do you handle asynchronous operations in JavaScript?
7. What is event delegation?
8. What are arrow functions, and how do they differ from regular functions?
9. How does the `this` keyword behave in arrow functions?
10. What is a `callback hell` and how do you avoid it?
11. How do you handle errors in promises?
12. What is the event loop in JavaScript?
13. What are microtasks and macrotasks in JavaScript?
14. How does `async` and `await` work in JavaScript?
15. How do you create a custom error in JavaScript?
16. What is `debouncing` and `throttling` in JavaScript?
17. How do you implement debouncing in JavaScript?
18. How do you implement throttling in JavaScript?
19. What is a `prototype` in JavaScript?
20. How do you create a prototype chain?
21. What is the difference between `call`, `apply`, and `bind`?
22. What is the difference between function declaration and function expression?
23. How do you use `Object.create` method?
24. What are ES6 classes in JavaScript?
25. How do you use `extends` and `super` keywords in JavaScript?
26. How do you create a singleton in JavaScript?
27. What is `typeof` vs `instanceof`?

28. What is the `new` keyword in JavaScript?
29. What is the difference between `Object.keys`, `Object.values`, and `Object.entries`?
30. How do you freeze an object in JavaScript?
31. What is a mixin in JavaScript?
32. How do you create a module in JavaScript?
33. What are `import` and `export` in JavaScript?
34. How do you use dynamic imports in JavaScript?
35. What is the `fetch` API?
36. How do you handle HTTP requests using `fetch`?
37. What are cookies in JavaScript?
38. How do you set and get cookies in JavaScript?
39. What is the `document.cookie` API?
40. What is `CORS`, and how do you handle it in JavaScript?
41. How do you use `async` and `await` with `fetch`?
42. What are `Generators` in JavaScript?
43. How do you create and use a generator function?
44. What is the purpose of the `yield` keyword?
45. What is `Symbol` in JavaScript?
46. How do you use symbols to create private properties?
47. What are iterators in JavaScript?
48. How do you create a custom iterator?
49. What is a `Proxy` object in JavaScript?
50. How do you use `Reflect` in JavaScript?
51. What are template literals and tagged templates?
52. What is `Map` and `Set` in JavaScript?
53. How do you use `WeakMap` and `WeakSet`?
54. What is a `Promise.all` and how does it work?
55. How do you use `Promise.race`?
56. What are `Web Workers` in JavaScript?
57. How do you create and use a Web Worker?
58. How does message passing work with Web Workers?
59. What is the difference between synchronous and asynchronous code?
60. How do you use `Promise.allSettled`?
61. What is memoization in JavaScript?
62. How do you implement memoization in JavaScript?
63. What is a `Pure` function?
64. How do you handle deep cloning of objects?
65. What is functional programming in JavaScript?
66. How do you use `currying` in JavaScript?
67. What is the `bind` method used for?
68. How does the `apply` method differ from `call`?
69. What is `JSONP`, and how does it differ from `fetch`?
70. How do you work with binary data in JavaScript?

71. What is `ArrayBuffer` and `TypedArray` in JavaScript?
72. How do you use `DataView` in JavaScript?
73. What is the difference between `window` and `document` objects?
74. How do you use the `MutationObserver` API?
75. What are `WebSockets` in JavaScript?
76. How do you create a WebSocket connection?
77. How do you handle binary data with WebSockets?
78. What is `Service Worker` in JavaScript?
79. How do you register a service worker?
80. How do you use the `Cache` API with service workers?
81. What is `IndexedDB`, and how do you use it?
82. What are promises vs observables?
83. How do you handle cross-browser compatibility in JavaScript?
84. What is a polyfill in JavaScript?
85. How do you use Babel for transpiling JavaScript code?
86. What is the difference between `map` and `WeakMap`?
87. How do you use `Object.assign` for shallow cloning?
88. How do you create a recursive function in JavaScript?
89. How do you use the `Intl` object for localization?
90. What is the purpose of `Promise.resolve` and `Promise.reject`?
91. How do you handle long-running tasks in JavaScript?
92. What are `async iterators`?
93. How do you create and use an async iterator?
94. What is the purpose of the `eval` function?
95. How do you prevent XSS attacks in JavaScript?
96. How do you manage memory in JavaScript?
97. What is a garbage collector, and how does it work?
98. What is event-driven programming in JavaScript?
99. How do you use `CustomEvent` in JavaScript?
00. What is the `IntersectionObserver` API?

Advanced Level (100 Questions)

1. What is the JavaScript engine, and how does it work?
2. How does JavaScript's event loop differ from other languages?
3. What are the different phases of the event loop?
4. How do you optimize the performance of JavaScript applications?
5. How does JavaScript handle concurrency?
6. What is the difference between stack and heap memory?
7. How does the JavaScript engine handle memory leaks?
8. How do you profile JavaScript code performance?
9. How do you optimize DOM manipulation in JavaScript?
10. What are the different types of scopes in JavaScript?
11. How does the `with` statement affect scope?
12. How does JavaScript's lexical scoping work?

13. How do you implement lazy loading in JavaScript?
14. How do you create and use a `proxy` object?
15. What are `Symbols`, and why are they used?
16. How does the garbage collector work in V8?
17. What is a `WeakMap` vs a `Map`, and when would you use each?
18. How does JavaScript handle tail call optimization?
19. How do you implement a linked list in JavaScript?
20. What is an `abstract` class, and how do you implement it in JavaScript?
21. How do you use the `Reflect` API for metaprogramming?
22. What are private fields in JavaScript classes?
23. How do you implement a priority queue in JavaScript?
24. What is the difference between `map`, `filter`, and `reduce`?
25. How do you write a polyfill for `Promise.all`?
26. What is the `typeof` operator's behavior with `null`?
27. How do you implement an event emitter in JavaScript?
28. How does the `delete` operator work with object properties?
29. What are the performance implications of using `eval`?
30. How do you handle deep equality checks for objects?
31. How do you implement a binary search algorithm in JavaScript?
32. How do you optimize a large number of DOM updates?
33. What are the different ways to handle immutability in JavaScript?
34. How do you implement a debounced function?
35. What is the `BigInt` type, and how does it differ from `Number`?
36. How do you handle precision issues with floating-point numbers?
37. How do you implement a cache in JavaScript?
38. How does JavaScript handle binary data and bitwise operations?
39. How do you implement a custom iterator in JavaScript?
40. How do you use the `Reflect` API for managing objects?
41. What is the difference between `Object.create` and `new`?
42. How do you implement a hash table in JavaScript?
43. What are `generator` functions, and how do you use them?
44. How does `async` and `await` differ from `Promises`?
45. How do you implement memoization with a cache in JavaScript?
46. How do you create a custom `Promise`?
47. What is a `WeakSet`, and when would you use it?
48. How do you optimize the performance of event listeners?
49. How do you use `Map` to implement a key-value store?
50. What is the difference between `shallow copy` and `deep copy`?
51. How do you implement function currying in JavaScript?
52. How does tail call optimization work in JavaScript?
53. How do you implement a decorator function?
54. How do you optimize array operations in JavaScript?
55. How do you use `WeakMap` to manage memory for objects?

56. What is the difference between `for...in` and `for...of`?
57. How do you implement a binary heap in JavaScript?
58. How do you optimize recursive functions in JavaScript?
59. How do you implement a custom iterable object?
60. How do you implement function throttling?
61. What is the difference between `ArrayBuffer` and `SharedArrayBuffer`?
62. How do you use the `Atomics` object for safe multi-threading?
63. How do you optimize JavaScript code for performance?
64. How do you implement a simple state machine in JavaScript?
65. What is the difference between `NodeList` and `HTMLCollection`?
66. How do you handle circular references in JavaScript objects?
67. How do you optimize the performance of complex animations?
68. What are the implications of using `Object.freeze`?
69. How do you implement a basic publish-subscribe pattern?
70. How does the `in` operator work in JavaScript?
71. How do you implement a custom event system?
72. How do you manage memory in large-scale JavaScript applications?
73. How do you optimize the rendering performance of web pages?
74. What is the difference between `Promise.all` and `Promise.race`?
75. How do you implement a binary search tree in JavaScript?
76. How do you handle backpressure in JavaScript streams?
77. How do you implement a custom `Promise.all`?
78. How do you optimize the performance of web workers?
79. How do you use the `Performance` API for profiling JavaScript code?
80. How do you implement a trie data structure in JavaScript?
81. How do you handle large data sets in JavaScript?
82. How do you optimize JavaScript for mobile devices?
83. How do you implement an LRU cache in JavaScript?
84. How do you handle precision issues with `BigInt`?
85. How do you implement a simple virtual DOM in JavaScript?
86. How do you optimize the memory usage of JavaScript applications?
87. How do you implement a basic templating engine in JavaScript?
88. How do you use the `SharedArrayBuffer` for multi-threading?
89. How do you optimize the performance of large-scale applications?
90. How do you implement a custom event loop in JavaScript?
91. How do you optimize the garbage collection in JavaScript?
92. How do you implement a custom `fetch` function?
93. How do you optimize the performance of WebAssembly in JavaScript?
94. How do you implement a reactive system in JavaScript?
95. How do you optimize the performance of service workers?
96. How do you use the `BroadcastChannel` API for inter-tab communication?
97. How do you implement a custom `localStorage` system?
98. How do you optimize the performance of JavaScript animations?

99. How do you implement a priority queue in JavaScript?
00. How do you handle edge cases in asynchronous code?

Beginner Level (101-200 Questions)

01. What is a function expression?
02. How do you check if a string includes a specific substring in JavaScript?
03. How do you create a number from a string in JavaScript?
04. How do you round a number to a specific number of decimal places?
05. How do you check if a variable is an array in JavaScript?
06. What is the difference between `Array.prototype.push()` and `Array.prototype.unshift()`?
07. How do you merge two arrays in JavaScript?
08. How do you copy an array in JavaScript?
09. What is the `indexOf` method in JavaScript?
10. How do you check if an array includes a specific element?
11. How do you remove duplicates from an array in JavaScript?
12. What is the `splice` method in JavaScript?
13. How do you find the length of an array?
14. What is the difference between `Array.prototype.slice()` and `Array.prototype.splice()`?
15. How do you remove the first element of an array in JavaScript?
16. How do you add an element at the beginning of an array?
17. What is a multidimensional array?
18. How do you create a multidimensional array in JavaScript?
19. How do you flatten a multidimensional array in JavaScript?
20. How do you find the index of an object in an array?
21. How do you loop through an array in JavaScript?
22. What is the `reduce` method in JavaScript?
23. How do you check if every element in an array passes a test?
24. What is the `some` method in JavaScript?
25. How do you reverse an array in JavaScript?
26. How do you sort an array of numbers in ascending order?
27. How do you sort an array of numbers in descending order?
28. What is the `find` method in JavaScript?
29. How do you shuffle an array in JavaScript?
30. How do you create a new array from an existing array?
31. What is the `concat` method in JavaScript?
32. How do you split a string into an array in JavaScript?
33. How do you join an array into a string in JavaScript?

34. What is the `filter` method in JavaScript?
35. How do you use the `forEach` method in JavaScript?
36. What is the difference between `map` and `forEach`?
37. How do you check if an object is empty in JavaScript?
38. What is `Object.assign()` in JavaScript?
39. How do you clone an object in JavaScript?
40. How do you merge two objects in JavaScript?
41. What is the `hasOwnProperty` method in JavaScript?
42. How do you check if a property exists in an object?
43. How do you get all the keys of an object in JavaScript?
44. How do you get all the values of an object in JavaScript?
45. How do you loop through an object in JavaScript?
46. What is the `Object.freeze` method in JavaScript?
47. How do you make an object immutable in JavaScript?
48. How do you define a method in an object in JavaScript?
49. How do you access nested objects in JavaScript?
50. How do you use the `delete` operator in JavaScript?
51. What is an event listener in JavaScript?
52. How do you add an event listener in JavaScript?
53. What is the `click` event in JavaScript?
54. How do you remove an event listener in JavaScript?
55. How do you trigger an event in JavaScript?
56. What is the difference between `addEventListener` and `onclick`?
57. What is event delegation in JavaScript?
58. How do you prevent the default behavior of an event?
59. How do you stop event propagation in JavaScript?
60. What is the `keydown` event in JavaScript?
61. What is the `keyup` event in JavaScript?
62. How do you detect a key press in JavaScript?
63. How do you handle form submission in JavaScript?
64. What is the `submit` event in JavaScript?
65. How do you prevent form submission in JavaScript?
66. What is the `change` event in JavaScript?
67. How do you handle file uploads in JavaScript?
68. How do you validate form inputs in JavaScript?
69. How do you handle checkbox and radio button states in JavaScript?
70. How do you get the value of a form input in JavaScript?
71. How do you clear a form in JavaScript?
72. What is the `DOMContentLoaded` event in JavaScript?
73. How do you handle mouse events in JavaScript?
74. What is the `mouseover` event in JavaScript?
75. How do you handle drag-and-drop in JavaScript?
76. What is the `drop` event in JavaScript?

77. How do you handle touch events in JavaScript?
78. What is the `touchstart` event in JavaScript?
79. What is the `touchmove` event in JavaScript?
80. How do you detect a touch screen in JavaScript?
81. What is the `scroll` event in JavaScript?
82. How do you handle window resize events in JavaScript?
83. How do you handle window unload events in JavaScript?
84. What is the `beforeunload` event in JavaScript?
85. How do you handle focus and blur events in JavaScript?
86. What is the `focus` event in JavaScript?
87. How do you handle input focus in JavaScript?
88. What is the `blur` event in JavaScript?
89. How do you handle copy and paste events in JavaScript?
90. What is the `copy` event in JavaScript?
91. How do you handle clipboard data in JavaScript?
92. What is the `cut` event in JavaScript?
93. How do you detect the user's browser in JavaScript?
94. How do you detect the user's operating system in JavaScript?
95. How do you detect the user's device type in JavaScript?
96. How do you detect if JavaScript is enabled in the browser?
97. How do you detect the user's screen resolution in JavaScript?
98. What is the `navigator` object in JavaScript?
99. How do you detect the user's language in JavaScript?
00. How do you redirect a user to another page in JavaScript?

Intermediate Level (101-200 Questions)

01. How does == vs === work in JavaScript?
02. What is the difference between `null` and `undefined`?
03. How do you compare two objects in JavaScript?
04. How do you check if an object has a property in JavaScript?
05. What is the difference between `Object.keys()` and `Object.values()`?
06. How do you clone an object in JavaScript?
07. How do you merge two objects in JavaScript?
08. What is a `Promise` in JavaScript?
09. How do you chain promises in JavaScript?
10. How do you handle errors in promises?
11. What is `async/await` in JavaScript?
12. How do you create an `async` function in JavaScript?
13. How do you handle errors with `async/await`?
14. What is the difference between synchronous and asynchronous code?
15. What is an event loop in JavaScript?
16. How does JavaScript handle asynchronous operations?
17. What is a callback function in JavaScript?
18. How do you avoid callback hell in JavaScript?

19. What is event bubbling in JavaScript?
20. What is event capturing in JavaScript?
21. How do you stop event propagation in JavaScript?
22. What is `this` keyword in JavaScript?
23. How does `this` behave in different contexts?
24. What is `call`, `apply`, and `bind` in JavaScript?
25. How do you use `call` in JavaScript?
26. How do you use `apply` in JavaScript?
27. How do you use `bind` in JavaScript?
28. What is hoisting in JavaScript?
29. How does hoisting affect variables in JavaScript?
30. How does hoisting affect functions in JavaScript?
31. What is closure in JavaScript?
32. How do you create a closure in JavaScript?
33. What is the purpose of closures in JavaScript?
34. How do you use closures in JavaScript?
35. What is a higher-order function in JavaScript?
36. How do you create a higher-order function in JavaScript?
37. What is the `arguments` object in JavaScript?
38. How do you use the `arguments` object in JavaScript?
39. What is the difference between `for...in` and `for...of`?
40. How do you loop through an object in JavaScript?
41. How do you loop through an array in JavaScript?
42. What is the difference between `Array.prototype.map()` and `Array.prototype.forEach()`?
43. How do you use the `reduce` method in JavaScript?
44. How do you use the `filter` method in JavaScript?
45. What is a regular expression in JavaScript?
46. How do you create a regular expression in JavaScript?
47. How do you test a string against a regular expression?
48. How do you extract a substring using a regular expression?
49. What is the `test` method in JavaScript?
50. What is the `exec` method in JavaScript?
51. How do you split a string using a regular expression?
52. How do you replace a substring using a regular expression?
53. How do you validate a form using a regular expression?
54. What is a module in JavaScript?
55. How do you create a module in JavaScript?
56. How do you import and export modules in JavaScript?
57. What is the difference between `default` and named exports?
58. How do you use `export default` in JavaScript?
59. How do you use named exports in JavaScript?
60. How do you import a module in JavaScript?
61. What is the `import` statement in JavaScript?

62. How do you use dynamic imports in JavaScript?
63. What is the `import()` function in JavaScript?
64. How do you handle circular dependencies in JavaScript modules?
65. What is tree shaking in JavaScript?
66. How do you optimize JavaScript code for performance?
67. What is a polyfill in JavaScript?
68. How do you use Babel in JavaScript?
69. What is Webpack in JavaScript?
70. How do you use Webpack in JavaScript?
71. What is the difference between `var`, `let`, and `const`?
72. How do you choose between `let` and `const`?
73. What is the Temporal Dead Zone in JavaScript?
74. How do you handle exceptions in JavaScript?
75. What is a `try...catch` statement in JavaScript?
76. How do you use `finally` in JavaScript?
77. What is a custom error in JavaScript?
78. How do you create a custom error in JavaScript?
79. What is error handling in asynchronous code?
80. How do you handle errors in promises?
81. How do you handle errors with `async/await`?
82. What is the purpose of `Promise.reject()`?
83. How do you use `Promise.all()` in JavaScript?
84. How do you use `Promise.race()` in JavaScript?
85. What is a microtask in JavaScript?
86. How does the microtask queue work in JavaScript?
87. What is the `setTimeout` function in JavaScript?
88. What is the `setInterval` function in JavaScript?
89. How do you use `clearTimeout` in JavaScript?
90. How do you use `clearInterval` in JavaScript?
91. What is the difference between `setTimeout` and `setInterval`?
92. How do you debounce a function in JavaScript?
93. How do you throttle a function in JavaScript?
94. What is the difference between debouncing and throttling?
95. How do you implement a simple debounce function in JavaScript?
96. How do you implement a simple throttle function in JavaScript?
97. What is a generator function in JavaScript?
98. How do you create a generator function in JavaScript?
99. How do you use `yield` in a generator function?
00. How do you iterate over a generator function in JavaScript?

Advanced Level (101-200 Questions)

01. How does the JavaScript engine optimize the code during execution?
02. What is Just-In-Time (JIT) compilation in JavaScript?
03. How does JavaScript's garbage collection work?

04. How do you prevent memory leaks in JavaScript?
05. What are some common causes of memory leaks in JavaScript?
06. How do you profile memory usage in JavaScript?
07. What is the `performance` API in JavaScript?
08. How do you use the `Performance.now()` method?
09. How do you measure the time taken by a function in JavaScript?
10. What is the `requestAnimationFrame` method in JavaScript?
11. How do you use `requestAnimationFrame` for smooth animations?
12. What are the advantages of `requestAnimationFrame` over `setTimeout`?
13. How do you handle high-resolution timers in JavaScript?
14. What is a `Map` in JavaScript, and how is it different from an object?
15. What is a `Set` in JavaScript, and how is it different from an array?
16. How do you use `WeakMap` and `WeakSet` in JavaScript?
17. How does the `Proxy` object work in JavaScript?
18. How do you create a `Proxy` in JavaScript?
19. What are the use cases of a `Proxy` in JavaScript?
20. How do you use the `Reflect` API in JavaScript?
21. What is the difference between `Reflect.get` and `Proxy.get`?
22. How do you use `Proxy` for data validation?
23. What is an `Iterator` in JavaScript?
24. How do you create a custom iterator in JavaScript?
25. How does the `for...of` loop work with iterators?
26. What is the difference between an `Iterator` and a `Generator`?
27. How do you use `Symbol.iterator` in JavaScript?
28. How do you create a custom iterable object in JavaScript?
29. What is the `asyncIterator` in JavaScript?
30. How do you create an `asyncIterator` in JavaScript?
31. How do you implement asynchronous iteration with `for...await...of`?
32. How does JavaScript handle backpressure in streams?
33. What is the `ReadableStream` API in JavaScript?
34. How do you use the `ReadableStream` API?
35. What is the `WritableStream` API in JavaScript?
36. How do you use the `WritableStream` API?
37. What is the `TransformStream` API in JavaScript?
38. How do you use the `TransformStream` API?
39. How do you handle streaming data in JavaScript?
40. What are Web Workers in JavaScript?
41. How do you create a Web Worker in JavaScript?
42. How do you communicate with a Web Worker?
43. How do you handle errors in Web Workers?
44. How do you terminate a Web Worker in JavaScript?
45. What are Shared Workers in JavaScript?
46. How do you create a Shared Worker in JavaScript?

47. How do you handle communication in Shared Workers?
48. What are Service Workers in JavaScript?
49. How do you register a Service Worker?
50. How do you handle push notifications with Service Workers?
51. What is the Cache API in JavaScript?
52. How do you use the Cache API with Service Workers?
53. How do you handle background sync with Service Workers?
54. How do you implement offline-first web applications?
55. What is the `BroadcastChannel` API in JavaScript?
56. How do you use the `BroadcastChannel` API?
57. How do you handle cross-origin requests in JavaScript?
58. What is Cross-Origin Resource Sharing (CORS)?
59. How do you handle CORS in JavaScript?
60. How do you implement JSONP in JavaScript?
61. What is the Content Security Policy (CSP) in JavaScript?
62. How do you implement CSP in JavaScript?
63. How do you handle Same-Origin Policy (SOP) in JavaScript?
64. What is the `Subresource Integrity` (SRI) attribute in JavaScript?
65. How do you implement SRI in JavaScript?
66. What is the `async` attribute in script tags?
67. How does the `defer` attribute work in script tags?
68. What is the `module` type in script tags?
69. How do you handle script loading order in JavaScript?
70. How do you optimize JavaScript loading performance?
71. What are the security implications of using `eval` in JavaScript?
72. How do you prevent XSS attacks in JavaScript?
73. How do you sanitize user input in JavaScript?
74. What is Content Security Policy (CSP)?
75. How do you implement secure cookies in JavaScript?
76. How do you prevent CSRF attacks in JavaScript?
77. How do you implement CSRF tokens in JavaScript?
78. What is the `window.postMessage` method in JavaScript?
79. How do you use `window.postMessage` for secure communication?
80. What is the `window.localStorage` API?
81. How do you use `window.localStorage` securely?
82. What is the `window.sessionStorage` API?
83. How do you use `window.sessionStorage` securely?
84. How do you handle large datasets in JavaScript?
85. What is a `WebAssembly` in JavaScript?
86. How do you compile WebAssembly modules in JavaScript?
87. How do you load WebAssembly modules in JavaScript?
88. How do you use WebAssembly with JavaScript?
89. How do you optimize WebAssembly performance?

90. What are the security considerations with WebAssembly?
91. How do you debug WebAssembly in JavaScript?
92. How do you handle WebAssembly exceptions in JavaScript?
93. What is the `Atomics` object in JavaScript?
94. How do you use the `Atomics` object for safe multi-threading?
95. What is the `SharedArrayBuffer` in JavaScript?
96. How do you use `SharedArrayBuffer` for multi-threading?
97. What is the `BigInt` type in JavaScript?
98. How do you perform arithmetic with `BigInt`?
99. How do you handle precision issues with `BigInt`?
00. How do you handle large integers in JavaScript?

Beginner Level (201-300 Questions)

01. How do you detect when a DOM element is in the viewport?
02. How do you create a cookie in JavaScript?
03. How do you read a cookie in JavaScript?
04. How do you delete a cookie in JavaScript?
05. What is local storage in JavaScript?
06. How do you set an item in local storage?
07. How do you get an item from local storage?
08. How do you remove an item from local storage?
09. What is session storage in JavaScript?
10. How do you set an item in session storage?
11. How do you get an item from session storage?
12. How do you remove an item from session storage?
13. What is the difference between local storage and session storage?
14. How do you store an object in local storage?
15. How do you store an array in local storage?
16. How do you clear local storage?
17. How do you clear session storage?
18. What is the `history` object in JavaScript?
19. How do you navigate to a previous page using the `history` object?
20. How do you navigate to a next page using the `history` object?
21. How do you go to a specific page in the browser history?
22. What is the `location` object in JavaScript?
23. How do you get the current URL using the `location` object?
24. How do you redirect to another URL using the `location` object?

25. How do you reload the current page using the `location` object?
26. How do you get the protocol of the current page?
27. How do you get the hostname of the current page?
28. How do you get the pathname of the current page?
29. How do you get the query string of the current page?
30. How do you parse the query string in JavaScript?
31. What is the `navigator` object in JavaScript?
32. How do you detect the user's browser using the `navigator` object?
33. How do you detect the user's operating system using the `navigator` object?
34. How do you detect if cookies are enabled in the browser?
35. How do you detect if JavaScript is enabled in the browser?
36. What is a `console` object in JavaScript?
37. How do you log a message to the console in JavaScript?
38. How do you log an error message to the console?
39. How do you clear the console in JavaScript?
40. How do you time a function in JavaScript?
41. How do you count the number of times a function is called in JavaScript?
42. How do you assert a condition in JavaScript?
43. How do you group console logs together?
44. How do you log an object as a table in the console?
45. How do you add custom styling to console messages?
46. What is the `alert` function in JavaScript?
47. How do you display a confirmation dialog in JavaScript?
48. How do you display a prompt dialog in JavaScript?
49. How do you focus an input field when the page loads?
50. How do you select the text in an input field using JavaScript?
51. How do you disable an input field using JavaScript?
52. How do you enable an input field using JavaScript?
53. How do you change the placeholder text of an input field?
54. How do you change the value of an input field in JavaScript?
55. How do you check if a checkbox is checked in JavaScript?
56. How do you check all checkboxes on a page using JavaScript?
57. How do you uncheck all checkboxes on a page using JavaScript?
58. How do you get the selected value of a dropdown in JavaScript?
59. How do you set the selected value of a dropdown in JavaScript?
60. How do you get all the options of a dropdown in JavaScript?
61. How do you add a new option to a dropdown in JavaScript?
62. How do you remove an option from a dropdown in JavaScript?
63. How do you create a new HTML element in JavaScript?
64. How do you append a new element to the DOM in JavaScript?
65. How do you remove an element from the DOM in JavaScript?
66. How do you replace an existing element in the DOM?
67. How do you clone a DOM element in JavaScript?

68. How do you insert an element before another element in the DOM?
69. How do you insert an element after another element in the DOM?
70. How do you get the parent of a DOM element in JavaScript?
71. How do you get all the children of a DOM element?
72. How do you get the first child of a DOM element?
73. How do you get the last child of a DOM element?
74. How do you get the next sibling of a DOM element?
75. How do you get the previous sibling of a DOM element?
76. How do you get the computed style of a DOM element?
77. How do you change the style of a DOM element in JavaScript?
78. How do you add a class to a DOM element in JavaScript?
79. How do you remove a class from a DOM element in JavaScript?
80. How do you toggle a class on a DOM element in JavaScript?
81. How do you check if a DOM element has a specific class?
82. How do you set an attribute on a DOM element in JavaScript?
83. How do you get an attribute from a DOM element?
84. How do you remove an attribute from a DOM element?
85. How do you check if a DOM element has a specific attribute?
86. How do you set the innerHTML of a DOM element in JavaScript?
87. How do you get the innerHTML of a DOM element?
88. How do you set the text content of a DOM element?
89. How do you get the text content of a DOM element?
90. How do you handle DOMContentLoaded event in JavaScript?
91. How do you handle window load event in JavaScript?
92. How do you handle window resize event in JavaScript?
93. How do you handle window scroll event in JavaScript?
94. How do you handle form submit event in JavaScript?
95. How do you handle form reset event in JavaScript?
96. How do you prevent form submission in JavaScript?
97. How do you validate form input fields in JavaScript?
98. How do you display form validation errors in JavaScript?
99. How do you reset form input fields in JavaScript?
00. How do you submit a form using JavaScript?

Intermediate Level (201-300 Questions)

01. How does JavaScript handle integer overflow?
02. What is a **`Blob`** in JavaScript?
03. How do you create a **`Blob`** in JavaScript?
04. How do you read the contents of a **`Blob`** in JavaScript?
05. How do you convert a **`Blob`** to a **`File`** in JavaScript?
06. How do you upload a **`Blob`** to a server in JavaScript?
07. How do you download a **`Blob`** as a file in JavaScript?
08. What is the **`File`** API in JavaScript?
09. How do you read a file as text in JavaScript?

10. How do you read a file as binary data in JavaScript?
11. How do you create a file in JavaScript?
12. How do you handle file uploads in JavaScript?
13. How do you display an image preview before uploading it?
14. How do you drag and drop files in JavaScript?
15. How do you check the size of an uploaded file in JavaScript?
16. How do you check the type of an uploaded file in JavaScript?
17. How do you handle file upload errors in JavaScript?
18. How do you parse a CSV file in JavaScript?
19. How do you parse a JSON file in JavaScript?
20. How do you parse an XML file in JavaScript?
21. How do you generate a CSV file in JavaScript?
22. How do you generate a JSON file in JavaScript?
23. How do you generate an XML file in JavaScript?
24. What is the `FormData` object in JavaScript?
25. How do you append data to a `FormData` object?
26. How do you send a `FormData` object using `fetch`?
27. How do you send a `FormData` object using `XMLHttpRequest`?
28. How do you serialize a `FormData` object in JavaScript?
29. How do you deserialize a `FormData` object in JavaScript?
30. What is the `URLSearchParams` object in JavaScript?
31. How do you create a `URLSearchParams` object?
32. How do you append parameters to a URL in JavaScript?
33. How do you get the value of a query parameter in JavaScript?
34. How do you remove a query parameter from a URL?
35. How do you parse query parameters in JavaScript?
36. How do you encode a URL in JavaScript?
37. How do you decode a URL in JavaScript?
38. What is a `Promise.allSettled()` method?
39. How do you use `Promise.allSettled()` in JavaScript?
40. What is a `Promise.any()` method?
41. How do you use `Promise.any()` in JavaScript?
42. What is a `Promise.race()` method?
43. How do you use `Promise.race()` in JavaScript?
44. What is the `Promise.finally()` method in JavaScript?
45. How do you use `Promise.finally()` in JavaScript?
46. How do you cancel a `Promise` in JavaScript?
47. How do you timeout a `Promise` in JavaScript?
48. How do you chain multiple promises in JavaScript?
49. How do you handle multiple asynchronous operations in sequence?
50. How do you handle multiple asynchronous operations in parallel?
51. What is the `AbortController` API in JavaScript?
52. How do you use the `AbortController` to cancel a fetch request?

53. How do you use the `AbortController` to cancel multiple fetch requests?
54. How do you handle errors with `AbortController`?
55. How do you check if a fetch request has been aborted?
56. How do you retry a failed fetch request in JavaScript?
57. How do you handle fetch request timeouts in JavaScript?
58. How do you handle fetch request retries with exponential backoff?
59. What is a `ReadableStream` in JavaScript?
60. How do you create a `ReadableStream` in JavaScript?
61. How do you read data from a `ReadableStream` in JavaScript?
62. How do you handle `ReadableStream` errors in JavaScript?
63. How do you create a `WritableStream` in JavaScript?
64. How do you write data to a `WritableStream` in JavaScript?
65. How do you handle `WritableStream` errors in JavaScript?
66. What is a `TransformStream` in JavaScript?
67. How do you create a `TransformStream` in JavaScript?
68. How do you use a `TransformStream` to modify data?
69. How do you pipe streams in JavaScript?
70. How do you handle backpressure in streams?
71. How do you create a custom stream in JavaScript?
72. How do you implement stream-based file processing in JavaScript?
73. What is the `Stream.pipeTo()` method in JavaScript?
74. What is the `Stream.pipeThrough()` method in JavaScript?
75. How do you combine multiple streams in JavaScript?
76. How do you split a stream in JavaScript?
77. How do you buffer a stream in JavaScript?
78. What is the `fetch()` API in JavaScript?
79. How do you make a GET request using `fetch()`?
80. How do you make a POST request using `fetch()`?
81. How do you handle the response from a `fetch()` request?
82. How do you handle errors in a `fetch()` request?
83. How do you parse JSON from a `fetch()` response?
84. How do you send JSON data with `fetch()`?
85. How do you send form data with `fetch()`?
86. How do you send a file with `fetch()`?
87. How do you send multiple files with `fetch()`?
88. How do you handle CORS issues with `fetch()`?
89. How do you abort a `fetch()` request?
90. How do you retry a `fetch()` request?
91. How do you handle timeouts with `fetch()`?
92. How do you work with JSON Web Tokens (JWT) in JavaScript?
93. How do you encode a JWT in JavaScript?
94. How do you decode a JWT in JavaScript?
95. How do you validate a JWT in JavaScript?

96. How do you refresh a JWT in JavaScript?
97. How do you handle JWT expiration in JavaScript?
98. How do you store a JWT securely in JavaScript?
99. How do you use JWT for authentication in JavaScript?
00. How do you handle JWT in a RESTful API with JavaScript?

These questions, combined with the previous sets, cover a broad range of topics in JavaScript from basic syntax and concepts to advanced techniques and best practices.

Advanced Level (201-300 Questions)

01. What is the `Proxy` object in JavaScript, and how do you use it?
02. How do you create a custom iterator in JavaScript?
03. How do you create a custom `Symbol.iterator` for an object in JavaScript?
04. What is the `Reflect` API in JavaScript, and how is it used with `Proxy`?
05. How do you create a revocable `Proxy` in JavaScript?
06. What are weak references in JavaScript?
07. How do you create a `WeakRef` in JavaScript, and what are its use cases?
08. How do you handle garbage collection in JavaScript?
09. What is a `FinalizationRegistry`, and how do you use it?
10. How do you implement memoization in JavaScript?
11. How do you debounce a function in JavaScript?
12. How do you throttle a function in JavaScript?
13. What is a service worker in JavaScript, and how do you register one?
14. How do you cache assets using a service worker?
15. How do you handle background sync in a service worker?
16. How do you intercept network requests in a service worker?
17. What is the difference between service workers and web workers?
18. How do you handle push notifications in a service worker?
19. How do you manage offline functionality with service workers?
20. What is an `IndexedDB`, and how do you use it in JavaScript?
21. How do you create an `IndexedDB` database?
22. How do you create an object store in `IndexedDB`?
23. How do you add data to an `IndexedDB` object store?
24. How do you retrieve data from an `IndexedDB` object store?
25. How do you handle version changes in `IndexedDB`?
26. How do you delete an `IndexedDB` database?
27. How do you handle transactions in `IndexedDB`?
28. How do you use cursors in `IndexedDB`?
29. What are object relationships in `IndexedDB`?
30. How do you create and manage indexes in `IndexedDB`?

31. How do you work with promises in `IndexedDB`?
32. What is `WebAssembly`, and how do you use it in JavaScript?
33. How do you compile and instantiate a WebAssembly module in JavaScript?
34. How do you import and export functions between WebAssembly and JavaScript?
35. How do you handle memory in WebAssembly from JavaScript?
36. How do you work with typed arrays in JavaScript?
37. What are `ArrayBuffer` and `DataView` in JavaScript?
38. How do you manipulate binary data using `DataView`?
39. How do you share memory between WebAssembly and JavaScript?
40. How do you optimize JavaScript code for performance?
41. What are the best practices for minimizing memory leaks in JavaScript?
42. How do you profile JavaScript performance in the browser?
43. How do you optimize DOM manipulation for better performance?
44. What are the techniques to optimize rendering performance in JavaScript?
45. How do you handle large datasets in JavaScript efficiently?
46. How do you optimize JavaScript applications for mobile devices?
47. How do you implement lazy loading in JavaScript?
48. What are some strategies for optimizing JavaScript bundles?
49. How do you use `IntersectionObserver` for efficient lazy loading?
50. What is tree shaking in JavaScript, and how does it work?
51. How do you implement code splitting in JavaScript?
52. How do you optimize network performance in JavaScript?
53. How do you use `requestIdleCallback` for efficient background tasks?
54. What is a `SharedArrayBuffer`, and how do you use it in JavaScript?
55. How do you implement a producer-consumer model with `SharedArrayBuffer`?
56. How do you handle synchronization with `Atomics` in JavaScript?
57. What are `Atomics` in JavaScript, and how do they work?
58. How do you use the `BroadcastChannel` API in JavaScript?
59. How do you implement a messaging system with `BroadcastChannel`?
60. How do you use the `MessageChannel` API in JavaScript?
61. How do you transfer data between workers using `MessageChannel`?
62. How do you work with transferable objects in JavaScript?
63. What is the `OffscreenCanvas` API, and how do you use it?
64. How do you perform offscreen rendering in a web worker with `OffscreenCanvas`?
65. How do you optimize canvas animations using `requestAnimationFrame`?
66. How do you handle high-resolution time using the `Performance` API?
67. How do you measure script execution time with `Performance.now()`?
68. How do you use the `PerformanceObserver` to monitor performance?
69. How do you track resource loading with the `Resource Timing` API?
70. How do you handle long tasks with the `Long Task API`?
71. How do you handle responsive design with media queries in JavaScript?
72. How do you implement dark mode in JavaScript?
73. How do you use the `ResizeObserver` API in JavaScript?

74. How do you detect and respond to element resizing in JavaScript?
75. How do you manage state in large-scale JavaScript applications?
76. What are the best practices for managing global state in JavaScript?
77. How do you implement a custom state management system in JavaScript?
78. What are the principles of reactive programming in JavaScript?
79. How do you use the `RxJS` library for reactive programming in JavaScript?
80. How do you implement observables in JavaScript?
81. What is the role of functional programming in JavaScript?
82. How do you implement currying in JavaScript?
83. How do you create higher-order functions in JavaScript?
84. How do you handle side effects in functional programming with JavaScript?
85. How do you implement immutability in JavaScript?
86. How do you use monads in JavaScript for managing side effects?
87. How do you implement a custom event emitter in JavaScript?
88. What are the best practices for designing APIs in JavaScript?
89. How do you implement a RESTful API with JavaScript?
90. How do you secure a JavaScript-based API?
91. How do you handle authentication and authorization in a JavaScript API?
92. How do you use `OAuth` for securing JavaScript applications?
93. How do you implement rate limiting in JavaScript?
94. How do you handle pagination in JavaScript APIs?
95. How do you use GraphQL in JavaScript applications?
96. How do you design and implement a GraphQL API in JavaScript?
97. What are the best practices for error handling in JavaScript APIs?
98. How do you handle validation in JavaScript APIs?
99. How do you implement caching in JavaScript APIs?
00. How do you monitor and log JavaScript applications