

<https://github.com/sudheerj/javascript-interview-questions#what-is-the-difference-between-call-apply-and-bind>

<https://github.com/sudheerj/reactjs-interview-questions>

1. **What is JavaScript?** Explain what JavaScript is and its role in web development.
  - a. JavaScript is one of the most widely used web development languages for adding interactivity to the web page. It was designed to build dynamic web pages at first and now may be run on the server and on almost any that has the JS Engine installed. It enables client-side scripting to allow developers to enrich the user interfaces and enhance the user experiences.
2. **What are the differences between var, let, and const?** Discuss scope, hoisting, and reassignment.

a.

	Scope	Hoisting	Reassignment
var	Function-scope	Function-scope	Re-declare and updated
let	Block-scope	Block-scope	updated
const	Block-scope	Block-scope	No

3. **What are data types in JavaScript? Describe**

a. Primitive types

String - a series of characters and is written with quotes
Number - a number and can be written with or without decimals $(-2^{53}+1$ to $2^{53}-1)$
BigInt - store numbers which are above the limitation of the Number data type
Boolean - logical entity used for conditional testing
Undefined - when a variable is declared but not assigned.
Null - a non-existent or a invalid value
Symbol - store an anonymous and unique value

b. Reference types - stores as a reference in memory

Object - collection of data
Arrays

Functions
-----------

4. **Difference between var and let keyword in javascript.**
  - a. Var variables declared with var are initialized with undefined while let variables are not initialized until declaration.
  - b. When var is declared in the global scope, it becomes a property of the global object while let does not create a property on the global object.
5. **Explain Implicit Type Coercion in javascript.**
  - a. Implicit type coercion in javascript is the automatic conversion of value from one data type to another. It takes place when the operands of an expression are of different data types. Example,  $3 + "3" = "33"$ ,  $33 - "3" = 30$ ,  $3 == "3"$ ,
6. **Is javascript a statically typed or a dynamically typed language?**
  - a. JavaScript is a dynamically typed language. In a dynamically typed language, the type of a variable is checked during run-time in contrast to a statically typed language, where the type of a variable is checked during compile-time.
7. **What is NaN property in JavaScript?**
  - a. NaN property represents the “**Not-a-Number**” value. It indicates a value that is not a legal number. `Typeof(NaN)` will return a Number.
8. **What is a closure in JavaScript?** Explain how closures work and provide an example.
  - a.
9. **What is the difference between function declarations and function expressions?** Discuss hoisting and the context in which each can be used.
  - a.
10. **What are arrow functions and how do they differ from regular functions?** Talk about syntax, this binding, and use cases.
  - a.
11. **Explain hoisting in JavaScript.**
  - a. Hoisting is the default behavior of javascript where all the variable and function declarations are moved on top. Irrespective of where the variables and functions are declared they are moved on Top of the scope. Use “use strict” to avoid hoisting
12. **How do you create an object in JavaScript?** Discuss different ways to create objects (object literals, constructor functions, classes).
  - a.
13. **What are some methods to manipulate arrays?** Explain methods like push, pop, shift, unshift, map, filter, reduce, etc.

- a.
- 14. **What is the difference between == and ===?** Discuss type coercion and strict equality.
  - a. Both are comparison operators. “==” is used to compare values whereas “===” is used to compare both values and types.
- 15. **What is the event loop in JavaScript?** Explain how the event loop works with the call stack and callback queue.
  - a.
- 16. **What are Promises and how do they work?** Define promises, their states, and how to handle them with .then() and .catch().
  - a.
- 17. **What is async/await?** Explain how async functions work and how they relate to promises.
  - a.
- 18. **What are modules in JavaScript?** Discuss ES6 modules and CommonJS, including import and export.
  - a.
- 19. **What is the difference between synchronous and asynchronous code?** Explain the differences and provide examples.
  - a.
- 20. **Explain passed by value and passed by reference.**
  - a. In JavaScript, primitive data types are passed by value and non-primitive data types are passed by reference. Pass by value creates a copy of the value while pass by reference passes a reference to the original data.
- 21. **What is an Immediately Invoked Function in JavaScript?**
  - a.
- 22. **How can you improve the performance of a JavaScript application?** Discuss techniques like debouncing, throttling, minimizing DOM manipulations, etc.
  - a.
- 23. **What is the importance of immutability in JavaScript?** Explain how immutability can lead to fewer bugs and easier state management.
  - a.
- 24. **What are the common security issues in JavaScript?** Discuss issues like XSS, CSRF, and how to mitigate them.
  - a.
- 25. **Can you explain the concept of "this" in JavaScript?** Discuss how this is determined in different contexts (global, object methods, constructors, etc.).

a.

26. **Write a function to flatten an array.** Provide a sample implementation and discuss the approach.

27. **Why do we use the word “debugger” in javascript?**

- a. The debugger for the browser must be activated in order to debug the code. Built-in debuggers may be switched on and off, requiring the user to report faults. The remaining section of the code should stop execution before moving on the next line while debugging.

28. **What are the possible ways to create objects in JavaScript?**

a.

Object literal syntax	<pre>var object = {   name: "Sudheer",   age: 34 };</pre>
Object constructor	<pre>var object = Object();</pre>
Object's create method - used to create a new object by passing the specified prototype object and properties as arguments	<pre>var object = Object.create(null); let vehicle = {   wheels: '4', } let carProps = {   type: {     value: 'Volkswagen'   } } var car = Object.create(vehicle, carProps);</pre>
Function constructor	<pre>function Person(name) {   this.name = name;   this.age = 21; } var object = new Person("Sudheer");</pre>
Function constructor with prototypes	<pre>function Person() {} Person.prototype.name = "Sudheer"; var object = new Person();</pre>
Object's assign method - Used to copy all the properties from one or more source objects and stores them in to a target object	<pre>const orgObject = { company: 'XYZ Corp'}; const carObject = { name: 'Toyota'}; const staff = Object.assign({}, orgObject, carObject);</pre>

ES6 Class syntax	<pre>class Person {   constructor(name) {     this.name = name;   } }</pre> <pre>var object = new Person("Sudheer");</pre>
Singleton pattern - A Singleton is an object which can only be instantiated one time. Repeated calls to its constructor return the same instance	<pre>var object = new (function () {   this.name = "Sudheer"; })();</pre>

### 29. What is a prototype chain?

- Prototype chaining is used to build new types of objects based on existing ones. (Inheritance). It enables inheritance and method sharing among objects. It allows an object to access properties and methods of other objects. I.e, when you create an object using a constructor function or a class, the created object inherits properties from a prototype object.

### 30. What is the difference between Call, Apply and Bind?

- They are functions used to set the value of this for a function and invoke it.

Call	Invokes a function with a given this value and arguments provided one by one
Apply	Invokes the function with a given this value and allows you to pass in arguments as an array
Bind	Returns a new function, allowing you to pass any number of arguments

Call and Apply are pretty much interchangeable. Both execute the current function immediately, while call takes a comma separated list of arguments and Apply takes an array.

### 31. What is JSON and its common operations

- JSON is a text-based data format following JavaScript object syntax, which was popularized by Douglas Crockford. It is useful when you want to transmit data across a network. It is basically just a text file with an extension of .json, and a MIME type of application/json
- Parsing: Converting a string to a native object - `JSON.parse(text);`
- Stringification: Converting a native object to a string so that it can be transmitted across the network - `JSON.stringify(object);`

32. What is the purpose of the array slice method
33. What is the purpose of the array splice method
34. What is the difference between slice and splice
35. How do you compare Object and Map
36. What is the difference between == and === operators
37. 10 What are lambda expressions or arrow functions
38. 11 What is a first class function
39. 12 What is a first order function
40. 13 What is a higher order function
41. 14 What is a unary function
42. 15 What is the currying function
43. 16 What is a pure function
44. 17 What is the purpose of the let keyword
45. 18 What is the difference between let and var
46. 19 What is the reason to choose the name let as a keyword
47. 20 How do you redeclare variables in a switch block without an error
48. 21 What is the Temporal Dead Zone
49. 22 What is an IIFE (Immediately Invoked Function Expression)
50. 23 How do you decode or encode a URL in JavaScript?
51. 24 What is memoization
52. 25 What is Hoisting
53. 26 What are classes in ES6
54. 27 What are closures
55. 28 What are modules
56. 29 Why do you need modules
57. 30 What is scope in javascript
58. 31 What is a service worker
59. 32 How do you manipulate DOM using a service worker
60. 33 How do you reuse information across service worker restarts
61. 34 What is IndexedDB
62. 35 What is web storage
63. 36 What is a post message
64. 37 What is a Cookie
65. 38 Why do you need a Cookie
66. 39 What are the options in a cookie
67. 40 How do you delete a cookie
68. 41 What are the differences between cookie, local storage and session storage
69. 42 What is the main difference between localStorage and sessionStorage
70. 43 How do you access web storage

- 71. 44 What are the methods available on session storage**
- 72. 45 What is a storage event and its event handler**
- 73. 46 Why do you need web storage**
- 74. 47 How do you check web storage browser support**
- 75. 48 How do you check web workers browser support**
- 76. 49 Give an example of a web worker**
- 77. 50 What are the restrictions of web workers on DOM**
- 78. 51 What is a promise**
- 79. 52 Why do you need a promise**
- 80. 53 What are the three states of promise**
- 81. 54 What is a callback function**
- 82. 55 Why do we need callbacks**
- 83. 56 What is a callback hell**
- 84. 57 What are server-sent events**
- 85. 58 How do you receive server-sent event notifications**
- 86. 59 How do you check browser support for server-sent events**
- 87. 60 What are the events available for server sent events**
- 88. 61 What are the main rules of promise**
- 89. 62 What is callback in callback**
- 90. 63 What is promise chaining**
- 91. 64 What is promise.all**
- 92. 65 What is the purpose of the race method in promise**
- 93. 66 What is a strict mode in javascript**
- 94. 67 Why do you need strict mode**
- 95. 68 How do you declare strict mode**
- 96. 69 What is the purpose of double exclamation**
- 97. 70 What is the purpose of the delete operator**
- 98. 71 What is typeof operator**
- 99. 72 What is undefined property**
- 100. 73 What is null value**
- 101. 74 What is the difference between null and undefined**
- 102. 75 What is eval**
- 103. 76 What is the difference between window and document**
- 104. 77 How do you access history in javascript**
- 105. 78 How do you detect caps lock key turned on or not**
- 106. 79 What is NaN**
- 107. 80 What are the differences between undeclared and undefined variables**
- 108. 81 What are global variables**
- 109. 82 What are the problems with global variables**
- 110. 83 What is NaN property**

- 111. 84 What is the purpose of isFinite function
- 112. 85 What is an event flow
- 113. 86 What is event bubbling
- 114. 87 What is event capturing
- 115. 88 How do you submit a form using JavaScript
- 116. 89 How do you find operating system details
- 117. 90 What is the difference between document load and DOMContentLoaded events
- 118. 91 What is the difference between native, host and user objects
- 119. 92 What are the tools or techniques used for debugging JavaScript code
- 120. 93 What are the pros and cons of promises over callbacks
- 121. 94 What is the difference between an attribute and a property
- 122. 95 What is same-origin policy
- 123. 96 What is the purpose of void 0
- 124. 97 Is JavaScript a compiled or interpreted language
- 125. 98 Is JavaScript a case-sensitive language
- 126. 99 Is there any relation between Java and JavaScript
- 127. 100 What are events
- 128. 101 Who created javascript
- 129. 102 What is the use of preventDefault method
- 130. 103 What is the use of stopPropagation method
- 131. 104 What are the steps involved in return false usage
- 132. 105 What is BOM
- 133. 106 What is the use of setTimeout
- 134. 107 What is the use of setInterval
- 135. 108 Why is JavaScript treated as Single threaded
- 136. 109 What is an event delegation
- 137. 110 What is ECMAScript
- 138. 111 What is JSON
- 139. 112 What are the syntax rules of JSON
- 140. 113 What is the purpose JSON stringify
- 141. 114 How do you parse JSON string
- 142. 115 Why do you need JSON
- 143. 116 What are PWAs
- 144. 117 What is the purpose of clearTimeout method
- 145. 118 What is the purpose of clearInterval method
- 146. 119 How do you redirect new page in javascript
- 147. 120 How do you check whether a string contains a substring
- 148. 121 How do you validate an email in javascript
- 149. 122 How do you get the current url with javascript

150. 123 What are the various url properties of location object
151. 124 How do get query string values in javascript
152. 125 How do you check if a key exists in an object
153. 126 How do you loop through or enumerate javascript object
154. 127 How do you test for an empty object
155. 128 What is an arguments object
156. 129 How do you make first letter of the string in an uppercase
157. 130 What are the pros and cons of for loops
158. 131 How do you display the current date in javascript
159. 132 How do you compare two date objects
160. 133 How do you check if a string starts with another string
161. 134 How do you trim a string in javascript
162. 135 How do you add a key value pair in javascript
163. 136 Is the !-- notation represents a special operator
164. 137 How do you assign default values to variables
165. 138 How do you define multiline strings
166. 139 What is an app shell model
167. 140 Can we define properties for functions
168. 141 What is the way to find the number of parameters expected by a function
169. 142 What is a polyfill
170. 143 What are break and continue statements
171. 144 What are js labels
172. 145 What are the benefits of keeping declarations at the top
173. 146 What are the benefits of initializing variables
174. 147 What are the recommendations to create new object
175. 148 How do you define JSON arrays
176. 149 How do you generate random integers
177. 150 Can you write a random integers function to print integers within a range
178. 151 What is tree shaking
179. 152 What is the need of tree shaking
180. 153 Is it recommended to use eval
181. 154 What is a Regular Expression
182. 155 What are the string methods that accept Regular expression
183. 156 What are modifiers in regular expression
184. 157 What are regular expression patterns
185. 158 What is a RegExp object
186. 159 How do you search a string for a pattern
187. 160 What is the purpose of exec method
188. 161 How do you change the style of a HTML element
189. 162 What would be the result of 1+2+'3'

190. 163 What is a debugger statement
191. 164 What is the purpose of breakpoints in debugging
192. 165 Can I use reserved words as identifiers
193. 166 How do you detect a mobile browser
194. 167 How do you detect a mobile browser without regexp
195. 168 How do you get the image width and height using JS
196. 169 How do you make synchronous HTTP request
197. 170 How do you make asynchronous HTTP request
198. 171 How do you convert date to another timezone in javascript
199. 172 What are the properties used to get size of window
200. 173 What is a conditional operator in javascript
201. 174 Can you apply chaining on conditional operator
202. 175 What are the ways to execute javascript after page load
203. 176 What is the difference between proto and prototype
204. 177 Can you give an example of when you really need a semicolon
205. 178 What is a freeze method
206. 179 What is the purpose of freeze method
207. 180 Why do I need to use freeze method
208. 181 How do you detect a browser language preference
209. 182 How to convert string to title case with javascript
210. 183 How do you detect javascript disabled in the page
211. 184 What are various operators supported by javascript
212. 185 What is a rest parameter
213. 186 What happens if you do not use rest parameter as a last argument
214. 187 What are the bitwise operators available in javascript
215. 188 What is a spread operator
216. 189 How do you determine whether object is frozen or not
217. 190 How do you determine two values same or not using object
218. 191 What is the purpose of using object is method
219. 192 How do you copy properties from one object to other
220. 193 What are the applications of assign method
221. 194 What is a proxy object
222. 195 What is the purpose of seal method
223. 196 What are the applications of seal method
224. 197 What are the differences between freeze and seal methods
225. 198 How do you determine if an object is sealed or not
226. 199 How do you get enumerable key and value pairs
227. 200 What is the main difference between Object.values and Object.entries  
method
228. 201 How can you get the list of keys of any object

229.	202	How do you create an object with prototype
230.	203	What is a WeakSet
231.	204	What are the differences between WeakSet and Set
232.	205	List down the collection of methods available on WeakSet
233.	206	What is a WeakMap
234.	207	What are the differences between WeakMap and Map
235.	208	List down the collection of methods available on WeakMap
236.	209	What is the purpose of uneval
237.	210	How do you encode an URL
238.	211	How do you decode an URL
239.	212	How do you print the contents of web page
240.	213	What is the difference between uneval and eval
241.	214	What is an anonymous function
242.	215	What is the precedence order between local and global variables
243.	216	What are javascript accessors
244.	217	How do you define property on Object constructor
245.	218	What is the difference between get and defineProperty
246.	219	What are the advantages of Getters and Setters
247.	220	Can I add getters and setters using defineProperty method
248.	221	What is the purpose of switch-case
249.	222	What are the conventions to be followed for the usage of switch case
250.	223	What are primitive data types
251.	224	What are the different ways to access object properties
252.	225	What are the function parameter rules
253.	226	What is an error object
254.	227	When you get a syntax error
255.	228	What are the different error names from error object
256.	229	What are the various statements in error handling
257.	230	What are the two types of loops in javascript
258.	231	What is nodejs
259.	232	What is an Intl object
260.	233	How do you perform language specific date and time formatting
261.	234	What is an Iterator
262.	235	How does synchronous iteration works
263.	236	What is an event loop
264.	237	What is call stack
265.	238	What is an event queue
266.	239	What is a decorator
267.	240	What are the properties of Intl object
268.	241	What is an Unary operator

269.	242	How do you sort elements in an array
270.	243	What is the purpose of compareFunction while sorting arrays
271.	244	How do you reversing an array
272.	245	How do you find min and max value in an array
273.	246	How do you find min and max values without Math functions
274.	247	What is an empty statement and purpose of it
275.	248	How do you get metadata of a module
276.	249	What is a comma operator
277.	250	What is the advantage of a comma operator
278.	251	What is typescript
279.	252	What are the differences between javascript and typescript
280.	253	What are the advantages of typescript over javascript
281.	254	What is an object initializer
282.	255	What is a constructor method
283.	256	What happens if you write constructor more than once in a class
284.	257	How do you call the constructor of a parent class
285.	258	How do you get the prototype of an object
286.	259	What happens If I pass string type for getPrototype method
287.	260	How do you set prototype of one object to another
288.	261	How do you check whether an object can be extendable or not
289.	262	How do you prevent an object to extend
290.	263	What are the different ways to make an object non-extensible
291.	264	How do you define multiple properties on an object
292.	265	What is MEAN in javascript
293.	266	What Is Obfuscation in javascript
294.	267	Why do you need Obfuscation
295.	268	What is Minification
296.	269	What are the advantages of minification
297.	270	What are the differences between Obfuscation and Encryption
298.	271	What are the common tools used for minification
299.	272	How do you perform form validation using javascript
300.	273	How do you perform form validation without javascript
301.	274	What are the DOM methods available for constraint validation
302.	275	What are the available constraint validation DOM properties
303.	276	What are the list of validity properties
304.	277	Give an example usage of rangeOverflow property
305.	278	Is enums feature available in javascript
306.	279	What is an enum
307.	280	How do you list all properties of an object
308.	281	How do you get property descriptors of an object

309.	282	What are the attributes provided by a property descriptor
310.	283	How do you extend classes
311.	284	How do I modify the url without reloading the page
312.	285	How do you check whether an array includes a particular value or not
313.	286	How do you compare scalar arrays
314.	287	How to get the value from get parameters
315.	288	How do you print numbers with commas as thousand separators
316.	289	What is the difference between java and javascript
317.	290	Does JavaScript supports namespace
318.	291	How do you declare namespace
319.	292	How do you invoke javascript code in an iframe from parent page
320.	293	How do get the timezone offset from date
321.	294	How do you load CSS and JS files dynamically
322.	295	What are the different methods to find HTML elements in DOM
323.	296	What is jQuery
324.	297	What is V8 JavaScript engine
325.	298	Why do we call javascript as dynamic language
326.	299	What is a void operator
327.	300	How to set the cursor to wait
328.	301	How do you create an infinite loop
329.	302	Why do you need to avoid with statement
330.	303	What is the output of the following for loops
331.	304	List down some of the features of ES6
332.	305	What is ES6
333.	306	Can I redeclare let and const variables
334.	307	Does the const variable make the value immutable
335.	308	What are default parameters
336.	309	What are template literals
337.	310	How do you write multi-line strings in template literals
338.	311	What are nesting templates
339.	312	What are tagged templates
340.	313	What are raw strings
341.	314	What is destructuring assignment
342.	315	What are default values in destructuring assignment
343.	316	How do you swap variables in destructuring assignment
344.	317	What are enhanced object literals
345.	318	What are dynamic imports
346.	319	What are the use cases for dynamic imports
347.	320	What are typed arrays
348.	321	What are the advantages of module loaders

349.	322	What is collation
350.	323	What is for...of statement
351.	324	What is the output of below spread operator array
352.	325	Is PostMessage secure
353.	326	What are the problems with postmessage target origin as wildcard
354.	327	How do you avoid receiving postMessages from attackers
355.	328	Can I avoid using postMessages completely
356.	329	Is postMessages synchronous
357.	330	What paradigm is Javascript
358.	331	What is the difference between internal and external javascript
359.	332	Is JavaScript faster than server side script
360.	333	How do you get the status of a checkbox
361.	334	What is the purpose of double tilde operator
362.	335	How do you convert character to ASCII code
363.	336	What is ArrayBuffer
364.	337	What is the output of below string expression
365.	338	What is the purpose of Error object
366.	339	What is the purpose of EvalError object
367.	340	What are the list of cases error thrown from non-strict mode to strict mode
368.	341	Do all objects have prototypes
369.	342	What is the difference between a parameter and an argument
370.	343	What is the purpose of some method in arrays
371.	344	How do you combine two or more arrays
372.	345	What is the difference between Shallow and Deep copy
373.	346	How do you create specific number of copies of a string
374.	347	How do you return all matching strings against a regular expression
375.	348	How do you trim a string at the beginning or ending
376.	349	What is the output of below console statement with unary operator
377.	350	Does javascript uses mixins
378.	351	What is a thunk function
379.	352	What are asynchronous thunks
380.	353	What is the output of below function calls
381.	354	How to remove all line breaks from a string
382.	355	What is the difference between reflow and repaint
383.	356	What happens with negating an array
384.	357	What happens if we add two arrays
385.	358	What is the output of prepend additive operator on falsy values
386.	359	How do you create self string using special characters
387.	360	How do you remove falsy values from an array
388.	361	How do you get unique values of an array

389.	362	What is destructuring aliases
390.	363	How do you map the array values without using map method
391.	364	How do you empty an array
392.	365	How do you round numbers to certain decimals
393.	366	What is the easiest way to convert an array to an object
394.	367	How do you create an array with some data
395.	368	What are the placeholders from console object
396.	369	Is it possible to add CSS to console messages
397.	370	What is the purpose of dir method of console object
398.	371	Is it possible to debug HTML elements in console
399.	372	How do you display data in a tabular format using console object
400.	373	How do you verify that an argument is a Number or not
401.	374	How do you create copy to clipboard button
402.	375	What is the shortcut to get timestamp
403.	376	How do you flattening multi dimensional arrays
404.	377	What is the easiest multi condition checking
405.	378	How do you capture browser back button
406.	379	How do you disable right click in the web page
407.	380	What are wrapper objects
408.	381	What is AJAX
409.	382	What are the different ways to deal with Asynchronous Code
410.	383	How to cancel a fetch request
411.	384	What is web speech API
412.	385	What is minimum timeout throttling
413.	386	How do you implement zero timeout in modern browsers
414.	387	What are tasks in event loop
415.	388	What is microtask
416.	389	What are different event loops
417.	390	What is the purpose of queueMicrotask
418.	391	How do you use javascript libraries in typescript file
419.	392	What are the differences between promises and observables
420.	393	What is heap
421.	394	What is an event table
422.	395	What is a microTask queue
423.	396	What is the difference between shim and polyfill
424.	397	How do you detect primitive or non primitive value type
425.	398	What is babel
426.	399	Is Node.js completely single threaded
427.	400	What are the common use cases of observables
428.	401	What is RxJS

429. 402 What is the difference between Function constructor and function declaration
430. 403 What is a Short circuit condition
431. 404 What is the easiest way to resize an array
432. 405 What is an observable
433. 406 What is the difference between function and class declarations
434. 407 What is an async function
435. 408 How do you prevent promises swallowing errors
436. 409 What is deno
437. 410 How do you make an object iterable in javascript
438. 411 What is a Proper Tail Call
439. 412 How do you check an object is a promise or not
440. 413 How to detect if a function is called as constructor
441. 414 What are the differences between arguments object and rest parameter
442. 415 What are the differences between spread operator and rest parameter
443. 416 What are the different kinds of generators
444. 417 What are the built-in iterables
445. 418 What are the differences between for...of and for...in statements
446. 419 How do you define instance and non-instance properties
447. 420 What is the difference between isNaN and Number.isNaN?
448. 421 How to invoke an IIFE without any extra brackets?
449. 422 Is that possible to use expressions in switch cases?
450. 423 What is the easiest way to ignore promise errors?
451. 424 How do style the console output using CSS?
452. 425 What is nullish coalescing operator (??)?
453. 426 How do you group and nest console output?
454. 427 What is the difference between dense and sparse arrays?
455. 428 What are the different ways to create sparse arrays?
456. 429 What is the difference between setTimeout, setImmediate and process.nextTick?
457. 430 How do you reverse an array without modifying original array?
458. 431 How do you create custom HTML element?
459. 432 What is global execution context?
460. 433 What is function execution context?
461. 434 What is debouncing?
462. 435 What is throttling?
463. 436 What is optional chaining?
464. 437 What is an environment record?
465. 438 How to verify if a variable is an array?
466. 439 What is pass by value and pass by reference?

- 467. 440 What are the differences between primitives and non-primitives?
- 468. 441 How do you create your own bind method using either call or apply method?
- 469. 442 What are the differences between pure and impure functions?
- 470. 443 What is referential transparency?
- 471. 444 What are the possible side-effects in javascript?
- 472. 445 What are compose and pipe functions?
- 473. 446 What is module pattern?
- 474. 447 What is Function Composition?
- 475. 448 How to use await outside of async function prior to ES2022?
- 476. 449 What is the purpose of the this keyword in JavaScript?
- 477. 450 What are the uses of closures?
- 478. 451 What are the phases of execution context?
- 479. 452 What are the possible reasons for memory leaks?
- 480. 453 What are the optimization techniques of V8 engine?
- 481. 454 What are the examples of built-in higher order functions?
- 482. 455 What are the benefits higher order functions?
- 483. 456 How do you create polyfills for map, filter and reduce methods?
- 484. 457 What is the difference between map and forEach functions?
- 485. 458 Give an example of statements affected by automatic semicolon insertion?
- 486. 459 What are the event phases of a browser?
- 487. 460 What are the real world use cases of proxy?
- 488. 461 What are hidden classes?
- 489. 462 What is inline caching?
- 490. 463 What are the different ways to execute external scripts?
- 491. 464 What is Lexical Scope?
- 492. 465 How to detect system dark mode in javascript?
- 493. 466 What is the purpose of requestAnimationFrame method?
- 494. 467 What is the difference between substring and substr methods?
- 495. 468 How to find the number of parameters expected by a function?
- 496. 469 What is globalThis, and what is the importance of it?
- 497. 470 What are the array mutation methods?
- 498.