TOP 100 JAVASCRIPT INTERVIEW QUESTION

Prepared by: Devkant Bhagat

Linkedin I'd:- http://linkedin.com/in/devkantbhagat

Telegram: https://t.me/Techverse_hub

JavaScript Basics

1. **What is JavaScript?**

   JavaScript is a lightweight, interpreted programming language primarily used to create interactive and dynamic content on web pages.

2. **What are the different data types in JavaScript?**

   - **Primitive types**: String, Number, Boolean, Undefined, Null, BigInt, Symbol
   - **Non-primitive types**: Objects (including Arrays, Functions, etc.)

3. **What is the difference between var, let, and const?**

   - var: Function-scoped, can be redeclared and updated.
   - let: Block-scoped, cannot be redeclared, but can be updated.
   - const: Block-scoped, cannot be redeclared or updated.

4. **What are template literals?**

   Template literals use backticks (`) and allow embedding expressions using ${expression}. Example: `Hello, ${name}!`.

5. **What is the difference between == and ===?**

   - ==: Compares values with type coercion.
   - ===: Compares values without type coercion.

Functions

6. **What are arrow functions?**

   Arrow functions provide a concise syntax for writing functions. They don't bind their context.

7. **What is the difference between function declaration and function expression?**

   - **Function declaration**: Defined with the function keyword. Example: function greet() {}.
   - **Function expression**: Assigned to a variable. Example: const greet = function() {};.

8. **What is a callback function?**

   A function passed as an argument to another function and executed after the completion of that function.

9. **What is a pure function?**

   A pure function always produces the same output for the same input and has no side effects.

10. **What is the purpose of closures?**

    Closures allow a function to access variables from its outer scope even after the outer function has been executed.

Advanced JavaScript

11. **What is the difference between null and undefined?**
    - null: Explicitly set to indicate "no value."
    - undefined: A variable has been declared but not assigned a value.
12. **What are JavaScript promises?**

    Promises represent a value that may be available now, in the future, or never. They can be in one of three states: pending, fulfilled, or rejected.
13. **What is async/await?**

    It is a syntax for handling asynchronous code, making it look synchronous and easier to read.
14. **What are higher-order functions?**

    Functions that can take other functions as arguments or return them.
15. **What is the difference between call, apply, and bind?**
    - call: Invokes a function with a specified context and arguments passed individually.
    - apply: Similar to a call but takes arguments as an array.
    - bind: Returns a new function with this bound to a specified object.

DOM Manipulation

16. **How can you select elements in the DOM?**
    - getElementById()
    - getElementsByClassName()
    - getElementsByTagName()
    - querySelector()
    - querySelectorAll()
17. **What is the difference between a window and a document?**
    - window: Represents the browser window and global context.
    - document: Represents the DOM (content of the web page).
18. **What is the purpose of addEventListener?**

    It adds an event handler to an element without overwriting existing handlers.

19. **How do you create a DOM element in JavaScript?**

    Using document.createElement() and appending it to the DOM using methods like appendChild.

20. **What is event delegation?**

    Attaching a single event listener to a parent element to manage events for child elements.

ES6+ Features

21. **What are rest and spread operators?**
    - **Rest (...args)**: Collects arguments into an array.
    - **Spread (...array)**: Spreads elements of an array/object.
22. **What are JavaScript modules?**

    Modules allow code to be divided into reusable chunks using import and export.

23. **What is destructuring?**

    A syntax to extract values from arrays or objects into variables.

24. **What are generators?**

    Functions that can pause execution (yield) and resume later, returning an iterator.

25. **What is the difference between Map and WeakMap?**
    - Map: Stores key-value pairs. Keys can be any type.
    - WeakMap: Keys must be objects, and they are weakly referenced.

JavaScript Objects and Prototypes

26. **What is the difference between an object and a Map in JavaScript?**

- **Object**: Keys are strings or symbols, unordered.
- **Map**: Keys can be any type and maintain insertion order.

27. **What is prototypal inheritance?**
    Prototypal inheritance allows objects to inherit properties and methods from another object.
28. **What is the Object.create() method?**
    It creates a new object with the specified prototype object and properties.
29. **How do you clone an object in JavaScript?**

- Using Object.assign({}, obj)
- Using the spread operator: { ...obj }

30. **What is a JavaScript class?**
    Classes are syntactic sugar over JavaScript's prototype-based inheritance to define constructors and methods.

Error Handling

31. **What is the purpose of try...catch in JavaScript?**
    It handles exceptions by running the code inside try and executing catch if an error occurs.
32. **What is the final block used for?**
    It executes code after try and catch, regardless of the outcome.
33. **How can you create a custom error in JavaScript?**
    By using the Error class:

```
throw new Error('Custom error message');
```

34. **What is the error event in JavaScript?**
    It is triggered when a runtime error occurs in the script.
35. **How can you handle asynchronous errors?**
    By using .catch() for Promises or try...catch with async/await.

Asynchronous JavaScript

36. **What is the event loop in JavaScript?**
    It is a mechanism that handles the execution of multiple tasks, including callbacks and asynchronous code.
37. **What is the difference between microtasks and microtasks?**

- **Microtasks**: Processed before microtasks, e.g., Promises.
- **Macrotasks**: These include tasks like setTimeout and setInterval.

38. **How does setTimeout work in JavaScript?**
    It schedules a task to run after a specified delay.
39. **What is Promise? All?**
    It resolves when all promises passed as an array are resolved or rejected when any promise fails.
40. **What is the purpose of Promise? Race?**
    It resolves or rejects as soon as any one of the promises in the array is resolved or rejected.

JavaScript Arrays

41. **What are the different ways to iterate over an array?**

- for loop
- forEach()
- map()
- for...of loop

42. **What is the difference between map() and forEach()?**

- map(): Creates a new array with the results of calling a function.
- forEach(): Executes a function on each array element without returning a new array.

43. **What is the use of reduce() in JavaScript?**
    It reduces an array to a single value by applying a function on each element.

44. **What is the purpose of Array.from()?**
    It creates a new array from array-like or iterable objects.

45. **How do you remove duplicates from an array?**

- Using Set:
- [...new Set(array)];

Memory and Performance

46. **What is a memory leak in JavaScript?**
    It occurs when memory that is no longer needed is not released, e.g., unused variables with global references.

47. **What is garbage collection?**
    The process of automatically reclaiming memory by removing objects that are no longer in use.

48. **What is the difference between deep copy and shallow copy?**

- **Shallow copy**: Only top-level properties are copied.
- **Deep copy**: All levels of properties are copied recursively.

49. **How do you optimise JavaScript performance?**

- Minimize DOM access
- Use requestAnimationFrame for animations
- Debounce or throttle events
- Use asynchronous code effectively.

50. **What are web workers in JavaScript?**
    Web Workers allow you to run JavaScript in the background, parallel to the main thread.

Miscellaneous

51. **What is hoisting in JavaScript?**
    Hoisting moves variable and function declarations to the top of their scope during compilation.
52. **What is the purpose of this keyword?**
    It refers to the object that is currently calling the function.
53. **What is the difference between == and Object.is()?**

- ==: Compares values with type coercion.
- Object.is(): Strict equality comparison, but considers NaN equal to itself and -0 different from +0.

54. **What is currying in JavaScript?**
    Currying transforms a function with multiple arguments into a sequence of functions, each taking a single argument.
55. **What is memoization?**
    A technique to cache function results to avoid redundant calculations.

Event Handling

56. **What is event bubbling?**
    Event bubbling occurs when an event starts at the target element and propagates upward to its ancestors.
57. **What is event capturing?**
    Event capturing (or trickling) occurs when an event starts at the topmost element and propagates down to the target element.
58. **How can you stop event propagation?**
    Using event.stopPropagation().
59. **What is the difference between stopPropagation and preventDefault?**

- stopPropagation: Stops the event from propagating to parent elements.
- preventDefault: Prevents the default action associated with the event.

60. **What are synthetic events in React?**
    Synthetic events are React's cross-browser wrapper around the browser's native events.

JavaScript Strings

61. **What are template literals, and how are they used?**
    Template literals use backticks (`) and allow embedded expressions using ${}:

    const greeting = Hello, ${name};

62. **How do you check if a string contains a substring?**
    Using includes():

    const password = "Hello World".includes("World"); // true

63. **What is the difference between substrate and substring?**

- substring (start, length): Extracts part of a string based on a start index and length.
- substring(start, end): Extracts characters between start and end indices.

64. **What is the purpose of split() in strings?**
    Splits a string into an array based on a specified delimiter.

    "a,b,c".split(","); // ['a', 'b', 'c']

65. **How do you reverse a string in JavaScript?**

    const reversed = str.split('').reverse().join('');

Regular Expressions

66. **What is a regular expression in JavaScript?**
    A pattern used to match character combinations in strings. Example:

    const regex = /ABC/;

67. **What does the g flag do in a regular expression?**
    The g flag stands for "global," allowing the regex to find all matches instead of stopping at the first.

68. **What is the purpose of the test() method in regular expressions?**
    Tests whether a regex matches a string.

    /abc/.test("abcdef"); // true

69. **What is the exec() method in regular expressions?**
    Executes a regex search and returns the matched result or null.

70. **How can you replace text using a regular expression?**
    Using replace():

    "hello world".replace(/world/, "JavaScript");

JavaScript Numbers

71. **How do you check if a value is a number in JavaScript?**
    Using typeof or Number.isFinite():

    ```
    Number.isFinite(123);    // true
    ```

72. **How do you convert a string to a number?**

- Number("123")
- parseInt("123")
- parseFloat("123.45")

73. **What is the difference between parseInt and Number?**

- parseInt: Parses only the integer part of a string.
- Number: Converts the entire string into a number.

74. **What is NaN in JavaScript?**
    NaN stands for "Not a Number" and represents an invalid number.

    ```
    isNaN("abc");   // true
    ```

75. **How do you generate a random number in JavaScript?**
    Using Math.random():

    ```
    const randomNum = Math.random();    // Between 0 and 1
    ```

Miscellaneous

76. **What is the difference between windows? onload and document.addEventListener('DOMContentLoaded')?**

- window.onload: Fires after the entire page (including assets) is loaded.
- DOMContentLoaded: Fires when the DOM is fully loaded without waiting for assets.

77. **What are JavaScript cookies?**
    Cookies store small pieces of data on the client side. Example:

    ```
    document.cookie = "username=JohnDoe";
    ```

78. **What is the eval() function in JavaScript?**
    It executes a string of JavaScript code.
    Warning: Avoid using it for security and performance reasons.

79. **What is use strict?**
    A directive that enforces stricter parsing and error handling.

    Example: "use strict";

80. **What is the difference between setTimeout and setInterval?**

- setTimeout: Executes a function once after a delay.
- setInterval: Repeats a function at regular intervals.

TypeScript

81. **What is TypeScript?**
    TypeScript is a superset of JavaScript that adds static typing.
82. **What are the advantages of TypeScript?**

- Type safety
- Improved IDE support
- Easier debugging
- Scalability for large projects

83. **How do you declare a variable in TypeScript?**

```
let age: number = 25;
```

84. **What are interfaces in TypeScript?**
    Interfaces define the shape of an object.

```
interface Person {
name: string;
age: number;
}
```

85. **What is a union type in TypeScript?**
    Allows a variable to hold more than one type.

```
let value: string | number;
```

Best Practices

86. **What is the purpose of IIFE (Immediately Invoked Function Expression)?**
    To create a private scope and avoid polluting the global namespace.

```
(function() {
console.log("IIFE");
})();
```

87. **What are design patterns in JavaScript?**
    Reusable solutions to common problems, e.g., Singleton, Factory, and Module patterns.

88. **What is a polyfill?**
    A piece of code that provides modern functionality in older browsers.

89. **What is the purpose of deferring in a <script> tag?**
    It loads the script in parallel but executes it only after the DOM is fully parsed.

90. **What is the difference between local storage, session storage, and cookies?**

- localStorage: Data persists indefinitely.
- sessionStorage: Data persists until the session ends.
- Cookies: Data sent with every request, limited to 4KB.

Advanced JavaScript Concepts

91. **What is the difference between call(), apply(), and bind()?**

- **call()**: Invokes a function with a specified value and arguments provided one by one.
- **apply()**: Similar to call(), but arguments are provided as an array.
- **bind()**: Returns a new function with a specified value and arguments.

92. **What are generator functions in JavaScript?**
93. Generator functions, declared with function*, can pause execution using yield and resume later.

```
function* generator() {
yield 1;
yield 2;
}
```

93. **What is the difference between Object.seal() and Object.freeze()?**

- **Object.seal()**: Prevents adding or removing properties but allows modification of existing properties.
- **Object.freeze()**: Prevents adding, removing, or modifying properties.

94. **What is the purpose of the Reflect API in JavaScript?**
    The Reflect API provides methods for interceptable JavaScript operations, like property manipulation.
    Example:

```
Reflect.set(obj, 'key', 'value');
```

95. **What is the async and await syntax in JavaScript?**

- **async**: Declares a function that always returns a Promise.
- **await**: Pauses the execution of an async function until a Promise is resolved or rejected.
  Example:

  async function fetch data() {

  const response = await fetch(url);

  console.log(response);

  }

JavaScript DOM

96. **How do you select elements in the DOM?**

- document.getElementById('id')
- document.querySelector('.class')
- document.getElementsByTagName('tag')

97. **What is the difference between innerHTML and textContent?**

- **innerHTML**: Returns or sets the HTML content of an element.
- **textContent**: Returns or sets the text content, without HTML tags.

98. **What is the purpose of the addEventListener() method?**
  It attaches an event handler to an element.

  element.addEventListener('click', function() {

  console.log('Clicked!');

  });

JavaScript Modules

99. **What is the difference between named and default exports in JavaScript modules?**

- **Named exports**: Export multiple values by name.
- **Default exports**: Export a single value or function.
  Example:

  export const value = 42; // Named export

  export default function() {} // Default export

Prepared by: Devkant Bhagat

Linkedin I’d:-http://linkedin.com/in/devkantbhagat

Telegram: https://t.me/Techverse_hub