Summary and Analysis Report

Summary: The following is a list of thirty-five JavaScript and CSS questions, with answers, that an interviewer might ask during a coding interview. The questions cover a range of topics, including JavaScript fundamentals, HTML, CSS, JavaScript programming principles, and JavaScript algorithms and data structures:

1. What is the purpose of a meta tag?
2. What is the purpose of the charset attribute?
3. How does the http-equiv refresh tag work?
4. What does the charset attribute represent?
5. What is the purpose of semantic HTML tags like nav, header, and footer?
6. What is the difference between a class and a tag in terms of specificity in CSS?
7. What are CSS variables (aka CSS custom properties) and how do they differ from CSS properties?
8. What is type coercion in JavaScript and give an example?
9. How does hoisting work in JavaScript?
10. What is the difference between letting and varing a variable in JavaScript?
11. What is the this keyword in JavaScript?
12. How global variables in JavaScript can be accessed?
13.

Sentiments: negative review

classifications: Tech

paraphrase: Here is your paraphrased text:

1. Meta tags provide information about a page. Description provides more detailed information about the page. Http-equiv refreshes the page when a URL is provided and after a period of time, the page will refresh and open the given URL. Charset is the numeric representation of characters.
2. Semantic HTML helps users and browsers understand the content of the page better. Semantic tags include nav, header, and footer.
3. Specificity is a measure of how tightly a CSS selector matches a particular HTML element; ID-100, class-10, tag-1, and *-0.
4. In CSS, var is used to define a variable, while property is used to define a property of an element. The :root selector has higher specificity than var.
5. In JavaScript, type coercion attempts to convert values to a number from a string, boolean, or other type.
6. Hoisting allows you to use functions and variables before they are declared. Global scope refers to anything not defined inside a function, which in the browser becomes associated with the window object and can be accessed using window.varName or varName. "Undefined" indicates a variable that has been declared but not given a value, while "not defined" indicates a variable that does not exist.
7. JavaScript is a loosely typed language, meaning that you can change the type of a

variable after creating it.
8. Scope refers to the location in which you can access a specific function or variable.
9. A constructor in JavaScript is a function that creates an instance of a class, typically called an "object". The new keyword is used to create an object, and the super keyword is used to access properties on an object literal or class's prototype.
10. Prototype inheritance in JavaScript involves adding properties and methods to a constructor function, which objects then inherit from a prototype.
11. To horizontally center a block element like a div, use margin: auto;
12. When you give a div a border radius of 50%, it will create a circle.
13. A Higher-Order Function takes a function as an argument or returns a function as output. Examples include Array.prototype.map, filter, and reduce.
14. In JavaScript, promises are used to handle asynchronous operations and have four states: fulfilled, rejected, pending, and settled. They can be created using the Promise constructor.
15. Let and const declare block scope variables in ES6, but var declares function or global scope variables.
16. Synchronous tasks occur one after the other in order while asynchronous tasks can occur in any order or simultaneously.
17. In JavaScript, Currying is the process of transforming a function that takes multiple arguments into a function that takes a single argument and returns another function if more arguments are needed.
18. Closures are functions that are nested in other functions or have lexical scope and are bundled as a closure.
19. This refers to an object in JavaScript, depending on how it is invoked. In an object method, this refers to the object itself; alone, this refers to the global object; in a function, this refers to the global object; in a function in strict mode, this is undefined; and in an event, this refers to the element that received the event. Methods such as call(), apply(), and bind() can refer to any object using this.
20. In JavaScript, hoisting refers to the process of using a function or variable before it is declared.
21. In JavaScript, the this keyword refers to an object. The object depends on the context in which this is invoked.
22. Promises are used to handle asynchronous operations in JavaScript. They make handling multiple asynchronous operations easier by avoiding the problem of "callback hell", or unmanageable code. A promise has four states: fulfilled, rejected, pending, and settled.
23. Currying is the process of transforming a function that takes multiple arguments into a function that takes a single argument and returns another function if more arguments are needed.
24. In CSS, the position property defines an element's position in a document. It works with left, right, top, bottom, and z-index to determine the final position of an element on a page. The position property can take five values: static, relative, absolute, fixed, and sticky.
25. Padding refers to the space between an element's content and its border, while margin is the space around an element's border.
26. In JavaScript, the this keyword refers to an object, the object depending on the

context in which this is invoked.
27. Synchronous tasks are executed one at a time in the order they are received, while asynchronous tasks can be executed in any order or simultaneously.
28. In JavaScript, Higher-Order Functions take a function as an argument or return a function as output.
29. Promises are used to handle asynchronous operations in JavaScript. They have four states: fulfilled, rejected, pending, and settled. They can be created using the Promise constructor.
30. Let, var, and const are ways to declare variables in JavaScript. Let and const have block scope, while var has function or global scope.
31. Currying is the process of transforming a function that takes multiple arguments into a function that takes a single argument and returns another function if more arguments are needed.

Let me know if anything requires clarification or needs to be rephrased.
I can also provide more information or answer questions if needed. enjoy!