1. What are the differences between Java and JavaScript?

JavaScript is a client-side scripting language and Java is object Oriented Programming language. Both of them are totally different from each other.

JavaScript: It is a light-weighted programming language ("scripting language") for developing interactive web pages. It can insert dynamic text into the HTML elements. JavaScript is also known as the browser's language.

Java: Java is one of the most popular programming languages. It is an object-oriented programming language and has a virtual machine platform that allows you to create compiled programs that run on nearly every platform. Java promised, "Write Once, Run Anywhere".

2. What are JavaScript Data Types?

There are three major Data types in JavaScript.

Primitive

Numbers

Strings

Boolean

Symbol

Trivial

Undefined

Null

Composite

Objects

Functions

Arrays

3. Which symbol is used for comments in JavaScript?

Comments prevent the execution of statements. Comments are ignored while the compiler executes the code. There are two type of symbols to represent comments in JavaScript:

Double slash: It is known as a single-line comment.

// Single line comment

Slash with Asterisk: It is known as a multi-line comment.

/*
Multi-line comments
...
*/

4. What would be the result of 3+2+”7″?

Here, 3 and 2 behave like an integer, and “7” behaves like a string. So 3 plus 2 will be 5. Then the output will be 5+”7″ = 57.

5. What is the use of the isNaN function?

The number isNan function determines whether the passed value is NaN (Not a number) and is of the type “Number”. In JavaScript, the value NaN is considered a type of number. It returns true if the argument is not a number, else it returns false.

6. Which is faster in JavaScript and ASP script?

JavaScript is faster compared to ASP Script. JavaScript is a client-side scripting language and does not depend on the server to execute. The ASP script is a server-side scripting language always dependable on the server.

7. What is negative infinity?

The negative infinity is a constant value represents the lowest available value. It means that no other number is lesser than this value. It can be generate using a self-made function or by an arithmetic operation. JavaScript shows the NEGATIVE_INFINITY value as -Infinity.

8. Is it possible to break JavaScript Code into several lines?

Yes, it is possible to break the JavaScript code into several lines in a string statement. It can be broken by using the backslash n ‘\n’.
For example:

console.log("A Online Computer Science Portal\n for Geeks")

The code-breaking line is avoid by JavaScript which is not preferable.

```
let gfg= 10, GFG = 5,
Geeks =
gfg + GFG;
```

9. Which company developed JavaScript?

Netscape developed JavaScript and was created by Brenden Eich in the year of 1995.

10. What are undeclared and undefined variables?

Undefined: It occurs when a variable is declare not not assign any value. Undefined is not a keyword.

Undeclared: It occurs when we try to access any variable which is not initialize or declare earlier using the var or const keyword. If we use 'typeof' operator to get the value of an undeclare variable, we will face the runtime error with the return value as "undefined". The scope of the undeclare variables is always global.

12. What are global variables? How are these variables declared, and what are the problems associated with them?

In contrast, global variables are the variables that define outside of functions. These variables have a global scope, so they can be used by any function without passing them to the function as parameters.

Example:

javascript

```javascript
let petName = "Rocky"; // Global Variable

myFunction();


function myFunction() {

    console.log("Inside myFunction - Type of petName:", typeof petName);

    console.log("Inside myFunction - petName:", petName);

}


console.log("Outside myFunction - Type of petName:", typeof petName);
```

```
console.log("Outside myFunction - petName:", petName);
```

Output

Inside myFunction - Type of petName: string

Inside myFunction - petName: Rocky

Outside myFunction - Type of petName: string

Outside myFunction - petName: Rocky

It is difficult to debug and test the code that relies on global variables.

13. What do you mean by NULL in JavaScript?

The NULL value represents that no value or no object. It is known as empty value/object.

14. How to delete property-specific values?

The delete keyword deletes the whole property and all the values at once like

```
let gfg={Course: "DSA", Duration:30};
delete gfg.Course;
```

15. What is a prompt box?

The prompt box is a dialog box with an optional message prompting the user to input some text. It is often used if the user wants to input a value before entering a page. It returns a string containing the text entered by the user, or null.

16. What is the ‘this’ keyword in JavaScript?

Functions in JavaScript are essential objects. Like objects, it can be assign to variables, pass to other functions, and return from functions. And much like objects, they have their own properties. ‘this’ stores the current execution context of the JavaScript program. Thus, when it use inside a function, the value of ‘this’ will change depending on how the function is defined, how it is invoked, and the default execution context.

17. Explain the working of timers in JavaScript. Also elucidate the drawbacks of using the timer, if any.

The timer executes some specific code at a specific time or any small amount of code in repetition to do that you need to use the functions setTimout, setInterval, and clearInterval. If the JavaScript code sets the timer to 2 minutes and when the times are up then the page

displays an alert message “times up”. The setTimeout() method calls a function or evaluates an expression after a specified number of milliseconds.

18. What is the difference between ViewState and SessionState?

ViewState: It is specific to a single page in a session.

SessionState: It is user specific that can access all the data on the web pages.

19. How to submit a form using JavaScript?

You can use document.form[0].submit() method to submit the form in JavaScript.

20. Does JavaScript support automatic type conversion?

Yes, JavaScript supports automatic type conversion.

JavaScript Intermediate Interview Questions

21. What are all the looping structures in JavaScript ?

while loop: A while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating if statement.

for loop: A for loop provides a concise way of writing the loop structure. Unlike a while loop, for statement consumes the initialization, condition and increment/decrement in one line thereby providing a shorter, easy to debug structure of looping.

do while: A do-while loop is similar to while loop with the only difference that it checks the condition after executing the statements, and therefore is an example of Exit Control Loop.

22. How can the style/class of an element be changed?

To change the style/class of an element there are two possible ways. We use document.getElementByID method

```
document.getElementById("myText").style.fontSize = "16px;

document.getElementById("myText").className = "class";
```

23. Explain how to read and write a file using JavaScript?

The readFile() functions is used for reading operation.

```
readFile( Path, Options, Callback)
```

The writeFile() functions is used for writing operation.

writeFile(Path, Data, Callback)

24. What is called Variable typing in JavaScript ?

The variable typing is the type of variable used to store a number and using that same variable to assign a “string”.

```
Geeks = 42;
Geeks = "GeeksforGeeks";
```

25. How to convert the string of any base to integer in JavaScript?

In JavaScript, parseInt() function is used to convert the string to an integer. This function returns an integer of base which is specified in second argument of parseInt() function. The parseInt() function returns Nan (not a number) when the string doesn’t contain number.

26. Explain how to detect the operating system on the client machine?

To detect the operating system on the client machine, one can simply use navigator.appVersion or navigator.userAgent property. The Navigator appVersion property is a read-only property and it returns the string that represents the version information of the browser.

27. What are the types of Pop up boxes available in JavaScript?

There are three types of pop boxes available in JavaScript.

Alert

Confirm

Prompt

28. What is the difference between an alert box and a confirmation box?

An alert box will display only one button which is the OK button. It is used to inform the user about the agreement has to agree. But a Confirmation box displays two buttons OK and cancel, where the user can decide to agree or not.

29. What is the disadvantage of using innerHTML in JavaScript?

There are lots of disadvantages of using the innerHTML in JavaScript as the content will replace everywhere. If you use += like “innerHTML = innerHTML + ‘html’” still the old content is replaced by HTML. It preserves event handlers attached to any DOM elements.

30. What is the use of void(0) ?

The void(0) is used to call another method without refreshing the page during the calling time parameter “zero” will be passed.

For further reading, check out our dedicated article on Intermediate Javascript Interview Questions. Inside, you’ll discover over 20 questions with detailed answers.

JavaScript Interview Questions for Experienced

31. What is the ‘Strict’ mode in JavaScript and how can it be enabled?

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program or a function in a “strict” operating context. This strict context prevents certain actions from being taken and throws more exceptions. The statement “use strict” instructs the browser to use the Strict mode, which is a reduced and safer feature set of JavaScript.

32. How to get the status of a CheckBox?

The DOM Input Checkbox Property is used to set or return the checked status of a checkbox field. This property is used to reflect the HTML Checked attribute.

```
document.getElementById("GFG").checked;
```

If the CheckBox is checked then it returns True.

33. How to explain closures in JavaScript and when to use it?

The closure is created when a child functions to keep the environment of the parent’s scope even after the parent’s function has already executed. The Closure is a locally declared variable related to a function. The closure will provide better control over the code when using them.

JavaScript

```
// Explanation of closure

function foo() {

    let b = 1;

    function inner() {

        return b;

    }
```

```
    return inner;
}
let get_func_inner = foo();


console.log(get_func_inner());
console.log(get_func_inner());
console.log(get_func_inner());
```

Output

1

1

1

34. What is the difference between call() and apply() methods ?

Both methods are used in a different situation

call() Method: It calls the method, taking the owner object as argument. The keyword this refers to the ‘owner’ of the function or the object it belongs to. We can call a method that can be used on different objects.

apply() Method: The apply() method is used to write methods, which can be used on different objects. It is different from the function call() because it takes arguments as an array.

35. How to target a particular frame from a hyperlink in JavaScript ?

This can be done by using the target attribute in the hyperlink. Like

New Page

36. Write the errors shown in JavaScript?

There are three different types of errors in JavaScript.

Syntax error: A syntax error is an error in the syntax of a sequence of characters or tokens that are intended to be written in a particular programming language.

Logical error: It is the most difficult error to be traced as it is the error on the logical part of the coding or logical error is a bug in a program that causes to operate incorrectly and terminate abnormally.

Runtime Error: A runtime error is an error that occurs during the running of the program, also known as an exception.

37. What is the difference between JavaScript and Jscript?

JavaScript

It is a scripting language developed by Netscape.

It is used to design client and server-side applications.

It is completely independent of Java language.

Jscript

It is a scripting language developed by Microsoft.

It is used to design active online content for the word wide Web.

38. What does *var myArray = [[]];* statement declares?

In JavaScript, this statement is used to declare a two-dimensional array.

39. How many ways an HTML element can be accessed in JavaScript code?

There are four possible ways to access HTML elements in JavaScript which are:

getElementById() Method: It is used to get the element by its id name.

getElementsByClass() Method: It is used to get all the elements that have the given classname.

getElementsByTagName() Method: It is used to get all the elements that have the given tag name.

querySelector() Method: This function takes CSS style selector and returns the first selected element.

40. What is the difference between innerHTML & innerText?

The innerText property sets or returns the text content as plain text of the specified node, and all its descendants whereas the innerHTML property sets or returns the plain text or HTML

contents in the elements. Unlike innerText, inner HTML lets you work with HTML rich text and doesn't automatically encode and decode text.

41. What is an event bubbling in JavaScript?

Consider a situation an element is present inside another element and both of them handle an event. When an event occurs in bubbling, the innermost element handles the event first, then the outer, and so on.

Promises, Callbacks, and Asynchronous Programming

Explain the Event Loop in JavaScript. How does it handle asynchronous operations?

The Event Loop is a continuous process that monitors the Call Stack and the Callback Queue. When the Call Stack is empty, it takes the first callback from the queue and pushes it onto the stack. This is how JavaScript handles asynchronous operations.

What is the difference between synchronous and asynchronous programming?

Synchronous code executes line by line in order. Asynchronous code allows other code to execute while waiting for long-running operations.

Describe callback hell and how Promises address this issue.

Callback hell occurs when multiple nested callbacks create hard-to-read code. Promises provide a cleaner way to handle asynchronous operations by chaining .then() methods.

Explain the Promise lifecycle (pending, fulfilled, rejected).

A Promise starts in a pending state. It transitions to fulfilled if the operation succeeds or rejected if it fails.

How do you handle multiple Promises using Promise.all and Promise.race?

Promise.all waits for all Promises to resolve before executing the callback. Promise.race resolves as soon as the first Promise settles.

What is Promise chaining and how does it work?

Promise chaining involves linking multiple .then() methods together. The value returned from one .then() is passed to the next.

Explain the concept of async/await and how it simplifies asynchronous code.

async/await provides a syntactic sugar for Promises. It makes asynchronous code look more synchronous.

How do you handle errors in Promises?

Use .catch() to handle errors in Promises.

Differentiate between thenable and promise-like objects.

A thenable is any object with a then method. A promise-like object adheres to the Promise specification.

What is the difference between setTimeout and Promise-based asynchronous operations?

setTimeout schedules a callback to be executed after a specific delay. Promises represent the eventual completion of an asynchronous operation.

Advanced Topics

Explain closures and their use cases.

A closure is a function that has access to variables from its outer scope, even after the outer function has returned. They are used for creating private variables, modules, and currying.

What is hoisting in JavaScript? How does it work with var, let, and const?

Hoisting is JavaScript's behavior of moving variable and function declarations to the top of their scope. var is hoisted with undefined value, while let and const are not hoisted.

Describe the difference between == and === operators.

== performs type coercion, while === checks for strict equality without type conversion.

Explain the concept of prototype inheritance in JavaScript.

Prototype inheritance allows objects to inherit properties from their prototype.

What is the difference between call, apply, and bind methods?

call and apply invoke a function with a specified this value and arguments. bind creates a new function with a fixed this value.

How do you implement inheritance using classes in JavaScript?

Use the extends keyword to create subclasses.

What is the module pattern in JavaScript?

A design pattern that encapsulates code and data within a self-contained unit.

Explain the concept of currying and partial application.

Currying converts a function that takes multiple arguments into a chain of functions, each taking a single argument. Partial application creates a new function by pre-filling some arguments.

What is the difference between shallow and deep copy?

A shallow copy creates a new object but copies references to child objects. A deep copy creates a new object and recursively copies all child objects.

How do you implement memoization in JavaScript?

Memoization is a technique to optimize function calls by caching the results of previous calls.

ES6 and Beyond

What are arrow functions and their advantages?

Arrow functions provide a concise syntax for writing functions. They have lexical this binding.

Explain destructuring in JavaScript.

Destructuring allows extracting values from arrays or objects into distinct variables.

What is the spread operator and rest parameter?

The spread operator (`) allows an iterable to be expanded into individual elements. The rest parameter collects multiple arguments into an array.

What are default parameters?

Default parameters provide default values for function parameters.

Explain template literals.

Template literals allow embedding expressions within strings using backticks (`).

Advanced Topics

What is the module system in JavaScript (CommonJS, AMD, ES6 modules)?

Different ways to organize and manage code in JavaScript.

Explain the concept of higher-order functions.

Functions that take other functions as arguments or return functions.

What is function currying?

Converting a function that takes multiple arguments into a chain of functions, each taking a single argument.

Explain the concept of immutability.

Data that cannot be changed after creation.

What are generators in JavaScript?

Functions that can be paused and resumed, allowing for asynchronous and iterative operations.

Note: This list covers a wide range of topics. Focus on understanding the core concepts and be prepared to provide code examples and explanations.

Would you like to focus on a specific area or difficulty level?