**JAVASCRIPT INTERVIEW QUESTION**

1. **What is Hoisting in javascript?**

Hoisting is the default behaviour of javascript. where variable and function declarations are moved to the top of their containing scope during the compilation phase, i.e its is executed before its has been declared.

Ex-

ABC();

function ABC(){

var name = “hello world”

console.log(name);

}

// Outputs " Hello world! "

NOTE- even when the function is declared after calling

1. **Difference between “ == “ and “ === “ operator ?**

THEY Both are comparison operators.

The difference between both the operators is that “==” is used to compare values of variable whereas, “ === “ is used to compare both values and data types of variable.

Ex-

var x = 2;

var y = "2";

console.log(x == y)

output: true (since the value of both x and y is the same)

console.log (x === y)

output: false (since the typeof x is "number" and typeof y is "string")

**3) Is javascript static type language or dynamic type language?**

JavaScript is a dynamic type language. This means that variables in JavaScript can hold values of any data ype,

In javascript variable is checked during the run time , so Unlike static type languages, such as Java or C++, c , JavaScript does not require explicit type declarations for variable like char, int , float , double etc .

This make javascript more easier , faster and modified it on the fly .

**4).** **Difference between var and let and const keyword in javascript?**

Var is functional and global scope. Var introduced in es5 where let and const introduced in ES6.

the variable specified using the var keywod is accessible outside the scope or inside the scope or anywhere in the function. but outside the function, it is not accessible. and data can be modified later in var keyword.

Ex1- for(var i=0; i<=5 , i++) {

Console.log(x)

}

Output: 0, 1,2,3,4,5

Ex2- for(var i=0; i<=5 , i++) {

}

Console.log(x)

Output: 0, 1,2,3,4,5 (note: outside the scope its also give same value)

Ex3: function AB() {

Console.log(“hello world”)

}

AB();

Output: hello world

Ex4: function AB() {

}

Console.log(“hello world”)

AB();

Output: Undefined (outside the function its not accessible)

Whereas, the **let** keyword is a block scope Variables specified using the let keyword is accessible inside the sope , outside the scope its not accessible.

Ex1- for(var i=0; i<=5 , i++) {

Console.log(x)

}

Output: 0, 1,2,3,4,5

Ex2- for(var i=0; i<=5 , i++) {

}

Console.log(x)

Output: undefined (note: outside the scope its not accesible)

**Const** keyword is also a block scope , Variables declared specified with const keyword must be assigned a value at the time of declaration and after that it cannot be reassigned or modified to a different value. This means that once you assign a value to a const variable, you cannot change that value later in the code coz it give error. It does not allow redeclaration or reassignment.

However, when using const with objects or arrays, the variables are still mutable,.

**5)What is NaN property in JavaScript**?

NaN property represents the “Not-a-Number” value. It indicates a value that is not a legal number.

To check if a value is NaN, we use the isNaN() function, isNaN() function converts the given value to a Number type, and then equates to NaN.

isNaN("Hello") // Returns true

isNaN(345) // Returns false

isNaN('1') // Returns false, since '1' is converted to Number type which results in 0 ( a number)

isNaN(true) // Returns false, since true converted to Number type results in 1 ( a number)

isNaN(false) // Returns false

isNaN(undefined) // Returns true

**6) Explain HIGHER ORDER FUNCTION ?**

A higher-order function is a function that can accept other functions (its one or more then one) as a arguments, and return functions as results, or both. In other words, it treats functions as first-class citizens. This concept enables you to write more abstract, flexible, and reusable code.

This feature enables powerful functional programming paradigms and allows for more flexible and expressive code.

Some common examples of higher-order functions include map(), filter(), reduce(), and forEach(),

which are built-in array methods in JavaScript. example, map() is a higher-order function because it takes a function as its argument. The function passed to map() is applied to each element of the numbers array, and the result is a new array doubledNumbers.

**7) Explain the “this” keyword..?**

In Javascript, this keyword refers to the object its belong to or the owner object. It is a special variable that holds a reference to the object that is currently being operated on or the object that the function belongs to.

Alone, this refers to the global object. (ex- window)

In a function, this refers to the global object

In a function, in strict mode, this is undefined.

Ex – thiskeyword.js file

**9)What is DOM?**

DOM stands for Document Object Model. IT is a programming interface for HTML and XML documents.

When the browser tries to render an HTML document into a web page, the browser creates a dom object based on the HTML document which basically look like the tree structure of an HTML document.

Using this DOM, we can manipulate or change various elements inside the HTML document easily.

**10)What is BOM ?**

In JavaScript, BOM stands for Browser Object Model. It is a set of objects provided by web browsers to interact with the browser window.

The BOM allows JavaScript code to access and manipulate various aspects of the browser, such as the window, document, location, history, and navigator objects.

Here are some commonly used objects in the BOM:

Window: The top-level object in the BOM hierarchy(list), representing the browser window, It provides properties and methods to control and interact with the browser window, such as opening new windows, resizing browser screen size, moving, and closing windows.

navigator: Provides information about the client's browser and OPERATING system, like the browser name, version, and user-agent string.

screen: Gives details about the client's screen, such as width, height, and color depth.

location: Represents the current URL of the web page and allows navigation to different URLs.

history: Allows interaction with the browser's history, enabling forward and backward navigation.

document: It represents the web page loaded in the browser window and provides methods and properties to manipulate the content and structure of the page.

**19)Explain call(), apply() and, bind() methods**.?

Certainly! The call(), apply(), and bind() methods are all built-in methods in JavaScript that allow you to manipulate the execution context of a function.

call() method allows an object to use the method (function) of another object. the call() method is used to invoke a function with a specified ‘this’ keyword value.its takes a arguments individually separated by comma .

The apply method is similar to the call() method. The only difference is that, the call() method takes arguments separately.whereas, apply() method takes arguments as an array form.

bind(): the bind method make a copy of the object function and when later needed then called or use it. Bind method returns a new function, where the value of the “this” keyword will be bound to the owner object, which is provided as a parameter.

**17)What is Callback function in javascript ?**

a callback function is a function that is passed as an argument to another function. And it will be executed after outer function gets executed.

In other word ,A callback function in JavaScript is a function that is passed as an argument to another function and is executed after the parent function has completed its task .

The purpose of a callback function is to allow the calling function to perform additional task or actions or make decisions based on the result or state of the callback function.

Callbacks function are a way to achieve asynchronous programming and handle actions that may take some time to complete, such as fetching data from a server, reading a file, or performing animations.

**20) What is the Constructor function in JavaScript?**

In JavaScript, a construction function is a special type of function that is used to create and initialize objects. When we have to create a multiple objects with similar properties and methods then we used a constructor

Inside the constructor function, we can use the this keyword to refer to the current instance of the object being created.

these constructor functions can be used with the new keyword to create instances of particular objects. known as a constructor

The constructor function is created by using the Function() constructor which is in Pascl notation i.e. the first letter is capital.

**14) what is a Ternary operator ?**

Ternary Operator in Javascript is a short-handed way to write if else conditional statements. It is also known as a conditional operator. If condition 1 is true then expression 1 is executed otherwise expression 2 is executed.

Ex –

let age1 = 17;

let age = (age1>=18) ? “yes he can Vote : “Not applicable for vote ;

console.log(age);

output: Not applicable for vote

**13) Explain type of error in javascript.?**

In JavaScript, errors can occur during the execution of a program when there is something wrong with the code.

These errors can be categorized into several types based on their nature . There are 2 main type of error in javascript –

Syntax Errors: Syntax errors occur when the code violates the rules of the JavaScript language. This can be due to misspelled keywords, when u forgot to give parentheses, open and close square bracket, curely brackets, or semicolons, improper use of operators, etc. These errors prevent the code from being executed, and the browser or runtime environment will display an error message pointing to the line where the syntax issue is found.

Ex-

var x = 10;

if (x > 5) {

console.log("x is greater than 5");

// Missing closing curly brace for the if statemen : Error comes

else {

console.log("x is less than or equal to 5");

}

**Logic Errors:** Logic errors, also known as semantic errors, it occur due to Reasoning mistakes when the syntax is proper but the logic or program is incorrect. the code does not produce the expected output due to incorrect logical reasoning. and in this case The code runs without any error, but the results does not behave as its expected.

Ex-

function calculateArea(width, height) {

return width \* height;

note: Incorrect formula, should be width \* height / 2 for a triangle's area

}

console.log(calculateArea(4, 5));

// Output: 20, but it should be 10 (for triangle's area)

15) **What are the advantage of the javascript?**

JavaScript has several advantages that make it a popular programming language:

Versatility: JavaScript can be used for both front-end and back-end development. It is the primary language for web development, allowing developers to create interactive and dynamic websites. However, if you want to use JavaScript on the backend, you'll need to learn NodeJS. It is currently the only JavaScript framework that may be used on the backend.

Easy to Learn: JavaScript has a relatively simple syntax and is easy language for beginners to learn. Its similarity to other programming languages.

Platform Independence: JavaScript is a cross-platform language, meaning it can run on various operating systems and devices. It is supported by all major web browsers,including Chrome, Firefox, Safari, and Edge. making it highly accessible to a wide audience.

Supports Asynchronous Programming: JavaScript's support for asynchronous programming with features like callbacks, Promises, and async/await allows developers to write non-blocking code, enabling better performance and responsiveness in applications.

Regular Updates or Continuous evolution: JavaScript evolves continuously, with regular updates and new ECMAScript standards. These updates bring modern features, performance improvements, and security enhancements to the language, keeping it relevant and up-to-date. This ensures that developers have access to modern tools and techniques, allowing them to stay up-to-date with the latest trends in web development.

**16)What is the distinction between client-side and server-side JavaScript?**

The distinction between client-side and server-side JavaScript lies in where the code is executed and what tasks it performs. Both types of JavaScript serve different purposes in web development

Client-side JavaScript:

Client-side- Client-side JavaScript refers to JavaScript code that runs on the client-side, typically with in a web browser. It is primarily used to enhance the user experience and interactivity of the web pages. Client-side JavaScript is responsible for adding interactivity, manipulating the DOM (Document Object Model), handling user events, e.g., clicks, form handling, and submissions, and updating the content without requiring a page reload (AJAX) When a user visits a website, the web browser downloads the HTML, CSS, and JavaScript files associated with that page. Common client-side JavaScript frameworks include React, Angular, and Vue.js.

Server-side JavaScript: Server-side JavaScript refers to JavaScript code that runs on the server side, It is used to handle server-side logic, process requests, and generate dynamic web content. Server-side JavaScript is often used in web development to build server applications, APIs, and backend services. It can interact with databases, perform complex computations, handle authentication and authorization, and communicate with other services. Popular server-side JavaScript platforms include Node.js, Express.js, and Nest.

17**)What is currying in JavaScript?**

Currying is a technique in functional programming that allows you to transform a function which take multiple arguments at once into a sequence of functions, now each function taking a single argument.

In other words, instead of passing all the arguments to a function at once, currying allows you to pass them one by one, creating a like chain of functions until all the arguments are provided, and the final result is returned.

Ex – currying.js

**18)what is closure ?**

Closure function is a inner function that has access to the outer function variable even after the function has finished executing.

It is the combination of function that bundled together that’s why it is also called nested function

In other word we can also say that its preserve the data from outside.