4) javascript description? oops concepts explain

JavaScript is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted programming language that enables dynamic interactivity on websites when applied to an HTML document.

oops concepts

1) object: is a real time entity. Objects are the collection of key/value pairs that can be modified throughout the lifecycle of an object as similar to dictionary.

2) Class: collection of objects. Classes are used to define the blueprint for real-world object modeling and organize the code into reusable and logical parts.

3) inheritance: parent class inherit to child class.

4) polymorphism: one task performed different ways.

5) abstraction: hiding internal details and showing functaionality.

6) encapsulation: wrapping code and data together into a single unit.

8) what are the types of inheritance? Explain?

1. Prototypal Inheritance: Prototypal inheritance is a type of inheritance that adds new properties and methods to an existing object. This inheritance makes use of prototype objects i.e. object.prototype

2. Pseudoclassical Inheritance: Psuedoclassical inheritance is similar to prototypal inheritance. This type of inheritance emulates classical inheritance using prototypal inheritance. In psuedoclassical inheritance, we try to create a class with a function that is intended to be called with the new keyword.

3. Functional Inheritance: Functional inheritance involves inheriting features with the use of an augmenting function to an object instance:

14 **Export:** export a variable using the **export** keyword in front of that variable declaration. Statements allow the user to export his created objects and methods to other programs. For instance, if you assign a string literal then it will expose that string literal as a module. , it gets sorted to the bottom of the file. It can't be called conditionally, it always run in the beginning of the file. To run a program containing export statement you have to use experimental module feature flag.

Syntax for variable: export let variable_name;

Syntax for function: export function function_name() { // Statements }

Syntax for class: export class Class_Name { constructor() { // Statements } }

15 **Import:** You can import a variable using **import** keyword. You can specify one of all the members that you want to import from a JavaScript file.

1) how to applay javascript file in html file

To include an external JavaScript file, we can **use the script tag with the attribute src** . The value for the src attribute should be the path to your JavaScript file Scripts can be placed in the <body> or in the <head> section of an ***HTML*** page

2) Define closure. When a variable which is defined outside the scope in reference is accessed from some inner scope.

3) What is the difference between == and === and NAN?

The == operator checks equality only Compares two operands returns *true* if operands have the same data type and same value, returns *false* if the values differ. Also known as *loose equality* whereas === checks equality, and data type, i.e., a value must be of the same type. Compares two operands returns *true* only if operands are of same data type and same value, otherwise returns *false* Also known as *strict equality* NAN: NaN stands for Not a Number. It represents a value which is not a valid number. It can be used to check whether a number entered is a valid number or not a number NaN is Not-A-Number it is a variable in global scope. NaN is a number that is not a legal number.

5) hosting in javacript explain?

Hoisting is the default behavior of moving all the declarations at the top of the scope before code execution. Basically, it gives us an advantage that no matter where functions and variables are declared, they are moved to the top of their scope regardless of whether their scope is global or local

6) local and Global variable?

Local Variables - local variables are declared inside the functions**.** They have local scope, which means that they can only be used within the functions that define them.

Global Variables − global variables are declared outside the functions A global variable has a global scope which means it can be defined anywhere in your JavaScript code.

7 Constructor: is a special function that creates and initialized an object instance of a class. I JavaScript a constructor gets called when an object is created using the new keyword the purpose of constructor is to create a new object and set values for any existing object properties.

Types of constructor: Array and objects

9 Spread Operator: ES6 introduced a new operator referred to as a spread operator, which consists of three dots (...). It allows an iterable to expand in places where more than zero arguments are expected. It gives us the privilege to obtain the parameters from an array. it takes in an iterable (e.g an array) and expands it into individual elements. **syntax:**

var variablename1 = [...value];

Import is lexical, it gets sorted to the top of the file. It can't be called conditionally, it always run in the beginning of the file. To run a program containing import statement you have to use experimental module feature flag.

Statements are used to refer to an ES module. Other file types can't be imported with these statements. They are permitted only in ES modules and the specifier of this statement can either be a URL-style relative path or a package name.

Syntax: import member_to_import from "path_to_js_file";

17 API Methodes

Axios is a promise based HTTP client for the browser and Node.js. Axios makes it easy to send asynchronous HTTP requests to REST endpoints and perform CRUD operations. It can be used in plain JavaScript or React.

1. **Axios .GET** : The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.
2. **Axios .POST** : A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.
3. **PUT** : PUT is used to send data to a server to create/update a resource. Replaces all the current representations of the target resource with the uploaded content.
4. **PATCH** : PATCH is used to update partial resources. For instance, when you only need to update one field of the resource, Putting a complete resource representation might be utilizes more bandwidth
5. **HEAD** : HEAD is almost identical to GET, but without the response body. HEAD transfers the status line and the header section only.
6. **DELETE** : The DELETE method deletes the specified resource.

18 Array destructuring

Destructuring is a JavaScript expression that makes it possible to unpack values from arrays, or properties from objects, into distinct variables

Array destructuring extracts values from an array Destructuring means to break down a complex structure into simpler parts. With the syntax of destructuring, you can extract smaller fragments from objects and arrays. It can be used for assignments and declaration of a variable. Destructuring is an efficient way to extract multiple values from data that is stored in arrays or objects. When destructuring an array, we use their positions (or index) in an assignment.

object destructuring

10 Rest Operator: The rest operator allows us to represent an indefinite number of arguments as an array. By using the rest parameter, a function can be called with any number of arguments. The rest parameter is prefixed with three dots (...). Although the syntax of the rest parameter is similar to the spread operator, it is entirely opposite from the spread operator. The rest parameter has to be the last argument because it is used to collect all of the remaining elements into an array. Syntax: function fun(a, b, ...theArgs) { // statements }

11 Arrow Function Arrow function a basically anonymous function which accept a fixed number or arguments. **Anonymous function expression written with the "fat arrow" syntax (=>)**. Like traditional function expressions, arrow functions are not hoisted, and so you cannot call them before you declare them.

12 callback function

A callback function is a function passed into another function as an argument, which is then invoked inside the outer function to complete some kind of routine or action. a function which is to be executed after another function has finished execution.

19 ForEach The forEach method allows you to execute a function by iterating through each element of an array. forEach method doesn't return anything, and thus, if you try to get the return value of the forEach method, you will get undefined. Instead, it allows you to modify the values of an existing array by applying the callback function on each element of an array. it allows you to execute a function for each element of an array.

20 MAP() map method creates a new array using the return values of this function. map creates a new array by applying the callback function on each element of the source array. Since map doesn't change the source array, we can say that it's an immutable method. ES6 provides us a new collection type called Map, which holds the key-value pairs in which values of any type can be used as either keys or values. A Map object always remembers the actual insertion order of the keys. Keys and values in a Map object may be primitive or objects. It returns the new or empty Map.Maps are ordered, so they traverse the elements in their insertion order.

21 Filter() filter creates a new array by removing elements that don't belong. The filter() method is used to filters all the elements. The filter() method returns the element that matches and removes the element that does not match. The filter() method search through all the elements. The filter() method does not change the original array. The filter() method does not execute the function for empty elements. In filter() method a value is passed to the function as its this value Its syntax is -:array.filter(function(value, Index, array), thisValue)

When destructuring the objects, we use keys as the name of the variable. The variable name must match the property (or keys) name of the object. If it does not match, then it receives an **undefined** value. This is how JavaScript knows which property of the object we want to assign. In object destructuring, the values are extracted by the keys instead of position (or index). It is similar to array destructuring except that instead of values being pulled out of an array, the properties (or keys) and their corresponding values can be pulled out from an object. object destructuring extracts values from a JavaScript object.

Promise

Promises are used to handle asynchronous operations in JavaScript. They are easy to manage when dealing with multiple asynchronous operations where callbacks can create callback hell leading to unmanageable code. A promises is an object that keep track about whether a certain event has happened already or not. Determines what happen after the events has happened.
Promise has 3 states – resolved, rejected and pending. Error handling is done using .then() and .catch() methods. Promise is an object representing intermediate state of operation which is guaranteed to complete its execution at some point in future.

Async/Await

Async/wait it is work with promises in a more comfortable fashion, called async its is easy to understand and use the word async before a function means one simple thing a function always returns a promise. So the async keyword is added to functions to tell them to return a promise rather than directly returning the value. We can use wait when calling a function that returns a promise including web API functions. The keyword await makes JavaScript wait until that promise settles and returns its result.
It does not have any states. It returns a promise either resolved or rejected.
Error handling is done using .try() and .catch() methods. Async/Await is a syntactic sugar for promises, a wrapper making the code execute more synchronously.

21) What are the different data types present in JavaScript?
There are two types of data types in JavaScript:
Primitive data types
Non- Primitive data types
Primitive data types
1 String: sequence of characters. It is written within quotes
2 Number: numeric values and can be written with or without decimals.
3) Boolean: Boolean value, true or false.
4) BigInt: The BigInt data type is used to store numbers beyond the Number data type limitation. This data type can

find() The find() method is used to find all the descendant elements of the selected element. The find() method finds the element in the DOM tree by traversing through the root to the leaf. The find() method search through all the child elements only. It does not execute the function for empty elements. It does not change the original array. This method returns undefined if no elements are found. Its syntax is -: array.find(function(value, Index, array),thisValue)

Reduce() Reduce takes all of the elements in an array and reduces them into a single value. Just like map and filter reduce is defined on Array. prototype and so is available on any array, and you pass a callback as its first argument. Syntax: reduce((previousValue, currentValue, currentIndex, array) => { /* ... */ })

Types of Conditional Statements

1 if statement 2 if….else statement 3 if….else if….statement
4 nested if statement 5 switch statement
1 The if statement: If the condition evaluates to true, the code within **if statement** will execute, but if the condition evaluates to false, then the code after the end of **if statement (after the closing of curly braces)** will execute.
2 The if….else statement If the condition is true, then the statements inside **if block** will be executed, but if the condition is false, then the statements of the **else block** will be executed.
3 The if….else if…..else statement It is used to test multiple conditions. The **if statement** can have multiple or zero **else if statements** and they must be used before using the **else statement.** You should always be kept in mind that the **else statement** must come after the **else if statements.**
4 The nested if statement It is an if statement inside an if statement.
5 The switch statement it is mainly used when all branches depend upon the value of a single variable. It executes a block of code depending upon the different cases. The **switch statement** uses the **break** or **default** keywords, **break:** It is used within the **switch statement** for terminating the sequence of a statement. **default:** It specifies some code to run when there is no case match. There can be only a single default keyword in a switch

23 Array Methods ES6:

concat() every() filter() find() forEach() Array.from() shift()

store large integers and is represented by adding "n" to an integer literal. 5) **Undefined:** variable is declared but not assigned. The value of this data type is undefined,. 6) **Null:** invalid value i.e. no value 7) **Symbol:** Symbol is a new data type introduced in the ES6 version of JavaScript. It is used to store an anonymous and unique value. 8) **typeof:** The typeof operator is used to determine what type of data a variable or operand contains. **Non-Primitive data types** **Object:** it is a real world entity **Array:** group of similar values. **RegExp** represents regular expression	indexof() join() map() pop() push() reduce() sort() splice() unshift() **24JavaScript String Methods** **charAt() charCodeAt() concat() match() toString(): indexOf(): lastIndexOf(): replace(): search(): valueOf(): slice(): split(): substr() substring(): toLocaleLowerCase(): toLowerCase(): toUpperCase(): trim():

16) Differences between var, let, and const

Var **let** **Const**

The scope of a *var* keyword is the global or function scope. It means variables defined outside the function can be accessed globally and variables defined inside a particular function can be accessed within the function.	The scope of a *let* variable is block scope. It can be accessible outside the particular block.	The scope of a *const* variable is block scope. When users declare a const variable they need to initialize it otherwise it returns an error. The user cannot update the const variable once it is declared.
It can be updated and re-declared into the scope.	It can be updated but cannot be re-declared into the scope.	It cannot be updated or re-declared into the scope.
It can be declared without initialization	It can be declared without initialization.	It cannot be declared without initialization.
It can be accessed without initialization as its default value is "undefined".	It cannot be accessed without initialization, as it returns an error.	It cannot be accessed without initialization, as it cannot be declared without initialization.

for loop: The for loop is used to iterate a part of the program multiple times. If you have a fixed number of iterations, Initialization: It is the initial condition that is executed once at the starting of the loop. In this part, we initialize the variable, or it can also be used for the already initialized variable. It is an optional statement. Condition: It is executed each time for testing the condition of the loop. It continues the execution of the loop until the condition is false. It returns only Boolean values that are either true or false. It is also an optional statement. Increment/Decrement: It can increment or decrement the value of the variable, and it is also an optional statement. Statement: It represents the body of the loop, which is executed every time until the conditional expression is false.

setTimeout allows us to run a function once after the interval of time.

setInterval allows us to run a function repeatedly, starting after the interval of time, then repeating continuously at that interval.

What is an API?

Application Programming Interface. It is the software responsible for the connection for the communication and information exchange between two apps. API connects two devices or programs in order to facilitate the exchange of information between them. Examople: When you use an application on your mobile phone, the application connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to your phone. The application then interprets that data and presents you with the information you wanted in a readable way.

What is CSS grid system?

grid is **a set of intersecting horizontal and vertical lines defining columns and rows**. A grid system in graphic design **uses a two-dimensional framework to align and lay out design elements**. Breaking down a single design space into a grid can help position individual components in ways that can catch the eye

Reduced file sizes.

Speed of development

Responsiveness.

For Loop	for Each Loop
It is one of the original ways of iterating over an array.	It is a newer way with lesser code to iterate over an array.
It is faster in performance	It is slower performance
The break statement can be used to come out from the loop	The break statement cannot be used because of the callback function.
The parameters are the iterator, counter, and incrementor	The parameters are the iterator, index of item, and array to iterate.
It works with the await keyword.	The await keyword cannot be used due to the callback function. It may lead to incorrect output.

25 for..in	for..of
The JavaScript for in statement loops through the enumerable properties of an object. The loop will iterate over all enumerable properties of the object itself and those the object inherits from its constructor's prototype.	loop over the data structures that are inerrable such as Arrays, Strings, Maps, Node Lists, and more. It calls a custom iteration hook with instructions to execute on the value of each property of the object

22 While Loop	Do While Loop
1 A while loop evaluates the condition if the condition evaluates to true the code inside the while loop is executed. The condition is evaluated again this process continues until the condition is false when the condition evaluates to false the loop stop 2while loop is entry control loop	1 The body of the loop is executed at first then the condition is evaluated. If the condition evaluates to true the body of the loop inside the do statement is executed again. The condition is evaluated once again if the condition evaluates to true the body of the loop inside the do statement is executed again this process continues until the condition evaluates to false then the loop stops. 2 Do loop Exit control loop
3 while loop is a pre-test loop	3 Do loop is a post-test loop
4 Ther is no semicolon at the end of a While loop	4 Do loop the semicolon is compulsory at the end of the loop
5 While loop there is no condition at the end of the loop	5 Do loop there is a condition at the end of the loop
Ex: while(condition){ statement; body of the loop }	**Ex:** Do { statement; body of the loop } while(condition);