1. What is Javascript ?

=> JavaScript is a scripting or programming language that allows you to implement complex features on web pages ,It enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else. It is a client-side language, which means that it is executed by a user's web browser rather than on a web server. It is used to add interactivity to web pages, allowing users to interact with and manipulate elements on the page without needing to reload the page. JavaScript is also used on the server-side, using platforms such as Node.js, to create web servers and build back-end systems.

2. let const var difference in Javascript ?

=> let, const, and var are all keywords used in JavaScript to declare variables. var was the original keyword used to declare variables in JavaScript, and is still supported in modern versions of the language .

Variables declared with var are function-scoped, which means that they are only accessible within the function in which they are declared. Variables declared with let and const are block-scoped, which means that they are only accessible within the block in which they are declared.

3.What is closure in Javascript ?

=> In JavaScript, a closure is created when a function is defined inside another function and the inner function refers to a variable from the outer function. The inner function has access to the outer function's variables and parameters, even after the outer function has returned. The inner function maintains a reference to the outer function's scope, which is called a closure.

4. What are higher order  functions(map , filter , reduce , find , findIndex, foreach).

=> Higher-order functions are functions that take other functions as arguments, or that return functions as values

-> **map**: The map() function creates a new array by calling a function on each element in the original array. It returns a new array with the same length as the original array, but with each element transformed by the function.

-> **filter**: The filter() function creates a new array by calling a function on each element in the original array. It returns a new array containing only the elements that pass the test in the function.

-> **reduce**: The reduce() function reduces an array to a single value by calling a function on each element in the array. It returns a single value that is the result of the function.

->**find**: The find() function returns the first element in an array that passes a test in a given function.

**->findIndex**: The findIndex() function returns the index of the first element in an array that passes a test in a given function.

->**forEach**: The forEach() function executes a function on each element in an array.And It return Undefined.

5. What is arrow functions in Javascript ?

=> Arrow functions are a useful tool for writing more concise and expressive code in JavaScript, and they are commonly used in modern JavaScript development.Arrow functions are a shorthand syntax for creating functions in JavaScript. They were introduced in ES6 (ECMAScript 2015) as a way to write more concise and expressive functions.

6. Difference  in traditional fun and arrow functions ?

=>Arrow functions have a few key features: They have a concise syntax that is shorter than traditional function expressions. They do not have their own this keyword. Instead, they inherit the this value from the surrounding context. They cannot be used as constructors, meaning they cannot be called with the new keyword to create new objects.

7. All the js events and how they work

=>JavaScript supports a wide range of events that can be used to respond to user actions and other events in a web page.

**Click**: This event is triggered when a user clicks on an HTML element. It can be used to respond to user input, such as triggering an action when a button is clicked.

**Mouseover**: This event is triggered when the user moves the mouse over an HTML element. It can be used to display additional information or trigger animations.

8. all the array methods (sort , slice , splice ,concat , max , min )

**sort**(): This method is used to sort the elements of an array. By default, it sorts the elements in ascending order, but you can pass a function as an argument to sort the elements in any desired order.

**slice**(): This method is used to create a new array that contains a portion of the original array. It takes two arguments: the start index and the end index (which is not included in the resulting array).

**concat**(): This method is used to combine two or more arrays into a single array.

**max**(): This method is not a built-in method of JavaScript arrays, but you can use the Math.max() function in combination with the spread operator to find the maximum element in an array.

**min**(): Similarly, you can use the Math.min() function to find the minimum element in an array.

9. mutable and immutable

=>In programming, mutable and immutable are terms used to describe whether or not an object or data structure can be modified after it is created. An object or data structure that is mutable can be changed or modified after it is created, while an object or data structure that is immutable cannot be changed or modified after it is created.

10. primitive and non-primitive

A primitive data type is a data type that represents a single, simple value. There are six primitive data types in JavaScript:

- Number: A number data type represents numeric values, such as integers and floating-point numbers.
- String: A string data type represents textual data, such as a sequence of characters.
- Boolean: A boolean data type represents a logical value, either true or false.
- Undefined: An undefined data type represents a variable that has been declared but not assigned a value.
- Null: A null data type represents an intentional absence of any object value.
- Symbol: A symbol data type represents a unique identifier that can be used as the key of an object property.

A non-primitive data type is a data type that represents a complex value made up of multiple values, such as an object or an array.

- Object: An object data type represents a collection of key-value pairs, where the keys are strings and the values can be any data type, including other objects.
- Array: An array data type represents a collection of values stored in a sequential order.
- Function: A function data type represents a reusable block of code that can be executed by calling the function.

11. What is array , object ?

Arrays in JavaScript are defined using square brackets, with each element separated by a comma

Arrays have a number of built-in methods that allow you to add, remove, or modify elements in the array, as well as methods for sorting, filtering, and iterating over the array.

Objects in JavaScript are defined using curly braces, with each key-value pair separated by a comma. Here is an example of an object:

12. What is variable hoisting  in JS ?

Variable hoisting is a feature in JavaScript that allows variables to be declared after they are used in the code. In other words, variables can be accessed before they are declared. This happens because JavaScript moves all variable declarations to the top of the current scope, before the code is executed. This process is called "hoisting".

Ex- x = 10;

console.log(x);

var x;

13. Object method in Javascript .

In JavaScript, an object is a data structure that contains a collection of key-value pairs, where the keys are strings and the values can be any data type, including other objects. Objects can also have methods, which are functions that are associated with the object and can operate on its properties. To define a method on an object, you can simply add a function as a property of the object.

Methods can be used to perform operations on an object's properties, and are a powerful tool for organizing and manipulating data in JavaScript.

14. deep copy and shallow copy

A deep copy, on the other hand, creates a new object with new values that are copies of the original object's properties. This means that if you modify a property of the new object, the original object will not be modified. Deep copy can be done using the JSON.parse() and JSON.stringify() methods, or by using a custom function.

A shallow copy creates a new object that references the original object's properties. In other words, the new object contains references to the same values as the original object. This means that if you modify a property of the new object, the original object will also be modified. Shallow copy can be done using the Object.assign() method or the spread operator.

15. callback function in Javascript ?

In JavaScript, a callback function is a function that is passed as an argument to another function, and is executed by that function when a specific event occurs or when a certain condition is met.

Callbacks are a powerful tool in JavaScript for handling asynchronous operations and for enabling code to be executed in response to specific events or conditions. They are commonly used in modern web development, particularly in libraries and frameworks that handle complex operations such as animations, network requests, and user interactions.

16. promises in Javascript ?

In JavaScript, a promise is an object that represents the eventual completion (or failure) of an asynchronous operation, and can be used to handle the result or error when it becomes available. Promises are used to simplify the handling of asynchronous operations, and are a core feature of modern JavaScript.