1. **Explain the difference between var, let, and const in JavaScript.**

. first lets talk about var variable so when js run their code var variable are Hoisted in the top of their global and functional scope .this means that we can acess before the declaration line . although the value of var will be undefined until we assigned some value in it.

The now let make a move to let and const so let and const are not stored in a global space they are stored In a special memory called [der] declarative environment record so we cannot acces before declaration and both let and const have a tempralal dead zone the time period between the variable is hoisted until variable assign some value init when we try to access the var of let and const in a temporalal dead zone it will give ref error

And in const we cannot reassigned value when we create a variable with const

1. **What are JavaScript data types? List them with examples.**
   * 1. Primitive Data Types [immutable their values cannot be changed once created]
        1. Numbers
        2. Strings
        3. Boolean
        4. Undefined
        5. null
     2. Non-Primitive [can store collection of values and they are mutable]
        1. Object
        2. Array
        3. Functions
2. **What is a closure in JavaScript? Provide an example.**

Closure is when the inner function is access the variables of outer function Even when the outer function is complete execution then also the inner function remember the outer function scope.

Most important usecase is data hiding and encapsulation

1. **What is the difference between == and === in JavaScript?**
   * 1. == is used to compare only the value not a data type for example if we compare 2 ==”2” it will give true coz it will only check value
     2. === is more strict then double equal because it will check value itself also the data type of that value example 2===”2” will give false.
2. **What is the DOM and how can you manipulate it using JavaScript?**

The DOM is a hierarchical representation of the web page structure, where each element, attribute, and piece of text is represented as a node in a tree. It allows programs to interact with the content and structure of the page.

1. **Explain the difference between synchronous and asynchronous programming in JavaScript.**
   * 1. In **synchronous** programming in javascript the code will run in order and one by one the next operation will start when first operation will complete.
     2. In **asynchronous** programming in javascript the code will not wait for to complete the execution.operation will run in the background and the next operation will also start parallely.
        1. Example set timeout when we delay a set timeout the code below this function will not stop.the set timeout will go in a callback queue and after the timer is completed the settimeout function will go in call stack and it will execute.
2. **What are higher-order functions in JavaScript? Give an example.**

A function that’s take a function as an argument is called higher-order functions.

Example:

function a(fun) {

  fun();

}

function sayHi() {

  console.log("hello from function b");

}

a(sayHi);

1. **What are JavaScript Promises and how do they work?**

So before promises developers are using callback function to perform asynchronous operation in javascript that leads to inversion of control means we do not have controll over the code also it leads to callback hell.

But promises make it easier so promise basically is an object that represent the eventual completion or failure of an asyncronous operation.

Promise have 3 states pending , fulfilled or rejected

Promise works by chaining .then() , .catch() and .finally() so the() is used handle the promise when it is fulfilled

Catch() is used to handle when promise is rejected

Finally() is executed when promise are settled whether it will be fullfiled or rejected.

1. **Explain what the event loop is in JavaScript.**

The job of event loop is it continuous watch the call stack to check weather it is empty as soon as it founds empty call stack. The callback which are present in micro task queue will be pooped out and pushed it into call stack and when micro task queue is empty it will check in callback queue and popped all the callback from callback queue and pushed it into call stack.

1. **What is scope in js?**

Scope in js where you can acccess your variable and functions in your code.

* 1. Global scope

Where you can define your function and variable outside functions or block it can be accessible anywhere

* 1. Local scope

Variabale that can be declared inside the function are called local scope that cannot be accessed outside the function

* 1. Functional scope

Var is functional scope meaning they are availabe throughout the entire function even before they are declared

* 1. Block scope variable declared with let and const are block scope that can only be available inside a block {} like in if statement and for loop etc…

1. **What is event bubbling and event capturing?** 
   * 1. Event Bubbling

When a event happenes In an element first it triggers the element events and then it will move up to its parent element all the way up to the top of the page imagine a bubble rising through layers starting from innermost element to outermost element.

* + 1. Event Capturing.

Event capturing is oppsitie to event bubbling. the event starts from outermost element and works its way down to its targeted element where the event is happened.

For example: check gtihub commit

1. **What is event propogation?**

Event propagation in JavaScript is the process by which an event triggered on an element moves through the DOM in three phases: **capturing**, where the event moves from the top of the DOM down to the target element; **target**, where the event is processed at the target; and **bubbling**, where the event moves back up from the target to the top. You can stop this flow using stopPropagation().

1. **What is event delegation?**

Event delegation is a technique in JavaScript where instead of attaching event listeners to multiple individual elements, you attach a single event listener to a common parent element. This parent element takes advantage of event bubbling to listen for events on its children, allowing you to handle events more efficiently,

1. **What is json.stringyfy?**

JSON.stringyfy is used to store our data in javascript object notation in a object like structure but in a string it is used to store data in database in json format and also used to store our key value pair in our localstorage.

1. **What is json.parse?**

JSON.parse: This method is used to convert a JSON-formatted string into a JavaScript object. It essentially does the reverse of JSON.stringify. When you have data stored as a JSON string (like from localStorage or a server response), JSON.parse allows you to transform that string back into a usable JavaScript object.

1. **What is local storage?**

Stores data that lasts even after you close and reopen the browser. The data is available across all tabs and windows of the same site.

1. **What is Session Storage?**

Stores data only for the duration of a single browser tab or window. When you close that tab or window, the data is gone.

1. **What is Session Storage?** 
   1. This key word is a represent the object that is currently executed the code.
   2. This key word in global space will give the value of global object [in js env it will be window object].
   3. This Keyword value is depend on how the function has been called and the value of this inside the function will depend on strict mode.
   4. Js have a this subsutution in which if this keyword value is undefined or null this keyword will replaced by global object. [only on non-strict mode]
   5. Arrow functions do not have their own this binding. Instead, they inherit this from the surrounding lexical context (the scope in which they are defined).
2. **What callback function in js?**

A callback function is a function passed as an argument to another function and is called later in the execution.

1. **What is spread Operator ?**

The spread operator in JavaScript is written as ... and is used to "spread" the elements of an array, object, or iterable into individual elements or properties.