1. **What is JavaScript? What is the role of JavaScript engine?**

JavaScript is a programming language that is used for converting static web page to interactive and dynamic web pages.

2. **what is scope in JavaScript?**

Scope is a term that refers to the visibility and accessibility of variables, functions, and objects in a program.

JavaScript has three types of scope: global scope, function scope, and block scope.

3**. What is the type of a variable in JS when it is declared without using the var, let or const keyword?**

var is the implicit type of variable when a variable is declared without var, let or const keywords.

**ex:** if (true) {

variable = 10;

}

console.log(variable); // output: 10

**4.what is Hoisting in JS?**

Hoisting is a JS behaviour where function and variable declarations are moved to the top of

their respective scopes during the compilation phase.

|  |  |
| --- | --- |
| **Function Hoisting** | **Variable Hoisting** |
| myFunction (); // Fun Hoisting  function myFunction () {  console.log("Hello!");  } // output: Hello | x=10; // Var Hoisting  console.log(x);  var x; // output: 10 |

**5. what is JSON?**

JSON is a JavaScript Object Notation is a lightweight data interchange format.

JSON consists of key-value pairs.

**6. what are variables? difference btw var, let and const?**

var creates a function scoped variable.

let creates a block scoped variable.

const can be assigned only once, and its value cannot be changed afterwards.

**7. what is the diff btw undefined and null variable??**

|  |  |  |
| --- | --- | --- |
| **Undefined** | **Null** |  |
| Variable is declared but has **not been assigned a value**, it is automatically initialized with undefined. | Null variables are intentionally assigned to **null value.** |  |
| Undefined can be used when you do not have the value right now, but you will get it after some logic or operation | Null can be used, when you sure you do not have any value for the particular variable. |  |
| **Ex code:** let undefinedVarialbe;  console.log(undefinedVarialbe);  //output: undefined | **Ex code:** let nullVarialbe = null;  console.log(nullVarialbe);  //output: null |  |

**8. what is the difference between == and === ?**

**Loose equality (==)** operator compares two values for equality after performing type coercion.

**Strict equality (===)** operator compares two values for equality without performing type coercion.

**9. what is the difference btw Spread and Rest operator in JS?**

**Spread operator** (…) is used to **expand or spread elements** from an iterable (such as an array, string, or object) into individual elements.

copying an array, merging array, and passing multiple argumenta to a function.

|  |
| --- |
| const array = [1,2,3]; console.log (…array); /// output: 1,2,3 |

**Rest operator** is used in function parameters to collect all remaining arguments into an array.

|  |
| --- |
| display (1,2,3,4,5);  function display (first, second, …restArguments) {  console.log (first); /// output: 1 console.log (second); /// output: 2  console.log (remaining); /// output: [3,4,5]  } |

**10. What is the diff map () and forEach () array methods of an Array?**

Map () methods is used when you want to modify each element of an array and create a **new array** with the modified values.

|  |
| --- |
| let arr1 = [1,2,3];  let mapArray = arr1.map((e) => e \* 2);  console.log(mapArrayy); // output: [2, 4, 6] |

**forEach ()** methods is used when you want to perform some operation on each element of an array **without creating a** **new array**.

|  |
| --- |
| let arr1 = [1,2,3];  let mapArray = arr1.forEach((e) => e \* 2);  console.log(mapArrayy); // output: 2, 4, 6 |

**11. What is Array Destructuring in JS?**

Array destructuring allow you to extract elements from an array and assign them to individual variables in a single statement. Its introduced in ES6.

Ex: const fruits = [‘apple’, ‘banana’, ‘orange’]

const [firstFruit, secondFruit, thirdFruit] = fruits; // array destructuring

**12. Named and anonymous functions?**

|  |  |
| --- | --- |
| **Named Function** | **Anonymous Function** |
| Named fun have a name identifier. | Anonymous function do not have a **name identifier** and cannot be referenced directly by name |
| Ex function sum (a, b) {  Return a+b;  };  console.log (add(5, 3))// output: 8 | Ex: console.log (function (a, b) {  return a \* b;  ) (4, 5)); /// output: 20 |

**13. what is function expression in JS?**

A function expression is a way to define a function b assigning it to a variable.

|  |
| --- |
| Anonymous Function Expression  const add = function (a, b) {  return a + b;  };  console.log(add(5, 3));  // output: 8 |

**14. what are Arrow Function in JS? What is it use?**

Arrow functions also known as fat arrow functions, is a Simpler and shorter way for defining function in JavaScript.

Syndex: () => { }

**15. What are Callback Functions? What is it use?**

Callback function is a function that is passed as an argument to another function.

|  |
| --- |
| function display (x, y, operation) {  var result = operation (x, y) ;  console.log (result);  }  Display (10, 5, add); |

**16. What is Higher order function in JS?**

Higher order function takes one or more function as arguments or return a function as a result.

|  |
| --- |
| function hof (func) {  func ():  }  hof (sayHello);  function sayHello () {  console.log (‘Hello!’);  }  //output: Hello |

**17. what is the use of event handling in JS?**

Event handing is the process of responding to user actions in a web page.

Ex: Click event, mouseHover event, keyUp and keyDown, etc.

**18. What are Pure and Impure functions in JS?**

|  |  |
| --- | --- |
| **Pure Function** | **Impure Function** |
| A pure function is a function that always produces the same output for the same input. | An impure function, can produce different outputs for the same input. |
| Pure function cannot modify the state | Impure function can modify the state |
| Ex: function add (a, b) {  return a + b;  }  console.log (add (3, 5)); // output: 8  console.log (add (3, 5)); // output: 8 | let total = 0;  function addToTotal (value) {  total += value;  return total;  }  console.log (addToTotal (5)); //output: 5  console.log (addToTotal (5)); //output: 10 |

**19. What is DOM?**

DOM (Document Object Model) represent the web page as a tree-like structure that allows JavaScript to dynamically access and manipulate the content and structure of a web page.

### 21. What is event bubbling?

JavaScript allows DOM elements to be nested inside each other. In such a case, if the handler of the child is clicked, the handler of the parent will also work as if it were clicked too.

**22. What are the important JavaScript Array Methods?**

* length property –> If you want to know the number of elements in an array, you can use the length property.
* prototype property –> If you want to add new properties and methods, you can use the prototype property.
* reverse method –> You can reverse the order of items in an array using a reverse method.
* sort method –> You can sort the items in an array using sort method.
* pop method –> You can remove the last item of an array using a pop method.
* shift method –> You can remove the first item of an array using shift method.
* push method –> You can add a value as the last item of the array.

**23. What is closure** **in JavaScript?**

A closure is a function having access to the parent scope, even after the parent function has closed.

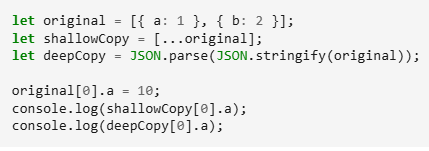
**24. Explain the concept of prototypes in JavaScript?**

Prototypes are a mechanism used by JavaScript objects for inheritance. Every JavaScript object has a prototype, which provides properties and methods that can be accessed by that object.

**25. What is Deep Copy and Shallow Copy?**

A **deep copy** means that all the values of the new variable are copied and **disconnected from the original**variable.

A **shallow copy** means that certain (sub-)values are **still connected** to the original variable.



Out put: 10, 1