1. What is JavaScript?

JavaScript is **a scripting language that enables you to create dynamically content. Javascript is a single-threaded language, meaning that just one line of code may be run at once**

2 What are the features of JavaScript?

It is lightweight and most commonly used as a part of web pages whose implementations allow client-side script to interact with the user and make dynamic pages.

3 What are the different data types in JavaScript?

The five most basic types of data are strings, numbers, booleans, undefined, and null

4 What is the difference between undefined and null in JavaScript?

– Undefined means the variable has been declared, but its value has not been assigned.

– null is an a empty value

5 What are variables in JavaScript and how are they declared?

– var , let , const

– var is a global scope

– let and const is a function scope

6 How do you declare and initialize an array in JavaScript?

Const arrayName = [“hello”, 12, “hiii”];

7 What is a function in JavaScript and how is it defined?

– A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses ();

– example Function hello() { console.log(“hello”) }

8 What is the call method in JavaScript?

– The call() method is a predefined JavaScript method. It can be used to invoke (call) a method.

– Example 1

```
const personName = {
   fullName: function() {
      return this.firstName+" "+this.lastName;
```

```
    }
}

const person1 = {
    firstName:"Balu", lastName:"vemula"
}

const person2 = {
    firstName:"dharan", lastName:"kumar"
}

console.log(personName.fullName.call(person1));
```

– Example 2

```
const personName = {
        fullName: function(a) {
            return this.firstName+" "+this.lastName + a
    + this.shortName;
        }
    }

    const person1 = {
        firstName:"Balu", lastName:"vemula"
    }

    const person2 = {
        firstName:"dharan", lastName:"kumar",
    shortName:”dk”
    }

    console.log(personName.fullName.call(person2,
    “hello”));
```

9 what is apply method in javascript?

– The apply() method calls the specified function with a given this value, and arguments provided as an array

– Example

```
const personName = {
        fullName: function() {
```

```
            return this.firstName+" "+this.lastName;
        }
    }

const person1 = {
        firstName:"Balu", lastName:"vemula"
    }

const person2 = {
        firstName:"dharan", lastName:"kumar"
    }

console.log(personName.fullName.apply(person2));
```

– Example 2

```
const personName = {
        fullName: function(a,b) {
            return this.firstName+" "+this.lastName +" "+
    a +" "+ b +" " + this.shortName;
        }
    }

    const person1 = ["balu", "vemula"];

    const person2 = {
        firstName:"dharan",
    lastName:"kumar",shortName:"dk"
    }

    console.log(personName.fullName.apply(person2,
    ["hello","i am"]));
```

10 What is the difference between call and apply in JavaScript?

– The call() method takes arguments separately.

– The apply() method takes arguments as an array

11 What are closures in JavaScript and how are they used?

– a closure gives you access to an outer function's scope from an inner function (while return a function its complete

function but in closures in inner function still have access outer function while return also)

– unction closureFun(){

```
let b = 10;
console.log(b);
return function innerFun() {
   let innerfun = b+1;
   return innerfun;
}
innerFun();
}
console.log(closureFun());
```

12 What is the difference between synchronous and asynchronous code in JavaScript?

– Asynchronous is a non-blocking architecture, so the execution of one task isn't dependent on another. Tasks can run simultaneously.

– Synchronous is a blocking architecture, so the execution of each operation is dependent on the completion of the one before it.

13 What is callback in JavaScript and how is it used?

– 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.

– example

14 What is event bubbling in JavaScript?

– Event Bubbling is a concept in the DOM (Document Object Model). It happens when an element receives an event, and that event bubbles up (or you can say is transmitted or propagated) to its parent and ancestor elements in the DOM tree until it gets to the root element

– Example

```
<div id=”grandparent”>
    <div id=”parent”>
        <div id=”child”>
```

```
            </div>
        </div>
    </div>

<script>
document.querySelector("#grandparent")
.addeventListener(click,
()=>{console.log("grandparent")});
document.querySelector("#parent").adde
ventListener(click,
()=>{console.log("parent")});
document.querySelector("#child").addev
entListener(click,
()=>{console.log("child")});
</script>
```

15 What is event capturing in JavaScript?

– In event capturing, an event propagates from the outermost element to the target element

– Example

```
<div id="grandparent">
    <div id="parent">
        <div id="child">

        </div>
    </div>
</div>

<script>
document.querySelector("#grandparent")
.addeventListener(click,
()=>{console.log("grandparent")},true)
;
document.querySelector("#parent").adde
ventListener(click,
()=>{console.log("parent")},true);
```

```
document.querySelector("#child").addev
entListener(click,
()=>{console.log("child")},true);
</script>
```

16 What is JSON in JavaScript and how is it used?

– JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax.

17 How do you handle errors in JavaScript?

– JavaScript provides error-handling mechanism to catch runtime errors using try-catch-finally block, similar to other languages like Java or C#.

– try: wrap suspicious code that may throw an error in try block.

– catch: write code to do something in catch block when an error occurs.

18 What is a Promise in JavaScript?

– promise is a object , its a feature in ES6

– in it it has three things

– panding → while create a new promise

– syntex → const a =
new promise((resolve,reject)
=> { });

– fulfilled → when the promise is resolved

– rejected → when the promise is reject

– Example

```
Function funName(){
    const x = new promise((resolve,reject) =>
{ });
    Return x;
}
x.then(()=>{ console.log("when it is resolved")});
x.catch(()=>{console.log("when it is rejected")});
x.finally(()=>{console.log("all time")});
```

19 What are the different types of operators in JavaScript?

– Arithmetic Operators.

- **Assignment Operators.**
- **Comparison Operators.**
- **Logical Operators.**
- **Conditional Operators.**

20 What is the difference between == and === operators in JavaScript?

– (==) is only check its value its not check its type

– (===) is check type also

21 What is the difference between let and var keywords in JavaScript?

– var is a global variable

– let is a block variable

– Variables declared by let are only available inside the block where they're defined. Variables declared by var are available throughout the function in which they're declared

22 What is the difference between const and let keywords in JavaScript?

– const creates "constant" variables that cannot be reassigned another value.

– In the var and let tab, when you run the code, you will see that there is no error and we can define new values to the var and let variables. In the const tab, when you run the code, you will get an error as a const variables value cannot be reassigned.

23 What is hoisting in JavaScript and how does it work?

– Hoisting is a JavaScript mechanism where variables and function declarations are moved to the top of their scope before code execution

```
– z = 10;
console.log(z);
var z;
– a = 10
console.log(a);
let a;
```