Top 20 Interview Questions

JavaScript

1. What Is JavaScript?

JavaScript is the most popular web scripting language, used for both client-side and server-side development. Supporting object-oriented programming abilities, the JavaScript code can be inserted into HTML pages that can be understood and executed by web browsers.

2. What is ECMAScript?

ECMAScript (European Computer Manufacturers Association Script) is a scripting language based on JavaScript.

3. Difference between " == " and " === " operators.

```
0==false // true, because false is equivalent of 0 .
0===false // false, because both operands are of different type.
2=="2" // true, auto type coercion, string converted into number.
2==="2" // false, since both operands are not of same type.
```

4. What are the data types supported by JavaScript?

- Undefined
- Null
- Boolean
- Object

- String
- Symbol
- Number

5. What Is JavaScript Strict Mode?

The default behavior of JavaScript is very forgiving if we make a minor mistake. That means that it will not display any error messages. But sometimes in development, we need to see all kinds of errors and warnings in order to debug the code.

6. What Is a Closure?

A closure in JavaScript is a function inside another function. The inner function has access to its own variables, the variables defined in the outer function as well as the global variables.

```
var a = 2;
function outer()
{
    var b = 4;
    function inner()
    {
        var c = 6;
        console.log(a + b + c);
    }

    inner();
}
outer();
```

7. What Is NaN?

In JavaScript, NaN stands for Not a Number. It is a special value that occurs when we are unable to perform an operation.

For example, what if we try to divide a string using a number (e.g., "Hello World" / 5)?

8. How can you create an object in JavaScript?

```
var emp = {
name: "Daniel",
age: 23
};
```

9. How can you create an Array in JavaScript?

```
var x = [];
var y = [1, 2, 3, 4, 5];
```

10. Define anonymous function?

```
var display=function()
{
  alert("Anonymous Function is invoked");
}
display();
```

11. What is this [[[]]]?

```
var myArray = [[[]]];
```

12. What are the features of JavaScript?

- Lightweight, interpreted programming language
- Cross-platform compatible
- Open-source
- Object-oriented
- Integration with other backend and frontend technologies
- Used especially for the development of network-based applications

13. What are the scopes of a variable in JavaScript?

There are two scopes of a variable:

Global Scope
Global variables, having global scope are available everywhere in a JavaScript code.

Local Scope
Local variables are accessible only within a function in which they are defined.

14. What are the arrow functions in JavaScript?

Arrow functions are a short and concise way of writing functions in JavaScript. The general syntax of an arrow function is as below:

```
const helloWorld = () => {
  console.log("hello world!");
};
```

15. What is the 'this' keyword in JavaScript?

The 'this' keyword in JavaScript refers to the currently calling object. It is commonly used in constructors to assign values to object properties.

16. Difference between var and let keyword.

let	var
let is block-scoped.	var is function scoped.
let does not allow to redeclare variables.	var allows to redeclare variables.
Hoisting does not occur in let.	Hoisting occurs in var.

17. Types of errors in javascript.

Javascript has two types of errors, Syntax error, and Logical error.

18. What is DOM?

DOM is a W3C (World wide web consortium) standard, when the HTML page loads in the browser, the browser creates the DOM (Document object model). It defines the HTML element as an object and allows scripts to dynamically manipulate the content, and the structure of the document.

19. What is the difference between Attributes and Property?

Attributes - provide more details on an element like id, type, value etc.

Property - is the value assigned to the property like type="text", value='Name' etc.

20. What is the difference between Call & Apply?

The call() method calls a function with a given this value and arguments provided individually.

```
fun.call(thisArg[, arg1[, arg2[, ...]]])
```

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

```
fun.apply(thisArg, [argsArray])
```

THANKS FOR READING

FOLLOW FOR MORE.

Jayesh Deshmukh