

1. What is JavaScript?

- a. JavaScript is a client-side and server-side scripting language inserted into HTML pages and understood by web browsers. JavaScript is also Object-Oriented Programming language.
- b. What makes JavaScript a scripting language?
 - i. JavaScript makes browser to do the dirty work. If you command an image to be replaced by another one, JavaScript tells a browser to go to do it. Because browser actually does the work, you only need to pull some strings by writing some relatively easy lines of code. That is what makes JavaScript an easy language to start with.
- c. What makes JavaScript an object - oriented language.
 - i. We should know unlike other Object Oriented Language there is no classes in JavaScript we have only object. To be more precise, JavaScript is a prototype based object oriented language, which means it does not have classes, rather it defines behavior using constructor function and then reuse it using the prototype.

2. Enumerate the differences between Java and JavaScript?

- a. Java is a complete programming language, JavaScript is a coded program that can be introduced to HTML pages. These two languages are not at all inter-dependent and are designed for different intent. Java is an object-oriented programming (OOPS) or structured programming languages like C++ or C, whereas JavaScript is a client-sided scripting language.

3. What are the JavaScript Data types?

a. Number

- i. Number represents a number and can be written with or without decimals

Example :

```
var x = 3; //without decimal
var y = 3.6; //with decimal
```

b. String

- i. String - represents a series of characters and is written with quotes. A string can be represented using a single or double quotes

Example :

```
var str = "Vivek Singh Bisht"; //using double quotes
var str2 = 'John Doe'; //using single quotes
```

c. Boolean

- i. Boolean represents a logical entity and can have only two values: true or false. Booleans are generally used for conditional testing

Example :

```
var a = 2;
var b = 3;
var c = 2;
(a == b) // returns false
(a == c) //returns true
```

d. Undefined

- i. When a variable is declared but not assigned, it has the value of undefined and its types is also undefined

Example :

```
var x; // value of x is undefined
var y = undefined; // we can also set the value of a variable as undefined
```

e. Null

- i. Null represents a non-existent or an invalid value

Example :

```
var z = null;
```

f. BigInt

- i. This data type is used to store numbers which are above the limitation of number data type. It can store large integers and is represented by adding "n" to an integer literal

Example :

```
var bigInteger = 234567890123456789012345678901234567890n;
```

g. Symbol

- i. It is a new data type introduced in the ES6 version of JavaScript. It is used to store an anonymous and unique value.

Example :

```
var symbol1 = Symbol('symbol');
```

h. Type of **primitive types**

```
typeof "John Doe" // Returns "string"
typeof 3.14 // Returns "number"
typeof true // Returns "boolean"
typeof 234567890123456789012345678901234567890n // Returns bigint
typeof undefined // Returns "undefined"
typeof null // Returns "object" (kind of a bug in JavaScript)
typeof Symbol('symbol') // Returns Symbol
```

Non-primitive types

Primitive data types can store only a single value. To store multiple and complex values, non-primitive data types are used. Object - used to store collection of data.

Note: It is important to remember that any data type that is not primitive data type, is of Object type in JavaScript

Example:

```
// Collection of data in key-value pairs

var obj1 = {
  x: 43,
  y: "Hello world!",
  z: function(){
    return this.x;
  }
}

// Collection of data as an ordered list

var array1 = [5, "Hello", true, 4.1];
```

4. What is the use in isNaN function?

- isNaN function returns true if the argument is not a number; otherwise, it is false
- The **isNaN** function is used to check whether a given value is an illegal number or not. It returns true if value is NaN else returns false. It is different from Number.isNaN() Method.

JavaScript

```
<script>
  document.write(isNaN(12) + "<br>");
  document.write(isNaN(0 / 0) + "<br>");
  document.write(isNaN(12.3) + "<br>");
  document.write(isNaN("Geeks") + "<br>");
  document.write(isNaN("13/12/2020") + "<br>");
  document.write(isNaN(-46) + "<br>");
  document.write(isNaN(NaN) + "<br>");
</script>
```

Output:

```
false
true
false
true
true
false
true
```

5. Which is faster between JavaScript and an ASP script?

- JavaScript is faster. JavaScript is a client-side language, but thus, it does not need the assistance of the webserver to execute. On the other hand, ASP is a server-side language and hence is always slower than JavaScript. JavaScript now is also a server-side language (nodejs).

6. What is negative Infinity?
 - a. Negative Infinity is a number in JavaScript which can be derived by dividing negative number by zero.
7. Is it possible to break JavaScript into several lines?
 - a. Breaking within a string statement can be done by using a backslash, '/' at the end of the first line.

Example:

```
document. Write ("This is \a program,");
```

- b. And if you change to a new line when not within a string statement, then JavaScript ignores the break in the line.

Example:

```
var x=1, y=2,  
z=  
x+y;
```

8. Which company first developed JavaScript?
 - a. Netscape is the software company that developed JavaScript
9. What are undeclared and undefined variables?
 - a. Undeclared variables are those that do not exist in a program and are not declared. If the program tries to read the values of an undeclared variable, then a runtime error is encountered.
 - b. Undefined variables are those that are declared in the program but have not been given any value. If the program tries to read the value of an undefined variable, an undefined value is returned.
10. Write the code for adding new elements dynamically?

```
<html>  
<head>  
<title>t1</title>  
<script type="text/javascript">  
    function addNode () { var newP = document. createElement("p");  
        var textNode = document.createTextNode(" This is a new text node");  
        newP.appendChild(textNode); document.getElementById("firstP").appendChild(newP);  
    }  
</script> </head>  
<body> <p id="firstP">firstP<p> </body>  
</html>
```

- New elements can be dynamically created in JavaScript with the help of **createElement() method**. The attributes of the created element can be set using the **setAttribute() method**. The examples given below would demonstrate this approach.

11. What are global variables? How are these variables declared?

- a. Global variables are available throughout the length of the code so that it has no scope. The **var** keyword is used to declare a local variable or object. If the var keyword is omitted, global variable is declared

Example:

```
// Declare a global: globalVariable = "Test";
```

The problem faced by using global variables are the clash of variable names of local and global scope. Also, it is difficult to debug and test the code that relies on global variables.

12. What is a prompt box?

- a. A prompt box is a box that allows the user to enter input by providing a text box. A label and box will be provided to enter the text or number.

13. What is "this" keyword in JavaScript?

- a. "this" keyword refers to the object from where it was called

14. What is the working of timers in JavaScript?

- a. Timers are used to execute a piece of code at a set time or repeat the code in a given interval. This is done by using the functions **setTimeout**, **setInterval**, and **clearInterval**
- b. The **setTime(function, delay)** function is used to start a timer that calls a particular function after the mentioned delay. The **setInterval(function, delay)** function repeatedly executes the given function in the mentioned delay and only halts when canceled. The **clearInterval(id)** function instructs the timer to stop.
- c. Timers are operated within a single thread, and thus events might queue up, waiting to be executed.

15. Which symbol is used for comments in JavaScript?

- a. `//` for single line comments and
- b. `/*` Multi Line Comment `*/`