JavaScript

Documents built using HTML are mostly static in nature .To add interactive to documents ,scripting elements need to be included

Scripting can be of two type

Client side scripting : This is done along with HTML document .

Server side scripting: These are scripts or programs that are executed from server side

Client Side scripting has some benefits

Client side scripts like java script are platform independent and light weight

Processing time in client side scripting is reduced as data need not be sent to server for processing

One of the most popular client side scripting language is JavaScript

JavaScript is easy to learn as most of the symentics are similar to other programming languages

Java script can also be said as default scripting laguage for a browser

Javascript can be placed in three ways over a document

Internal script

External script

Inline script

Internal script is placed inside script tag in head section of HTML document

```
<HTML>

<HEAD>

<SCRIPT>

 alert("java script")

</SCRIPT>

</HEAD>

</HTML>
```

in the above example an alert is kept inside script tag.This script will be loaded along with head tag.Internal script is applicable to document in which it is created

External Script:JavaScript can be created externally and attached to the document

Lets see an example

home.html

```
<HTML>
 <HEAD>
          <SCRIPT SRC="hello.js">
   </SCRIPT>
 </HEAD>
</HTML>
```

hello.js

```
alert("hello world")
```

In the above example an external java script file is created with .js extension.It is attached to html document using a script tag

script tag has a src attribute that has location of external javascript file

Inline javascript

Inline script is kept inside the body tag

```
<HTML>
  <BODY>
  <SCRIPT >
    document.write("Hello from script")
  </SCRIPT >
  </BODY>
</HTML>
```

In the above example script is kept inside body tag.script this way is usually used with event handling

Script tag has some attributes

these are

async :specifies that script can be executed asynchronously

defer:specifies that script should be executed when page has finished parsing

src:specifies src of external script

type :specifies media-type ,application/javascript(default) or application/ecmascript

Media type is the type of data to travel over web ,also known as MIME type (Multipurpose mail extension)

Javascript as Programming language

Javascript programming involves usage of variables,operators ,functions ,decision constructs ,loops and arrays

Lets understand these

'

======================

variable in java script can be declared with a simple var keyword or even without using a keyword

declaring a variable

syntax

[variable name]=[value]

Rules for naming a variable

These are the following rules for naming a JavaScript variable:

A variable name must start with a letter, underscore (_), or dollar sign ($).

A variable name cannot start with a number.

A variable name can only contain alpha-numeric characters (A-z, 0-9) and underscores.

A variable name cannot contain spaces.

A variable name cannot be a JavaScript keyword or a JavaScript reserved word.

diffrent values that can be assigned to a javascript variable and data type for a variable is decided by value assigned to it

Different data types are

Primitive

String

Number

Boolean

Composite

Object

Array

Function

Special

Undefined

Null

Lets see primitive datatypes

Working with String

```
<HTML>

<HEAD>

  <SCRIPT>
```

```
   mystring="Hello World"

   document.write(mystring)
  </SCRIPT>
 </HEAD>
</HTML>
```

the above example creates and displays a string

characters inside string can be accessed from index position

```
<SCRIPT>
   mystring="Hello World"

   document.write(mystring[0])
  </SCRIPT>
```

string has in built functions associated with it

charAt returns character at particular index position

```
<SCRIPT>
```

```
   mystring="Hello World"

   document.write(mystring.charAt(3))
 </SCRIPT>
```

charCodeAt

returns unicode of the character at particular position

```
<SCRIPT>
   mystring="Hello World"

   document.write(mystring.charCodeAt(3))
 </SCRIPT>
```

concat

concatenate two string

```
<SCRIPT>
   mystring="Hello World"
```

```
  document.write(mystring.concat("!Welcome to Javascript"))
 </SCRIPT>
```

endswith

checks if string ends with particular text and returns true or false

```
<SCRIPT>
  mystring="Hello World"

  document.write(mystring.endsWith("ld"))
 </SCRIPT>
```

fromCharCode

converts unicode number to character

```
<SCRIPT>


  document.write(String.fromCharCode(70))
 </SCRIPT>
```

includes

checks if string includes a particular string

```
<SCRIPT>
  mystring="Hello World"

  document.write(mystring.includes("Hello"))
 </SCRIPT>
```

indexOf

searches for a string inside another string and returns index position

```
 <SCRIPT>
  mystring="Hello World"

  document.write(mystring.indexOf("World"))
 </SCRIPT>
```

lastIndexOf

searches for last occurance of a string inside another string

```
<SCRIPT>
  mystring="Hello World"

  document.write(mystring.lastIndexOf("l"))
 </SCRIPT>
```

repeat

makes a new string by copying string as no of times mentioned in function

```
 <SCRIPT>
  mystring="Hello World"

  document.write(mystring.repeat(3))
 </SCRIPT>
```

replace

replace a string

```
 <SCRIPT>
   mystring="Hello World"


   document.write(mystring.replace("World","Peter"))
 </SCRIPT>
```

search

searches for a string inside another string and returns index position from where string starts

```
 <SCRIPT>
   mystring="Hello World"


   document.write(mystring.search("World"));
 </SCRIPT>
```

slice

returns part of a string

```
<SCRIPT>
   mystring="Hello World"
   document.write(mystring.slice(0,7));
```

```
 </SCRIPT>
```

split

splits a string into array based on delimiter

```
 <SCRIPT>
   mystring="Apple,Banana,Mango,Strawberry"
 var arr=mystring.split(",")
   document.write(arr[1]);
 </SCRIPT>
```

startsWith

checks if a string starts with particular string

```
<SCRIPT>
    name="Peter San Joes"
    document.write(name.startsWith("Peter"))
 </SCRIPT>
```

substr

extracts part of a string

```
<SCRIPT>
   name="Peter San Joes"
   document.write(name.substr(1,5))
 </SCRIPT>
```

toLowerCase

converts string to lower case

```
<SCRIPT>
   name="Peter San Joes"
   document.write(name.toLowerCase())
 </SCRIPT>
```

toUpperCase

converts string to upper case

trim

removes addittional space from end of string

```
<SCRIPT>
   name="Peter San Joes  "
   lastname="hendrik"
   document.write(name.trim().concat(lastname))
 </SCRIPT>
```

=================

Working with number

```
<SCRIPT>
   num=100
  document.write(num)
 </SCRIPT>
```

In the above given example a variable has been assigned a value of type number

inbuilt functions for number data type

isFinite

checks if the number is a finite number

```
<SCRIPT>

  document.write(Number.isFinite(10/0))
 </SCRIPT>
```

isInteger()

checks if a number is integer

```
<SCRIPT>

  document.write(Number.isInteger(10))
 </SCRIPT>
```

isNaN

checks if the number NaN

NaN means Not a Number

```
 <SCRIPT>

  document.write(Number.isNaN(0/0))
 </SCRIPT>
```

toExponential

converts a number to exponential notation

```
<SCRIPT>
   n=10
  document.write(n.toExponential(5))
 </SCRIPT>
```

Composite DataTypes

working with Objects

An object in JavaScript looks more like a key value pair ,we can say objects are collection of named values

Lets see an example

```
<SCRIPT>
        order={"ProductName":"Rice","Price":50,"Quantity":"2 KG"}
  document.write("Product Name "+order.ProductName)
  document.write(" Price "+order.Price)
  document.write(" Quantity "+order.Quantity)


   </SCRIPT>
```

Object Properties

An object properties are values associated with it

sytax to access a propery

[object].[property]

Properties can be add,delete,modified and viewed

lets see an example

```
<SCRIPT>
        order={"ProductName":"Rice","Price":50,"Quantity":"2 KG"}
  order.Date="12-3-2020"
document.write(order.ProductName+"<br/>")
document.write(order.Price+"<br/>")
document.write(order.Quantity+"<br/>")
document.write(order.Date+"<br/>")
   </SCRIPT>
```

In the above example an object order is created ,a new property Date is added to the object

An object property can be deleted using a delete keyword

after deletion property can be used without adding it again

Methods of object

Methods inside object are functions stored as properties

```
<SCRIPT>
 blog={"blogname":"",
"blog_content":"",
"blog_creation_date":"",

addBlog:function (blogname,blog_content,blog_creation_date)
{
   this.blogname=blogname
   this.blog_content=blog_content
   this.blog_creation_date=blog_creation_date
  return "<h1>"+this.blogname+"</h1>" +
"<h4>"+this.blog_creation_date+"</h4>"+"<h2>"+this.blog_content+"</h2>
"

 }

}
    title="Working with C#"
    datecreation="12-1-2020"
    content="C# is an object oriented programming language. that has
capabilites to fully implement all features of OOP"

  mynewblog=blog.addBlog(title,content,datecreation)

   document.write(mynewblog)

    </SCRIPT>
```

In the example above a method addBlog is created inside object to add values to blog object variables .A this keyword is used ,this represents owner of current instance ,that is the object that has been currently created

and values passed through this are passed on to variables of current object

Object Accessors

These allow you to created Object accessors also known as computed properties

In the above example ,a property blog_author can be added and accessed using accessors

```
get author()

 {

  return this.blog_author

 },

set author(name)

  {

  this.blog_author=name

 }
```

```
blog.author="A.Sailesh Padmanabhan"
  document.write(blog.author)
```

Constructor

Constructors in javascript can be used to create objects of multiple types

```
<SCRIPT>

function blog(blogname,blog_content,blog_date_creation)
 {
  this.blogname=blogname
  this.blog_content=blog_content
  this.blog_date_creation=blog_date_creation
 }

myblog=new blog("ADO.NET","Working with ADO.NET","12-01-2019")
document.write(myblog.blogname)
</SCRIPT>
```

Java Script functions

Java script functions are created using function keyword

```
function add(a ,b)
 {
 return a+b
 }
```

for execution function has to be called ,and required parameter has to be passes

```
document.write(add(12,23)
```

function may or may not have parameters,this totally depends on requrirement

function may or may not return value ,also based on requirement

function can be passed as parameter to another function

```
function PrintMessage(message)

{

  document.write("<h1>"+message+"</h1>")

 }


function ProcessMessage(fun,data)

{

 fun(data)

 }


ProcessMessage(message,"Hello World")
```

Operators and descion making constructs

there are different operators available in javascript that are common to other programming languages as well

Arithematic Operator		
+	Addition	x+y
-	Sutraction	x-y
*	Multiplicative	x*y
/	Division	x/y
%	Modulus (Reminder)	x%y
**	Exponent	x**y
++,--	increment,decrement	x++,x--

Arithematic Assignment	Example	
+=	x+=y	x=x+y
-=	x-=y	x=x-y
/=	x/=y	x=x/y
%=	x%=y	x=x%y
**=	x**=y	x=x**y

Comparison Operator	
==	equal to
>	greater then
<	less then
>=	greater then equal to
<=	less then equal to
!=	not equal to
?	ternery operator

Logical Operator	
&&	And
||	Or
!	Not

type operator	
Typeof	returns data type of a variable
Instanceof	checks if an object is an instance of a type

Conditional Constructs

JavaScript provides with conditional construts to implement logic in scripts

Different conditional constructs

if else

```
syntax if([condition])
 {
}
 else
{
}
```

Lets see an example

```
<SCRIPT>
num1=10
num2=20
if (num1 > num2 )
  {
  document.write(num1 + " is greater then "+num2);
```

```
 }
 else
{
 document.write(num2 + " is greater then " + num1);
 }

   </SCRIPT>
```

The above example is a simple implementation of if else

Lets see another construct switch case

switch case is used for a menu driven approach

syntax for switch case

```
switch([variable])
{
 case [constant]..
   [statement]
   break

    <SCRIPT>
color="GREEN"
switch(color)
```

```
{
 case  "RED":
                    document.write("STOP")
     break;
 case "GREEN":
                    document.write("GO")
     break;
 case "YELLOW":
                    document.write("WAIT")
     break;
  default:
                    document.write("Lights off")
 }



  </SCRIPT>
```

In the above example ,value of variable color is checked and matching case is executed

the break keyword is used to terminate the case ,else case will fall down and executed other cases

Loop constructs

Loop construtcs are used for creating a iterative block of statements

Java script provides different loop constructs

for loop

while loop

do while loop

foreach

lets see for loop

A for loop executes for a fixed no of turns

syntax

```
for(initializatio;test condition;increment/decrement expression)
 {
 }
```

Lets see an example

```
<SCRIPT>
 document.write("<select> ");

for(i=0;i<10;i++)
```

```
{
  document.write("<option>")
  document.write(i);
  document.write("</option>")
}
document.write("</select>")
   </SCRIPT>
```

the above example creates a dropdown list in document ,by and fills value in drop down list using for loop

Another way to write a for loop is for an array .We can use a for loop to iterate through an array

Lets see an example

```
<SCRIPT>
    num=[12,11,19,23,44,56]
   for(i in num)
 {
 document.write(i+ "  "+num[i]+"<br/>")
}

</SCRIPT>
```

In the above program a for loop is iterating through an array using an in operator ,the variable used in for loop gets the current index position of the array ,the value in array can be accessed using index position

While loop :A while loop executes as long as the provided condition is true

syntax

```
while([condition])
{
 }
```

lets see an example

```
<SCRIPT>
    i=0;
   while(i<10)
 {
  document.write("<input type='file' name='file"+i+"' /><br/>")
  i++;
 }
   </SCRIPT>
```

In the above example a while loop is constructed.it checks the value of variable I .Inside loop file control is getting generated and value of i is getting incremented

do while loop

A do while loop executes loop body first and then checks for condition.So a do while loop will get executed at least once even if condition is false

```
do
 {
  }while([condition])
```

Lets see an example

```
<SCRIPT>
     i=10;
   do
 {
  document.write(i)
 }while(i<10);
```

the above loop will get executed atleast once even if condition i< 10 is false

foreach loop

A foreach loop works for an array .It accepts a function as a parameter and applies function over each value of the array

```
<SCRIPT>
     num=[12,11,19,23,44,56]


    num.forEach(printnumber)
```

```
   function printnumber(i)
 {
 document.write(i+"<br/>")
  }
</SCRIPT>
```

In the above example a foreach function accepts function name

Array

An array in JavaScript is group of values .Array is used to store multiple values in a single variable

syntax for array

[variable]=[values,…]

Lets see an example

```
                        <SCRIPT>
  fruits=["Apple","Banana","Mango","Strawberry","Pappaya"]
   document.write(fruits)
<SCRIPT>
```

In the above example an array fruits is created and printed on browser

Members of an array can be accessed using index position

```
    <SCRIPT>
  fruits=["Apple","Banana","Mango","Strawberry","Pappaya"]
   document.write(fruits[1])
   document.write(fruits[3])

</SCRIPT>
```

Array in Java Script has some in built functions and variables

*assuming fruits to be an array

fruits=["Apple","Banana","Mango","Strawberry","Pappaya"]

Function name	Usage	Example
length	Gets length of an array	n=fruits.length , otuput:5

push	sends a value inside array	fruits=["Apple","Banana","Mango","Strawberry","Pappaya"]
		
		fruits.push("Orange")
		document.write(fruits)
		output:
		Apple,Banana,Mango,Strawberry,Pappaya,Orange

pop	gets a value out of array	fruits=["Apple","Banana","Mango","Strawberry","Pappaya"]
		
		document.write(fruits.pop())
		output:Pappaya

sort	sorts an array	fruits=["Banana","Mango","Apple","Strawberry","Pappaya"]
		
		fruits.sort()
		document.write(fruits)
		output: Apple,Banana,Mango,Pappaya,Strawberry

reverse	reverses an array	fruits=["Banana","Mango","Apple","Strawberry","Pappaya"]
		
		fruits.sort()
		fruits.reverse()
		document.write(fruits)
		output: Strawberry,Pappaya,Mango,Banana,Apple

Event Handling

Events in JavaScript are generated in response to some action by the user ,

like click event ,which is generated when user clicks on mouse

Some of the common events are

onclick :when user clicks on any html element

Lets see an example

```
<HTML>

   <HEAD>

    <SCRIPT>

function clickbuttonaction()

{

alert('button clicked')

 }

</SCRIPT>


  </HEAD>

<body>

    <input type='button' value='click' onclick='clickbuttonaction()'/>

 </body>

<HTML>
```

in the above example a button is created with an onclick function that is pointing to another function clickbuttonaction

onload: when document is loaded

```
<HTML>
```

```
  <HEAD>

    <SCRIPT>

     function loadfunction()

  {

  alert('document loaded')

  }

   </SCRIPT>

  </HEAD>

 <body onload='loadfunction()'>

 </body>

</HTML>
```

In above example function loadfunction is executed on call of onload event mentioned in body tag

onchage:When html element has changed

```
<HTML>

  <HEAD>

    <SCRIPT>

     function viewchange()

 {

  selected=document.getElementById("list").value;

 document.getElementById("mypara").innerHTML="text selected
"+selected
```

```
 }
   </SCRIPT>
  </HEAD>
 <body>

 <select id="list" onchange='viewchange()'>
 <option>
 Apple
 </option>
 <option>
Banana
 </option>
<option>
Orange
</option>
 <option>
Mango
</option>
 </select>
 <p id="mypara">
 </p>
</body>
</HTML>
```

In the example above a dropdown list is created,value from dropdown list is selected and in function viewchage.selected value is shown on browser

viewchange function is called on onchange event.Whenever new value is selected ochange event is triggered

onmouseover

When user moves a mouse over an html element

```
<HTML>

  <HEAD>

    <SCRIPT>

      function highlight()

{

 txt=document.getElementById("mypara").innerHTML


 document.getElementById('mypara').innerHTML="<font color='red'>" +txt
+"</font>"

 }

   </SCRIPT>

  </HEAD>

 <body>




 <p id="mypara" onmouseover="highlight()">
```

```
hello user
 </p>
</body>
</HTML>
```

onkeydown

when user pushes a keyboard key

```
<HTML>
   <HEAD>
     <SCRIPT>
       function keypressed()
{
  txt=document.getElementById("mytxt").value

  document.getElementById('mypara').innerHTML=txt
 }
    </SCRIPT>
   </HEAD>
  <body>

<input type='text' id='mytxt' onkeydown='keypressed()'/>

 <p id="mypara">
```

```
hello user
 </p>
</body>
</HTML>
```

In the above example textfield's onkeydown event is given a function keypressed ,that displays the text entered inside text field in a para

---

javascript browser object

The inbuilt objects of Javascript allow to communicate directly with browser

These are global objects and accessible across script

Lets see some of these objects

Window

Window object represents window

inside window we have certain properties and function

document object

this is available throgh window and used manipulate main html document

some functions of document are

write : writes on browser

getElementById :retrieves element based on Id given to the element

getElementsByName:gets all elements by specified name

Popup boxes

window object provides with different kind of popup boxes

alert ,confirm and prompt

alert box shows a text and a button

alert('hello to user')

confirm

A confitmbox contains an ok and a cancel button to accept or reject any event

confirm('click on ok or cancel')

prompt

A prompt is used to accept a value from the user before entering any page

var name=prompt("plese enter your name ")

summary

Documents built using HTML are mostly static in nature .To add interactive to documents ,scripting elements need to be included

Client side scripts like java script are platform independent and light weight

Processing time in client side scripting is reduced as data need not be sent to server for processing

Java script can also be said as default scripting laguage for a browser

Javascript can be placed in three ways over a document Internal script External script Inline script

Javascript programming involves usage of variables,operators ,functions ,decision constructs ,loops and arrays

variable in java script can be declared with a simple var keyword or even without using a keyword

An object in JavaScript looks more like a key value pair ,we can say objects are collection of named values

Methods inside object are functions stored as properties

Constructors in javascript can be used to create objects of multiple types

there are different operators available in javascript that are common to other programming languages as well

JavaScript provides with conditional construts to implement logic in scripts

Loop construtcs are used for creating a iterative block of statements

Java script provides different loop constructs

for loop ,while loop,do while loop,foreach

A foreach loop works for an array .It accepts a function as a parameter and applies function over each value of the array

An array in JavaScript is group of values .Array is used to store multiple values in a single variable

FAQ

1. What is JavaScript?

JavaScript is a client-side as well as server side scripting language that can be inserted into HTML pages and is understood by web browsers. JavaScript is also an Object based Programming language

2. Enumerate the differences between Java and JavaScript?

Java is a complete programming language. In contrast, 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 the different intent. Java is an object - oriented programming (OOPS) or structured programming language like C++ or C whereas JavaScript is a client-side scripting language.

3. What are JavaScript Data Types?

Following are the JavaScript Data types:

- Number
- String
- Boolean
- Object
- Undefined

4. What is the use of isNaN function?

isNan function returns true if the argument is not a number otherwise it is false.

5. Between JavaScript and an ASP script, which is faster?

JavaScript is faster. JavaScript is a client-side language and thus it does not need the assistance of the web server 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?

Negative Infinity is a number in JavaScript which can be derived by dividing negative number by zero.

7. Is it possible to break JavaScript Code into several lines?

Breaking within a string statement can be done by the use of a backslash, '\', at the end of the first line

Example:

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

And if you change to a new line when not within a string statement, then javaScript ignores break in line.

Example:

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

The above code is perfectly fine, though not advisable as it hampers debugging.

8. Which company developed JavaScript?

Netscape is the software company who developed JavaScript.

9. What are undeclared and undefined variables?

Undeclared variables are those that do not exist in a program and are not declared. If the program tries to read the value of an undeclared variable, then a runtime error is encountered.

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>
```

11. What are global variables? How are these variable declared and what are the problems associated with using them?

Global variables are those that are available throughout the length of the code, that is, these have no scope. The var keyword is used to declare a local variable or object. If the var keyword is omitted, a global variable is declared.

Example:

// Declare a global globalVariable = "Test";

The problems that are 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 prompt box is a box which allows the user to enter input by providing a text box. Label and box will be provided to enter the text or number.

13. What is 'this' keyword in JavaScript?

'This' keyword refers to the object from where it was called.

14. Explain the working of timers in JavaScript? Also elucidate the drawbacks of using the timer, if any?

Timers are used to execute a piece of code at a set time or also to repeat the code in a given interval of time. This is done by using the functions **setTimeout, setInterval** and **clearInterval**.

The **setTimeout(function, delay)** function is used to start a timer that calls a particular function after the mentioned delay. The **setInterval(function, delay)** function is used to repeatedly execute the given function in the mentioned delay and only halts when cancelled. The **clearInterval(id)** function instructs the timer to stop.

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?

// for Single line comments and

/* Multi

Line

Comment

*/

16. What is the difference between ViewState and SessionState?

'ViewState' is specific to a page in a session.

'SessionState' is specific to user specific data that can be accessed across all pages in the web application.

17. What is === operator?

=== is called as strict equality operator which returns true when the two operands are having the same value without any type conversion.

18. Explain how can you submit a form using JavaScript?

To submit a form using JavaScript use document.form[0].submit();

```
document.form[0].submit();
```

19. Does JavaScript support automatic type conversion?

Yes JavaScript does support automatic type conversion, it is the common way of type conversion used by JavaScript developers

20. How can the style/class of an element be changed?

It can be done in the following way:

```
document.getElementById("myText").style.fontSize = "20?;
```

or

```
document.getElementById("myText").className = "anyclass";
```

21. Explain how to read and write a file using JavaScript?

There are two ways to read and write a file using JavaScript

- Using JavaScript extensions

- Using a web page and Active X objects

22. What are all the looping structures in JavaScript?

Following are looping structures in Javascript:

- For
- While
- do-while loops

23. What is called Variable typing in Javascript?

Variable typing is used to assign a number to a variable and the same variable can be assigned to a string.

Example

```
i = 10;
i = "string";
```

This is called variable typing.

24. How can you convert the string of any base to integer in JavaScript?

The parseInt() function is used to convert numbers between different bases. parseInt() takes the string to be converted as its first parameter, and the second parameter is the base of the given string.

In order to convert 4F (of base 16) to integer, the code used will be -

```
parseInt ("4F", 16);
```

25. Explain the difference between "==" and "==="?

"==" checks only for equality in value whereas "===" is a stricter equality test and returns false if either the value or the type of the two variables are different.

26. What would be the result of 3+2+"7"?

Since 3 and 2 are integers, they will be added numerically. And since 7 is a string, its concatenation will be done. So the result would be 57.

27. Explain how to detect the operating system on the client machine?

In order to detect the operating system on the client machine, the navigator.platform string (property) should be used.

28. What do mean by NULL in Javascript?

The NULL value is used to represent no value or no object. It implies no object or null string, no valid boolean value, no number and no array object.

29. What is the function of delete operator?

The delete keyword is used to delete the property as well as its value.

Example

```
var student= {age:20, batch:"ABC"};
delete student.age;
```

30. What is an undefined value in JavaScript?

Undefined value means the

- Variable used in the code doesn't exist
- Variable is not assigned to any value
- Property doesn't exist

31. What are all the types of Pop up boxes available in JavaScript?

- Alert
- Confirm and
- Prompt

32. What is the use of Void(0)?

Void(0) is used to prevent the page from refreshing and parameter "zero" is passed while calling.

Void(0) is used to call another method without refreshing the page.

33. How can a page be forced to load another page in JavaScript?

The following code has to be inserted to achieve the desired effect:

```
<script language="JavaScript" type="text/javascript" >
```

```
<!-- location.href="http://newhost/newpath/newfile.html"; //--></script>
```

34. What is the data type of variables of in JavaScript?

All variables in the JavaScript are object data types.

35. What is the difference between an alert box and a confirmation box?

An alert box displays only one button which is the OK button.

But a Confirmation box displays two buttons namely OK and cancel.

36. What are escape characters?

Escape characters (Backslash) is used when working with special characters like single quotes, double quotes, apostrophes and ampersands. Place backslash before the characters to make it display.

Example:

```
document.write "I m a "good" boy"
document.write "I m a \"good\" boy"
```

37. What are JavaScript Cookies?

Cookies are the small test files stored in a computer and it gets created when the user visits the websites to store information that they need. Example could be User Name details and shopping cart information from the previous visits.

38. Explain what is pop()method in JavaScript?

The pop() method is similar as the shift() method but the difference is that the Shift method works at the start of the array. Also the pop() method take the last element off of the given array and returns it. The array on which is called is then altered.

Example:

```
var cloths = ["Shirt", "Pant", "TShirt"];
cloths.pop();
```

//Now cloth becomes Shirt,Pant

39. Whether JavaScript has concept level scope?

No. JavaScript does not have concept level scope. The variable declared inside the function has scope inside the function.

40. Mention what is the disadvantage of using innerHTML in JavaScript?

If you use innerHTML in JavaScript the disadvantage is

- Content is replaced everywhere
- We cannot use like "appending to innerHTML"
- Even if you use +=like "innerHTML = innerHTML + 'html'" still the old content is replaced by html
- The entire innerHTML content is re-parsed and build into elements, therefore its much slower
- The innerHTML does not provide validation and therefore we can potentially insert valid and broken HTML in the document and break it

41. What is break and continue statements?

Break statement exits from the current loop.

Continue statement continues with next statement of the loop.

42. What are the two basic groups of dataypes in JavaScript?

They are as –

- Primitive
- Reference types.

Primitive types are number and Boolean data types. Reference types are more complex types like strings and dates.

43. How generic objects can be created?

Generic objects can be created as:

```
var I = new object();
```

44. What is the use of type of operator?

'Typeof' is an operator which is used to return a string description of the type of a variable.

45. Which keywords are used to handle exceptions?

Try… Catch---finally is used to handle exceptions in the JavaScript

```
Try{
        Code
}
Catch(exp){
        Code to throw an exception
}
Finally{
        Code runs either it finishes successfully or after catch
}
```

46. Which keyword is used to print the text in the screen?

document.write("Welcome") is used to print the text – Welcome in the screen.

47. What is the use of blur function?

Blur function is used to remove the focus from the specified object.

48. What is variable typing?

Variable typing is used to assign a number to a variable and then assign string to the same variable. Example is as follows:

```
i= 8;
i="john";
```

49. How to find operating system in the client machine using JavaScript?

The '**Navigator.appversion'** is used to find the name of the operating system in the client machine.

50. What are the different types of errors in JavaScript?

There are three types of errors:

- **Load time errors**: Errors which come up when loading a web page like improper syntax errors are known as Load time errors and it generates the errors dynamically.
- **Run time errors**: Errors that come due to misuse of the command inside the HTML language.
- **Logical Errors**: These are the errors that occur due to the bad logic performed on a function which is having different operation.

51. What is the use of Push method in JavaScript?

The push method is used to add or append one or more elements to the end of an Array. Using this method, we can append multiple elements by passing multiple arguments

52. What is unshift method in JavaScript?

Unshift method is like push method which works at the beginning of the array. This method is used to prepend one or more elements to the beginning of the array.

53. What is the difference between JavaScript and Jscript?

Both are almost similar. JavaScript is developed by Netscape and Jscript was developed by Microsoft .

54. How are object properties assigned?

Properties are assigned to objects in the following way -

```
obj["class"] = 12;
```

or

```
obj.class = 12;
```

55. What is the 'Strict' mode in JavaScript and how can it be enabled?

Strict Mode adds certain compulsions to JavaScript. Under the strict mode, JavaScript shows errors for a piece of codes, which did not show an error before, but might be problematic and potentially unsafe. Strict mode also solves some mistakes that hamper the JavaScript engines to work efficiently.

Strict mode can be enabled by adding the string literal "use strict" above the file. This can be illustrated by the given example:

```
function myfunction() {
    "use strict";
    var v = "This is a strict mode function";
}
```

56. What is the way to get the status of a CheckBox?

The status can be acquired as follows -

alert(document.getElementById('checkbox1').checked);

If the CheckBox will be checked, this alert will return TRUE.

57. How can the OS of the client machine be detected?

The navigator.appVersion string can be used to detect the operating system on the client machine.

58. Explain window.onload and onDocumentReady?

The onload function is not run until all the information on the page is loaded. This leads to a substantial delay before any code is executed.

onDocumentReady loads the code just after the DOM is loaded. This allows early manipulation of the code.

59. How will you explain closures in JavaScript? When are they used?

Closure is a locally declared variable related to a function which stays in memory when the function has returned.

For example:

```
function greet(message) {

    console.log(message);

}

function greeter(name, age) {

    return name + " says howdy!! He is " + age + " years old";

}

// Generate the message

var message = greeter("James", 23);

// Pass it explicitly to greet

greet(message);
```

```
This function can be better represented by using closures

function greeter(name, age) {

    var message = name + " says howdy!! He is " + age + " years old";

    return function greet() {

        console.log(message);

    };

}

// Generate the closure

var JamesGreeter = greeter("James", 23);

// Use the closure

JamesGreeter();
```

60. How can a value be appended to an array?

A value can be appended to an array in the given manner -

arr[arr.length] = value;

61. Explain the for-in loop?

The for-in loop is used to loop through the properties of an object.

The syntax for the for-in loop is -

```
for (variable name in object){
        statement or block to execute
}
```

In each repetition, one property from the object is associated to the variable name, and the loop is continued till all the properties of the object are depleted.

62. Describe the properties of an anonymous function in JavaScript?

A function that is declared without any named identifier is known as an anonymous function. In general, an anonymous function is inaccessible after its declaration.

Anonymous function declaration -

```
var anon = function() {
        alert('I am anonymous');
};
anon();
```

63. What is the difference between .call() and .apply()?

The function .call() and .apply() are very similar in their usage except a little difference. .call() is used when the number of the function's arguments are known to the programmer, as they have to be mentioned as arguments in the call statement. On the other hand, .apply() is used when the number is not known. The function .apply() expects the argument to be an array.

The basic difference between .call() and .apply() is in the way arguments are passed to the function. Their usage can be illustrated by the given example.

```
var someObject = {
myProperty : 'Foo',

myMethod : function(prefix, postfix) {

        alert(prefix + this.myProperty + postfix);
}
};
someObject.myMethod('<', '>'); // alerts '<Foo>'
var someOtherObject  = {

        myProperty : 'Bar'

};
someObject.myMethod.call(someOtherObject, '<', '>'); // alerts '<Bar>'

someObject.myMethod.apply(someOtherObject, ['<', '>']); // alerts '<Bar>'
```

64. Define 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 parent will also work as if it were clicked too.

65. Is JavaScript case sensitive? Give an example?

Yes, JavaScript is case sensitive. For example, a function parseInt is not same as the function Parseint.

66. What boolean operators can be used in JavaScript?

The 'And' Operator (&&), 'Or' Operator (||) and the 'Not' Operator (!) can be used in JavaScript.

*Operators are without the parenthesis.

67. How can a particular frame be targeted, from a hyperlink, in JavaScript?

This can be done by including the name of the required frame in the hyperlink using the 'target' attribute.

```
<a href="/newpage.htm" target="newframe">>New Page</a>
```

68. What is the role of break and continue statements?

Break statement is used to come out of the current loop while the continue statement continues the current loop with a new recurrence.

69. Write the point of difference between web-garden and a web-farm?

Both web-garden and web-farm are web hosting systems. The only difference is that web-garden is a setup that includes many processors in a single server while web-farm is a larger setup that uses more than one server.

70. How are object properties assigned?

Assigning properties to objects is done in the same way as a value is assigned to a variable. For example, a form object's action value is assigned as 'submit' in the following manner - Document.form.action="submit"

71. What is the method for reading and writing a file in JavaScript?

This can be done by Using JavaScript extensions (runs from JavaScript Editor), example for opening of a file -

```
fh = fopen(getScriptPath(), 0);
```

72. How are DOM utilized in JavaScript?

DOM stands for Document Object Model and is responsible for how various objects in a document interact with each other. DOM is required for developing web pages, which includes objects like paragraph, links, etc. These objects can be operated to include actions like add or delete. DOM is also required to add extra capabilities to a web page. On top of that, the use of API gives an advantage over other existing models.

73. How are event handlers utilized in JavaScript?

Events are the actions that result from activities, such as clicking a link or filling a form, by the user. An event handler is required to manage proper execution of all these events. Event handlers are an extra attribute of the object. This attribute includes event's name and the action taken if the event takes place.

74. Explain the role of deferred scripts in JavaScript?

By default, the parsing of the HTML code, during page loading, is paused until the script has not stopped executing. It means, if the server is slow or the script is particularly heavy, then the webpage is displayed with a delay. While using Deferred, scripts delays execution of the script till the time HTML parser is running. This reduces the loading time of web pages and they get displayed faster.

75. What are the various functional components in JavaScript?

The different functional components in JavaScript are-

First-class functions: Functions in JavaScript are utilized as first class objects. This usually means that these functions can be passed as arguments to other functions, returned as values from other functions, assigned to variables or can also be stored in data structures.

Nested functions: The functions, which are defined inside other functions, are called Nested functions. They are called 'everytime' the main function is invoked.

76. Write about the errors shown in JavaScript?

JavaScript gives a message if it encounters an error. The recognized errors are -

- Load-time errors: The errors shown at the time of the page loading are counted under Load-time errors. These errors are encountered by the use of improper syntax, and thus are detected while the page is getting loaded.
- Run-time errors: This is the error that comes up while the program is running. It is caused by illegal operations, for example, division of a number by zero, or trying to access a non-existent area of the memory.
- Logic errors: It is caused by the use of syntactically correct code, which does not fulfill the required task. For example, an infinite loop.

77. What are Screen objects?

Screen objects are used to read the information from the client's screen. The properties of screen objects are -

- AvailHeight: Gives the height of client's screen
- AvailWidth: Gives the width of client's screen.
- ColorDepth: Gives the bit depth of images on the client's screen
- Height: Gives the total height of the client's screen, including the taskbar
- Width: Gives the total width of the client's screen, including the taskbar

78. Explain the unshift() method ?

This method is functional at the starting of the array, unlike the push(). It adds the desired number of elements to the top of an array. For example -

```
var name = [ "john" ];
name.unshift( "charlie" );
name.unshift( "joseph", "Jane" );
console.log(name);
```

The output is shown below:

```
[" joseph "," Jane ", " charlie ", " john "]
```

79. Define unescape() and escape() functions?

The escape () function is responsible for coding a string so as to make the transfer of the information from one computer to the other, across a network.

For Example:

```
<script>
        document.write(escape("Hello? How are you!"));
```

```
</script>
```

Output: Hello%3F%20How%20are%20you%21

The unescape() function is very important as it decodes the coded string.

It works in the following way. For example:

```
<script>
        document.write(unescape("Hello%3F%20How%20are%20you%21"));
</script>
```

Output: Hello? How are you!

80. What are the decodeURI() and encodeURI()?

EncodeURl() is used to convert URL into their hex coding. And DecodeURI() is used to convert the encoded URL back to normal.

```
<script>
        var uri="my test.asp?name=ståle&car=saab";

        document.write(encodeURI(uri)+ "<br>");

        document.write(decodeURI(uri));
</script>
```

Output -

my%20test.asp?name=st%C3%A5le&car=saab

my test.asp?name=ståle&car=saab

81. Why it is not advised to use innerHTML in JavaScript?

innerHTML content is refreshed every time and thus is slower. There is no scope for validation in innerHTML and, therefore, it is easier to insert rouge code in the document and, thus, make the web page unstable.

82. What does the following statement declares?

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

It declares a three dimensional array.

83. How are JavaScript and ECMA Script related?

ECMA Script are like rules and guideline while Javascript is a scripting language used for web development.

84. What is namespacing in JavaScript and how is it used?

Namespacing is used for grouping the desired functions, variables etc. under a unique name. It is a name that has been attached to the desired functions, objects and properties. This improves modularity in the coding and enables code reuse.

85. How can JavaScript codes be hidden from old browsers that don't support JavaScript?

For hiding JavaScript codes from old browsers:

Add "<!--" without the quotes in the code just after the <script> tag.

Add "//-->" without the quotes in the code just before the <script> tag.

Old browsers will now treat this JavaScript code as a long HTML comment. While, a browser that supports JavaScript, will take the "<!--" and "//-->" as one-line comments.