JavaScript

Why we study JavaScript?

- **HTML**: HTML is used to define the content of web pages
- **CSS:** CSS is used to specify the layout of web pages
- **JavaScript:** JavaScript is used to program the behaviour of web pages

Difference between JavaScript and Java

Java	JavaScript
Java creates applications that run in a virtual machine or browser	JavaScript code is run on a browser only.
Java is purely an object-oriented language.	JavaScript is not purely an object-oriented language.
Java is complied and interpreted language.	JavaScript is interpreted language. It is not compiled language, the clients directly interprets it.
Java is a strongly typed language.	JavaScript is a freely typed language.

Introduction

- It is a lightweight programming language.
- It is a scripting language.
- It is an interpreted language.
- It was developed by Netscape Corporation.
- Initially it was known as "Live Script"
- It was designed to add interactivity to HTML pages.
- It is usually directly embedded into HTML pages.
- It supports client side scripting.
- It also supports server-side scripting, separately known as Livewire.
- Initially supported by Netscape Navigator 2.0.
- Currently supported by most of the web browser Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, etc.
- It was also referred to as ECMA (European Computer Manufacturer's Association) Script.

Limitations of JavaScript

1. Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
2. JavaScript cannot be used for networking applications because there is no such support available.
3. JavaScript doesn't have any multithreading or multiprocessor capabilities.
4. **Strength of Java Script**

Strength of Java Script/ What JavaScript can do?

1. **Read and write HTML elements:** It can read and change the content of a HTML element.
2. **Validate data:** It can be used to validate form data before it is submitted to a server. This saves the server from extra processing and network congestion.
3. **Put dynamic text into a HTML page:** A JavaScript statement can write a variable text into an HTML page.
4. **Detect the visitor's browser**: It can be used to detect the visitor's browser, and depending on the browser load another page specifically designed for that browser.
5. **React to events:** It can be set to execute when something happens, like when a page has finished loading or when a user clicks on a HTML element.
6. **Create cookies**: It can be used to store and retrieve information on the visitor's computer.

Advantages of JavaScript

1. **An Interpreted Language:** It requires no compilation steps. This provides an easy development process. The code is directly interpreted by any JavaScript enabled web browser.
2. **Embedded Within HTML:** It doesn't require any special software, coding can be done in any text editor, and saved with .html extension.
3. **Easy to learn:** By learning few commands and simple syntax, complete applications can be built using JavaScript.
4. **Performance:** JavaScript program files are fairly compact and quite small. This minimizes storage requirements on the web server and downloads time for the client.
5. **Procedural Capabilities:** It provides condition checking, Looping and Branching statements which add procedural capabilities to web page (HTML) coding.
6. **Programming User Events:** It supports Object/Event based programming. It recognizes when a form *Button* is pressed. This event can have suitable JavaScript code attached, which will execute when *Button Pressed* event occurs.
7. **Easy Debugging and Testing:** Being an interpreted language, Java Script is tested line by line. Errors are also listed with an appropriate message along with the line number. It is thus easy to locate error, make changes and test it again without overhead and delay of compiling.
8. **Platform Independent:** It is completely independent of the hardware on which it works. JavaScript application work on any machine that has an appropriate JavaScript enabled web browser installed.
9. **Less Server Interaction:** It can validate user input before sending the page data to the server. This saves network traffic and fewer loads on server.

JavaScript Syntax

- It starts with <SCRIPT> tag and end with the </SCRIPT> tag in a web page.
- It can be placed anywhere in web page, but it is recommended to keep in the head section of webpage.

Syntax:

```
<SCRIPT ……>
        JavaScript code
</SCRIPT>
```

The script tag takes two attributes:

i. **Language**: This attribute specifies which scripting language we are using. Typically, its value will be JavaScript. Although recent versions of HTML have phased out the use of this attribute.

ii. **Type**: This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

Syntax:

```
<SCRIPT language="javascript" type="text/javascript">
        JavaScript code
</SCRIPT>
```


Placement of JavaScript in HTML file

- There is a flexibility given to include JavaScript code anywhere in an HTML document.
- The most preferred ways to include JavaScript in your HTML file are:
    - <HEAD>...</HEAD> section.
    - <BODY>...</BODY> section.
    - In an external file.

1. <HEAD>...</HEAD> section

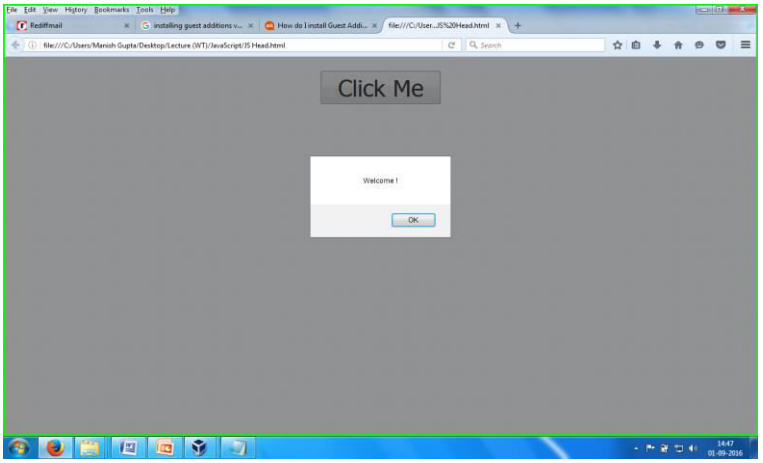
Syntax:

```
<HTML>
<HEAD>
<SCRIPT type="text/javascript">
        JavaScript code….
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Example: 1

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
function Welcome()
{
        alert("Welcome !")
}
</SCRIPT>
</HEAD>
 <BODY>
<CENTER>
<INPUT TYPE="button"
onclick="Welcome()" value="Click Me">
</CENTER>
</BODY>
</HTML>
```

Output: 1



2. <BODY>...</BODY> section

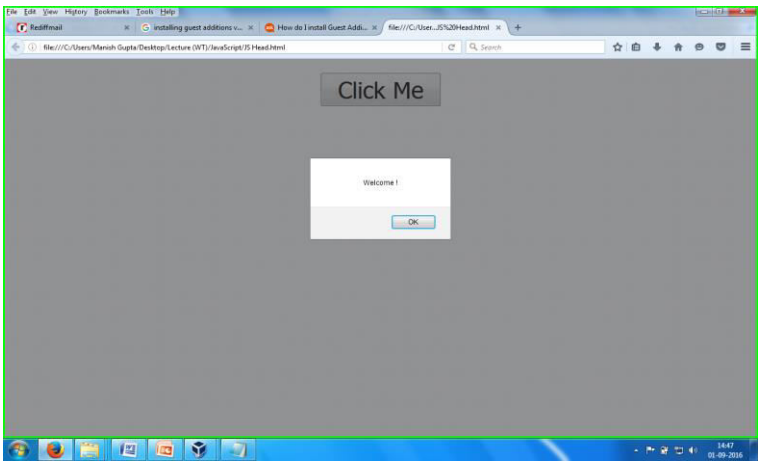
Syntax:

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<SCRIPT type="text/javascript“>
        JavaScript code….
</SCRIPT>
<P>This is a paragraph</P>
</BODY>
</HTML>
```


Example: 2

```
<HTML>
<BODY>
<SCRIPT TYPE="text/javascript">
function Welcome()
{
        alert("Welcome !")
}
</SCRIPT>
<CENTER>
<INPUT TYPE="button"
onclick="Welcome()" value="Click Me">
</CENTER>
</BODY>
</HTML>
```

Output: 2



3. In an external file

Syntax:

```
<HTML>
<HEAD>
<SCRIPT type="text/javascript" src="filename.js”>
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Example: 3

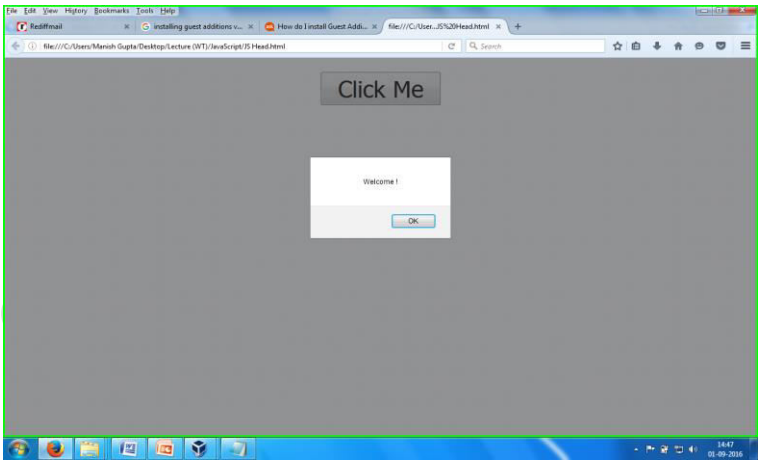
Code for myscript.js

```
function Welcome()
{
        alert("Welcome !")
}
```

Code for HTML file

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript"
SRC="myscript.js">
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<INPUT TYPE="button" onclick="Welcome()" value="Click Me">
</CENTER>
</BODY>
</HTML>
```

Output: 3



Variables

- JavaScript variables are named containers.
- Variable name must start with an alphabet, $, _.
- Variable names are case sensitive.
- Variable name should not be a JavaScript reserved word.
- Variable name does not contain space.
- Variables can also hold the value of any data type.
- Variables can be declared using ***var*** keyword.
- Variable has dynamic types.

Syntax:

var <variable name>=value;

Example:

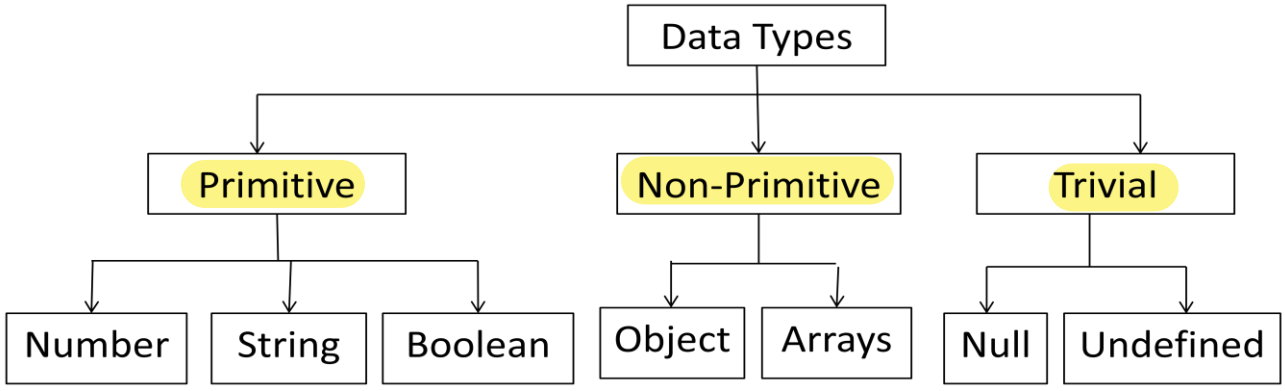
(i) To assign a text value use single or double quotes around the value.

var x="Web Technology";

(ii) To assign a numeric value don't use any quotes around the value.

var x=700.75;

Data Types



Number

- JavaScript has only one type of numbers.
- Numbers can be written with, or without decimals.

Example:

```
var x=700.75;          //Written with decimals
var y=11;              //Written without decimals
```

- Extra large or extra small numbers can be written with scientific (exponential) notation.

Example:

```
var x=123e5    //12300000
var y=123e-5   //0.00123
```

String

- A string is a series of characters.
- A string can be written inside quotes (single, double).

Example:

```
var name = "Amit";    // Using double quotes
var name = 'Sumit';   // Using single quotes
```

- We can also use quotes inside a string, but it should not match with the quotes surrounding the string.

Example:

```
var name = "My name is 'Amit'";
var name = 'My name is "Sumit"';
```

Boolean

- Booleans can only have two values: *true* or *false*.
- Booleans are often used in conditional testing.

Example:

```
var x = true;
var y = false;
```

Object

- JavaScript objects are written with curly braces.
- Object properties are written as *name : value* pairs, separated by commas.

Example:

```
var person = {Name:“ABC”, Age:50, Gender:“Male”};
```

The object (person) has 3 properties: Name, Age, Gender.

Array

- JavaScript arrays are written with square brackets.
- Array items are separated by commas.
- Array indexing starts with zero.

Example:

```
var name = [“ABC", “PQR", “XYZ”];
```

The code creates an array called name, containing three values.

NULL

- JavaScript *NULL* is "nothing". It is supposed to be something that doesn't exist.
- The data type of NULL is an object.

Example:

```
var person = null;      // Value is null, but type is still an object
```

Undefined

- A variable without a value has the value undefined.
- The data type of undefined is an undefined.
- Any variable can be emptied, by setting the value to undefined.

Example:

```
var person;                 // Value is undefined, type is undefined
var person = undefined;     // Value is undefined, type is undefined
```

Declaring Variable Type

- We can declare a new variable and its type using the "new" keyword.
- JavaScript variables are all objects.
- When we declare a variable we create a new object.

Example:

```
var x =new String;
var x= new Number;
var x= new Boolean;
var x= new Array;
var x =new Object;
```

Operators

- JavaScript operators are symbols that are used to perform operations on operands.
- There are following types of operators in JavaScript:
  - Arithmetic Operators
  - Relational Operators
  - Bitwise Operators
  - Logical Operators
  - Assignment Operators

Arithmetic Operators

Operators	Description	Example
+	Addition	1+2=3
-	Subtraction	5-2=3
*	Multiplication	2*3=6
/	Division	10/2=5
%	Modulus	8%3=2
++	Increment	var a=10; a++; Now a=11
--	Decrement	var a=10; a--; Now a=9

Relational Operator

Operators	Description	Example
= =	Is equal to	10==20 = false
!=	Not equal to	10!=20 = true
>	Greater than	20>10 = true
>=	Greater than or equal to	20>=10 = true
<	Less than	20<10 = false
<=	Less than or equal to	20<=10 = false
= = =	Identical	“5” = = = 5 =false
!= =	Not Identical	“20”! = =20 =true

Bitwise Operator

Operators	Description	Example
&	Bitwise AND	5 & 3= 1
|	Bitwise OR	5 | 3=7
^	Bitwise XOR	5^3=6
~	Bitwise NOT	(~10)=-11
<<	Bitwise Left Shift	5 << 2=20

Operators	Description	Example
>>	Bitwise Right Shift	5 >> 2=1
>>>	Bitwise Right Shift with Zero fill	5 >>> 2=1

Logical Operator

Operators	Description	Example
&&	Logical AND	(10==20 && 20==30) = false
‖	Logical OR	(10==20 ‖ 20==30) =false
!	Logical Not	! (10==20) =true

Assignment Operator

Operators	Description	Example
=	Assignment	10+10=20

Dialog Boxes

In JavaScript there are three kinds of dialog boxes:

- Alert box
- Confirm box
- Prompt box

Alert Box

- An alert dialog box is mostly used to give a warning message to the users.
- An alert box can also be used for friendlier messages.
- Alert box gives only one button "OK" to select and proceed.

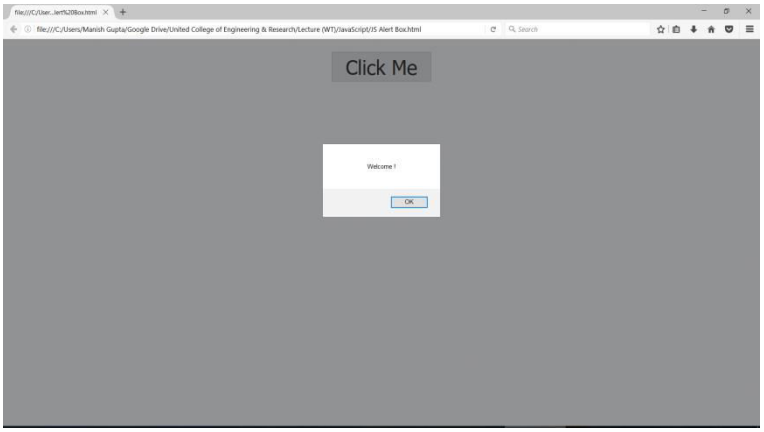
Syntax:

```
alert("sometext")
```

Example: 1

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
function Welcome()
{
        alert("Welcome !")
}
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<INPUT TYPE="button" onclick="Welcome()" value="Click Me">
</CENTER>
</BODY>
</HTML>
```

Output: 1



Confirm Box

- A confirmation dialog box is mostly used to take user's consent on any option.
- It displays a dialog box with two buttons: ***OK*** and ***Cancel***.
- If the user clicks on the OK button, the window method confirm() will return true.
- If the user clicks on the Cancel button, then confirm() returns false.

Syntax:

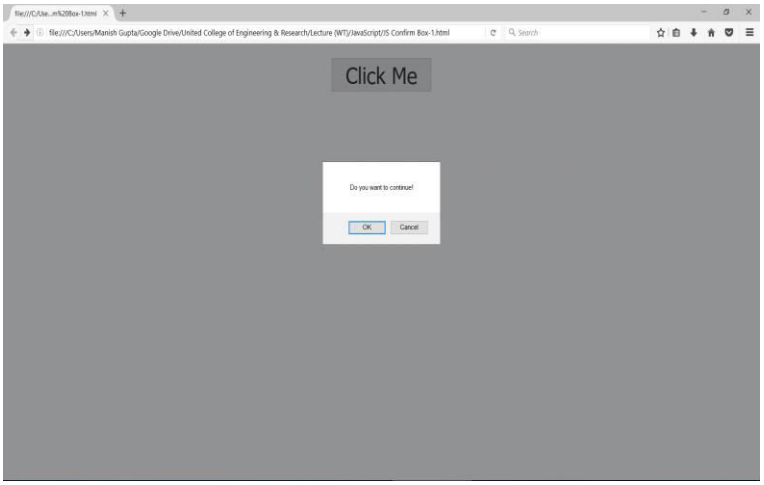
```
confirm("sometext")
```

Example: 2

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
function Welcome()
{
        var value=confirm("Do you want to continue!");
        if(value==true)
```

```
            document.write("Welcome");
        else
            document.write("Thank You");
}
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<INPUT TYPE="button"
onclick="Welcome()" value="Click Me">
</CENTER>
</BODY>
</HTML>
```

Output: 2



Prompt Box

- A prompt box is often used if we want the user to input a value before entering a page.
- This dialog box has two buttons: **OK** and **Cancel**.
- If the user clicks "OK" the box returns the input value.
- If the user clicks "Cancel" the box returns null.
- This dialog box is displayed using a method called **prompt()** which takes two parameters:
    - a label which you want to display in the text box
    - a default string to display in the text box.

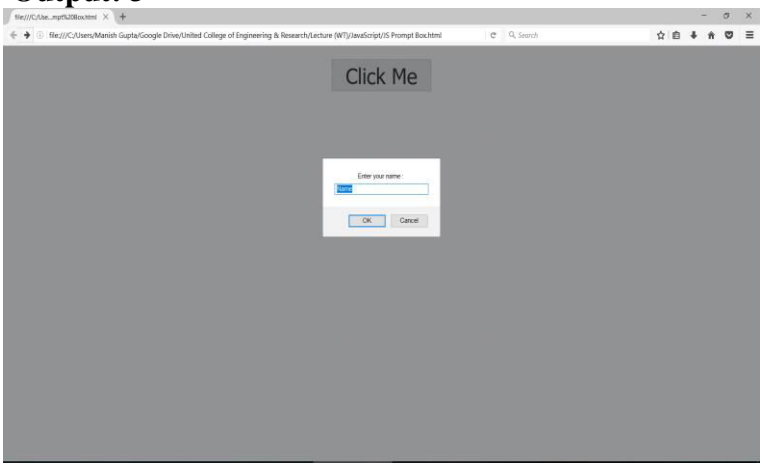
Syntax:

```
prompt("sometext","defaultvalue“)
```

Example: 3

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
function Welcome()
{
   var value=prompt("Enter your name :
","Name");
   if(value==null)
      alert("Thank You");
   else
      document.write("Welcome "+value);
}
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<INPUT TYPE="button" onclick="Welcome()" value="Click Me">
</CENTER>
</BODY>
</HTML>
```

Output: 3



Conditional Checking

Syntax- 1: if-else

```
if (condition)
{
        Statements;     //Executed if condition is true
}
else
{
        Statements;     //Executed if  condition is false
}
```

Example:

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
var value=1;
if(value==1)
        document.write("One");

else
        document.write("Other");
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Output

One

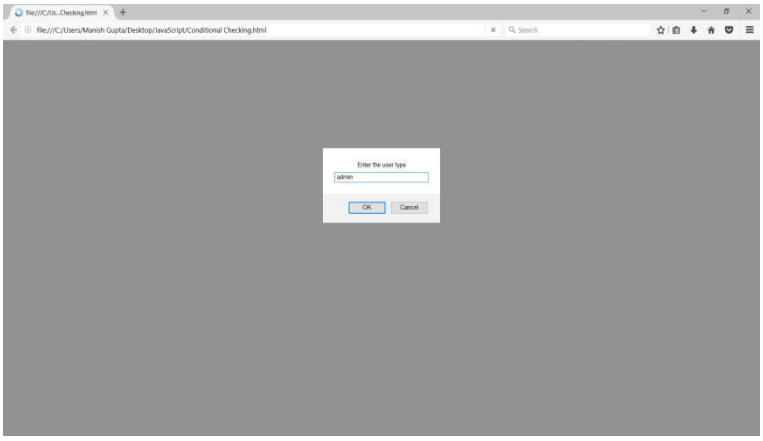
Syntax- 2: if-elseif-else

```
if(condition1)
{
   Statements;    //Executed if condition1 is true
}
else if (condition2)
{
   Statements;    //Executed if condition1 and condition2 are true
}
else
{
    Statements;    //Executed if condition1 and condition2 are false
}
```

Example:

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
var type;
type=prompt("Enter the user type","");
if(type=="admin")
        document.write("Welcome
        Administrator");
else if(type=="user")
        document.write("Welcome user");
else
```

Output



```
        document.write("Sorry…");
</SCRIPT>
</HEAD>
</HTML>
```

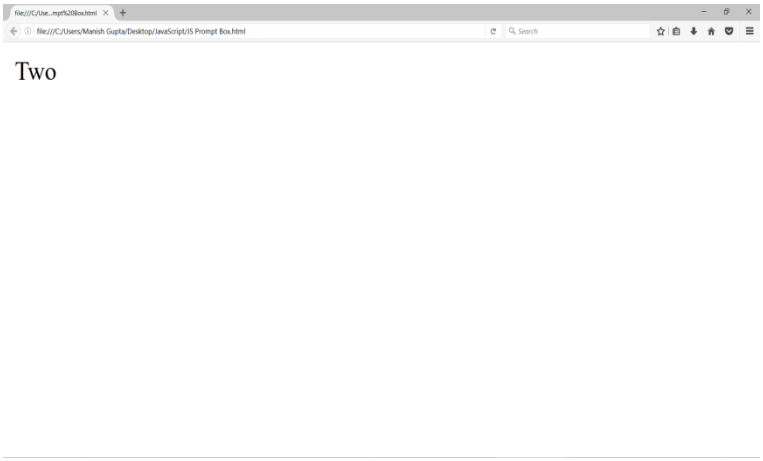
Syntax-3: switch-case

```
switch(expression)
{
        case n:
                        code block
                        break;
        case n:
                        code block
                        break;
        default:
                        default code block
}
```

Example:

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
var value=2;
switch(value)
{
        case 1:
                document.write("One");
                break;
        case 2:
                document.write("Two");
                break;
        default:
                document.write("Other");
}
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Output



Functions

- A function is a block of code designed to perform a particular task.
- To keep the browser from executing a script when the page loads, we can put our script into a function.
- A function is executed when "someone" calls it.
- A function is a reusable code-block that will be executed by an event, or function call.
- A function contains code that will be executed by an event or by a call to that function.
- We can call a function from anywhere within the page or even from other pages (if function is embedded in an external .js file).

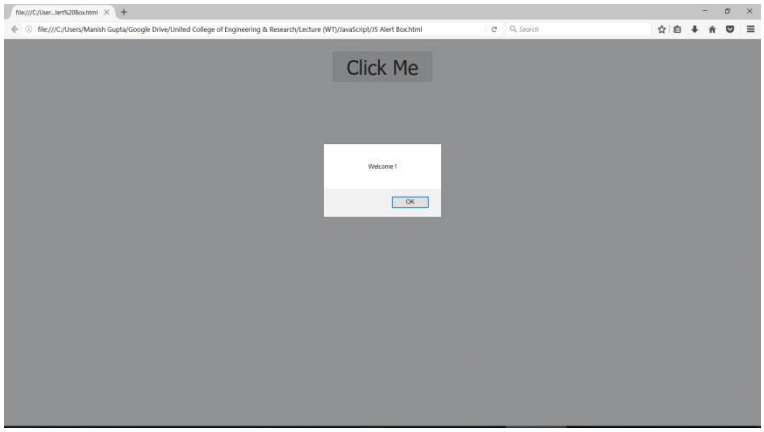
Syntax:

function *functionname*()
{
    some code to be executed
}

Example:

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
function Welcome()
{
    alert("Welcome !")
}
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<INPUT TYPE="button"
onclick="Welcome()" value="Click Me">
</CENTER>
</BODY>
</HTML>
```

Output



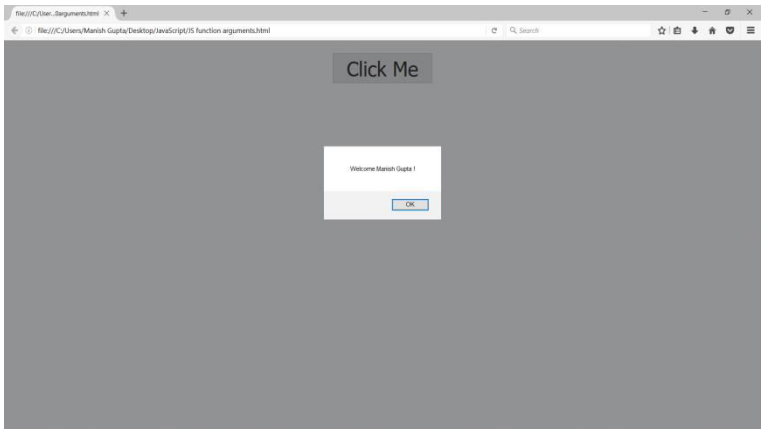
Calling a Function with Arguments

- We can call a function by passing some values to it; these values are called *arguments* or *parameters*.
- These arguments/parameters can be used inside the function.

Example:

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
function Welcome(name)
{
    alert("Welcome "+name+" !")
}
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<INPUT TYPE="button"
onclick="Welcome('Manish Gupta')" value="Click Me">
</CENTER>
</BODY>
</HTML>
```

Output



Functions with a Return Value

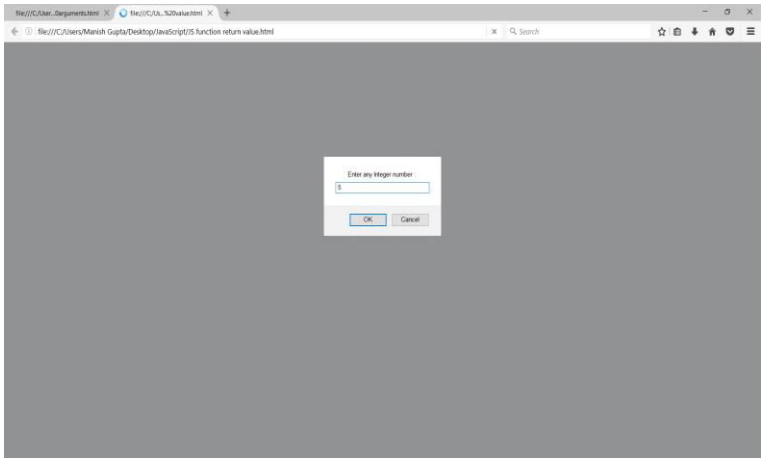
- Sometimes we want our function to return a value.

- This is possible by using the return statement.
- When using the return statement, the function will stop executing, and return the specified value.

Example:

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
function Calculate()
{
    var fact=1;
    var num=prompt("Enter any Integer
number : ")
    for(var i=1;i<=num;i++)
    {
        fact=fact*i;
    }
    return fact;
}
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT TYPE="text/javascript">
document.write("The factorial of given
number is " +Calculate());
</SCRIPT>
</BODY>
</HTML>
```

Output



The factorial of given number is 120

Data Type Conversions

- JavaScript is a dynamically typed language.
- We can convert a number into String and vice-versa.
- **Converting a String into number:**
    - For converting a String into number 'parseInt' method is used.
    - **For Example:**
        - int a = parseInt("21");
        - int b = parseInt("5.7");
- **Converting a number into String:**
    - For converting a number into String simply add an empty quote to that number.
    - **For Example:**
        - String a= ""+21;
        - String b= ""+5.7;

Types of Iteration Statements

JavaScript supports different kinds of loops:

- for loop
  - Fixed number of iterations.
- while loop
  - Variable number of iterations.
- do/while (do-while)
  - Variable number of iterations.
- for/in (for…in)
  - Variable number of iterations.

for loop

Example:

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
var i;
for(i=1;i<=5;i++)
    document.write(i+" ");
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```


Output:

1 2 3 4 5

while loop

Example:

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
var i=1;
while(i<=5)
{
        document.write(i+" ");
        i++;
}
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Output:

1 2 3 4 5

do-while loop

Example:

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
var i=0;
do
{
        i++;
        document.write(i+" ");
} while(i<5);
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Output:

1 2 3 4 5

for...in loop

- The for/in statement is used to loop through the elements of an array or through the properties of an object.
- The code in the body of the for/in loop is executed once for each element/property.

Syntax:

```
for (variable in object)
{
 code to be executed
}
```

Example: for...in loop with array

```
<HTML>
<BODY>
<SCRIPT TYPE="text/javascript">
var x
var fname = new Array()
fname[0] = "Amit"
fname[1] = "Sumit"
fname[2] = "Ravi"
for (x in fname)
{
        document.write(fname[x] + "<br>")
}
```

```
</SCRIPT>
</BODY>
</HTML>
```

Output:

```
Amit
Sumit
Ravi
```

Example: for...in loop with object

```
<HTML>
<BODY>
<SCRIPT TYPE="text/javascript">
var x;
var person = {name:"Amit", age:25, gender:"Male", place:"Allahabad"};
for (x in person)
{
   document.write(person[x] + "<br>")
}
</SCRIPT>
</BODY>
</HTML>
```


Output:

```
Amit
25
Male
Allahabad
```

Events

- HTML events are "things" that happen to HTML elements.
- It can be something the browser does, or something a user does.
- Examples of HTML Events:
    - HTML page has finished loading.
    - HTML input field was changed.
    - HTML button was clicked.
- When JavaScript is used in HTML pages, it can "react" on these events.
- Often, when events occur, we may want to do something.
- Events are normally used in combination with functions.
- Every HTML element has some restricted events associated with it.
- Example:
    - onchange: Select, Text, Text area
    - click: Button, Checkbox, Document, Links, Radio, Reset, Submit

Mouse Events

- **onclick**

- Occurs when the user clicks on an element.
- **ondblclick**
  - Occurs when the user double –clicks on an element.
- **onmousedown**
  - Occurs when a user presses a mouse button over an element.
- **onmousemove**
  - Occurs when the pointer is moving while it is over an element.
- **onmouseover**
  - Occurs when the pointer is moved onto an element.
- **onmouseout**
  - Occurs when a user moves the mouse pointer out of an element.
- **onmouseup**
  - Occurs when a user releases a mouse button over an element.

Keyboard Events

- **onkeydown**
  - Occurs when the user is pressing a key.
- **onkeypress**
  - Occurs when the user presses a key.
- **onkeyup**
  - Occurs when the user releases a key.

Form Events

- **onblur**
  - Occurs when a form element loses focus.
- **onchange**
  - Occurs when the content of a form element have changed.
- **onfocus**
  - Occurs when an element gets focus.
- **onreset**
  - Occurs when a form is reset.
- **onsubmit**
  - Occurs when a form is submitted.

document.getElementsByName()

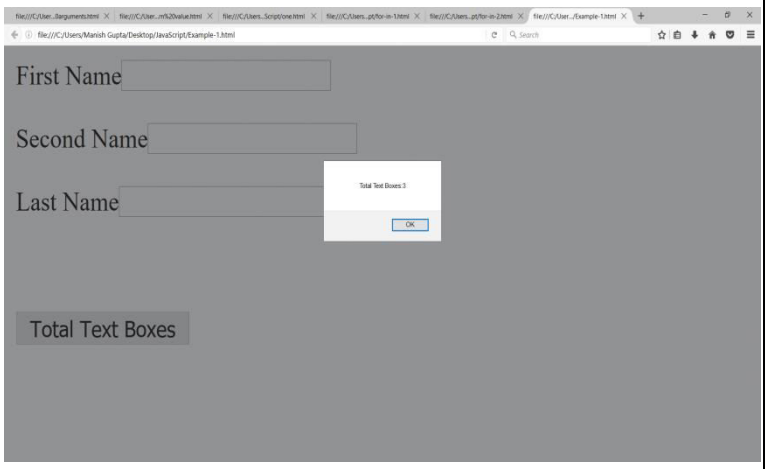
- This method returns all the element of specified name.

Syntax:

```
document.getElementsByName("name");
```

Example: **Output**

```
<HTML>
<HEAD>
<script type="text/javascript">
function calculate()
```




```
{
        var total=
        document.getElementsByName
        ("T1");
        alert("Total Text
        Boxes:"+total.length);
}
</script>
</HEAD>
<BODY>
<form>
First Name<input type="text" name="T1"><BR><BR>
Second Name<input type="text" name="T1"><BR><BR>
Last Name<input type="text" name="T1"><BR><BR>
<BR><BR>
<input type="button" onclick="calculate()" value="Total Text Boxes">
</form>
</BODY>
</HTML>
```

document.getElementById()

- This method returns the element of specified id.

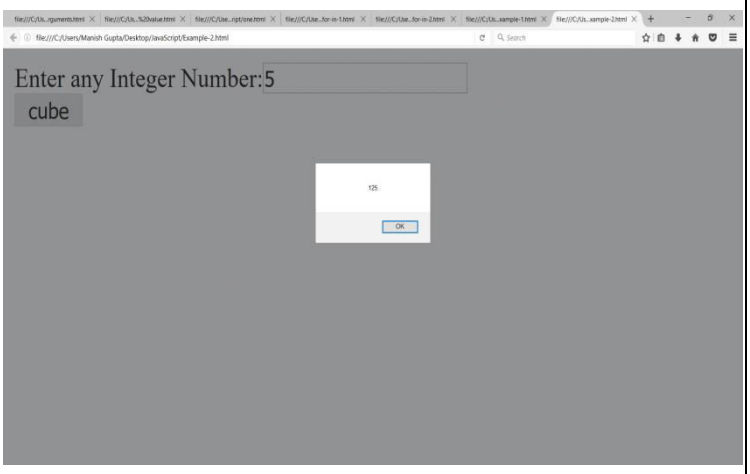
Syntax:

document.getElementById("id");

Example:

```
<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript">
function calculate()
{
    var number=
    document.getElementById
    ("number").value;
   alert(number*number*number);
}
</SCRIPT>
</HEAD>
<BODY>
<FORM>
Enter any Integer Number:<INPUT TYPE="text" ID="number"><br>
<INPUT TYPE="button" VALUE="cube" ONCLICK="calculate()">
</FORM>
```

Output



```
</BODY>
</HTML>
```

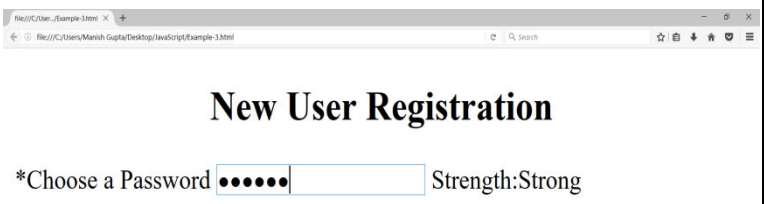
innerText Property

- It is used to write the dynamic text on the html document.
- The written text will not be interpreted as html text but a normal text.
- It is mostly used to generate the dynamic content on web page. (e.g. writing the validation message, password strength)

Example:

```
<HTML>
<HEAD>
<SCRIPT language="javascript"
TYPE="text/javascript">
function verify()
{
    var message;
    if(document.form1.P1.value.length>5)
    {
        message="Strong";
    }
    else
    {
        message="Weak";
    }
    document.getElementById('one').innerText=message;
}
</SCRIPT>
</HEAD>
<BODY>
<CENTER><H2>New User Registration</H2></CENTER>
<FORM NAME="form1">
*Choose a Password
<INPUT TYPE="password" NAME="P1" onkeyup="verify()">
Strength:<SPAN ID="one">no strength</SPAN>
</FORM>
</BODY>
</HTML>
```

Output



innerHTML Property

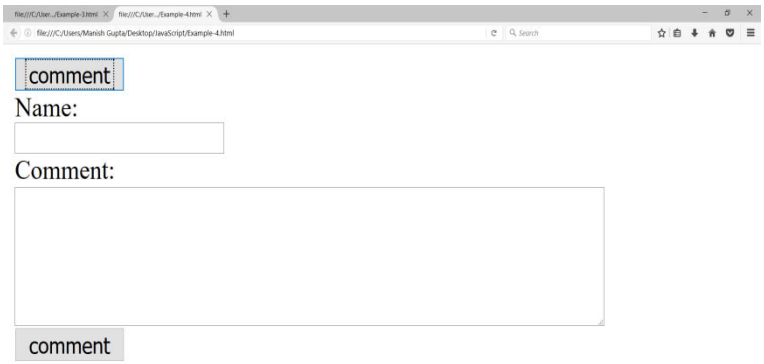
- It is used to write the dynamic html on the html document.
- It is mostly used to generate the dynamic html such as registration form, comment form, links etc.

Example:

```
<html>
<head>
<script type="text/javascript" >
function show()
{
    var data="Name:<br>
    <input type='text' name='name'><br>
    Comment:<br>
    <textarea rows='5'
cols='50'></textarea><br>
    <input type='submit'
value='comment'>";

document.getElementById('one').innerHTML=data;
 }
</script>
</head>
<body>
<form name="myForm">
<input type="button" value="comment" onclick="show()">
<div id="one"></div>
</form>
</body>
</html>
```

Output



Form Validation

- JavaScript can be used to validate data in HTML forms before sending the content to server.
- JavaScript can check the following:
    - Mandatory fields are entered?
    - Valid e-mail address entered?
    - Valid date is entered?
    - Valid numeric data is entered?

Example:

```
<HTML>
<HEAD>
<SCRIPT language="JavaScript">
function verify()
{
        if(!check_name())
                return false;
        if(!check_date())
                return false;
        if(!check_month())
                return false;
        if(!check_year())
```

```
                return false;
        if(!checkdate())
                return false;
        if(!check_address())
                return false;
        if(!check_contact())
                return false;
        if(!check_email())
                return false;
        return true;
}
function check_name()
{
        val=document.myform.T1.value;
        len=val.length;
        if(val=="")
        {
                alert("Please enter your name");
                document.forms[0].elements[0].focus();
                return(false);
        }
        if(len>20)
        {
                alert("Name exceeds 20 characters");
                document.forms[0].elements[0].focus();
                return(false);
        }
        if(!isCharsInBag(val,"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTU
VWXYZ "))
        {
                alert("Name must only contain alphabets");
                document.forms[0].elements[0].focus();
                return(false);
        }
        return(true);
}
function check_date()
{
        dval=document.forms[0].elements[1].value;
        if(dval=="0")
        {
                alert("Please select the date");
                document.forms[0].elements[1].focus();
                return(false);
        }
        return(true);
}
function check_month()
{
        mval=document.forms[0].elements[2].value;
```

```
	if(mval=="0")
	{
		alert("Please select the month");
		document.forms[0].elements[2].focus();
		return(false);
	}
	return(true);
}
function check_year()
{
	yval=document.forms[0].elements[3].value;
	if(yval=="0")
	{
		alert("Please select the year");
		document.forms[0].elements[3].focus();
		return(false);
	}
	return(true);
}
function checkdate()
{
	dtval=document.forms[0].elements[1].value;
	monval=document.forms[0].elements[2].value;
	yval=document.forms[0].elements[3].value;
	if(dtval==31)
	{
		if((monval==2) || (monval==4) || (monval==6) || (monval==9) ||
(monval==11))
		{
			alert("Invalid date.");
			document.forms[0].elements[1].focus();
			return(false);
		}
	}
	if(dtval==29)
	{
		if(((yval%4!=0) && ((yval%100==0) || (yval%400!=0)))&&(monval==2))
		{
			alert("Invalid date.");
			document.forms[0].elements[1].focus();
			return(false);
		}
	}
	if(dtval==30)
	{
		if(monval==2)
		{
			alert("Invalid date.");
			document.forms[0].elements[1].focus();
			return(false);
```

```
                }
        }
        return(true);
}
function check_address()
{
        add=document.forms[0].elements[6].value;
        len=document.forms[0].elements[6].value.length;
        if(add=="")
        {
                alert("Please enter the address");
                document.forms[0].elements[6].focus();
                return(false);
        }
        if(len>200)
        {
                alert("Address exceeds 200 characters");
                document.forms[0].elements[6].focus();
                return(false);
        }
        return(true);
}
function check_contact()
{
        con=document.forms[0].elements[7].value;
        len=document.forms[0].elements[7].value.length;
        if(con=="")
        {
                alert("Please enter the contact number");
                document.forms[0].elements[7].focus();
                return(false);
        }
        if(len!=10)
        {
                alert("Invalid number");
                document.forms[0].elements[7].focus();
                return(false);
        }
        if(!isCharsInBag(con,"0123456789"))
        {
                alert("Invalid Number");
                document.forms[0].elements[7].focus();
                return(false);
        }
        return(true);
}
function check_email()
{
        mail=document.forms[0].elements[8].value;
        len=document.forms[0].elements[8].value.length;
```

```
	if(mail=="")
	{
		alert("Please enter the email id");
		document.forms[0].elements[8].focus();
		return(false);
	}
	if(!isCharsInBag(mail,"abcdefghijklmnopqrstuvwxyz0123456789@._#$"))
	{
		alert("Invalid email id");
		document.forms[0].elements[8].focus();
		return(false);
	}
	count=0;
	for(i=0;i<len;i++)
	{
		if(mail.charAt(i)=="@")
			count++;
	}
	if(count!=1)
	{
		alert("Invalid email id");
		document.forms[0].elements[8].focus();
		return(false);
	}
	return(true);
}
function isCharsInBag(string,bag)
{
	var i;
	for(i=0;i<string.length;i++)
	{
		var charval=string.charAt(i);
		if(bag.indexOf(charval)==-1)
		{
			return (false);
		}
	}
	return (true);
}
</SCRIPT>
</HEAD>
<BODY>
<CENTER><H2>New User Registration</H2></CENTER>
<form method="get" onSubmit="return verify()" action="a.html" name="myform">
*Name <input type="text" name="T1">     (not more than
20 characters)
<BR>
<BR>
*Date of Birth
<select name="S1">
```

```
<option selected value="0">Select</option>
<option value="01">01</option>
<option value="02">02</option>
<option value="03">03</option>
<option value="04">04</option>
<option value="05">05</option>
<option value="06">06</option>
<option value="07">07</option>
<option value="08">08</option>
<option value="09">09</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
<option value="13">13</option>
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>
<option value="18">18</option>
<option value="19">19</option>
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
</select>
 <select name="S2">
<option selected value="0">Select</option>
<option value="1">January</option>
<option value="2">February</option>
<option value="3">March</option>
<option value="4">April</option>
<option value="5">May</option>
<option value="6">June</option>
<option value="7">July</option>
<option value="8">August</option>
<option value="9">September</option>
<option value="10">October</option>
<option value="11">November</option>
<option value="12">December</option>
</select>
```

```
<select name="S3">
<option selected value="0">Select</option>
<option value="1985">1985</option>
<option value="1986">1986</option>
<option value="1987">1987</option>
<option value="1988">1988</option>
<option value="1989">1989</option>
<option value="1990">1990</option>
<option value="1991">1991</option>
<option value="1992">1992</option>
<option value="1993">1993</option>
<option value="1994">1994</option>
<option value="1995">1995</option>
<option value="1996">1996</option>
<option value="1997">1997</option>
<option value="1998">1998</option>
<option value="1999">1999</option>
<option value="2000">2000</option>
<option value="2001">2001</option>
<option value="2002">2002</option>
<option value="2003">2003</option>
<option value="2004">2004</option>
<option value="2005">2005</option>
<option value="2006">2006</option>
<option value="2007">2007</option>
<option value="2008">2008</option>
<option value="2009">2009</option>
<option value="2010">2010</option>
<option value="2011">2011</option>
<option value="2012">2012</option>
<option value="2013">2013</option>
<option value="2014">2014</option>
<option value="2015">2015</option>
</select>
<BR>
<BR>
*Gender
<input name="R1" type="radio" value="Male" checked="checked">Male
<input name="R1" type="radio" value="Female">Female
<BR>
<BR>
*Address <textarea name="textarea" cols="30" rows="7" ></textarea>
     (not more than 200 characters)
<BR>
<BR>
*Contact Number
<input type="text" name="T2">
<BR>
<BR>
```

```
*Email Id
<input type="text" name="T3">
<BR>
<BR>
*Desired User Name
<input type="text" name="T4">
<BR>
<BR>
*Choose a Password<input type="password" name="P1">
     (It must be 5 to 15 characters)
<BR>
<BR>
*Photo
<input type="file" name="fileToUpload" id="fileToUpload">
<BR>
<BR>
<CENTER>
<input type="submit" name="submit" value="Submit">
     
<input type="reset" name="reset" value="Reset">
</CENTER>
</FORM>
</BODY>
</HTML>
```


Questions asked in semester paper

Question- What are scripting languages and why java script is used? Write a JavaScript function for validating form data like Mandatory Fields and Email field?

[2017-2018]

Question- How a Java Script works?

[2016-2017]

Question- Compare and Contrast Java and Java Script.

[2016-2017]

Question-What is the difference between Java and Java Script? Write a Java script function for e-mail address validation, that is, to check if the content has the general syntax of an e-mail or not.

[2015-2016]

Question- How do you perform client side validation using JavaScript? Illustrate with suitable example.

[2014-2015]

Question- What is difference between Java and JavaScript? How is Java strongly associated with internet? Why do you think JavaScript plays important role in web designing?

[2014-2015]

Question-Discuss in detail about JavaScript? Write a JavaScript program to find largest among 5 numbers.

[2014-2015]

Question-How event handling is done in JavaScript? Write a program to create an applet, window completely black in color to start with that becomes brighter and brighter as you go on clicking the mouse.

[2014-2015]

Question-What is the difference between Java and Java Script? Describe the strengths and weakness of JAVA Script.

[2013-2014]

Question-How do we handle in JavaScript? What is DHTML?

[2013-2014]

Question-How JavaScript can be embedded into HTML, with neat example briefly explain the form validation using JavaScript..

[2012-2013]

Question-How is array creation done in JavaScript? Write an example.

[2012-2013]

Question-What is the difference between undefined and null in Java Script?

[2012-2013]

Question- Illustrate the importance of documents, statements and functions in JavaScript and VB Script.

[2012-2013]

Question-Write a JavaScript program to find largest among 5 numbers.

[2012-2013]

Question-Write a script using Java Script that checks the given input for a valid name, password length more than 6 characters and age field in the range of 1 to 99. Also write the appropriate HTML code to implement this script.

[2012-2013]

Question-Write a Java Script function to validate user name.

[2012-2013]

Question-What are the advantages and disadvantages of scripting language? Write a script for any application using java scripting language.

[2012-2013]

Question-What is the difference between JAVA and JAVA SCRIPT? How is Java strongly associated with internet? Draw a flowchart to show how various Java tools are used in application development.

[2011-2012]

Question-Write a program in Java script to build up a clock.

[2010-2011]

Question- What do you mean by event? How event handling is done in JAVA SCRIPT? Write a program to create an applet, window completely black in colour to start with that becomes brighter and brighter as you go on clicking the mouse.

[2010-2011]

Question-What do you mean by an event? How event handling is done in Java Script? Explain with suitable examples.

[2007-2008]

Question-Write a Java Script function int find Sum Square(f), to find the sum of the squares of the first n integers. The number n is the value of an element called number in the form f.

[2007-2008]

Question-Differentiate the following pair: Java and Java Script

[2007-2008]

Question-Write a JavaScript function int findsum (f) to find the sum of the first n prime integers. The number n is the value of an element called number in the form f.

[2006-2007]

Question-What is Java Script? Why do you think java script plays important role in web designing?

[2005-2006]