

4.7 UNDERSTAND THE PROCESS OF DEBUGGING JAVASCRIPT CODE

- ✓ A bug can be defined as a mistake made in program or script. This mistake can be identified and fixed by a process called debugging.
- ✓ Different methods of debugging java script as follows:
 1. By using java script validator.
 2. By using java script debugger.
 3. By adding debugging code to the program.

By using java script validator

- ✓ The java script is run by passing it through the program known as validators or validating parsers.
- ✓ This validators usually contains HTML and java script editors and they are responsible for ensuring that the code follows the syntax rules of the language.
- ✓ One of the user-friendly java script validator is Douglas crockford's javascript link which is free of cost.

By using java script debugger

- ✓ In this method, a debugger application is used that helps the programmer to control every aspect of script execution.
- ✓ With a debugger, you can also set breakpoints(places where code execution can be stopped) and examine variables while the code is executing.
- ✓ The **debugger** keyword stops the execution of java script, and calls the debugging function.
- ✓ Normally, we activate debugging in your browser with the F12 key, and select "Console" in the debugger menu.

By adding debugging code to the program.

- ✓ In this method, alert() or document. Write() methods can be used for debugging the java script code.

4.8 IMPLEMENT FUNCTIONS

- ✓ A function is a set of statements used to perform a particular task.
- ✓ Set of statements in a function are executed when the **function is called** in a program.
- ✓ This eliminates the need of writing the same code again and again.
- ✓ Functions allow a programmer to divide a big program into a number of small and manageable functions.
- ✓ In Java script, functions can be placed under the head and body sections in HTML.

4.8.1 DEFINE AND CALL A FUNCTION

Defining a function:The most common way to define a function in JavaScript is by using the **function** keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.

```
Syntax:<script type="text/javascript">  
function functionname(parameter-list)  
{  
Statements;  
}</script>
```

Calling a function

- To call a function we use the function name and place the parameters inside the function.

```
<script type="text/javascript">  
function sayHello()  
{
```

```
document.write("Hello there!");  
}  
    sayHello(); //calling function  
</script>
```

4.8.2.ILLUSTRATE PARAMETER PASSING

- ✓ Parameters are the values passed to the function while calling it.
- ✓ Number and order of arguments passed in the function call should match with the number and order of arguments that you define in the function.

Syntax:

```
<script type="text/javascript">  
function functionname(parameter_name 1,parameter_name2)  
{  
Block of Statements;  
}</script>
```

There are two types of parameter passing

- 1.Functions without parameters
- 2.Functions with parameters.

1.Functions without parameters

- ✓ It does not contain any parameters.
- ✓ The code written in these functions is static, implying that the values of members of these functions cannot be changed.

sample program

```
<script type="text/javascript">  
function total()  
{  
    var a=5,b=6,c=7;  
    var sum=a+b+c;  
    document.write ("Total sum is" +sum);  
}  
total(); //calling function with out parameters  
</script>
```

2.Functions with parameters

- ✓ It contains parameters in its parenthesis.
- ✓ Values of members of these functions can be changed in run time by passing different values to their parameters.
- ✓ Generally, parameters can be passed in two ways. A)pass by value and B)pass by reference.
- ✓ Java script supports only pass by value method.

sample program

```
<script type="text/javascript">  
function total(a,b,c)  
{  
    var sum=a+b+c;  
    document.write ("Total sum is" +sum);  
}
```

```
total(4,5,6); //calling function with parameters
</script>
```

Return statement

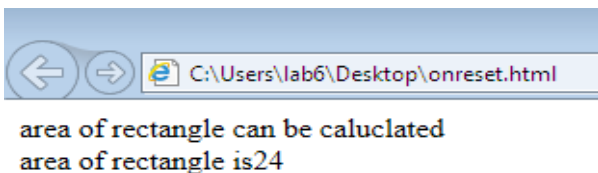
- ✓ A return statement specifies the value that is returned from a function.
- ✓ The execution of a return statement is the final act of the function.
- ✓ After the execution of return statement, control of execution passes back to calling function.

Syntax: return value;

Write a Java script program to find area of rectangle using functions

```
<html>
<head>
<script type="text/javascript">
function area(l,b)
{
    var rectangle=l*b;
    return rectangle;
}
document.write("area of rectangle can be caluclated<br>");
var rect=area(4,6);
document.write("area of rectangle is"+rect);
</script></head></body></html>
```

Output



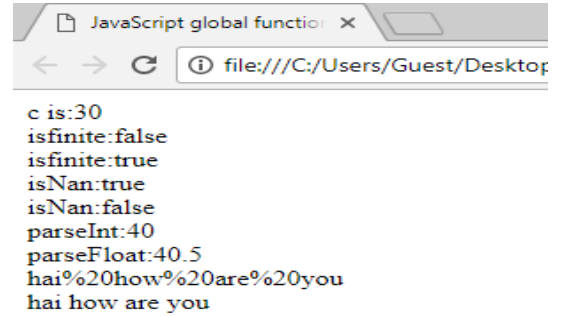
4.8.3 LIST AND EXPLAIN GLOBAL FUNCTIONS USED IN JAVASCRIPT

- ✓ **Alert():** Displays information in message box. This function displays an error message box when you validate a form
Ex: alert("Hai");
- ✓ **Prompt():** Displays a message consisting of **ok** and **cancel** button. This function returns a text **string** when the **Ok** button is clicked and **null** is returned when **Cancel** is clicked.
Ex: prompt("hai", "");
- ✓ **Confirm():** Displays a message box with two buttons **ok** and **cancel** buttons. When you click on **OK** button the function returns true. When you click on **cancel** button the function returns false.
- ✓ **Number():** It converts a value of an object in to a number. If object contains Boolean value, then it returns 1-true, 0-false.
- ✓ **Eval():** It evaluates and executes a string and returns a result (evaluation function).
Ex: eval("x*y");
- ✓ **Isfinite():** It returns a boolean value true or false indicating whether the argument passed to is finite (or) infinite.
Ex: isFinite("2"); // output is **false**
- ✓ **IsNaN():** It determines the entered value is illegal number or not. **Nan**-not a number
Ex: isNaN("ab"); //output is **true**
- ✓ **parseInt():** This function parses the string and returns the first integer found in string.
Ex: parseInt("10.45") result=10;
- ✓ **parseFloat():** This function parses the string and returns the first float found in string.
Ex: parseInt("60.5 number") result=60.5;

- ✓ **Escape():** Encodes a string or Encrypt the string.
- ✓ **Unescape():** Decodes a string that is encoded by escape() function.

Example program:

```
<html>
<head>
<title>JavaScript global functions</title>
</head>
<body>
<script type="text/javascript">
vara=5,b=2;
var c=eval("a*b")+eval("a+5")+eval("5+5");
document.writeln("c is "+c+"<br>");
document.write("isfinite"+isFinite("hello")+"<br>");
document.write("isfinite"+isFinite(123)+"<br>");
document.write("isNan"+isNaN("hai")+"<br>");
document.write("isNan"+isNaN(2)+"<br>");
document.write("parseInt"+parseInt("40.5 years")+"<br>");
document.write("parseFloat"+parseFloat("40.5 years")+"<br>");
document.write(escape("hai how are you")+"<br>");
document.write(unescape("hai how are you")+"<br>");
</script>
</body>
</html>
```



4.8.4 SCOPE AND LIFETIME OF VARIABLE:

Scope of variable: The scope of a variable can be defined as a process of locating (or) referencing the variable it from undesired modifications. (or) It is an area within a function and the accessibility in a program.

Scopes are two types

1. Local scope
2. Global scope

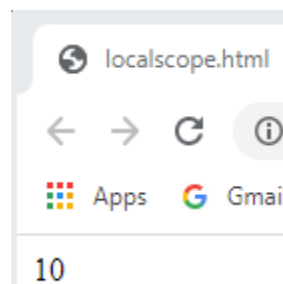
Local scope

- ✓ Scope of a variable is limited to the block of code.
- ✓ The variables that are declared inside a block are not visible from outside of a block.
- ✓ This means the scope of these variables is only within the block.

EX:

```
<html>
<head>
<title>local</title>
</head>
<body>
<script type="text/javascript">
function f1()
{
var a=10;
document.writeln(a);
}
function f2()
{
document.writeln(a);
}
f1();
f2();
```

OUT PUT:



```
</script>
</body>
</html>
```

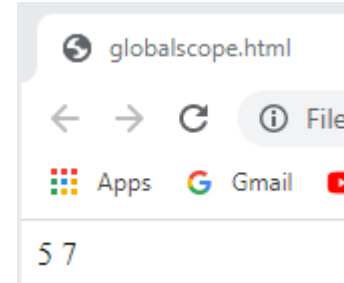
Global scope

- ✓ Variable that are declared outside of a block (or) function are said to have global scope.
- ✓ Variable which can be declared in the beginning of method can be accessible throughout the script (or) program.

EX:

```
<html>
<head>
<title>global</title>
</head>
<body>
<script type="text/javascript">
var a=5;
function f1( )
{
document.writeln(a);
}
function f2( )
{
a=a+2;
document.writeln(a);
}
f1();
f2();
</script>
</body>
</html>
```

OUT PUT:



Lifetime of variable: Lifetime is also called as duration. It is a period in which a variable is available in memory.

The duration can be of two types

a. **Automatic duration**

- ✓ It is a method of conversion of memory as it creates the variable when the program control enters into the function and destroys the variable when program control exists the function.
- ✓ The variable that are local have automatic duration.

b. **Static duration:**

- ✓ Variable that are defined in the <head> section which is available starting from </head> section until the browser is terminated.
- ✓ Variables are available even after closing the <head> section but they are not allowed to be used in entire script.
- ✓ Variables that are global variables have static duration.

4.8.5 WRITE SMALL PROGRAMS USING RECURSION

Recursion: A function that calls itself is called recursive function.

sample program using recursion-[factorial]

```
<html>
<head>
<title> recursion </title>
```

```

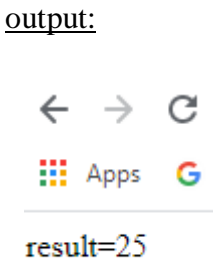
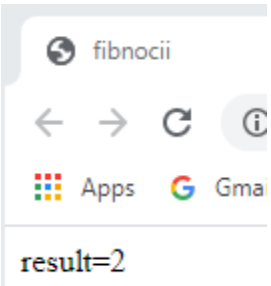
</head>
<body>
<h1> this is recursion program on java script </h1>
<script type="text/JavaScript">
function fact (num)
{
if (num>1)
return num*fact (num-1);
else
return num;
}
var res=fact (5);
document.write ("factorial of 5 is="+res);
</script>
</body>
</html>

```

OUTPUT



sample program on x power y and Fibonacci

X power y	fibonacci
<pre> <html> <head> <title>x power y</title> </head> <body> <script type="text/javascript"> function power(x,y) { if(y==0) { return 1; } else { return(x*power(x,y-1)); } } var x,y,result; x=prompt("enter value of x"); y=prompt("enter value of y"); result=power(x,y); document.writeln("result="+result); </script> </body> </html> </pre> <p style="text-align: right;"><u>output:</u></p> 	<pre> <html> <head> <title>fibonacci</title> </head> <body> <script type="text/javascript"> var n,result; n=prompt("enter n value"); result=fib(n); function fib(n) { if(n==1 n==2) return 1; else return(fib(n-1)+fib(n-2)); } document.writeln("result="+result); </script> </body> </html> </pre> <p style="text-align: right;"><u>output:</u></p> 

4.9 IMPLEMENT ARRAYS

- ✓ Array is used for storing different data in continuous location.
- ✓ In java script array can be hold different types of data types in a single array.

4.9.1 explain single and multi-dimensional array

- **Single dimensional array:** onedimensional array have a only one column. these array will have only one subscript while in use.

Array can be created in the following ways

- 1) Using array constructor.
- 2) Using array literal notation.

Array declaration: Using array constructor

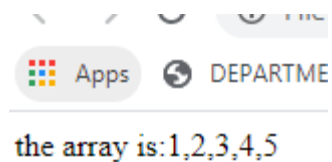
```
var arrayname=new Array(array size);  
var myarray=new Array(5);
```

creaing array elements

```
var myarray=new Array("sun",2,"tues",4);
```

sample program

```
<html>  
<head>  
<title>arrays</title>  
</head>  
<body>  
<script type="text/javascript">  
var arr=new Array(1,2,3,4,5);  
document.writeln("the array is:"+arr);  
</script>  
</body>  
</html>
```



Array declaration: Using array literal

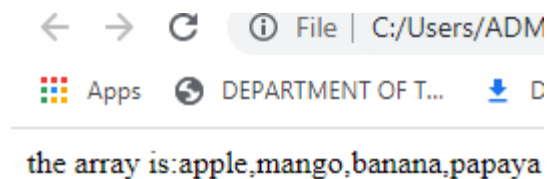
```
var arrayname=[];  
var myarray=[];
```

creaing array elements

```
var myarray=["sun",2,3,"weds"];
```

sample program

```
<html>  
<head>  
<title>arrays</title>  
</head>  
<body>  
<script type="text/javascript">  
var arr=["apple","mango","banana","papaya"];  
document.writeln("the array is:"+arr);  
</script>  
</body>  
</html>
```



- **Multi dimensionalarray:** It contains more than one columns and we also use multiple subscripts in array.

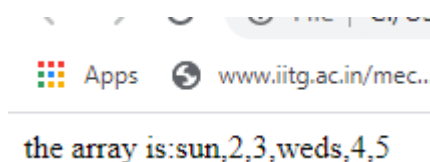
```
Var myarray=[][];
```

creaing array elements

```
var myarray=[[ "sun",2],[3,"weds"]][4,5];
```

sample program:

```
<html>  
<head>  
<title>arrays</title>  
</head>  
<body>  
<script type="text/javascript">  
var myarray=[[ "sun",2],[3,"weds"],[4,5];  
document.writeln("the array is:"+myarray);
```



</script>

</body>

</html>

4.9.3 Methods of Array object(or)Array Manipulation methods

Method

Description

Concat() -> Joins two or more arrays and returns the joined array.

Join() -> Joins all the elements of an array into a string.

Reverse() -> Reverse the order of list of elements in array.

Sort() -> Sorts the elements of an array.

Length() -> Count the number of elements in an array.

toString() -> Converts an array to a string of comma separator.

Pop() -> Removes the last element of array.

Push() -> Insert the new element into array as last element of array.

Shift() -> It removes the first element of array.

Push() -> Insert the new element into array as first element of array.

4.9.4 Sample program using array

```
<html>
```

```
<head>
```

```
<title>array</title>
```

```
</head>
```

```
<body>
```

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

```
myarray1=new Array("sunday","monday","tuesday","wednesday","thursday");
```

```
myarray2=new Array(1,2,3,4,5);
```

```
document.write("Array 1:"+myarray1+"<br/>")
```

```
document.write("Array 2:"+myarray2+"<br/>")
```

```
document.write("concatenated two arrays: "+myarray1.concat(myarray2)+"<br/>");
```

```
document.write("joined two arrays:"+myarray1.join(myarray2)+"<br/>");
```

```
document.write("Shifted array 1:"+myarray1.shift()+"<br/>");
```

```
document.write("sorted array2:"+myarray2.sort()+"<br/>");
```

```
document.write("Shifted array 2:"+myarray2.push(10)+"<br/>");
```

```
document.write("pop array 2:"+myarray2.pop()+"<br/>");
```

```
document.write("reversed array2:"+myarray2.reverse()+"<br/>");
```

```
</script>
```

```
</body>
```

```
</html>
```

Output:


```
Array 1:sunday,monday,tuesday,wednesday,thursday
Array 2:1,2,3,4,5
concatenated two arrays: sunday,monday,tuesday,wednesday,thursday,1,2,3,4,5
joined two arrays:sunday1,2,3,4,5monday1,2,3,4,5tuesday1,2,3,4,5wednesday1,2,3,4,5thursday
Shifted array 1:sunday
sorted array2:1,2,3,4,5
Shifted array 2:6
pop array 2:10
reversed array2:5,4,3,2,1
```

Addition of two arrays

```
<html>
<head>
<body>
<script type="text/javascript">
Var a=[1,2,3,4,5];
Var b=[6,7,8,9,1];
Var c=[];
C[0]=a[0]+b[0];
C[1]=a[1]+b[1];
C[2]=a[2]+b[2];
document.write("Array is:"+c+"<br/>")
</script>
</body>
</html>
```

Sorting an array

```
<html>
<body>
<script type="text/javascript">
Var a=[1,2,3,4,5];
a.sort();
document.write("Sorting Array is:"+a);
</script>
</body>
</html>
```

4.10 LIST VARIOUS OBJECTS PROVIDED BY JAVASCRIPT

- ✓ JavaScript is an object based scripting language that is used to develop object –oriented web applications.
- ✓ Object Oriented programming is based on the concepts of classes, objects.
- ✓ In Java script objects are of two types
 - Built in objects(Number,Array,Boolean,String,Math,Date)
 - User defined objects

1.String object

- ✓ A string is a sequence of characters. In java script all strings are represented as instances of the string object.

METHOD	DESCRIPTION
length	Returns the length of string
charAt()	Returns character in specified index
concat()	Combines the text of two strings and returns a new string.
indexOf()	Return the position of first occurrence of the specified character in a string.

lastIndexOf()	Return the position of last occurrence of the specified character in a string.
Search()	Searches for a match between regular expression and a string, and returns a positions of the matches.
Slice(start,end)	Returns a part of a string as new string.
replace()	Used to find a match between a regular expression and a string, and to replace the matched substring with a new substring.
split()	Splits a String object into an array of strings by separating the string into substrings.
substr(start,end)	Returns the characters in a string beginning at the specified location through the specified number of characters.
toLowerCase()	Returns the calling string value converted to lower case.
toUpperCase()	Returns the calling string value converted to upper case.

Sample program:

```

<html>
<head>
<title>JavaScript String Method</title>
</head>
  <body>
    <script type="text/javascript">
      Var str = new String( "This is string" );
      document.writeln("str.charAt(0) is:" + str.charAt(0)+"<br>");
      document.write("length is:" + str.length+"<br>");
      document.write("lower case is:"+str.toLowerCase( )+"<br>");
      document.write("upper case is:"+str.toUpperCase( ));
      document.write("substr is:"+str.substr(1,2)+"<br>");
      document.write("split is:"+str.split('web'));

    </script>
  </body>
</html>

```

Output:



2.DateObject: Date object is used to display date on a web page.

Syntax: var date1=new Date();

Method	Description
getDate()	Returns the day of the month for the specified date according to local time.
getDay()	Returns the day of the week for the specified date according to local time.

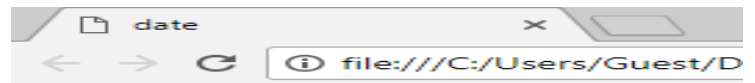
getFullYear()	Returns the year of the specified date according to local time.
getHours()	Returns the hours range from 0 to 23
getMinutes()	Returns minutes the ranges from 0 to 59
getMilliseconds	Returns milliseconds the ranges from 0 to 999
getMonth()	Returns the numerical equivalence of month that ranges from 0 to 11
setTime()	Set the time in webpage.
setDate()	Sets the day of the month for a specified date according to local time.
setFullYear()	Sets the full year for a specified date according to local time.
setMonth()	Sets the month ranges from 0 to 11
setHours()	Sets the hours for a specified date according to local time.
setMinutes()	sets minutes the ranges from 0 to 59
setMilliseconds	sets milliseconds the ranges from 0 to 999

Sample programs(getmethods)

```

<html>
<head>
<title>date</title>
</head>
<body>
<h1>program in java script using date
function</h1>
<script type="text/javascript">
var a=new date();
document.write("date is"+a.getDate()+"<br>");
document.write("day is"+a.getDay()+"<br>");
document.write("fullyear
is"+a.getFullYear()+"<br>");
document.write("hours is"+a.getHours()+"<br>");
document.write("minutes is"+a.getMinutes()+"<br>");
document.write("month is"+(a.getMonth()+1)+"<br>");
document.write("seconds is"+a.getSeconds()+"<br>");
document.write("milliseconds is"+a.getMilliseconds()+"<br>");
</script>
</body>
</html>

```



Date function

```

date is10
day is6
fullyear is2018
hours is16
minutes is54
month is2
seconds is55
milliseconds is295

```

Sample programs(setmethods)

```

<html>
<head>
<title>date</title>
</head>
<body>
<h1>program in java script using date function</h1>
<script type="text/javascript">
var d=new Date();
document.write("date is"+d+"<br>");
d.setDate(25);

```

program in java script using date function

```

date isFri Jan 31 2020 09:31:04 GMT+0530 (India Standard Time)
date isSat Jan 25 2020 09:31:04 GMT+0530 (India Standard Time)
fullyear isTue Jan 25 2022 09:31:04 GMT+0530 (India Standard Time)
hours isTue Jan 25 2022 02:31:04 GMT+0530 (India Standard Time)
minutes isTue Jan 25 2022 02:35:04 GMT+0530 (India Standard Time)
month isWed May 25 2022 02:35:04 GMT+0530 (India Standard Time)
seconds isWed May 25 2022 02:35:56 GMT+0530 (India Standard Time)
milliseconds isWed May 25 2022 02:35:56 GMT+0530 (India Standard Time)

```

```

document.write("date is"+d+"<br>");
d.setFullYear(2022);
document.write("fullyear is"+d+"<br>");
d.setHours(2);
document.write("hours is"+d+"<br>");
d.setMinutes(35);
document.write("minutes is"+d+"<br>");
d.setMonth(4);
document.write("month is"+d+"<br>");
d.setSeconds(56);
document.write("seconds is"+d+"<br>");
d.setMilliseconds(45);
document.write("milliseconds is"+d+"<br>");
</script>
</body>
</html>

```

3.MathObject:Used to perform simple and complex arithmetic operations.

<u>Method</u>	<u>Description</u>
abs(x)	returns absolute value of x
ceil(x)	rounds up x to nearest biggest integer
cos(x),sin(x),tan(x)	returns respective trigonometric values
exp(x)	exponential value of x
floor(x)	rounds up 'x' to nearest smaller integer
pow(x,y)	returns x to the power of y
sqrt(x)	square root of x
random(x)	returns random value between 0 and 1
min(x,y,z)	returns the minimum value among x,y and z
max(x,y,z)	returns maximum value among x,y and z

Sample program

```

<html>
<head>
<title>math object</title>
</head>
<body>
<script type="text/javascript">
document.write("floor value is:"+Math.floor(12.6)+"<br/>");
document.write("square root of 16 is "+Math.sqrt(16)+"<br/>");
document.write("round value is:"+Math.round(12.6)+"<br/>");
document.write("power valu is:"+Math.pow(10,2)+"<br/>");
document.write("Maximum value is:"+Math.max(12,3,4,15,6)+"<br/>");
document.write("Minimum value is:"+Math.min(12,34,15,65)+"<br/>");
</script>
</body>
</html>

```



```

floor value is:12
square root of 16 is 4
round value is:13
power valu is:100
Maximum value is:15
Minimum value is:12

```

4.Boolean Object

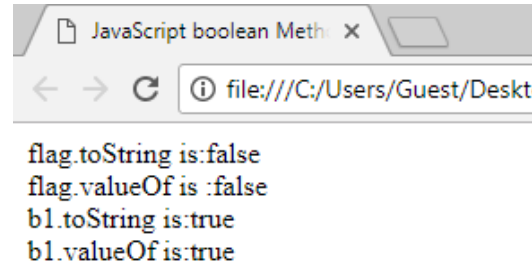
- ✓ It is used to convert non Boolean values into Boolean values. The **Boolean** object represents two values, either "true" or "false".
- ✓ Boolean object returns false when the object is passed with values such as **0,false,null**.

✓ Var bool1=new Boolean(num);

Method	Description
toString()	Returns a string of either "true" or "false" depending upon the value of the object.
valueOf()	Returns the primitive value of the Boolean object.

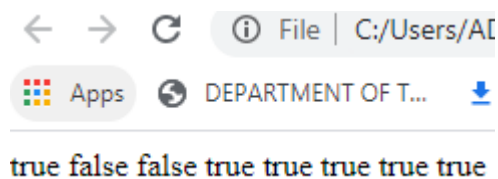
Sample program

```
<html>
<head>
<title>JavaScript boolean Method</title>
</head>
<body>
<script type="text/javascript">
var flag = new Boolean(false);
var b1 = new Boolean("hai");
document.write( "flag.toString is : " +
flag.toString()+"<br>");
document.write( "flag.valueOf is : " + flag.valueOf()+"<br>");
document.write( "b1.toString is : " + b1.toString()+"<br>");
document.write( "b1.valueOf is : " + b1.valueOf());
</script>
</body>
</html>
```



Sample program

```
<html>
<head>
<title>date</title>
</head>
<body>
<script type="text/javascript">
var b1=new Boolean(true);
var b2=new Boolean(false);
var b3=new Boolean(0);
var b4=new Boolean(1);
var b5=new Boolean(6);
var b6=new Boolean(-9);
var b7=new Boolean("hai");
var b8=new Boolean(" ");
document.write(b1+" "+b2+" "+b3+" "+b4+" "+b5+" "+b6+" "+b7+" "+b8);
</script>
</body>
</html>
```



5.NumberObject: It is used to perform any numerical functions like exponential, fixed, precision.

Method	Description
toExponential()	Converts a number into a exponential notation
toFixed()	Rounds up a number to x digits after the decimal.

toFixed()	Rounds up a number to a length of x digits
toString()	Returns a string value for the number object
valueOf()	Returns a primitive value for the number object

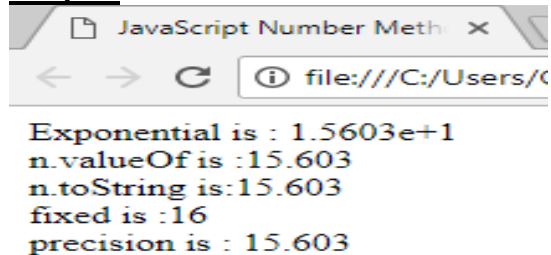
Sample Program:

```

<html>
<head>
<title>JavaScript Number Method</title>
</head>
<body>
<script type="text/javascript">
var n= new Number("15.603");
document.write( "Exponential is : " + n.toExponential()+"<br>");
document.write( "n.valueOf is : " + n.valueOf()+"<br>");
document.write( "n.toString is : " + n.toString()+"<br>");
document.write( "fixed is : " +n.toFixed()+"<br>");
document.write( "precision is : " +n.toPrecision(3)+"<br>");
</script>
</body>
</html>

```

Output:



Exploring pop up boxes

A popup box is a window that displays a message along with an ok button the popup box may also contains a cancel button. In addition it can prompt users to enter some text javascript supports three types of popup boxes

The alert box

The confirm box

The prompt box

Alert box: it is generally used to display an alert message while javascript code executing

Confirm box: it is used to display a message as well as return a true or false value .if it displays a dialogue box with two buttons ok and cancel

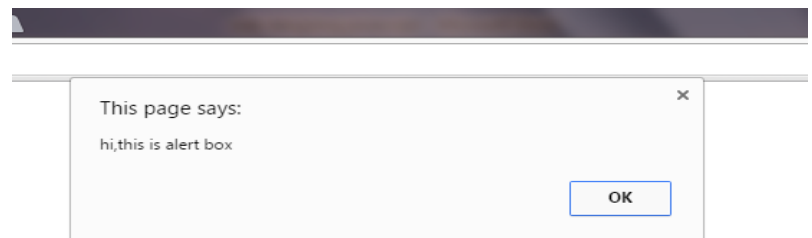
Prompt box: it is used to input a value from a user. it contains text box and ok and cancel buttons

ALERT BOX

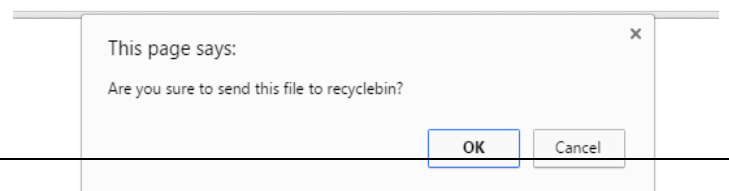
```

<html>
<head>
<title>alert box</title></head>
<body bgcolor="green">
<h1> using alert box </h1>
<script type="text/javascript">
alert("hi,this is alert box");
</script>
</body>
</html>

```



CONFIRM BOX



```

<html>
<head>
<title>confirm box</title>
</head>
<body>
<h1> using confirm box </h1>
<script type="text/javascript">
varqry=confirm("Are you sure to send this file to recyclebin?");
if(qry)
{
document.write("welcomen to javascript<br/>you have moved the file to recyclebin");
}
else
document.write("welcomen to javascript<br/>the file is not move to recyclebin");
</script>
</body>
</html>

```

Prompt box

```

<html>
<head>
<title> prompt box </title>
</head>
<body>
<h1> using prompt box </h1>
<script type="text/javascript">
var name=prompt("enter your name");
if(name==null||name==" ")
{
document.write("you have not entered name");
}
else
{
document.write("hi"+name+"welcome to javascript<br/>");
}
</script>
</body>
</html>

```

