3. ADVANCED JAVASCRIPT

Index:

- Features of JavaScript, difference between client side scripting and server side scripting.
- Looping structures.
- DOM Objects and window object in JavaScript.
- Inbuilt objects such String, Math, Array, Date and Number with its properties and Methods.
- Simple JavaScript programs to do validations and user interaction.

From:
Mrs. Shailaja Rajesh Pande (IT Teacher)
S. B . Patil college of Science & Commerce, Ravet

Introduction

- Javascript is a client side scripting language. It is used to develop dynamic website.
- Program is a set of instructions used to produce various kinds of outputs
- JavaScript is an interpreted scripting language. An interpreted language is a type of programming language that executes its instructions directly and freely.
- JavaScript was initially created to "make webpages alive". The programs in this language are called scripts.
- Javascript is a Case sensitive language.

- **3.1.1 Features of JavaScript**

- JavaScript is light weight scripting language because it does not support all features of object oriented programming languages.
- No need of special software to run JavaScript programs
- JavaScript is object oriented scripting language and it supports event based programming facility.

3.1.1 Features of JavaScript

- JavaScript helps the browser to perform input validation without wasting the user's time by the Web server access.
- It can handle date and time very effectively.
- Most of the JavaScript control statements syntax is same as syntax of control statements in other programming languages.
- An important part of JavaScript is the ability to create new functions within scripts. Declare a function in JavaScript using function keyword.
- Software that can run on any hardware platform (PC, Mac, SunSparc etc.) or software platform (Windows, Linux, Mac OS etc.) is called as platform independent software. JavaScript is platform independent scripting language.
- Due to different features Javascript is Known as Universal client side scripting language.

TYPES OF SCRIPTING LANGUAGE

There are two types of scripting language.

1] Client – Side scripting language

In this type script resides on client computer (i.e. Browser) and that can run on client . It is placed inside on HTML document

Ex. – Javascript , Vbscript

2] Server-side scripting language

In this type script resides on web server and it is activated by client.

Ex.- PHP, ASP

3.1.2 Difference between Server side scripting and client side scripting

Server Side scripting	Client side scripting
1. Server-side scripting is used at the backend, where the source code is not visible or hidden at the client side (browser). Server-side scripting is more secure than client-side scripting	1. client- side scripting is used at the frontend which users can see source code from the browser.
2. When a server-side script is processed it communicates to the server.	2. client-side scripting does not need any server interaction.
3. programming languages such as PHP, ASP.net, Ruby, ColdFusion, Python, C# etc. are server side scripting languages.	3. The client-side scripting language involves languages such as HTML5, JavaScript etc.
4. Server-side scripting is useful in customizing the web pages and implements the dynamic changes in the websites.	4. client- side scripts are generally used for validation purpose and effectively minimize the load to the server.
5. Special software (web server software) is required to execute server-side script	5. client side scripts requires web browser as an interface.

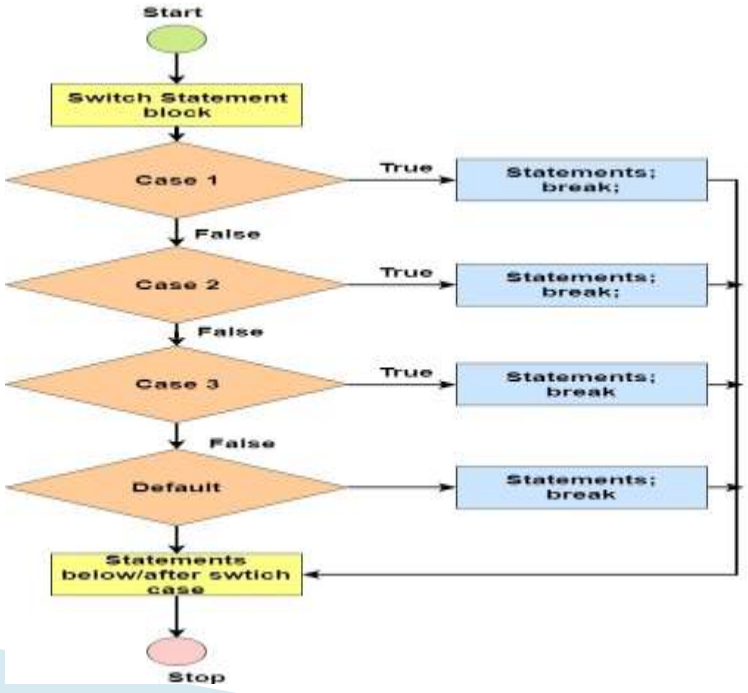
3.2 Switch case and Looping Structures

3.2.1 Switch Case statement

- The switch statement is used to perform different actions based on different conditions.
- JavaScript has a built–in multiway decision statement known as Switch. The switch statement test the value of given expression against a list of case values and when match is found, a block of statement associated with that case is executed.
- There should not be duplicity between the cases.
- The value for the case must be similar data type as the variable in switch.
- The default statement is not mandatory.

Syntax :

```
switch(expression)
{
case value1:
statement block 1;   break;
case value2:
statement block 2; break;
……..
case value n:
statement block n; break;
default:
statement block ;
}
```



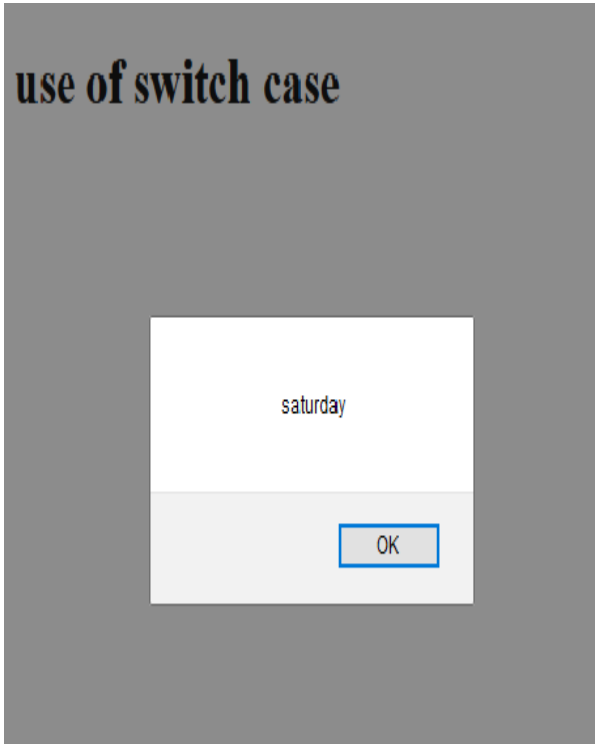
- This is how it works:
- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.
- If there is no match, the default code block is executed.

BREAK Statement

1] break statement is used to jump out of loop.

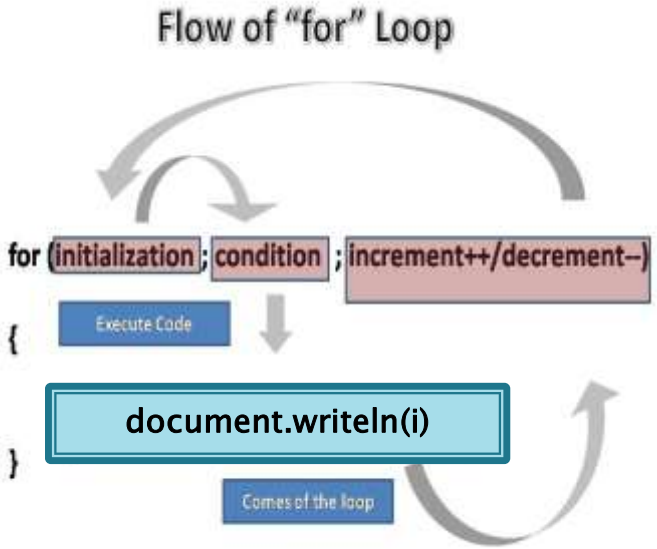
2] It is used to make an early exit from loop

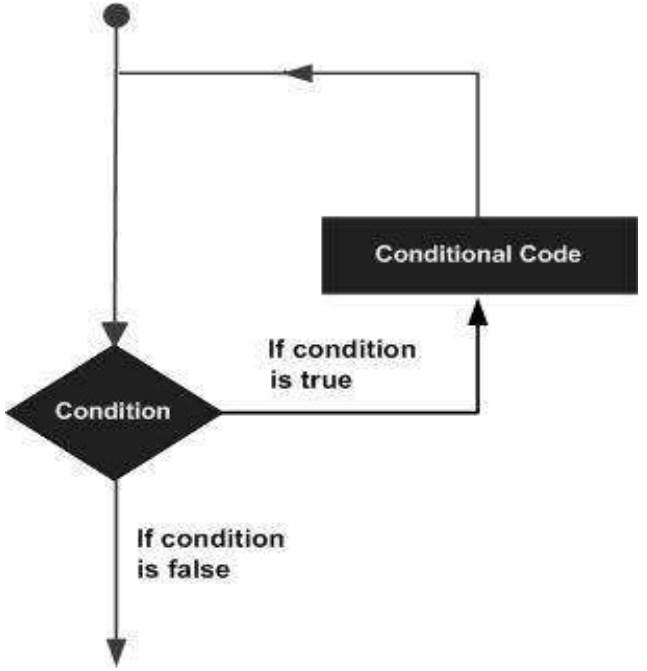
```
<!DOCTYPE html>
<head><title>Javascript Program
</title></head>
<body>
<h1> use of switch case </h1>
<script  type="text/javascript">
var day=6;
switch(day)
{
case 1: alert(“Monday”); break;
case 2: alert(“Tuesday”); break;
case 3: alert(“Wednesday”); break;
case 4: alert(“Thursday”); break;
case 5: alert(“Friday”); break;
case 6: alert(“Saturday”); break;
case 7: alert(“Sunday”); break;
default: alert("Invalid day");
}
</script></body></html>
```



3.2.2 Looping statement

- Iteration refers to the execution of statement or a group of statements of code for a fixed number of times or till the condition is satisfied.
- The condition should be Boolean condition.
- Some commonly used JavaScript looping statements are:
  1)For loop
  2)While loop



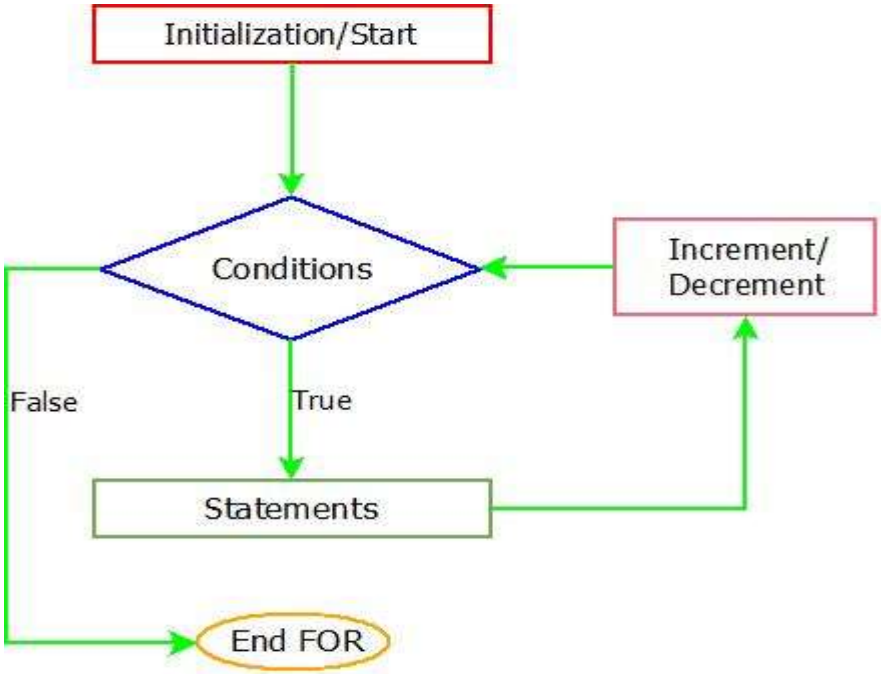


1. for…….loop

This loop executes statements as long as condition becomes true, for-loop is that it combines initialization, condition and loop iteration (increment or decrement) in single statement.

Syntax :

```
for(initialization;condition;iteration)
{
statement block;
}
```



Note : 'language' attribute of <Script> is replaced by 'type' attribute in all the programs as it is standardized.

3.2.2 example of for loop

```
<!DOCTYPE html>
<head><title>Javascript for loop</title></head>
<body><h1> use of for loop</h1>
<script type="text/javascript">
for(i=1;i<=5;i++)
{
document.writeln(i);
}
</script>
</body>
</html>
```

OUTPUT


```
<!DOCTYPE html>
<head><title>Javascript for loop</title></head>
<body><h1> use of for loop</h1>
<script type="text/javascript">
for(i=5;i>=1;i--)
{
document.writeln(i);
}
</script>
</body>
</html>
```

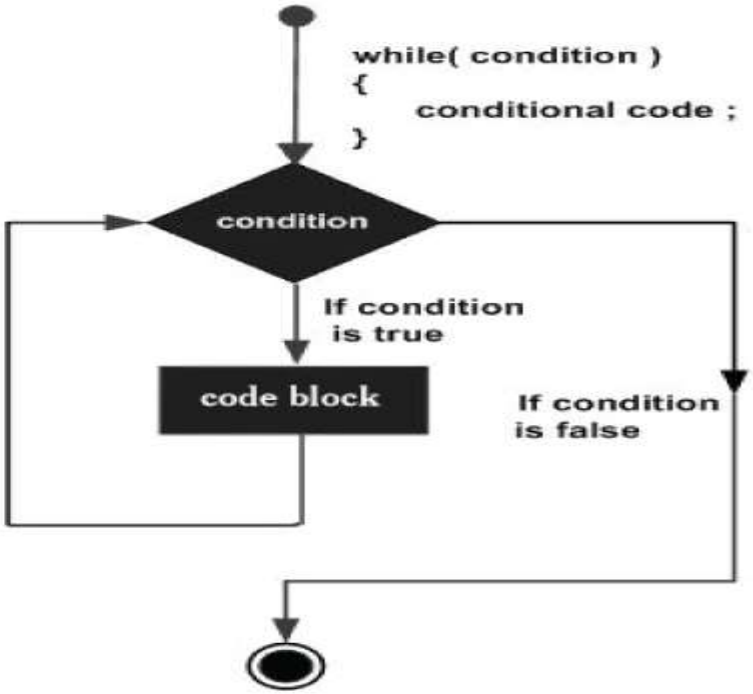
OUTPUT

2. While.....loop

This loop executes statements as long as the condition is true. As soon as condition becomes false control comes out of the loop.

Syntax:

initialization; while(condition)
{
statement block;
}



3.1.3 example of while loop

```
<!DOCTYPE html>
<head><title>Javascript while loop</title></head>
<body><h1> use of while  loop</h1>
<script type="text/javascript">
var i=1
while(i<=5)
{
document.writeln(i);
 i=i+1;
}
</script></body> </html>
```

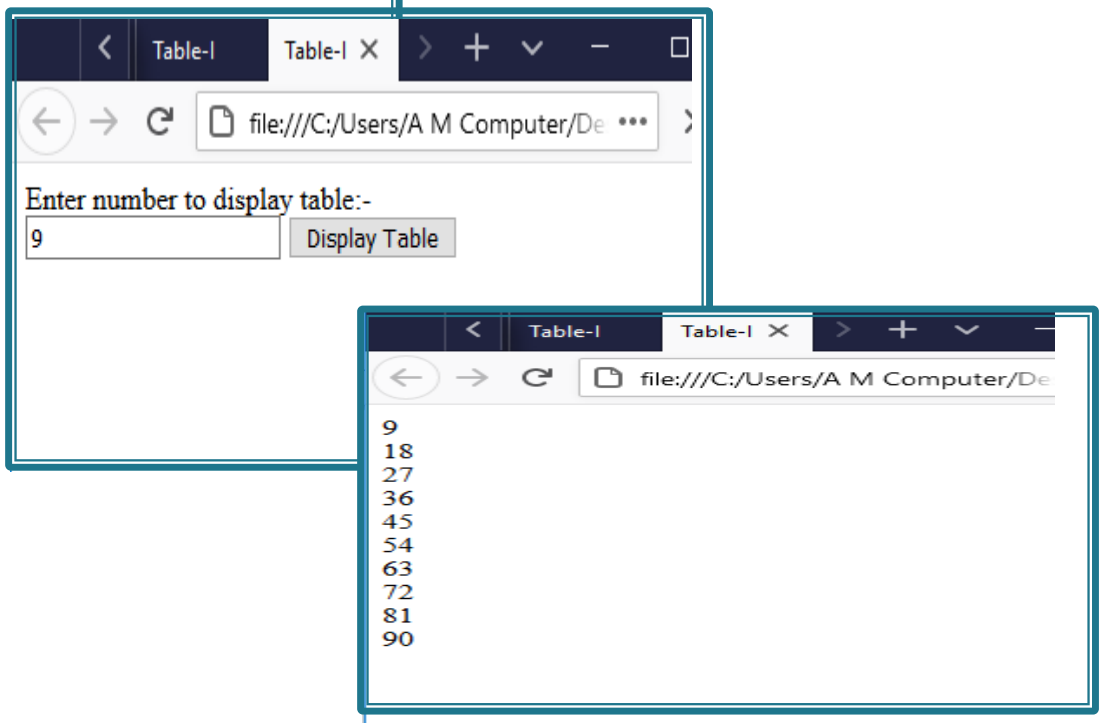
OUTPUT


```
<!DOCTYPE html>
<head><title>Javascript while loop</title></head>
<body>   <h1> use of while  loop</h1>
<script type="text/javascript">
var i=5
while(i>=1)
{
document.writeln(i);
 i=i-1;
}
</script>  </body>      </html>
```

OUTPUT

Accept number and display the table of that number

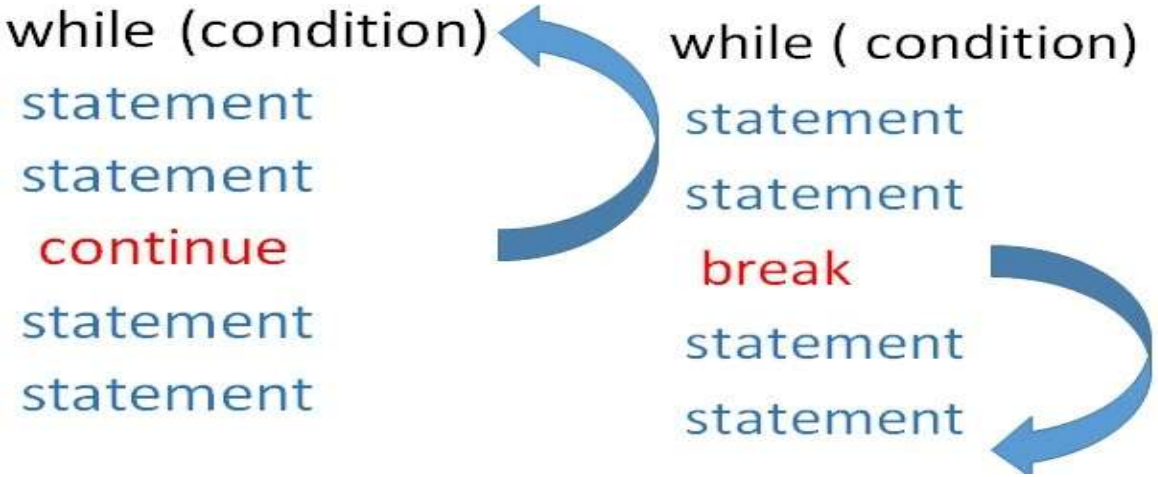
```
<!DOCTYPE html>
<head><title>Table-I</title>
<script type="text/javascript">
function display()
      {
var  i, a;      a=form1.t1.value
 for(i=1;i<=10;i++)
         {
         document.write(a*i + "<br/>");
         }
     }
</script></head>     <body>
Enter number to display table:-
<form name="form1">
<input type="text" name="t1">
<input type="button" value="Display Table" onClick="display()">
</body></html>
```



• Break and continue statements

- **Break statement** is used to jump out of loop. It is used to make an early exit from a loop. When keyword break is encountered inside the loop, control automatically passes to the next statement after the loop.

- **Continue statement:** In looping it may be necessary to skip statement block and take the control at the beginning for next iteration. This is done by using **'continue'** statement in JavaScript.



Note : 'language' attribute of <Script> is replaced by 'type' attribute in all the programs as it is standardized.

Break and continue statements

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Loop with break
statement</h2>
<script type="text/javascript">
var i;
for (i = 0; i < 5; i++) {
  if (i == 3)
  { break; }
  document.write("<br>"+i);
}
</script>
</body>
</html>
```

OUTPUT

JavaScript Loop with break statement

0
1
2

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Loop with continue
statement</h2>
<script type="text/javascript">
var i;
for (i = 0; i < 5; i++) {
  if (i ==3)
  { continue;}
  document.write("<br>"+i);
}
</script>
</body>
</html>
```

OUTPUT

JavaScript Loop with continue statement

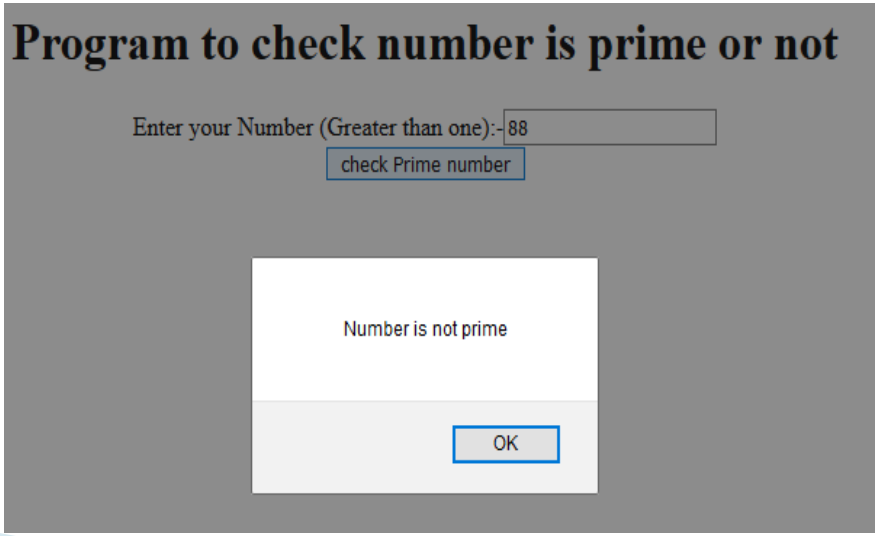
0
1
2
4

Note : 'language' attribute of <Script> is replaced by 'type' attribute in all the programs as it is standardized.

Number is prime or not

```
<!DOCTYPE html>
<html><head><title>Prime number</title>
<script type="text/javascript"> function display()
{
var a,ans;
a=parseInt(form1.t1.value); ans=1;
for(i=2;i<a; i++)
{
if(a%i==0)
{
ans=0; break;
}
}
if(ans==1)
alert("Number is prime"); else
alert("Number is not prime");
}
</script></head>
```

```
<body>
<h1 align="center"> Program to check number is prime or not
</h1>
<form name="form1" style="text-align:center">
Enter your Number (Greater than one):-
<input type="text" name="t1"> <br>
<input type="button" value="check Prime number"
onClick="display()">
</body>
</html>
```



3.3 Objects in JavaScript

- JavaScript is an object based scripting language. Almost everything is an object in JavaScript.
- A JavaScript object is an entity having state (properties) and behavior (methods). An object can group data together with functions needed to manipulate it.
- Examples of real world objects. as table, board, television, bicycle, shop, bus, car, monitor etc. All these tangible things are known as objects.

Properties and methods of object's are accessed with '.' operator.

JavaScript supports 2 types of objects built-in objects and user defined objects.

1. **Built in objects** such as Math, String, Array, Date etc.
2. As per user requirements can **create user defined objects**
3. **The 'new' keyword is used to create new object in JavaScript.**
4. **e.g.** d= new Date();

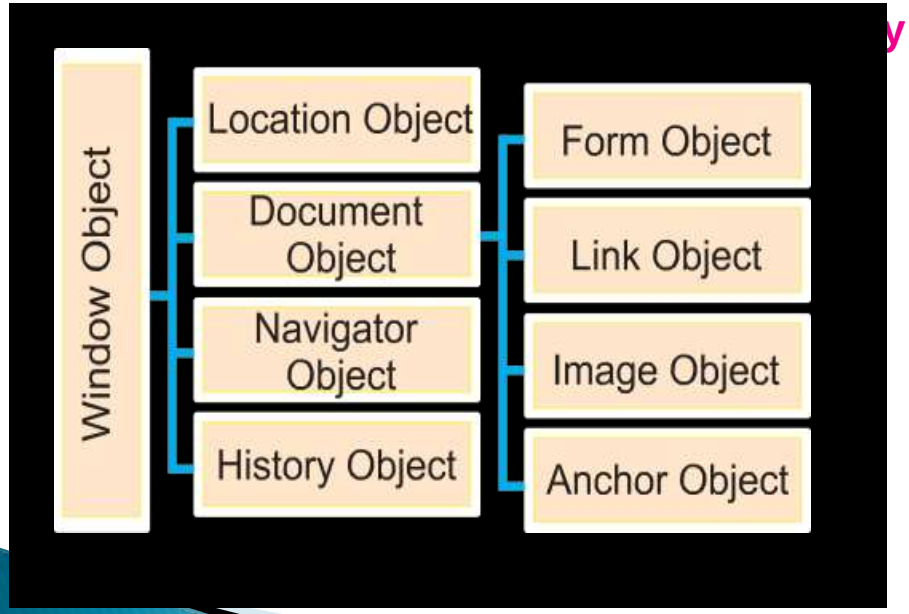
Object	Properties	Methods
car	car.name=Ferrari	car.start()
	car.model=F430	car.drive()
	car. weight=1517kg	car.brake()
	car.color=red	car.stop()

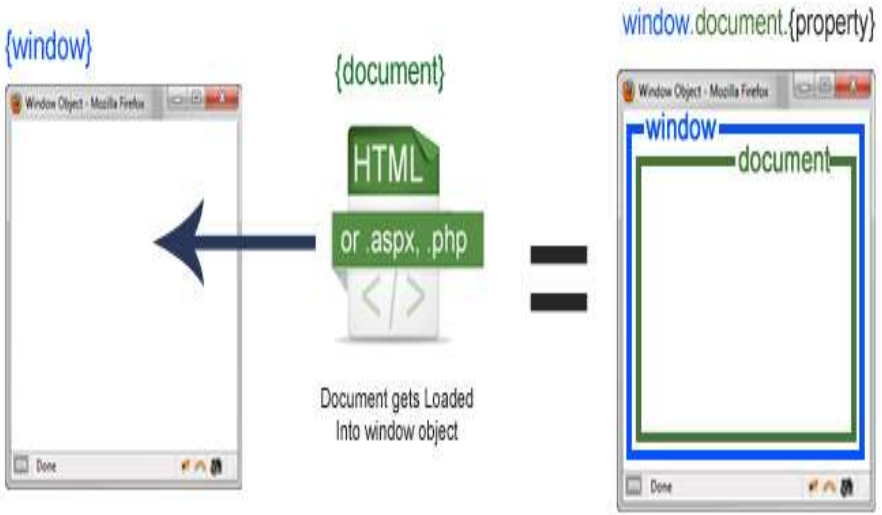
- Take an example of car object.
- It has properties like name, model, weight, color etc. and methods like start, stop, brake etc.
- All cars have same properties but contain different values from car to car.
- All cars have same methods but perform differently.

Note : 'language' attribute of <Script> is replaced by 'type' attribute in all the programs as it is standardized.

DOM (Document Object Model) :

- When HTML document is loaded into a web browser, it becomes a document
- The W3C Document Object Model (DOM) is a platform and language- neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."



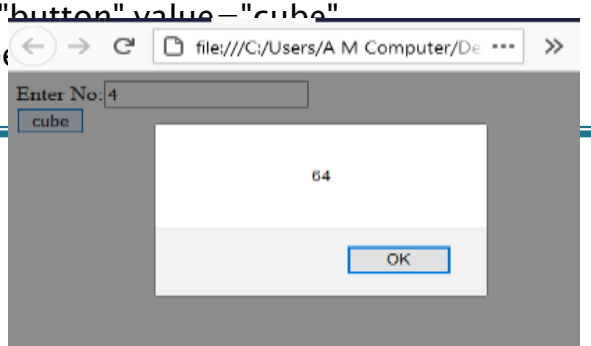


Property	Description
head	Returns the <head> element of the document
title	Sets or returns title of the document.
URL	Returns full URL of the HTML document.
body, img	Returns <body>, <img> elements respectively.
Method	**Description**
write()	Writes HTML expressions or JavaScript code to a document.
writeln()	Same as write(), but adds a newline character after each statement.
getElementById()	There are many ways of accessing form elements, of which the easiest is by getElementById() method. In which id property is used to find an element.

form1.name.value to get the value of the input value.
Instead of this,
document.getElementById() to get value of the input text.
But we need to define id for the input field.

```
<!Doctype html>
<html>
<head>
<script type="text/javascript">
   function getcube(){
   var
number=document.getElementById("number").value;
   alert(number*number*number);
   }
   </script>
</head>
<body>
   <form>
   Enter No:<input type="text" id="number"
name="number"/><br/>
   <input type="button" value="cube"
onclick="getcube
   </form>
</body>
</html>
```

output



The innerHTML Property : used to write the dynamic html on the html document.

The innerHTML property is useful for getting html element and changing its content. The innerHTML property can be used to get or change any HTML element, including <html> and <body>.

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
 function changeText3()
{
var style="<h1 style= 'color:green'>";
var text="Welcome to the HTML5 and Javascript";
var closestyle="</h1>";
document.getElementById('para').innerHTML=style+text+closestyle;
}
</script></head>
<body style="background-color:cyan">
<h1 align="center">
<p id="para">Welcome to the site</p>
<input type="button" onclick="changeText3()" value="click this
button to change above text">
</h1>
</body>
</html>
```

Output

Before button click

After button click

Window Object :

- At the very top of the object hierarchy is the window object. Window object is parent object of all other objects. It represents an open window in a browser.
- An object of window is created automatically by the browser. Window object represents an open window in a browser.

Following table shows some of the methods and properties for window object.

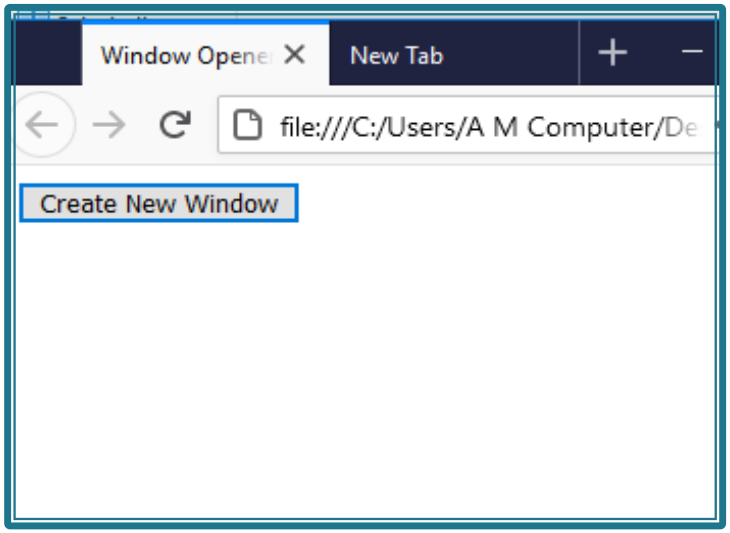
Property	Description
name	Sets or returns the name of a window.
location	Returns the Location object for the window.
document	Returns the Document object for the window.
status	Sets or returns the text in the status bar of a window.
closed	Returns a Boolean value indicating whether a window has been closed or not.
Method	**Description**
alert()	Displays the alert box containing message with ok button.
confirm()	Displays the confirm dialog box containing message with ok and cancel button.
prompt()	Displays a dialog box to get input from the user.
open()	Opens the new window.
close()	Closes the current window.
blur()	Removes focus from the current window.
focus()	Sets focus to the current window.
print()	Prints the content of current window.
setTimeout()	Calls a function or evaluates an expression after a specified number of milliseconds.

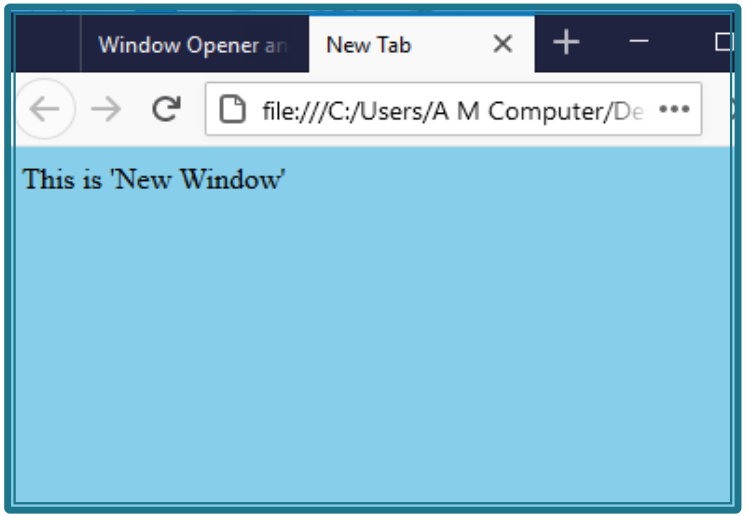
Program :

```
<!DOCTYPE html>
<html>
<head>  <title>  Window Opener and Closer </title>
<script type="text/javascript">
function makeNewWindow()
{
var newwin=window.open();
 newwin.document.write("<p>This is 'New Window'</p>");
   newwin.document.body.style.backgroundColor = "skyblue";
}
</script></head>
<body>
<form>
<input type="button" value="Create New Window"
onClick="makeNewWindow()">
</form>    </body>
</html>
```

Output

Before button click





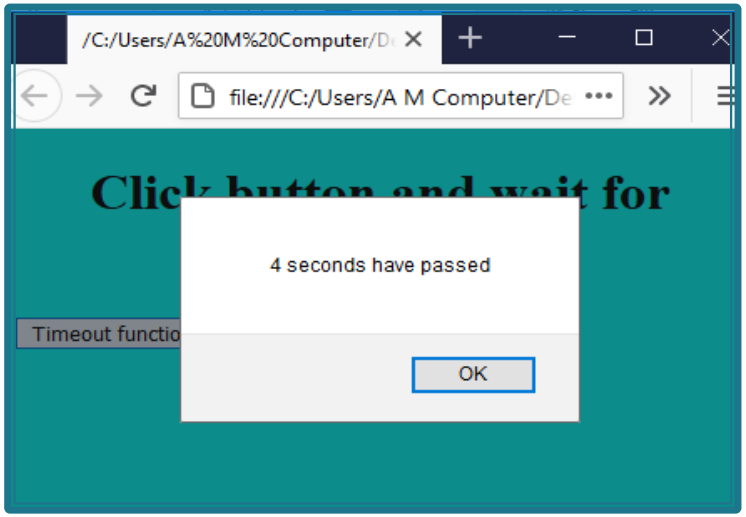
After button click

Program :

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
 function sampleFunction()
{
        window.setTimeout(next(), 4000);
    }
function next()
{
         alert("4 seconds have passed");
}
</script></head>
<body style="background -color:cyan">
<h1 align="center"> Click button and wait for message    </h1>
<input type="button" value="Timeout function"
onClick="sampleFunction()">
</body>    </html>
```

Output

Before button click



After button click

3.4 JavaScript Event

Events are actions done by the user or an application that occurs on the webpage. In previous year we studied different keyboard events (onKeypress, onKeydown, onkeyup) and mouse events (onClick, onMousemove, onMouseout, onMouseover). Similarly there are some more events used with form objects.

Event handler	Description
onblur	It occurs when user leaves field or losses focus of an element.
onfocus	It occurs when an element gets focus.
onchange	It occurs when user changes content of an element or selects dropdown value. E.g. for textbox, password, select box, textarea etc.
onselect	It occurs when user selects some text of an element.
onsubmit	It occurs when user clicks submit button.
onreset	It occurs when user clicks reset button.
onload	It occurs when page/image has been loaded.
onunload	It occurs when document/page has been unloaded or closes.

3.5 JavaScript Built-in Objects

JavaScript has several built-in or core language objects. These built-in objects are available regardless of window content and operates independently of whatever page browser has loaded. These objects provide different properties and methods that are useful while creating live web pages.

String Object :

String is used to store zero or more characters of text within single or double quotes.
String object is used to store and manipulate text.

Property	Description
length	Returns the number of characters in a string
Method	**Description**
charAt()	Returns the character at the specified position (in Number).
indexOf()	Returns the index of the first occurence of specified character in given string, or -1 if it never occurs, so with that index you can determine if the string contains the specified character.
lastIndexOf()	Returns the index of the last occurence of specified character in given string.
substr()	Returns the characters you specified: (14,7) returns 7 characters, from the 14th character.
substring()	Returns the characters you specified: (7,14) returns all characters between the 7th and the 14th.
trim()	The trim() method removes whitespace from both sides of a string
toLowerCase()	Converts a string to lower case
toUpperCase()	Converts a string to upper case

Example :

```
var str="Information Technology";
document.write ("length of string is :-" + str.length);
document.write ("Substring is :-" + str.substr (12,10));
Document.write("Character at 1st location is :"+str.charAt(1));
```

Output :

Length of string is :-22

Substring is :- Technology

Character at 1st location is : n

Example : Convert the value of txt to upper case.

```
var txt = "Hello World";
var txt1 = txt.toUpperCase();
var txt2=txt.toLowerCase();
```

Output: txt1 = HELLO WORLD

txt2= helloworld

Example :

Find the position of the character h in the string txt.

```
var txt = "abcdefghijklm";
var pos = txt. indexOf("h");
```

Output : pos= 8

Example :

```
var str = "Apple, Banana, Kiwi";
var res = str.substr(7);
```

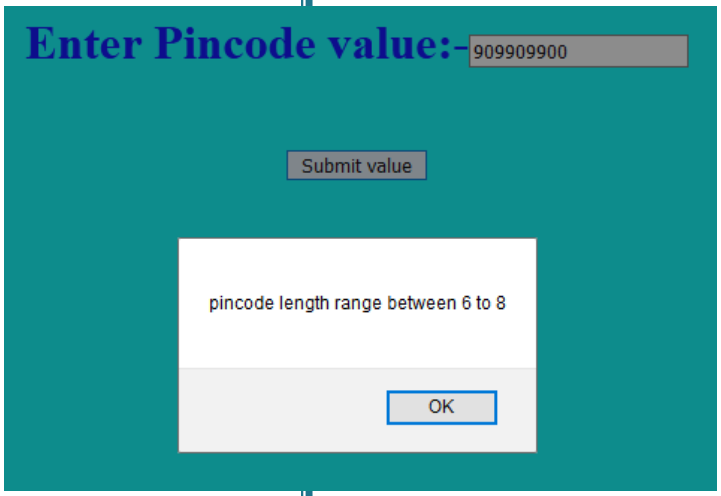
Output: res= Banana, Kiwi

Example :

```
var str = "Hello planet earth, you are a great planet.";
 var n = str.lastIndexOf("planet");
```

Output n= 36

```
<!DOCTYPE html>
<html><head><title>Pincode Validation</title></head>
<body style="color:blue;background-color:cyan">
                <form name="form1">
                <h1 align="center">
Enter Pincode value:-<input type="text" name="t1"><br><br>
<input type="button" value="Submit value" onClick="validate()">
                 </h1>
<script type="text/javascript"> function validate()
{
var pincode;
pincode=form1.t1.value;
                 if(pincode.length==0)
                                {
                                alert("please check, enter value"); form1.t1.focus();
                                }
                else if(isNaN(pincode))
                                {
                                alert("please, enter integer number only"); form1.t1.focus();
                                }
                else if(pincode.length<6||pincode.length>8)
                                {
                                alert("pincode length range between 6 to 8"); form1.t1.focus();  }
                                else
                                alert("Pincode is accepted");
                                }
</script> </body></html>
```



Summary

- JavaScript is light weight scripting language. It is platform independent language.
- There are two types of scripts; client side script and server side scripts. Client side scripts reside on client machine and server side script resides on web server.
- JavaScript provide 'switch…case' as multi way decision statement.
- For....loop, while…loop and do…while are commonly used looping structures in JavaScript.
- DOM (Document Object Model) is a programming interface for HTML and XML documents. It defines logical structure of document.
- Window object is parent object of all other objects hence its methods can be used without specifying it.
- JavaScript is event based language support objects events such as onBlur, onFocus, onChange, onSelect, onSubmit, onLoad, onUnload, onResize etc.
- JavaScript supports built-In objects such as Date, String, Math, Number and array etc. These objects contain number of properties and methods that are useful while creating interacting web pages.

Thank you !