By: N. Selvakumar
Email: anselva@yahoo.com
Web: www.nselva.com

DHTML (Dynamic HTML)

The major components of Dynamic HTML technology are:

1. CSS (Cascading Style Sheets/ Style Sheet)
2. Scripting Languages (**JavaScript**, VBScript, JScript, PerlScript, & etc.)

Scripting Languages:

Scripts that are executed when a document is loaded may be able to modify the document's contents dynamically. JavaScript is a simple programming language that can be written directly into HTML documents to allow for increased interactivity with the user. For example, JavaScript can be used to create online calculator, shows the current date and time, form validation and shows last modified date. The ability to do so depends on the scripting language itself. (Scripting Languages: JavaScript, VBScript, Jscript)

What is JavaScript?

- JavaScript is a scripting language
- A scripting language is a lightweight programming language
- A JavaScript is lines of executable computer code
- A JavaScript can be inserted into an HTML page
- JavaScript is an open scripting language that anyone can use without purchasing a license
- JavaScript is supported by all major browsers like Netscape Navigator & Internet Explorer
- JavaScript is a case sensitive language.

How does it work?

When a JavaScript is inserted into the HTML document, the Internet browser will read the HTML and interpret the JavaScript codes. The JavaScript can be executed immediately or at a later event.

What can a JavaScript do? (JavaScript gives HTML designers a programming tool)

HTML authors are normally not programmers, but JavaScript is a very light programming language with a very simple syntax. Almost anyone can start putting small piece of code into their HTML documents.

How to write JavaScript code?

Browsers that recognize JavaScript also recognize the special <SCRIPT> ... </SCRIPT> tag. This tag goes in the <HEAD> or <Body>of your HTML document.

How to Handle Older Browsers

Older browsers that do not support scripts will display the script as page content. To prevent them from doing this, you can use the HTML comment tag:

```
<script language="javascript">
<!--
   some statements
//-->
</script>
```

The two forward slashes in front of the end of comment line (//) are a JavaScript comment symbol, and prevent the JavaScript from trying to compile the line. Note that you can't put // in front of the first comment line (like //<!--), because older browsers will display it.

Ending Statements with a Semicolon?

With the traditional programming languages C++ and Java, each code statement has to end with a semicolon. Many programmers continue this habit when writing JavaScript, but in general, semicolons are optional and are required only if you want to put more than one statement on a single line.

JavaScript can put dynamic text into an HTML page

A JavaScript statement like this: document.write("<h1>Bill Gates</h1>") can write a text into the display of an HTML page, just like the static HTML text: <h1>Bill Gates</h1> does.

JavaScript can react to events

A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element. (eg: buttons, links, ….)

JavaScript can be used to validate data

JavaScripts can be used to validate data in a form before it is submitted to a server. This function is particularly well suited to save the server from extra processing.

Where to put the JavaScript:

- JavaScript in the head section - <script language="javascript"> ….. </script>
  Scripts to be executed when they are called, or when an event is triggered, go in the head section. When you place a script in the head section, you will ensure that the script is loaded before anyone uses it.

- JavaScript in the body section - <script language="javascript"> ….. </script>
  Scripts to be executed when the page loads go in the body section. When you place a script in the body section it generates the content of the page. (eg: shows today date & time)

- An external JavaScript - <script src="includefiles/jsfunctions.js"></script>
  Sometimes you might want to run the same script on several pages, without writing the script on each and every page. To simplify this you can write a script in an external file, and save it with a .js file extension.

Basic JavaScript

- Write text with JavaScript
  Eg: document.write("Hello World!")
- Format text with HTML tags
  Eg: document.write("<h1>Hello World!</h1>")

The following are examples of comments in JavaScript.

```
// This is a single line comment.
/* This is a multiple line comment with only one line. */
```

Variables

A variable is a "container" for information you want to store. A variable's value can change during the script. You can refer to a variable by name to see its value or to change its value.

Declare a variable, assign a value to it, and display it. (var is a keyword)

```
var name = "Sivakumar"
document.write(name)
x="Welcome to UOC"
document.write(x)
```

Rules for Variable names: Variable names are case sensitive, must begin with a letter or the underscore character.

Lifetime of Variables:

When you declare a variable within a function, the variable can only be accessed within that function. When you exit the function, the variable is destroyed. These variables are called local variables. You can have local variables with the same name in different functions, because each is recognized only by the function in which it is declared. If you declare a variable outside a function, all the functions on your page can access it. The lifetime of these variables starts when they are declared, and ends when the page is closed.

Functions

A function contains some code that will be executed by an event or a call to that function. A function is a set of statements. You can reuse functions within the same script, or in other documents. You define functions at the beginning of a file (in the head section), and call them later in the document.

Eg:
```
function myfunction (argument1, argument2)
{
      alert("HELLO")
}
```

- The function keyword, function name, comma-separated list of arguments to the function in parentheses
- The statements in the function in curly braces: { }

Conditionals (Conditional statements in JavaScript are used to perform different actions based on different conditions)

- if statement
- if ... else statement
- if .... else if .... else if .... else ... statement
- switch ... case statement

Looping (Looping statements in JavaScript are used to execute the same block of code a specified number of times)

- for loop - for (var i=1; i<=10; i++) { }
- while loop - while (i<10){ }
- do ... while loop - do { } while (i<10)

String Object

- Count characters in a string (length propery)
- Test if a string contains a specified character (indexOf() method)
  Eg: email= anselva@yahoo.com, email.indexOf('@') = 7, email.indexOf('b') = -1
- Return a specified part of a string (substring() method)
  Eg: email.substring(0,6) = anselva
- Uppercase or lowercase characters? (toLowerCase() and toUpperCase() methods)

Array Object

- Make an array that stores names
- Find out how many elements an array contains

Eg: myArray = new Array(10); // myArray[0], myArray[1], myArray[2],.., myArray[9]
var dayArray=new Array ("Sun","Mon","Tues","Wednes","Thurs","Fri","Satur")
var monthArray=new Array ("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")

Date Object

Today's date, current local time, set date, display current weekday. The Date object is used to work with dates and times.

var d1 =new Date()
document.write (d1.getDate())
getDate(), getDay() : 0-sun, 6-satur, getHours(), getMinutes(), getMonth(), getSeconds(), getYear()

setDate() - Sets the day of the month.
setHours() - Sets the hour for the current time.
setMinutes() - Sets the minutes for the current time.
setMonth() - Sets the month with an integer from zero (January) to 11 (December).
setSeconds() - Sets the seconds for the current time.
setTime Sets the value of a date object.

```
<script language="javascript">
   <!-- Hide this code from non JavaScript browsers
   currentTime = new Date();
   if (currentTime.getHours() < 12)
     document.write("Good morning");
   else if (currentTime.getHours() < 17)
     document.write("Good afternoon");
   else
     document.write("Good evening");
   // End of JavaScript code -->
</script>
```

Window

- Display an alert box
  Eg: alert("Your email is an invalid!")

- Display an alert box with line breaks
  Eg: alert("Your email is an invalid!\n Please re enter the email address!")
  \n – new line, \t – tab, \r- enter key, \" – ", \' – '

- Disable right-click IE only
  Eg: **Disable right click mouse**

```
function rightClickMouseDisable (e) {
if (e.button == 2 || e.button == 3) {
        alert("Sorry, you do not have permission to right click!");
        return false;
}
<body onMouseDown="rightClickMouseDisable(event)">
```

- Display a confirm box

```
if (confirm ("Do you want to delete this record!"))
        .......
else
        .......
```

- Display a prompt box
  var age=prompt("Please enter your age:")

- Open a new window

```
function openwindow()
{
window.open("http://www.uoctamil.com","my_new_window","width=400 height=400")
}
```

- Open a new window when clicking on a button and control its appearance
  onClick=window.open("http://www.uoctamil.com","_blank","toolbar=yes, location=yes, directories=no, status=no, menubar=yes, scrollbars=yes, resizable=no, copyhistory=yes, width=400, height=400, top=200, left=100")

- Close a window

```
<input type="button" value="Close" onClick="self.close()">
<input type="button" value="Close" onClick="window.close()">
```

- Refresh page

  Eg: `<input type="button" value="Refresh" onClick="location.reload()">`

- Change text in the status bar

To display something on the status bar when the mouse is moved over this link:

```
<A HREF="mylink.htm" onMouseOver="window.status='Click here to know more about
me'; return true" onMouseOut="window.status=''; ">Click here</A>
```

- Print page

```
<input type="submit" value="Print" onclick="window.print()">
```

Window Resize

```
onresize="self.resizeTo(490,200)"
<body onresize="self.resizeTo(490,185)">
```

Back to last web page

```
<a href="javascript:history.back()"><img src="back.gif"></a>
<a href="javascript:history.go(-1)"><img src="back.gif"></a>
<input type=”button” value=”Back” onClick=” history.back()”>
```

Focus text box

```
onload="document.login_form.login.focus()"
```

onSubmit

```
<FORM action="something.asp" method=”post” onSubmit="return checkscript()">
```

Full Screen

```
<body onload="window.moveTo(0,0),window.resizeTo(screen.availWidth,screen.availHeight)">
```

Control the special characters

```
onKeypress="if ( (event.keyCode > 32 && event.keyCode < 48 ) || (event.keyCode > 57 &&
event.keyCode < 65) || (event.keyCode > 90 && event.keyCode < 97) || (event.keyCode >
122 && event.keyCode < 255)) event.returnValue = false;"
```

How to capture form values in javascript

```
document.formname.fieldname.value
eg: document.frm1.name.value
```

Forms

- Validate an e-mail address
- Validate a value
- Validate a length of an input

To displaying "Last Updated" date on html file

```
<script language="javascript">
<!--
    document.write("Last updated :" + document.lastModified);
// -->
</script>
```

```
function todayDate()
{
   var tdate = new Date()
   var d1=tdate.getDate()
   var sup="th"
   if (d1==1 || d1==21 || d1==31)
```

```
        sup="st"
   else if (d1==2 || d1==22)
        sup="nd"
   else if (d1==3 || d1==23)
        sup="rd"
    var cdate=dayArray[tdate.getDay()]+"day, "+d1+ sup +" "+ monthArray[tdate.getMonth()]
+" "+ tdate.getYear()+""
   document.write(cdate)
}
function movein(which)
{
  which.style.background='#666666'
  which.style.color='#000000'

}
function moveout(which)
{
  which.style.background='#333333'
  which.style.color='#FBFCE2'

}
```

Handling Events

JavaScript applications in Navigator are largely event-driven. *Events* are actions that usually occur as a result of something the user does. For example, clicking a button is an event, as is changing a text field or moving the mouse over a link. For your script to react to an event, you define *event handlers*, such as onChange and onClick.

The following table summarizes the JavaScript events (most important events).

Event	**Applies to**	**Occurs when**	**Event handler**
Change	text fields, select lists	User changes value of element	onChange
Click	buttons, radio, checkboxes, submit, reset buttons, links	User clicks form element or link	onClick
KeyPress	documents, images, links, text areas	User presses or holds down a key	onKeyPress
Load	document body	User loads the page in the Navigator	onLoad
MouseOut	areas, links	User moves cursor out of a client-side image map or link	onMouseOut
MouseOver	links	User moves cursor over a link	onMouseOver
Resize	windows	User or script resizes a window	onResize
Submit	forms	User submits a form	onSubmit
Unload	document body	User exits the page	onUnload