

APPENDIX
C
JAVASCRIPT

BASIC JAVASCRIPT PROGRAMMING TECHNIQUES AND SAMPLE CODE

This appendix contains a list of commonly used JavaScript techniques. Sample code is provided to illustrate how these techniques could be used within a Web application. Because JavaScript is commonly used to manipulate the objects, properties, and methods in the Document Object Model, this appendix also contains a list of the objects, properties, and methods commonly used. For more in-depth information on the Document Object Model, please visit the World Wide Web Consortium at <http://www.w3c.org>. This appendix is not meant to be an exhaustive reference on JavaScript, nor does it contain the complete reference and documentation for JavaScript. For the complete documentation on JavaScript, please visit Netscape at <http://developer.netscape.com/tech/javascript/index.html>. JavaScript is case sensitive. For information on ECMAScript, please visit European Computer Manufacturers Association (ECMA) at <http://www.ecma.ch/>. Information on JScript can be found at the Microsoft Windows Script Technologies Web site at <http://msdn.microsoft.com/scripting/>.

Comments and the JavaScript Basics

To add a one-line comment, use two forward slashes. You can add comments inline.	<code>// This is a one-line comment</code>
To add multiline comments, you need two sets of forward slashes and asterisks.	<code>/* This comment can continue over several lines */</code>
The semicolon (;) is used to indicate the end of a JavaScript statement. In general, JavaScript is case sensitive.	<code>document.write("Hello");</code>
To concatenate variables and strings, use the ampersand (&) as the concatenation operator.	<code>document.write("Cost = " + intCost);</code>

JavaScript Reserved Words

abstract	else	instanceof	static
boolean	extends	int	super
break	false	interface	switch
byte	final	long	synchronized
case	finally	native	this
catch	float	new	throw
char	for	null	transient
class	function	package	true
const	goto	private	try
continue	if	protected	var
default	implements	public	void
do	import	return	while
double	in	short	with

Using Variables

To declare a variable	<code>var strProduct, intQuantity;</code>
To assign a variable to a value	<code>intQuantity = 5; myDataType = "Shirts"; myDataType = true;</code>
To assign a variable to a form field	<code>document.frm.txtname.value = name;</code>
To retrieve the value of a variable and use it in an expression	<code>document.write(intCost);</code>

Arithmetic and Unary Operators

Category	Operator	Description	Example
Arithmetic	+	Addition	<code>x + y</code>
Arithmetic	-	Subtraction	<code>x - y</code>
Arithmetic	*	Multiplication	<code>x * y</code>
Arithmetic	/	Division	<code>x / y</code>
Unary	++	Increment	<code>i = ++ i</code>
Unary	--	Decrement	<code>i = -- i</code>
Unary	-	Negation	<code>-i</code>

The Date

To create a new date object with the current date	<code>var d = new Date();</code>
To create a new date object with a specific date	<code>var d = new Date("January 14, 2002");</code> <code>var d = new Date("January 14, 2002 04:35:00");</code> <code>var d = new Date("02, 1, 14");</code> <code>var d = new Date("02, 1, 14, 00, 35, 00");</code>
To get the date	<code>getDate(d)</code>
To get the day of the week	<code>getDay(d)</code>
To get the month	<code>getMonth(d)</code>
To get all four digits of the year	<code>getFullYear(d)</code>
To get the date in Universal Time Code (UTC)	<code>getUTCDate(d)</code>
Converts the date object to a string	<code>toString(d)</code>

Arrays

To declare and populate a one-dimensional array	<code>var Products = new Array(1);</code> <code>Products[0] = "Pencils";</code> <code>Products[1] = "Pens";</code>
To write an array element value to a Web page	<code>document.write(Products[0]);</code>

Comparison Operators

Condition being tested	Operator	Sample
Equal to	<code>==</code>	<code>var1 == var2</code>
Not equal	<code>!=</code>	<code>var1 != var2</code>
Less than	<code><</code>	<code>var1 < var2</code>
Greater than	<code>></code>	<code>var1 < var2</code>
Less than or equal to	<code><=</code>	<code>var1 <= var2</code>
Greater than or equal to	<code>>=</code>	<code>var1 >= var2</code>

Using Built-in Functions

To retrieve the data type of a variable, use the <code>typeof</code> function.	<pre>var dt; dt = "Total Cost"; document.write("Data Type is: " + typeof(dt));</pre>
To detect if the value of a variable is not a number, use the <code>isNaN</code> function.	<pre>strNum = isNaN(strQuantity); document.write(strNum);</pre>
To change a number string into a number, you can use several built-in functions.	<pre>strQ = parseInt(strQuantity); strQ = parseFloat(strQuantity); strQ = eval(strQuantity);</pre>
To call a built-in mathematical function in JavaScript	<pre>ceil(5.05);</pre>

Escape Sequence Characters in JavaScript

Special Character	Escape Character	Escape Sequence	Sample
Carriage return	r	\r	<code>window.alert "Welcome " & strName & "\r" & "to our Website"</code>
Backslash	\	\\	<code>"document.write("Your Webpage is located at: " & "C:\\InetPub\\wwwroot\\home.htm"</code>
New line	n	\n	<code>window.alert "Welcome " & strName & "\n" & "to our Website"</code>
Tab space	t	\t	<code>"document.write("\n" & "Your total cost is \$4.50. "</code>
Backspace	b	\b	<code>"document.write("Click to continue.\b?"</code>
Single quotation mark	'	\'	<code>window.alert "Welcome \' " & strName & "\'"</code>
Double quotation mark	"	\"	<code>window.alert "Welcome \" " & strName & "\""</code>
Form feed	f	\f	<code>"document.write("Click to continue.\f"</code>

Using Functions

To create a function you must use the function keyword. The curly brackets enclose the code.	<pre>function calcTotal(){ var cost; cost = 2; document.write(cost); }</pre>
To call a function	<pre>calcTotal()</pre>
To return a result from a function	<pre>function calcTotal(){ var cost; cost = 2; return cost; }</pre>

C

Control Structures

To create an If Then Else statement	<pre>if (a=1) { window.alert("a"); } else{ window.alert("not a"); }</pre>
To create a Switch Case statement	<pre>switch (aNumber){ case 1: window.alert("1"); break; case 2: window.alert("2"); break; default: window.alert("other"); break; }</pre>
To create a While Do statement	<pre>var z, i; z= 5; i = 0; while (i != z){ window.alert(i); i++; }</pre>

Control Structures (continued)

To create a Do While statement	<pre>var z = 5; var i = 0; do{ window.alert(i); i++; } while(i != z);</pre>
To create a For Next Loop statement	<pre>var z; z = 5; var i; i = 0; while (i != z){ window.alert(i); i++; }</pre>
To create a For In Loop statement	<pre>for(var element in document){ document.write(element + " = "); document.writeln(document[element]); document.writeln("
"); }</pre>

Logical Operators

Operator	Sample	Description
&&	if ((a == 1) && (b == 2))	If both expressions are true, the result is true. If either expression is false, the result is false.
	if ((a == 1) (b == 2))	If either expression is true, or both are true, the result is true. If both are false, the result is false.
!	if (! (a == 1))	Used to negate a single expression. If the expression is true, the result is false. If the expression is false, the result is true.

OBJECTS, PROPERTIES, AND METHODS OF THE DOCUMENT OBJECT MODEL

The Window Object

Description	Sample
To alter the status message bar	<code>window.status = "Welcome"</code>
To create an alert dialog box	<code>window.alert("Warning!")</code>
To create a confirm dialog box	<code>window.confirm("Are you ready?")</code>
To create a prompt dialog box	<code>window.prompt("Your Name:", "Enter your name here")</code>
To open a new window and alter the appearance	<code>window.open("MyPage.htm", "windowName", "height=200, width=400, status=no, toolbar=no, menubar=no, location=no")</code>
To close a window	<code>window.close()</code>

The Navigator Object

To display the browser name	<code>navigator.appName</code>
To display the browser version	<code>navigator.appName</code>
To display the user agent	<code>navigator.userAgent</code>
To display language (VBScript only)	<code>navigator.userLanguage</code>
To display system language (VBScript only)	<code>navigator.systemLanguage</code>
To go to the default home page (JavaScript only)	<code>navigator.home()</code>

The Location Object

To display the complete URL of the current document	<code>location.href</code>
To display pieces of the URL, such as the hostname	<code>location.hostname</code>
To redirect the window to a new document	<code>location.href = ("home.htm")</code>

The History Object

To display the number of entries in the history list	<code>window.history.length</code>
To move to the previous window in the history list	<code>window.history.back()</code>
To move to the next window in the history list	<code>window.history.next()</code>
To move to a specific entry in the history list	<code>window.history.go(2)</code>

The Document Object

To write to the document in the current window	<code>document.write("Welcome")</code>
To write to the page in a specific window	<code>winName.document.write("Name")</code>
To write the contents to the window and append a carriage return after the line of text	<code>document.writeln("Welcome")</code>
To write the results of an expression or function to the window	<code>document.write (select1.getAttribute("name"))</code>
To display the document properties, such as the document title, referring page, and last modified date	<code>document.title document.referrer document.lastmodified</code>
To change the document properties, such as the background and foreground colors	<code>document.bgcolor = "green" document.bgcolor = "white"</code>
To handle events such as when the document loads or closes	<code><body onLoad = "alert('Welcome')" onUnload = "alert('Goodbye!')"></code>
To close a page	<code>document.close()</code>

The Form Object

To display the form properties, such as the name of the form	<code>window.document.loginform.name</code>
To submit the form without the user's involvement	<code>submit()</code>
To handle events such as when the form is submitted	<code><form name = "form" method = "get" action = "go.asp" onsubmit = "alert('Thanks!')"></code>
To display an object property such as the name	<code>window.document.frm.txtName.name</code>
To display an object property such as the value	<code>window.frm.prod.value</code>
To change the contents of a property such as the value property <code>readOnly</code>	<code>window.frm.prod.value = "2" window.txtName.readOnly = "True"</code>

The Form Object (continued)

To change the focus to the object	<code>txtName.focus()</code>
To remove the focus	<code>txtName.blur()</code>
To change the focus to the object and highlight the contents of the object	<code>txtName.select()</code>
To handle events such as when the object is clicked (onClick)	<code><input type = button value = "Change" onClick = "ChangeImage()"></code>

C

Event Handlers

To handle events such as when the object is clicked (onClick)	<code><input type = button value = "Change" onClick = "ChangeImage()"></code>
To change the contents of a property of the images collection, such as the SRC property	<code><input type = button value = "Change" onClick = "window.document.jumper1.src = 'jumper1.jpg';"></code>
To change the image source property when the user places the mouse over a hyperlink	<code> Jumper </code>
To change the status message property when the user places the mouse over an image. The hyperlink goes nowhere.	<code> Jumper </code>
To change the status message property when the user places the mouse over a hyperlink	<code> Shirts </code>

Events

Event Handler	Description
onfocus	Action performed when an object has the focus
onblur	Action performed when an object loses focus
onselect	Action performed when the user highlights text
onchange	Action performed when the user changes a value and moves off the object
onsubmit	Action performed when the user clicks the Submit button
onclick	Action performed when the user clicks a button
onmouseover	Action performed when the user moves the mouse over an object
onmouseout	Action performed when the mouse moves off an object
onload	Action performed when a document object is loaded
onunload	Action performed when a document object is unloaded
onabort	Action performed when the user clicks the Stop button in the browser
onerror	Action performed when an error occurs in the script
onreset	Action performed when the user clicks the Reset button
ondragdrop	Action performed when the user drags and drops an object into the window object
onkeydown	Action performed when the user presses a key
onkeypress	Used with onkeydown; action performed while an object holds a key
onkeyup	Action performed when the user releases a key
onmousedown	Action performed when the user presses the mouse button
onmousemove	Action performed when the user moves the mouse
onmouseup	Action performed when the user releases the mouse button
onresize	Action performed when a window is resized