

**Chapter
2**

CONSOLE I/O

*Get on the
Fast Track!*



TM

**SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.**

Objectives

You will learn:

- C MsgBox function.
- C InputBox function.
- C Displaying information from client scripts.
- C Displaying information on the page.
- C Displaying information in an HTML control.
- C Sending formatting tags to the browser.
- C Displaying special characters.
- C Using the status bar for displaying messages.

1 MsgBox Function

The MsgBox function displays a message in a dialog box, waits for the user to click a button, and returns a value indicating which button the user clicked.

Syntax:

```
MsgBox(prompt[, buttons][, title][, helpfile, context])
```

Argument	Description
prompt	String expression displayed as the message in the dialog box. The maximum length of prompt is approximately 1024 characters, depending on the width of the characters used. If prompt consists of more than one line, the lines can be separated using a carriage return character (Chr(13)), a linefeed character (Chr(10)), or carriage return–linefeed character combination (Chr(13) & Chr(10)) between each line.
buttons	Numeric expression that is the sum of values specifying the number and type of buttons to display, the icon style to use, the identity of the default button, and the modality of the message box.
title String	Expression displayed in the title bar of the dialog box. If title is omitted, the application name is placed in the title bar.
helpfile	String expression that identifies the Help file used to provide context-sensitive Help for the dialog box. If helpfile is provided, context must also be provided. It is not available on 16-bit platforms.
context	Numeric expression that identifies the Help context number assigned by the Help author to the appropriate Help topic. If context is provided, helpfile must also be provided. Not available on 16-bit platforms.





1.1 How Many/What Buttons?

The buttons argument settings are:

Constant	Value	Description
vbOKOnly	0	Display OK button only.
vbOKCancel	1	Display OK and Cancel buttons.
vbAbortRetryIgnore	2	Display Abort, Retry, and Ignore buttons.
vbYesNoCancel	3	Display Yes, No, and Cancel buttons.
vbYesNo	4	Display Yes and No buttons.
vbRetryCancel	5	Display Retry and Cancel buttons.

1.2 Which Icon?

An icon can be displayed by adding the appropriate value from the following table to buttons argument:

Constant	Value	Icon
vbCritical	16	 Display Critical Message icon.
vbQuestion	32	 Display Warning Query icon.
vbExclamation	48	 Display Warning Message icon.
vbInformation	64	 Display Information Message icon.

1.3 Default Button

One of the displayed buttons can be set as the default button by adding the appropriate value to the buttons argument.

Constant	Value	Default Button
vbDefaultButton1	0	First button is default.
vbDefaultButton2	256	Second button is default.
vbDefaultButton3	512	Third button is default.
vbDefaultButton4	768	Fourth button is default.

1.4 Modal Type

It is possible to specify whether the message box will be application modal or system modal by adding the appropriate value to the buttons argument.

Constant	Value	Description
vbApplicationModal	0	Application modal; the user must respond to the message box before continuing work in the current application.
vbSystemModal	4096	System modal; all applications are suspended until the user responds to the message box.

When adding numbers to create a final value for the argument buttons, use only one number from each group.

1.5 Return Values

The MsgBox function has the following return values:

Constant	Value	Button
vbOK	1	OK
vbCancel	2	Cancel
vbAbort	3	Abort
vbRetry	4	Retry
vbIgnore	5	Ignore
vbYes	6	Yes
vbNo	7	No

If the dialog box displays a Cancel button, pressing the ESC key has the same effect as clicking Cancel.

If the dialog box contains a Help button, context-sensitive Help is provided for the dialog box.

However, no value is returned until one of the other buttons is clicked.

Example:

This code uses the MsgBox function to display a message box and return a value describing which button was clicked:

```
1 Dim MyVar
2 MyVar = MsgBox ("Hello World!", 65, "MsgBox Example")
```



2 InputBox Function

InputBox function displays a prompt in a dialog box, waits for the user to input text or click a button, and returns the contents of the text box.

Syntax:

```
InputBox(prompt[, title][, default][, xpos][, ypos][, helpfile, context])
```

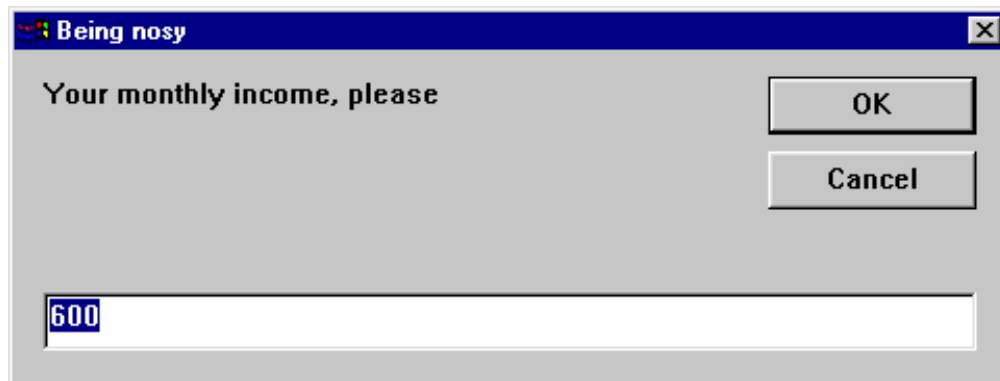
Argument	Description
prompt	String expression displayed as the message in the dialog box. The maximum length of prompt is approximately 1024 characters, depending on the width of the characters used. If prompt consists of more than one line, the lines can be separated using a carriage return character (Chr(13)), a linefeed character (Chr(10)), or carriage return–linefeed character combination (Chr(13) & Chr(10)) between each line.
title	String expression displayed in the title bar of the dialog box. If title is omitted, the application name is placed in the title bar.
default	String expression displayed in the text box as the default response if no other input is provided. If the default is omitted, the text box is displayed empty.
xpos	Numeric expression that specifies, in twips, the horizontal distance of the left edge of the dialog box from the left edge of the screen. If xpos is omitted, the dialog box is horizontally centered.
ypos	Numeric expression that specifies, in twips, the vertical distance of the upper edge of the dialog box from the top of the screen If ypos is omitted, the dialog box is vertically positioned approximately one-third of the way down the screen.
helpfile	String expression that identifies the Help file to use to provide context-sensitive Help for the dialog box. If helpfile is provided, context must also be provided.
context	Numeric expression that identifies the Help context number assigned by the Help author to the appropriate Help topic.

- C When both helpfile and context are supplied, a Help button is automatically added to the dialog box.
- C If the user clicks OK or presses ENTER, the InputBox function returns whatever is in the text box.
- C If the user clicks Cancel, the function returns a zero-length string ("").

Example:

This code uses the InputBox function to display an input box and assign the string to the variable nMonthly:

```
1 Dim nMonthly
2 nMonthly = InputBox("Your monthly income, please", "Being nosy",600)
3 MsgBox ("Your annual income is " & nMonthly*12,64,"Financial status" )
```



3 Displaying Information from Client Scripts

Since client scripts run on the browser, they provide flexibility in how information should be displayed to the user. Message boxes can be displayed the same way stand-alone applications on a computer do.

3.1 document.Write Method

Information from client scripts can be displayed directly on the page, intermingled with the HTML text.

To display information in the page from client scripts:

Call the document.write method; this puts text at the location in the page where the script is executing.

Example:

```
1     <SCRIPT LANGUAGE="VBScript">
2         document.Write "The current time is " & time
3     </SCRIPT>
```

If a page contains an HTML text box or text area control, the contents of the box displaying information can be changed.

3.2 document.WriteLine Method

The WriteLine method sends strings, followed by a newline character, to the Immediate window of the Microsoft Script Debugger at run time. If the script is not being debugged, the WriteLine method has no effect.

```
document.WriteLine([str1 [, str2 [, ... [, strN]]]])
```

Argument	Description
str1, str2, ... , strN	Strings to send to the script debugger.

The WriteLine method is almost identical to the Write method. The only difference is that the Write method does not send a newline character after sending the strings.

Example:

This code uses the WriteLine method to display the value on an HTML document:

```
1      Dim counter
2      counter = 42
3      document.WriteLine "The value of counter is " & counter
```

3.3 Displaying Information in an HTML Control

To display information inside an HTML control, set the value property of the text box or text area control.

Example:

```
1      <SCRIPT LANGUAGE="VBScript">
2      Function btnShowTime_onclick()
3          document.Form1.txtTime.value = time
4      End Function
5      </SCRIPT>
```

3.4 innerText Property

If an application will be running in browsers that support Dynamic HTML, it is possible to directly set the text of any tag that has a name or ID.

To set the text of a tag using DHTML, set the tag's innertext property. The tag must have an ID or a name that you can reference in the script.

Example:

This page illustrates a tag and how to set it.

```
1      <HEAD>
2      <SCRIPT LANGUAGE="VBScript">
3      Function btnChangeText_onclick()
4          para1.innerText = "The new time is: " & time
5      End Function
6      </SCRIPT>
7
8      </HEAD>
9      <BODY>
10     <P ID=para1>This text will be replaced.</P>
11     <P><INPUT TYPE="button" NAME="btnChangeText" VALUE="Change text"></P>
12 </BODY>
```

3.5 Formatting the Displayed Text

To format the text being displayed, HTML tags can be included:

Example:

```
1     <SCRIPT LANGUAGE="VBScript">
2         document.write "<P>The current time is <B>" & time & "</B></P>"
3         document.write "<P>The current date is <B>" & date & "</B></P>"
4     </SCRIPT>
```

3.6 Displaying Reserved Characters

If the information to be displayed includes characters that are reserved in HTML — such as < and > — they can not be directly be included in the string to display.

To display reserved characters, use the HTML syntax for ASCII characters, such as < or < for the opening angle bracket (<):

```
1     <SCRIPT LANGUAGE="VBSCRIPT">
2         document.write "&lt;Click here&gt;"
3     </SCRIPT>
```

4 Status Bar

Messages can be set or retrieved from the status bar at the bottom of the window.

Syntax:

```
window.status[ = status]
```

The value of this read-write property can be any valid string.

Since the current window object is implied, it is not necessary to prefix the property with the window keyword. However, many people prefer to use the keyword to ensure that their scripts are clear and readable.

Example:

```
status = "Hello, visitor"
```

is the same as saying:

```
window.status = "Hello, visitor"
```

and both will work when referring to the current window object.

