The MsgBox function provides a way to display messages to the user and have them respond. The message box displays a message to the user and waits for a response. The message box then returns a value to the program. This value depends on which button is picked by the user.

A sample message box is shown in Figure 1. In this case, the message box has two buttons, a text message, an icon, and a title.

![Figure 1: Sample Message Box](image)

The syntax for the MsgBox function is shown below. The values within the brackets, [ ], are optional.

```vba
MsgBox message [, [buttons][, title][, helpfile, context ]]
```

- **message**: The string expression displayed as the message in the dialog box. The maximum length is 1024 characters.
- **buttons**: A numeric expression that is the sum of values that describe the display and behavior of the message box.
- **title**: String expression that will appear in the dialog box title bar. If omitted, the project name appears.
- **helpfile**: String naming the help file associated with this message.
- **context**: Required if helpfile is used. Provides the context for context-sensitive help.

The statement used to create the dialog box in Figure 1 is:

```vba
MsgBox “The value is too high.”, 17, “Error”
```

The **buttons** option is a number that tells VB what buttons and icon to display as well as which button is the default button. The value is the sum of three numbers: the button component + the icon component + the default button component. Only one number can be used for each component. See the table on the next page for the component values.

**Example**: Compute the value to display both an OK and Cancel button along with the Stop icon and having the Cancel button (second button) be the default button.

The argument is computed as: \(1 + 16 + 256 = 273\). Alternately, you can add the constants together which would be: vbOKCancel + vbCritical + vbDefaultButton2. The second method is easier to remember, but longer to type.
The response from the button pick can be assigned to a variable. For instance, the following command starts the message box and saves the value of the button picked (1 for OK or 2 for Cancel) to the integer variable 'response.'

```vba
response = MsgBox("The value is too high.", 17, "Error")
```

This code can be used in conjunction with an If-block to react to the user’s response.
Example: Write a procedure that asks the user if they want to print the worksheet when they pick the menu option to close the workbook.

```vba
Sub Workbook_BeforeClose ()
    Response = MsgBox ("Do you want to print this sheet before you quit?", 36, "Print Option")
    If Response = 6 Then
        Selection.PrintOut Copies:=1, Collate:=True
    End If
End Sub
```

If the user picks the YES button (6) then the worksheet prints before the workbook closes. Otherwise, the workbook closes without printing.