**Module 8: User Interaction**

* **InputBox Function**

The InputBox function is used to collect information (input) from the user.  
**Syntax:**

InputBox(Prompt, [Title], [Default], [XPos], [YPos], [HelpFile], [Context])

* **Prompt** → Message displayed to the user.
* **Title** → Text in the title bar (optional).
* **Default** → Default value in the input box (optional).
* **XPos, YPos** → Position of the input box on the screen (optional).
* **HelpFile, Context** → Help file references (rarely used).

**Example:**

Sub GetUserName()

Dim userName As String

userName = InputBox("Enter your name:", "User Information", "Guest")

MsgBox "Welcome, " & userName & "!", vbInformation, "Greeting"

End Sub

👉 This code asks the user for their name and then greets them.

* **MsgBox Function**

The MsgBox function is used to display information and optionally collect responses.

Syntax:

MsgBox(Prompt, [Buttons], [Title])

* Prompt → Message displayed.
* Buttons → Type of buttons/icons (optional).
* Title → Text in the title bar (optional).

Button Options:

* vbOKOnly → OK button (default).
* vbOKCancel → OK and Cancel buttons.
* vbYesNo → Yes and No buttons.
* vbYesNoCancel → Yes, No, and Cancel buttons.
* vbInformation, vbExclamation, vbCritical, vbQuestion → Icons.

Example 1: Simple MsgBox

Sub ShowMessage()

MsgBox "Process completed successfully!", vbInformation, "Status"

End Sub

Example 2: MsgBox with User Response

Sub ConfirmAction()

Dim response As VbMsgBoxResult

response = MsgBox("Do you want to continue?", vbYesNo + vbQuestion, "Confirmation")

If response = vbYes Then

MsgBox "You chose YES", vbInformation

Else

MsgBox "You chose NO", vbExclamation

End If

End Sub

* **Collecting and Displaying Information**

You can combine **InputBox** and **MsgBox** to make interactive programs.

**Example:**

Sub CollectAndDisplay()

Dim age As String

age = InputBox("Enter your age:", "Age Input")

If age <> "" Then

MsgBox "You are " & age & " years old.", vbInformation, "User Info"

Else

MsgBox "You did not enter your age.", vbExclamation, "Missing Info"

End If

End Sub

✅ **Summary:**

* **InputBox** → Collects user input.
* **MsgBox** → Displays messages and gets responses.
* Together, they make VBA programs interactive and user-friendly.

**Some real life examples**

**Dynamic Discount Calculator using inputbox**

Sub Real\_life\_Example\_Button1\_Click()

Dim price As Double, discount As Double, finalPrice As Double

price = InputBox("Enter product price:", "Discount Calculator")

discount = InputBox("Enter discount percentage:", "Discount Calculator", 10) 'Default = 10%

finalPrice = price - (price \* discount / 100)

MsgBox "Final price after " & discount & "% discount = " & finalPrice, vbInformation

End Sub

**Mini Project: Sales Report Filter with InputBox**

**Scenario**

**You have a Sales Data sheet in Excel with these columns:**

* **A → Date**
* **B → Product**
* **C → Sales Amount**

**You want a VBA program that:**

1. **Asks the user for a Start Date and End Date using InputBox.**
2. **Filters the sales data within that range.**
3. **Shows the Total Sales for that period in a MsgBox.**

**VBA Code Example**

Sub SalesReportFilter()

Dim startDate As Date, endDate As Date

Dim ws As Worksheet

Dim lastRow As Long

Dim i As Long

Dim totalSales As Double

'Set worksheet

Set ws = ThisWorkbook.Sheets("SalesData")

'Find last row

lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row

'Get user input

startDate = InputBox("Enter Start Date (dd-mm-yyyy):", "Sales Report Filter")

endDate = InputBox("Enter End Date (dd-mm-yyyy):", "Sales Report Filter")

'Loop through data

totalSales = 0

For i = 2 To lastRow

If ws.Cells(i, 1).Value >= startDate And ws.Cells(i, 1).Value <= endDate Then

totalSales = totalSales + ws.Cells(i, 3).Value

End If

Next i

'Show result

MsgBox "📊 Total Sales from " & startDate & " to " & endDate & " = ₹" & totalSales, vbInformation, "Sales Report"

End Sub