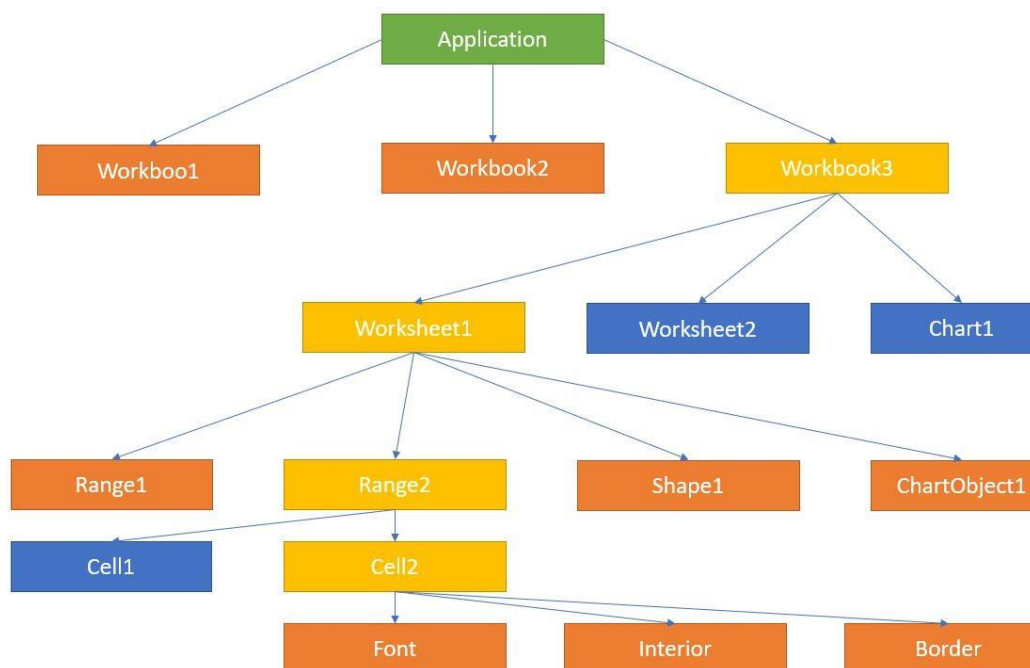


09.1 Excel Objects

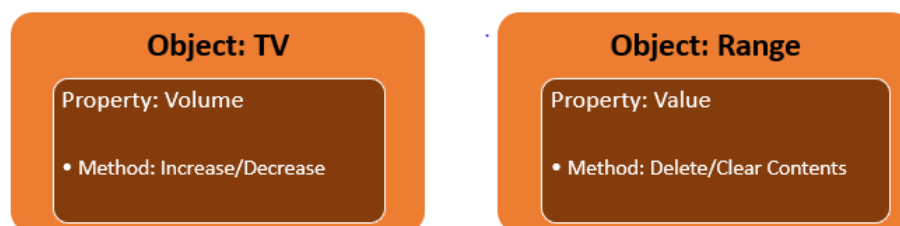
Excel objects collectively form the Excel Object Model. This model provides a way to interact with various elements within an Excel spreadsheet using code. Each object has its own properties (characteristics) and methods (actions you can perform on it).

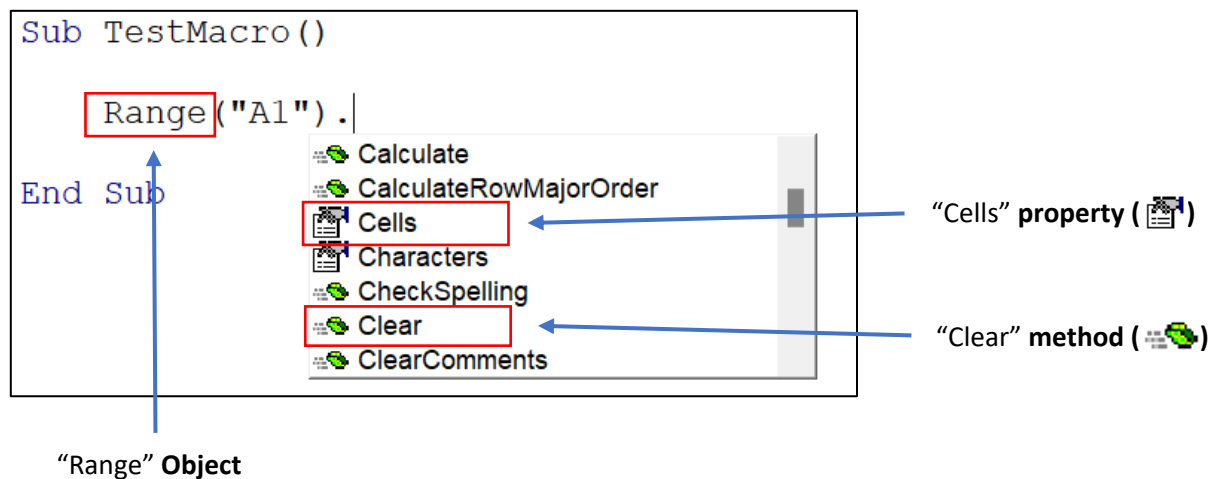
Examples of Excel Objects:

- **Workbook:** Represents the entire Excel file, containing one or more worksheets.
- **Worksheet:** An individual sheet within the workbook where you organize your data in rows and columns.
- **Range:** A selection of cells, which can be a single cell, a row, a column, a block of cells, or even noncontiguous selections. It's the fundamental building block for working with data in your worksheet.
- **Chart:** A visual representation of your data using elements like bars, lines, or pie charts.
- **Cell:** The individual intersection of a row and column where you enter or display data.



By understanding and using these objects and their **properties** and **methods**, you can automate tasks in VBA.





Activity 09.3

Open a new excel workbook and save it as a macro enabled excel file, giving the name "Activity 09-3". Perform the following in the new workbook.

01. Create a new macro named "MyName".

(Path – **Developer** Tab → **Macros** → Provide the macro name → **Create**)

02. Write a VB code to display your name in "B2" cell.

03. Run the macro by selecting the relevant code segment (in VB editor).

01. Create a new macro named "CopyValues".

02. Write a VB code to copy your name from "B2" cell to "D2" cell.

03. Run the "CopyValues" macro by assigning to a button (Rename the button as "Copy Name")

01. Select "D4" cell.

02. Create a new macro named "Active_Cell_Value".

03. Write a VB code to assign number 50 to active cell.

04. Run the macro by selecting the relevant code segment (in VB editor).

01. Add a new worksheet.

02. Create a new macro named "CopyBetweenSheets".

03. Write a VB code to copy your name to "B3" cell in "Sheet2" worksheet.

04. Run the "CopyBetweenSheets" macro by assigning to a button (Rename the button as "Copy to Sheet")

01. Insert a new button to "Sheet1" worksheet. (You can rename the button later)

02. Create a new macro at the time of inserting the button, named "ClearData".

03. Write a VB code to clear "A1:E10" cell range.

04. Rename the button as "Clear" and run the VB code by clicking the button.



Activity 09.4

Open a new excel workbook and try out the following.

| Code | Description |
|---|--|
| ActiveCell.Interior.Color = RGB(0, 100, 255) | Add a fill color to the Active Cell |
| ActiveCell.Font.Color= RGB(255, 255, 255) | Change the font color in the Active Cell |
| Application.Quit | Close excel application |
| Worksheets("Sheet1").Cells(2, 3).Value = "2nd Row 3rd Column" | Alternative method of selecting a cell and inserting a value |
| Rows(1).EntireRow.Clear | Clear contents of the first row |
| Rows(1).EntireRow.Delete | Delete the first row |
| Columns(1).EntireColumn.Clear | Clear contents of the first column |
| Columns(1).EntireColumn.Delete | Delete the first column |

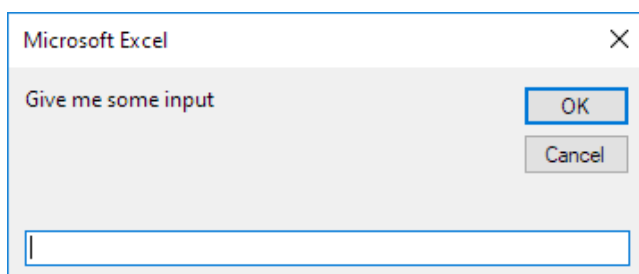
09.2 Message Box and Input Box

09.2.1 Message Box

A message box is a pop-up dialog created using the “**MsgBox**” function. It's a powerful tool for interacting with users of your macro. Message Box displays messages to inform users about the macro's progress or results and get user input through buttons like Yes/No, OK/Cancel, or Retry/Abort.



09.2.2 Input Box



An input box in Excel VBA is a dialog box you can use in your macros to prompt the user for input. It allows users to enter information during the macro's execution. You use the “**InputBox**” function within your VBA code.



Activity 09.4 (Part A)

Open a new excel workbook and save it as a macro enabled excel file, giving the name “Activity 09-4”. Perform the following in the new workbook.

01. Enter your hometown in “B2” cell.
02. Create a new macro named “DisplayHometown”.
03. Write a VB code to display your hometown in a message box as follows.
Your Hometown is : [Your Hometown]
04. Run the “DisplayHometown” macro by assigning to a button (Rename the button as “Display Hometown”).

01. Create a new macro named “EnterID”.
02. Write a VB code to enter your NIC number in “D2” cell using an input box.
(You need to display “Enter your NIC” message in the Input box)
03. Run the “EnterID” macro by assigning it to a button (Rename the button as “Enter NIC”).

09.3 Move between cells

Syntax : **Offset(no of rows , no of columns).Select**



Activity 09.4 (Part B)

Use the “Activity 09-4” workbook created above and perform the following.

01. Create 4 Buttons and name them as “Left”, “Right”, “Up” and “Down”
02. Create the macros below and write the VB code to perform the given action.
 - a. “MoveLeft” macro - Move active cell to the left side by one cell.
 - b. “MoveRight” macro - Move active cell to the right side by one cell.
 - c. “MoveUp” macro - Move active cell up by one cell.
 - d. “MoveDown” macro - Move active cell down by one cell.
03. Assign the macros to the relevant buttons.
04. Change the ‘Offset’ values and check the behavior of active cell.

09.4 Variables

Variables are 'places' that you create to store "data" when you run your program. Variables have a name (you think of) and a type (such as Integer, String, Decimal etc.).

- In VBA first you need to declare the variable. This is done (mostly) with the word "Dim"

E.g.- **Dim** intNumber **As Integer**.

- After declaring, you can initialize the variable. That means assigning a value to the variable.

E.g.- intNumber = 12

Rules for naming variables

- A variable name must start with a letter.
- The name of a variable cannot contain spaces.
- You cannot use special characters as #
- You cannot use reserved names such as save and print.

09.4.1 Basic Data Types

| | Data Type Name | Type | Data Range and Remarks |
|-------------|----------------|----------|---|
| Non-numeric | String | Text | Text. |
| | Date | Date | Date and time. |
| | Boolean | Boolean | True or False. |
| | Variant | Any type | Default type if the variable is not declared with any data type. It will accept any kind of data. |
| Numeric | Byte | Numeric | Whole number between 0 and 255. |
| | Integer | Numeric | Whole number between -32,768 and 32,767. |
| | Long | Numeric | Whole number between – 2,147,483,648 and 2,147,483,647. |
| | Single | Numeric | Floating decimal number between -3.402823E38 and 3.402823E38. |
| | Double | Numeric | Floating decimal number between -1.79769313486232D308 and 1.79769313486232D308. |

09.4.2 Scope of Variables

Procedure Level - Visible only within a Sub or Function.

E.g.-

```
Sub procedureLevel()  
    Dim exchangeRate As Integer  
    exchangeRate = 180  
    MsgBox "USD to LKR " & exchangeRate  
End Sub
```

Module Level Scope - Visible within a module and all procedures within the module.

E.g.-

```
Dim exchangeRate As Integer  
  
Sub moduleLevel_subroutine1()  
    exchangeRate = 180  
End Sub  
  
Sub moduleLevel_subroutine2()  
    MsgBox ("USD to LKR " & exchangeRate)  
End Sub
```



Activity 09.5

Open a new excel workbook and save it as a macro enabled excel file, giving the name “Activity 09-5”.

Part A

Create an excel macro that calculates the area of a rectangle using variables. The user should input the length and width, and the macro should display the calculated area in message box. (Name the macro as “CalArea” and assign it to a button named “Calculate Area”)

Part B

Type your first name in “C2” cell and last name in “D2” cell. Create an excel macro to show your full name (as “Full name : _____”) in “C4” cell. You need to use variables for this.

09.5 Conditional logics (IF Statement)

Conditional expressions evaluate to being either true or false and are important to signal the computer to take one action (or set of actions) if the condition is true and another set of actions if the condition is false.



Activity 09.6

Open a new excel workbook and save it as a macro enabled excel file, giving the name “Activity 09-6”.

Part A

Assume you have a product price in “A1” cell. Create a macro that calculates a 10% discount for products priced above Rs. 5000 and leaves the price unchanged if it's Rs. 5000 or less. Display the discounted price in “B1” cell. Name the macro as “CalculateDiscount”.

Part B

Create the following interface in the worksheet and check the eligibility for the course which meets the following conditions.

If Subject 01 and 02 marks are greater than 50, then eligible for the course. If either of the subjects has scored more than 90 marks, then eligible for the course. If any of the above conditions are not met, consider as not-eligible. You need to show the result in “E4” cell based on given conditions.

| | A | B | C | D | E | F |
|---|---|------------|------------|---|----------------------------|---|
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | Subject 01 | Subject 02 | | Eligibility for the Course | |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |
| 7 | | | | | Check Eligibility | |
| 8 | | | | | | |