**Module 9: Working with Ranges & Cells**

* **Selecting Ranges**

In VBA, you can select specific ranges of cells.

Sub SelectRange()

Range("A1:C5").Select 'Selects cells from A1 to C5

End Sub

Sub SelectCells()

Cells(10, 8).Select 'Selects cell at Row 10, Column 8 (H10)

End Sub

**Note:** Using .Select is not always necessary—you can directly work with ranges.

* **Formatting Ranges**

You can format cells (like font, color, borders).

Sub FormatRange()

With Range("A1:A5")

.Font.Bold = True

.Font.Color = vbWhite

.Interior.Color = vbBlue

.NumberFormat = "0.00" '2 decimal places

End With

End Sub

* **Clearing Ranges**

Different ways to clear cell content.

Sub ClearRange()

Range("B1:B10").ClearContents 'Only clears values

End Sub

Sub ClearFormat()

Range("C1:C10").ClearFormats 'Only clears formatting

End Sub

Sub Clear()

Range("D1:D10").Clear 'Clears everything (content + formatting)

End Sub

* **Dynamic Ranges**

Sometimes you don’t know the last row/column. VBA can find it dynamically.

Sub DynamicRangeExample()

Dim lastRow As Long

lastRow = Cells(Rows.Count, 1).End(xlUp).Row 'Last used row in Column A

Range("A1:A" & lastRow).Select 'Selects all data in Column A

End Sub

* **Copying & Pasting with VBA**

Copy and paste ranges with values, formats, or both.

Sub CopyPasteExample()

Range("A1:A5").Copy Destination:=Range("C1") 'Copy values & formatting

End Sub

'Alternative way

Sub CopyPasteValue()

Range("B1:B5").Copy

Range("D1").PasteSpecial xlPasteValues 'Only values

End Sub

Sub CopyPasteFormat()

Range("E1").PasteSpecial xlPasteFormats 'Only formatting

End Sub

**Summary of this module**

* Use Range and Cells to select cells.
* .Font, .Interior, .NumberFormat for formatting.
* .ClearContents, .ClearFormats, .Clear for clearing cells.
* Use Rows.Count and End(xlUp) to handle dynamic ranges.
* Use .Copy and .PasteSpecial for flexible copying.

**Real life example:-**

Create a sales report from raw data with proper formatting and header color and at the end give summary of total revenue.

|  |  |  |
| --- | --- | --- |
| **Product Name** | **Units Sold** | **Revenue** |
| Product 1 | 120 | 15,000.50 |
| Product 2 | 95 | 12,350.75 |
| Product 3 | 140 | 21,000.00 |
| Product 4 | 60 | 7,800.25 |
| Product 5 | 200 | 35,000.00 |
| **Total** | **615** | **91,151.50** |

Sub Sheet3\_Button2\_Click()

Dim ws As Worksheet

Dim lastRow As Long

Dim tbl As ListObject

Dim rng As Range

' Use active sheet to avoid subscript errors

Set ws = ActiveSheet

' Find last row based on Column A

lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

' Clear existing table if any

On Error Resume Next

ws.ListObjects(1).Unlist

On Error GoTo 0

' Define data range

Set rng = ws.Range("A1:C" & lastRow)

' Create table

Set tbl = ws.ListObjects.Add(xlSrcRange, rng, , xlYes)

tbl.Name = "SalesTable"

tbl.TableStyle = "TableStyleMedium9"

' Format columns

ws.Range("B2:B" & lastRow).NumberFormat = "0"

ws.Range("C2:C" & lastRow).NumberFormat = "?#,##0.00"

' Add Totals

tbl.ShowTotals = True

tbl.ListColumns("Revenue").TotalsCalculation = xlTotalsCalculationSum

tbl.ListColumns("Units Sold").TotalsCalculation = xlTotalsCalculationSum

' Format header manually (optional)

With ws.Range("A1:C1")

.Interior.Color = vbBlue

.Font.Color = vbWhite

.Font.Bold = True

End With

MsgBox "Report generated successfully!", vbInformation

End Sub