**Module 7: Debugging & Error Handling**

* **Debugging in VBA**

When you write VBA code, it may not work as expected the first time. Debugging tools help you **find and fix errors**.

* **Debug Toolbar**
* Found in the VBA editor (View > Toolbars > Debug).

1.Place a [command button](https://www.excel-easy.com/vba/create-a-macro.html#command-button) on your worksheet and add the following code lines:

Sub Test\_Error()

x = 2  
Range("A1").Valu = x

End Sub

Click the command button on the sheet.

Result:

A screenshot of a computer

AI-generated content may be incorrect.

2. Place a [command button](https://www.excel-easy.com/vba/create-a-macro.html#command-button) on your worksheet and add the following code lines:

Option Explicit

Sub Button3\_Click()

'Dim x As Integer

x = 2

Range("A6").Value = x

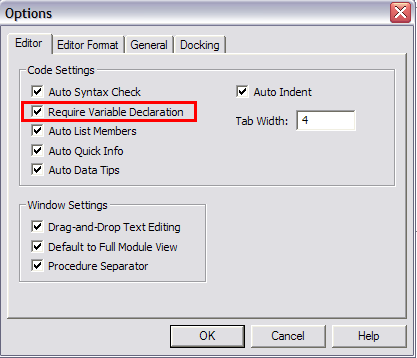
End Sub

A screenshot of a computer error

AI-generated content may be incorrect.

Steps to Enable Option Explicit Automatically

1. Open the VBA Editor
   * In Excel (or Word, PowerPoint, Access), press Alt + F11.
2. Go to the menu:  
   Tools → Options…
3. In the Options dialog box, under the Editor tab,  
   ✅ Check the box "Require Variable Declaration".
4. Click OK.



* **Debugging in Excel VBA**

This example teaches you how to debug code in Excel VBA.

**Single Step**

By pressing F8, you can single step through your code. This is very useful because it allows you to see the effect of each code line on your worksheet.

Place a [command button](https://www.excel-easy.com/vba/create-a-macro.html#command-button) on your worksheet and add the following code lines:

Sub Button4\_Click()

Dim i As Integer, j As Integer

For i = 1 To 2 ' Rows 1 to 2

For j = 1 To 5 ' 5 columns wide

' Start from row i, column 6 (F) and move right

Cells(i, j + 5).Value = WorksheetFunction.RandBetween(20, 100)

Next j

Next i

End Sub

Result when you click the command button on the sheet:

Excel VBA enters a random number between 20 and 100 into the cell at the intersection of row 1 and column 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 78 | 90 | 38 | 70 | 89 |
| 74 | 95 | 99 | 89 | 90 |

Open the [Visual Basic Editor](https://www.excel-easy.com/vba/create-a-macro.html#visual-basic-editor) and reduce the size of the screen so that you can see the Visual Basic Editor and worksheet at the same time.

place your cursor before Private and press Ctrl+F8.

The first line turns yellow.

Than press Shift+F8

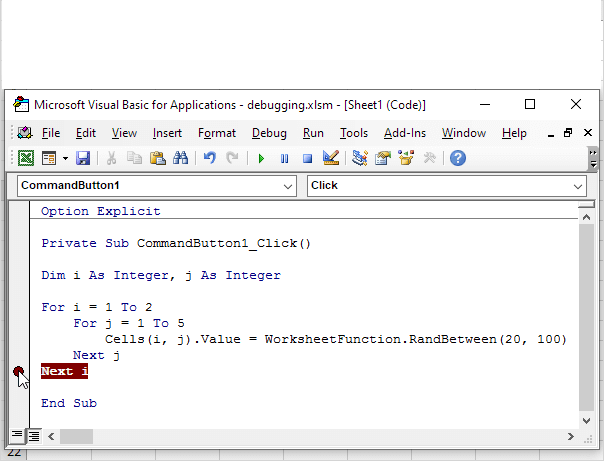
You will see how program run step by step and fill the random no. in each cell.

**Breakpoint**

You set a breakpoint to halt execution at a specific code line.

1. Empty the range F1:J2.

2. To set a breakpoint, click on the left margin (in grey) where you want to place the breakpoint. A red dot appears.



Click on the green arrow to execute the macro until the breakpoint.

Only part of the macro (for i = 1) has been executed.

 To remove the breakpoint, click on the red dot. Next, click on the green arrow to continue execution.

Note:- If you receive a macro from someone else, remember, debugging is great way to understand the code. You can easily see the effect of each code line on your worksheet.



* **Error handling using On Error Resume Next & On Error GoTo**

Below we will look at two programs in Excel VBA. One program simply ignores errors. The other program continues execution at a specified line upon hitting an error.

Both programs calculate the [square root](https://www.excel-easy.com/examples/square-root.html) of numbers.

Square Root 1

Add the following code lines to the 'Square Root 1' command button.

Sub Button5\_Click()

Dim rng As Range, cell As Range

'Work on selected cells

Set rng = Selection

'Loop through each cell in the selection

For Each cell In rng

On Error Resume Next 'skip errors (e.g., text or negative numbers)

cell.Value = Sqr(cell.Value)

On Error GoTo 0 'reset error handling

Next cell

End Sub

Conclusion: Excel VBA has ignored cells containing invalid values such as negative numbers and text.

Square Root 2

Add the following code lines to the 'Square Root 2' command button.

Sub Button6\_Click()

Dim rng As Range, cell As Range

'Work on selected cells

Set rng = Selection

'Loop through each cell in the selection

For Each cell In rng

On Error GoTo InvalidValue

cell.Value = Sqr(cell.Value)

Next cell

Exit Sub 'Prevents running the error handler if no error occurred

InvalidValue:

MsgBox "Can't calculate square root at cell " & cell.Address

Resume Next 'Continue with the next cell

End Sub

Conclusion: Excel VBA has not ignored cells containing invalid values such as negative numbers and text.