ADD TRANSPOSE TO MICROSOFT WORD

This guide contains full code for the macro to Transpose tables in word. This guide goes one step further and will walk you through inserting a module, pasting the code, and then creating a new button in your Ribbon. This way, the transpose functionality will always be just one click away!
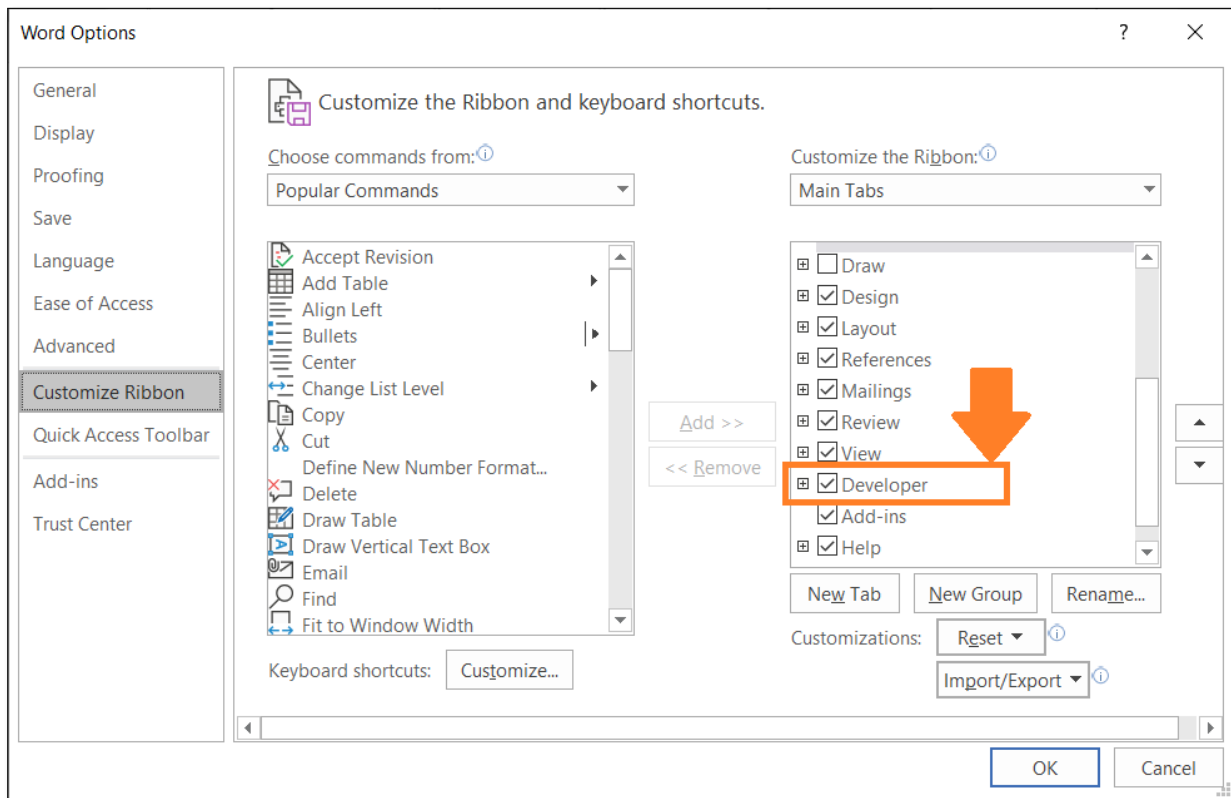
Know the basics? Skip everything download code from https://coderecode.com/transpose-code or end of this eBOOK.

CONTENTS

HOW TO ENABLE DEVELOPER TAB

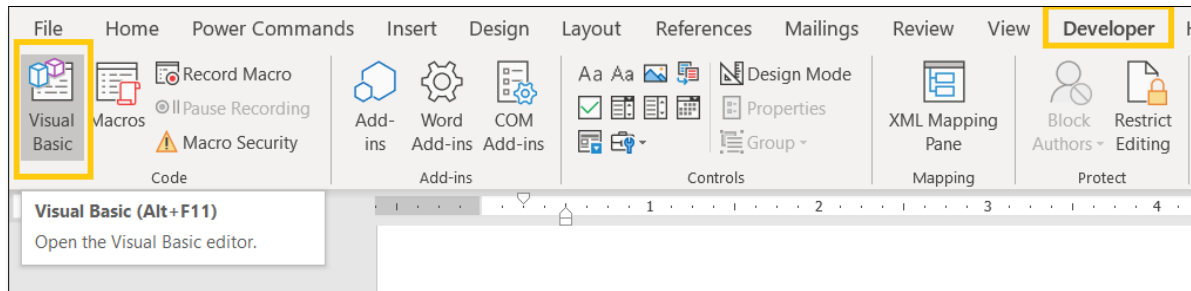1. Right click anywhere on the Ribbon and select Customize the Ribbon.

1 - Enable Developer Tab

2. Select Developer on the right column, as shown below:
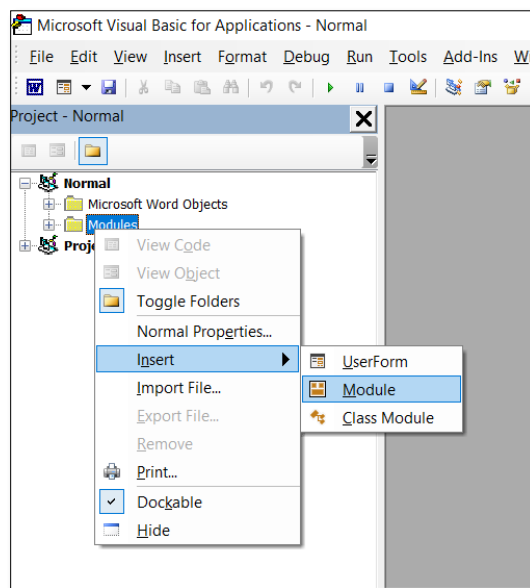
HOW TO INSERT MODULE AND PASTE CODE

1. Click Visual Basic under Developer toolbar.

Hint: You can directly press ALT + F11 to open Visual Basic Editor, even without enabling Developer Tab. This is the fastest way and works on all version of Word.

2. Insert a new module under Normal template and paste the code. Even if you already have a new module, it helps to create one more module to keep the code organized.
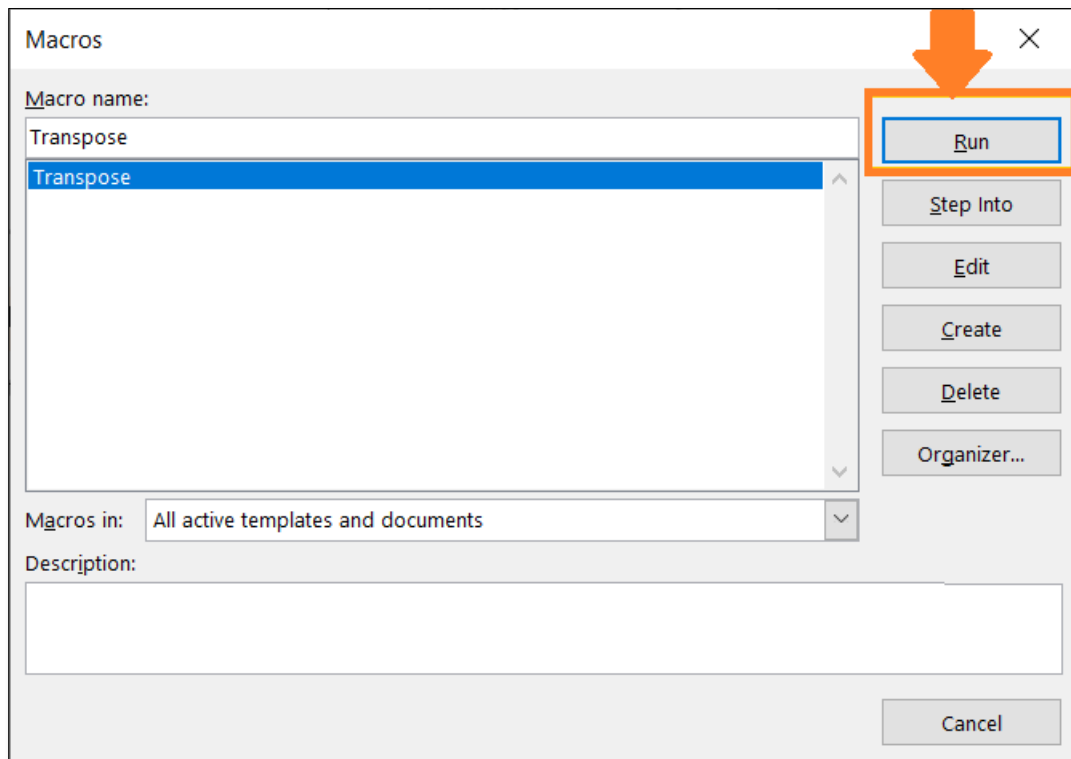

3 - Insert a New Module

3. Paste the code in the new module. The code is in the last section of this eBook.

HOW TO RUN THE MACRO

There are multiple ways to run a macro.

Option 1. Go to Developer tab and click Macros. This will bring the following dialog. Select Transpose and click Run.

Option 2: Pres `Alt + F8` keys and this will bring the same Run dialog. Select Transpose and click Run.
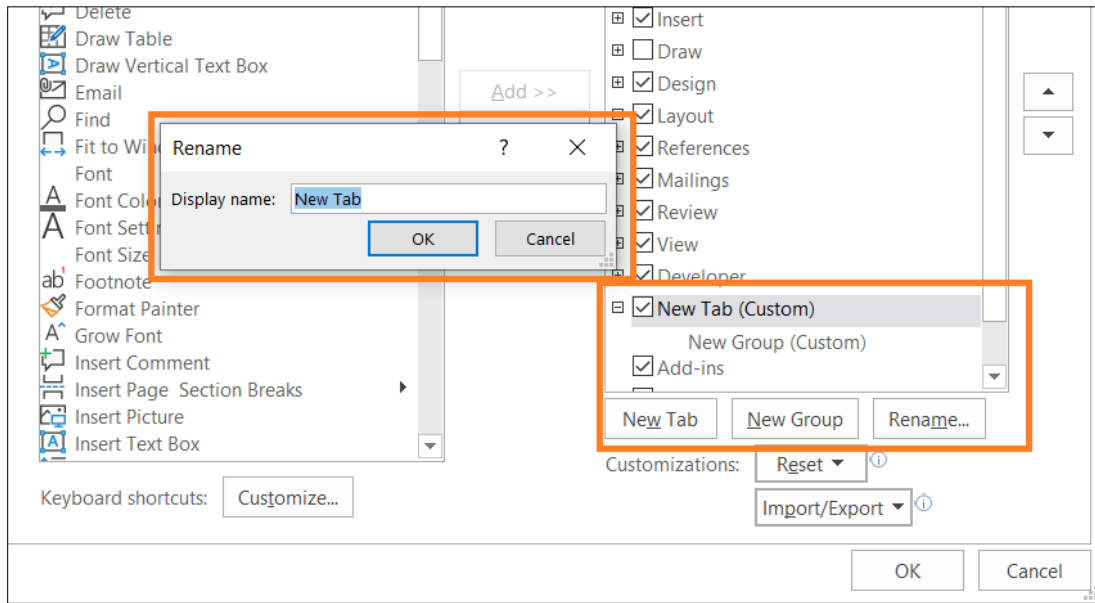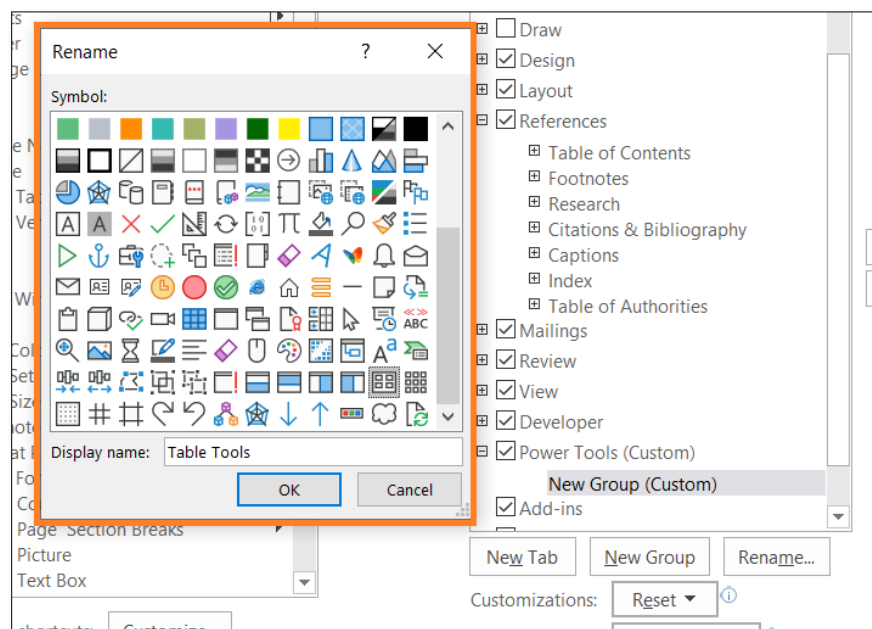

4 - **Macros Dialog**

Option 3: Add a shortcut in the Ribbon to have this functionality nicely added to MS Word. Jump to the next section to understand how to do this.

ADD A SHORTCUT IN THE RIBBON

1. Right click anywhere on the Ribbon and select Customize the Ribbon. You will see the dialog seen earlier 1 - Enable Developer Tab
2. Select New Tab to create a new Tab and New Group to create a new group. Select New Tab and click Rename and type a name that you like.
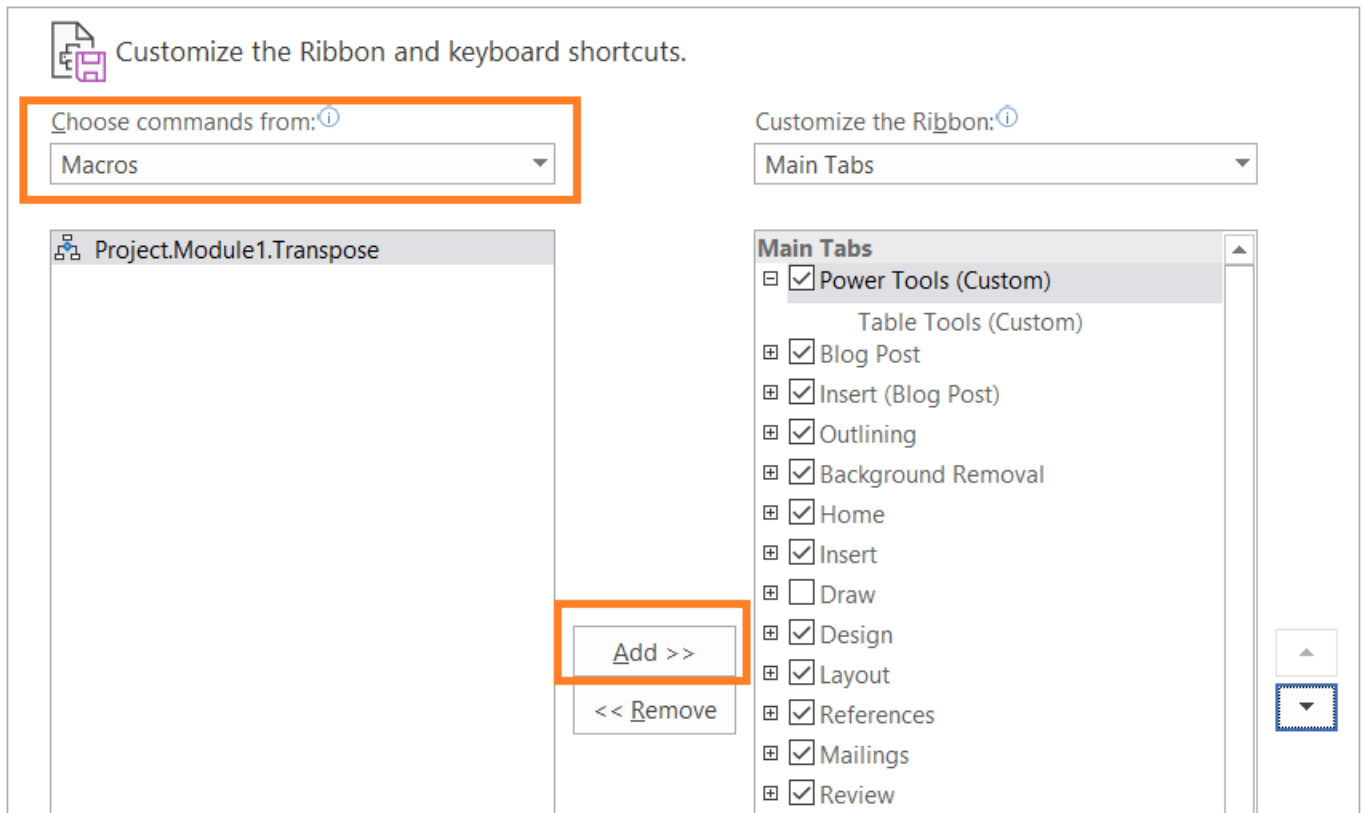

3. Then select New Group and click Rename to rename the group. In this Rename dialog, you can even assign an icon for the group. This would be a good place to logically group functionalities. This is NOT the name of the macro.
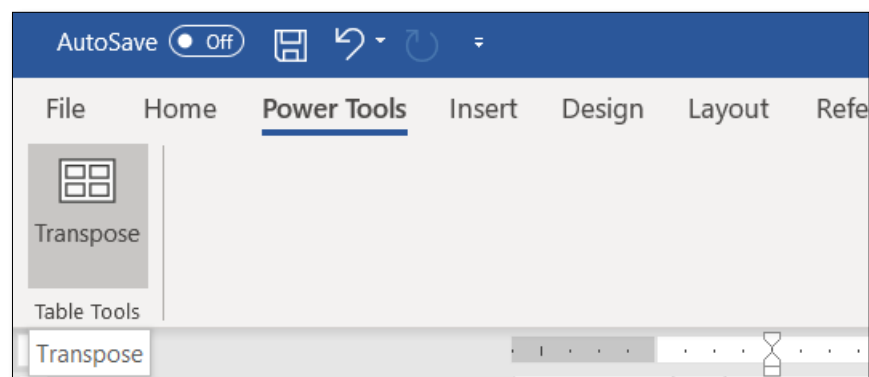

5 - Create New Tab

4. At the top of the of the **left** column, you would see a drop down with default value as Popular Command. Change it to Macro.
5. Important: Confirm that on the right column, your new newly created Group under your group is selected and click Add.

6 - Add Macro to the Group

6. Select the macro name on the right column and Rename it and select any Icon.
7. Click OK to close the dialog

Congratulations! You just added a new Tab with a button that executes a macro.

THE CODE

Copy the code and paste in the module created in previous step.

```vba
Public Sub Transpose()
    'Declare Variables
    Dim SourceTable As Table
    Dim RowCount As Long, ColumnCount As Long
    Dim TableRange As Range
    Dim i As Long, j As Long 'Loop Counters
    Dim RowDataAsArray() As String
    Dim NewTable As Table
    Dim SourceTableStyle As Style
    Dim TableAsArray() As String 'Will contain the table text in memory

    'If cursor is not in a table, exit macro
    If Not Selection.Information(wdWithInTable) Then
      MsgBox "Cursor should be in a table"
      Exit Sub
    End If

    Set SourceTable = Selection.Tables(1)
    RowCount = SourceTable.Rows.Count
    ColumnCount = SourceTable.Columns.Count
    Set SourceTableStyle = SourceTable.Style

   'Redefine array as a two dimensional array with exact row and column count
    ReDim TableAsArray(1 To RowCount, 1 To ColumnCount)

    For i = 1 To RowCount
      RowDataAsArray = Split(Expression:=SourceTable.Rows(i).Range.Text, _
                Delimiter:=vbCr)
      For j = 1 To ColumnCount
        'Last item in RowDataAsArray is vbCr, thus j - 1 to ignore that
        TableAsArray(i, j) = RowDataAsArray(j - 1)
      Next j
    Next

    Set TableRange = SourceTable.Range
    TableRange.Collapse wdCollapseEnd
    SourceTable.Delete

    'Create a new table at the same position
    Set NewTable = TableRange.Tables.Add(TableRange, ColumnCount, RowCount)
    'Fill data in the new table
    For i = 1 To RowCount
      For j = 1 To ColumnCount
        NewTable.Rows(j).Cells(i).Range.Text = TableAsArray(i, j)
      Next
    Next
    'Apple Style to the new table
    NewTable.Style = SourceTableStyle
  End Sub
```