Excel numberformat percentage

04-02-2011, 08:05 #1 Hi, I have this cell on a spreadsheet called abc, and I changed the properties of this cell to a percentage. And I have this code in VBA format (range (abc), Percent) When I make an msgbox in the line above, all I see is .10 Can someone tell me what the problem is? 04-02-2011, 08:12 PM #2 fixed it I changed it to 0% 0% 04-02-2011, 08:25 PM #3 Doctortt, Here's what I found works: [vba]Sub MyLittleSub() Dim wb As a Workbook, ws Worksheet As Set wb = ActiveWorkbook Set ws = ActiveSheet With ws . Area (abc). NumberFormat = 0.00% End With End Sub [/vba] When the encoding effort does not work, I save the macro so that the data is in a percentage formatting cell with a value of 0.1. No, no, no, no. Sub Macro1 () ' Macro1 Macro ' Selection.NumberFormat = 0.00% End Sub [/vba] Then I adjust the code to use what I learned from the macro, in this case numberFormat = 0.00% and not Percent. I hope this helps. Now that I've sent my answer, I can see that I updated your findings, mostly what I found. Cheers, Home » VBA Functions » VBA-FormatPercent-FunctionDescription VBA FormatPercent function uses the percentage format in a numeric expression and returns the result as a string. The function syntax is:FormatPercent(Expression, [NumDigitsAfterDecimal], [IncludeLeadingDigit],[UseParensForNegativeNumbers], [GroupDigits])Where the function arguments are:Expression-Formattable

Expression. [NumDigitsAfterDecimal]-Optional numeric value that specifies the number of digits displayed after decimal place. If [NumDigitsAfterDecimal] is omitted, the default value is -1, which means that the regional settings on the computer must be used. [IncludeLeadingDigit]-Optional vbTriState enumerated value that specifies whether fraction values are displayed with a fore.1. This can have one of the following values:vbFalse-Do not display the leading zero.vbTrue-Display preference setting zero.vbUseDefault-Use the default computer settings. If no argument is specified, [IncludeLeadingDigit] is set to vbUseDefault. [UseParensForNegativeNumbers]-Optional vbTriState enumerated value that specifies whether negative numbers should be encased in parentheses. This can have one of the following values:vbFalse-Do not enter negative numbers in parentheses.vbTrue-Encase negative numbers in parentheses.vbUseDefault-Use the default computer settings. If no argument is specified, the [UseParensForNegativeNumbers] argument is set to vbUseDefault. [GroupDigits]-Optional vbTriState enumeration value that specifies whether the number should be grouped (to thousands, etc.) using the group separator specified by the regional settings on the computer. This can have one of the following values:vbFalse-Do not group numbers.vbTrue-Group digits.vbUseDefault-Use the default computer settings. If no value is specified, the [GroupDigits] argument is set to vbUseDefault.The following example shows how to use the VBA FormatPercent function to format numeric values. Percentages. Each example uses different formatting rules. Dim pc1 StringDim as pc2 StringDim pc3 as StringDim pc4 as Stringpc1 = FormatPercent(10)pc2 = FormatPercent(10, , , vbFalse)pc3 = FormatPercent(0,559, 0)pc4 = FormatPercent(-0,5, , vbTrue)Note that above in the call of the FormatPercent function:When only the Expression argument is given, the function uses the default percentage format for the current system. When [GroupDigits] is set to vbFalse, 10 is returned as a percent string of 1000.00% (without commas separating the number groups). When [NumDigitsAfterDecimal] is set to 0, the given expression 0.559 is rounded to 56% (zero decimal places) before being returned as 56%. When [UseParensForNegativeNumbers] is set to vbTrue, the negative value of -0.5 is returned as a string (50.00%). Also note that in each case, the result returned from the FormatPercent function is a String data type. VBA FormatPercent function errorIf the expression given to the FormatPercent function is a text string that cannot be converted to a numeric value, you will get an error:Run time error '13': Type conflict Error on the VBA function page Return to the Excel VBA tutorial page that I have been looking for this and it looks so simple, but I can't make it work. I have a table, and one column is formatted as a percentage. Below is my code, but it doesn't format the cells, it just leaves them decimal. I think this is because although the cell is defined as a range, it is actually the value of the cell, so I don't know how to refer to it as a range. The ReturnRebate variable is defined as a range and the loop rotates through the correct range. Code: Dim the cell as a range, p Double for each cell in returnRebate p = cell. Value If p <> 0 and p > 1, p = p * 0.01 cells. Value = p-cell. NumberFormat = Percentage 'No Formatting Here ElseIf p < 1 and p > 0 Then 'don't do anything about it Other cell. Value = vbNullString End If you enter 0.751 in the column next, it will appear as 75.1% If you enter 0.75, it will appear 75.% What do you want? I'm sorry, I put up a code I was messing with. All I want at the moment is for 15.00% to read 15%. But for future use, I'd like to adapt it to read any decimal place I want. Thank you, Shg Columns (B). NumberFormat = 0% (no decimal) Columns(B). NumberFormat = 0.0% (1 decimal place) Columns(B). NumberFormat = 0.00% (2 decimal places) Excel Tutorials For Macros Link 15 to the best tutorials has been sent to you, check your email to download it! (If you don't see the email, check your spam or campaign folder and make sure you add us as a contact so you can get our email in the future.) Build Professional - Unbreakable - Forms Excel 45 Tutorials - 5+ Hours - Downloadable Excel Files Instant Access! - Lifetime use! Edit more to favorites Favorites This free Excel macro formats the selection cells as a percentage in Excel. This simply changes the number format of the cell to Percentage Formatting. This tells you any number in 100 to display the correct percentage. This Excel macro is great for quickly formatting large cell, row, or column selections in Excel. Macro installation location: Module for formatting Excel macro cells as a percentage in sub Format_Cell_Percentage() in Excel number formatting. NumberFormat = 0.00% End Sub Link to the 15 best tutorials has been sent to you, check your email to download it! (If you don't see the email, check your spam or campaign folder and make sure you add us as a contact so you can get our email in the future.) Install the macro selector and copy text from the gray box above. Open the Microsoft Excel file in which you want the macro to work. Press Alt + F11 - This will open the Visual Basic Editor - Works on all versions of Excel. Or if you want other ways to get there, click here. Versions of Excel before Excel 2007 Go to tools > macros > Visual Basic Editor For Excel 2007 Go to Office button > Excel Options > Popular > Select the Ribbon View Developer tab. Then go to the Developer tab on the ribbon menu and click On the left side of the new window that opens Visual Basic, go to the left side where the vertical pane is located. Locate the Excel file. it is called VBAProject (YOUR FILE NAME HERE) and click here. If the macro moves to the module, click Here, otherwise continue to step 8. Go to the menu at the top of the window and click more > Another window should have opened in the Visual Basic Editor window. Paste the macro code into this new window. Make sure you paste the code under everything else in the window. Go to step 8. If the macro moves to a workbook or ThisWorkbook, click Here, otherwise continue to step 8. Open the drop-down list by clicking the Microsoft Excel Objects folder icon directly under an Excel file named VBAProject (file name here). Then, in the list that appears, double-click thisWorkbook text. A new window appears inside the Visual Basic Editor window. In this new window, paste the macro code. Be sure to paste this code under the last line of any other code already in the window. Go to step 8. If the macro moves to the worksheet code, click here, otherwise continue to step 8. Open the drop-down list by clicking the Microsoft Excel Objects folder icon directly under an Excel file named VBAProject (file name here). The list that appears displays all worksheets in the Excel file. They are listed as such: Sheet1 (SHEET NAME HERE) and below it is Sheet2 (SHEET NAME HERE). Select the table in which you want the macro to run, and then double-click the table. New window inside Visual Basic window appears. In this new window, paste the macro code. Be sure to paste this code under the last line of any other code already in the window. Repeat steps (b) and (c) for each sheet on which you want the macro to work. Placing a macro in one table does not apply it to other sheets in the workbook. Go to step 8. Close the Microsoft Visual Basic Editor window and save the Excel file. When you close the Visual Basic Editor window, a standard Excel window does not close. You can now run the macro. Macro.