

## Excel 2016 Function Macro Basics

Microsoft Excel© contains many useful functions that simplify calculations. Sometimes Excel doesn't have a built-in function that fits the problem. A *function macro* is the answer when calculations must be performed repeatedly and there isn't an existing Excel function. This document is a short introduction to making function macros using Visual Basic.

If the **Developer** tab is not available, do the following to display it: Click the **File** tab, and then click **Options**. Click on the **Customize Ribbon** option and under the **Main Tabs** category, check **Developer** and then click **OK**. On the **Developer** tab, in the **Code** group, click **Visual Basic** or it can be started by typing **Alt** and **F11**. In the left column entitled "Project - VBA Project" double-click "VBAProject (Book1)." Then open the **Microsoft Excel Objects** file and right click on **Sheet1(Sheet1)** → **Insert** → **Module**. Underneath "VBAProject (Book1)" an icon entitled "Module 1" will appear in the Modules folder and a window will open entitled "Book1 - Module 1 (Code)." This is where the function macro will be created.

The easiest way to learn how to write function macros is to take an example, and then modify it. In the code window type the following practice function.

```
Function Volume (length, width, height)
Volume = length * width * height
End Function
```

This function macro creates a new function that will calculate the volume of a box. To use this function, close the Visual Basic Editor, select a cell in your Excel spreadsheet, and then enter "=Volume(1,2,3)." The function will multiply the three numbers within the parenthesis and display the result in the selected cell. The order of the arguments of the function within the spreadsheet will coincide with the order of the variables within the function macro's argument list. For this example, the value of 1 will coincide with variable "length", 2 with "width", and 3 with "height".

Now save the Excel document. If using a University computer, save the file to the Temp folder on the C drive or a USB memory stick.

Return to Visual Basic Editor and insert another module. Now enter the following code to create a function that calculates a fine from a speeding ticket. To define variables using more than one word, use the underscore character ( `_` ) instead of a space.

```
'A function macro to calculate the cost of a speeding ticket
Function Fine(speed_limit, my_speed, cost_per_mph_over)
'This is how much over the speed limit I was going
Excess = my_speed - speed_limit
'I wasn't speeding
If Excess < 0 Then Fine = 0
'I was going the speed limit
If Excess = 0 Then Fine = 0
'I was speeding
If Excess > 0 Then Fine = Excess * cost_per_mph_over
End Function
```

An apostrophe is used to make comments in Visual Basic code. The variable "Excess" is not an input variable for the function. It is used in an intermediate calculation, but more importantly, it is used to determine a positive whether or not the individual was speeding.

More complex functions are possible. Practice with function macros is beneficial.

For extensive information on Microsoft© Visual Basic, visit Microsoft's online library.