Master Microsoft Excel Macros and VBA

Project #2 – Sorting a List

Author: Kyle Pew (Microsoft Certified Trainer – MCT)

Course Reviews

+1 *"This is a great course. I love how the lessons are only about 4 minutes each.* ***It makes it possible to learn a lot in a short amount of time.*** *It seems targeted to new learners but it also makes for a great review even if you are familiar with Microsoft Office… 5 stars!" - Wendy*

+1 *"Great visual over the shoulder presentations by a very articulate instructor. The simple tips on Word and PowerPoint alone were well worth taking the course. What I learned will not only save time, but will end much of the frustration I have experienced with PDF's. Recommend the course." – Bill*

+1 *"These presentations are very well put together.* ***The instructor keeps you engaged and is easy to follow.*** *" - Karen*

Contents

Project #2 - Overview

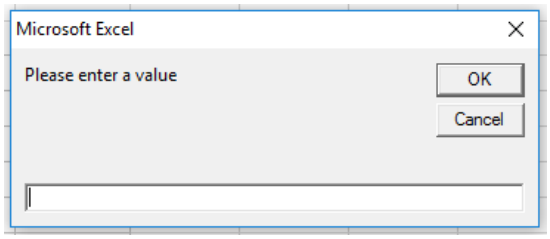
During this project you will be mastering 3 VBA concepts that you will use throughout your Excel Macro/VBA career.

- Gathering Input from your Macro Users
- Using IF Statements to build logic in your code
- Using Message Boxes (MsgBox) to message information to the users

To practice these concepts you will be taking the list from the precious project and creating an interface that will allow the users to sort the list in 1 of 3 ways.

Prompting the User for Input Using an InputBox

Built into the Visual Basic for Applications language is an object called an InputBox.



Using an InputBox is a great way to get input/values from the user of your macro. The InputBox comes with a few parameters that you will need to fill out in order to use the InputBox.

InputBox(prompt, title, default, xpos, ypos, helpfile) as string

- **Prompt**: The message to the user that appears on the Input Box (ex. Please enter a value)
- **Title**: The text that shows up at the top of the Input Box (ex. Sort Box)
- **Default**: Any default value that you want the Input Box to contain (ex. Enter a Value here)
- **Xpos/Ypos**: The position of the InputBox on the screen
- **HelpFile**: Any associated help file for the user

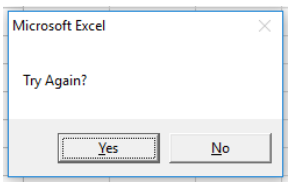
Building Logic in Your Macros Using an IF Statement

A huge part of creating Macros with VBA is the ability to add logic based code to your Macros. One of the most common forms of logic based code comes through an IF Statement.

```
If userInput = "1" Then
    ' Do Something
ElseIf userInput = "2" Then
    ' Do something else
Else
    ' Do this if the above isn't true
End If
```

Using a MsgBox to Message the User

Where InputBoxes ask the user to fill in a value a Message Box (MsgBox) object displays information to the user and gives them optional buttons to click.



Much like the InputBox, the MsgBox has a few parameters that you can fill in when you use it.

MsgBox(prompt, buttons, title, helpfile, context)

- **Prompt**: The message to the user
- **Buttons**: a value representing the type of buttons on the message box (ex. vbYesNo)
- **Title**: Text that appears at the top of the message box
- **HelpFile**: Any associated help file for the user
- **Context**: Numeric value to associated help file

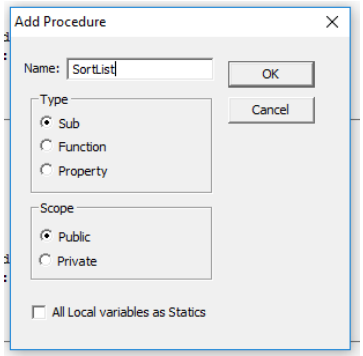
A common practice of a MsgBox is to the prompt the user with a message and then build logic to detect which of the buttons the user pressed. For example; an error may have occurred, so you alert the user with a MsgBox and then prompt them to Try Again. If they click the Yes button the MsgBox returns a 6 back to the code.

```
If userInput = "1" Then
    ' Do Something
    test = MsgBox("Try Again?", vbYesNo) ' if user click yes the message box returns a 6
ElseIf userInput = "2" Then
    ' Do something else
Else
    ' Do this if the above isn't true
End If
```

Level Up by doing it yourself!

You decide to take the Expense report list that your co-worker has given you and add an input box prompting the user for a value to sort the list. You will build logic in your macro, using an IF statement to check what value the user input and based on that value you will sort the list appropriately.

1. Download and open the file: **SortingRecords.xlsm** provided in this lecture
2. Open the Visual Basic Editor window **VIEW – MACROS – VIEW MACROS – EDIT (Alt + F11 is the shortcut key)**
3. Make sure **MODULE1** is open. Double Click **Module1** in the **Project** window on the left
4. Observe the 3 Macros/Procedures that are in **Module1.** Each of these macros sort the list based on a different column
    a. **DivisionSort**: Sorts the list by Division
    b. **CategorySort**: Sorts the list by the Category
    c. **TotalSort**: Sorts the list by the Total Column
5. Insert a new **PROCEDURE** into **Module1**
    a. **INSERT -- PROCEDURE**
    b. Name the **PROCEDURE "SortList".** Leave the other options as default
    c. Click OK
6. Between the **SUB** and **END SUB** lines, enter the following code



```
Dim userInput As String
Dim tryAgain As Integer

userInput = InputBox("1 = Sort by Division, 2 = Sort by Category, 3 = Sort by Total Sales")

If userInput = "1" Then
    DivisionSort
ElseIf userInput = "2" Then
    CategorySort
ElseIf userInput = "3" Then
    TotalSort
Else
    tryAgain = MsgBox("Incorrect Value. Try Again?", vbYesNo)

    If tryAgain = 6 Then
        SortList
    End If
End If
```

7. Close the Visual Basic Editor window
8. Add a button to the **Quick Access Toolbar** to run your new Macro
9. Test the Macro by clicking the new button

Instructor BIO:

Kyle is a Microsoft Certified Trainer (MCT) and a certified Microsoft Office Master Instructor and has been teaching and consulting for the past 10+ years on various computer applications, including;

1. Microsoft Office Suite 1997, 2000, XP, 2003, 2007, 2010, 2013
    1. Excel, Word, PowerPoint, Outlook, Access and Visio
2. SharePoint End-User 2007, 2010, 2013
3. VBA (Excel and Access)
4. Adobe Suite
    1. Photoshop, Illustrator, InDesign
5. Maya (Modeling and Animation)
6. Unity3d (Game Design)
7. HTML, CSS and JavaScript
8. Crystal Reports

Kyle is a graduate of the San Francisco Art Institute in the Media Arts and Animation Program. He has worked as a Game Designer for Electronic Arts, designing on games such as Nerf N-Strike, Nerf N-Strike Elite, both for the Wii, and a Sims 3 Expansion Pack for the PC.

Kyle has facilitated courses that range from 1-on-1 interactions to large scale groups of 100+ participants, including; live in person classes, webinar style classes online and live online full courses. He is consistently ranked top in reviews for each of the courses he teaches.

What Students have said about Kyle:

- ***"Kyle was off the chart "GOOD""***
- ***"ONE OF THE BEST COURSES THAT I'VE HAD… (IN 12+ YEARS)."***
- ***"Awesome trainer because I'm computer "stupid" and he helped me understand it."***
- ***"Kyle Pew is very knowledgeable and presented information with exceptional skill."***

In his 10+ years of training (corporate training, 1-on-1 consulting and college courses), Kyle has taught 1000's of courses and 10's of thousands of students all the while maintaining a high level of delivery and satisfaction from the student he has taught.

Teaching Philosophy:

Kyle believes that student's best learn through the application of real-life business situations through exercises. Allowing students to guide the class with their specific situations allows for the quickest and easiest adaptation to new technology and skills.

Check out my Udemy profile for more information and more courses.
https://www.udemy.com/user/kyle-pew/