



## Master Microsoft Excel Macros and VBA

### Project #1 – Inserting and Formatting Text

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 #1 - Overview.....	3
Using the Macro Recorder .....	3
Running a Macro .....	4
Editing a Macro (VBE) .....	4
Level Up by doing it yourself! .....	4
Instructor BIO:.....	6

# Project #1 - Overview

During this project you will utilize Excel's Macro Recorder tool to add and format headers for a simple list in Excel.

East	Technical Support	800	650	700	2150
East	Telephone	900	850	850	2600
East	Copying	4850	3200	1155	9205
East	Overhead	1250	1250	1250	3750
East	Software	2025	2200	1650	5875
East	Maintenance	1350	1500	1700	4550
East	Supplies	3300	3500	3700	10500
East	Telemarketing	3825	3725	3750	11300
East	Contractors	8900	10315	5250	24465
East	Consultants	6250	6000	6500	18750
East	Rent	8000	8000	8000	24000
East	Miscellaneous	11500	12500	12500	36500
East	Advertising	12250	12250	12750	37250
East	Clerical Support	25000	24000	26390	75390

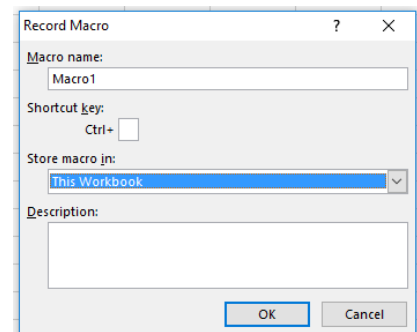
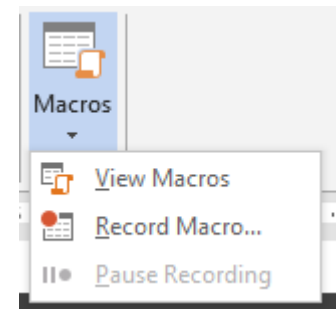
1st Qtr Expense Report						
Division	Category	Jan	Feb	Mar	Total	
East	Technical Support	800	650	700	2150	
East	Telephone	900	850	850	2600	
East	Copying	4850	3200	1155	9205	
East	Overhead	1250	1250	1250	3750	
East	Software	2025	2200	1650	5875	
East	Maintenance	1350	1500	1700	4550	
East	Supplies	3300	3500	3700	10500	
East	Telemarketing	3825	3725	3750	11300	
East	Contractors	8900	10315	5250	24465	
East	Consultants	6250	6000	6500	18750	
East	Rent	8000	8000	8000	24000	
East	Miscellaneous	11500	12500	12500	36500	
East	Advertising	12250	12250	12750	37250	
East	Clerical Support	25000	24000	26390	75390	

## Using the Macro Recorder

Excel's Macro Recorder tool is the fastest way to create automation in Excel. Upon activating the Macro Recorder, Excel will record your actions within the application, writing down each step you take in Visual Basic for Applications (VBA) code. All you need to do is click RECORD MACRO, fill in the options, and start recording any Excel commands that will would like to automate

### VIEW – MACROS – RECORD MACRO

- **MACRO NAME**
  - Up to 255 Characters
  - No Space or Special Characters (@, #, \$, &)
  - Don't Start your Macro Name with a numeric value
- **Short-Cut Key**
  - You can use CTRL + a character or CTRL + SHIFT and a character
  - Stay away from using short-cut keys already used by Excel as this will over-ride the default before and will run your macro instead
- **STORE MACRO IN**
  - **This Workbook:** Storing Macro here will give you access to the macro in this workbook, the workbook you are creating the macro in
  - **Personal Macro Workbook:** Storing the Macro here will give you access to the macro in any workbook on your system
  - **New Workbook:** This creates a new workbook at time of creating the macro and the macro will be available if that workbook is open
- **Description:** Leave some notes behind for yourself and any co-workers that may use your macro



## Running a Macro

There are many ways to run Macros from within Excel. Below are a few ways you can run Macros.

- **Use a Shortcut Key:** The shortcut key is created when you first create the Macro
- **Create a Button on the Worksheet:** Buttons can be added to a worksheet and then tied to a Macro
  - You can find the **COMMAND BUTTON** control on the developer tab within Excel
- **Add a button on the Quick Access Toolbar**

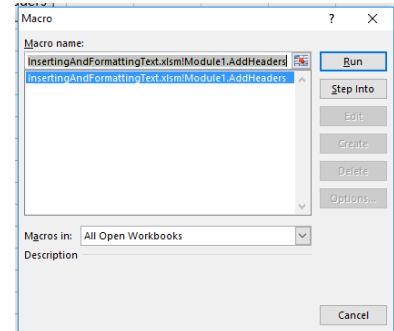


## Editing a Macro (VBE)

In order to edit Macros that you have created using the Macro Recorder tool you must open the **Visual Basic Editor window**.

### VIEW – MACROS – VIEW MACROS – EDIT

The **Visual Basic Editor** or **VBE** window is where all the code that the Macro Recorder wrote is stored. The code is stored in an object called a **MODULE**.



### Level Up by doing it yourself!

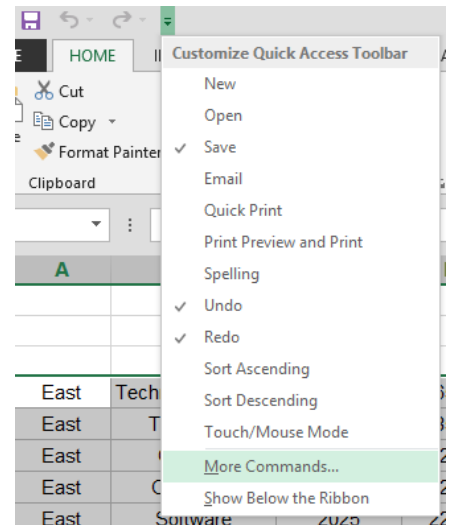
A co-worker has given you an Excel document that contains raw data, a list of expense records. The list contains several pieces of data but is missing the headers for each of its columns and a Worksheet title. Each time you receive one of these documents you have to add the headers and the worksheet title manually. You decide to automate the adding of the column headers and worksheet title by creating a Macro using Excel's Macro Recorder tool.

### RECORD THE MACRO

1. Download and open the file: **InsertingAndFormattingText.xlsx** provided in this lecture
2. Observe the records on the worksheet titled **EAST RECORDS** the list contains data about expenses but is missing the appropriate headers and a Worksheet title
3. Record yourself, using the Macro Recorder tool, add the headers and the worksheet title
4. Goto **VIEW—MACROS – RECORD MACRO**
5. Call the Macro "**AddHeaders**"
6. Add the following headers to the list in ROW 3
  - a. Division
  - b. Category
  - c. Jan
  - d. Feb
  - e. Mar
  - f. Total Expense
7. Add a worksheet title in CELL C1, "EAST EXPENSE REPORT"
8. Format the headers and the worksheet title
9. Stop Recording – **VIEW – MACROS – STOP RECORDING**

## ADD A BUTTON TO RUN MACRO

1. Add a button to the **QUICK ACCESS TOOLBAR** that runs your Macro
2. Test the Macro with the button on the **RUN WITH BUTTON** worksheet



## 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/>



## 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.....	3
Prompting the User for Input Using an InputBox .....	3
Building Logic in Your Macros Using an IF Statement.....	3
Using a MsgBox to Message the User.....	3
Level Up by doing it yourself! .....	4
Instructor BIO:.....	6



## Project #2 - Overview

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

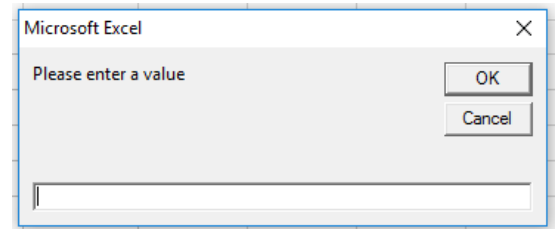
- Gathering Input from your Macro Users
- Using IF Statements to build logic in your code
- Using Message Boxes (MsgBox) to message information to the users

To practice these concepts you will be taking the list from the previous 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

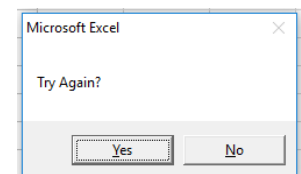
A huge part of creating Macros with VBA is the ability to add logic based code to your Macros. One of the most common forms of logic based code comes through an IF Statement.

```
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 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.

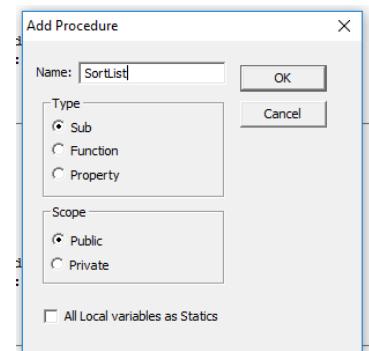
```
If userInput = "1" Then
    ' Do Something
    test = MsgBox("Try Again?", vbYesNo) ' if user click yes the message box returns a 6
ElseIf userInput = "2" Then
    ' Do something else
Else
    ' Do this if the above isn't true
End If
```



### 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/>



## Master Microsoft Excel Macros and VBA

### Project #3 – Preparing and Cleaning Up Data

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 #3 - Overview.....	3
Working with Excel VBA Loops .....	3
Level Up by doing it yourself! .....	4
Instructor BIO: .....	6

## Project #3 - Overview

In order to complete project #3 you are going to combine using both the Macro Recorder and writing your own custom Excel VBA code. The Macro Recorder will be used to record routine tasks that are easily performed in the Excel interface. Then you will move into the Visual Basic Editor to write a custom procedure that makes calls to the recorded macros adding in looping logic to repeat the macros for each sheet in the workbook.

East	Technical Support	800	650	700	2150
East	Telephone	900	850	850	2600
East	Copying	4850	3200	1155	9205
East	Overhead	1250	1250	1250	3750
East	Software	2025	2200	1650	5875
East	Maintenance	1350	1500	1700	4550
East	Supplies	3300	3500	3700	10500
East	Telemarketing	3825	3725	3750	11300
East	Contractors	8900	10315	5250	24465
East	Consultants	6250	6000	6500	18750
East	Rent	8000	8000	8000	24000
East	Miscellaneous	11500	12500	12500	36500
East	Advertising	12250	12250	12750	37250
East	Clerical Support	25000	24000	26390	75390

1st Qtr Expense Report					
Division	Category	Jan	Feb	Mar	Total
East	Technical Support	800	650	700	2150
East	Telephone	900	850	850	2600
East	Copying	4850	3200	1155	9205
East	Overhead	1250	1250	1250	3750
East	Software	2025	2200	1650	5875
East	Maintenance	1350	1500	1700	4550
East	Supplies	3300	3500	3700	10500
East	Telemarketing	3825	3725	3750	11300
East	Contractors	8900	10315	5250	24465
East	Consultants	6250	6000	6500	18750
East	Rent	8000	8000	8000	24000
East	Miscellaneous	11500	12500	12500	36500
East	Advertising	12250	12250	12750	37250
East	Clerical Support	25000	24000	26390	75390

### Macro Recorder Macros

Using the Macro recorder you will record two macros.

1. **AddHeaders:** This macro will add the headers to the top of the list (Division, Category, Jan...)
2. **FormatList:** This macro will format the headers and the Jan-Mar data as currency

### Custom VBA Procedures

After the 2 initial macros are created using the Macro Recorder you will then build a custom procedure in the Visual Basic Editor where you will create a loop that performs the two recorded macros on each sheet in the workbook.

## Working with Excel VBA Loops

The Macro Recorder tool in Excel is a great tool, making the creation of Macros super easy. But, the Macro Recorder has some limitations. If you would like to add any logic or repeat task a fixed or unknown number of times, you will need to move away from the Macro Recorder and get into Excel VBA.

There are a few different approaches to creating loops within Excel VBA.

- Do While Loop
- For Next Loop

Based on the logic you wish to perform in your procedure will really dictate the type of loop you'll work with.

### DO WHILE LOOP

```
Do While ActiveCell.Value <> ""
    ' do something
Loop
```

A Do While Loop will continue to repeat a block of code while a condition is true. In the above example the code between the Do line and Loop line will continue to repeat until the active cell's value is not empty. This is a great setup for looping through a set of records while checking for specific conditions (ex. Format records where a date has expired)

## DO UNTIL LOOP

The counterpart to the Do While Loop is the Do Until Loop. This loop is very similar. The difference is in the logic the loop uses to determine if the condition to loop is still true.

```
Do Until ActiveCell.Value = ""  
    ' do something  
Loop
```

Rather than looping **while** a condition is still true, this one will continue to loop **until** a condition is true.

Both loops are valid techniques to use in Excel VBA. Which one you use is based on the logic needed to perform the loop.



## Level Up by doing it yourself!

The macro you created earlier that added and formatted headers on a single worksheet has impressed your co-workers so much that they would now like to repeat those same steps but on multiple worksheets in a workbook. You decide to record the actions of adding the headers and formatting the data, but in order to repeat those actions for each worksheet in the workbook you decide to create a loop in Visual Basic.

1. Download and open the file: **CleaningUpData.xlsx** provided in this lecture
2. Note the worksheets (**East, West, North, and South Records**)
  - a. The **RECORDIG** worksheet will be used to record the two initial macros
3. Each worksheets contains similar data but specific to the division

### RECORD THE ADDHEADERS MACRO

1. Activate the **RECORDING** worksheet
2. Go to **VIEW – MACROS – RECORD MACRO**
3. Name the Macro **ADDHEADERS**, leave the other settings as default
4. Record yourself adding the headers to the list
  - a. **DIVISION, CATEGORY, JAN, FEB, MAR, TOTAL EXPENSE**
5. Go to **VIEW – MACROS – STOP RECORDING**

### RECORD THE FORMATDATA MACRO

1. Activate the **RECORDING** worksheet
2. Go to **VIEW – MACROS – RECORD MACRO**
3. Name the Macro **FORMATLIST**, leave the other settings as default
4. Record yourself doing the following;
  - a. Formatting the headers in row 1 (**ex. Bold, background color, font color, borders**)
  - b. Format the Jan-Mar numeric data as currency
    - i. Click in cell **C2**



- ii. Press **CTRL+SHIFT+DOWN ARROW**
  - iii. Press **CTRL+SHIFT+RIGHT ARROW**
  - iv. Format the data as Currency
5. Go to **VIEW – MACROS – STOP RECORDING**

### CREATE THE LOOP

1. Open the Visual Basic Editor window (**ALT+F11**)
2. Note the 2 macros you recorded in the previous steps
3. Insert a new Procedure
  - a. **INSERT – PROCEDURE**
  - b. Name the Procedure **WorksheetLoop**
  - c. Leave the other settings as default
  - d. Add the following code to the procedure

```
For i = 1 To Worksheets.Count
    Worksheets(i).Select

    AddHeaders
    FormatList

Next i
```

### RUN THE PROCEDURE

1. Close the VBE Window
2. Delete the **RECORDINGS** worksheet
3. Go to **VIEW – MACROS – VIEW MACROS**
4. Select the Macro **WORKSHEET LOOP**
5. Click **RUN**

## 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/>



## Master Microsoft Excel Macros and VBA

### Project #4 – Using VBA to Automate Excel Formulas

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 #4 - Overview.....	3
Finding and Building Dynamic Content with VBA.....	3
Level Up by doing it yourself! .....	3
Instructor BIO: .....	5

## Project #4 - Overview

During this exercise you will create a custom VBA procedure that will find a range of cells dynamically in order to SUM that range of cells. The problem is that each sheet that you want to SUM the range of cells on can vary in the number of records to SUM. To fix this problem you will need to programmatically find the last cell in the Total Expense column so that you know where the range ends and where the calculation should go.

You will be creating 2 custom VBA procedures to automate the SUM function and loop the automation over several worksheets.

1. **AutoSum**: This procedure will dynamically find the range of cells to sum and build the calculation
2. **LoopSum**: This procedure will loop the AutoSum procedure through all the worksheets in the workbook

## Finding and Building Dynamic Content with VBA

Below are a few techniques that you will be using in the **AutoSum** procedure.

### **Selection.End(xlDown).Select**

Because you don't know how many records each worksheet has you need a way to find the last cell in the list. The line of code above will travel from an active cell, such as F2, traveling DOWN until it finds the last cell that contains a value. This will assist you in finding that last cell in the range to SUM.

### **ActiveCell.Address**

Once the last cell in the range has been located you then need to get the Address of that cell to include in the formula that will SUM the range. The line of code above will return the address of the active cell (ex. F15). This can be stored in a variable that can then be used at a later time to build the formula.

+

There will be many times in your Excel VBA procedures where you will need to combine multiple String/Text values together. In Excel VBA you can use the Plus Symbol (+) to combine multiple values together.

**=SUM(F2:" + lastCell + ")**



### Level Up by doing it yourself!

Your co-workers are excited about all the time you have saved them with the previous macros that they now would like you add a macro that loops through all the division worksheets and SUMs up the Total Expense column. Because this will need to loop through multiple worksheets you decide to write your own custom VBA procedure to accomplish this.

1. Download and open the file: **AutomateSumFunction.xlsx** provided in this lecture
2. Note the 4 worksheets in the workbook
  - a. Each worksheet contains a Total Expense column
  - b. 2 worksheets have 15 records the other 2 contain 16 records
3. Open the Visual Basic Editor (**ALT+F11**)
4. Add a new **MODULE**

- a. **INSERT – MODULE**
5. Add a New **PROCEDURE**
  - a. **INSERT – PROCEDURE**
  - b. Name the procedure **AutoSum**
  - c. Leave the other settings as default
6. Add the following code to the new procedure

```
Dim lastCell As String

Range("F2").Select

Selection.End(xlDown).Select
lastCell = ActiveCell.Address

ActiveCell.Offset(1, 0).Select
ActiveCell.Value = "=sum(F2:" + lastCell + ")"
```

Now you will create a 2nd procedure that will loop the AutoSum procedure through each worksheet in the workbook

1. Insert a new **PROCEDURE**
  - a. **INSERT – PROCEDURE**
  - b. Name the procedure **LoopSum**
  - c. Leave the other settings as default
2. Add the following code to the new procedure

```
Dim i As Integer

For i = 1 To Worksheets.Count
    Worksheets(i).Select
    AutoSum
Next i
```

3. Close the **VBE** window
4. Run the **LoopSum** procedure
  - a. Go to **VIEW – MACROS – VIEW MACROS**
  - b. Select the **LoopSum** Macro
  - c. Click **RUN**

## 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/>



## Master Microsoft Excel Macros and VBA

### Project #5 – Creating the Final Report

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.....	3
Prompting the User for Input Using an InputBox.....	<b>Error! Bookmark not defined.</b>
Level Up by doing it yourself! .....	3
Instructor BIO: .....	5

## Project #5 - Overview

Project #5 is going to take you through combining all the techniques that you have learned through the previous projects in order to create a final report based on content from multiple worksheets.

## Selecting, Copying and Pasting with VBA

Below are a few techniques that you will use in order to create the final report.

### Selection.Copy

Once a selection has been made, such as **Selection.CurrentRegion.Select**, the line of code above will copy the selection.

### ActiveSheet.Paste

As long as something is still marked as being copied the line of code above will paste the copied content into the active cell of the active worksheet.



### Level Up by doing it yourself!

You decide to

1. Download and open the file: **QuarterlyReport.xlsm** provided in this lecture
2. Click the **ENABLE** button once the document has been opened (**This document contains VBA code and must be enabled before you can use the code as part of this exercise**)
3. Note the 5 worksheets in the workbook (**EAST, WEST, NORTH, SOUTH RECORDS and YEARLY REPORT**)
4. The YEARLY REPORT worksheet is where you will be pasting the data from the other worksheets to create the final reportOpen the VBE (**ALT+F11**)
5. Review the 3 procedures found in Module1
  - a. **AutoSum()**
  - b. **AddHeaders()**
  - c. **FormatData()**
6. These are 3 Macros/Procedures that you have created in earlier projects in this course
7. Create a new Procedure

- a. **INSERT – PROCEDURE**
- b. Name the new Procedure **FinalReport**
- c. Leave the rest of the setting as default
- d. Add the following code to the **FinalReport** procedure

```
Dim i As Integer

For i = 1 To Worksheets.Count - 1
    Worksheets(i).Select

    AutoSum
    AddHeaders
    FormatData

    Range("A2").Select
    Selection.CurrentRegion.Select
    Selection.Copy

    Worksheets("Yearly Report").Select

    Range("A30000").Select
    Selection.End(xlUp).Select

    ActiveCell.Offset(2, 0).Select

    ActiveSheet.Paste

Next i
```

8. Close the **VBA** Window
9. Run the **FinalReport** Procedure
  - a. Go to **VIEW – MACROS – VIEW MACROS**
  - b. Select the **FinalReport** Macro
  - c. Click **RUN**

## 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/>