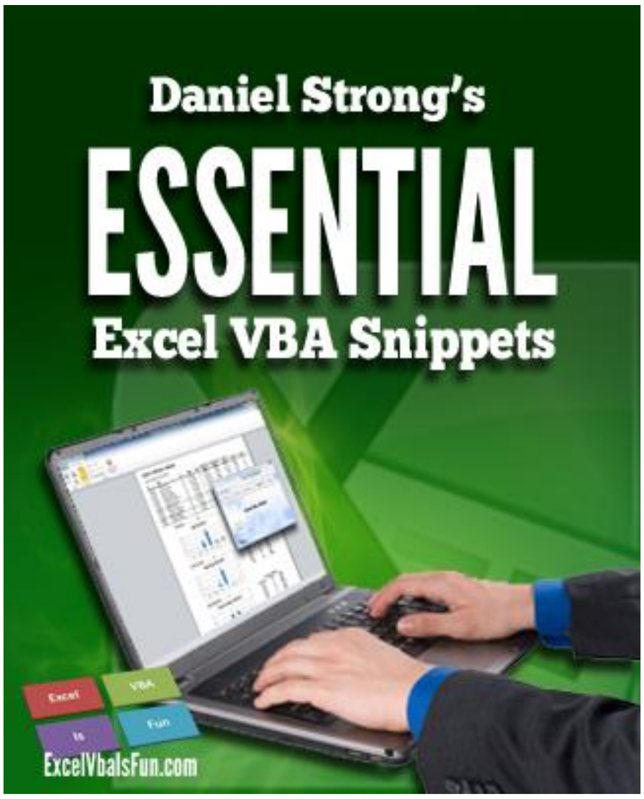

Daniel Strong's
ESSENTIAL
Excel VBA Snippets
Excel
VBA
Is
Fun
ExcelVbaIsFun.com

Acknowledgements

Thank you Mike Gervin of ExcelIsFun (www.youtube.com/excelisfun) for your wonderful support and for inspiring so many people with your lessons. Big thanks to mom and dad and my wife Brittney for putting up with me! Thanks to God for also putting up with me. Thanks to you the reader for picking up this book and my students worldwide for your support and desire to continue learning and improving! For more free and premium training, visit https://ExcelVbaIsFun.com

Preface

VBA is the window into Excel's soul - unveiling the Wizard behind the curtain. Okay, it's actually some simple code that allows you to **Automate almost anything in Excel.**

You've heard of recording macros - NOW learn to make them from scratch and make them more dynamic, more user-friendly and much much FASTER and efficient. If you use Excel (and you understand some basics), you're ready to dive into some simple codes to make Excel work for you - anyway you want.

This book assumes you know some Excel VBA Basics, at least enough to know how to make a new Sub that you can shove code into. If not, please click here to find out how this works: http://www.youtube.com/watch?v=AIhKNNXzZLM&list=PLw8O1w0Hv2ztGjIkrW7suD6oNDaOk3vbR&index=1 This book is interactive – all sample materials are available in the Sample Workbook, so you don't have to start from scratch and can benefit from real life examples.

Table of Contents

Chapter 1: Beginner Snippets

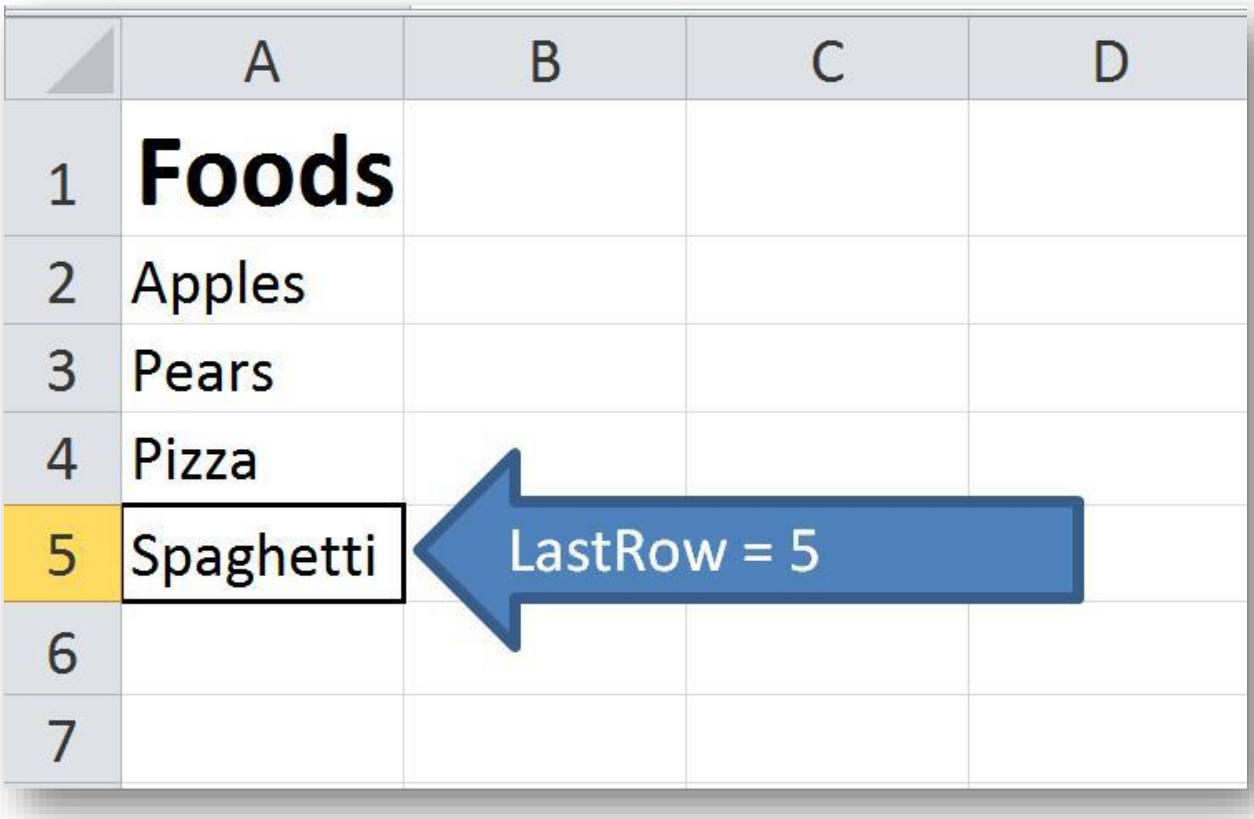
How do you eat an elephant? Simple, one bite at a time. We all have to start somewhere, and these code snippets will help you out if you're new. And if you're not new, you may still want to copy and paste them from this book.

Either way, enjoy the code snippets and remember – You CAN do anything you set your mind to. You ARE a success, just believe it and think like that no matter what. Remember, the mind controls the body, so create your reality through positive thoughts. Enough about that, let's get coding!

1. Finding the Last Row

This one is super-duper essential! Example: To find the last row in use on column one:

```
LastRow = ThisWorkbook.Sheets("Sheet1").Cells(Rows.Count,1).End(xlUp).Row
```



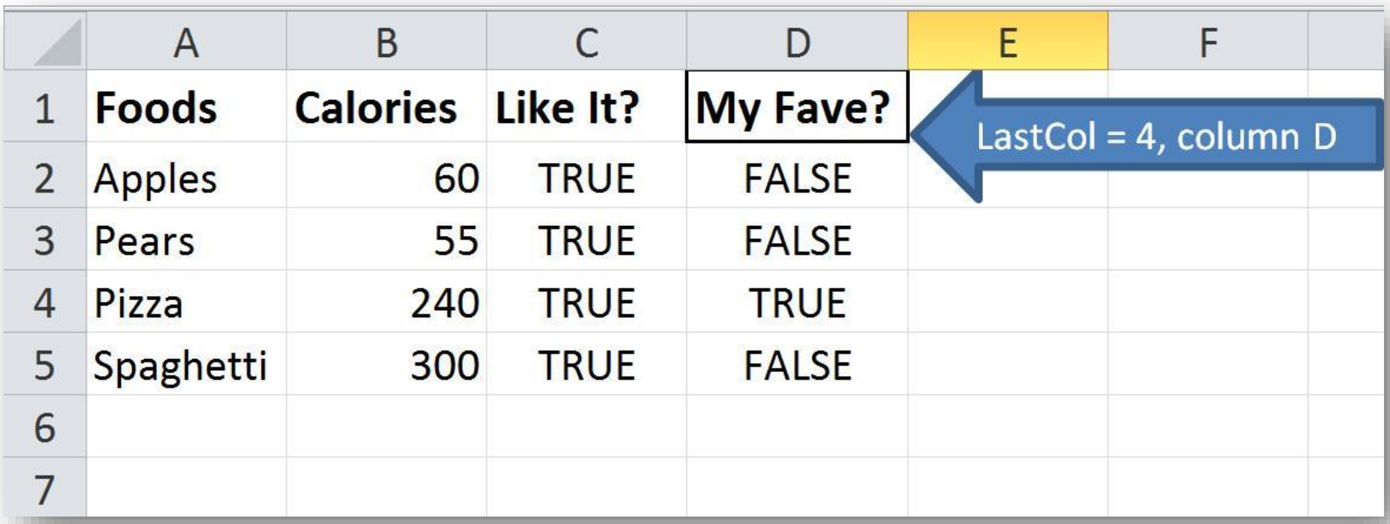
	A	B	C	D
1	**Foods**			
2	Apples			
3	Pears			
4	Pizza			
5	Spaghetti			
6				
7				

For more info, watch Video1 or [click here to view online](#).:

2. *Finding the Last Column*

If you need to know what the last used column number is, or how many you've got, use this code snippet! This is if you want the last used column on Row 1.

```
LastCol = ThisWorkbook.Sheets("Sheet1").Cells(1,Columns.Count).End(xlToLeft).Column
```



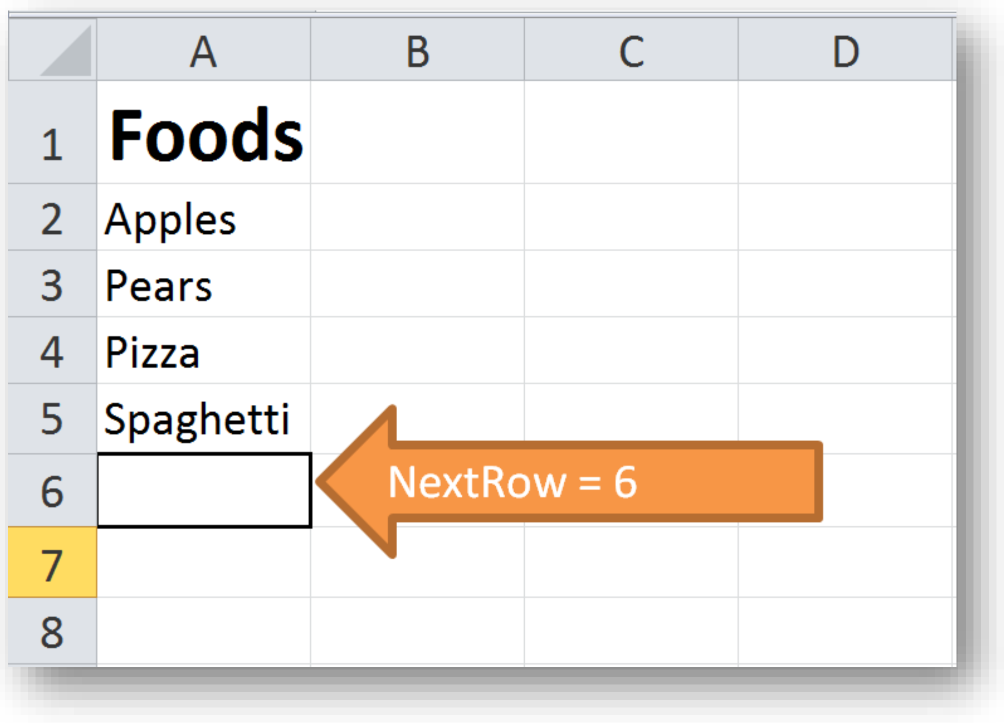
	A	B	C	D	E	F
1	**Foods**	**Calories**	**Like It?**	**My Fave?**		
2	Apples	60	TRUE	FALSE		
3	Pears	55	TRUE	FALSE		
4	Pizza	240	TRUE	TRUE		
5	Spaghetti	300	TRUE	FALSE		
6						
7						

For more info, watch Video1 or click here to view online.
https://www.youtube.com/watch?v=K0VqyXLJBOw&list=PLw8O1w0Hv2ztGjIkrW7suD6oNDaOk3vbR

3. *Finding the Next Row*

When you're wanting to write to the next available Row, this code snippet is a life saver! Example: To find the next available row in column one:

```
NextRow = ThisWorkbook.Sheets("Sheet1").Cells(Rows.Count,1).End(xlUp).Row + 1
```



	A	B	C	D
1	**Foods**			
2	Apples			
3	Pears			
4	Pizza			
5	Spaghetti			
6				
7				
8				

For more info, watch Video2 or [click here to view online](https://www.youtube.com/watch?v=egYkGw6kuUI&list=PLw8O1w0Hv2ztGjIkrW7suD6oNDaOk3vbR&index=43).
https://www.youtube.com/watch?v=egYkGw6kuUI&list=PLw8O1w0Hv2ztGjIkrW7suD6oNDaOk3vbR&index=43

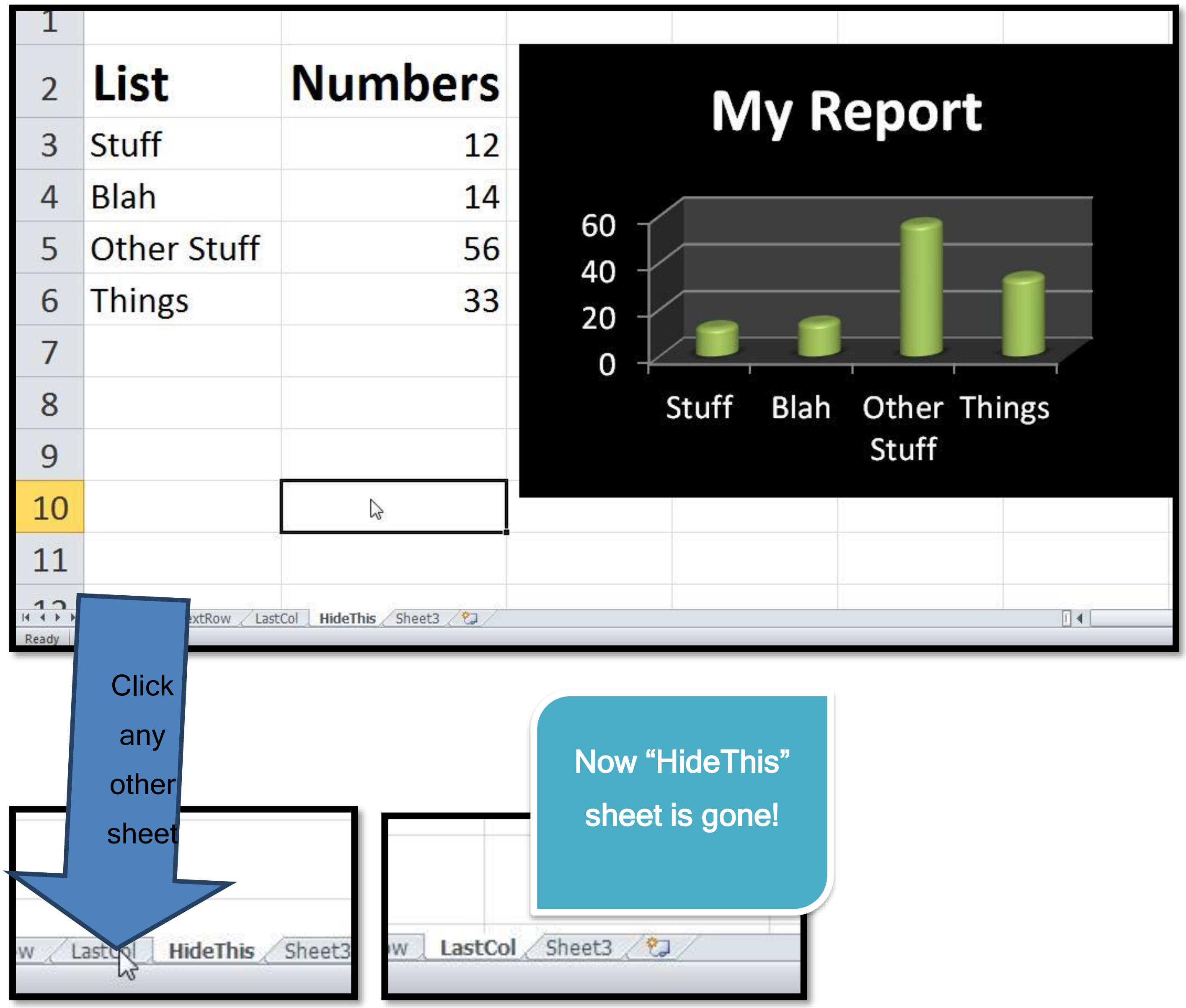
4. Auto-Hide Current Sheet when Selecting Another

Especially Cool when you have a zillion sheets (like reports) that you only need unhidden when you're using them.

Right-click on the sheet you want and click on "View Code". Now copy/paste this code in there (Assuming the sheet you want to hide is called **HideThis,** otherwise use your sheet name):

```
Private Sub Worksheet_Deactivate()
        ThisWorkbook.Sheets("HideThis").Visible = False
End Sub
```

When you click away from that sheet, it goes invisible!



List	Numbers
Stuff	12
Blah	14
Other Stuff	56
Things	33

For more info, watch Video4 or click here to view online.
https://www.youtube.com/watch?v=z8BHx86HSN8

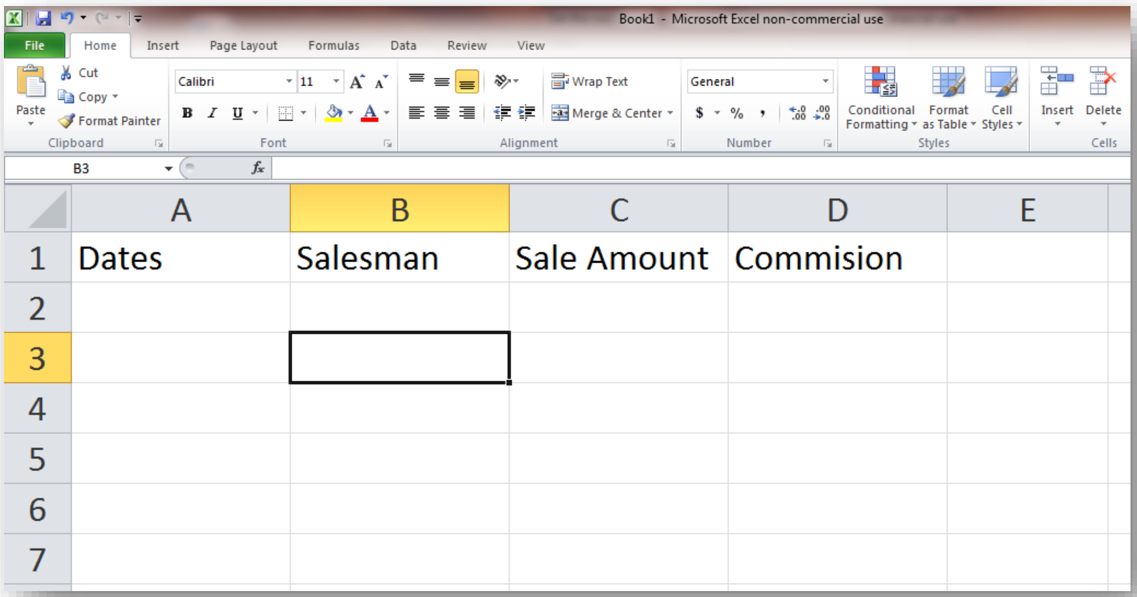
5. *Set the worksheet as a Variable (abbreviate it)*

This will save a gazillion keystrokes of the lifetime of a programmer. Instead of typing this to change cell A1 to “Dates”, and then change b1 to “Salesman”, and c1 to “Sale Amount”:

```
ThisWorkbook.Sheets(“Sheet1”).Range(“A1”) = “Dates”
ThisWorkbook.Sheets(“Sheet1”).Range(“B1”) = “Salesman”
ThisWorkbook.Sheets(“Sheet1”).Range(“C1”) = “Sale Amount”
ThisWorkbook.Sheets(“Sheet1”).Range(“D1”) = “Commission”
```

Try this instead:

```
Dim ws as worksheet
Set ws = ThisWorkbook.Sheets(“Sheet1”)

ws.Range(“A1”) = “Dates”
ws.Range(“B1”) = “Salesman”
ws.Range(“C1”) = “Sale Amount”
ws.Range(“D1”) = “Commission”
```

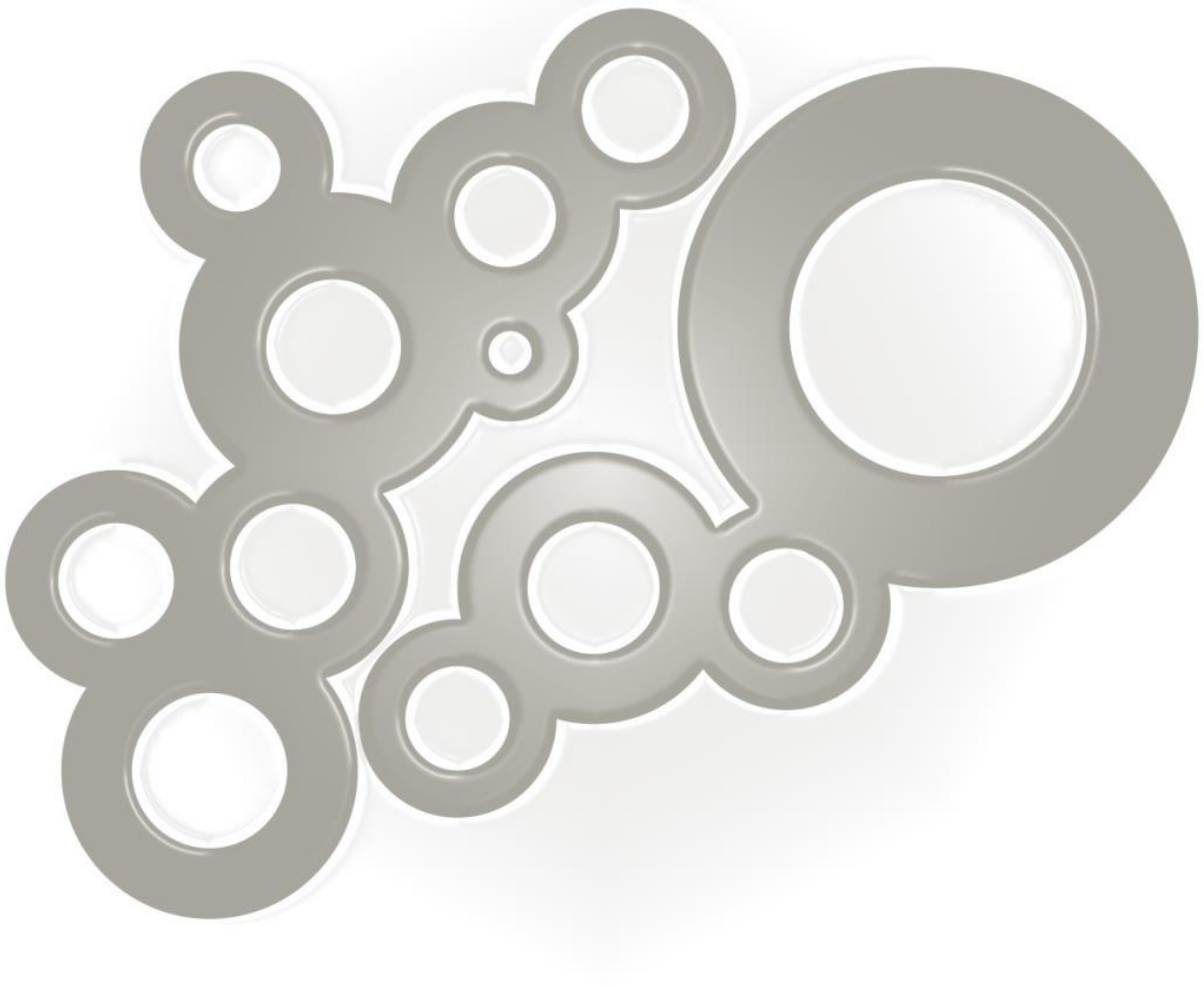


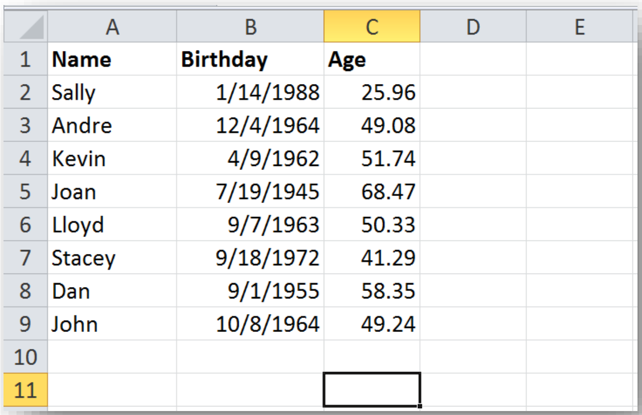
	A	B	C	D	E
1	Dates	Salesman	Sale Amount	Commision	
2					
3					
4					
5					
6					
7					

For more info, watch Video4 or [click here to view online](http://www.youtube.com/watch?v=S55cKcgF1jk).
http://www.youtube.com/watch?v=S55cKcgF1jk

Chapter 2: Loops

Simply put – loops are how to dig information out of loose data. When you search a worksheet or table full of data for specific info, you as a human would scan from the top to the bottom, and you'd search one or more columns, perhaps both before moving on. This chapter has some great code snippets to help you on your journey through Loops!





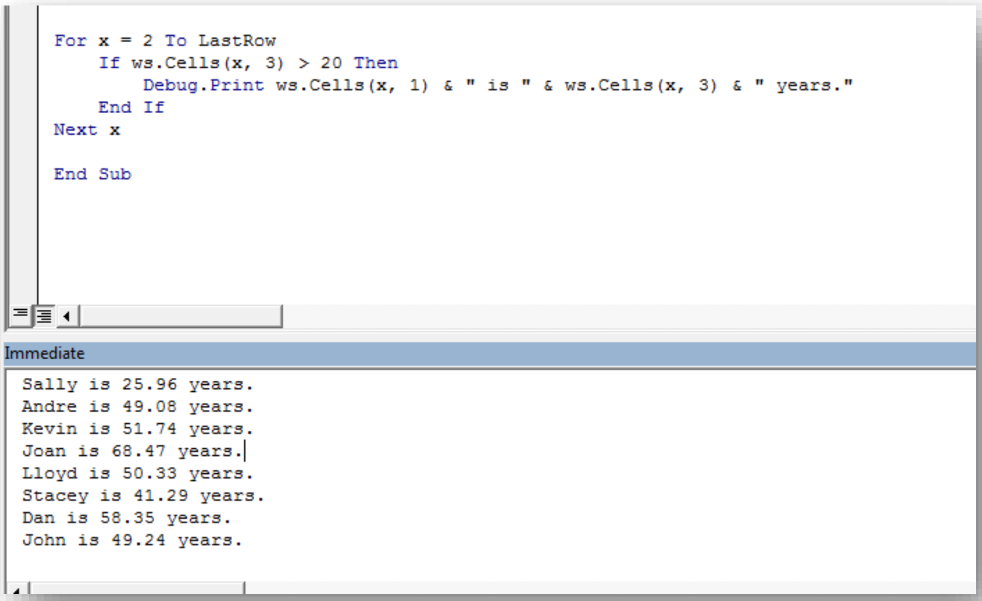
	A	B	C	D	E
1	Name	Birthday	Age		
2	Sally	1/14/1988	25.96		
3	Andre	12/4/1964	49.08		
4	Kevin	4/9/1962	51.74		
5	Joan	7/19/1945	68.47		
6	Lloyd	9/7/1963	50.33		
7	Stacey	9/18/1972	41.29		
8	Dan	9/1/1955	58.35		
9	John	10/8/1964	49.24		
10					
11					

6. *Loop Through Your Database*

So, you want to have excel review each row in your database/table starting with row 2 (because we don't want to analyse the header row 1) all the way to the end of the data. Let's say we want to see anyone who's older than 20 years. We'll loop through our worksheet using the code below. See snippet #1 **Finding the Last Row** for more info on the first part of the procedure.

```
Dim ws As Worksheet
Set ws = ThisWorkbook.Sheets("Sheet3")
LastRow = ws.Cells(Rows.Count, 1).End(xlUp).Row

For x = 2 To LastRow
    If ws.Cells(x, 3) > 20 Then
        Debug.Print ws.Cells(x, 1) & " is " & ws.Cells(x, 3) & " years."
    End If
Next x
```

This code loops through row 2 through 9 and finds those that are greater than 20! If they are, it spits out something like: "John is 56.77 years"



```
For x = 2 To LastRow
    If ws.Cells(x, 3) > 20 Then
        Debug.Print ws.Cells(x, 1) & " is " & ws.Cells(x, 3) & " years."
    End If
Next x

End Sub
```

Immediate

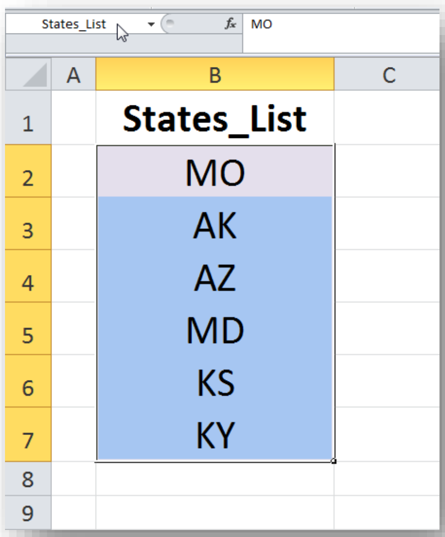
```
Sally is 25.96 years.
Andre is 49.08 years.
Kevin is 51.74 years.
Joan is 68.47 years.
Lloyd is 50.33 years.
Stacey is 41.29 years.
Dan is 58.35 years.
John is 49.24 years.
```

For more info, watch Video4 or click here to view online.

http://www.youtube.com/watch?v=JEVh3hhEiRA

7. *Loop Through Every Cell in a Named Range*

This is handy when you have a named range in your workbook that you want to refer to. This will go through each Cell in the range, left to right, top to bottom. In the example below, we have a custom list of States that we'd like to loop through, named "States_List", found on sheet "LNR".



	A	B	C
1		**States_List**	
2		MO	
3		AK	
4		AZ	
5		MD	
6		KS	
7		KY	
8			
9			

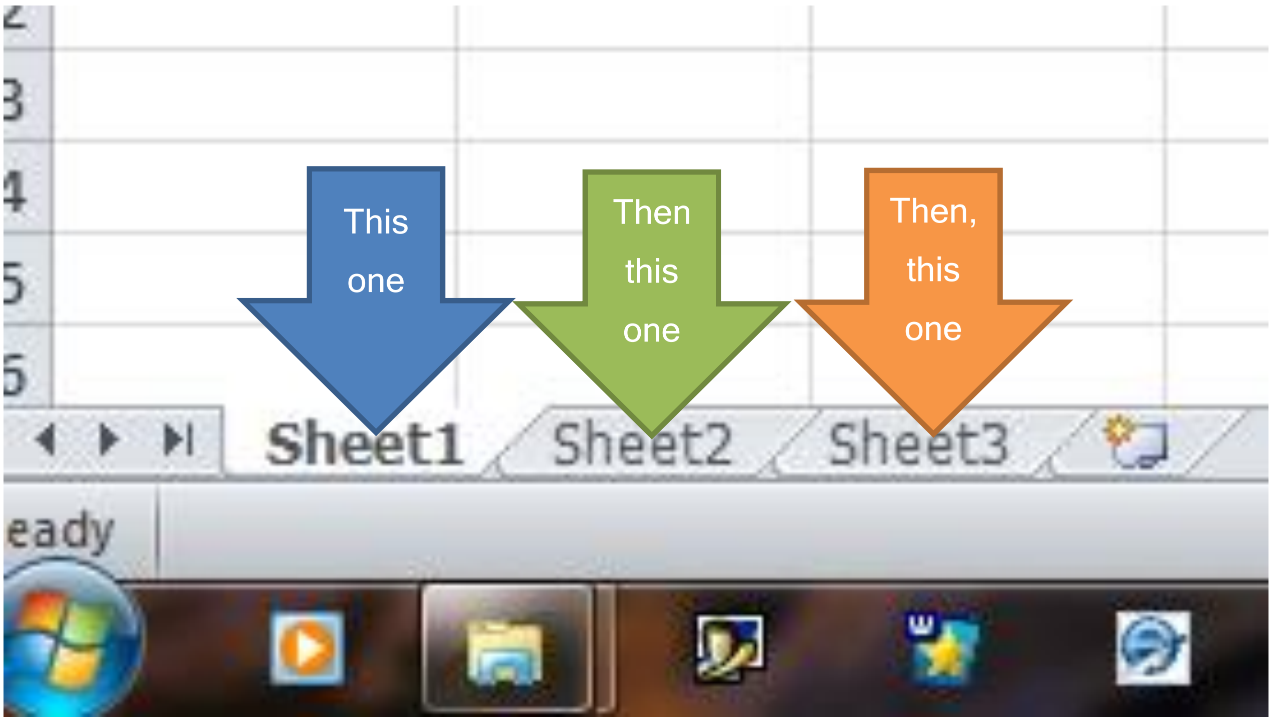
To loop through a named range, try this code:

```
For Each cell In ThisWorkbook.Sheets("LNR").Range("States_List")
   'do something
Next cell
```

For more info, watch Video___ or click here to view online.
http://www.youtube.com/watch?v=629dqqN3gCc

8. *Loop Through All Worksheets*

If you want to cycle through each worksheet to see if one of them has this property or this one's visible or not, or see which one starts with this letter – you get the picture. This code snippet allows you to loop through each worksheet in your workbook. Nifty!!



In this example, we want the names of each sheet printed in the Immediate Window. Use this (it's a dandy):

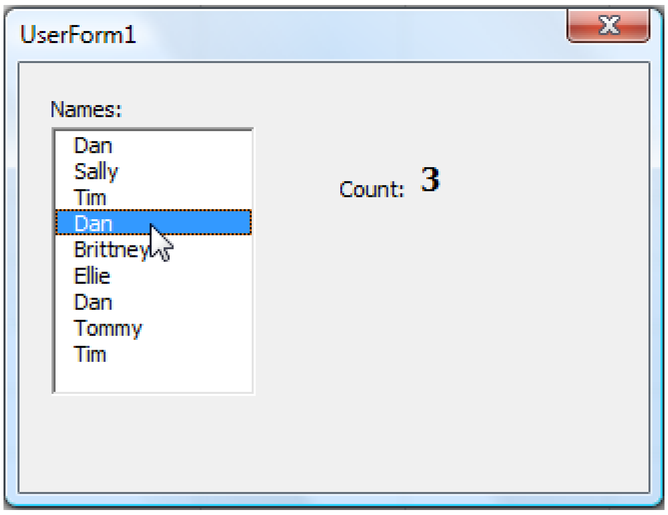
```
Dim ws As Worksheet

For Each ws In ThisWorkbook.Sheets
        'list the names
        Debug.Print ws.Name
Next ws
```

For more info, watch Video___ or click here to view online.
http://www.youtube.com/watch?v=-75HKRrIMI8

9. *Loop Through ListBox Items*

Sometimes you need to go through each item in a ListBox, maybe to refresh or tweak the list. Maybe you're filtering to remove an entry or two. In this example, we'll count the occurrences of the name "Dan" in our list. We'll put a click event in our ListBox (lbxNames) to trigger when clicked. Whichever name we click on will be counted.

```
Private Sub lbxNames_Click()
SelName = Me.lbxNames.Value 'grab the current selection

For x = 0 To Me.lbxNames.ListCount - 1
    If Me.lbxNames.List(x, 0) = Me.lbxNames.Value Then
        myCounter = myCounter + 1 'increment each time it matches
    End If
Next x
'here at the end, fill the label with the number
Me.lblCount = myCounter
End Sub
```



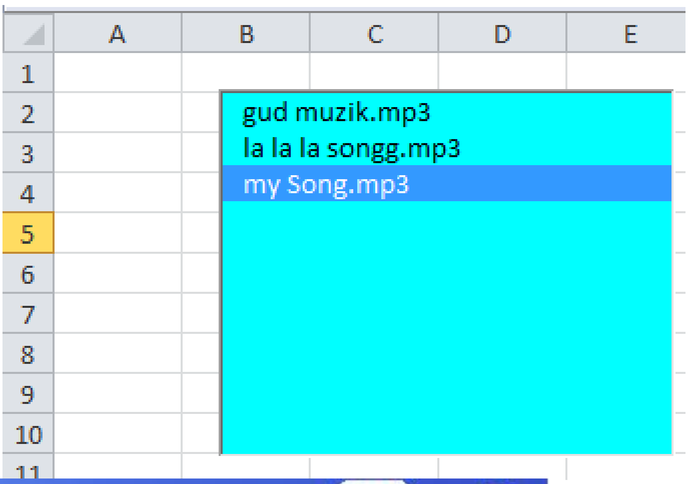
10. Loop Through Files in a Folder

When you're dealing with files and folders with your macros, it's nice to have this in your back pocket, in case you need to have VBA search through each file name and analyze them. Maybe you're searching for files that end in .mp3, or maybe you're needing to find a specific file, maybe you want any file that contains a term or string of text you need. Who knows, play around with it!

In this example, we're going to fill a listbox with anything on the Desktop that ends in ".mp3". Let's see how many songs are on the Desktop! Try this macro within the Sheet that your Listbox lives in.

```
Sub FindMp3s()
Dim FSO As Object
Dim File As Object
Dim Fldr As Object
Set FSO = CreateObject("scripting.FileSystemObject")

Set Fldr = FSO.getfolder("C:\Users\KenLaptop\Desktop")
For Each File In Fldr.Files
    If Right(File.Name, 4) = ".mp3" Then
        'add to listbox
        Me.ListBox1.AddItem File.Name
    End If
Next File
End Sub
```

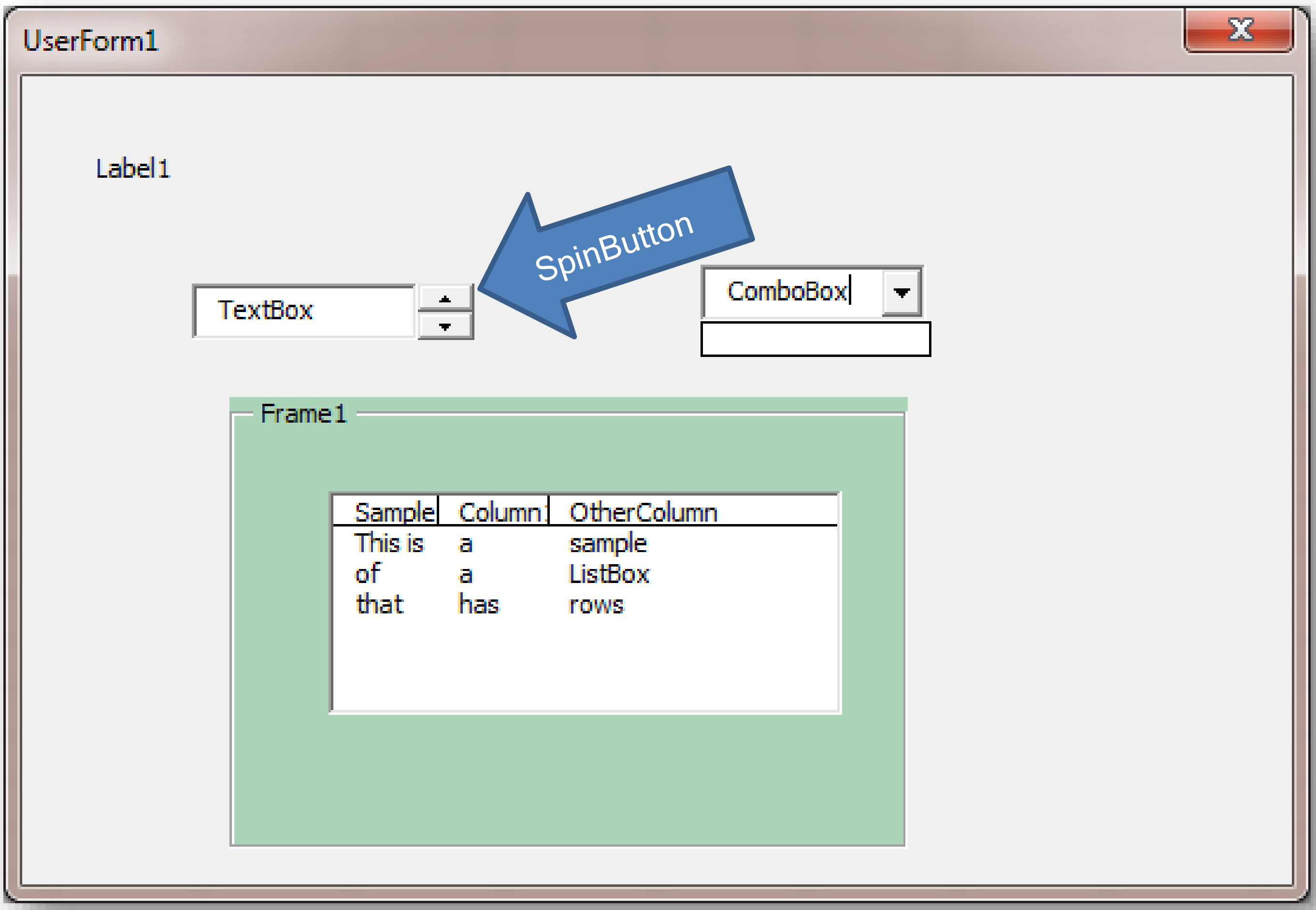
Here's the listbox after the macro is run:


gud muzik.mp3
la la la songg.mp3
my Song.mp3

Here's what my Desktop looks like below;
As you can see it found the 3 mp3 files!

Chapter 3: Controls

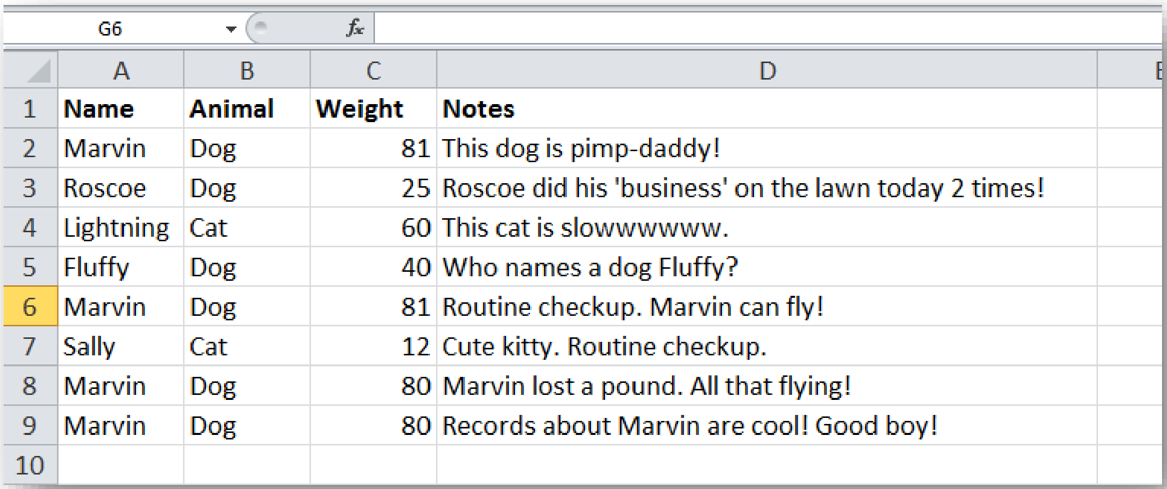
Using controls in Userforms and Worksheets is another essential skill for the Excel Programmer. Controls make navigation and triggering macros easier and just give the user easier access to data or manipulation of that data! Controls include (but are not limited to) ComboBoxes, ListBoxes, TextBoxes, CheckBoxes, Option Buttons, Frames, Labels, and SpinButtons. Check out these tasty little treats from the oven!



Sample	Column1	OtherColumn
This is	a	sample
of	a	ListBox
that	has	rows

11. Fill a ListBox From a Table/Database

Listboxes are amazing, because you can emulate tables with them and make it look like a portion of a worksheet, allowing the viewer to select one or more records and do stuff with them! You can Fill a ListBox with a named range or with a regular range by using the RowSource (on a Userform) or the ListFillRange (on a Sheet). But here's how to customize the results in your ListBox. In this example, we fill a ListBox with the records that say "Marvin" in the name column.

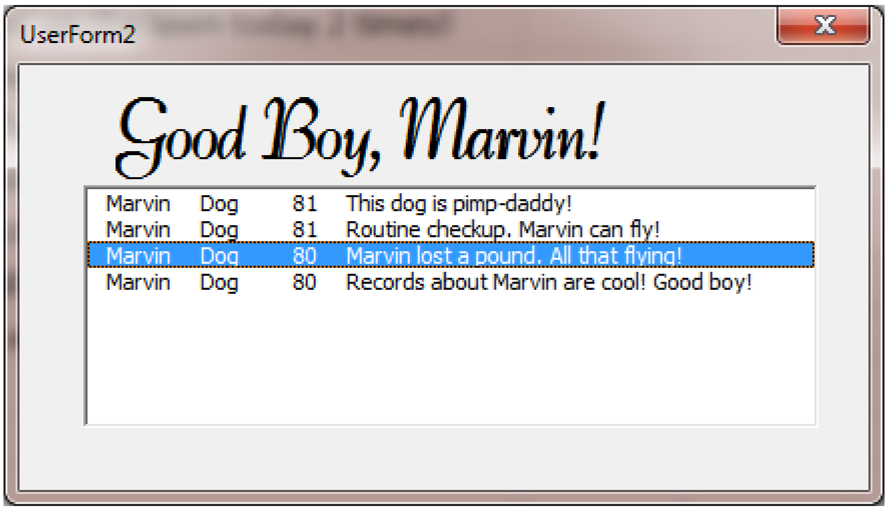


G6

	A	B	C	D
1	**Name**	**Animal**	**Weight**	**Notes**
2	Marvin	Dog	81	This dog is pimp-daddy!
3	Roscoe	Dog	25	Roscoe did his 'business' on the lawn today 2 times!
4	Lightning	Cat	60	This cat is slowwwwww.
5	Fluffy	Dog	40	Who names a dog Fluffy?
6	Marvin	Dog	81	Routine checkup. Marvin can fly!
7	Sally	Cat	12	Cute kitty. Routine checkup.
8	Marvin	Dog	80	Marvin lost a pound. All that flying!
9	Marvin	Dog	80	Records about Marvin are cool! Good boy!
10				

This code is directed at the ListBox within the my Userform:

```
Dim ws As Worksheet
Set ws = ThisWorkbook.Sheets("MrvnSht")
wLR = ws.Cells(Rows.Count, 1).End(xlUp).Row

For x = 2 To wLR
   If ws.Cells(x, 1) = "Marvin" Then
      'add to lbox
      Me.lbxMarvin.AddItem ws.Cells(x, 1)
      Me.lbxMarvin.List(Me.lbxMarvin.ListCount - 1, 1) = ws.Cells(x, 2)
      Me.lbxMarvin.List(Me.lbxMarvin.ListCount - 1, 2) = ws.Cells(x, 3)
      Me.lbxMarvin.List(Me.lbxMarvin.ListCount - 1, 3) = ws.Cells(x, 4)
   End If
Next x
```



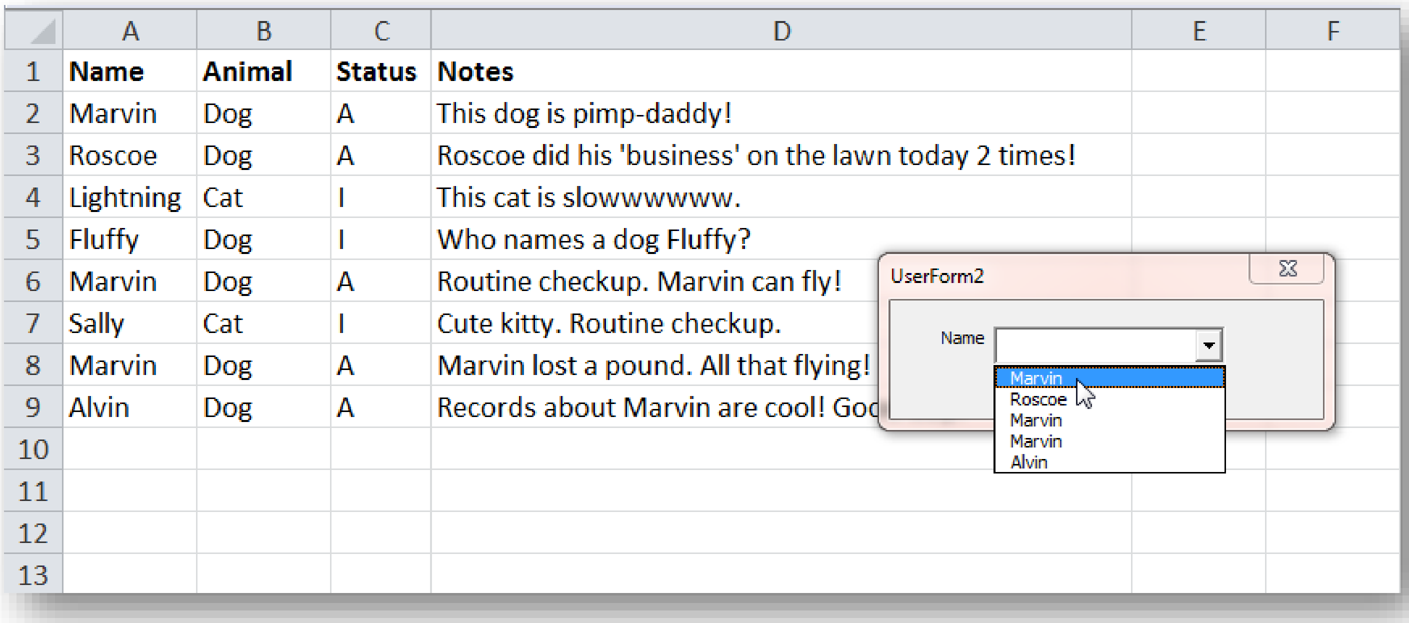
```
Marvin   Dog   81   This dog is pimp-daddy!
Marvin   Dog   81   Routine checkup. Marvin can fly!
Marvin   Dog   80   Marvin lost a pound. All that flying!
Marvin   Dog   80   Records about Marvin are cool! Good boy!
```

12. Fill a ComboBox

Comboboxes are called drop-down menus/boxes in the non-VBA vernacular. They are actually a combination of TextBox and ListBox, because although you may select one of the items as a regular dropdown or list, you may also type freely and it will help you auto-select the item you want, but you can also veer from the list and enter something else.

Here's a code snippet where we fill a combobox with entries that are "Active" (records that have the letter "A" in the Status column).

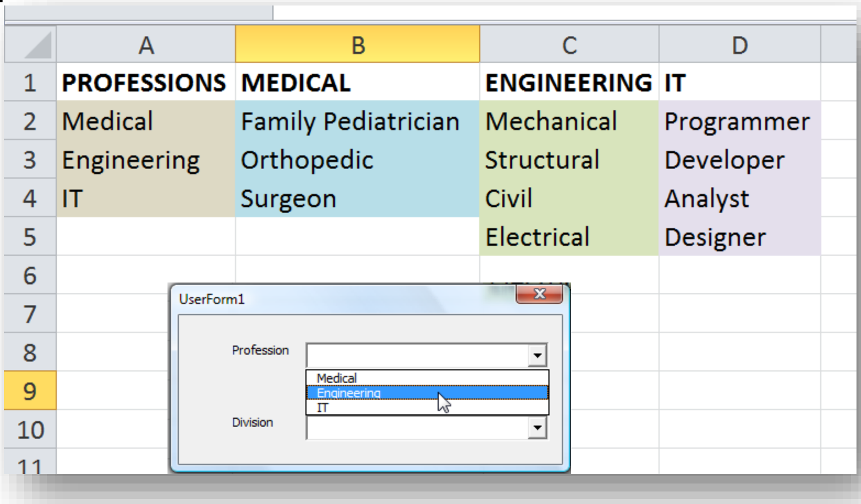
```
Dim ws As Worksheet
Set ws = ThisWorkbook.Sheets("blah")
wLR = ws.Cells(Rows.Count, 1).End(xlUp).Row

For x = 2 To wLR
    If ws.Cells(x, 3) = "A" Then
        'add to lbox
        Me.cmbStatus.AddItem ws.Cells(x, 1)
    End If
Next x
```



	A	B	C	D	E	F
1	**Name**	**Animal**	**Status**	**Notes**		
2	Marvin	Dog	A	This dog is pimp-daddy!		
3	Roscoe	Dog	A	Roscoe did his 'business' on the lawn today 2 times!		
4	Lightning	Cat	I	This cat is slowwwwwww.		
5	Fluffy	Dog	I	Who names a dog Fluffy?		
6	Marvin	Dog	A	Routine checkup. Marvin can fly!		
7	Sally	Cat	I	Cute kitty. Routine checkup.		
8	Marvin	Dog	A	Marvin lost a pound. All that flying!		
9	Alvin	Dog	A	Records about Marvin are cool! Goo		
10						
11						
12						
13						

13. *Fill ComboBox Based on Another ComboBox*

Say you have a UserForm and 2 ComboBoxes. The First one will have the 3 main categories, Medical, Engineering and IT. Based on which is selected the other 2 comboboxes will fill with those lists - one a list of Medical types, the next Engineering types and so on. We've named the ranges as follows: "Professions", "Medical", "Engineering", and "IT". We'll use the **RowSource** property in the userform combobox to set the lists this time, rather than using .AddItem like last time.
We'll put the code in the change event for the first combobox named "Combobox1", this way anytime someone clicks the first combobox, the next one will be changed to include those specific entries for that selection.

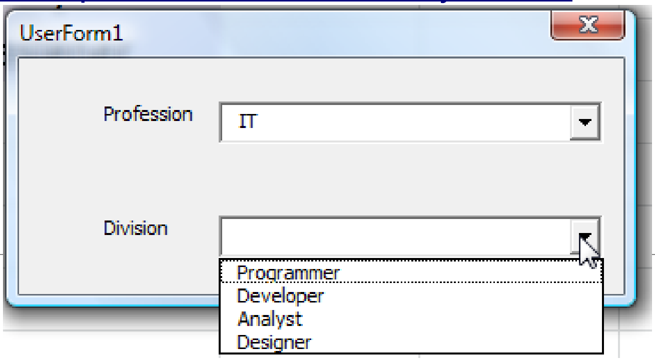


	A	B	C	D
1	PROFESSIONS	MEDICAL	ENGINEERING	IT
2	Medical	Family Pediatrician	Mechanical	Programmer
3	Engineering	Orthopedic	Structural	Developer
4	IT	Surgeon	Civil	Analyst
5			Electrical	Designer

```
Private Sub ComboBox1_Change()
Me.ComboBox2 = ""

Select Case Me.ComboBox1
    Case "Medical"
        Me.ComboBox2.RowSource = "Medical"
    Case "Engineering"
        Me.ComboBox2.RowSource = "Engineering"
    Case "IT"
        Me.ComboBox2.RowSource = "IT"
    Case Else
        'do nothing
End Select
End Sub
```

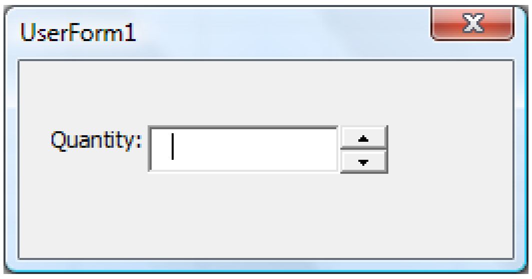
So when "IT" is selected in Combobox1, Combobox2 now is blanked out and filled with the list called "IT".

Video here: http://www.youtube.com/watch?v=qvakbdZaeYU&feature=youtu.be

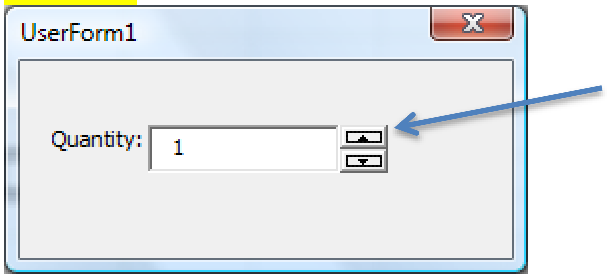


14. *Spinbuttons to Increase/Decrease Userform Textbox Value*

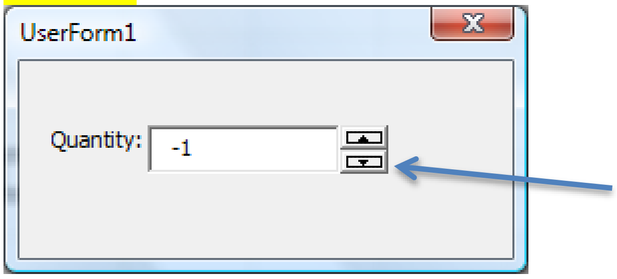
Spin buttons are basically left/right arrows or up/down arrow sets. You can direct what the spin-up click does or the spin-down. Spin up controls the right click if you have it set horizontal (right/left) and spin-down controls the left click if horizontal. This is fun to play with. In the example below, we're going to make the textbox on the userform named "Userform1" go up in value when you spin up and go down 1 when you spin-down. The Textbox is called "tbQty" and the spinbutton is called "sbQty"



```
Private Sub sbQty_SpinUp()
  On Error Resume Next
  If Me.tbQty = "" Then Me.tbQty = 0
  Me.tbQty = Me.tbQty + 1
End Sub
```



```
Private Sub sbQty_SpinDown()
  On Error Resume Next
  If Me.tbQty = "" Then Me.tbQty = 0
  Me.tbQty = Me.tbQty - 1
End Sub
```

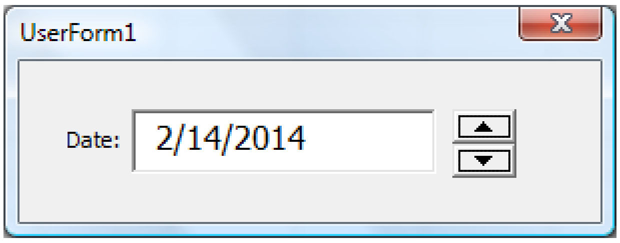


More info [here](http://www.youtube.com/watch?v=bPtWjdE9trE) (a video): http://www.youtube.com/watch?v=bPtWjdE9trE

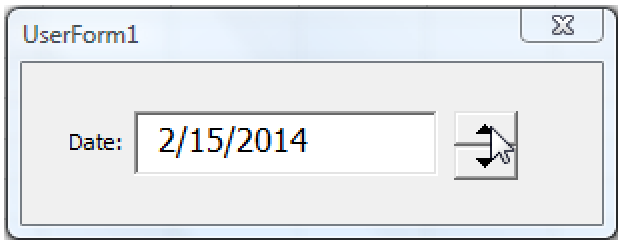
15. Use a Spinbutton to Increase/Decrease (Userform) Textbox Dates

When using a spinbutton to increase/decrease the date, we'll use a similar procedure as increasing/decreasing the value by one. One difference is that we'll be using "IsDate" and "CDate" in order to identify or convert dates (respectively). Keep in mind you can use this code to increase weekly (+/- 7), biweekly (+/-14), and there are ways to get the beginning date of the next/previous month or the end of a month. In this specific example, we'll increase or decrease the date by one day. Let's start with Valentines day (Feb 14) 2014!

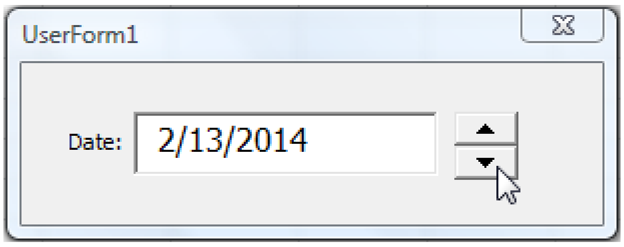
We'll name the textbox "tbDate" and the spinbutton "sbDate". Double click on the spinbutton but change the event to "SpinUp" and "SpinDown" instead of the change event that's default. Here's the code I use for each.



```
Private Sub sbDate_SpinUp()
On Error Resume Next
Me.tbDate = CDate(Me.tbDate) + 1
End Sub

Private Sub sbDate_SpinDown()
On Error Resume Next
Me.tbDate = CDate(Me.tbDate) - 1
End Sub
```



or



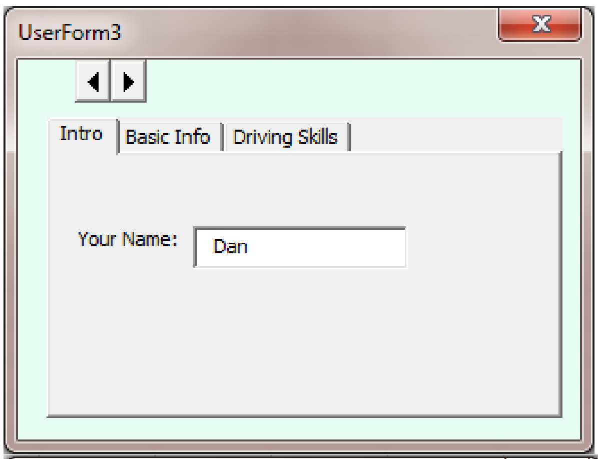
The first line tells it to not freak out if something's wrong, like a textbox being blank or NOT a date, etc. The next line tells it to take whatever's in the textbox and have it become whatever it is (converted from text string to a date using CDate) plus or minus one day.

Video tutorial found Here: http://www.youtube.com/watch?v=6r4w5LDCgkU

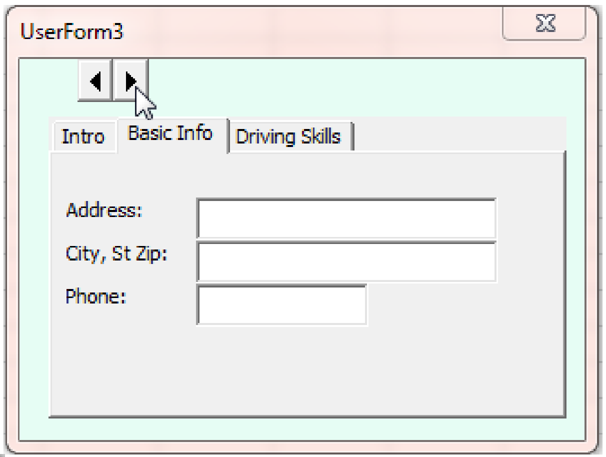
16. *Use a Spinbutton to Cycle Through (Userform) Tabs*

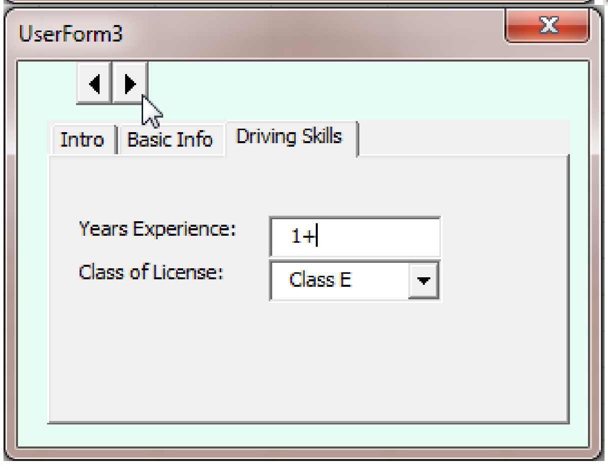
Navigation is another big deal in programming. Here’s a starting point on how to navigate and manipulate the Userform tabs using a spin button. Please note that the Multipage in the example has 3 tabs, so the Multipage values are 0, 1, and 2. Here are the complete macros for the spin-up and spin-down snippets within the userform. The spinbutton is named SpinButton1.

```
Private Sub SpinButton1_SpinDown()
If Me.MultiPage1.Value = 0 Then
   'do nothing
Else
   Me.MultiPage1.Value = Me.MultiPage1.Value - 1
End If
End Sub

Private Sub SpinButton1_SpinUp()
If Me.MultiPage1.Value = 2 Then
   'do nothing
Else
   Me.MultiPage1.Value = Me.MultiPage1.Value + 1
End If
End Sub
```



Your Name: Dan





Years Experience: 1+

Class of License: Class E

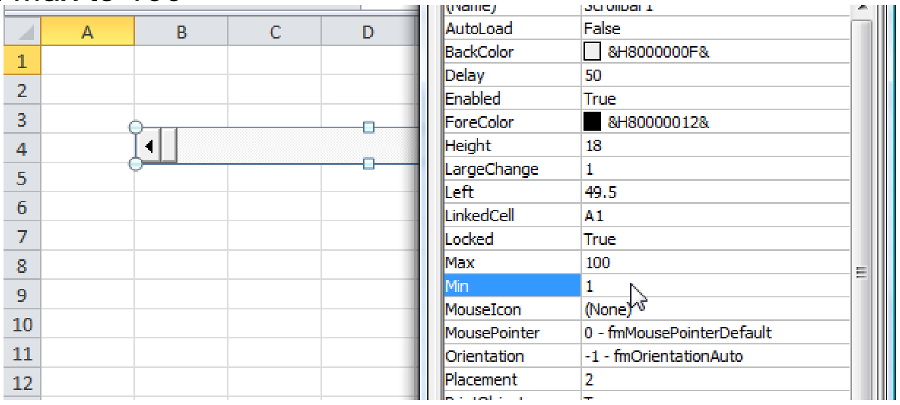
17. *ScrollBar with Worksheet numbers*

Scroll bars can be used pretty similarly as the spin buttons, the main difference being that Scroll bars can be dragged to pinpoint a value in addition to the top/bottom ends. This is fun to play with and manipulate. In this example, we'll pin (link) a cell to receive the value of the scroll bar and we'll set the scroll bar to be from 1 to 100.

Step 1: Link the scroll bar to cell "A1"

- Right click on scroll bar>Properties
- Type "A1" in the LinkedCell field

Step 2: Set min to 1 and Max to 100



AutoLoad	False
BackColor	&H8000000F&
Delay	50
Enabled	True
ForeColor	&H80000012&
Height	18
LargeChange	1
Left	49.5
LinkedCell	A1
Locked	True
Max	100
Min	1
MouseIcon	(None)
MousePointer	0 - fmMousePointerDefault
Orientation	-1 - fmOrientationAuto
Placement	2

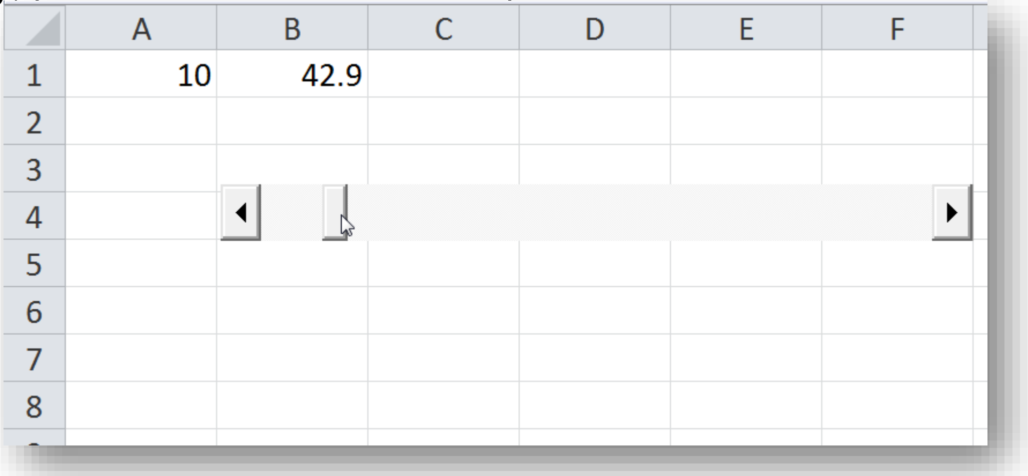
Now when you click up or down OR drag the scrollbar middle piece, the value in cell A1 will reflect it. Try putting a function in cell B1, like **=A1 * 4.29** to see how many weeks are in the number of months in A1.

If you don't want to link to cell A1:

In design mode, double click on the scroll bar and put this code in the Sub, assuming the scroll bar is ScrollBar1. Otherwise adjust this code:

```
Private Sub ScrollBar1_Change()
        Range("a1") = Me.ScrollBar1
        Range("b1") = Range("a1") * 4.29
End Sub
```

Either way, your sheet will look like this as you scroll. . .



	A	B	C	D	E	F
1	10	42.9				

18. Insert Title Here

Got an idea for a Snippet?! Let me know, check out
https://excelvbaisfun.com/contact/

```
Dim ws As Worksheet
Set ws = ThisWorkbook.Sheets("blah")
wLR = ws.Cells(Rows.Count, 1).End(xlUp).Row

For x = 2 To wLR
    If ws.Cells(x, 3) = "A" Then
        'add to lbox
        Me.cmbStatus.AddItem ws.Cells(x, 1)
    End If
```

Chapter 4: Make It Run FASTER!

If you really want your program in Excel to be awesome, it needs to be FAST. I'm going to show you some of my favourite snippets and tips to make your macros Blazing FAST, because nobody wants to click a button and then wait 40 minutes, or even ten seconds! Especially if they click that button often. Let's dive in, shall we?

21. Disable/Enable ScreenUpdating

When a macro is running it has to obey your commands in order. It usually runs these commands much faster than the eye can see, and sometimes faster than the screen can keep up with. In order for the screen to process all the changes, sometimes this slows down the progress of the macro in order to compensate for the slow screen. The solution is to allow the screen to NOT need to keep up with the macro. Disable the screen's updating until the macro has run its course. Now it can go fast and not have to worry about whether the screen is keeping up! Put this at the beginning and ending of your macro:

Beginning:

```
Application.ScreenUpdating = False
```

At the End of your code:

```
Application.ScreenUpdating = True
```

EXAMPLES:

```
Sub mySlowMacro()
For x = 2 to 10000
        Cells(x,1) = x
        Cells(x,2) = x  + (x*.10)
Next x
End Sub
```

```
Sub myFasterMacro()
Application.ScreenUpdating = False
For x = 2 to 10000
        Cells(x,1) = x
        Cells(x,2) = x  + (x*.10)
Next x
Application.ScreenUpdating = True
End Sub
```

22. Disable/Enable Calculation

Sometimes the macro runs slowly because every time a cell changes in Excel, the workbook decides to recalculate EVERY CELL in the workbook so it won't miss anything important it needs to know. . . When we're running a gazillion cell changes for report generation, we don't want it recalculate a gazillion times, just turn calculation mode back on when the macro is over with. Try this:

Beginning:

```
Application.Calculation = False
```

At the End of your code:

```
Application.Calculation = True
```

EXAMPLES:

```
Sub mySlowMacro()

For x = 2 to 10000
        Cells(x,1) = x
        Cells(x,2) = x  + (x*.10)
Next x

End Sub
```

```
Sub myFasterMacro()
Application.ScreenUpdating = False
Application.Calculation = False
For x = 2 to 10000
        Cells(x,1) = x
        Cells(x,2) = x  + (x*.10)
Next x
Application.ScreenUpdating = True
Application.Calculation = True
End Sub
```

23. Disable/Enable Events

Same deal here: if you're worksheet needs to re-perform a macro every time a cell is selected in a worksheet, it may run a gazillion times over the course of a macro (especially for reports being run). Speed it up by turning off these worksheet or workbook events. Note: you can't turn off events in Userforms. You have to get a bit sneakier about doing that. . .

Beginning:

```
Application.EnableEvents = False
```

At the End of your code:

```
Application.EnableEvents = True
```

EXAMPLES:

```
Sub mySlowMacro()

For x = 2 to 10000
        Cells(x,1) = x
        Cells(x,2) = x  + (x*.10)
Next x

End Sub


Sub myFasterMacro()
Application.ScreenUpdating = False
Application.Calculation = False
Application.EnableEvents = False
For x = 2 to 10000
        Cells(x,1) = x
        Cells(x,2) = x  + (x*.10)
Next x
Application.ScreenUpdating = True
Application.Calculation = True
Application.EnableEvents = True
End Sub
```

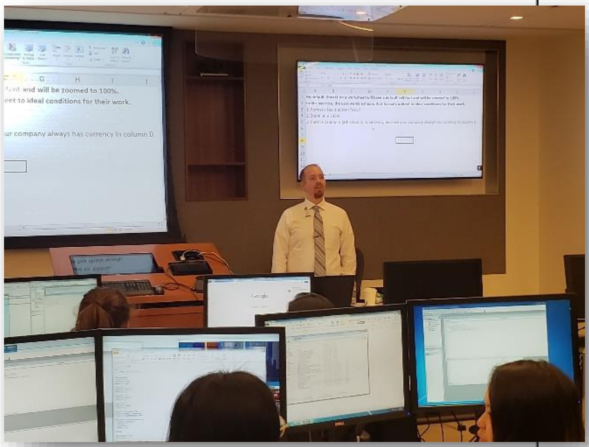
Check out the Video Here: http://www.youtube.com/watch?v=Wf7ci_obxaM

24. Binary, Baby!

HOT Tip!
If you save your workbook as an ".xlsb" instead of ".xlsm", it will run faster and take up less hard drive space. This is a Binary style workbook instead of the regular Macro Enabled Workbook. Cool!

View the Video Lesson to speed up your workbook Even more here:
http://www.youtube.com/watch?v=DsJOvT0Fzi4

Personal Message from Daniel Strong:



I hope you find this eBook useful and that it helps you to create your own extremely cool Excel programs!

Be sure to check out my Free and Premium Video Courses, resources, special Excel controls, and more at https://excelvbaisfun.com/shop/

God Bless and Happy Programming!

Dan Strong, ExcelVbaIsFun

CHECK FOR MORE DEALS @ https://excelvbaisfun.com/