

Excel Tips, 1 to 20

1

ARE WE CONNECTING?

Excel contains a number of drawing tools. One of them is the connector, which is great to use when creating flowcharts or organizational charts. Connectors attach to handles on other drawing objects, stretching and adjusting nicely if the objects they're attached to are moved on the worksheet. If you haven't used them, you should try them out!

Follow these steps:

1. Click the Drawing toolbar button to make sure that the drawing toolbar is visible (or select View | Toolbars | Drawing).
2. Draw a couple of shapes on the worksheet.
3. Click AutoShapes and select a style from the Connectors list.
4. Connect the shapes by drawing the connector from a handle on one shape to a handle on another.

Now move the shapes around: The connectors will adjust automatically!

2

ARE WE CONNECTING?

Excel contains a number of drawing tools. One of them is the connector, which is great to use when creating flowcharts or organizational charts.

Connectors attach to handles on other drawing objects, stretching and adjusting nicely if the objects they're attached to are moved on the worksheet. If you haven't used them, you should try them out!

Follow these steps:

1. Click the Drawing toolbar button to make sure that the drawing toolbar is visible (or select View | Toolbars | Drawing).
2. Draw a couple of shapes on the worksheet.
3. Click AutoShapes and select a style from the Connectors list.
4. Connect the shapes by drawing the connector from a handle on one shape to a handle on another.

Now move the shapes around: The connectors will adjust automatically!

3

AUDIT YOURSELF WITH THESE TOOLS

Excel's Auditing tools can help you understand how a worksheet is constructed--and they can help you trace the cause of errors. This is an invaluable set of tools when you're trying to figure out the mechanisms of a complex spreadsheet that someone else has constructed.

To familiarize yourself with these tools, it's a good idea to check out the Auditing toolbar (choose Tools | Auditing | Show Auditing Toolbar).

You can select a cell and display arrows to all formulas that depend on this cell. If a formula on another sheet depends on the cell, double-click the arrow to see the exact location of the dependent formula.

When a cell with a formula is selected, you can display arrows to the cells that this formula depends on. If an error value is the result of a formula, click the Trace Error button to locate the cause of the error.

4

SHIFT YOUR FOCUS WITH THESE KEYBOARD TRICKS

Many of the toolbar buttons in Excel have a different meaning if you click them while pressing the [Shift] key. Here are some examples:

- * If you press [Shift] while clicking the Align Left button on the Standard toolbar, it has the same effect as clicking Align Right.
- * If you'd like to see a print preview, hold down [Shift] and click the Print button.
- * Pressing [Shift] while clicking the Underline button makes a double underline.

If you'd like to see what a specific key combination does, point your mouse at a toolbar button and press the [Shift] key while holding down the left mouse button. The button image will change to the icon of what the combination will do. Move the mouse off the button before releasing the mouse button. Pressing [Shift] and clicking the File and Edit menus will display a few menu options that aren't usually available.

And here's a dangerous combination to avoid: When you quit Excel, you're asked if you want to save changes to the workbook. Clicking No while pressing [Shift] means no for all currently open workbooks, and Excel closes down without saving any of the changed workbooks.

5

THE QUICKEST WAY TO ENTER THE SAME VALUE

This is a very simple little trick, which can be useful when you want to enter the same value into several cells.

Follow these steps:

1. Select the cells where you want the value entered. To select several ranges, hold down [Ctrl] while selecting cells with the mouse.
2. Type a value.
3. Press [Ctrl][Enter].

All selected cells will display the value you just entered--no need for copying and pasting!

6

IMMEDIATE CALCULATIONS IN THE FORMULA BAR

If you enter a formula in the formula bar and press [F9], the result of the formula will be calculated, and you'll see only the result of the calculation.

If only a part of the formula's text is selected, that part will be evaluated when [F9] is pressed. If a range reference is selected, pressing [F9] replaces the reference with its values.

To see how it works, try out these three examples.

* Type the following in A1.

=25+11+9

Press [F9] and then press [Enter]--45 is entered into the cell.

* Type the following in A2.

=25+(1/8)

Select (1/8) with the mouse, and press [F9] and then [Enter]. The formula =25+.125 is written into A2.

* Enter a list of numbers in the range B1:B5. Select cell B6 and click the AutoSum button. The range reference B1:B5 is selected. Press [F9] and see what happens. Then press [Enter] to see what happens.

7

WHERE AM I?

If you want your users to know exactly where they're working in a workbook, use this trick to put the workbook's path, filename, or the active sheet's name into a cell.

Here's what to do: In an Excel workbook, enter the following formula in a cell.

=CELL("filename")

The result of this simple formula is the path of the workbook, the workbook's filename within square brackets, and the sheet name.

If you want to return only the path, filename, or worksheet name, you can use Excel's text functions. Here's how.

* To return only the path name, enter the following formula in a cell.

```
=LEFT(CELL("filename"),SEARCH("[",CELL("filename"))-1)
```

* To return only the filename, enter the following in a cell.

```
=MID(CELL("filename"),SEARCH("[",CELL("filename"))+1,SEARCH("[",CELL("filename"))-SEARCH("[",CELL("filename"))-1)
```

* To return only the worksheet name, enter this formula in a cell.

```
=RIGHT(CELL("filename"),LEN(CELL("filename"))-SEARCH("[",CELL("filename")))
```

8

COUNT WITH TWO CRITERIA

The COUNTIF function makes it easy to count the number of items that meet a certain condition. But what if you want to count the number of items that meet two or more criteria? Here's another function to use.

Follow these steps:

1. Enter State, Gender, and Age in A1:C1.
2. Enter state abbreviations, gender (M or F), and ages in A2:C15.

Use the following formula to count all rows where the state is TX and the gender is F.

```
=SUMPRODUCT(N(A2:A15="TX"),N(B2:B15="F"))
```

Use this formula to count all rows where the state is FL, the gender is M, and the age is above 50.

```
=SUMPRODUCT(N(A2:A15="FL"),N(B2:B15="M"),N(C2:C15>50))
```

9

USE A 3D RANGE TO TOTAL WORKSHEETS

Excel allows you to refer to ranges that continue through several sheets. This can shorten your formulas considerably when you perform calculations on multiple worksheets.

To see how it works, follow these steps:

1. Create a new workbook.

2. Add worksheets until you have 13 sheets.
3. Name the sheets by month (January through December) and name the last one Total.
4. Right-click a sheet tab and choose Select All Sheets. By selecting multiple sheets, you can modify several sheets at once.
5. Enter Apples, Bananas, and Pears in B1:D1.
6. Enter Bought and Sold in A2:A3.
7. Enter values in B2:D3.

You now have 13 identical sheets. Let's add up the first 12 sheets on the last sheet. Follow these steps:

1. Select the Total worksheet.
2. Enter the following formula in cell B2.

```
=SUM(January:December!B2)
```

3. Copy the formula and paste it into B2:D3.

The formula uses a 3D range. It adds the B2:D3 cells for all sheets from January to December.

[Back to TOP](#)

10

CONVERT FORMULAS TO VALUES

Here's a method that converts formulas to values so that you can delete the old values after the new ones have been calculated. For example, let's say you have employees' full names entered in one column, but you'd like to separate them into first and last name columns.

Here's what you can do. Follow these steps:

1. Enter these values in A1:A5.

Pat Johnson
Mike Baker
Pat Brown
Jim Freeman
Rob Pearson

2. Enter these formulas in B1 and C1.

```
=LEFT(A1,SEARCH(" ",A1)-1)  
=MID(A1,SEARCH(" ",A1)+1,LEN(A1))
```

3. Select B1:C1 and use the fill handle to copy the formulas down to B5:C5. The range B1:C5 now contains formulas, which refer to A1:A5. Let's convert these formulas to values.

4. With B1:C5 selected, point the mouse at the range border (not the fill handle). Hold down the right mouse button and drag the range a little to the right and then back again.

5. Select Copy Here As Values Only from the shortcut menu.

You can now delete the values in A1:A5. B1:C5 no longer contains formulas that rely on those values.

11

GET TO KNOW JULIAN DATES

Some computer applications store dates in Julian date format. Julian dates are written as a five-digit number: The first two digits are the year, and the last three digits are the day number of the year. For example, Jan. 1, 1999 has the Julian date 99001. Feb. 2, 2000 has the Julian date 00033.

Excel has no built-in date format for Julian dates, but here's a formula that will convert a date in cell A1 to a Julian date.

```
=TEXT(RIGHT(YEAR(A1),2),"00") & TEXT(A1-DATE(YEAR(A1),1,0),"000")
```

To convert a Julian date in cell B1 to an Excel date, you can use the following formula. The cell that includes the Julian date must be formatted as a text string.

```
=DATE(IF(LEFT(B1,2)<"30",2000,1900)+LEFT(B1,2),1,RIGHT(B1,3))
```

12

SPLIT TEXTS IN CELLS INTO MULTIPLE LINES

When typing text into a cell, you can add line breaks within the cell by pressing [Alt][Enter]. The cell formatting is automatically set to wrap text, and a line break character is inserted. You don't see the character, but it's still there.

Let's try it. Follow these steps:

1. Select cell A1.
2. Type Apple, press [Alt][Enter], and type Pie.
3. Select cell B1 and enter the following formula.

```
=LEN(A1)
```

The formula in cell B1 tells us that the text in cell A1 is nine characters long. A line break accounts for one character.

Select cell A1 and choose Format | Cells. Click the Alignment tab: You'll notice that the Wrap Text check box has been selected.

13

Today's Excel Tip

KEEP A RUNNING TOTAL

To create a running total, you can use a combination of absolute and relative references. The fixed reference will stay as it is when you copy the formula, but the relative reference will change to reflect the new position of the formula.

To see how it works, follow these steps:

1. Enter numbers in the range A1:A10.
2. Enter the following formula in cell B1.

```
=SUM($A$1:A1)
```

\$A\$1 is an absolute reference. If a dollar sign [\$] precedes something, the reference won't change when the formula is copied to a new location.

A1 is a relative reference. When we copy the formula to another cell, the reference will change to reflect the new location of the formula.

Try it out: Copy cell B1 down to B10 by using the fill handle. Examine the formulas in B2:B10 and see how they've changed. This is a very practical and easy way to calculate a running total.

14

USE APPLICATION.INTERACTIVE TO CONTROL EXCEL

If you want to make sure that your VBA code has complete control over the Excel environment and that your users don't interfere by making manual changes, you can set the Application.Interactive property to False.

This will block all keyboard and mouse input to Excel--except to the message boxes, dialogs, and forms displayed by the VBA code. But make sure you set this property back to True afterward, or you won't be able to use Excel.

Below are two sample procedures that use this property. The first procedure locks Excel, displays a message in the status bar, and schedules the backup procedure to run at 05:00. The second procedure clears the status bar, copies a file, and unlocks Excel.

```
Sub LockExcelAndStartTimer()  
Application.Interactive = False  
Application.StatusBar = "Excel is waiting until 05:00"  
Application.OnTime TimeValue("05:00:00"), "BackUp"
```

```
End Sub
```

```
Sub BackUp()  
Application.StatusBar = False  
FileCopy Source:="c:\test\test.xls", Destination:="d:\test\test.xls"  
Application.Interactive = True  
End Sub
```

15

USE GETOPENFILENAME TO SELECT FILES

You can use the Application.GetOpenFilename method to select files that you want to do something with. The selected files are not actually opened, but their names are returned in a variable so that your application can use them.

The sample procedure below lets the user select workbooks that are copied to files with the extension .bkp.

```
Sub BackUpSelectedFiles()  
'Declare variables.  
Dim SelectedFiles As Variant  
Dim OldName As String  
Dim NewName As String  
Dim x As Integer  
  
'Set active drive and directory.  
ChDrive "c:"  
ChDir "c:\test"  
  
'Select files.  
SelectedFiles = Application.GetOpenFilename( _  
FileFilter:="Excel Workbooks (*.xls), *.xls", _  
Title:="Select files to back up", _  
MultiSelect:=True)  
  
'Check if Cancel was clicked.  
If SelectedFiles = False Then Exit Sub  
  
'Loop through selected files and copy them.  
For x = LBound(SelectedFiles) To UBound(SelectedFiles)  
OldName = SelectedFiles(x)  
NewName = Application.Substitute(OldName, ".xls", ".bkp")  
FileCopy Source:=OldName, Destination:=NewName  
Next x  
End Sub
```

16

EVENT HANDLING PROCEDURES

Excel can react to certain events and execute code when these events occur.

Typical events include when a workbook is opened, when a worksheet cell is changed, or when a user activates or deactivates a sheet. Event handling procedures can easily be added in the VBA code for a sheet or for thisWorkbook.

To try it out, follow these steps:

1. In Excel, press [Alt][F11] to activate the Visual Basic Editor.
2. Double-click ThisWorkbook to show its code module.
3. Select Workbook from the first drop-down list at the top of the code window, and select BeforePrint from the second drop-down list.

The Workbook_BeforePrint procedure is created automatically for you. If you try to print any sheet in the workbook, Excel will fire the code that's in the workbook_BeforePrint procedure. Some event procedures have a Cancel variable that can be set to True to cancel the event that caused the procedure to run. The following procedure will stop you from printing the workbook.

```
Private Sub Workbook_BeforePrint(Cancel As Boolean)
MsgBox "No, no!"
Cancel = True
End Sub
```

17

VBA CAN MODIFY A WORKSHEET EVEN WHEN IT'S PROTECTED

You may have used the sheet protection that's available from the Tools menu in Excel. Sometimes it would be handy if users aren't allowed to change the sheet, but VBA code is still able to modify the sheet--without having to unprotect it first. This can be done quite easily! Follow these steps:

1. Start Excel with a new workbook.
2. Press [Alt][F11] to open the Visual Basic Editor.
3. Type the following in the Immediate window.

```
Sheet1.Protect Password:="carrot", UserInterfaceOnly:=True
```

Sheet1 is now protected. You can't enter data into the sheet or modify it in any other way, except through VBA. Let's try it:

1. Press [Alt][F11] to return to Excel.
2. Try to enter a value in cell A1.
3. Try setting cell A1 to bold.

No good. Go back to the VB Editor again and type the following in the Immediate window.

```
Range("A1").Value="xxx"
```

```
Range("A1").Font.Bold=True
```

Switch back to Excel, and you'll see that the VBA statements were able to modify the sheet.

18

CREATE YOUR OWN FUNCTIONS

User-defined functions can come in handy when you want to perform complex calculations. By creating your own functions, you can extend the functionality of Excel and make your worksheet formulas more readable.

A couple of weeks ago, we introduced you to the Julian date format and offered formulas to help convert to and from these dates ("Get to know Julian dates," Dec. 22, 2000). But the formulas are rather long and can be hard to remember if you're not using them frequently. To show how user-defined functions work, let's create two functions that convert to and from Julian dates.

Follow these steps:

1. In Excel, open the Visual Basic Editor by pressing [Alt][F11].
2. Select Insert | Module.
3. Type the following code into the module:

```
Function Julian(d As Date) As String
Julian = Format(d, "YY") & Format(d - DateSerial(Year(d), 1, 0),
"000")
End Function
```

```
Function UnJulian(j As String) As Date
If Left(j, 2) < "30" Then
UnJulian = DateSerial(2000 + Left(j, 2), 1, Right(j, 3))
Else
UnJulian = DateSerial(1900 + Left(j, 2), 1, Right(j, 3))
End If
End Function
```

After you've entered the code, you can use the functions in a workbook formula, just as you use Excel's built-in functions.

19

LINK UP YOUR CELLS

Here's a little trick to use if you want to make a two-way link. In this example, we want to make sure that cells A1 and D1 always contain the same value. If you change A1, D1 should also change and vice versa.

1. In Excel, right-click the worksheet tab and select View Code from the shortcut menu. This will open the VB Editor.
2. Enter the following code:

```
Private Sub Worksheet_Change(ByVal Target As Range)
'Turn off event handling so that the procedure
'doesn't start itself up again when changing cells
Application.EnableEvents = False
```

```
'If cell A1 is in the changed range...
If Not Intersect(Target, Range("A1")) Is Nothing Then
Range("D1").Value = Range("A1").Value
```

```
'Else, if cell D1 is in the changed range...
Elseif Not Intersect(Target, Range("D1")) Is Nothing Then
Range("A1").Value = Range("D1").Value
```

```
End If
```

```
'Turn event handling back on
Application.EnableEvents = True
End Sub
```

Go back to Excel by pressing [Alt][F11]. See what happens when you type something in A1 or D1.

20

USING THE USED RANGE PROPERTY

The UsedRange property returns the range from the first used row and column to the last used row and column. You should be aware that this range doesn't necessarily start on row 1 or in column A. The number of rows or columns in the used range may not be equal to the number of the last used row or column.

To see how it works, follow these steps:

1. Enter values in the range D3:G9 on an empty sheet.
2. Press [Alt][F11] to open the Visual Basic Editor.
3. Type the following code in the Immediate window.

```
ActiveSheet.UsedRange.Select
```

4. Press [Alt][F11] to return to Excel. As you see, the first three columns and the first two rows of the worksheet are not selected.
5. Go back to the VB Editor and type the following in the Immediate window.

```
MsgBox ActiveSheet.UsedRange.Rows.Count
```

Of course, the result is 7--that's the number of rows in UsedRange. You can use the following methods to find the number of the last used row or column in a sheet.

MsgBox ActiveSheet.Cells.SpecialCells(xlCellTypeLastCell).Row
MsgBox ActiveSheet.Cells.SpecialCells(xlCellTypeLastCell).Column