Getting Familiar with the Insert Function Dialog Box

The Insert Function dialog box (shown in Figure 2-1) is designed to simplify the task of using functions in your worksheet. The dialog box not only helps you locate the proper function for the task at hand, but also provides information about the arguments that the function takes. If you use the Insert Function dialog box, you don't have to type functions directly in worksheet cells. Instead, the dialog box guides you through a (mostly) point-and-click procedure — a good thing, because if you're anything like me, you need all the help you can get.

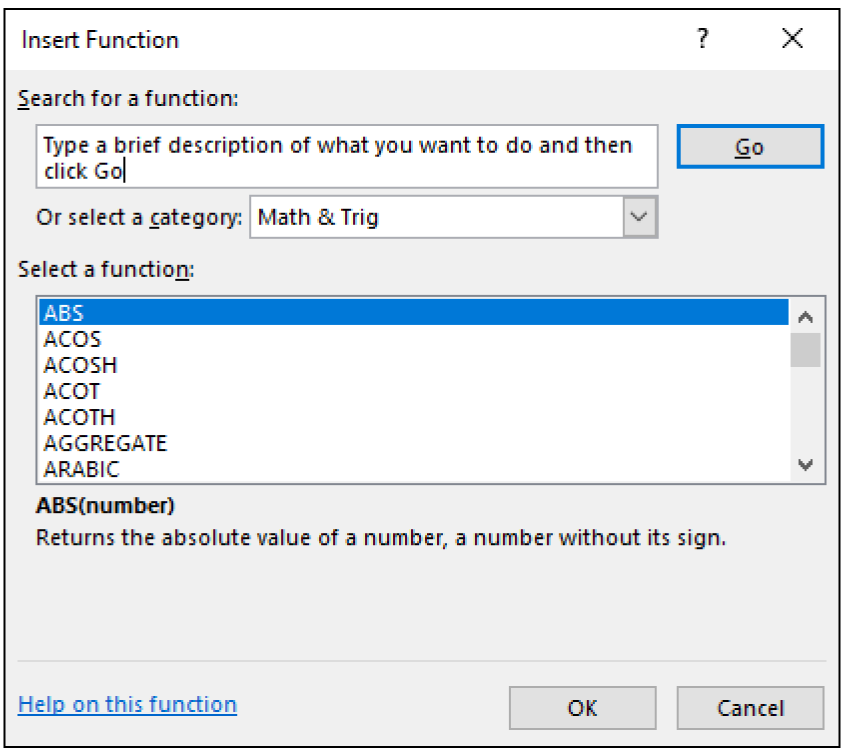


FIGURE 2-1: Use the Insert Function dialog box to easily enter functions in a worksheet.

In the Insert Function dialog box, you can browse functions by category or scroll the complete alphabetical list. A search feature — you type a word or phrase in the Search for a Function box, click the Go button, and see what comes up — is helpful. When you highlight a function in the Select a Function box, a brief description of what the function does appears under the list. You can also click the Help on This Function link at the bottom of the dialog box to view more detailed information about the function.

You can display the Insert Function dialog box in three ways:

- Click the Insert Function button on the Formulas tab.
- On the Formula Bar, click the smaller Insert Function button (which looks like *fx*).
- Click the small arrow to the right of the AutoSum feature on the Formulas tab, and select More Functions (see Figure 2-2). AutoSum has a list of commonly used functions that you can insert with a click. If you select More Functions, the Insert Function dialog box opens.

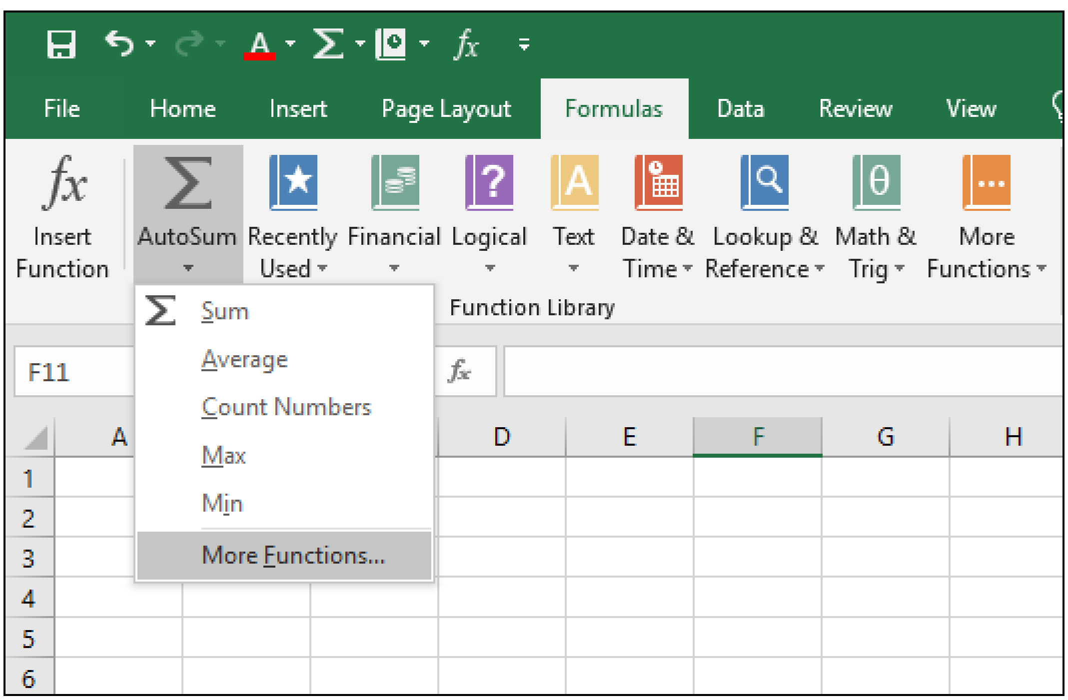


FIGURE 2-2: The AutoSum button offers quick access to basic functions and the Insert Function dialog box.

Finding the Correct Function

The first step in using a function is finding the one you need! Even when you do know the one you need, you may not remember all the arguments it takes. You can find a function in the Insert Function dialog box in two ways:

- **Search:** Type one or more keywords or a phrase in the Search for a Function box, then click the Go button.
  - If a match is made, the Or Select a Category drop-down menu displays Recommended, and the Select a Function box displays a list of the functions that match your search.
  - If no match is made, the Or Select a Category drop-down menu displays Most Recently Used functions, and the most recently used functions appear in the Select a Function dialog box. The Search for a Function box displays a message to rephrase the text entered for the search.
- **Browse:** Click the Or Select a Category down arrow, and from the drop-down menu, select All or an actual function category. When an actual category is selected, the Select a Function box updates to show just the relevant functions. You can look through the list to find the function you want. Alternatively, if you know the category, you can select it on the Formulas tab.

Table 2-1 lists the categories in the Or Select a Category drop-down menu. Finding the function you need is different from knowing which function you need. Excel is great at giving you the functions, but you do need to know what to ask for.

TABLE 2-1 Function Categories in the Insert Function Dialog Box

Category	*Type of Functions*
Most Recently Used	The last several functions you used.
All	The entire function list, sorted alphabetically.
Financial	Functions for managing loans, analyzing investments, and so on.
Date & Time	Functions for calculating days of the week, elapsed time, and so on.
Math & Trig	A considerable number of mathematical functions.
Statistical	Functions for using descriptive and inferential statistics.
Lookup & Reference	Functions for obtaining facts about and data on worksheets.

Database	Functions for selecting data in structured rows and columns.
Text	Functions for manipulating and searching text values.
Logical	Boolean functions (AND, OR, and so on).
Informati on	Functions for getting facts about worksheet cells and the data therein.
Web	A few functions that are useful when sharing data with web services.
Engineeri ng	Engineering and some conversion functions. These functions are also provided in the Analysis ToolPak.
Cube	Functions used with online analytical processing (OLAP) cubes.

Compatibility	Some functions were updated as of Excel 2010 and Excel 2013. The functions in this category are the older versions that remain compatible with Excel 2007 and earlier versions.
User Defined	Any available custom functions created in VBA code or from add-ins. This category may not be listed.

Entering Functions Using the Insert Function Dialog Box

Now that you've seen how to search for or select a function, it's time to use the Insert Function dialog box to actually insert a function. The dialog box makes it easy to enter functions that take no arguments and functions that *do* take arguments. Either way, the dialog box guides you through the process of entering the function.

Sometimes, function arguments are not values, but references to cells, ranges, named areas, or tables. That this is also handled in the Insert Function dialog box makes its use so beneficial.

SELECTING A FUNCTION THAT TAKES NO ARGUMENTS

Some functions return a value, period. No arguments are needed for these functions. This means you don’t have to have some arguments ready to go. What could be easier? Here’s how to enter a function that does not take any arguments. The TODAY function is used in this example:

1. **Position the cursor in the cell where you want the results to appear.**
2. **Click the Insert Function button on the Formulas tab to open the Insert Function dialog box.**
3. **Select All in the Or Select a Category drop-down menu.**
4. **Scroll through the Select a Function list until you see the TODAY function, and click it.**

   Figure 2-3 shows what the screen looks like.
5. **Click OK.**

   The Insert Function dialog box closes, and the Function Arguments dialog box opens. The dialog box tells you that function does not take any arguments. Figure 2-4 shows how the screen looks now.
6. **Click OK.**

   Doing this closes the Function Arguments dialog box, and the function entry is complete.

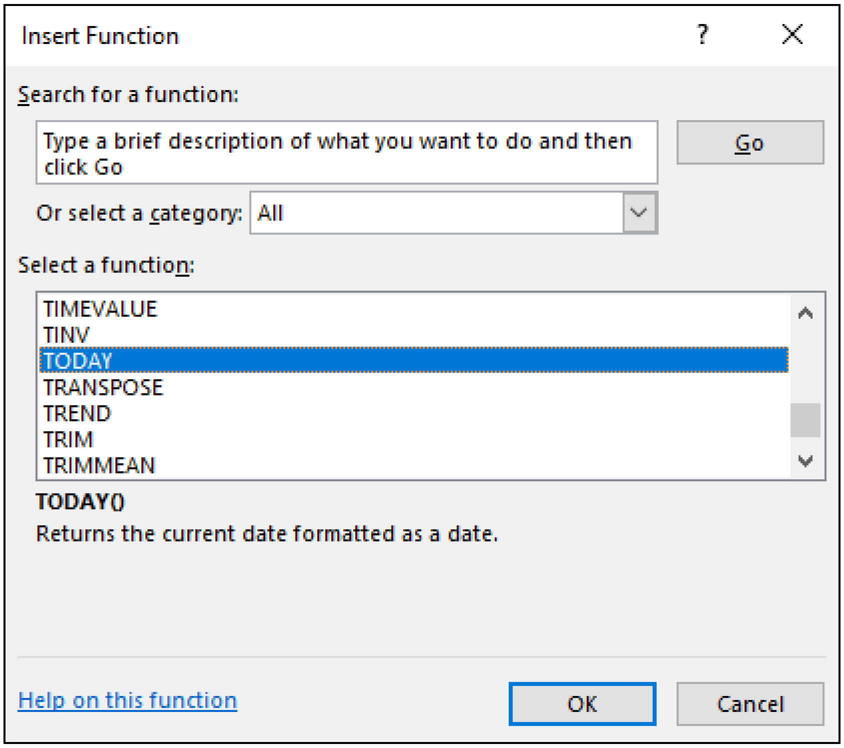


FIGURE 2-3: Selecting a function.

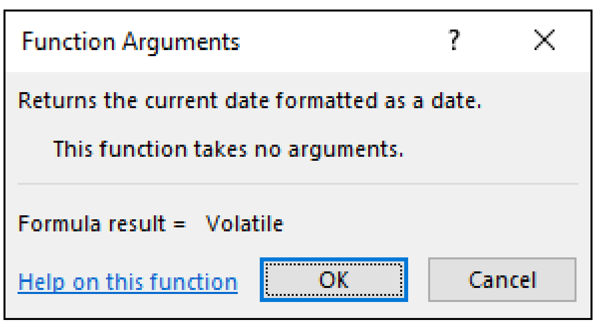


FIGURE 2-4: Confirming that no arguments exist with the Function Arguments dialog box.

You may have noticed that the Function Arguments dialog box says that the Formula result will equal `Volatile`. This is nothing to be alarmed about! This just means the answer can be different each time you use the function. For example, TODAY will return a different date when used tomorrow.

Figure 2-5 shows how the function’s result has been returned to the worksheet. Cell B2 displays the date when I wrote this example. The date you see on your screen is the current date.

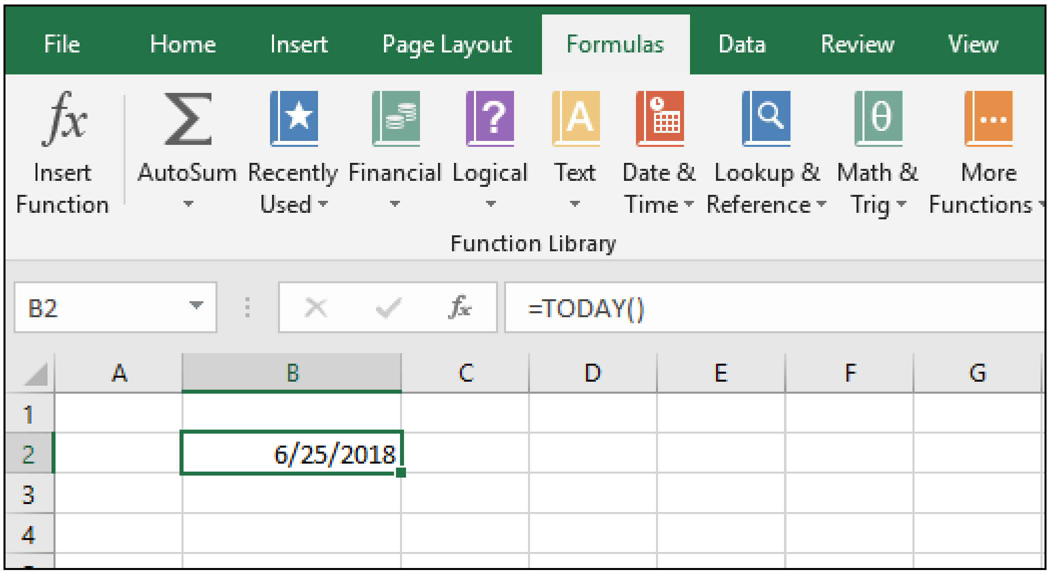


FIGURE 2-5: Populating a worksheet cell with today’s date.

REMEMBER Most functions do take arguments. The few that *do not* take arguments can return a result without needing any information. For example, the TODAY function just returns the current date. It doesn’t need any information to figure this out.

SELECTING A FUNCTION THAT USES ARGUMENTS

Most functions take arguments to provide the information that the functions need to perform their calculations. Some functions use a single argument; others use many. *Taking arguments* and *using arguments* are interchangeable terms. Most functions take arguments, but the number of arguments depends on the actual function. Some functions take a single argument, and some can take up to 255.

The following example shows how to use the Insert Function dialog box to enter a function that *does* use arguments. The example uses the PRODUCT function. Here's how to enter the function and its arguments:

1. **Position the cursor in the cell where you want the results to appear.**
2. **Click the Insert Function button on the Formulas tab.**
   Doing this opens the Insert Function dialog box.
3. **Select Math & Trig in the Or Select a Category drop-down menu.**
4. **Scroll through the Select a Function list until you see the PRODUCT function and then click it.**
   Figure 2-6 shows what the screen looks like.
5. **Click OK.**
   The Insert Function dialog box closes, and the Function Arguments dialog box opens. Figure 2-7 shows what the screen looks like. The dialog box tells you that this function can take up to 255 arguments, yet there appears to be room for only 2. As you enter arguments, the dialog box provides a scroll bar to manage multiple arguments.
6. **In the Function Arguments dialog box, enter a number in the Number1 box.**
7. **Enter another number in the Number2 box.**
   You are entering actual arguments. As you enter numbers in the dialog box, a scroll bar appears, letting you add arguments. Enter as many as you like, up to 255. Figure 2-8 shows how I entered eight arguments. Also look at the bottom left of the dialog box. As you enter functions, the formula result is instantly calculated. Wouldn't it be nice to be that smart?
8. **Click OK to complete the function entry.**
   Figure 2-9 shows the worksheet's result.

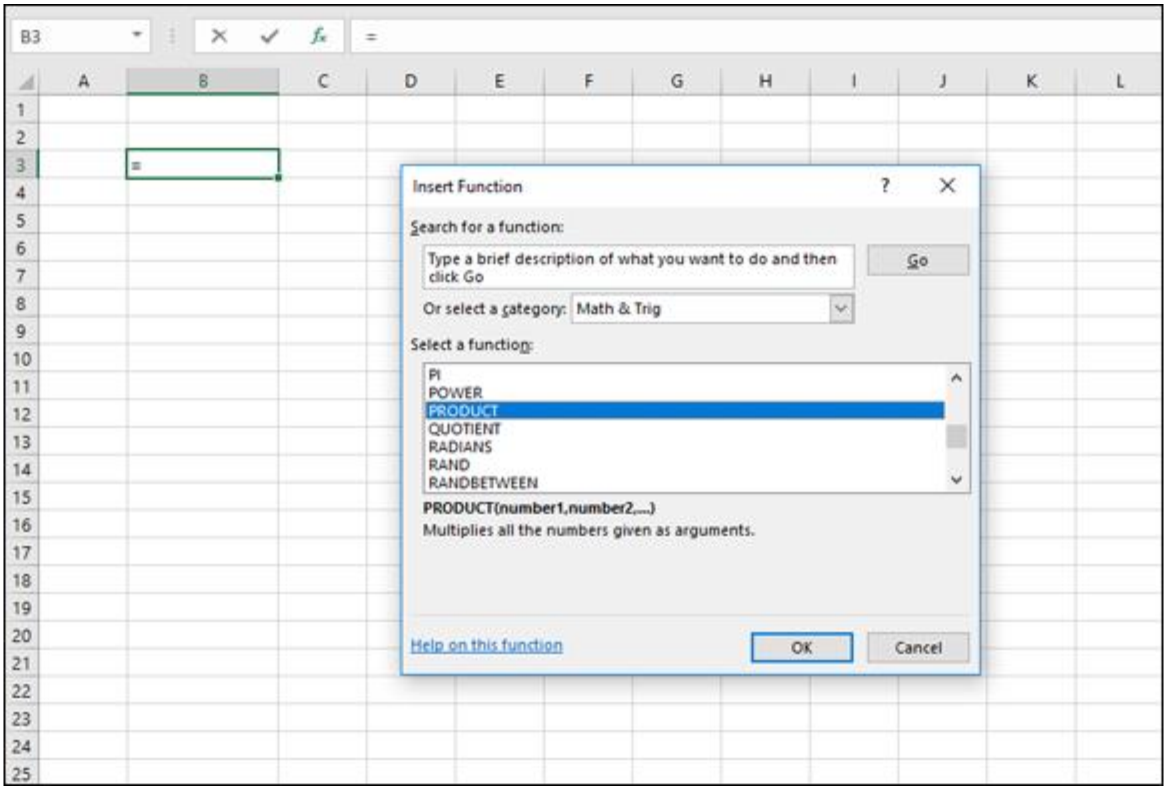


FIGURE 2-6: Preparing to multiply some numbers with the PRODUCT function.

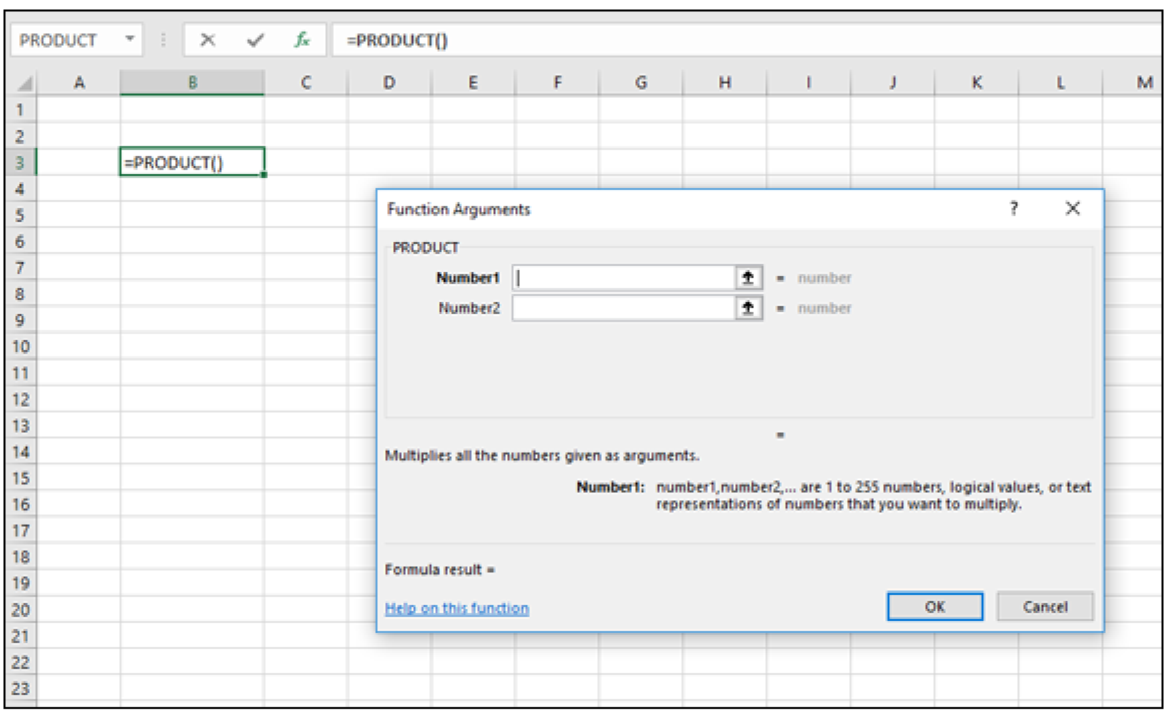


FIGURE 2-7: Ready to input function arguments.

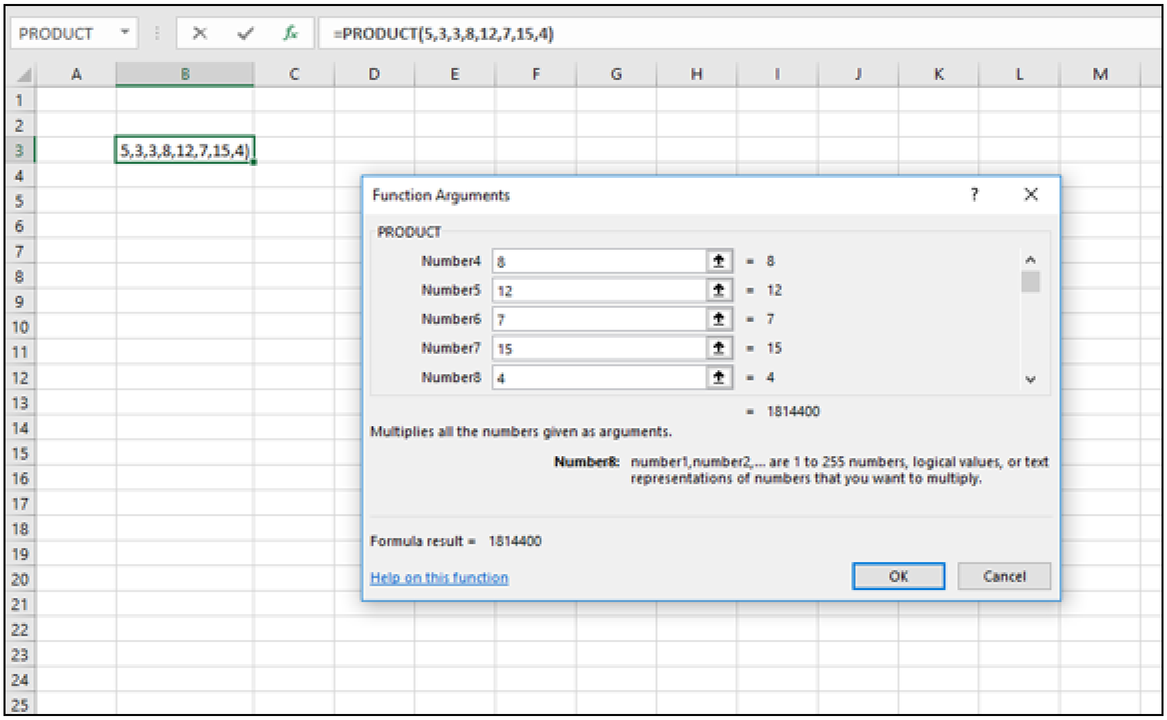


FIGURE 2-8: Getting instant results in the Function Arguments dialog box.

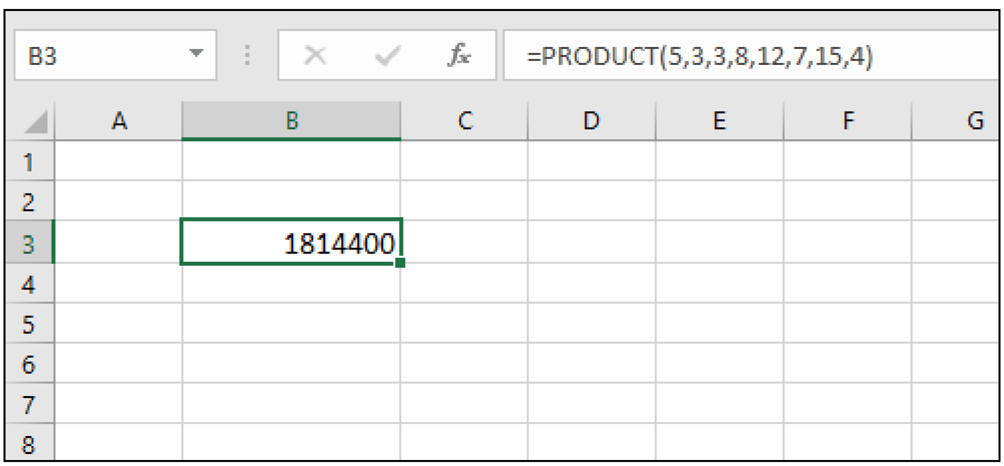


FIGURE 2-9: Getting the final answer from the function.

ENTERING CELLS, RANGES, NAMED AREAS, AND TABLES AS FUNCTION ARGUMENTS

Excel is so cool. You can not only provide single cell references as arguments, but also, in many cases you can enter an entire range reference, or the name of an area or table, as a single argument! What's more, you can enter these arguments by using either the keyboard or the mouse.

This example demonstrates using both single cell and range references as well as a named area and table as arguments. For this example, I use the SUM function. Here's how to use the Insert Function dialog box to enter the function and its arguments:

1. **Enter some numbers in a worksheet in contiguous cells.**
2. **Select the cells and then click the Table button on the Insert tab.**
   The Create Table dialog box opens.
3. **Click OK to complete making the table.**
   The Ribbon should display table style and other options. (If not, look along the Excel title bar for Table Tools, and click it.) On the left end of the Ribbon is the name that Excel gave the table. You can change the name of the table, as well as the appearance. Jot down the name of the table. You need to re-enter the table name further in these steps.
4. **Somewhere else on the worksheet, enter numbers in contiguous cells.**
5. **Select the cells and then click the Define Name button on the Formulas tab.**
   The New Name dialog box opens.
6. **Enter a name for the area.**
   I used the name MyArea. See Figure 2-10 to see how the worksheet is shaping up.
7. **Enter some more numbers in contiguous cells, either across a row or down a column.**
8. **Enter a single number in cell A1.**

9. **Click an empty cell where you want the result to appear.**
10. **Click the Insert Function button on the Formulas tab.**

    The Insert Function dialog box opens.
11. **Select the SUM function.**

    SUM is in the All or Math & Trig category, and possibly in the Recently Used category.
12. **Click OK.**

    The Function Arguments dialog box opens.

    REMEMBER To the right of each Number box is a small fancy button — a special Excel control sometimes called the *RefEdit*. It allows you to leave the dialog box, select a cell or range on the worksheet, and then go back to the dialog box. Whatever cell or range you click or drag over on the worksheet is brought into the entry box as a reference.

    You can type cell and range references, named areas, and table names directly in the Number boxes as well. You can also click directly on cells or ranges on the worksheet. The RefEdit controls are there to use if you want to work with the mouse instead.
13. **Click the first RefEdit.**

    The dialog box shrinks so that the only thing visible is the field where you enter data. Click cell A1, where you entered a number.
14. **Press Enter.**

    The Function Arguments dialog box reappears.
15. **In the second entry box, type the name of your named area.**

    If you don’t remember the name you used, use the RefEdit control to select the area on the worksheet.
16. **In the third entry box, enter your table name and press Enter.**
17. **If you don’t remember the name you used, use the RefEdit control to select the table.**
18. **In the fourth entry box, enter a range from the worksheet where some values are located.**

    It does not matter if this range is part of a named area or table. Use the

RefEdit control if you want to just drag the mouse over a range of numbers. Your screen should look similar to Figure 2-11.

19. **Click OK.**

The final sum from the various parts of the worksheet displays in the cell where the function was entered. Figure 2-12 shows how the example worksheet turned out.

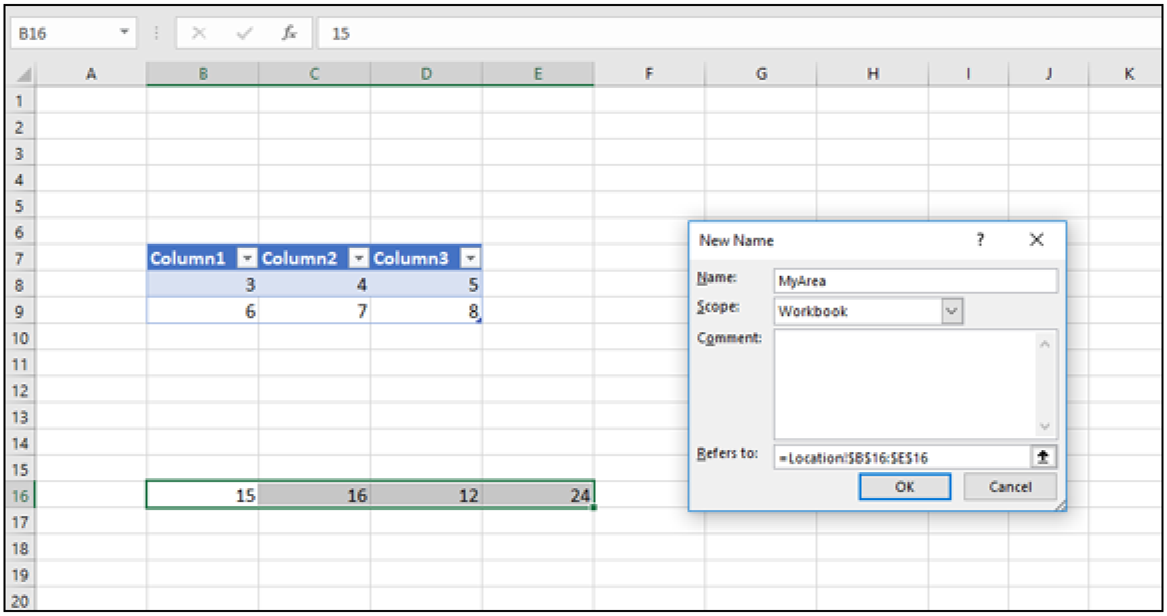


FIGURE 2-10: Adding a table and a named area to a worksheet.

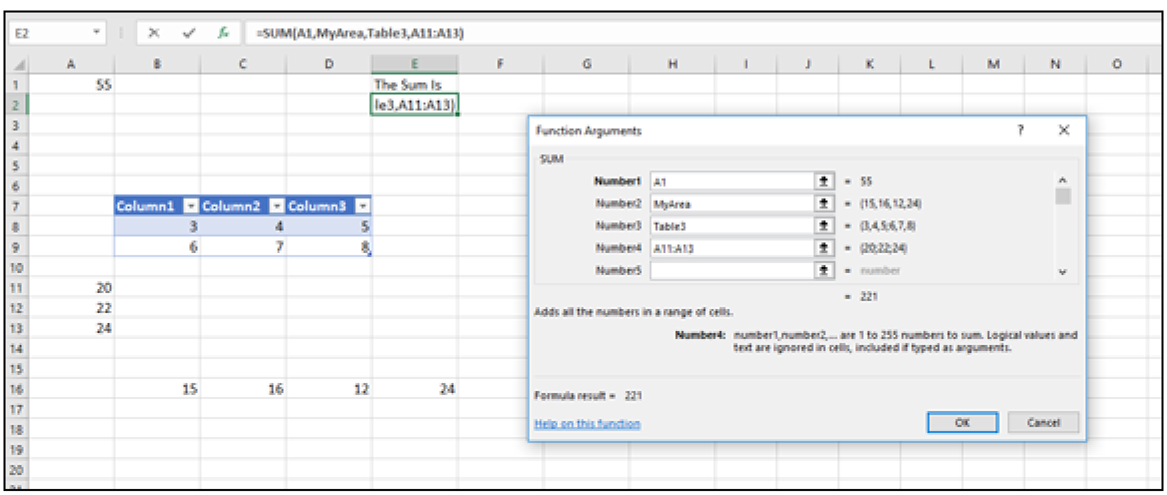


FIGURE 2-11: Entering arguments.

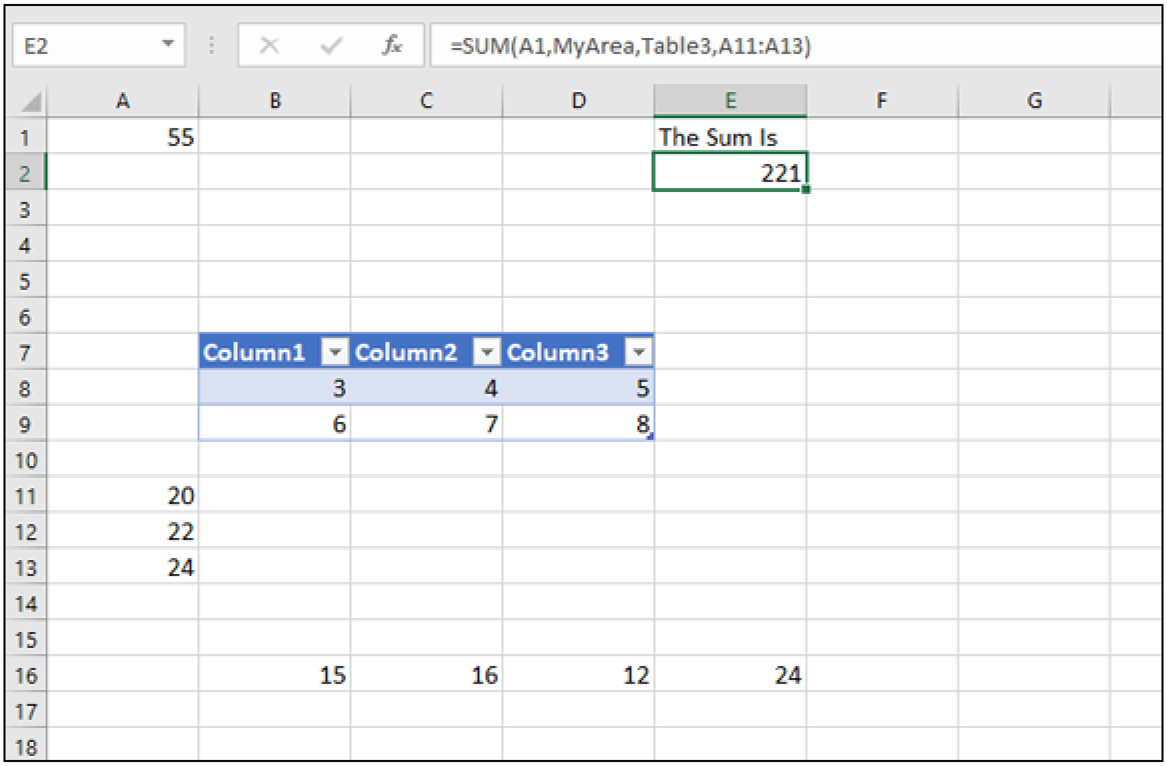


FIGURE 2-12: Calculating a sum based on cell and range references.

Congratulations! You did it. You successfully inserted a function that took a cell reference, a range reference, a named area, and a table name. You're harnessing the power of Excel. Look at the result — the sum of many numbers located in various parts of the worksheet. Just imagine how much summing you can do. You can have up to 255 inputs, and if necessary, each one can be a *range* of cells.

TIP

You can use the Insert Function dialog box at any time while entering a formula. This is helpful when the formula uses some values and references in addition to a function. Just open the Insert Function dialog box when the formula entry is at the point where the function goes.

GETTING HELP IN THE INSERT FUNCTION DIALOG BOX

The number of functions and their exhaustive capabilities give you the power to do great things in Excel. However, from time to time, you may need guidance on how to get functions to work. Luckily for you, help is just a click away.

Both the Insert Function and Function Arguments dialog boxes have a link to the Help system. At any time, you can click the Help on This Function link in the lower-left corner of the dialog box and get help on the function you're using. The Help system has many examples. Often, reviewing how a function works leads you to other, similar functions that may be better suited to your situation.

USING THE FUNCTION ARGUMENTS DIALOG BOX TO EDIT FUNCTIONS

Excel makes entering functions with the Insert Function dialog box easy. But what do you do when you need to change a function that has already been entered in a cell? What about adding arguments or taking some away? There is an easy way to do this! Follow these steps:

1. **Click the cell with the existing function.**
2. **Click the Insert Function button.**
   The Function Argument dialog box appears. This dialog box is already set to work with your function. In fact, the arguments that have already been entered in the function are displayed in the dialog box as well!
3. **Add, edit, or delete arguments, as follows:**
   - To add an argument (if the function allows), use the RefEdit control to pick up the extra values from the worksheet. Alternatively, if you click the bottom argument reference, a new box opens below it, and you can enter a value or range in that box.
   - To edit an argument, simply click it and change it.
   - To delete an argument, click it and press the Backspace key.

4. **Click OK when you're finished.**
   The function is updated with your changes.

Directly Entering Formulas and Functions

As you get sharp with functions, you will likely bypass the Insert Function dialog box altogether and enter functions directly. One place you can do this is in the Formula Bar. Another way is to just type in a cell.

ENTERING FORMULAS AND FUNCTIONS IN THE FORMULA BAR

When you place your entry in the Formula Bar, the entry is really going into the active cell. However, because the active cell can be anywhere, you may prefer entering formulas and functions directly in the Formula Bar. That way, you know that the entry will land where you need it. Before you enter a formula in the Formula Box (on the right end of the Formula Bar), the Name Box on the left lets you know where the entry will end up. The cell receiving the entry may be not be in the visible area of the worksheet. Gosh, it could be a million rows down and thousands of columns to the right! After you start entering the formula, the Name Box becomes a drop-down menu of functions. This menu is useful for nesting functions. As you enter a function in the Formula Box, you can click a function in the Name Box, and the function is inserted into the entry you started in the Formula Box. Confused? Imagine what I went through explaining that! Seriously, though, this is a helpful way to assemble nested functions. Try it, and get used to it; it will add to your Excel smarts.

When your entry is finished, press Enter or click the little check-mark Enter button to the left of the Formula Box.

Figure 2-13 makes this clear. A formula is being entered in the Formula Box, and the Name Box follows along with the function(s) being entered. Note, though, that the active cell is not in the viewable area of the worksheet. It must be below and/or to the right of the viewable area because the top-left portion of the worksheet is shown in Figure 2-13.

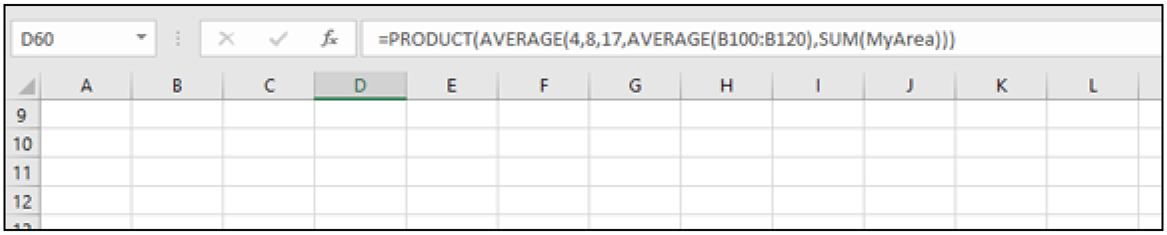


FIGURE 2-13: Entering a formula in the Formula Box has its conveniences.

In between the Name Box and the Formula Box are three small buttons. From left to right, they do the following:

- Cancel the entry.
- Complete the entry.
- Display the Insert Function dialog box.

REMEMBER The Cancel and Enter Function buttons are enabled only when you enter a formula, a function, or just plain old values on the Formula Bar or directly in a cell.

ENTERING FORMULAS AND FUNCTIONS DIRECTLY IN WORKSHEET CELLS

Perhaps the easiest entry method is typing the formula directly in a cell. Just type formulas that contain no functions and press Enter to complete the entry. Try this simple example:

1. **Click a cell where the formula is to be entered.**

2. **Enter this simple math-based formula:**
   =6 + (9/5) *100
3. **Press Enter.**
   The answer is 186. (Don't forget the order of operators; see Chapter 18 for more information about the order of mathematical operators.)

Excel makes entering functions in your formulas as easy as a click. As you type the first letter of a function in a cell, a list of functions starting with that letter is listed immediately (see Figure 2-14).

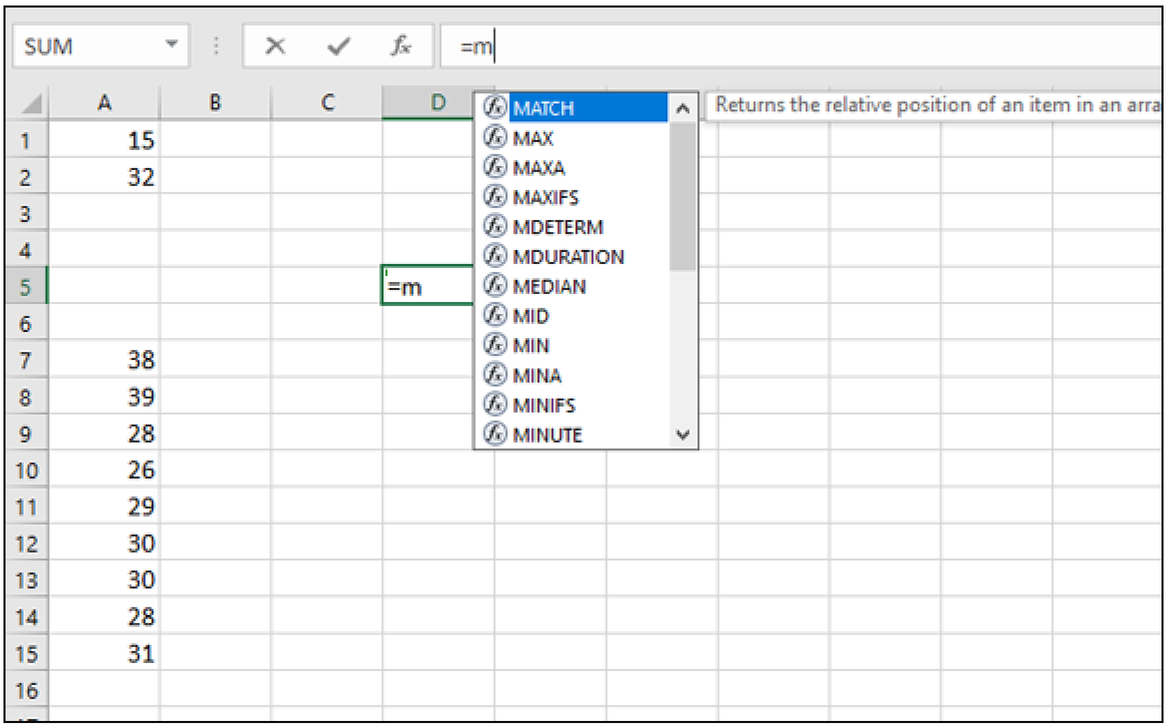


FIGURE 2-14: Entering functions has never been this easy.

The desired function in this example is MIN, which returns the minimum value from a group of values. As soon as you type *M* (first enter the equal sign if this is the start of a formula entry), the list in Figure 2-14 appears, showing all the M functions. Now that an option exists, either keep typing the full function name, or scroll to MIN and press the Tab key. Figure 2-15 shows just what happens when you do the latter. MIN is completed and provides the required syntax structure — not much thinking involved! Now your brain can concentrate on more interesting things, such as poker odds. (Will Microsoft ever create a function category for calculating poker odds? Please?) In Figure 2-15, the MIN function is used to find the minimum value in the range A7:A15 (which is multiplied with the sum of the values in A1 plus A2). Entering the closing parenthesis and then pressing Enter completes the function. In this example, the answer is 1222.

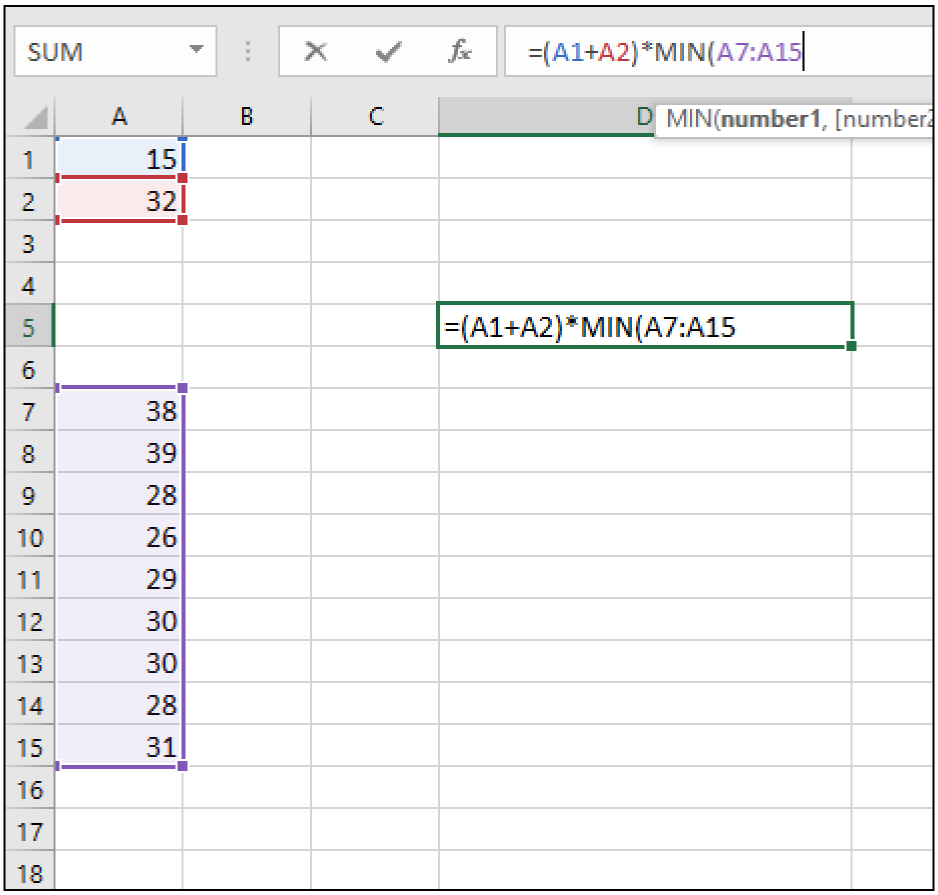


FIGURE 2-15: Completing the direct-in-the-cell formula entry.

A1: 15

A2: 32

A7: 38

A8: 39

A9: 28

A10: 26

A11: 29

A12: 30

A13: 30

A14: 28

A15: 31

The formula in D5 is =(A1+A2) * MIN(A7:A15).

REMEMBER Excel’s capability to show a list of functions based on spelling is called *Formula AutoComplete.*

You can turn Formula AutoComplete on or off in the Excel Options dialog box by following these steps:

1. **Click the File tab at the top left of the screen.**
2. **Click Options.**
3. **In the Excel Options dialog box, select the Formulas tab.**
4. **In the Working with Formulas section, select or deselect the Formula AutoComplete check box. See Figure 2-16.**
5. **Click OK.**

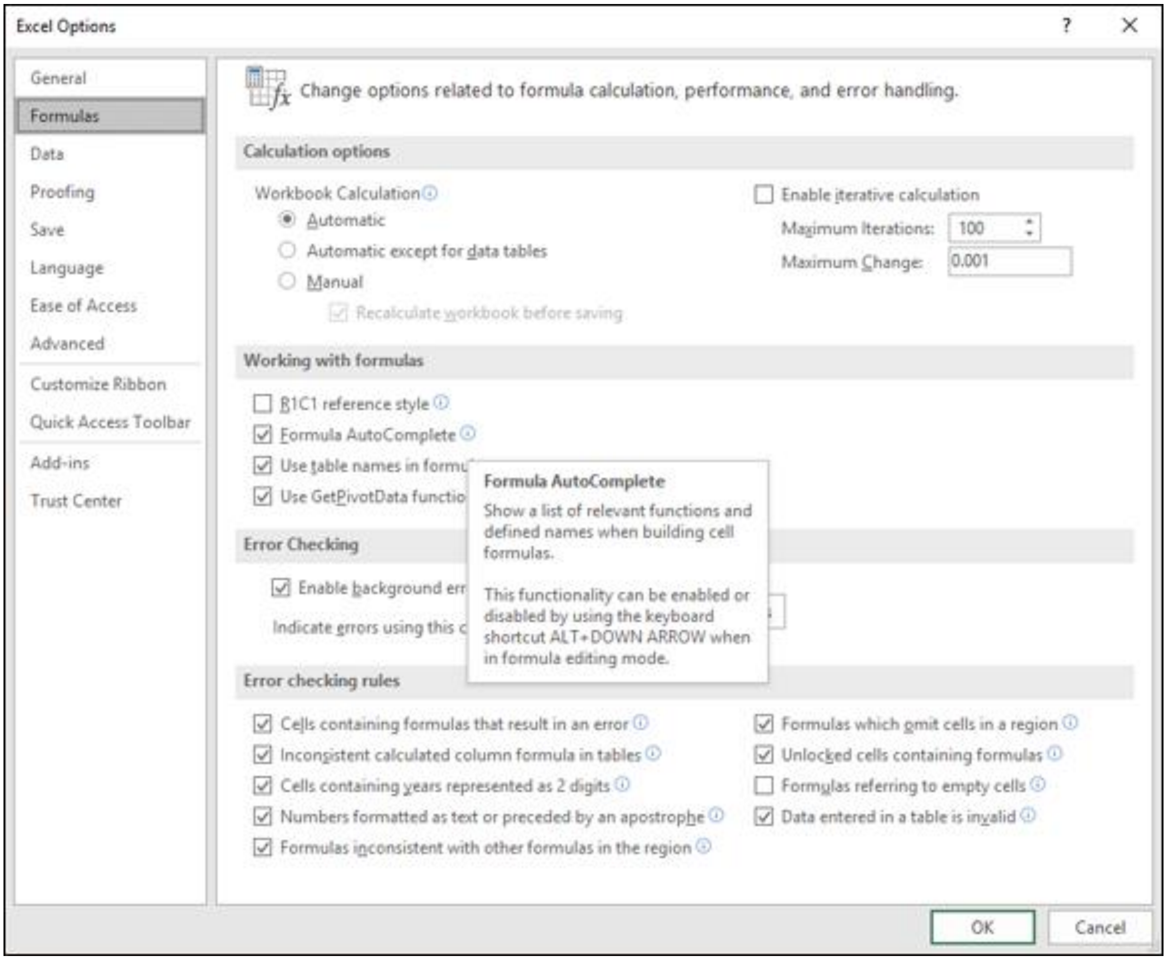


FIGURE 2-16: Setting Formula AutoComplete.

Using Basic Math Functions

Adding It All Together with the SUM Function

Just adding numbers together is something Excel is great at. Oh, you can use your calculator to add numbers as well, but think about it: On a calculator you enter a number, then press the + button, then enter another number, then press the + button, and so on. Eventually you press the = button, and you get your answer. But if you made an entry mistake in the middle, you have to start all over!

The SUM function in Excel adds numbers together in a more efficient way. First, you list all your numbers on the worksheet. You can see them all and verify that they're correct. Then you use the SUM function to add them all together. Here's how:

1. **Enter some numbers in a worksheet.**
   These numbers can be both integer and *real* (decimal) values. You can add labels to adjacent cells to identify the values, if you want.
2. **Position the cursor in the cell where you want the results to appear.**
3. **Type** =SUM (**to begin the function entry.**
4. **Click a cell where you entered a number.**
5. **Type a comma (,).**
6. **Click a cell where you entered another number.**

7. **Repeat steps 5 and 6 until all the numbers have been entered into the function.**
8. **Type a) and press Enter.**

Figure 7-1 shows an example of how these steps help sum up amounts that are not situated next to one another on a worksheet. Cell F6 contains the sum of values in cells C2, E2, G2, and I2.

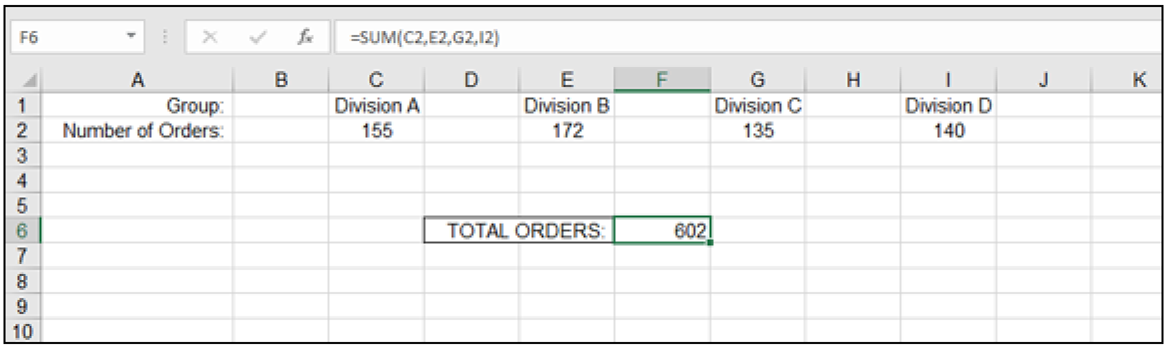


FIGURE 7-1: Using the SUM function to add noncontiguous numbers.

Using SUM is even easier when the numbers you're adding are next to one another in a column or row. The SUM function lets you enter a range of cells in place of single cells in the arguments of the function. So adding a list of contiguous numbers is as easy as giving SUM a single argument. Here's how you enter a range as a single argument:

1. **Enter some numbers in a worksheet.**
   Be sure the numbers are continuous in a row or column. You can add labels to adjacent cells to identify the values, if desired, but this doesn't affect the SUM function.
2. **Position the cursor in the cell where you want the results to appear.**
3. **Type** =SUM(**to begin the function entry.**
4. **Enter the range address that contains the numbers.**
   Alternatively, you can click the first cell with a number, hold down the left mouse button, and drag the mouse over the range of cells.

5. **Type a) and press Enter.**

Using a range address in the function is a real time saver — and is easier on the fingers, too. Figure 7-2 shows how a single range is used with the SUM function. Look at the Formula Bar, and you'll see that the entire function's syntax is `=SUM(B6:B12)`. A single range takes the place of multiple individual cell addresses.

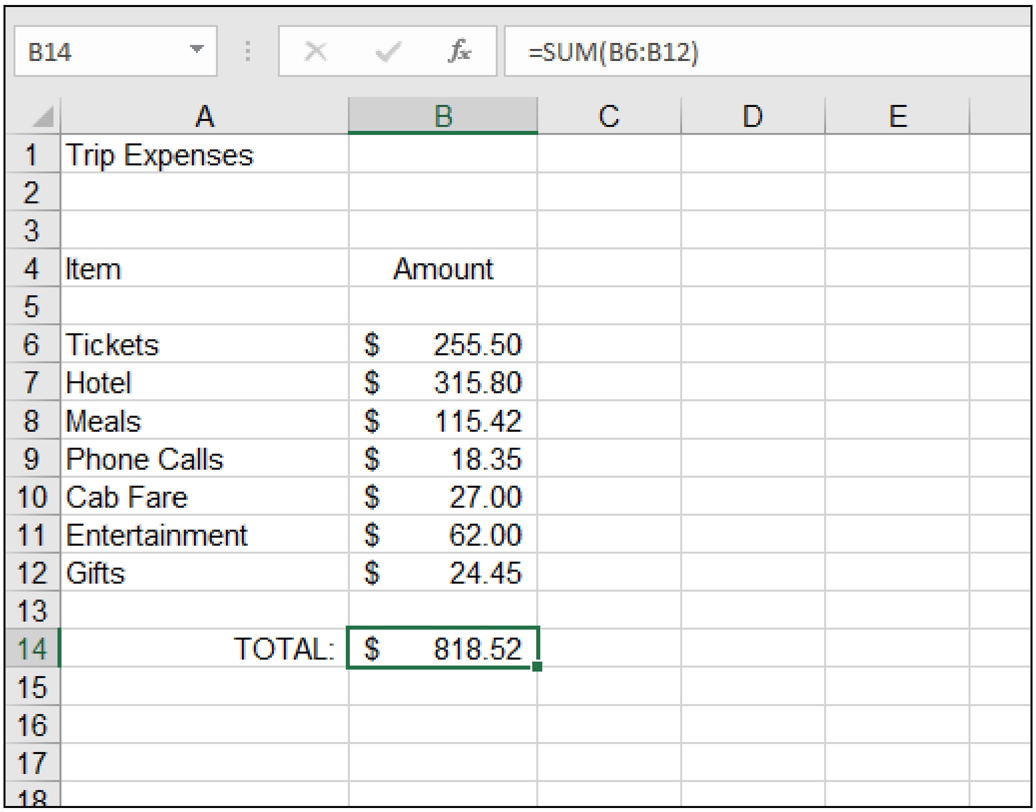
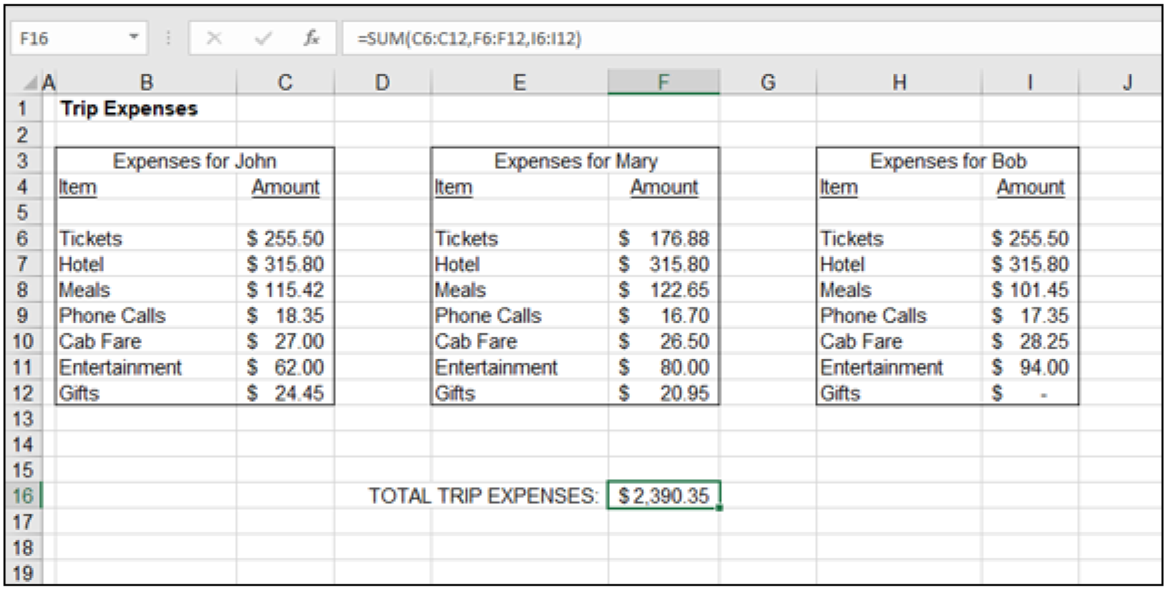


FIGURE 7-2: Calculating a sum from a range of cells.

You can sum multiple ranges in a single formula, which is great when multiple distinct contiguous cell ranges all must feed a grand total. Figure 7-3 shows just such a situation.


F16 =SUM(C6:C12,F6:F12,I6:I12)

	A	B	C	D	E	F	G	H	I	J
1		**Trip Expenses**								
2										
3		Expenses for John			Expenses for Mary			Expenses for Bob		
4		Item	Amount		Item	Amount		Item	Amount	
5										
6		Tickets	$ 255.50		Tickets	$ 176.88		Tickets	$ 255.50	
7		Hotel	$ 315.80		Hotel	$ 315.80		Hotel	$ 315.80	
8		Meals	$ 115.42		Meals	$ 122.65		Meals	$ 101.45	
9		Phone Calls	$ 18.35		Phone Calls	$ 16.70		Phone Calls	$ 17.35	
10		Cab Fare	$ 27.00		Cab Fare	$ 26.50		Cab Fare	$ 28.25	
11		Entertainment	$ 62.00		Entertainment	$ 80.00		Entertainment	$ 94.00	
12		Gifts	$ 24.45		Gifts	$ 20.95		Gifts	$ -	
13										
14										
15										
16					TOTAL TRIP EXPENSES:	$ 2,390.35				
17										
18										
19										

FIGURE 7-3: Calculating a sum of multiple ranges.

Here's how you use SUM to add the values in multiple ranges:

1. **Enter some lists of numbers in a worksheet.**
   You can add labels to adjacent cells to identify the values, if desired.
2. **Position the cursor in the cell where you want the results to appear.**
3. **Type** =SUM(**to begin the function entry.**
4. **Click the first cell in a range, hold down the left mouse button, drag the mouse over all the cells in the range, and then release the mouse button.**
5. **Type a comma (,).**
6. **Click the first cell in another range, hold down the left mouse button, drag the mouse over all the cells in this range, and then release the mouse button.**
7. **Repeat steps 5 and 6 until all the ranges have been entered into the function.**
8. **Type a) and press Enter.**

The completed function entry should look similar to the entry shown in the Formula Bar in Figure 7-3. Range are separated by commas, and a grand sum is in the cell where the function was entered.

TIP When entering ranges into a formula, you can either type them or use the mouse to drag over the range.

Excel has a special button, the AutoSum button, that makes it easier to use the SUM function. The AutoSum button is on both the Home tab and the Formulas tab of the Ribbon. The AutoSum feature works best with numbers that are in a vertical or horizontal list. In a nutshell, AutoSum creates a range reference for the SUM function to use. AutoSum makes its best guess about what the range should be. Often, it gets it right — but sometimes, you have to help it along.

Using AutoSum is as easy as clicking and then pressing Enter. Figure 7-4 shows that the AutoSum button on the Ribbon has been clicked, and Excel, in its infinite wisdom, guessed correctly that the operation is to sum cells B6:B13. At this point, the operation is incomplete. Pressing Enter finishes the formula.

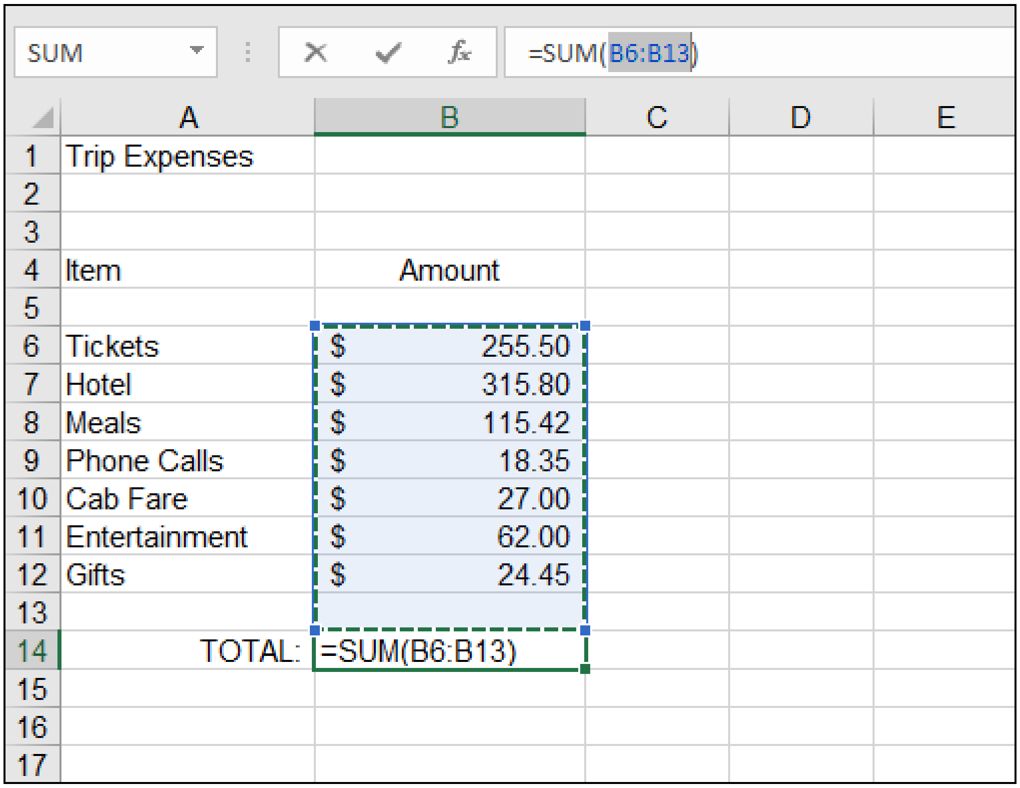


FIGURE 7-4: Using AutoSum to guess a range for the SUM function.

TIP

You can click the check mark to the left of the formula, in the Formula Bar, to complete the operation.

Follow these steps to use AutoSum:

1. **Enter some lists of numbers in a worksheet.**

   You can add labels to adjacent cells to identify the values, if desired.
2. **Position the cursor in the cell where you want the results to appear.**
3. **Click the AutoSum button.**

   AutoSum has entered a suggested range in the SUM function.
4. **Change the suggested range, if necessary, by entering it with the keyboard or using the mouse to drag over a range of cells.**

5. **Press Enter or click the check mark on the Formula Bar to complete the function.**

TIP It's easy to use AutoSum to tally multiple ranges, such as those shown in Figure 7-3. Before ending the function with the Enter key or the check mark in the Formula Bar, instead enter a comma and then drag the mouse over another range. Do this for as many ranges as you need to sum. Finally, finish the function by pressing Enter or clicking the check mark in the Formula Bar.

By the way, the AutoSum button can do more than addition. If you click the down arrow on the button, you have a choice of a few other key functions, such as Average (see Figure 7-5).

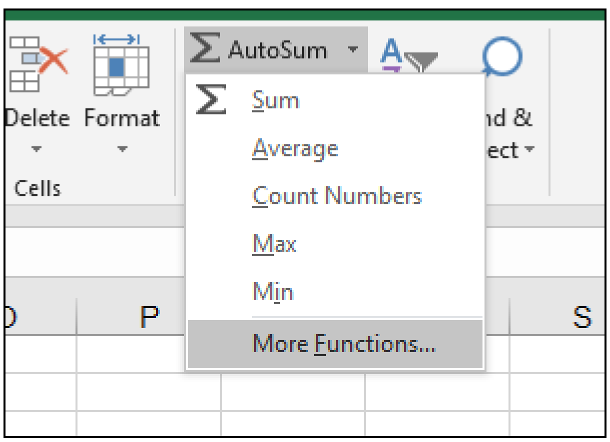


FIGURE 7-5: Using AutoSum to work with other popular functions.

Rounding Out Your Knowledge

Excel calculates answers to many decimal places. Unless you're doing rocket science, you probably don't need such precise answers. Excel has a great set of functions for rounding numbers so they're usable for the rest of us.

Excel's rounding functions are really helpful. The other day, my son had a couple of his friends over. I ordered a large pizza for their lunch. That's eight slices for three hungry boys. How many slices does each boy get? Presto magic, I went over to the computer where Excel was already running (okay, I am an Excel nut, after all), and I entered this simple formula: `=8/3`.

Of course, Excel gave me the perfect answer. Each boy gets 2.66667 slices. Have you ever tried to cut 66,667/100,000ths of a slice of pizza? Not easy! This is the type of answer that rounding is used for. To tell you the truth, I did solve the pizza problem a different way. I gave each boy two slices, and I ate the last two (pretty good with mushrooms!).

JUST PLAIN OLD ROUNDING

Easy to use, the ROUND function is the old tried-and-true method for rounding off a number. It takes two arguments. One argument is the number to round (typically, this is a cell reference), and the other argument indicates how many decimal places to round to.

The ROUND function rounds up or down, depending on the number being rounded. When the value is less than the halfway point of the next significant digit, the number is rounded down. When the value is at or greater than the halfway point, the number is rounded up, as follows:

- 10.4 rounds down to 10.
- 10.6 rounds up to 11.
- 10.5 also rounds up to 11.

Table 7-1 shows some examples of the ROUND function.

TABLE 7-1 Using the ROUND Function

Example of Function	*Result*	*Comment*
`=ROUND(12.3456, 1)`	12.3	The second argument is 1. The result is rounded to a single decimal place.
`=ROUND(12.3456, 2)`	12.35	The second argument is 2. The result is rounded to two decimal places. Note that the full decimal of .3456 becomes .35 because the .0456 portion of the decimal value rounds to the closest second-place decimal, which is .05.

`=ROUND(12.3456, 3)`	12.346	The second argument is 3. The result is rounded to three decimal places. Note that the full decimal or .3456 becomes .346 because the .0056 portion of the decimal value rounds to the closest third-place decimal, which is .006.
`=ROUND(12.3456, 4)`	12.3456	The second argument is 4. There are four decimal places. No rounding takes place.
`=ROUND(12.3456, 0)`	12	When the second argument is 0, the number is rounded to the nearest integer. Because 12.3456 is closer to 12 than to 13, the number rounds to 12.

`=ROUND(12.3456, -1)`	10	When negative values are used in the second argument, the rounding occurs on the left side of the decimal (the integer portion). A second argument value of –1 tells the function to round to the closest value of 10. In this example, that value is 10 because 12 is closer to 10 than to 20.

Here's how to use the ROUND function:

1. **In a cell of your choice, enter a number that has a decimal portion.**
2. **Position the cursor in the cell where you want the results to appear.**
3. **Type** =ROUND(**to begin the function entry.**
4. **Click the cell where you entered the number.**
5. **Type a comma (,).**
6. **Enter a number to indicate how many decimal places to round to.**
7. **Type a) and press Enter.**

REMEMBER Rounding functions make the most sense when the first argument is a cell reference, not an actual number. Think about it: If you know what a number should appear as, you would just enter the number. You would not need a function to round it.

ROUNDING IN ONE DIRECTION

Excel has a handful of functions that always round numbers either up or always down. That is, when Excel is rounding a number, the functions that round down always give a result that is lower than the number itself. Functions that round up, of course, always give a higher number. These functions are useful when letting the good ol’ ROUND function determine which way to round just isn’t going to do.

A few of these rounding functions not only round in the desired direction, but also allow you to specify some additional ways of rounding. The EVEN and ODD functions, for example, round to the closest even or odd number, respectively. The CEILING and FLOOR functions let you round to a multiple. EVEN, ODD, CEILING, and FLOOR are discussed later in this section.

Directional rounding, pure and simple

ROUNDUP and ROUNDDOWN are similar to the ROUND function. The first argument to the function is the cell reference of the number to be rounded. The second argument indicates the number of decimal places to round to. But unlike with plain old ROUND, the rounding direction is not based on the halfway point of the next significant digit, but on which function you use.

For example, `=ROUND(4.22,1)` returns 4.2, but `=ROUNDUP(4.22,1)` returns 4.3. ROUNDDOWN, however, returns 4.2 because 4.2 is less than 4.22. Table 7-2 shows some examples of ROUNDUP and ROUNDDOWN.

TABLE 7-2 Using the ROUNDUP and ROUNDDOWN Functions

Example of Function	*Result*	*Comment*
`=ROUNDUP(150.255,0)`	151	The second argument is 0. The result is rounded up to the next higher integer, regardless of the fact that the decimal portion would normally indicate the rounding would go to the next lower integer.
`=ROUNDUP(150.255,1)`	150.3	The second argument is 1. The result is rounded to a single decimal point. Note that the full decimal of .255 rounds up to .3. This would also happen with the standard ROUND function.

`=ROUNDUP(150.255,2)`	150.26	The second argument is 2. The result is rounded to two decimal places. Note that the full decimal of .255 becomes .26. This would also happen with the standard ROUND function.
`=ROUNDUP(150.255,3)`	150.255	The second argument is 3, and there are three decimal places. No rounding takes place.

`=ROUNDDOWN(155.798,0)`	155	The second argument is 0. The result is rounded down to the integer portion of the number, regardless of the fact that the decimal portion would normally indicate that the rounding would go to the next higher integer.
`=ROUNDDOWN(155.798,1)`	155.7	The second argument is 1. The result is rounded to a single decimal place. Note that the full decimal of .798 rounds down to .7. The standard ROUND function would round the decimal up to .8.

=ROUNDDOWN(155.798,2)	155.79	The second argument is 2. The result is rounded to two decimal places. Note that the full decimal of .798 becomes .79. The standard ROUND function would round the decimal up to .8.
=ROUNDDOWN(155.798,3)	155.798	The second argument is 3, and there are three decimal places. No rounding takes place.

Here's how to use the ROUNDUP and ROUNDDOWN functions:

1. **In a cell of your choice, enter a number with a decimal portion.**
2. **Position the cursor in the cell where you want the results to appear.**
3. **Type** =ROUNDUP(**or** =ROUNDDOWN(**to begin the function entry.**
4. **Click the cell where you entered the number.**
5. **Type a comma (,).**
6. **Enter a number to indicate how many decimal places to round to.**
7. **Type a) and press Enter.**

Rounding to the multiple of choice

The FLOOR and CEILING functions take directional rounding to a new level. With these functions, the second argument is a multiple to which to round to. What does that mean?

Well, imagine this: You're a human-resources manager, and you need to prepare a summary report of employee salaries. You don't need the figures to be reported down to the last penny — just rounded to the closest $250 multiple. Either FLOOR or CEILING can do this. For this example, FLOOR can be used to round down to the closest multiple of $250 that is less than the salary, or CEILING can be used to round up to the next $250 multiple greater than the salary. Figure 7-6 shows how FLOOR and CEILING return rounded values.

C6 =FLOOR(B6,250)

	A	B	C	D	E
1			Salary Rounded to Nearest 250	Salary Rounded to Nearest 250	
2			Using FLOOR	Using CEILING	
3					
4					
5	Employee ID	Salary			
6	W234	$ 54,677	$ 54,500	$ 54,750	
7	N552	$ 36,125	$ 36,000	$ 36,250	
8	P310	$ 28,900	$ 28,750	$ 29,000	
9	B533	$ 31,950	$ 31,750	$ 32,000	
10	R390	$ 48,305	$ 48,250	$ 48,500	
11	R418	$ 78,500	$ 78,500	$ 78,500	
12	W602	$ 60,252	$ 60,250	$ 60,500	
13	C177	$ 58,900	$ 58,750	$ 59,000	
14	T542	$ 36,550	$ 36,500	$ 36,750	
15	T833	$ 38,740	$ 38,500	$ 38,750	
16	M405	$ 52,580	$ 52,500	$ 52,750	
17					
18					

FIGURE 7-6: Using FLOOR or CEILING to round to a desired multiple.

FLOOR and CEILING exceed the rounding ability of ROUND, ROUNDUP, and ROUNDDOWN. These three functions can use the positioning of digit placeholders in how they work. For example, `=ROUND(B4,-3)` tells the ROUND function to round on the thousandth position. On the other hand, FLOOR and CEILING can round to whatever specific multiple you set.

The FLOOR function rounds toward 0, returning the closest multiple of the second argument that is lower than the number itself.

The CEILING function works in the opposite direction. CEILING will round its first argument, the number to be rounded, to the next multiple of the second number that is in the direction away from 0.

Certainly, a few examples will make this clear! Table 7-3 shows ways that FLOOR and CEILING can be used.

TABLE 7-3 Using FLOOR and CEILING for Sophisticated Rounding

Example of Function	*Result*	*Comment*
`=FLOOR(30.17,0.05)`	30.15	The second argument says to round to the next 0.05 multiple, in the direction of 0.
`=FLOOR(30.17,0.1)`	30.1	The second argument says to round to the next 0.1 multiple, in the direction of 0.

`=FLOOR(-30.17,-0.1)`	–30.1	The second argument says to round to the next 0.1 multiple, in the direction of 0.
`=CEILING(30.17,0.05)`	30.2	The second argument says to round to the next 0.05 multiple, away from 0.
`=CEILING(30.17,0.1)`	30.2	The second argument says to round to the next 0.1 multiple, away from 0.
`=CEILING(-30.17,-0.1)`	–30.2	The second argument says to round to the next 0.1 multiple, away from 0.

FLOOR and CEILING can be used to round negative numbers. FLOOR rounds toward 0, and CEILING rounds away from 0. FLOOR decreases a positive number as it rounds it toward 0 and also decreases a negative number toward 0, although in absolute terms, FLOOR actually increases the value of a negative number. Weird, huh?

CEILING does the opposite. It increases a positive number away from 0 and also increases a negative number away from 0, which in absolute terms means the number is getting smaller.

WARNING For both the FLOOR and CEILING functions, the first and second arguments must match signs. Trying to apply a positive number with a negative multiple, or vice versa, results in an error.

Here's how to use the FLOOR and CEILING functions:

1. **Enter a number in any cell.**
2. **Position the cursor in the cell where you want the results to appear.**
3. **Type** =FLOOR(**or** =CEILING(**to begin the function entry.**
4. **Click the cell where you entered the number.**
5. **Type a comma (,).**
6. **Enter a number that is the next multiple you want to round the number to.**

   For example, to get the floor value, at the ones place, make sure 1 is the second argument. The first argument should, of course, be a number larger than 1 and should be a decimal value, like this: `=Floor(19.77, 1)`. This returns 19 as the floor, but hey — don’t hit the ceiling about it!
7. **Type a) and press Enter.**

Rounding to the next even or odd number

The EVEN and ODD functions round numbers away from 0. The EVEN function rounds a number to the next highest even integer. ODD rounds a number to the next highest odd integer. Table 7-4 has examples of how these functions work.

TABLE 7-4 Rounding to Even or Odd Integers

Example of Function	*Result*	*Comment*

=EVEN(3)	4	Rounds to the next even integer, moving away from 0.
=EVEN(4)	4	Because 4 is an even number, no rounding takes place. The number 4 itself is returned.
=EVEN(4.01)	6	Rounds to the next even integer, moving away from 0.
=EVEN(–3.5)	–4	Rounds to the next even integer, moving away from 0.
=ODD(3)	3	Because 3 is an odd number, no rounding takes place. The number 3 itself is returned.
=ODD(4)	5	Rounds to the next odd integer, moving away from 0.
=ODD(5.01)	7	Rounds to the next odd integer, moving away from 0.

`=ODD(-3.5)`	–5	Rounds to the next odd integer, moving away from 0.

The EVEN function is helpful in calculations that depend on multiples of two. Say you're in charge of planning a school trip. You need to figure out how many bus seats are needed for each class. A seat can fit two children. When a class has an odd number of children, you still have to count that last seat as taken, even though only one child will sit there.

Say the class has 17 children. This formula tells you how many seats are needed: `=EVEN(17)/2`. The EVEN function returns the number 18 (the next higher integer), and that result is divided by 2 because 2 children fit on each seat. The answer is 9 seats are needed for a class of 17.

Here’s how to use the EVEN and ODD functions:

1. **Position the cursor in the cell where you want the results to appear.**
2. **Type** =EVEN(**or** =ODD(**to begin the function entry.**
3. **Click a cell where you entered a number, or enter a number.**
4. **Type a**) **and press Enter.**

Leaving All Decimals Behind with INT

The INT function rounds a number down to the next lowest integer. The effect is as if the decimal portion is just dropped, and often, INT is used to facilitate just that: dropping the decimal.

INT comes in handy when all you need to know is the integer part of a number or the integer part of a calculation's result. For example, you may be estimating what it will cost to build a piece of furniture. You have the prices for each type of raw material, and you just want a ballpark total.

Figure 7-7 shows a worksheet in which a project has been set up. Column A contains item descriptions, and column B has the price for each item. Columns C and D contain the parameters for the project. That is, column C contains the count of each item needed, and column D has the amount to be spent for each item — that is, the price per item multiplied by the number of items needed.

D8 =INT(SUM(D3:D5))

	A	B	C	D	E
1	Item	Price per Unit	Units Needed	Cost	
2					
3	Lumber	$ 6.99	12	$ 83.88	
4	Hinges	$ 4.49	24	$ 107.76	
5	Knobs	$ 4.99	4	$ 19.96	
6					
7					
8			Project Cost	$ 211.00	
9					
10					

FIGURE 7-7: Using INT to drop unnecessary decimals.

The sums to be spent are then summed into a project total. If you added the item sums as they are — 83.88, 107.76, and 19.96 — you get a total of $211.60. Instead, the INT function is used to round the total to a ballpark figure of $211.

In cell D8, INT is applied to the total sum, like this:

```
=INT(SUM(D3:D5))
```

The INT function effectively drops the decimal portion, .60, and returns the integer part, 211. The project estimate is $211.

INT takes only the number as an argument. INT can work on positive or negative values but works a little differently with negative numbers. INT actually rounds down a number to the next lower integer. When INT is working with positive numbers, the effect appears the same as just dropping the decimal. With negative numbers, the function drops the decimal portion and subtracts 1.

With negative numbers, the function produces an integer that is farther away from 0. Therefore, a number such as –25.25 becomes –26. Here are some examples:

- `INT(25.25)` returns 25.
- `INT(25.75)` returns 25.
- `INT(-25.25)` returns –26.
- `INT(-25.75)` returns –26.

Here's how to use the INT function:

1. **In a cell of your choice, enter a number that has a decimal portion.**
2. **Position the cursor in the cell where you want the results to appear.**
3. **Type** =INT(**to begin the function entry.**
4. **Click the cell where you entered the number.**
5. **Enter a closing parenthesis to end the function and press Enter.**

TIP INT can also be used to return just the decimal part of a number. Subtracting the integer portion of a number from its full value leaves just the decimal as the answer. For example, `10.95-INT(10.95)` equals 0.95.

Leaving Some Decimals Behind with TRUNC

The TRUNC function drops a part of a number. The function takes two arguments. The first argument is the number to be changed. The second argument indicates how much of the number is to be dropped. A value of 2 for the second argument says to leave 2 decimal places remaining. A value of 1 for the second argument says to leave 1 decimal place remaining.

TRUNC does no rounding as it truncates numbers. Here are some examples:

- `=TRUNC(212.65, 2)` returns 212.65.
- `=TRUNC(212.65, 1)` returns 212.6.
- `=TRUNC(212.65, 0)` returns 212.

You can even use TRUNC to drop a portion of the number from the integer side. To do this, you enter negative values for the second argument, like this:

- `=TRUNC(212.65, -1)` returns 210.
- `=TRUNC(212.65, -2)` returns 200.

TIP The INT and TRUNC functions work exactly the same way for positive numbers. The only difference is when negative numbers are being changed. Then INT's rounding produces a different result than TRUNC’s truncation.

Looking for a Sign

Excel's SIGN function tells you whether a number is positive or negative. The SIGN function does not alter the number in any way but is used to find out information about the number.

SIGN does actually return a number, but it isn't a variation of the number being tested in the function. SIGN returns only three numbers:

- 1 if the number being tested is positive
- –1 if the number being tested is negative
- 0 if the number being tested is 0

Consider these examples:

- `=SIGN(5)` returns 1.
- `=SIGN(–5)` returns –1.
- `=SIGN(0)` returns 0.

Using SIGN in combination with other functions presents sophisticated ways of working with your information. As an example, you may be tallying up a day's receipts from your store. You want to know the total value of sold merchandise and the total value of returned merchandise. Sales are recorded as positive amounts, and returns are recorded as negative amounts.

Figure 7-8 shows a worksheet with these facts. Column A shows individual transaction amounts. Most amounts are sales and are positive. A few returns occurred during the day, entered as negative amounts.

B18 =SUMIF(B3:B15,1,A3:A15)

	A	B	C	D	E	F
1	**Daily Reciepts**					
2						
3	$ 34.95	1				
4	$ 24.82	1				
5	$ 90.63	1				
6	$ 44.50	1				
7	$ (17.24)	-1				
8	$ 12.00	1				
9	$ 25.90	1				
10	$ 28.99	1				
11	$ (30.15)	-1				
12	$ 115.99	1				
13	$ 104.10	1				
14	$ (16.79)	-1				
15	$ 32.35	1				
16						
17						
18	Sales:	$ 514.23				
19	Returns:	$ 64.18				
20						

FIGURE 7-8: Using SIGN to sum amounts correctly.

Just summing the whole transaction list would calculate the net revenue of the day, but often, a business needs better information. Instead, two sums are calculated: the sum of sales and the sum of returns.

For each value in column A, there is a value in column B. The column B values are the result of using the SIGN function. For example, cell B3 has this formula: `=SIGN(A3)`.

As shown in Figure 7-8, values in column B equal 1 when the associated value in column A is positive. Column B displays –1 when the associated value is negative. This information is then used in a SUMIF function, which selectively sums information from column A.

In cell B18 is this formula: `=SUMIF(B3:B15,1,A3:A15)`.

In cell B19 is this formula: `=ABS(SUMIF(B3:B15,-1,A3:A15))`.

The SUMIF function is used to indicate a criterion to use in determining which values to sum. For the sum of sales in cell B18, the presence of the value 1 in column B determines which values to sum in column A. For the sum of returns in cell B19, the presence of the value –1 in column B determines which values to sum in column A.

Also, the ABSOLUTE function (ABS) is used to present the number in cell B19 as a positive number. The answer in cell B19 is the sum of merchandise returns. You would say there was $64.18 (not –$64.18) in returned merchandise, if you were asked.

The SUMIF function is covered in Chapter 8. The ABS function is covered next in this chapter.

Here's how to use the SIGN function:

1. **Position the cursor in the cell where you want the results to appear.**
2. **Type** =SIGN(**to begin the function entry.**
3. **Click a cell where you entered a number, or enter a number.**
4. **Type a**) **and press Enter.**

Ignoring Signs

The ABS function returns the absolute value of a number. The absolute number is always a positive. The absolute of a positive number is the number itself. The absolute of a negative number is the number but with the sign changed to positive. For example, `=ABS(100)` returns 100, as does `=ABS(-100)`.

The ABS function is handy in a number of situations. For example, sometimes imported data comes in as negative values, which need to be converted to their positive equivalents. Or, when you're working with cash flows as discussed in Chapter 3, you can use the ABS function to present cash flows as positive numbers.

A common use of the ABS function is to calculate the difference between two numbers when you don’t know which number has the greater value to begin with. Say you need to calculate the difference between scores for two contestants. Score 1 is in cell A5, and score 2 is in cell B5. The result goes in cell C5. The formula in cell C5 would be `=A5-B5`.

Plugging in some numbers, assume that score 1 is 90 and score 2 is 75. The difference is 15. Okay, that’s a good answer. What happens when score 1 is 75 and score 2 is 90? The answer is –15. This answer is mathematically correct but not presented in a useful way. The difference is still 15, not –15. When you use the ABS function, the result is always returned as positive. Therefore, for this example, the best formula coding is this: `=ABS(A5-A6)`.

Now, whether score 1 is greater than score 2 or score 2 is greater than score 1, the correct difference is returned.

Here’s how to use the ABS function:

1. **Position the cursor in the cell where you want the results to appear.**
2. **Type** =ABS(**to begin the function entry.**
3. **Click a cell where you entered a number, or enter a number.**
4. **Type a**) **and press Enter.**

Using PI to Calculate Circumference and Diameter

Pi is the ratio of a circle's circumference to its diameter. A circle's *circumference* is its outer edge and is equal to the complete distance around the circle. A circle's *diameter* is the length of a straight line running from one side of the circle, through the middle, and reaching the other side.

Dividing a circle's circumference by its diameter returns a value of approximately 3.14159, known as *pi*. Pi is represented with the Greek letter pi and the symbol π.

Mathematicians have proved that pi is an *irrational number* — in other words, that it has an infinite number of decimal places. They have calculated the value of pi to many thousands of decimal places, but you don't need that level of precision in most calculations. Many people use the value 3.14159 for pi, but the PI function in Excel does a bit better than that. Excel returns a value of pi accurate to 15 digits — that is 14 decimal places in addition to the integer 3. This function has no input arguments. The function uses this syntax:

```
=PI()
```

TIP In Excel, the PI function always returns 3.14159265358979, but initially, it may look like some of the digits after the decimal point are missing. Change the formatting of the cell to display numbers with 14 decimal places to see the entire number.

If you know the circumference of a circle, you can calculate its diameter with this formula:

diameter = circumference ÷ pi

If you know the diameter of a circle, you can calculate its circumference with this formula:

circumference = diameter × pi

If you know the diameter of a circle, you can calculate the area of the circle. A component of this calculation is the *radius*, which equals one half of the diameter. The formula is

area = (diameter × 0.5)^2 × pi

Generating and Using Random Numbers

Random numbers are, by definition, unpredictable. That is, given a series of random numbers, you can't predict the next number from what has come before. Random numbers are quite useful for trying formulas and calculations. Suppose that you're creating a worksheet to perform various kinds of data analysis. You may not have any real data yet, but you can generate random numbers to test the formulas and charts in the worksheet.

For example, an actuary may want to test some calculations based on a distribution of people's ages. Random numbers between 18 and 65 can be used for this task. You don't have to manually enter fixed values between 18 and 65, because Excel can generate them automatically via the RAND function.

THE ALL-PURPOSE RAND FUNCTION

The RAND function is simple; it takes no arguments and returns a decimal value between 0 and 1. That is, RAND never actually returns 0 or 1; the value is always in between these two numbers. The function is entered like this:

```
=RAND()
```

The RAND function returns values such as 0.136852731, 0.856104058, or 0.009277161. "Yikes!" you may be thinking. "How do these numbers help if you need values between 18 and 65?" Actually, it's easy with a little extra math.

There is a standard calculation for generating random numbers within a determined range. The calculation follows:

```
= RAND() * (high number - low number) + low number
```

Using 18 and 65 as a desired range of numbers, the formula looks like `=RAND()*(65-18)+18`. Some sample values returned with this formula follow:

- 51.71777896
- 27.20727871
- 24.61657068
- 55.27298686
- 49.93632709
- 43.60069745

Almost usable! But what about the long decimal portions of these numbers? Some people lie about their ages, but I've never heard someone say he’s 27.2 years old!

All that is needed now for this 18-to-65 age example is to include the INT or ROUND function. INT simply discards the decimal portion of a number. ROUND allows control of how to handle the decimal portion.

The syntax for using the INT function with the RAND function follows:

```
= INT((high number – low number + 1) * RAND() + low
number)
```

The syntax for using the ROUND function with the RAND function follows:

```
=ROUND(RAND() * (high number-low number) + low
number,0)
```

Try it yourself! Here’s how to use RAND and INT together:

1. **Position the pointer in the cell where you want the results displayed.**
2. **Type** =INT((**to begin the formula.**

3. **Click the cell that has the highest number to be used, or enter such a value.**
4. **Type – (a minus sign).**
5. **Click the cell that has the lowest number to be used, or enter such a value.**
6. **Type** +1) * RAND( ) + **.**
7. **Again, click the cell that has the lowest number to be used, or enter the value.**
8. **Type a) and press Enter.**

A random number, somewhere in the range between the low and high number, is returned.

Table 8-1 shows how returned random numbers can be altered with the INT and ROUND functions.

TABLE 8-1 Using INT and ROUND to Process Random Values

Value	*Value Returned with INT*	*Value Returned with ROUND*
51.71777896	51	52
27.20727871	27	27
24.61657068	24	25

55.27298686	55	55
49.93632709	49	50
43.60069745	43	44

Table 8-1 points out how the INT and ROUND functions return different numbers. For example, 51.71777896 is more accurately rounded to 52. Bear in mind that the second argument in the ROUND function, 0 in this case, has an effect on how the rounding works. A 0 tells the ROUND function to round the number to the nearest integer, up or down to whichever integer is closest to the number.

WARNING Random values are volatile. Each time a worksheet is recalculated, the random values change. You can prevent this behavior by typing the formula directly in the Formula Bar, pressing the F9 key, and then pressing Enter.

A last but not insignificant note about using the RAND function: It is subject to the recalculation feature built into worksheets. In other words, each time the worksheet calculates, the RAND function is rerun and returns a new random number. The calculation setting in your worksheet is probably set to automatic. You can check this by looking at the Formulas tab of the Excel Options dialog box. Figure 8-1 shows the calculation setting. On a setting of Automatic, the worksheet recalculates with every action. The random generated numbers keep changing, which can become quite annoying if this is not what you intended to have happen. However, I bet you did want the number to change; otherwise, why use something “random” in the first place?

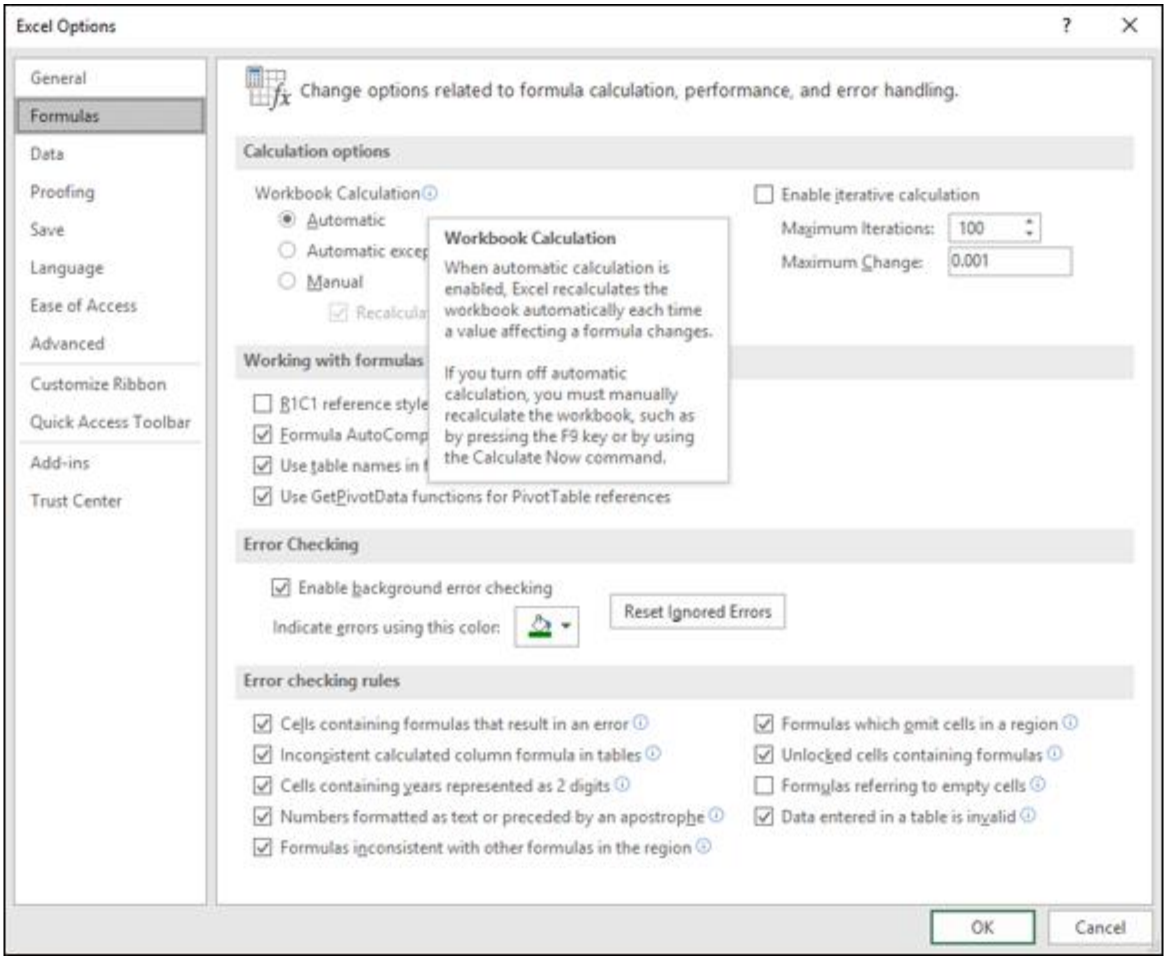


FIGURE 8-1: Setting worksheet calculation options.

REMEMBER Luckily, you can generate a random number but have it remain fixed regardless of the calculation setting. The method is to type the RAND function, along with any other parts of a larger formula, directly in the Formula Bar. After you type your formula, press the F9 key and then press Enter. This tells Excel to calculate the formula and enter the returned random number as a fixed number instead of a formula. If you press Enter or finish the entry in some way without pressing the F9 key, you have to enter it again.

PRECISE RANDOMNESS WITH RANDBETWEEN

Using the RAND function returns a value between 0 and 1, and when you use it with other functions, such as ROUND, you can get a random number within a range that you specify. If you just need a quick way to get an integer (no decimal portion!) within a given range, use RANDBETWEEN.

The RANDBETWEEN function takes two arguments: the low and high numbers of the desired range. It works only with integers. You can put real numbers in the range, but the result will still be an integer.

To use RANDBETWEEN, follow these steps:

1. **Position the pointer in the cell where you want the results displayed.**
2. **Type** =RANDBETWEEN(**to begin the formula.**
3. **Click the cell that has the low number of the desired range, or enter such a value.**
4. **Type a comma (,).**
5. **Click the cell that has the highest number of the desired range, or enter such a value.**
6. **Type a**) **and press Enter.**

For example, =RANDBETWEEN(10,20) returns a random integer between 10 and 20.

Ordering Items

Remember the Beatles? John, Paul, George, and Ringo? If you’re a drummer, you may think of the Beatles as Ringo, John, Paul, and George. The order of items in a list is known as a *permutation.* The more items in a list, the more possible permutations exist.

Excel provides the PERMUT function. It takes two arguments: the total number of items to choose among and the number of items to be used in determining the permutations. The function returns a single whole number. The syntax of the function follows:

```
=PERMUT(total number of items, number of items to
use)
```

TIP Use permutations when the order of items is important.

WARNING The total number of items must be the same as or greater than the number of items to use; otherwise, an error is generated.

You may be confused about why the function takes two arguments. On the surface, it seems that the first argument is sufficient. Well, not quite. Getting back to the Beatles, anyone have a copy of *Abbey Road* I can borrow? If we plug in 4 as the number for both arguments

```
=PERMUT(4,4)
```

Twenty-four permutations are returned:

- John Paul George Ringo
- John Paul Ringo George
- John George Paul Ringo
- John George Ringo Paul
- John Ringo Paul George
- John Ringo George Paul
- Paul John George Ringo
- Paul John Ringo George

- Paul George John Ringo
- Paul George Ringo John
- Paul Ringo John George
- Paul Ringo George John
- George John Paul Ringo
- George John Ringo Paul
- George Paul John Ringo
- George Paul Ringo John
- George Ringo John Paul
- George Ringo Paul John
- Ringo John Paul George
- Ringo John George Paul
- Ringo Paul John George
- Ringo Paul George John
- Ringo George John Paul
- Ringo George Paul John

Altering the function to use 2 items at a time from the total of 4 items — `PERMUT(4,2)` — returns just 12 permutations:

- John Paul
- John George
- John Ringo
- Paul John
- Paul George
- Paul Ringo
- George John
- George Paul
- George Ringo
- Ringo John
- Ringo Paul
- Ringo George

Just for contrast, using the number 2 for both arguments — `PERMUT(2,2)` — returns just two items! When using PERMUT, make sure you've selected the correct numbers for the two arguments; otherwise, you'll end up with an incorrect result and may not be aware of the mistake. The PERMUT function simply returns a number. The validity of the number is in your hands.

Combining

Combinations are similar to permutations but with a distinct difference. The order of items is intrinsic to permutations. Combinations, however, are groupings of items in which the order doesn't matter. For example, "John Paul George Ringo" and "Ringo George Paul John" are two distinct permutations but identical combinations.

TIP *Combinations* are groupings of items, regardless of the order of the items.

The syntax of the function follows:

```
=COMBIN(total number of items, number of items to use)
```

The first argument is the total number of items to choose among, and the second argument is the number of items to be used in determining the combinations. The function returns a single whole number. The arguments for the COMBIN function are the same as those for the PERMUT function. The first argument must be equal to or greater than the second argument.

Plugging in the number 4 for both arguments — `COMBIN(4,4)` — returns 1. Yes, there is just one combination of four items selected from a total of four items! Using the Beatles once again, just one combination of the four musicians exists, because the order of names doesn't matter.

Selecting to use two items from a total of four — `COMBIN(4,2)` — returns 6. Selecting two items out of two — `COMBIN(2,2)` — returns 1. In fact, whenever the two arguments to the COMBIN function are the same, the result is always 1.

Raising Numbers to New Heights

There is an old tale about a king who loved chess so much, he decided to reward the inventor of chess by granting any request he had. The inventor asked for a grain of wheat for the first square of the chessboard on Monday, two grains for the second square on Tuesday, four for the third square on Wednesday, eight for the fourth square on Thursday, and so on, each day doubling the amount until the 64th square was filled with wheat. The king thought this was a silly request. The inventor could have asked for riches!

What happened was that the kingdom quickly ran out of wheat. By the 15th day, the number equaled 16,384. By the 20th day, the number was 524,288. On the 64th day, the number would have been an astonishing 9,223,372,036,854,780,000, but the kingdom had run out of wheat at least a couple of weeks earlier!

This "powerful" math is literally known as raising a number to a power. The *power*, in this case, means how many times a number is to be multiplied by itself. The notation is typically a superscript (2^3 for example). Another common way of noting the use of a power is with the caret symbol: 2^3. The verbiage for this is *two to the third power, or two to the power of three.*

In the chess example, 2 is raised to a higher power each day. Table 8-2 shows the first 10 days.

TABLE 8-2 The Power of Raising Numbers to a Power

Day	***Power That 2 Is Raised To***	***Power Notation***	***Basic Math Notation***	***Result***
1	0	2^0	1	1
2	1	2^1	2	2
3	2	2^2	2 × 2	4
4	3	2^3	2 × 2 × 2	8
5	4	2^4	2 × 2 × 2 ×2	16
6	5	2^5	2 × 2 × 2 × 2 × 2	32
7	6	2^6	2 × 2 × 2 × 2 × 2 × 2	64
8	7	2^7	2 × 2 × 2 × 2 × 2 × 2 × 2	128

9	8	2^8	2 × 2 × 2 × 2 × 2 × 2 × 2 × 2	256
10	9	2^9	2 × 2 × 2 × 2 × 2 × 2 × 2 × 2 × 2	512

The concept is easy enough. Each time the power is incremented by 1, the result doubles. Note that the first entry raises 2 to the 0 power. Isn't that strange? Well, not really. Any number raised to the 0 power = 1. Also note that any number raised to the power of 1 equals the number itself.

Excel provides the POWER function, whose syntax follows:

```
=POWER(number, power)
```

Both the number and power arguments can be integer or real numbers, and negative numbers are allowed.

TIP In a worksheet, either the POWER function or the caret can be used. For example, in a cell you can enter `=POWER(4,3)`, or `=4^3`. The result is the same either way. You insert the caret by holding Shift and pressing the number 6 key on the keyboard.

Multiplying Multiple Numbers

The PRODUCT function is useful for multiplying up to 255 numbers at a time. The syntax follows:

```
=PRODUCT (number1, number2,…)
```

Cell references can be included in the argument list, as well as actual numbers, and of course, they can be mixed. Therefore, all these variations work:

```
=PRODUCT(A2, B15, C20)
=PRODUCT(5, 8, 22)
=PRODUCT(A10, 5, B9)
```

In fact, you can use arrays of numbers as the arguments. In this case, the notation looks like this:

```
=PRODUCT(B85:B88,C85:C88, D86:D88)
```

Here's how to use the PRODUCT function:

1. **Enter some values in a worksheet.**
   You can include many values, going down columns or across in rows.
2. **Position the pointer in the cell where you want the results displayed.**
3. **Type** =PRODUCT(**to begin the function.**
4. **Click a cell that has a number.**
   Alternatively, you can hold down the left mouse button and drag the pointer over a range of cells with numbers.
5. **Type a comma (,).**
6. **Repeat steps 4 and 5 up to 254 times.**
7. **Type a) and press Enter.**

The result you see is calculated by multiplying all the numbers you selected. Your fingers would probably hurt if you had done this on a calculator.

Figure 8-2 shows this on a worksheet. Cell C10 shows the result of multiplying 12 numbers, although only three arguments, as ranges, have been used in the function.

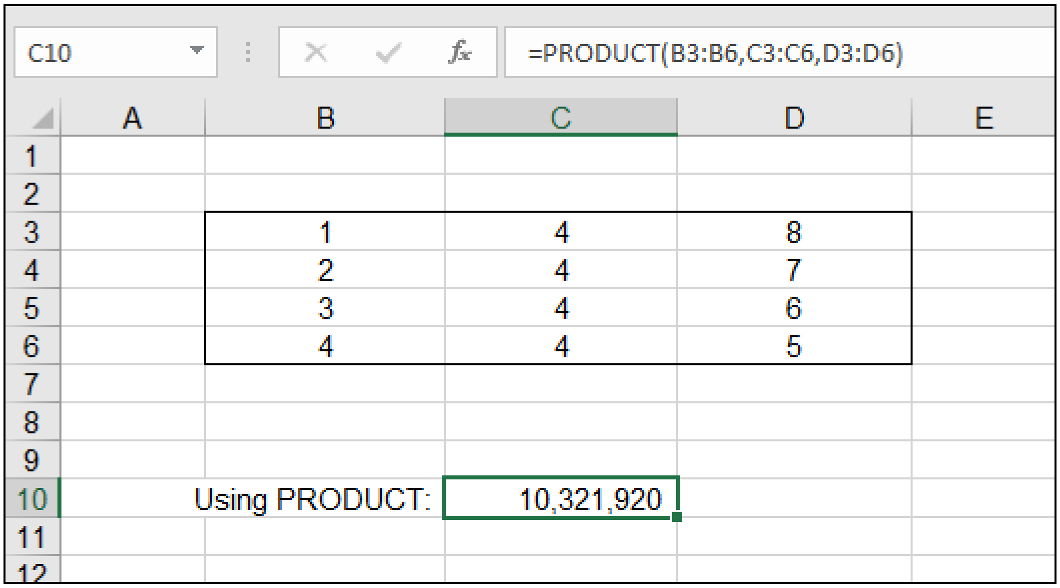


FIGURE 8-2: Putting the PRODUCT function to work.

Using What Remains with the MOD Function

The MOD function returns the remainder from an integer division operation. This remainder is called the *modulus*, hence the function's name. The function has two arguments: the number being divided and the number being used to divide the first argument. The second argument is the divisor. The syntax follows:

```
=MOD(number, divisor)
```

These are examples of the MOD function:

- `=MOD(12,6)` returns 0.
- `=MOD(14,5)` returns 4.
- `=MOD(27,7)` returns 6.
- `=MOD(25,10)` returns 5.
- `=MOD(25,-10)` returns –5.

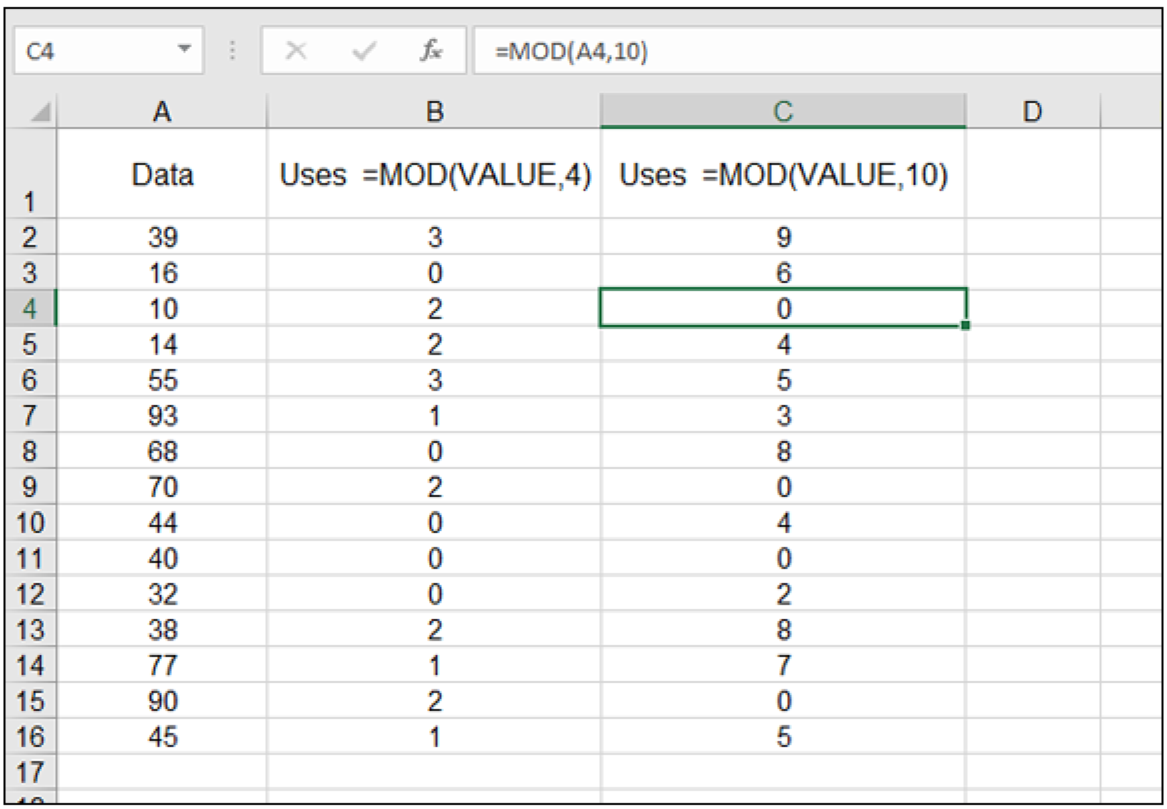
- `=MOD(15.675,8.25)` returns 7.425.

The returned value is always the same sign as the divisor.

You can use MOD to tell whether a number is odd or even. If you simply use a number 2 as the second argument, the returned value will be 0 if the first argument is an even number and 1 if it is not.

But what's so great about that? You can just look at a number and tell whether it’s odd or even. The power of the MOD function is apparent when you’re testing a reference or formula, such as `=MOD(D12 – G15,2)`. In a complex worksheet with many formulas, you may not be able to tell when a cell will contain an odd or even number.

Taking this a step further, the MOD function can be used to identify cells in a worksheet that are multiples of the divisor. Figure 8-3 shows how this works.


C4 =MOD(A4,10)

	A	B	C	D
1	Data	Uses =MOD(VALUE,4)	Uses =MOD(VALUE,10)	
2	39	3	9	
3	16	0	6	
4	10	2	0	
5	14	2	4	
6	55	3	5	
7	93	1	3	
8	68	0	8	
9	70	2	0	
10	44	0	4	
11	40	0	0	
12	32	0	2	
13	38	2	8	
14	77	1	7	
15	90	2	0	
16	45	1	5	
17				

FIGURE 8-3: Using MOD to find specific values.

Row 1 of the worksheet in Figure 8-3 shows examples of the formulas that are entered in the successive rows of columns B and C, starting from the second row. Column A contains numbers that will be tested with the MOD function. If you're looking for multiples of 4, the MOD function has 4 as the divisor, and when a value is a multiple of 4, MOD returns 0. This is evident when you compare the numbers in column A with the returned values in column B.

The same approach is used in column C, only here the divisor is 10, so multiples of 10 are being tested for in column A. Where a 0 appears in column C, the associated number in column A is a multiple of 10.

In this way, you can use the MOD function to find meaningful values in a worksheet.

Summing Things Up

Aha! Just when you think you know how to sum up numbers (really, haven't you been doing this since your early school years?), I present a fancy-footwork summing that makes you think twice before going for that quick total.

The functions here are very cool — very "in" with the math crowd. To be a true Excel guru, try the SUBTOTAL, SUMPRODUCT, SUMIF, and SUMIFS functions shown here and then strut your stuff around the office!

USING SUBTOTAL

The SUBTOTAL function is very flexible. It doesn't perform just one calculation; it can do any of 11 calculations depending on what you need. What's more, SUBTOTAL can perform these calculations on up to 255 ranges of numbers. This gives you the ability to get exactly the type of summary you need without creating a complex set of formulas. The syntax of the function follows:

```
=SUBTOTAL(function number, range1, range2,…)
```

The first argument determines which calculation is performed. It can be any of the values shown in Table 8-3. The remaining arguments identify the ranges containing the numbers to be used in the calculation.

TABLE 8-3 **Argument Values for the SUBTOTAL Function**

Function Number for First Argument	***Function***	***Description***
1	AVERAGE	Returns the average value of a group of numbers
2	COUNT	Returns the count of cells that contain numbers and also numbers within the list of arguments
3	COUNTA	Returns the count of cells that are not empty and only nonempty values within the list of arguments
4	MAX	Returns the maximum value in a group of numbers
5	MIN	Returns the minimum value in a group of numbers

6	PROD UCT	Returns the product of a group of numbers
7	STDEV .S	Returns the standard deviation from a sample of values
8	STDEV .P	Returns the standard deviation from an entire population, including text and logical values
9	SUM	Returns the sum of a group of numbers
10	VAR.S	Returns variance based on a sample
11	VAR.P	Returns variance based on an entire population

Figure 8-4 show examples of using the SUBTOTAL function. Raw data values are listed in column A. The results of using the function in a few variations are listed in column C. Column E displays the actual function entries that returned the respective results in column C.

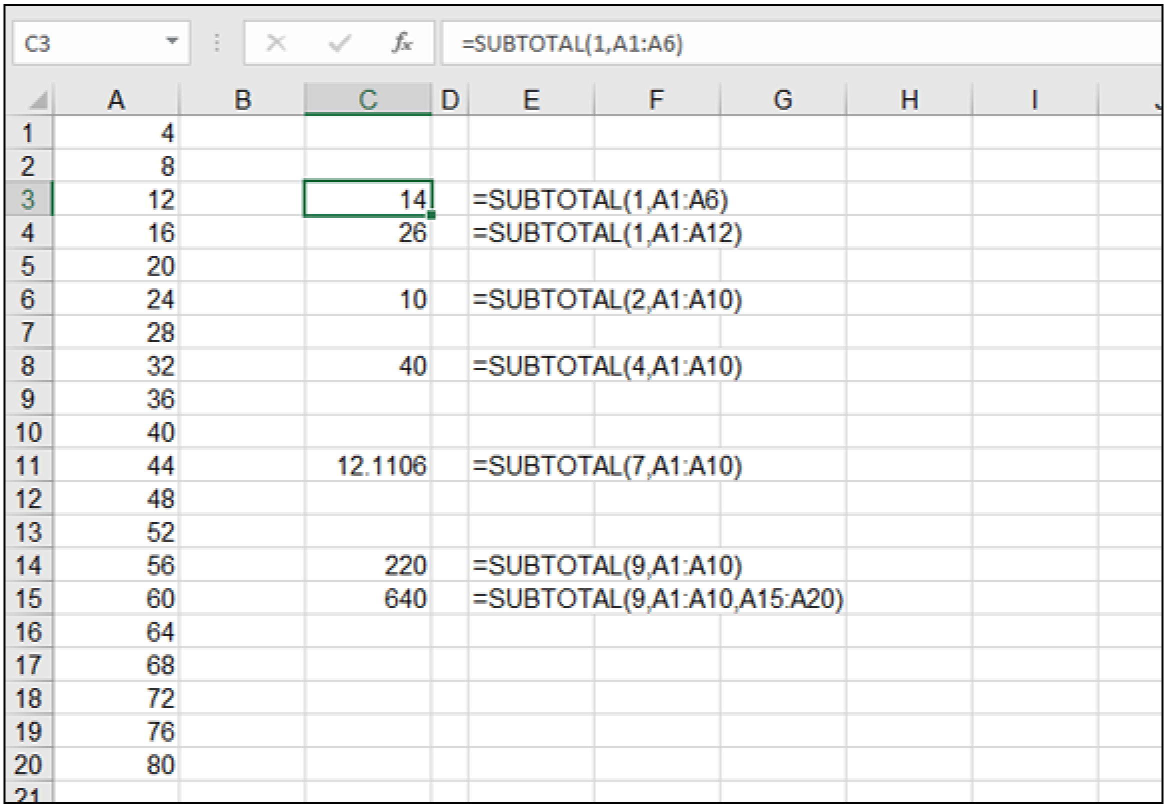


FIGURE 8-4: Working with the SUBTOTAL function.

TIP

Using named ranges with the SUBTOTAL function is useful. For example, `=SUBTOTAL(1, October_Sales, November_Sales, December_sales)` makes for an easy way to calculate the average sale of the fourth quarter.

A second set of numbers can be used for the Function Number (the first argument in the SUBTOTAL function). These numbers start with 101 and are the same functions as shown in Table 8-3. For example, 101 is AVERAGE, 102 is COUNT, and so on.

The 1 through 11 Function Numbers consider all values in a range. The 101 through 111 Function Numbers tell the function to ignore values that are in hidden rows or columns. Figure 8-5 shows SUBTOTAL in use with both Function Number systems. Comparing Figure 8-5 to Figure 8-4, you can see that row 2 has been set to hidden. In Figure 8-5, the values in column B are calculated using the same Function Numbers as in Figure 8-4; column G shows SUBTOTAL using the Function Numbers that start with 101. For example, cell B3 still shows the average of the numbers in the range A1:A6 as equal to 14. The result in cell G3 shows the average of A1:A6 equal to 15.2. The value of 8 in cell A2 is not used because it is hidden.

G3 =SUBTOTAL(101,A1:A6)

	A	B	C	D	E	F	G	H	I	J
1	4									
3	12		14		=SUBTOTAL(1,A1:A6)		15.2		=SUBTOTAL(101,A1:A6)	
4	16		26		=SUBTOTAL(1,A1:A12)		27.63636364		=SUBTOTAL(101,A1:A12)	
5	20									
6	24		10		=SUBTOTAL(2,A1:A10)		9		=SUBTOTAL(102,A1:A10)	
7	28									
8	32		40		=SUBTOTAL(4,A1:A10)		40		=SUBTOTAL(104,A1:A10)	
9	36									
10	40									
11	44		12.1106		=SUBTOTAL(7,A1:A10)		11.7379		=SUBTOTAL(107,A1:A10)	
12	48									
13	52									
14	56		220		=SUBTOTAL(9,A1:A10)		212		=SUBTOTAL(109,A1:A10)	
15	60		640		=SUBTOTAL(9,A1:A10,A15:A20)		632		=SUBTOTAL(109,A1:A10,A15:A20)	
16	64									
17	68									
18	72									
19	76									
20	80									

FIGURE 8-5: Getting SUBTOTAL to ignore hidden values.

USING SUMPRODUCT

The SUMPRODUCT function provides a sophisticated way to add various products — across ranges of values. It doesn't just add the products of separate ranges; it produces products of the values positioned in the same place in each range and then sums up those products. The syntax of the function follows:

```
=SUMPRODUCT(Range1, Range2, …)
```

The arguments to SUMPRODUCT must be ranges, although a range can be a single value. What is required is that all the ranges be the same size, both rows and columns. Up to 255 ranges are allowed, and at least 2 are required.

SUMPRODUCT works by first multiplying elements, by position, across the ranges and then adding all the results. To see how this works, take a look at the three ranges of values in Figure 8-6. I put letters in the ranges instead of numbers to make this easier to explain.

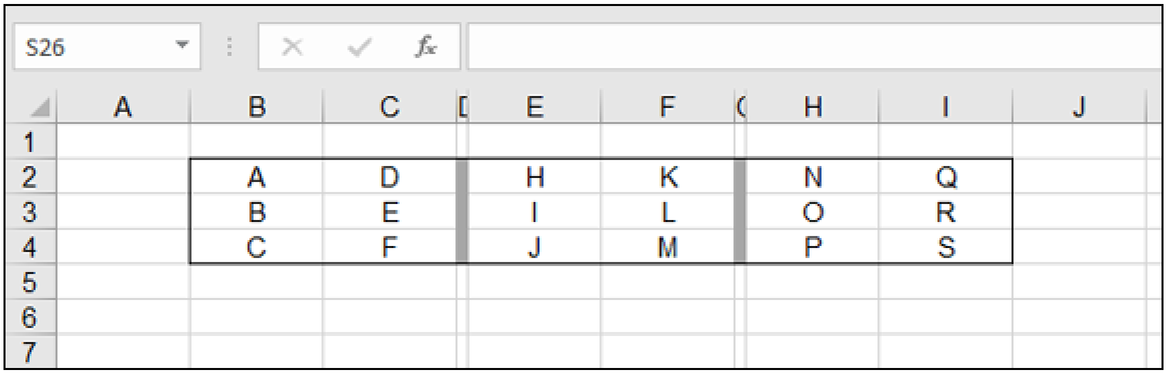


FIGURE 8-6: Following the steps used by SUMPRODUCT.

Suppose that you entered the following formula in the worksheet:

```
=SUMPRODUCT(B2:C4, E2:F4, H2:I4)
```

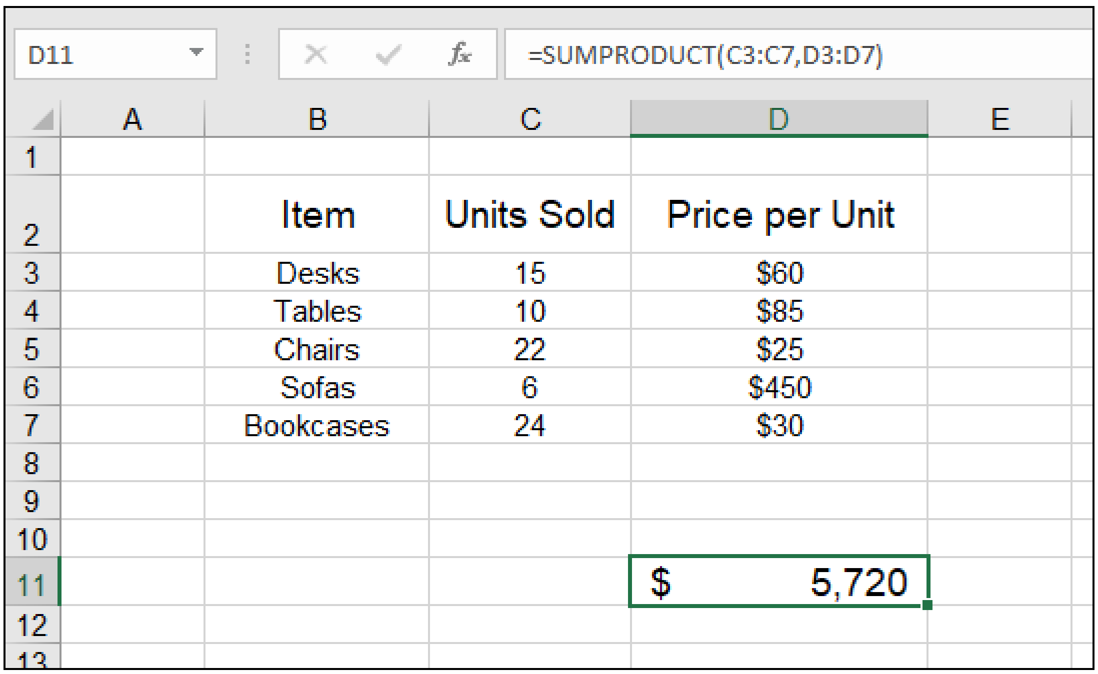
The result would be calculated by the following steps:

1. **Multiplying A times H times N and saving the result**
2. **Multiplying D times K times Q and saving the result**
3. **Multiplying B times I times O and saving the result**
4. **Multiplying E times L times R and saving the result**
5. **Multiplying C times J times P and saving the result**
6. **Multiplying F times M times S and saving the result**
7. **Adding all six results to get the final answer**

WARNING Be careful when you're using the SUMPRODUCT function. It's easy to mistakenly assume that the function adds products of individual ranges. It doesn't. SUMPRODUCT returns the sums of products across positional elements.

As confusing as SUMPRODUCT seems, it actually has a sophisticated use. Imagine that you have a list of units sold by product and another list of the products' prices. You need to know total sales (that is, the sum of the amounts), in which an amount is units sold times the unit price.

In the old days of spreadsheets, you would use an additional column to first multiply each unit sold figure by its price. Then you would sum those intermediate values. Now, with SUMPRODUCT, the drudgery is over. The single use of SUMPRODUCT gets the final answer in one step. Figure 8-7 shows how one cell contains the needed grand total. No intermediate steps are necessary.


D11 =SUMPRODUCT(C3:C7,D3:D7)

	A	B	C	D	E
1					
2		Item	Units Sold	Price per Unit	
3		Desks	15	$60	
4		Tables	10	$85	
5		Chairs	22	$25	
6		Sofas	6	$450	
7		Bookcases	24	$30	
8					
9					
10					
11				$ 5,720	
12					
13					

FIGURE 8-7: Being productive with SUMPRODUCT.

USING SUMIF AND SUMIFS

SUMIF is one of the real gemstones of Excel functions. It calculates the sum of a range of values, including only those values that meet a specified criterion. The criterion can be based on the same column that is being summed, or it can be based on an adjacent column.

Suppose that you use a worksheet to keep track of all your food-store purchases. For each shopping trip, you put the date in column A, the amount in column B, and the name of the store in column C. You can use the SUMIF function to tell Excel to add all the values in column B only where column C contains `"Great Grocery"`. That's it. SUMIF gives you the answer. Neat!

Figure 8-8 shows this example. The date of purchase, place of purchase, and amount spent are listed in three columns. SUMIF calculates the sum of purchases at Great Grocery. Here is how the function is written for the example:

```
=SUMIF(C3:C15,"Great Grocery",B3:B15)
```

C18 =SUMIF(C3:C15,"Great Grocery",B3:B15)

	A	B	C	D	E	F
1	DATE	AMOUNT	STORE			
2						
3	4/12/2018	$ 15.04	Great Grocery			
4	4/16/2018	$ 26.90	Shoppers World			
5	4/15/2018	$ 42.25	Great Grocery			
6	4/17/2018	$ 12.10	The Food Stand			
7	4/22/2018	$ 26.95	Great Grocery			
8	4/30/2018	$ 55.00	Barry's Bistro			
9	5/4/2018	$ 20.25	The Food Stand			
10	5/5/2018	$ 18.55	Shoppers World			
11	5/9/2018	$ 7.95	The Food Stand			
12	5/15/2018	$ 35.00	Great Grocery			
13	5/18/2018	$ 38.80	Shoppers World			
14	5/22/2018	$ 42.00	Barry's Bistro			
15	5/26/2018	$ 43.75	Great Grocery			
16						
17						
18	Amount spent at Great Grocery:		$ 162.99			
19						
20						

FIGURE 8-8: Using SUMIF for targeted tallying.

REMEMBER Here are a couple of important points about the SUMIF function:

- The second argument can accommodate several variations of expressions, such as including greater than (>) or less than (<) signs or other operators. For example, if a column has regions such as North, South, East, and West, the criteria could be `<>North`, which would return the sum of rows that are *not* for the North region.
- Unpredictable results occur if the ranges in the first and third arguments do not match in size.

Try it yourself! Here's how to use the SUMIF function:

1. **Enter two ranges of data in a worksheet.**
   At least one should contain numerical data. Make sure both ranges are the same size.
2. **Position the pointer in the cell where you want the results displayed.**
3. **Type** =SUMIF(**to begin the function.**
4. **Hold down the left mouse button and drag the pointer over one of the ranges.**
   This is the range that can be other than numerical data.
5. **Type a comma (,).**
6. **Click one of the cells in the first range.**
   This is the criterion.
7. **Type a comma (,).**
8. **Hold down the left mouse button and drag the pointer over the second range.**
   This is the range that must contain numerical data.
9. **Type a**) **and press Enter.**

The result you see is a sum of the numeric values where the items in the first range matched the selected criteria.

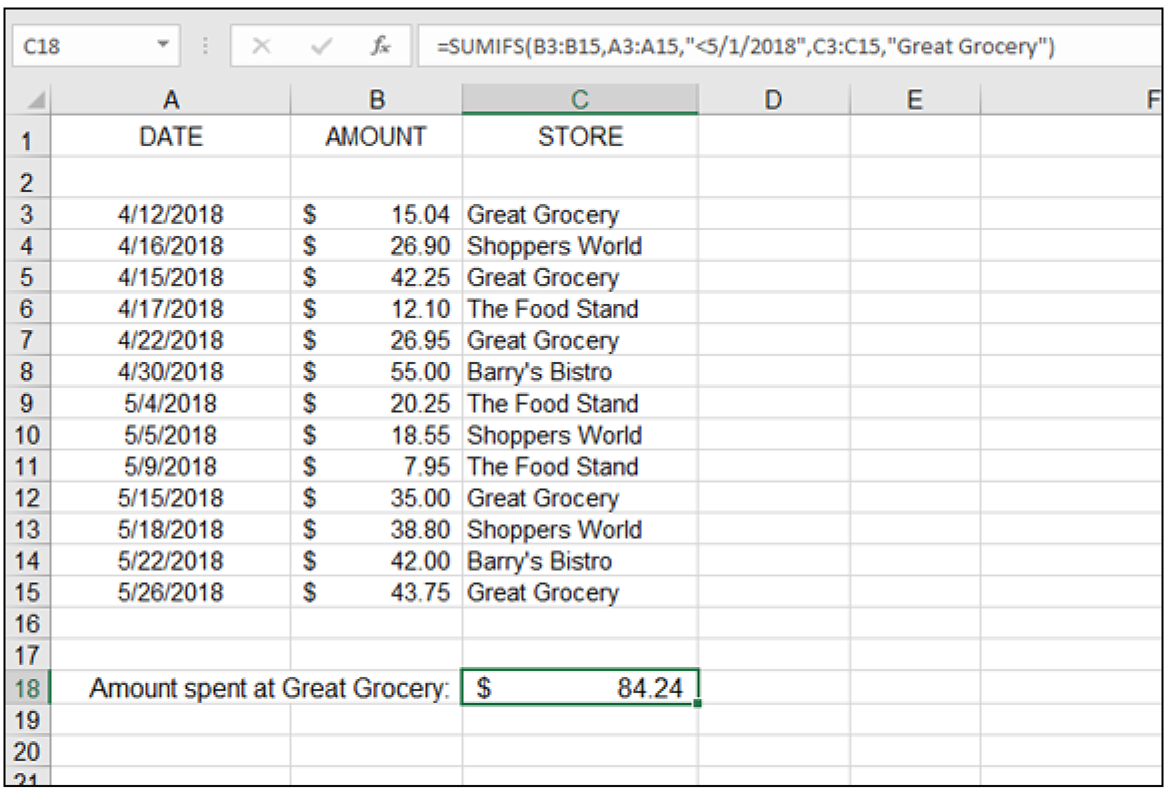
The example in Figure 8-8 sums values when the store is Great Grocery but does not use the date in the calculation. What if you need to know how much was spent at Great Grocery in April only? Excel provides a function for this, of course: SUMIFS.

SUMIFS lets you apply multiple "if" conditions to a sum. The format of SUMIFS is a bit different from SUMIF. SUMIFS uses this structure:

```
=SUMIFS(range to be summed, criteria range 1,
criteria 1, criteria range 2, criteria 2)
```

The structure requires the range of numerical values to be entered first, followed by pairs of criteria ranges and the criteria itself. In Figure 8-9, the formula is

```
=SUMIFS(B3:B15,A3:A15,"<5/1/2018",C3:C15,"Great
Grocery")
```



C18 =SUMIFS(B3:B15,A3:A15,"<5/1/2018",C3:C15,"Great Grocery")

	A	B	C	D	E	F
1	DATE	AMOUNT	STORE			
2						
3	4/12/2018	$ 15.04	Great Grocery			
4	4/16/2018	$ 26.90	Shoppers World			
5	4/15/2018	$ 42.25	Great Grocery			
6	4/17/2018	$ 12.10	The Food Stand			
7	4/22/2018	$ 26.95	Great Grocery			
8	4/30/2018	$ 55.00	Barry's Bistro			
9	5/4/2018	$ 20.25	The Food Stand			
10	5/5/2018	$ 18.55	Shoppers World			
11	5/9/2018	$ 7.95	The Food Stand			
12	5/15/2018	$ 35.00	Great Grocery			
13	5/18/2018	$ 38.80	Shoppers World			
14	5/22/2018	$ 42.00	Barry's Bistro			
15	5/26/2018	$ 43.75	Great Grocery			
16						
17						
18	Amount spent at Great Grocery:		$ 84.24			
19						
20						

FIGURE 8-9: Using SUMIFS to get a multiple filtered sum.

The function uses B3:B15 as the source of values to sum. A1:A15 is the first criteria range, and "<5/1/2018" is the criteria. This tells the function to look for any date that is earlier than May 1, 2018 (which filters the dates to just April). This is followed by a second criteria range and value: In C3:C15, look just for Great Grocery. The final sum of $84.24 adds just three numbers — 15.04, 42.25, and 26.95 — because these are the only values in April for Great Grocery.

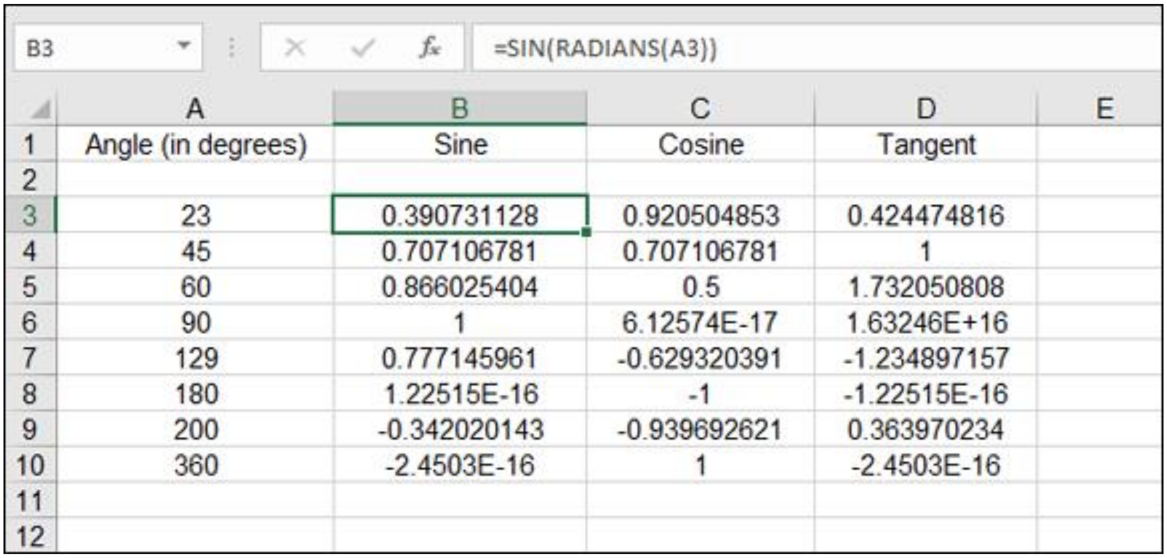
Getting an Angle on Trigonometry

Did you think Excel was not up to snuff to provide some tricks for trigonometry? Then think again. Who can resist playing around with such exciting things like cosines and tangents? All right, I admit this is not for everyone, but here are the trig functions nonetheless. Besides, even if the concepts are difficult, you can always throw around the terms at a party and be recognized as the brainiest person there.

THREE BASIC TRIGONOMETRY FUNCTIONS

The sine, cosine, and tangent of an angle are likely the most used values in trigonometry calculations. They provide answers about the relationships of a triangle's angles to the sides of the triangle. (See how I boiled down the bulk of trigonometry into a single sentence!)

Figure 8-10 shows a handful of angles in column A and their corresponding sine, cosine, and tangent values in columns B, C, and D, respectively.



B3 =SIN(RADIANS(A3))

	A	B	C	D	E
1	Angle (in degrees)	Sine	Cosine	Tangent	
2					
3	23	0.390731128	0.920504853	0.424474816	
4	45	0.707106781	0.707106781	1	
5	60	0.866025404	0.5	1.732050808	
6	90	1	6.12574E-17	1.63246E+16	
7	129	0.777145961	-0.629320391	-1.234897157	
8	180	1.22515E-16	-1	-1.22515E-16	
9	200	-0.342020143	-0.939692621	0.363970234	
10	360	-2.4503E-16	1	-2.4503E-16	
11					
12					

FIGURE 8-10: Using SIN, COS, and TAN functions.

The SIN, COS, and TAN functions take just the single argument of a number (the angle) and return the converted values. The functions look like this if the angles are in radians:

```
=SIN(angle)
=COS(angle)
=TAN(angle)
```

If the angles are in degrees, they need to be converted to radians by the RADIANS function. In this case, you would use these:

```
=SIN(RADIANS(angle))
=COS(RADIANS(angle))
=TAN(RADIANS(angle))
```

Which leads you to …

DEGREES AND RADIANS

An angle can be expressed in degrees or radians. A degree is more common to us non–rocket scientists. Most everyone know that there are 360 degrees in a circle, that a right angle is 90 degrees, and even that “doing a 180” means turning completely around and going the other way.

One radian = 180/pi degrees, and one degree = pi/180 radians. All this talk about pi is making me hungry! For lowdown and quick conversion: 1 radian = 57.3 degrees.

REMEMBER Pi is approximately equal to 3.14159. See the beginning of this chapter for a forkful of pi.

Excel provides the RADIANS and DEGREES functions to convert a number from radians to degrees or vice versa. I think the real reason Excel did this was to keep pi out of the picture so you would concentrate on work and not dessert.

The functions are the single argument type:

```
=RADIANS(angle in degrees)
=DEGREES(angle in radians)
```

Using some numbers, you can see that 90 degrees = 1.5707963267949 radians and so 1.5707963267949 radians must equal 90 degrees. And it does.

Understanding How Excel Handles Dates

Imagine that on January 1, 1900, you started counting by ones, each day adding one more to the total. This is just how Excel thinks of dates. January 1, 1900, is one; January 2, 1900, is two; and so on. We'll always remember 25,404 as the day man first walked on the moon, and 36,892 as the start of the new millennium!

TECHNICAL STUFF The millennium actually started on January 1, 2001. The year 2000 is the last year of the 20th century. Representing dates as a serial number — specifically, the number of days between January 1, 1900, and the date in question — may seem odd, but there are very good reasons for it. Excel can handle dates from January 1, 1900, to December 31, 9999. Using the serial numbering system, that's 1 through 2,958,465!

Because Excel represents dates in this way, it can work with dates in the same manner as numbers. For example, you can subtract one date from another to find out how many days are between them. Likewise, you can add 14 to today's date to get a date two weeks in the future. This trick is very useful, but people are used to seeing dates represented in traditional formats, not as numbers. Fortunately, Excel uses date serial numbers only behind the scenes, and what you see in your workbook are dates in the standard date formats such as Jan 20, 2018 and 1/20/18.

TIP In Excel for the Mac, the serial numbering system begins on January 1, 1904.

The way years are handled requires special mention. When a year is fully displayed in 4 digits, such as 2018, there is no ambiguity. However, when a date is written in a shorthand style, such as in 3/1/02, it isn't clear what the year is. It could be 2002 or it could be 1902. Suppose that 3/1/02 is a shorthand entry for someone's birthday. On March 1, 2005, he is either 3 years old or 103 years old. In those countries that write dates as dd/mm/yy, this would be January 3, 1902 or January 3, 2002.

Excel and the Windows operating system have a default way of interpreting shorthand years. Windows has a setting in the Customize Regional Options dialog box located in the Control Panel. This setting guides how Excel interprets years. If the setting is 1930 through 2029, 3/1/18 indicates the year 2018, but 3/1/45 indicates the year 1945, not 2045. Figure 12-1 shows this setting.

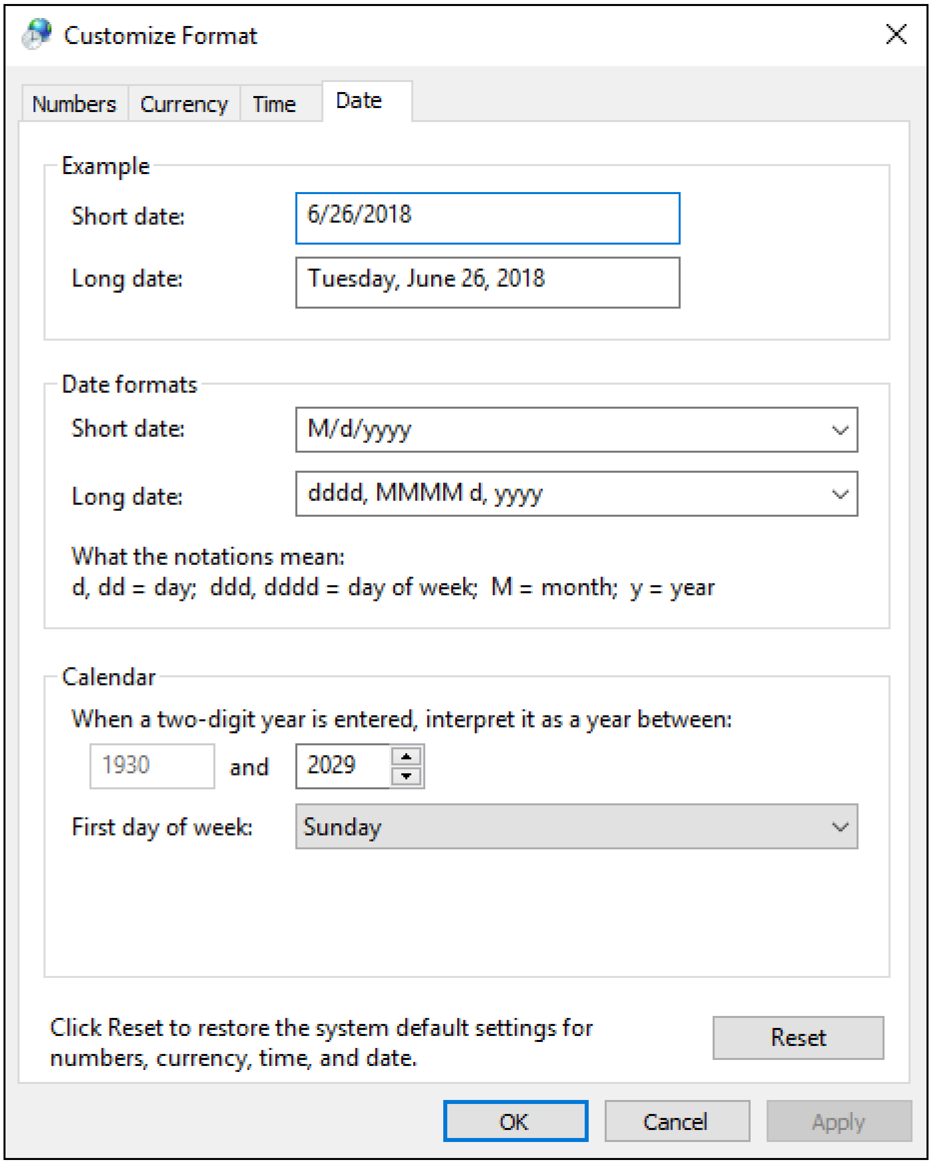


FIGURE 12-1: Setting how years are interpreted in the Customize Format dialog box.

Here's how to open and set it:

1. **Use the Windows search feature to find and open Control Panel.**
2. **Select Clock and Region.**
3. **Select Region.**

   The Region dialog box opens.

4. **Click the Formats tab.**
5. **Click the Additional Settings button.**

   The Customize Format dialog box opens.
6. **Click the Date tab.**
7. **In the Calendar section, select a 4-digit ending year (such as 2029) to indicate the latest year that will be used when interpreting a 2-digit year.**
8. **Click OK to close each dialog box.**

TIP To ensure full accuracy when working with dates, always enter the full 4 digits for the year.

Formatting Dates

When you work with dates, you probably need to format cells in your worksheet. It’s great that Excel tells you that June 1, 2018, is serially represented as 43252, but you probably don’t want that in a report. To format dates, you use the Format Cells dialog box, shown in Figure 12-2.

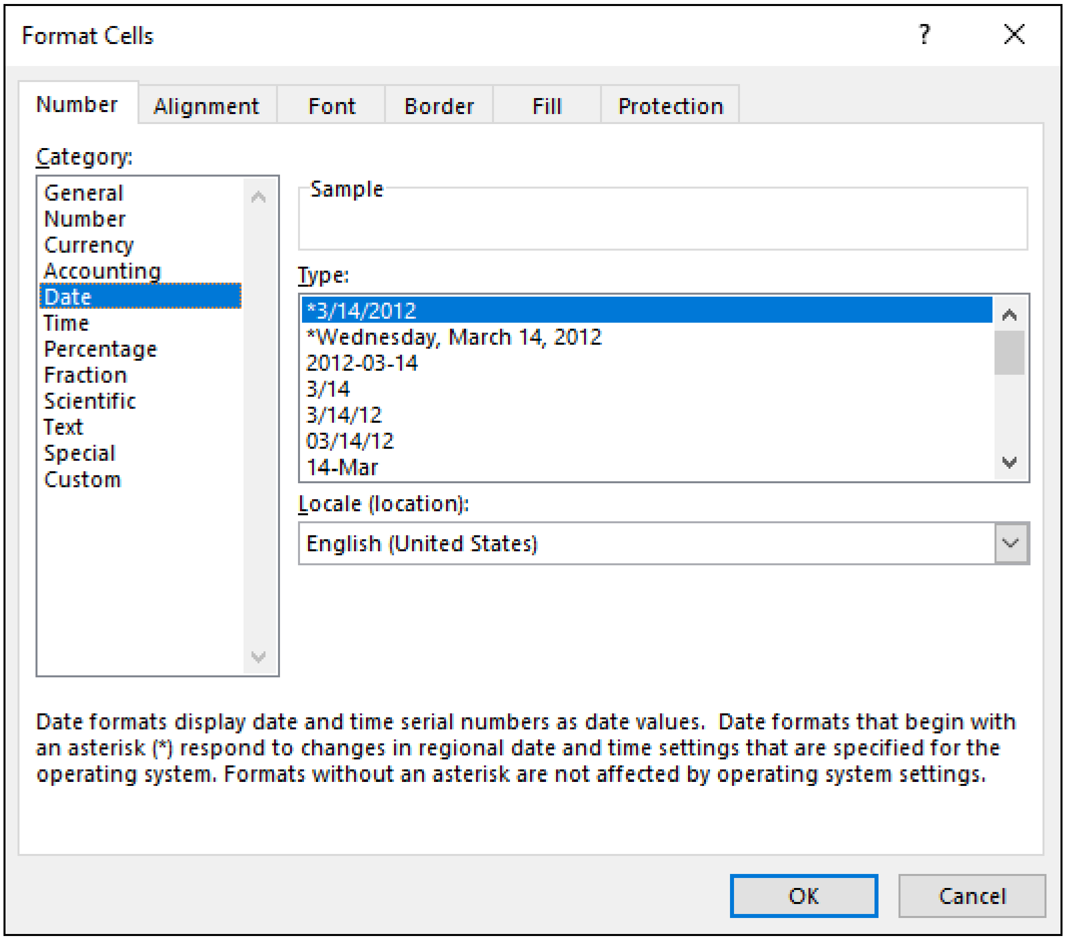


FIGURE 12-2: Using the Format Cells dialog box to control how dates are displayed.

To format the currently selected cells as dates, follow these steps:

1. **If it's not already displayed, click the Home tab at the top of the Excel screen.**
2. **Click the small arrow at the bottom-right corner of the Number section.**

   The Format Cells dialog box appears, revealing the Number tab.
3. **Select Date from the Category list.**
4. **Select an appropriate format from the Type list.**

Now you can turn the useful but pesky serial dates into a user-friendly format.

TIP When you enter a date in a cell using one of the standard date formats, Excel recognizes it as a date and automatically assigns a Date format to the cell. You may want to use the Number tab in the Format Cells dialog box to assign a different Date format.

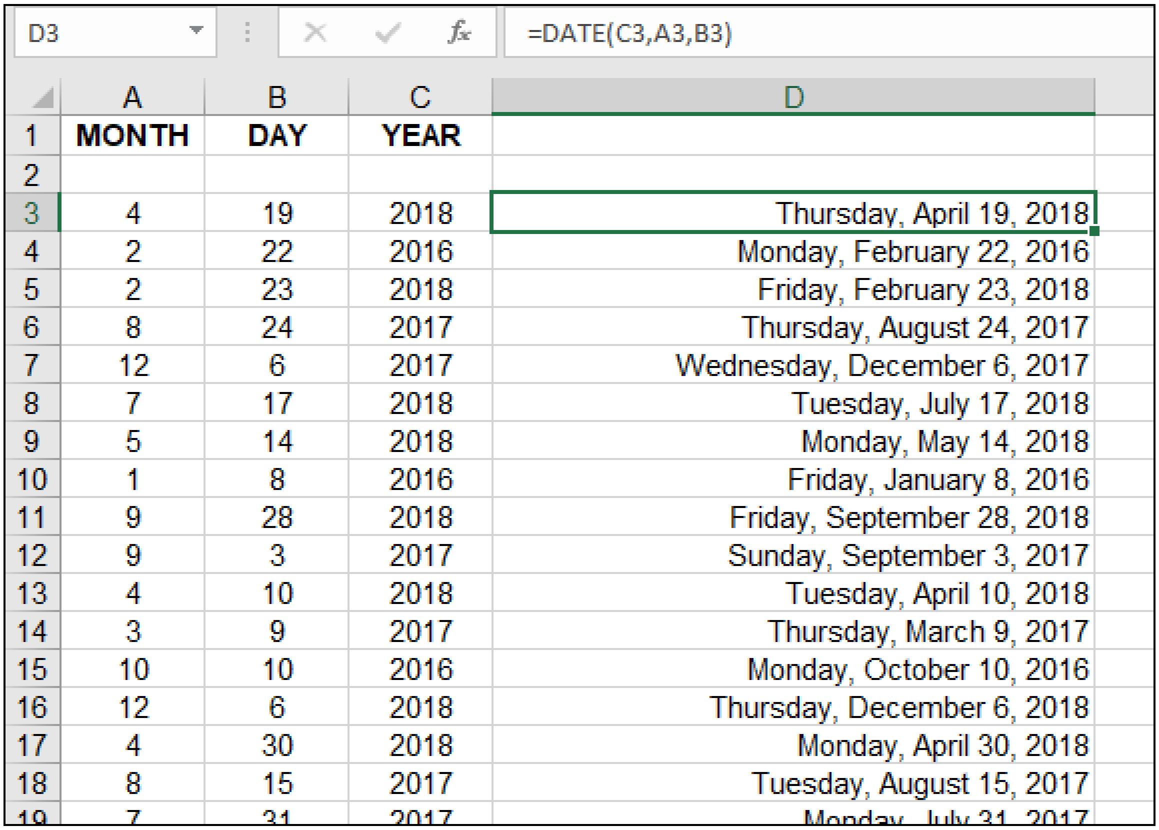
Making a Date with DATE

You can use the DATE function to create a complete date from separate year, month, and day information. The DATE function can be useful because dates don’t always appear as, well, dates, in a worksheet. You may have a column of values between 1 and 12 that represents the month and another column of values between 1 and 31 for the day of the month. A third column may hold years — in either 2-digit shorthand or the full 4 digits.

The DATE function combines individual day, month, and year components into a single usable date. This makes using and referencing dates in your worksheet easy. Follow along to use the DATE function:

1. **Select the cell where you want the results displayed.**
2. **Type** =DATE(**to begin the function entry.**
3. **Click the cell that has the year.**
4. **Type a comma (,).**
5. **Click the cell that has the number (1–12) that represents the month.**
6. **Type a comma (,).**
7. **Click the cell that has the number (1–31) that represents the day of the month.**
8. **Type a**) **and press Enter.**

Figure 12-3 displays a fourth column of dates that were created by using DATE and the values from the first three columns. The fourth column of dates has been formatted so the dates are displayed in a standard format, not as a raw date serial number.



D3 =DATE(C3,A3,B3)

	A	B	C	D
1	MONTH	DAY	YEAR	
2				
3	4	19	2018	Thursday, April 19, 2018
4	2	22	2016	Monday, February 22, 2016
5	2	23	2018	Friday, February 23, 2018
6	8	24	2017	Thursday, August 24, 2017
7	12	6	2017	Wednesday, December 6, 2017
8	7	17	2018	Tuesday, July 17, 2018
9	5	14	2018	Monday, May 14, 2018
10	1	8	2016	Friday, January 8, 2016
11	9	28	2018	Friday, September 28, 2018
12	9	3	2017	Sunday, September 3, 2017
13	4	10	2018	Tuesday, April 10, 2018
14	3	9	2017	Thursday, March 9, 2017
15	10	10	2016	Monday, October 10, 2016
16	12	6	2018	Thursday, December 6, 2018
17	4	30	2018	Monday, April 30, 2018
18	8	15	2017	Tuesday, August 15, 2017
19	7	31	2017	Monday, July 31, 2017

FIGURE 12-3: Using the DATE function to assemble a date from separate month, day, and year values.

DATE provides some extra flexibility with the month number. Negative month numbers are subtracted from the specified year. For example, the function `=DATE(2018, -5, 15)` returns the date July 15, 2017, because July 2017 is 5 months before the first month of 2018. Numbers greater than 12 work the same way. `=DATE(2018, 15, 1)` returns March 1, 2019, because March 2019 is 15 months after the first month of 2018.

Day numbers work the same way. Negative day numbers are subtracted from the first of the specified month, and numbers that are greater than the last day of the specified month wrap into later months. Thus, `=DATE(2018, 2, 30)` returns March 2, 2018, because February does not have 30 days. Likewise, `=DATE(2018, 2, 40)` returns March 12, 2018.

Breaking a Date with DAY, MONTH, and YEAR

That which can be put together can also be taken apart. In the preceding section, I show you how to use the DATE function to create a date from separate year, month, and day data. In this section, you find out how to do the reverse: Split a date into individual year, month, and day components by using the DAY, MONTH, and YEAR functions. In Figure 12-4, the dates in column A are split apart by day, month, and year, respectively, in columns B, C, and D.

B3 =DAY(A3)

	A	B	C	D	E
1	Date	DAY	MONTH	YEAR	
2	1/1/2014	1	1	2014	
3	5/22/2012	22	5	2012	
4	3/13/2013	13	3	2013	
5	9/28/2013	28	9	2013	
6	9/3/2012	3	9	2012	
7	4/10/2013	10	4	2013	
8	3/9/2012	9	3	2012	
9	8/15/2012	15	8	2012	
10	7/31/2012	31	7	2012	
11	11/14/2012	14	11	2012	
12	5/8/2013	8	5	2013	
13	8/2/2012	2	8	2012	
14	2/26/2012	26	2	2012	
15	1/20/2013	20	1	2013	
16	12/1/2012	1	12	2012	
17	4/22/2012	22	4	2012	
18	12/12/2013	12	12	2013	
19	6/17/2013	17	6	2013	
20	5/31/2012	31	5	2012	
21	1/31/2013	31	1	2013	

FIGURE 12-4: Splitting apart a date with the DAY, MONTH, and YEAR functions.

ISOLATING THE DAY

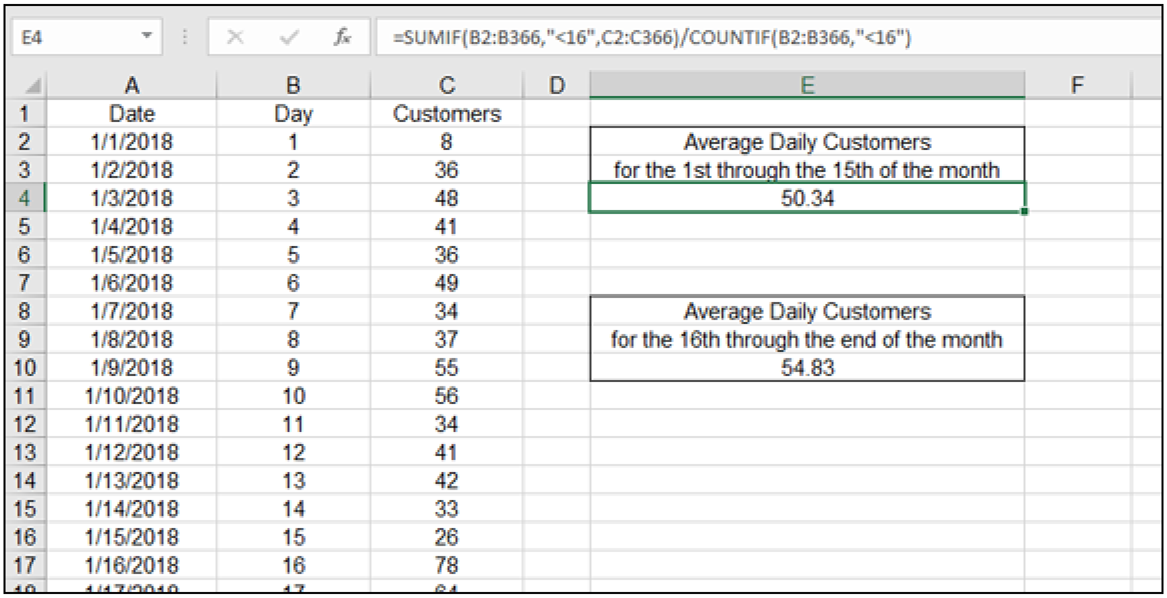
Isolating the day part of a date is useful in applications in which just the day, but not the month or year, is relevant. Suppose that you own a store and want to figure out whether more customers come to shop in the first half or the second half of the month. You're interested in this trend over several months. So the task may be to average the number of sales by the day of the month only.

The DAY function is useful for this because you can use it to return just the day for a lengthy list of dates. Then you can examine results by the day only.

Here’s how you use the DAY function:

1. **Position the pointer in the cell where you want the results displayed.**
2. **Type** =DAY(**to begin the function entry.**
3. **Click the cell that has the date.**
4. **Type a**) **and press Enter.**
   Excel returns a number between 1 and 31.

Figure 12-5 shows how the DAY function can be used to analyze customer activity. Column A contains a full year’s sequential dates (most of which are not visible in the figure). In column B, the day part of each date has been isolated. Column C shows the customer traffic for each day.



E4 | =SUMIF(B2:B366,"<16",C2:C366)/COUNTIF(B2:B366,"<16")

	A	B	C	D	E	F
1	Date	Day	Customers			
2	1/1/2018	1	8		Average Daily Customers	
3	1/2/2018	2	36		for the 1st through the 15th of the month	
4	1/3/2018	3	48		50.34	
5	1/4/2018	4	41			
6	1/5/2018	5	36			
7	1/6/2018	6	49			
8	1/7/2018	7	34		Average Daily Customers	
9	1/8/2018	8	37		for the 16th through the end of the month	
10	1/9/2018	9	55		54.83	
11	1/10/2018	10	56			
12	1/11/2018	11	34			
13	1/12/2018	12	41			
14	1/13/2018	13	42			
15	1/14/2018	14	33			
16	1/15/2018	15	26			
17	1/16/2018	16	78			

FIGURE 12-5: Using the DAY function to analyze customer activity.

This is all the information you need to analyze whether there is a difference in the amount of customer traffic between the first half and second half of the month.

Cells E4 and E10 show the average daily customer traffic for the first half and second half of the month, respectively. The value for the first half of the month was obtained by adding all the customer values for day values in the range 1 to 15 and then dividing by the total number of days. The value for the second half of the month was done the same way, using day values in the range 16 to 31.

The day parts of the dates, in column B, were key to these calculations:

- In cell E4, the calculation is `=SUMIF(B2:B366,"<16",C2:C366)/COUNTIF(B2:B366"<16")`.
- In cell E10, the calculation is `=SUMIF(B2:B366,">15",C2:C366)/COUNTIF(B2:B366,">15")`.

The SUMIF function is discussed in Chapter 8. The COUNTIF function is discussed in Chapter 9.

The DAY function has been instrumental in showing that more customers visit the fictitious store in the second half of the month. This type of information is great for helping a store owner plan staff assignments, sales specials, and so on.

ISOLATING THE MONTH

Isolating the month part of a date is useful in applications in which just the month, but not the day or year, is relevant. For example, you may have a list of dates on which more than five of your employees call in sick and need to determine whether this event is more common in certain months than others.

You could sort the dates and then count the number for each month. That would be easy enough, but sorting may not be an option based on other requirements. Besides, why manually count when you have, right in front of you, one of the all-time greatest counting software programs ever made?

Figure 12-6 shows a worksheet in which the MONTH function has extracted the numeric month value (1–12) into column B from the dates in column A. Cell B2 contains the formula `=MONTH(A2)` and so on down the column. Columns C and D contain a summary of dates per month. The formula used in cell D3 is `=COUNTIF($B$2:$B$260,1)`.

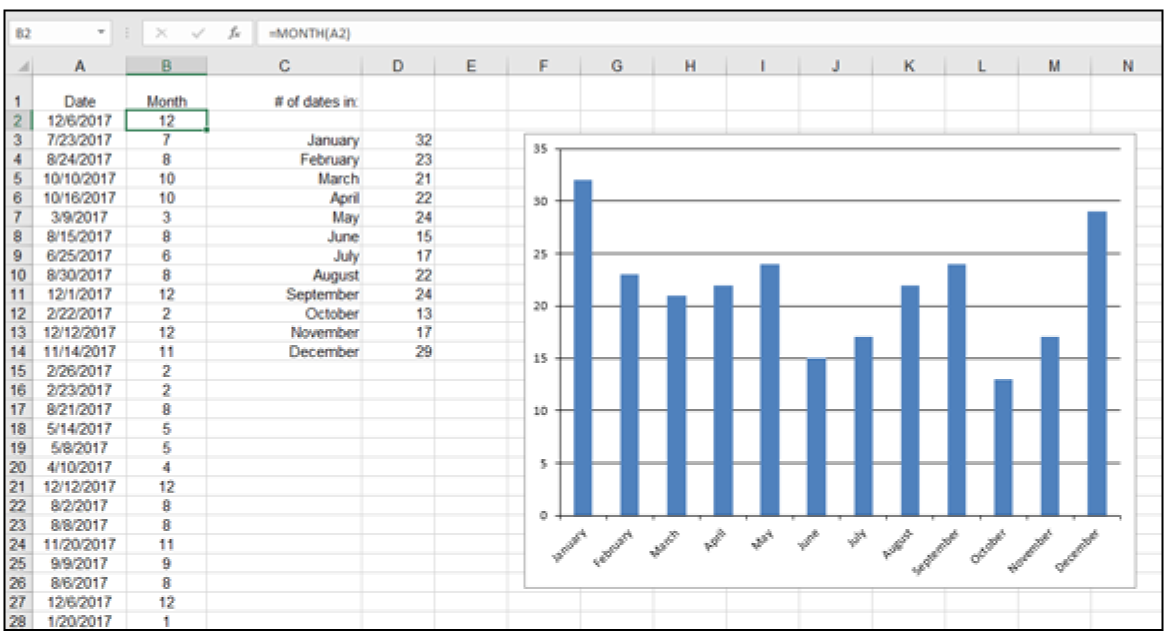


FIGURE 12-6: Using the MONTH function to count the number of dates falling in each month.

This counts the number of dates in which the month value is 1 — in other words, January. Cells D4 through D14 contain similar formulas for month values 2 through 12. The figure's data plot makes it clear that calling in sick is more prevalent in December and January. See Chapter 9 for information on the COUNTIF function.

Use the MONTH function this way:

1. **Select the cell where you want the results displayed.**
2. **Type** =MONTH(**to begin the function entry.**
3. **Click the cell that has the date.**
4. **Type a**) **and press Enter.**

   Excel returns a number between 1 and 12.

ISOLATING THE YEAR

Isolating the year part of a date is useful in applications in which only the year, but not the day or month, is relevant. In practice, this is less used than the DAY or MONTH functions because date data is often — though not always — from the same year.

Follow these steps to use the YEAR function:

1. **Select the cell where you want the results displayed.**
2. **Type** =YEAR(**to begin the function entry.**
3. **Click the cell that has the date.**
4. **Type a**) **and press Enter.**

   Excel returns the 4-digit year.

Converting a Date from Text

You may have data in your worksheet that looks like a date but is not represented as an Excel date value. For example, if you enter 01-24-18 in a cell, Excel would have no way of knowing whether this is January 24, 2018, or the code for your combination lock. If it looks like a date, you can use the DATEVALUE function to convert it to an Excel date value.

In practice, any standard date format entered into a cell is recognized by Excel as a date and converted accordingly. However, there may be cases, such as when text dates are imported from an external data source or data is copied and pasted into Excel, for which you need DATEVALUE.

TECHNICAL STUFF Why not enter dates as text data? Although they may look fine, you can't use them for any of Excel's powerful date calculations without first converting them to date values.

The DATEVALUE function recognizes almost all commonly used ways that dates are written. Here are some ways that you may enter August 14, 2018:

- 8/14/18
- 14-Aug-2018
- 2018/08/14

DATEVALUE can convert these and several other date representations to a date serial number.

After you've converted the dates to a date serial number, you can use the dates in other date formulas or perform calculations with them as described in other parts of this chapter.

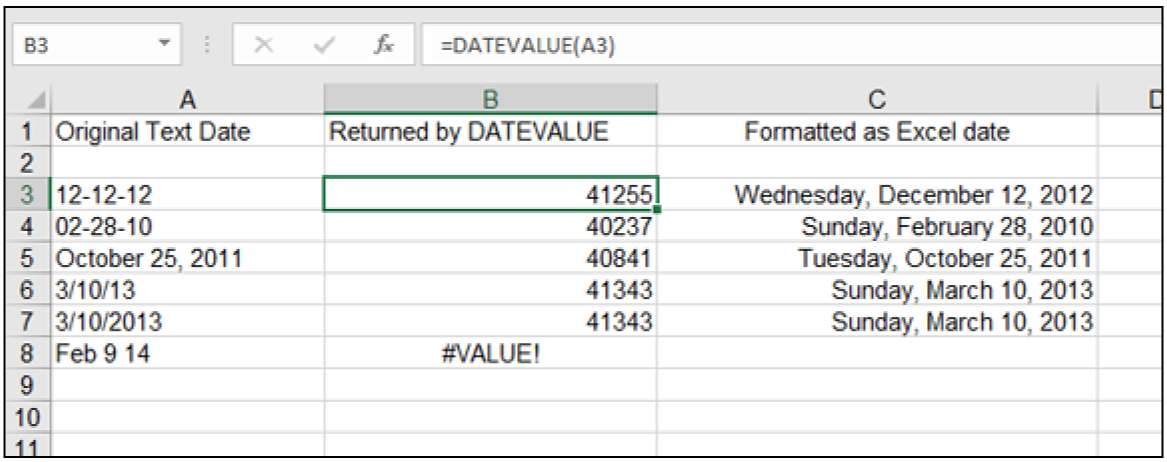
To use the DATEVALUE function, follow these steps:

1. **Select the cell where you want the date serial number located.**

2. **Type** =DATEVALUE(**to begin the function entry.**
3. **Click the cell that has the text format date.**
4. **Type a) and press Enter.**

   The result is a date serial number unless the cell where the result is displayed has already been set to a date format.

Figure 12-7 shows how some nonstandard dates in column A have been converted to serial numbers with the DATEVALUE function in column B. Then column C displays these serial numbers formatted as dates.



B3 =DATEVALUE(A3)

	A	B	C
1	Original Text Date	Returned by DATEVALUE	Formatted as Excel date
2			
3	12-12-12	41255	Wednesday, December 12, 2012
4	02-28-10	40237	Sunday, February 28, 2010
5	October 25, 2011	40841	Tuesday, October 25, 2011
6	3/10/13	41343	Sunday, March 10, 2013
7	3/10/2013	41343	Sunday, March 10, 2013
8	Feb 9 14	#VALUE!	
9			
10			
11			

FIGURE 12-7: Converting dates to their serial equivalents with the DATEVALUE function.

Do you notice something funny in Figure 12-7? Normally, you aren't able to enter a value such as the one in cell A4 — 02-28-10 — without losing the leading 0. The cells in column A had been changed to the Text format. This format tells Excel to leave your entry as is. The Text format is one of the choices in the Category list in the Format Cells dialog box (refer to Figure 12-2).

Note also that the text date in cell A8, `Feb 9 14`, could not be converted by DATEVALUE, so the function returns the error message `#VALUE#`. Excel is great at recognizing dates, but I did not say it is perfect! In cases such as this, you have to format the date another way so DATEVALUE can recognize it.

Finding Out What TODAY Is

When working in Excel, you often need to use the current date. Each time you print a worksheet, for example, you may want the day's date to show. The TODAY function fills the bill perfectly. It simply returns the date from your computer's internal clock. To use the TODAY function, follow these steps:

1. **Position the pointer in the cell where you want the result.**
2. **Type** =TODAY( )**.**
3. **Press Enter to end the function.**

That's it! You now have the date from your computer. If your computer's clock is not set correctly, don't blame Excel. As with all dates in Excel, what you really end up with is a serial number, but the Date formatting displays the date in a readable fashion.

As with all functions in Excel, you can embed functions in other functions. For example, if you need to know just the current date's month, you can combine the TODAY function with the MONTH function, like this:

```
=MONTH(TODAY())
```

COUNTING THE DAYS UNTIL YOUR BIRTHDAY

After a certain age, a lot of people wish their birthdays would not come around so often, but if you still like birthdays, you can use Excel to keep track of how many days are left until the next one. Entered in a cell, this formula tells you how many days are left until your birthday (assuming that your next birthday is May 5, 2019):

```
=DATE(2019,5,5) - TODAY()
```

Use the DATE function to enter the day, month, and year of your next birthday. This prevents Excel from interpreting a shorthand entry, such as 5/5/2019, as a mathematical operation on its own.

If the formula were `=5/5/2019 – TODAY()`, Excel would calculate an incorrect answer because the formula effectively says, “Divide 5 by 5, then divide that result by 2019, then subtract the serial number of today’s date.” The answer would be incorrect.

REMEMBER Using the DATE function to represent dates in which a mathematical operation is performed is a good idea.

COUNTING YOUR AGE IN DAYS

When your birthday finally rolls around, someone may ask how old you are. Maybe you’d rather not say. Here’s a way to respond, but in a way that leaves some doubt: Answer by saying how old you are in days!

Excel can help you figure this out. All you have to do is count the number of days between your birth date and the current date. A simple formula tells you this:

```
=TODAY() - DATE(birth year, birth month, birth day)
```

Here's an example, assuming that your birthday is March 18, 1976:

```
=TODAY() - DATE(1976, 3, 18).
```

Determining the Day of the Week

The Beatles recorded a song called “Eight Days a Week,” but for the rest of us, 7 days is the norm. The WEEKDAY function helps you figure out which day of the week a date falls on. Now you can figure out whether your next birthday falls on a Friday. Or you can make sure that a planned business meeting does not fall on a weekend.

Here is how you use the WEEKDAY function:

1. **Select the cell where you want the results displayed.**
2. **Type** =WEEKDAY(**to begin the function entry.**
3. **Select the cell that has the date for which you want to find the weekday.**
4. **Type a) and press Enter.**
   WEEKDAY returns a number between 1 and 7. Table 12-1 shows what the returned number means.

TABLE 12-1 WEEKDAY Returned Values

Returned Value	*Weekday*
1	Sunday
2	Monday
3	Tuesday
4	Wednesday

5	Thursday
6	Friday
7	Saturday

Don’t confuse the returned numbers with actual dates! Just because Table 12-1 shows a value of 4 indicating Wednesday doesn’t mean that the fourth day of a month is a Wednesday. The values of the returned numbers are also a bit confusing because most people consider Monday, not Sunday, to be the first day of the week. You can go argue the point with Microsoft, if you like! Better yet, you can include a second, optional, argument that tells WEEKDAY to return 1 for Monday, 2 for Tuesday, and so on:

```
=WEEKDAY(A1, 2)
```

REMEMBER The numbers 1 through 7, returned from the WEEKDAY function, are not the same as the first through seventh of the month.

The WEEKDAY function lets you extract interesting information from date-related data. For example, maybe you’re on a diet, and you’re keeping a tally of how many calories you consume each day for a month. Then you start wondering “On which days do I eat the most?” Figure 12-8 shows a worksheet that calculates the average calories consumed on each day of the week over a month’s time. A glance at the results shows that Saturdays and Sundays are not your high-calorie-consumption days; it’s Monday and Tuesday that you have to watch out for.

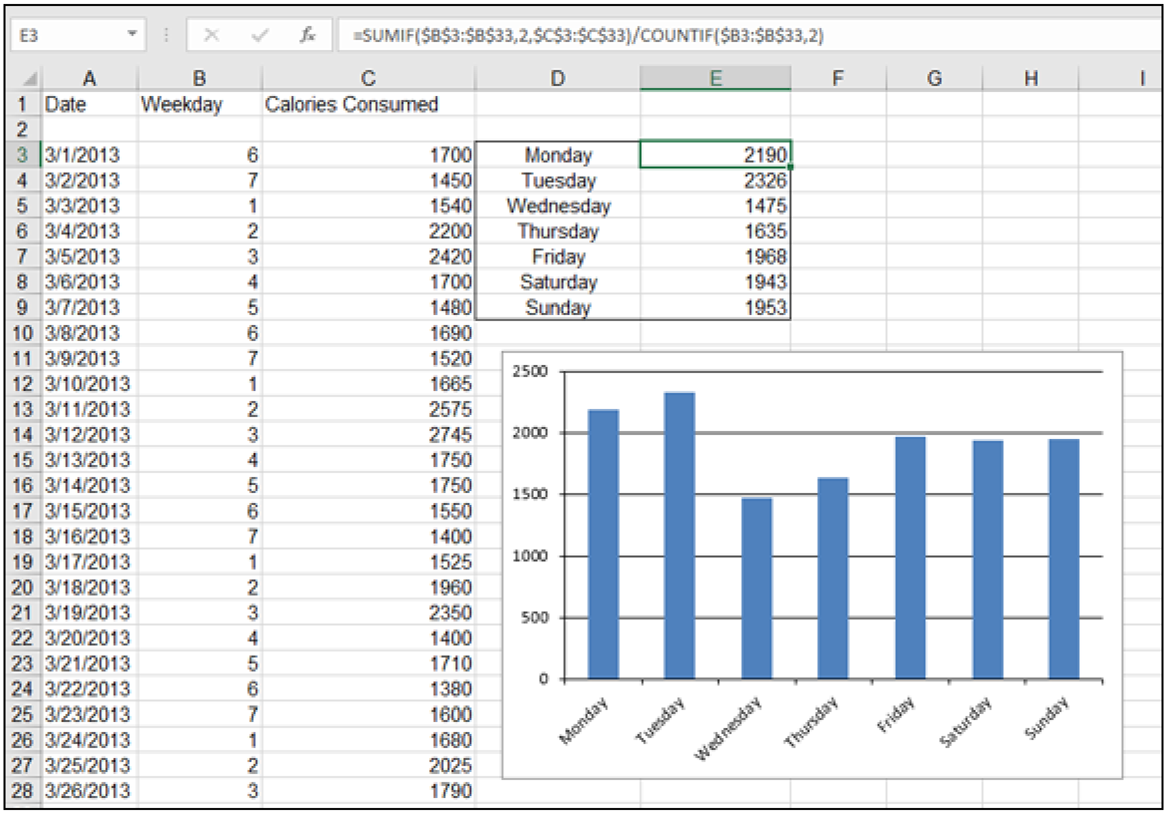


FIGURE 12-8: Using WEEKDAY tells you which day of the week a date falls on.

Working with Workdays

Most weeks have 5 workdays — Monday through Friday — and 2 weekend days. (I know; some weeks seem to have 20 workdays, but that's just your imagination!) Excel has two functions that let you perform workday-related calculations.

DETERMINING WORKDAYS IN A RANGE OF DATES

The NETWORKDAYS function tells you how many working days are in a range of dates. Do you ever sit at your desk and stare at the calendar, trying to count how many working days are left in the year? Excel can answer this vital question for you!

NETWORKDAYS counts the number of days, omitting Saturdays and Sundays, in a range of dates that you supply. You can add a list of dates that should not be counted, if you want. This optional list is where you can put holidays, vacation time, and so on.

Figure 12-9 shows an example using NETWORKDAYS. Cells C3 and C4 show the start and end dates, respectively. In this example, the start date is provided the TODAY function. Therefore, the result always reflects a count that starts from the current date. The end date is the last day of the year. The function in cell C6 is `=NETWORKDAYS(C3,C4,C10:C23)`.

C6 | fx =NETWORKDAYS(C3,C4,C10:C23)

	A	B	C	D
1				
2				
3		Today's Date	6/26/2018	
4		End of Year	12/31/2018	
5				
6		**Remaining Work Days:**	**127**	
7				
8				
9				
10		New Year's Day	1/1/2018	
11		Martin Luther King Jr. Day	1/15/2018	
12		President's Day	2/19/2018	
13		Memorial Day	5/28/2018	
14		Independence Day	7/4/2018	
15		Labor Day	9/3/2018	
16		Vacation	9/20/2018	
17		Vacation	9/21/2018	
18		Vacation	9/22/2018	
19		Vacation	9/23/2018	
20		Vacation	9/24/2018	
21		Columbus Day	10/8/2018	
22		Thanksgiving	11/22/2018	
23		Christmas	12/25/2018	
24				

FIGURE 12-9: Counting workdays with NETWORKDAYS.

The function includes the cells that have the start and end dates. Then there is a range of cells: C10 through C23. These cells have dates that should not be counted in the total of workdays: holidays and vacations. You can put anything in these cells, but they do have to be Excel dates. If a date specified in this list falls on a workday, NETWORKDAYS does not count it. If it falls on a weekend, it would not be counted anyway, so it is ignored.

To use NETWORKDAYS, follow these steps:

1. **Select the cell where you want the results displayed.**
2. **Type** =NETWORKDAYS(**to begin the function entry.**
3. **Click the cell that has the start date for the range of dates to be counted.**
4. **Type a comma (,).**
5. **Click the cell that has the end date for the range of dates to be counted.**

   If you want to add a list of dates to exclude, continue to steps 6 and 7; otherwise, go to Step 8.
6. **Type a comma (,).**
7. **Click and drag the pointer over the cells that have the dates to exclude.**
8. **Type a**) **and press Enter.**

   The result is a count of days, between the start and end dates, that do not fall on Saturday or Sunday and are not in an optional list of exclusion dates.

WORKDAYS IN THE FUTURE

Sometimes, you are given a deadline ("Have that back to me in 20 working days"), or you give it to someone else. Fine, but what is the date 20 working days from now? The WORKDAY function comes to the rescue. You specify a start date, the number of working days, and an optional list of holidays that are not to be counted as working days. (This list works just the same as for the NETWORKDAYS function, discussed in the previous section.)

To use WORKDAYS, follow these steps:

1. **Select the cell where you want the results displayed.**
2. **Type** =WORKDAY(**to begin the function entry.**
3. **Click the cell that has the start date for the calculation.**
4. **Type a comma (,).**
5. **Click the cell that has the number of workdays or enter the number directly in the formula.**

   If you want to add a list of dates to exclude in the count, continue to steps 6 and 7; otherwise, go to Step 8.
6. **Type a comma (,).**
7. **Click and drag the pointer over the cells that have the dates to be excluded.**
8. **Type a**) **and press Enter.**

   The result is a date that is the specified number of workdays from the start date, not counting dates in the optional list of exclusion dates.

Calculating Time Between Two Dates with the DATEDIF Function

Excel provides the DATEDIF function to calculate the number of days, months, or years between two dates. This is an undocumented function; that is, you won't see it in the Insert Function dialog box, and you cannot find it in the Excel Help system. Why is it undocumented? Beats me — but it sure can be useful! Impress your friends and coworkers. The only thing you have to do is remember how to enter it. For that, you might want to keep this book handy to look it up.

DATEDIF takes three arguments:

- Start date
- End date
- Interval

The interval argument tells the function what type of result to return, summarized in Table 12-2.

TABLE 12-2 Settings for the Interval Argument of DATEDIF

Value	***What It Means***	***Comment***
`"d"`	Days	The count of inclusive days from the start date through the end date.
`"m"`	Months	The count of complete months between the dates. Only those months that fully occur between the dates are counted. For example, if the first date starts after the first of the month, that first month is not included in the count. For the end date, even when it is the last day of the month, that month is not counted. See Figure 12-10 for an example.

"y"	Years	The count of complete years between the dates. Only those years that fully occur between the dates are counted. For example, if the first date starts later than January 1, that first year is not included in the count. For the end date, even when it is December 31, that year is not counted. See Figure 12-10 for an example.
"yd"	Days excluding years	The count of inclusive days from the start date through the end date, but as if the two dates are in the same year. The year is ignored.
"ym"	Months excluding years	The count of complete months between the dates, but as if the two dates are in the same year. The year is ignored.
"md"	Days excluding months and years	The count of inclusive days from the start date through the end date, but as if the two dates are in the same month and year. The month and year are ignored.

C13 =DATEDIF(A13,B13,"d")

	A	B	C	D	E	F	G	H
1								
2	Start Date	End Date	"d"	"m"	"y"	"yd"	"ym"	"md"
3								
4	1/1/2013	1/1/2013	0	0	0	0	0	0
5	1/1/2013	1/2/2013	1	0	0	1	0	1
6	1/1/2013	12/31/2013	364	11	0	364	11	30
7	1/1/2013	1/1/2014	365	12	1	0	0	0
8	1/1/2013	5/10/2014	494	16	1	129	4	9
9	1/1/2013	1/1/2015	730	24	2	0	0	0
10								
11								
12								
13	1/2/2013	12/31/2013	363	11	0	363	11	29
14	1/2/2013	1/1/2014	364	11	0	364	11	30
15	1/2/2013	5/10/2014	493	16	1	128	4	8
16	1/2/2013	3/3/2015	790	26	2	60	2	1
17	1/2/2013	12/31/2015	1093	35	2	363	11	29
18	1/2/2013	1/1/2016	1094	35	2	364	11	30
19								
20								

FIGURE 12-10: Counting days, months, and years with DATEDIF.

Figure 12-10 shows some examples of using DATEDIF. column A has start dates. Column B has end dates. Columns C through H contain formulas with DATEDIF. The DATEDIF function uses the start and end dates on each given row, and the interval is labeled at the top of each column, C through H.

Here’s how to use DATEDIF:

1. **Select the cell where you want the results to appear.**
2. **Type** =DATEDIF(**to begin the function entry.**
3. **Click a cell where you have a date or enter its address.**
4. **Type a comma (,).**
5. **Click a cell where you want another date.**

   This date must be the same or greater than the first date from Step 3; otherwise, you get an error.
6. **Type a comma (,).**

7. **Enter an interval.**

   Refer to Table 12-2 for the list of intervals that you can use with the function. Make sure that the interval is enclosed in double quotes.
8. **Type a) and press Enter.**

Understanding How Excel Handles Time

In Chapter 12, I explain how Excel uses a serial number system to work with dates. Well, guess what? The same system is used to work with time. The key difference is that although dates are represented by the integer portion of a serial number, time is represented by the decimal portion.

What does this mean? Consider this: 43466. That is the serial number representation for January 1, 2019. Notice, though, that there is no indication of the time of day. The assumed time is 12 a.m. (midnight), the start of the day. You can, however, represent specific times if needed.

Excel uses the decimal side of the serial number to represent time as a fraction of the 24-hour day. Thus, 12 p.m. (noon) is 0.5, and 6 p.m. is 0.75. Table 13-1 shows some more examples and reveals how dates and time information are combined in a single serial number.

TABLE 13-1 How Excel Represents Time

Date and Time	*Serial Format*
January 1, 2019 12:00 a.m.	43466

January 1, 2019 12:01 a.m.	43466.00069
January 1, 2019 10:00 a.m.	43466.41667
January 1, 2019 12:00 p.m.	43466.5
January 1, 2019 4:30 p.m.	43466.6875
January 1, 2019 10:00 p.m.	43466.91667
January 1, 2019 11:59 p.m.	43466.99931

Time is represented in a decimal value — up to five digits to the right of the decimal point. A value of 0 is the equivalent of 12 a.m. A value of .5 is the equivalent of 12 p.m. — the midpoint of the day. The value of .99931 is the same as the 23rd hour and *start* of the 59th minute. A value of .99999 is the same as the 23rd hour, the 59th minute, and the 59th second — in other words, 1 second before the start of the next day.

Can you represent time without a date? You bet! Use a value less than 1 for this purpose. For example, the serial number 0.75 represents 6 p.m. with no date specified.

Representing time as a serial number provides the same advantages as it does for dates: the ability to add and subtract times. For example, given a date/time serial number, you can create the serial number for the date/time one and a half days later by adding 1.5 to it.

Formatting Time

When you work with time values, you probably need to format cells in your worksheet so the times display in a standard format that people will understand. The decimal numbers don’t make sense to us human folk. To format time, you use the Number tab of the Format Cells dialog box, shown in Figure 13-1.

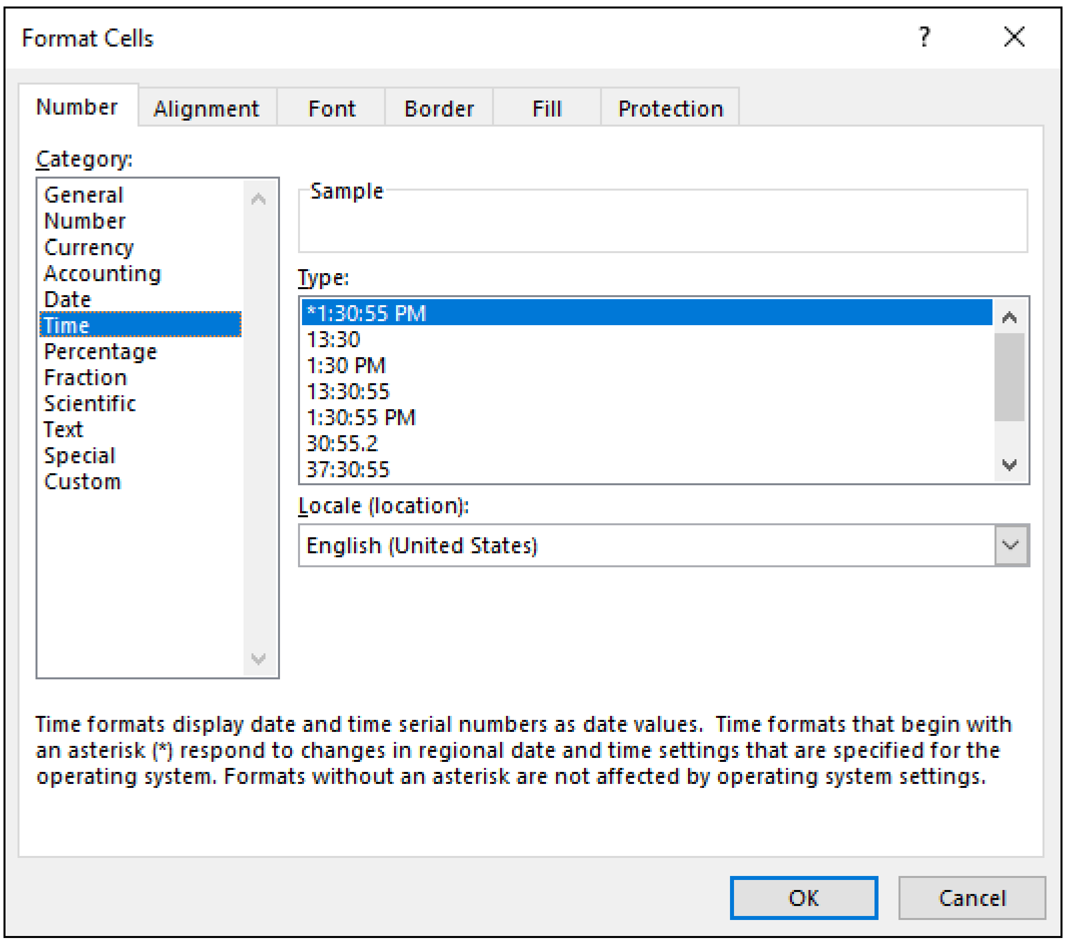


FIGURE 13-1: Using the Format Cells dialog box to specify how time values are displayed.

To format time, follow these steps:

1. **If it’s not already displayed, click the Home tab at the top of the Excel screen.**
2. **Click the small arrow in the bottom-right corner of the Number section.**
   The Format Cells dialog box appears, with the Number tab displayed.
3. **Select Time in the Category list.**
4. **Select an appropriate format in the Type list.**

TIP You can display time in several ways. Excel can format time so that hours in a day range from 1 a.m. to 12 a.m. and then 1 p.m. to 12 p.m. Alternatively, the hour can be between 0 and 23, with values 13 through 23 representing 1 p.m. through 11 p.m. The latter system, known to some as *military time or 24-hour time*, is commonly used in computer systems.

Note that Excel stores a date and time together in a single serial number. Therefore, some of the formatting options in the time and date categories display a complete date and time.

Keeping TIME

You can use the TIME function to combine hours, minutes, and seconds into a single usable value. Figuring out the serial number representation of a particular moment in time isn’t easy. Luckily, the TIME function does this for you. You provide an hour, minute, and second, and TIME tells you the serial value. To do this, follow these steps:

1. **Select the cell where you want the result displayed.**
2. **Type** =TIME(**to begin the function entry.**

3. **Click the cell that has the hour (0–23) or enter such a value.**
4. **Type a comma (,).**
5. **Click the cell that has the minute (0–59) or enter such a value.**
6. **Type a comma (,).**
7. **Click the cell that has the second (0–59) or enter such a value.**
8. **Type a**) **and press Enter.**
   The result is a decimal serial number, or a readable time if the cell is formatted properly.

You should be aware that the minute and second values “wrap.” A value of 60 or greater for seconds wraps to the next minute. For example, 75 seconds is interpreted as 1 minute 15 seconds. Likewise, a minute value of 90 is interpreted as 1 hour 30 minutes. Hours wrap, too. An hour value of 26 is interpreted as 2 a.m.

Converting Text to Time with TIMEVALUE

If you enter a time in a standard format in a cell, Excel recognizes it as a time. It is converted to a serial number, and the cell is assigned the default time format. If you are pasting or importing data from another application, you may encounter times in text format, such as `"2:28 PM"`. You can convert these to a time serial number by using the TIMEVALUE function. Here's how:

1. **Select the cell where you want the result displayed.**
2. **Type** =TIMEVALUE(**to begin the function entry.**
3. **Click the cell that contains the time in text format.**
4. **Type a**) **and press Enter.**

TIP TIMEVALUE works just with text. If the TIMEVALUE function returns the error code `#VALUE#`, it probably means one of two things:

- The time is in a text format that Excel does not recognize, such as `"2:28PM"` (no space before PM) instead of `"2:28 PM"`.
- The time is not actually in text format but is an Excel time serial number formatted to look that way. Change the cell format to General to check.

Deconstructing Time with HOUR, MINUTE, and SECOND

Any moment in time really is a combination of an hour, a minute, and a second. In the preceding section, I show you how the TIME function puts these three components together. In this section, I show you how to break them apart by using the HOUR, MINUTE, and SECOND functions. The worksheet in Figure 13-2 shows a date and time in several rows going down column A. The same dates and times are shown in column B, with a different format. Columns C, D, and E show the hour, minute, and second, respectively, from the values in column A.

C5 =HOUR(A5)

	A	B	C	D	E	F
1	Date and Time	Date and Time	Hour	Minute	Second	
2	In General Format	in Date and Time Format				
3						
4						
5	40375.89935	7/16/2010 9:35:04 PM	21	35	4	
6	40328.55234	5/30/2010 1:15:22 PM	13	15	22	
7	40435.2303	9/14/2010 5:31:38 AM	5	31	38	
8	40185.3452	1/7/2010 8:17:05 AM	8	17	5	
9	40186.33802	1/8/2010 8:06:45 AM	8	6	45	
10	40282.6486	4/14/2010 3:33:59 PM	15	33	59	
11	40496.36667	11/14/2010 8:48:00 AM	8	48	0	
12	40407.839	8/17/2010 8:08:10 PM	20	8	10	
13	40359.13837	6/30/2010 3:19:15 AM	3	19	15	
14	39168.04307	3/27/2007 1:02:01 AM	1	2	1	
15						
16						

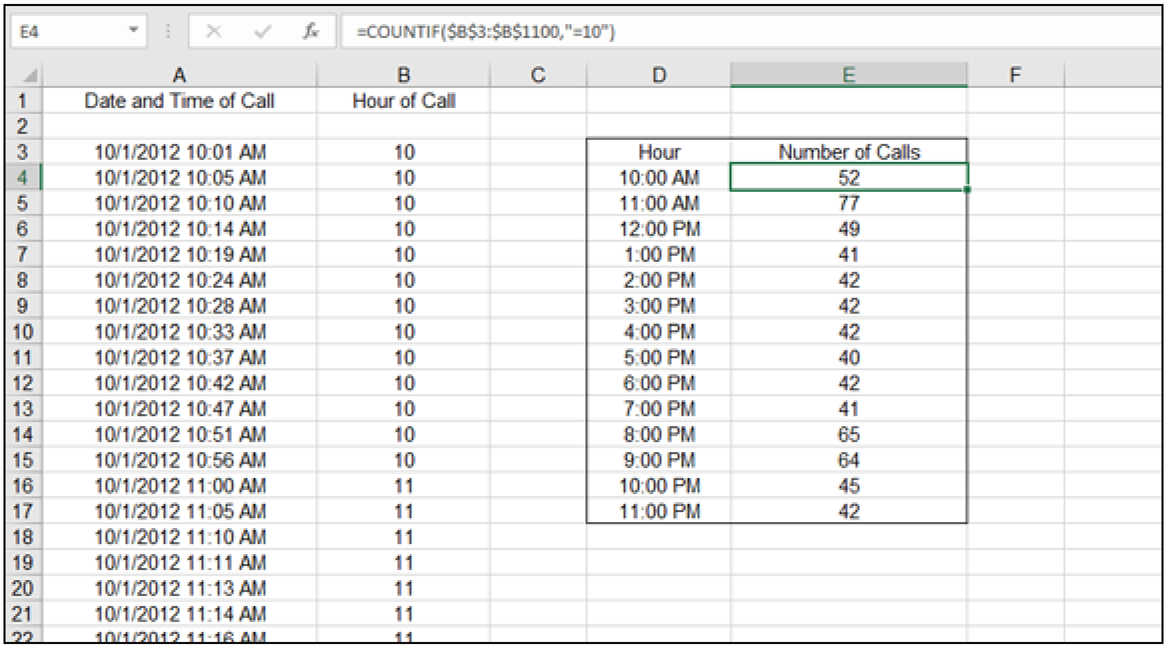
FIGURE 13-2: Splitting time with the HOUR, MINUTE, and SECOND functions.

TIP Note that if the date/time serial number contains a date part, HOUR, MINUTE, and SECOND ignore it; all they care about is the time part.

ISOLATING THE HOUR

Extracting the hour from a time is useful in applications that tally hourly events. A common use of this occurs in call centers. If you've ever responded to an infomercial or a pledge drive, you may realize that a group of workers wait for incoming phone calls such as the one you made. (I hope you got a good bargain.) A common metric in this type of business is the number of calls per hour.

Figure 13-3 shows a worksheet that summarizes calls per hour. Calls have been tracked for October 2012. The incoming call dates and times are listed in column A. In column B, the hour of each call has been isolated with the HOUR function. Column D is a summary of calls per hour over the course of the month.



E4 =COUNTIF($B$3:$B$1100,"=10")

	A	B	C	D	E	F
1	Date and Time of Call	Hour of Call				
2						
3	10/1/2012 10:01 AM	10		Hour	Number of Calls	
4	10/1/2012 10:05 AM	10		10:00 AM	52	
5	10/1/2012 10:10 AM	10		11:00 AM	77	
6	10/1/2012 10:14 AM	10		12:00 PM	49	
7	10/1/2012 10:19 AM	10		1:00 PM	41	
8	10/1/2012 10:24 AM	10		2:00 PM	42	
9	10/1/2012 10:28 AM	10		3:00 PM	42	
10	10/1/2012 10:33 AM	10		4:00 PM	42	
11	10/1/2012 10:37 AM	10		5:00 PM	40	
12	10/1/2012 10:42 AM	10		6:00 PM	42	
13	10/1/2012 10:47 AM	10		7:00 PM	41	
14	10/1/2012 10:51 AM	10		8:00 PM	65	
15	10/1/2012 10:56 AM	10		9:00 PM	64	
16	10/1/2012 11:00 AM	11		10:00 PM	45	
17	10/1/2012 11:05 AM	11		11:00 PM	42	
18	10/1/2012 11:10 AM	11				
19	10/1/2012 11:11 AM	11				
20	10/1/2012 11:13 AM	11				
21	10/1/2012 11:14 AM	11				
22	10/1/2012 11:16 AM	11				

FIGURE 13-3: Using the HOUR function to summarize results.

In Figure 13-3, the values in column D are calculated by the COUNTIF function. There is a COUNTIF for each hour from 10 a.m. through 11 p.m. Each COUNTIF looks at the range of numbers in column B (the hours) and counts the values that match the criteria. Each COUNTIF uses a different hour value for its criteria. Following is an example:

```
=COUNTIF($B$3:$B$1100,"=16")
```

Here is how to use the HOUR function:

1. **Select the cell where you want the result displayed.**
2. **Type** =HOUR(**to begin the function entry.**
3. **Click the cell that has the full time (or date/time) entry.**
4. **Type a**) **and press Enter.**

   Excel returns a number between 0 and 23.

ISOLATING THE MINUTE

Isolating the minute part of a time is necessary in applications that track activity down to the minute. A timed test is a perfect example. Remember when the teacher would yell, “Pencils down”?

Excel can easily calculate how long something takes by subtracting one time from another. In the case of a test, the MINUTE function helps with the calculation because how long something took in minutes is being figured out. Figure 13-4 shows a list of times it took for students to take a test. All students started the test at 10 a.m. Then, when each student finished, the time was noted. The test *should* have taken a student no more than 15 minutes.

D3 =IF(MINUTE(C3)-MINUTE(B3)<=15,"Yes","No")

	A	B	C	D	E	F
1	Student ID	Start Time	End Time	Finished on Time		
2						
3	S2223	10:00 AM	10:18 AM	No		
4	G7854	10:00 AM	10:12 AM	Yes		
5	A4973	10:00 AM	10:14 AM	Yes		
6	M4211	10:00 AM	10:20 AM	No		
7	H7840	10:00 AM	10:22 AM	No		
8	G4381	10:00 AM	10:10 AM	Yes		
9	J4009	10:00 AM	10:11 AM	Yes		
10	T5545	10:00 AM	10:15 AM	Yes		
11	W9329	10:00 AM	10:13 AM	Yes		
12	M8050	10:00 AM	10:16 AM	No		
13	S2377	10:00 AM	10:23 AM	No		
14	R1967	10:00 AM	10:14 AM	Yes		
15						
16						

FIGURE 13-4: Calculating minutes elapsed with the MINUTE function.

For each data row, column D contains a formula that subtracts the minute in the end time, in column C, from the start time, in column B. This math operation is embedded in an IF statement. If the result is 15 or less, `Yes` appears in column D; otherwise, `No` appears.

```
=IF(MINUTE(C3)-MINUTE(B3)<=15,"Yes","No")
```

Like the HOUR function, the MINUTE function takes a single time or date/time reference as its argument.

ISOLATING THE SECOND

Isolating the second from a date value is useful in situations in which highly accurate time calculations are needed. In practice, this isn't a common requirement in Excel applications.

Follow these steps to use the SECOND function:

1. **Position the pointer in the cell where you want the results displayed.**
2. **Type** =SECOND(**to begin the function entry.**
3. **Click the cell that has the time value or enter a time value.**
4. **Type a**) **to end the function, and press Enter.**

Finding the Time NOW

Sometimes when you're working in Excel, you need to access the current time. For example, you may be working on a client project and need to know how much time you've spent on it. Use the NOW function when you first open the workbook, and use it again when you're finished. Subtracting one value from the other provides the elapsed time.

Here's how to use the NOW function:

1. **Select the cell where you want the result.**
2. **Type** =NOW( )**.**
3. **Press Enter to end the function.**

TIP You must take one additional step to make the preceding NOW time calculation work. When you get the current time at the start, copy the value and then use Paste Special to paste it back as a value. This strategy prevents the time from constantly updating. You can also do this by selecting the cell, clicking the Formula Bar, and then pressing F9.

NOW provides not just the current time, but also the current date. This is similar to the TODAY function. TODAY returns the current date — without the current time. NOW returns the full current date and time. See Chapter 12 for more information on the TODAY function.

Calculating Elapsed Time Over Days

Each day has 24 hours. Multiplying 24 by 7 tells you that there are 168 hours in a week. How many hours are in a month? This is not as easy to tell. A month may have 28, 29, 30, or 31 days.

Counting elapsed time, in hours, could require a complex algorithm. Although Excel has no single function for this task, you can combine a couple of functions in a formula to get the answer. This is another benefit of the fact that Excel represents date/time values as serial numbers. This makes it easy to calculate the number of hours that have passed between two date/time values.

Figure 13-5 shows a worksheet with start and end date/time values in two columns. A third column shows the calculated number of elapsed hours for each start/end pair.

C9 =(INT(B9)-INT(A9))*24+HOUR(B9)-HOUR(A9)

	A	B	C
1	Start Date and Time	End Date and Time	Count of Elapsed Hours
2			
3	11/15/2012 9:00	11/15/2012 9:30	0
4	11/15/2012 9:00	11/15/2012 21:00	12
5	11/15/2012 9:00	11/15/2012 23:59	14
6	11/15/2012 9:00	11/16/2012 0:00	15
7	11/15/2012 9:00	11/16/2012 2:00	17
8	11/15/2012 9:00	11/16/2012 9:00	24
9	11/15/2012 9:00	11/23/2012 11:30	194
10	11/15/2012 9:00	12/2/2012 15:30	414
11	11/15/2012 9:00	12/30/2012 1:45	1072
12	11/15/2012 9:00	1/15/2013 16:00	1471
13	11/15/2012 9:00	1/18/2013 10:00	1537
14	11/15/2012 9:00	3/15/2013 9:00	2880
15	11/15/2012 9:00	11/15/2013 8:58	8759
16	11/15/2012 9:00	11/15/2013 9:00	8760
17			

FIGURE 13-5: Calculating elapsed time.

In column A and column B are dates and times. These dates and times are really just serial numbers with a decimal portion. Using the INT function, Excel counts the difference in days, even if the span pops over to a new year. Then it uses the HOUR function to calculate the difference of the decimal portion. The formula for the first row is

```
=(INT(B3)-INT(A3))*24 + HOUR(B3) - HOUR(A3)
```

Each successive row has the same formula in column C, but with the cell references pointed to the values on that row. The first part of the formula calculates the difference in days and multiplies this by 24 for the total number of hours in the number of days.

The trick is to correctly calculate the time between the start and end values. The hour portion of both the start and end values is determined with the HOUR function; then one value is subtracted from the other. The result of this subtraction is added to the precalculated number of hours from the count of days. Note that minutes are ignored. Perhaps you can figure out how to modify the formula to take seconds into account!

Testing on One Condition

The IF function is like the Swiss Army knife of Excel functions. Really, it is used in many situations. Often, you can use it with other functions, which I do often in this chapter. IF, structurally, is easy to understand. The function takes three arguments:

- A test that gives a true or false answer. For example, the test `"is the value in cell A5 equal to the value in cell A8"` can have only one of two possible answers, yes or no. In computer talk, that's true or false. This is not a calculation, mind you, but a comparison.
- The data to be returned by the IF function if the test is true.
- The data to be returned by the IF function if the test is false.

Sounds easy enough. Here are some examples:

Function	***Comment***
`=IF(D10>D20, D10, D20)`	If the value in D10 is greater than the value in D20, the value in D10 is returned because the test is true. If the value in D10 is not greater than — that is, smaller or equal to — the value in D20, the value in D20 is returned. If the values in D10 and D20 are equal, the test returns false, and the value in D20 is returned.

`=IF(D10>D20, "Good news!", "Bad news!")`	If the value in D10 is greater than the value in D20, the text “`Good News!`” is returned. Otherwise, “`Bad News!`” is returned.
`=IF(D10>D20, "", "Bad news!")`	If the value in D10 is greater than the value in D20, *nothing* is returned. Otherwise, “`Bad News!`” is returned. Note that the second argument is a pair of empty quotes.
`=IF(D10>D20, "Good news!", "")`	If the value in D10 is greater than the value in D20, “`Good News!`” is returned. Otherwise, *nothing* is returned. Note that the third argument is empty quotes.

TIP An important aspect to note about using IF: letting the second or third argument return nothing. An empty string is returned, and the best way to do this is to place two double quote marks together with nothing in the middle. The result is that the cell containing the IF function remains blank.

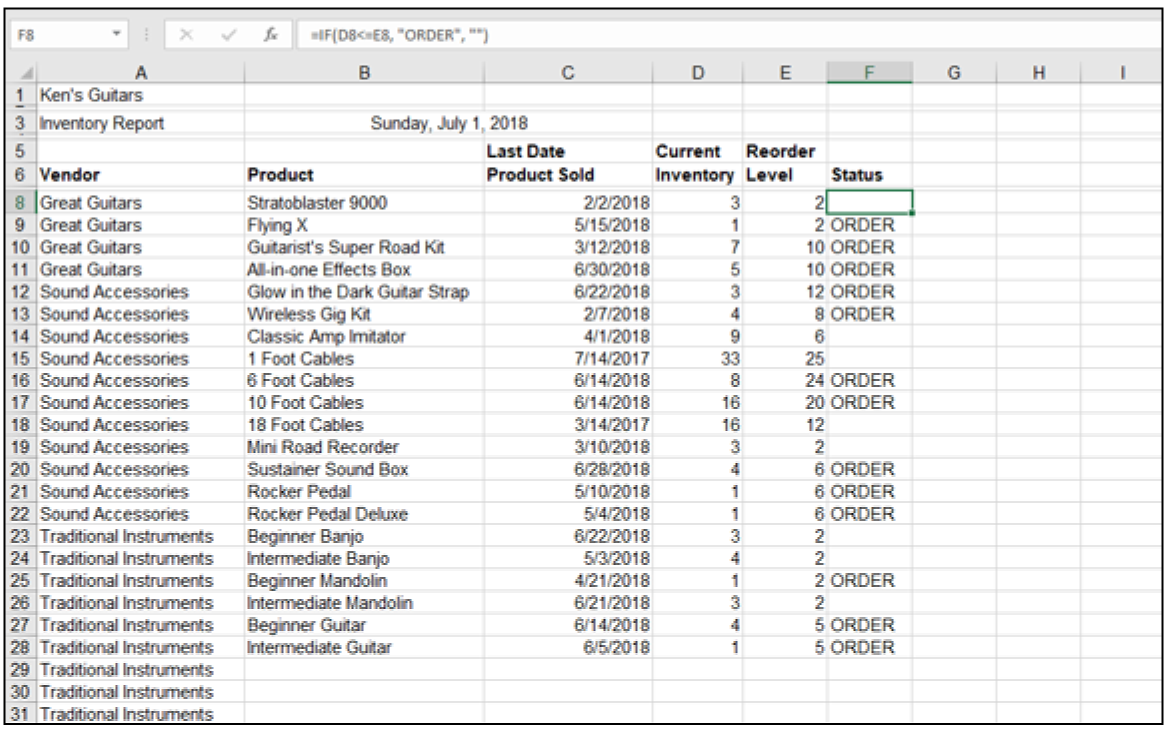
IF, therefore, lets you set up two results to return: one for when the test is true and another for when the test is false. Each result can be a number, some text, a function or formula, or even blank.

As you see in the preceding example, a common use of IF is to see how two values compare and return either one value or the other, depending on how you set up the test in the first argument.

IF is often used as a validation check to prevent errors. Suppose that you have a financial worksheet that uses a variable percentage in its calculations. The user must enter this percentage each day, but it must never be greater than 10 percent. To prevent the chance of errors, you could use the IF function to display an error message in the adjacent cell if you mistakenly enter a value outside the permitted range. Assuming that the percentage is entered in cell A3, here's the required IF function:

```
=IF(A3>.1, "ERROR: the % in A3 IS TOO LARGE", "")
```

Figure 14-1 shows how IF can be put to good use in a business application. A fictitious store shop — Ken's Guitars (kinda snappy, don't you think?) — keeps tabs on inventory in an Excel worksheet.



F8 =IF(D8<=E8, "ORDER", "")

	A	B	C	D	E	F
1	Ken's Guitars					
3	Inventory Report	Sunday, July 1, 2018				
5			Last Date	Current	Reorder	
6	Vendor	Product	Product Sold	Inventory	Level	Status
8	Great Guitars	Stratoblaster 9000	2/2/2018	3	2	
9	Great Guitars	Flying X	5/15/2018	1	2	ORDER
10	Great Guitars	Guitarist's Super Road Kit	3/12/2018	7	10	ORDER
11	Great Guitars	All-in-one Effects Box	6/30/2018	5	10	ORDER
12	Sound Accessories	Glow in the Dark Guitar Strap	6/22/2018	3	12	ORDER
13	Sound Accessories	Wireless Gig Kit	2/7/2018	4	8	ORDER
14	Sound Accessories	Classic Amp Imitator	4/1/2018	9	6	
15	Sound Accessories	1 Foot Cables	7/14/2017	33	25	
16	Sound Accessories	6 Foot Cables	6/14/2018	8	24	ORDER
17	Sound Accessories	10 Foot Cables	6/14/2018	16	20	ORDER
18	Sound Accessories	18 Foot Cables	3/14/2017	16	12	
19	Sound Accessories	Mini Road Recorder	3/10/2018	3	2	
20	Sound Accessories	Sustainer Sound Box	6/28/2018	4	6	ORDER
21	Sound Accessories	Rocker Pedal	5/10/2018	1	6	ORDER
22	Sound Accessories	Rocker Pedal Deluxe	5/4/2018	1	6	ORDER
23	Traditional Instruments	Beginner Banjo	6/22/2018	3	2	
24	Traditional Instruments	Intermediate Banjo	5/3/2018	4	2	
25	Traditional Instruments	Beginner Mandolin	4/21/2018	1	2	ORDER
26	Traditional Instruments	Intermediate Mandolin	6/21/2018	3	2	
27	Traditional Instruments	Beginner Guitar	6/14/2018	4	5	ORDER
28	Traditional Instruments	Intermediate Guitar	6/5/2018	1	5	ORDER
29	Traditional Instruments					
30	Traditional Instruments					
31	Traditional Instruments					

FIGURE 14-1: Keeping an eye on inventory at the guitar shop.

Column D shows the inventory levels, and column E shows the reorder levels. It works this way: When a product's inventory level is the same or less than the reorder level, it is time to order more of the product. (I don't know about you, but I love the thought of being surrounded by a bunch of guitars!) The cells in column F contain a formula.

The formula in cell F8 is `=IF(D8<=E8,"ORDER","")`. It says that if the number of Stratoblaster 9000 guitars in stock is the same or less than the reorder level, return `Order`. If the number in stock is greater than the reorder level, return nothing. Nothing is returned because three are in stock and the reorder level is two. In the next row, the number of Flying Xs is equal to the reorder level; therefore, cell F9 displays `Order`.

Using IF is easy. Follow these steps:

1. **Enter two values in a worksheet.**
   These values should have some meaning to you, such as the inventory levels example in Figure 14-1.
2. **Click the cell where you want the result to appear.**
3. **Type** =IF(**to start the function.**
4. **Decide what test you want to perform.**
   You can see whether the two values are equal; whether one is larger than the other; whether subtracting one from the other is greater than, equal to, or less than 0; and so on. For example, to determine whether the first value equals the second value, click the first cell (or enter its address), enter an equal sign (=), and then click the second cell (or enter its address).
5. **Type a comma (,).**
6. **Enter the result that should appear if the test is *true*.**
   REMEMBER For example, enter **"The values are equal"**. Text must be enclosed in quotes.
7. **Type a comma (,).**
8. **Enter the result that should appear if the test is *false*.**
   For example, enter **"The values are not equal"**.

9. **Type a) and press Enter.**

The IF function can do a whole lot more. Nested IF functions give you a lot more flexibility in performing tests on your worksheet data. A bit of perseverance is necessary to get through this. *Nested* means that you can place an IF function inside another IF function. That is, the inner IF is placed where the true or false argument in the outer IF goes (or even use internal IFs for both of the arguments). Why would you do this?

The other night, we were deciding where to go for dinner. We were considering Italian and decided that if we went to an Italian place and it served manicotti, we would have manicotti. Otherwise, we decided to eat pizza.

Logically, this decision looks like this:

```
If the restaurant is Italian, then
If the restaurant serves manicotti, then
we will have manicotti
else
we will have pizza
```

TECHNICAL STUFF This looks a lot like programming code. I have left out the `End If` statements on purpose to prevent confusion because the IF function has no equivalent value.

That's it! Make note that the inner IF statement has a result for both the true and false possibilities. The outer IF does not. Here is the structure as nested Excel IF statements:

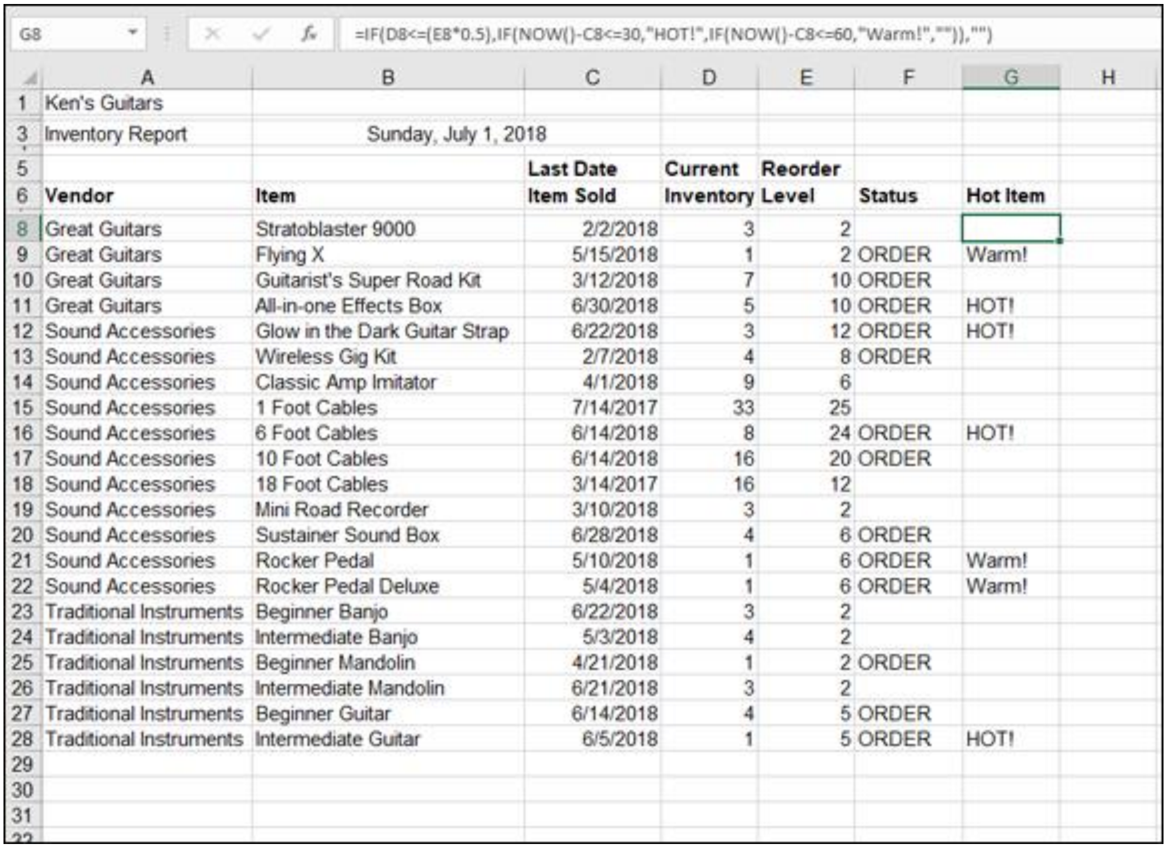
```
=IF(Restaurant=Italian, IF(Restaurant serves
manicotti, "manicotti", "pizza"), "")
```

If the restaurant were not Italian, it wouldn't matter what we ate (as indicated by the third argument of the outer IF being empty).

TIP You can nest up to 64 IF statements, although things are likely to get very complicated once you go beyond 4 or 5.

You can apply a nested IF statement to increase the sophistication of the inventory worksheet from Figure 14-1. Figure 14-2 has an additional column: Hot Item. A Hot Item can take three forms:

- If the inventory level is half or less of the reorder level and the last sale date is within the last 30 days, this is a *Hot Item*. The point of view is that in 30 days or less the stock sold down to half or less than the reorder level. This means that the inventory is turning over at a fast pace.
- If the inventory level is half or less of the reorder level and the last sale date is within the last 31–60 days, this is a *Warm Item*. The point of view is that in 31–60 days the stock sold down to half or less than the reorder level. This means that the inventory is turning over at a medium pace.
- If neither of the preceding two conditions is met, the item is not assigned any special status.



G8 =IF(D8<=(E8*0.5),IF(NOW()-C8<=30,"HOT!",IF(NOW()-C8<=60,"Warm!","")),"")

	A	B	C	D	E	F	G	H
1	Ken's Guitars							
3	Inventory Report	Sunday, July 1, 2018						
5			**Last Date**	**Current**	**Reorder**			
6	**Vendor**	**Item**	**Item Sold**	**Inventory**	**Level**	**Status**	**Hot Item**	
8	Great Guitars	Stratoblaster 9000	2/2/2018	3	2			
9	Great Guitars	Flying X	5/15/2018	1	2	ORDER	Warm!	
10	Great Guitars	Guitarist's Super Road Kit	3/12/2018	7	10	ORDER		
11	Great Guitars	All-in-one Effects Box	6/30/2018	5	10	ORDER	HOT!	
12	Sound Accessories	Glow in the Dark Guitar Strap	6/22/2018	3	12	ORDER	HOT!	
13	Sound Accessories	Wireless Gig Kit	2/7/2018	4	8	ORDER		
14	Sound Accessories	Classic Amp Imitator	4/1/2018	9	6			
15	Sound Accessories	1 Foot Cables	7/14/2017	33	25			
16	Sound Accessories	6 Foot Cables	6/14/2018	8	24	ORDER	HOT!	
17	Sound Accessories	10 Foot Cables	6/14/2018	16	20	ORDER		
18	Sound Accessories	18 Foot Cables	3/14/2017	16	12			
19	Sound Accessories	Mini Road Recorder	3/10/2018	3	2			
20	Sound Accessories	Sustainer Sound Box	6/28/2018	4	6	ORDER		
21	Sound Accessories	Rocker Pedal	5/10/2018	1	6	ORDER	Warm!	
22	Sound Accessories	Rocker Pedal Deluxe	5/4/2018	1	6	ORDER	Warm!	
23	Traditional Instruments	Beginner Banjo	6/22/2018	3	2			
24	Traditional Instruments	Intermediate Banjo	5/3/2018	4	2			
25	Traditional Instruments	Beginner Mandolin	4/21/2018	1	2	ORDER		
26	Traditional Instruments	Intermediate Mandolin	6/21/2018	3	2			
27	Traditional Instruments	Beginner Guitar	6/14/2018	4	5	ORDER		
28	Traditional Instruments	Intermediate Guitar	6/5/2018	1	5	ORDER	HOT!	
29								
30								
31								

FIGURE 14-2: Looking for hot inventory items.

There are Hot Items, and there are Warm Items. Both must meet the common criterion that the inventory is 50 percent or less of the reorder level. Only after this first condition is met does the second criterion — the number of days since the last order — come into play. Sounds like a nested IF to me! Here is the formula in cell G8:

```
=IF(D8<=(E8]]>×0.5),IF(NOW()-
C8<=30,"HOT!",IF(NOW()-C8<=60,"Warm!","")),"")
```

Okay, take a breath. I leave no Excel user behind!

The outer IF tests whether the inventory in column D is equal to or less than half (50 percent) of the reorder level. The piece of the formula that does that is `=IF(D8<=(E8×0.5)`. This test, of course, produces a true or false answer. If it is false, the false part of the outer IF is taken (which is just an empty string found at the end of the formula: `,"")`).

That leaves the whole middle part to wade through. Stay with it!

If the first test is true, the true part of the outer IF is taken. It just so happens that this true part is another IF function:

```
IF(NOW()-C8<=30,"HOT!",IF(NOW()-C8<=60,"Warm!",""))
```

The first argument of the inner IF tests whether the number of days since the last order date (in column C) is less than or equal to 30. You do this by subtracting the last order date from `today`, as obtained from the NOW function.

If the test is true, and the last order date is within the last 30 days, `HOT!` is returned. A hot seller indeed! If the test is false … wait, what's this? Another IF function! Yes: an IF inside an IF inside an IF. If the number of days since the last order date is greater than 30, the next nested IF tests whether the number of days is within the last 60 days:

```
IF(NOW()-C8<=60
```

If this test is true, `Warm!` is returned. If the test is false, nothing is returned.

A few key points about this triple-level IF statement:

- The IF that tests whether the number of elapsed days is 30 or fewer has a value to return if true (`HOT!`) and a value to return for false (whatever is returned by the next nested IF).
- The outer IF and the innermost IF return nothing when their test is false.
- On the surface, the test for 60 or fewer days *also would catch a date that is 30 days or fewer since the last order date*. This is not really what is meant to be. The test should be whether the number of elapsed days is 60 or fewer *but more than 30*. You do not have to actually spell it out this way, because the formula got to the point of testing for the 60-day threshold only because the 30-day threshold already failed. Gotta watch out for these things!

Choosing the Right Value

The CHOOSE function is ideal for converting a value to a *literal*. In plain-speak, this means turning a number, such as 4, into a word, such as April. CHOOSE takes up to 30 arguments. The first argument acts as key to the rest of the arguments. In fact, the other arguments do not get processed per se by the function. Instead, the function looks at the value of the first argument and, based on that value, returns one of its other arguments.

The first argument must be, or evaluate to, a number. This number in turn indicates which of the following arguments to return. For example, the following returns `Two`:

```
=CHOOSE(2, "One", "Two", "Three")
```

TIP The first argument is the number `2`. This means that the function will return the second argument in the list of arguments *following* the first argument. But watch out — this is not the same as returning the second argument of the function. It means to return the second argument, counting from the second argument.

Figure 14-3 shows a useful example of CHOOSE. Suppose that you have a column of months that are in the numerical form (1 through 12). You need to have these displayed as the month names (January through December). CHOOSE to the rescue!

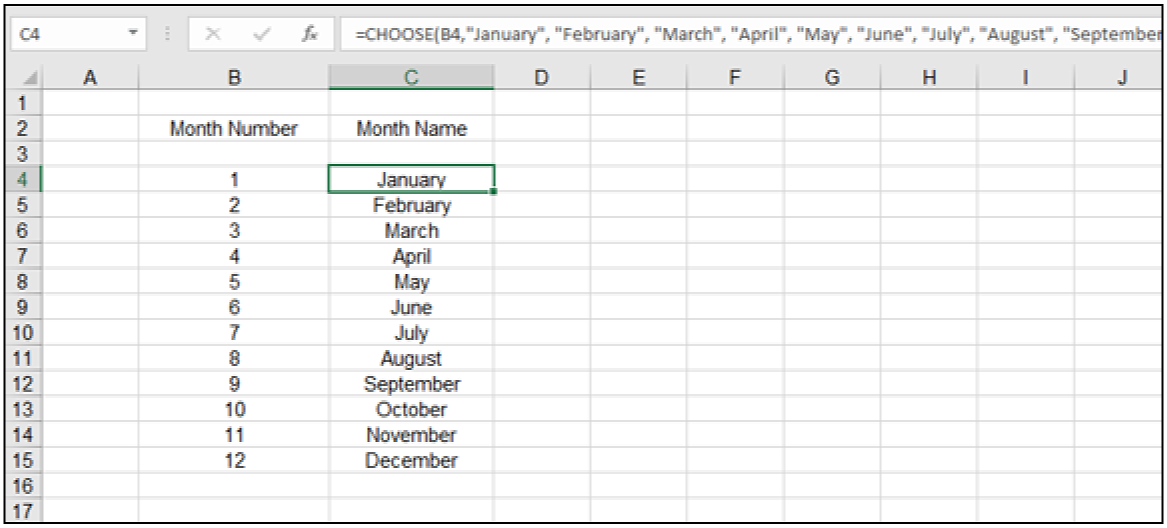


FIGURE 14-3: Choosing what to see.

Cells C4:C15 contain formulas with the CHOOSE function. The formula in cell C4 follows:

```
=CHOOSE(B4,"January", "February", "March", "April",
"May", "June", "July", "August", "September",
"October", "November", "December")
```

Cell B4 contains the value `1`, so the first argument starting in the list of possible returned strings (that is, `"January"`) is returned.

TIP CHOOSE is most often used to return meaningful text that relates to a number, such as returning the name of a month from its numeric value. But CHOOSE is not restricted to returning text strings. You can use it to return numbers.

Try it yourself! Here's how:

1. **Enter a list of numeric values in a worksheet column.**

   These values should all be small, such as 1, 2, 3, and so on.
2. **Click the cell to the right of the first value.**

3. **Type** =CHOOSE(**to start the function.**
4. **Click the cell to the left (the one that has the first value) or enter its address.**
5. **Type a comma (,).**
6. **Enter a list of text strings that each have an association with the numbers you entered in Step 1.**
   Each text string should be in double quotes and separated with commas (for example, **"January", "February", "March"**).
7. **Type a**) **and press Enter.**
   The cell to the right of the first item displays the returned text.
8. **Use the fill handle from the first cell with the formula, and drag the formula down to all the other cells adjacent to list entries.**

Let's Be Logical

I once worked on a grammar problem that provided a paragraph with no punctuation and asked that the punctuation be added:

That that is is not that that is not is not that it it is

The answer follows:

That that is, is not that that is not. Is not that it? It is.

So true! That that is, such as an apple, is not that that is not, such as an orange. (Is your head spinning yet?) These logic operators help you work with your data and also be like Mr. Spock. That's a logical win-win!

NOT

NOT is a logical operator. It is used to reverse a logical value, turning true to false or false to true.

Type this formula in a cell:

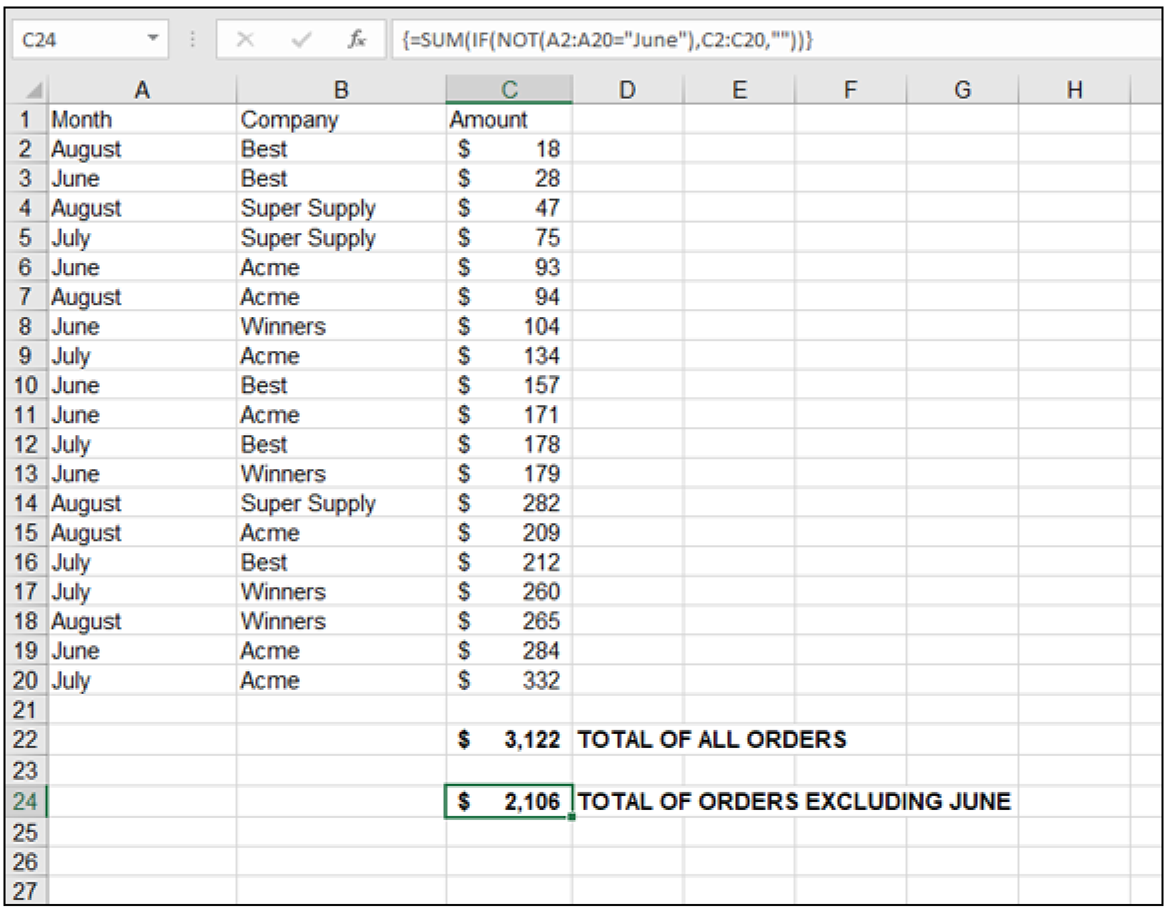
```
= 5 + 5 = 10
```

The result is the word `TRUE`. Makes sense. The math checks out. Now try this:

```
=NOT(5 + 5 = 10)
```

What happens? The word `FALSE` is returned.

The NOT function provides greater flexibility when you're designing the test portion of a SUMIF function (which you read about in Chapter 8). Sometimes, it is easier to define what you want omitted from the sum than to define what you want included. Figure 14-4 shows an example of how this works. The task is to sum up all orders except those in June. Column A lists the months, and column C lists the amounts.



C24 | {=SUM(IF(NOT(A2:A20="June"),C2:C20,""))}

	A	B	C	D	E	F	G	H
1	Month	Company	Amount					
2	August	Best	$ 18					
3	June	Best	$ 28					
4	August	Super Supply	$ 47					
5	July	Super Supply	$ 75					
6	June	Acme	$ 93					
7	August	Acme	$ 94					
8	June	Winners	$ 104					
9	July	Acme	$ 134					
10	June	Best	$ 157					
11	June	Acme	$ 171					
12	July	Best	$ 178					
13	June	Winners	$ 179					
14	August	Super Supply	$ 282					
15	August	Acme	$ 209					
16	July	Best	$ 212					
17	July	Winners	$ 260					
18	August	Winners	$ 265					
19	June	Acme	$ 284					
20	July	Acme	$ 332					
21								
22			**$ 3,122**	**TOTAL OF ALL ORDERS**				
23								
24			**$ 2,106**	**TOTAL OF ORDERS EXCLUDING JUNE**				
25								
26								
27								

FIGURE 14-4: Being selective with summing.

Cell C22 calculates the full sum with this formula:

```
=SUM(C2:C20)
```

The total is $3,122.

On the other hand, the formula in cell C24 is

```
{=SUM(IF(NOT(A2:A20="June"),C2:C20,""))}
```

This says to sum values in the range C2:C20 only when the associated month in column A is not June.

TIP Note that this formula is an array formula. When entered, the entry was completed by pressing Ctrl+Shift+Enter instead of just plain Enter. See Chapter 3 for more information on array formulas.

AND AND OR

Next are the AND and OR functions. AND and OR both return a single logical answer, true or false, based on the values of two or more logical tests (such as the way IF works):

- The AND function returns true if *all* the tests are true. Otherwise, false is returned.
- The OR function returns true if *any* one or more of the tests is true. Otherwise, false is returned.

The syntax of both AND and OR is to place the tests inside the function's parentheses; the tests themselves are separated by commas. Here is an example that returns true if the value in cell D10 equals 20 *or* 30 *or* 40:

```
=OR(D10=20,D10=30,D10=40)
```

Check out how this works. In Figure 14-3 earlier in this chapter, you see how you can use the CHOOSE function to return the name of a month derived from the number of the month. That works okay, but what if you type a wrong number or even a non-numerical value as the first argument in CHOOSE?

As is, the CHOOSE function shown in Figure 14-3 returns the `#VALUE!` error if the first argument is a number greater or less than the number of arguments (not counting the first argument). So as is, the function only works when the first argument evaluates to a number between 1 and 12. If only life were that perfect!

The next-best thing, then, is to include a little validation in the function. Think this through. Both statements must be true:

- The first argument must be greater than 0.
- The first argument must be less than 13.

The formula that uses CHOOSE needs an overhaul, and here it is, courtesy of the AND function:

```
=IF(AND(B4>0,B4<13),CHOOSE(B4,"January",
"February", "March", "April", "May", "June",
"July", "August", "September", "October",
"November", "December"),"That is not a month!")
```

Wow, that's a mouthful (or a cell-full). The CHOOSE function is still there, but it is nested inside an IF. The IF has a test (which I explain shortly). If the test returns true, the CHOOSE function returns the name of the month. If the IF test returns false, a simple `That is not a month!` message is returned. Figure 14-5 shows this in action.

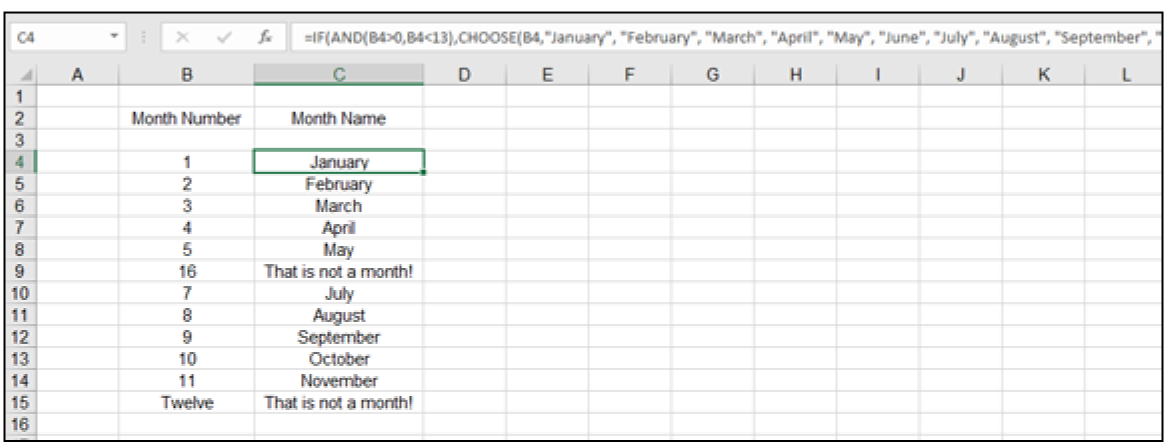


FIGURE 14-5: Being logical about what to choose.

The test part of the IF function is this:

```
AND(B4>0,B4<13)
```

The AND returns true if the value in Cell B4 is both greater than 0 *and* less than 13. When that happens, the true part of the IF statement is taken, which uses the CHOOSE statement to return a month name. Otherwise, the `"That is not a month!"` statement is displayed. In Figure 14-5, this is just what happens in cells C9 and C15, which look at the data values in cells B9 and B15, respectively.

Can you figure out how to accomplish the same thing by using OR instead of AND? Think for a moment and then look at the answer:

```
=IF(OR(B4>1,B4<12),"That is not a month!",
CHOOSE(B4,"January", "February", "March", "April",
"May", "June", "July", "August", "September",
"October", "November", "December"))
```

REMEMBER AND returns true when every condition is true. OR returns true when any condition is true.

Here's how to use AND or OR:

1. **Click a cell where you want the result to appear.**
2. **Type either** =AND(**or** =OR(**to start the function.**
3. **Enter one or more logical tests.**

   A test typically is a comparison of values in two cells or an equation, such as A1 = B1 or A1 + B1 = C1. Separate the tests with commas.
4. **Type a**), **and press Enter.**

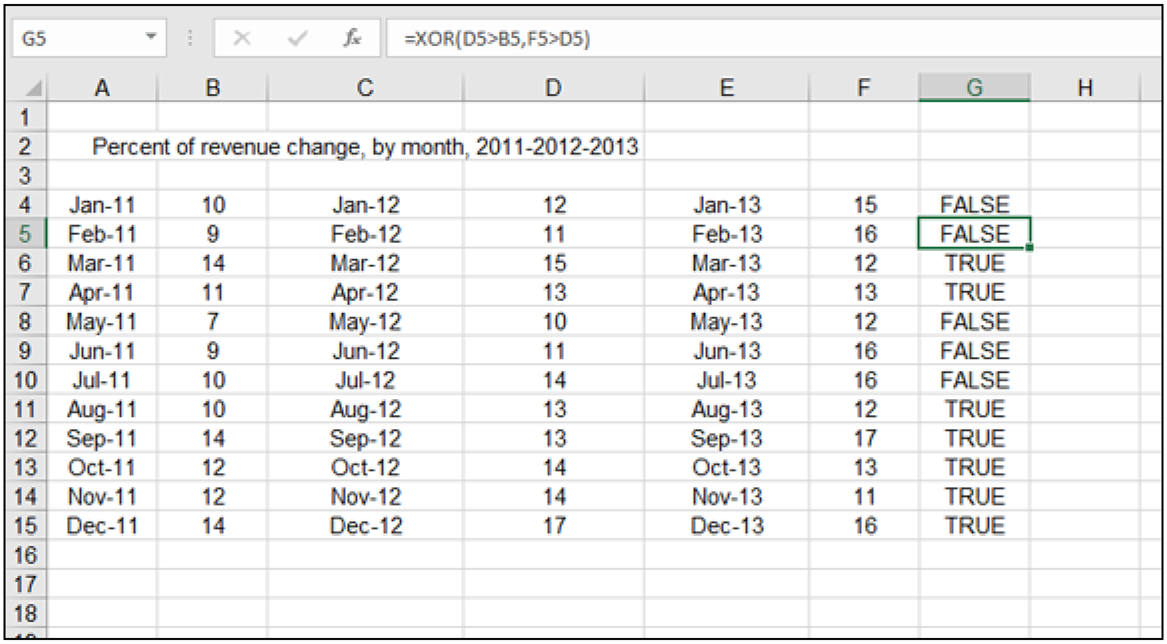
   If you enter the AND function, the result is true if all the tests are true. If you enter the OR function, the result is true if at least one of the tests is true.

XOR

OR, shown earlier in this chapter, returns TRUE when at least one condition is true. This makes sense, considering that it's the word *or* — as in "this or that." So what does XOR mean? XOR is an acronym for Exclusive Or. XOR returns TRUE or FALSE depending on the number of conditions. This can be confusing! (Please don't shoot me; I'm just the messenger!)

Seriously, you really must wonder what is going on here. I have always found that the best way to think of XOR is that it is a variation of OR. Only one condition needs to be true in an OR test to have it return TRUE. With XOR, each condition's logical value and the number of conditions both determine if XOR returns TRUE or FALSE. Believe it or not, there are useful applications for this.

Figure 14-6 shows a worksheet that compares the percentage of change in revenue month to month over a 3-year span. For example, Feb 2011 had an increase of 9 percent from the same period in the previous year. Feb 2012 has an increase of 11 percent from the same period in 2011, and Feb 2013 had an increase of 16 percent from Feb 2012. In a nutshell, revenue has been increasing each February compared with the previous February. This is the type of news that makes business manager types all tingly and ready to go out dancing.



G5 =XOR(D5>B5,F5>D5)

	A	B	C	D	E	F	G	H
1								
2	Percent of revenue change, by month, 2011-2012-2013							
3								
4	Jan-11	10	Jan-12	12	Jan-13	15	FALSE	
5	Feb-11	9	Feb-12	11	Feb-13	16	FALSE	
6	Mar-11	14	Mar-12	15	Mar-13	12	TRUE	
7	Apr-11	11	Apr-12	13	Apr-13	13	TRUE	
8	May-11	7	May-12	10	May-13	12	FALSE	
9	Jun-11	9	Jun-12	11	Jun-13	16	FALSE	
10	Jul-11	10	Jul-12	14	Jul-13	16	FALSE	
11	Aug-11	10	Aug-12	13	Aug-13	12	TRUE	
12	Sep-11	14	Sep-12	13	Sep-13	17	TRUE	
13	Oct-11	12	Oct-12	14	Oct-13	13	TRUE	
14	Nov-11	12	Nov-12	14	Nov-13	11	TRUE	
15	Dec-11	14	Dec-12	17	Dec-13	16	TRUE	
16								
17								
18								

FIGURE 14-6: Using XOR to find where data is not what was expected.

The revenue percentage change is shown for all 12 months over the 3-year span. The XOR is put in column G and is used with the two conditions in the same row. In other words, cell G5 contains an XOR that has two conditions — a test to see whether the percentage change from Feb 2011 to Feb 2012 is an increase and a test to see whether the percentage change of Feb 2012 to Feb 2013 is also an increase. The revenue percentage change has been increasing — good news — and the XOR returns the word `FALSE`. When a manager looks over this report, he can scan column G, and if he sees the word `FALSE`, that's a signal to ignore. The question is, did revenue dip somewhere along the 3 years, in February? The answer is no — that is, the answer is `FALSE`. The manager skips looking any further at that line. The formula in cell G5 looks like this:

```
=XOR(D5<B5, F5>D5)
```

To the manager's eye, other lines in the worksheet are worthy of attention. For example, cell G15 contains an XOR that looks at the revenue change for December over the 3-year period. Sure enough, the revenue percentage change went up and then down — not good news. The XOR function returns TRUE.

Finding Where the Data Is

You can find a plethora of things with the ADDRESS, ROW, ROWS, COLUMN, COLUMNS, and OFFSET functions.

ADDRESS

The ADDRESS function takes a row number and a column number as arguments and returns a standard cell reference (cell address). For example, if you pass the row number 4 and the column number 3, the function returns C4. ADDRESS can return an absolute or relative reference in either of Excel's two reference formats. Before I get to the details, I review the differences between absolute and relative cell references:

- A *relative reference* is expressed as just the column letter and row number (for example, M290). When you copy a formula that contains a relative cell reference, the reference — the row number and the column letter — is adjusted to reflect the location to which you copied the formula.
- An *absolute reference* has a dollar sign in front of the column letter and the row number (for example, $M$290). When you copy a formula that contains an absolute cell reference, the reference does not change.
- A *mixed reference* has a dollar sign in front of the column letter or the row number (for example, $M290 or M$290). When you copy a formula that contains a mixed cell reference, the part of the reference with the dollar sign does not change, but the other part does.

Figure 14-7 shows a worksheet in which entering a formula with a relative cell reference causes a problem. Totals are the result of adding the tax to the amount. The tax is a percentage (0.075) for a 7.5 percent tax rate. This percentage is in cell C1 and is referenced by the formulas. The first formula that was entered is in cell C7 and looks like this: `=B7*(1 + C1)`.

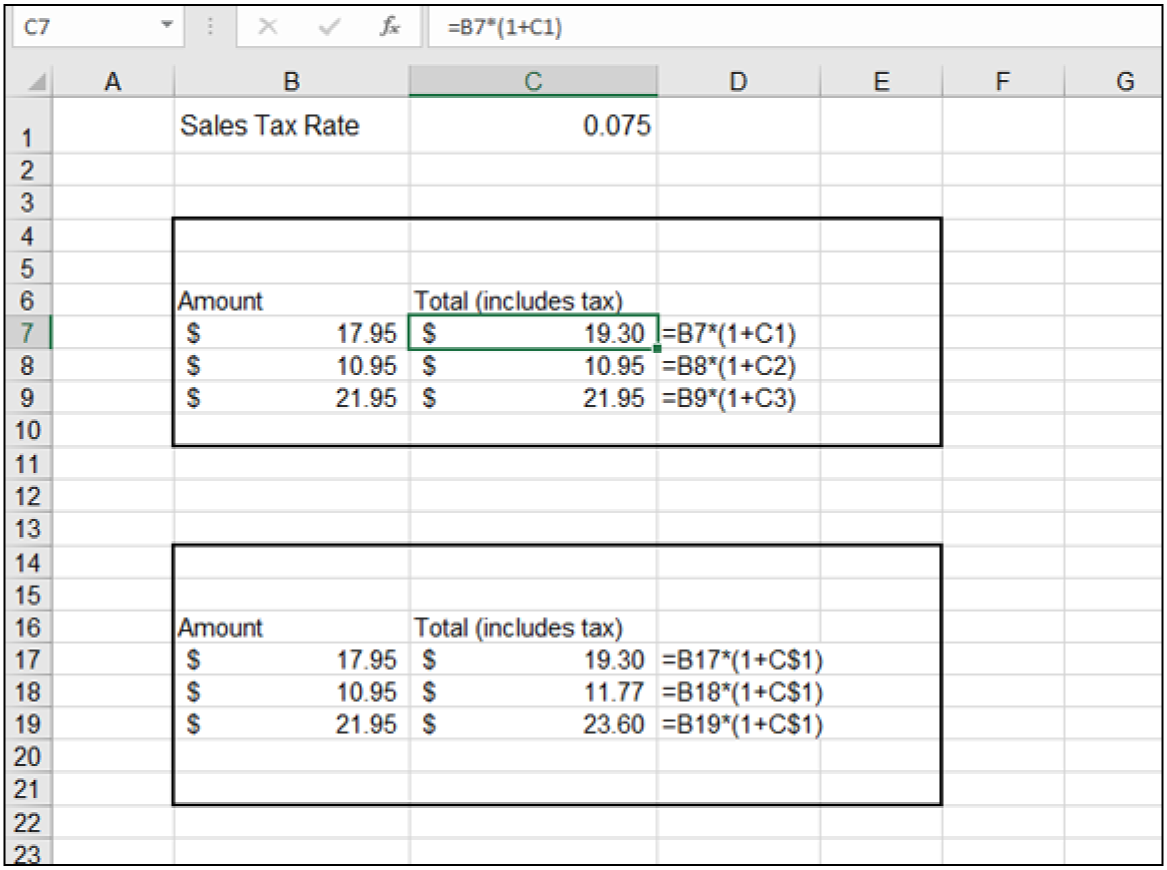


FIGURE 14-7: Changing a reference from relative to absolute.

The formula in cell C7 works correctly. It references cell C1 to calculate the total. But if you use the fill handle to copy the formula from cell C7 to cells C8 and C9, there's a problem. The reference to cell C1 changed to cell C2 and C3. Because these cells are empty, the results in cells C8 and C9 are incorrect; they are the same as the amounts to the left. (No tax is added.)

To better understand, column D displays the formulas that are in column C. When the formula in cell C7 was dragged down, the C1 reference changed to C2 in cell C8, and to C3 in cell C9. Often, this is what you want — for Excel to automatically change cell references when a formula is copied. But sometimes, as in this situation, it is *not* what you want. You need an absolute cell reference.

The formula in cell C17 is almost identical to the one in cell C7 except that the reference to cell C1 has been made row absolute by placing a dollar sign in front of the row number. The formula in cell C17 looks like this: `=B17*(1 +`

`C$1)`. When this formula was dragged down into C18 and C19, the reference was not adjusted but stayed pointing at cell C1. Note that in this example, only the row part of the reference is made absolute. That's all that is necessary. You could have made the reference completely absolute by doing this: `=B17*(1 + $C$1)`. The result would be the same, but it's not required in this example.

REMEMBER Put a dollar sign in front of the column letter of a cell reference to create an absolute column reference. Put a dollar sign in front of the row number to create an absolute row reference.

Excel supports two cell reference styles: the good old A1 style and the R1C1 style. You see the A1 style — a column letter followed by a row number — throughout this book (D4 or B2:B10, for example). The R1C1 style uses a numerical system for both the row and the column, such as this: R4C10. In this example, R4C10 means row 4 column 10.

To change the cell reference style, choose File ⇒ Options and check the R1C1 reference style in the Working with Formulas area on the Formulas tab. Using the R1C1 format also forces the columns on the worksheet to display as numbers instead of the lettering system. This is useful when you're working with a large number of columns. For example, column CV positionally is the 100th column. Remembering 100 is easier than remembering CV.

To get back to the ADDRESS function, it takes up to five arguments:

- The row number of the reference
- The column number of the reference
- A number that tells the function how to return the reference. The default is 1, but it can be
  - 1 for full absolute
  - 2 for absolute row and relative column
  - 3 for relative row and absolute column

    - 4 for full relative
- A value of 0 or 1 to tell the function which reference style to use:
    - 0 uses the R1C1 style.
    - 1 (the default if omitted) uses the A1 style.
- A worksheet or external workbook and worksheet reference

Only the first two arguments are required: the row number and column number being addressed. The function returns the specified reference as text. Table 14-1 shows some examples of using the ADDRESS function.

TABLE 14-1 Using the ADDRESS Function

Syntax	***Result***	***Comment***
`=ADDRESS(5,2)`	`$B$5`	Only the column and row are provided as arguments. The function returns a full absolute address.

=ADDRESS(5,2,1)	$B$5	When a 1 is used for the third argument, a full absolute address is returned. This is the same as leaving out the third argument.

`=ADDRESS(5,2,2)`	`B$5`	When a 2 is used for the third argument, a mixed reference is returned, with the column relative and the row absolute.

`=ADDRESS(5,2,3)`	`$B5`	When a 3 is used for the third argument, a mixed reference is returned, with the column absolute and the row relative.
`=ADDRESS(5,2,4)`	`B5`	When a 4 is used for the third argument, a full relative reference is returned.

=ADDRESS(5,2,1,0)	R5C2	When the fourth argument is false, an R1C1-style reference is returned.
=ADDRESS(5,2,3,0)	R[5]C2	This example tells the function to return a mixed reference in the R1C1 style.

=ADDRESS(5,2,1,,"Sheet4")	Sheet4!$B$5	The fifth argument returns a reference to a worksheet or external workbook. This returns an A1-style reference to cell B5 on Sheet 4.
=ADDRESS(5,2,1,0,"Sheet4")	Sheet4!R5C2	This returns an R1C1-style reference to B5 on

Sheet
4.

Use ADDRESS this way:

1. **Click a cell where you want the result to appear.**
2. **Type** =ADDRESS(**to start the function.**
3. **Enter a row number, a comma (**,**), and a column number.**
   You can also enter references to cells where those values are located.
4. **If you want the result to be returned in a mixed or full reference, enter a comma (**,**) and the appropriate number:** 2, 3, **or** 4**.**
5. **If you want the result to be returned in R1C1 style, enter a comma (**,**) and enter** 0**.**
6. **If you want the result to be a reference to another worksheet, enter a comma and put the name of the worksheet in double quote marks.**
   If you want the result to be a reference to an external workbook, enter a comma (**,**) and enter the workbook name and worksheet name together. The workbook name goes in brackets, and the entire reference goes in double quote marks, such as this: `"[Book1]Sheet2"`.
7. **Type a**) **and press Enter.**

TIP Instead of entering a row number and column number directly in ADDRESS, you can enter cell references. However, the values you find in those cells must evaluate to numbers that can be used as a row number and column number.

ROW, ROWS, COLUMN, AND COLUMNS

The ADDRESS function is rarely used on its own. Most often, it is used as part of a more complex formula. A useful example of ADDRESS follows the discussion of ROW, ROWS, COLUMN, and COLUMNS.

ROW and COLUMN are passed a reference to a cell or range and return the row number or the column number, respectively. Sounds simple enough. These functions take a single optional argument. The argument is a reference to a cell or range. The function returns the associated row number or column number. When the reference is a range, it is the first cell of the range (the upper left) that is used by the function.

ROW and COLUMN are particularly useful when the argument is a name (for a named area). When you use ROW or COLUMN without an argument, it returns the row number or column number of the cell the function is in. Here are examples of ROW and COLUMN:

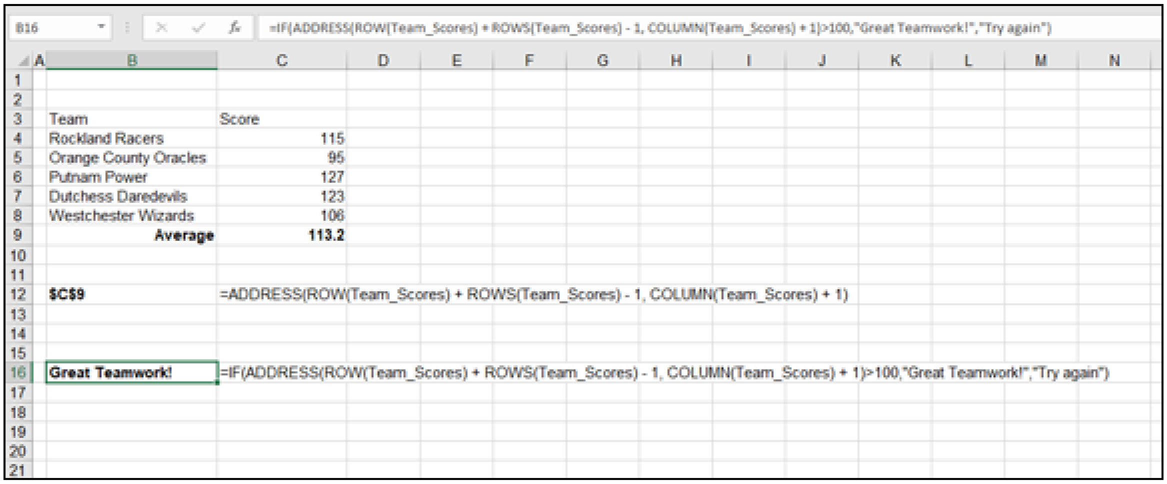
Formula	***Result***
`=ROW(D3)`	3
`=ROW(D3:G15)`	3
`=COLUMN(D3)`	4
`=COLUMN(D3:G15)`	4

`=ROW(Team_Scores)`	The first row of the Team_Scores range
`=COLUMN(Team_Scores)`	The first column of the Team_Scores range

The ROWS and COLUMNS functions (notice that these are now plural), respectively, return the number of rows or the number of columns in a reference:

Formula	***Result***
`=ROWS(Team_Scores)`	Number of rows in the Team_Scores range
`=COLUMNS(Team_Scores)`	Number of columns in the Team_Scores range

Now you are getting somewhere. You can use these functions with ADDRESS to do something useful. Here's the scenario: You have a named range in which the bottom row has summary information, such as averages. You need to get at the bottom row but don't know the actual row number. Figure 14-8 shows this situation. The Team_Scores range is B3:C9. Row 9 contains the average score. You need that value in a calculation, even if another team is added to the list and the row number changes.



B16 =IF(ADDRESS(ROW(Team_Scores) + ROWS(Team_Scores) - 1, COLUMN(Team_Scores) + 1)>100,"Great Teamwork!","Try again")

	A	B	C
1			
2			
3		Team	Score
4		Rockland Racers	115
5		Orange County Oracles	95
6		Putnam Power	127
7		Dutchess Daredevils	123
8		Westchester Wizards	106
9		**Average**	**113.2**
10			
11			
12		**$C$9**	=ADDRESS(ROW(Team_Scores) + ROWS(Team_Scores) - 1, COLUMN(Team_Scores) + 1)
13			
14			
15			
16		**Great Teamwork!**	=IF(ADDRESS(ROW(Team_Scores) + ROWS(Team_Scores) - 1, COLUMN(Team_Scores) + 1)>100,"Great Teamwork!","Try again")
17			
18			
19			
20			
21			

FIGURE 14-8: Using reference functions to find a value.

Cell B12 uses a combination of ADDRESS, ROW, ROWS, and COLUMN to determine the cell address where the average score is calculated. That formula follows:

```
=ADDRESS(ROW(Team_Scores) + ROWS(Team_Scores) - 1,
COLUMN(Team_Scores) + 1)
```

- ROW returns the row number of the first cell of Team_Scores. That row number is 3.
- ROWS returns the number of rows in the named range. That count is 7.

Adding these two numbers is not quite right. A 1 is subtracted from that total to give the last row (9). In this example, you need only COLUMN to get the column number because it understood that the range's second column is the column of scores. In other words, you have no idea how many rows the range has, so ROW and ROWS are both used, but you do know the scores are in the range's second column. This tells you that cell C9 contains the average score. Now what?

Cell B16 contains an IF that uses the address to perform its calculation:

```
=IF(ADDRESS(ROW(Team_Scores) + ROWS(Team_Scores) -
1, COLUMN(Team_Scores) + 1)>100,"Great
Teamwork!","Try again")
```

The IF function tests whether the average score is greater than 100. If it is, the `"Great Teamwork!"` message is displayed. This test is possible because the ADDRESS, ROW, ROWS, and COLUMN functions all help give the IF function the address of the cell where the average score is calculated.

Using ROW, ROWS, COLUMN, or COLUMNS is easy. Here's how:

1. **Click the cell where you want the results to appear.**
2. **Type** =ROW(, =ROWS(, =COLUMN(, **or** =COLUMNS(**to start the function.**
3. **Enter a reference or drag the mouse over an area of the worksheet.**
4. **Type a**) **and press Enter.**

Again, these functions are rarely used alone; they are almost always used in a more complex formula, as in the preceding example.

OFFSET

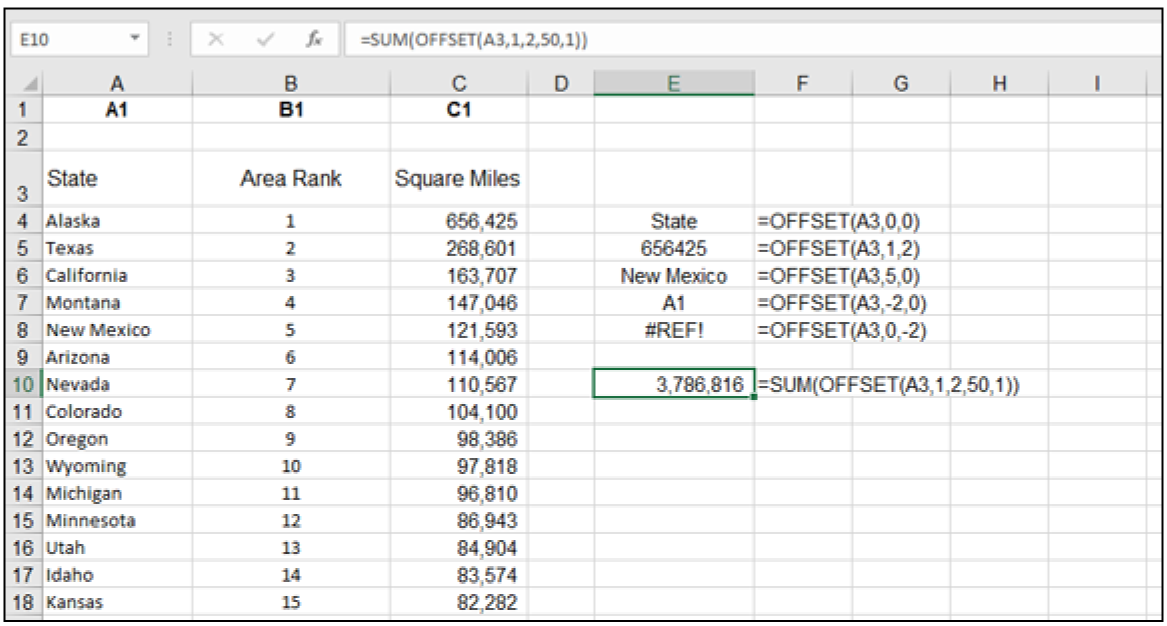
The OFFSET function lets you get the address of the cell that is offset from another cell by a certain number of rows and/or columns. For example, cell E4 is offset from cell B4 by three columns because it is three columns to the right. OFFSET takes up to five arguments. The first three are required:

- **A cell address or a range address:** Named ranges are not allowed.
- **The number of rows to offset:** This can be a positive or negative number. Use 0 for no row offset.
- **The number of columns to offset:** This can be a positive or negative number. Use 0 for no column offset.
- **The number of rows in the returned range:** The default is the number of rows in the reference range (the first argument).

- **The number of columns to return:** The default is the number of columns in the reference range.

If you omit the last two arguments, OFFSET returns a reference to a single cell. If you include a value greater than 1 for either or both, the function's return references a range of the specified size with the top-left cell at the specified offset.

Figure 14-9 shows some examples of using OFFSET. Columns A through C contain a ranking of the states in the United States by size in square miles. Column E shows how OFFSET has returned different values from cells that are offset from cell A3.


E10 =SUM(OFFSET(A3,1,2,50,1))

	A	B	C	D	E	F	G	H	I
1	A1	B1	C1						
2									
3	State	Area Rank	Square Miles						
4	Alaska	1	656,425		State	=OFFSET(A3,0,0)			
5	Texas	2	268,601		656425	=OFFSET(A3,1,2)			
6	California	3	163,707		New Mexico	=OFFSET(A3,5,0)			
7	Montana	4	147,046		A1	=OFFSET(A3,-2,0)			
8	New Mexico	5	121,593		#REF!	=OFFSET(A3,0,-2)			
9	Arizona	6	114,006						
10	Nevada	7	110,567		3,786,816	=SUM(OFFSET(A3,1,2,50,1))			
11	Colorado	8	104,100						
12	Oregon	9	98,386						
13	Wyoming	10	97,818						
14	Michigan	11	96,810						
15	Minnesota	12	86,943						
16	Utah	13	84,904						
17	Idaho	14	83,574						
18	Kansas	15	82,282						

FIGURE 14-9: Finding values by using the OFFSET function.

Some highlights follow:

- Cell E4 returns the value of cell A3 because both the row and column offset is set to 0: `=OFFSET(A3,0,0)`.

- Cell E7 returns the value you find in cell A1 (the value also is `A1`). This is because the row offset is –2. From the perspective of A3, minus two rows is row number 1: `=OFFSET(A3,-2,0)`.
- Cell E8 displays an error because OFFSET is attempting to reference a column that is less than the first column: `=OFFSET(A3,0,-2)`.
- Cell E10 makes use of the two optional OFFSET arguments to tell the SUM function to calculate the sum of the range C4:C53: `=SUM(OFFSET(A3,1,2,50,1))`.

Here's how to use the OFFSET function:

1. **Click a cell where you want the result to appear.**
2. **Type** =OFFSET(**to start the function.**
3. **Enter a cell address or click a cell to get its address.**
4. **Type a comma (,).**
5. **Enter the number of rows you want to offset where the function looks for a value.**
   This number can be a positive number, a negative number, or 0 for no offset.
6. **Type a comma (,).**
7. **Enter the number of columns you want to offset where the function looks for a value.**
   This can be a positive number, a negative number, or 0 for no offset.
8. **Type a) and press Enter.**

OFFSET is another of those functions that can be used alone but is usually used as part of a more complex formula.

Looking It Up

Excel has a neat group of functions that let you extract data from lists and tables. What is a table? A *table* is a dedicated matrix of rows and columns that collectively form a cohesive group of data. Tables usually have labels in the top row or the left column that identify the columns and rows of data. The remainder of the table contains the data itself.

HLOOKUP AND VLOOKUP

The HLOOKUP and VLOOKUP functions extract the data from a particular cell in a table. HLOOKUP starts by searching across the first row of the table to find a value that you specify. When it finds that value, it goes down the column a specified number of rows and returns the value in the target cell. VLOOKUP works the same way except that it searches down the first column of the table and then moves across a specified number of columns.

HLOOKUP takes four arguments, and the first three are required:

- **The value to find in the top row of the table:** This can be text or a number.
- **The address of the table itself:** This is either a range address or a named range.
- **The row offset from the top row:** This is not a fixed row number but rather the number of rows relative from the top row.
- **A true or false value:** If true (or omitted), a partial match is acceptable for Step 1 (see steps below). If false, only an exact match is allowed.

Figure 14-10 shows how HLOOKUP pulls values from a table and displays them elsewhere in the worksheet. This function is quite useful if you need to print a report with a dedicated print area and must include some, but not all, of the data in the table. This example uses the HLOOKUP function to extract the desired data and display it for printing.

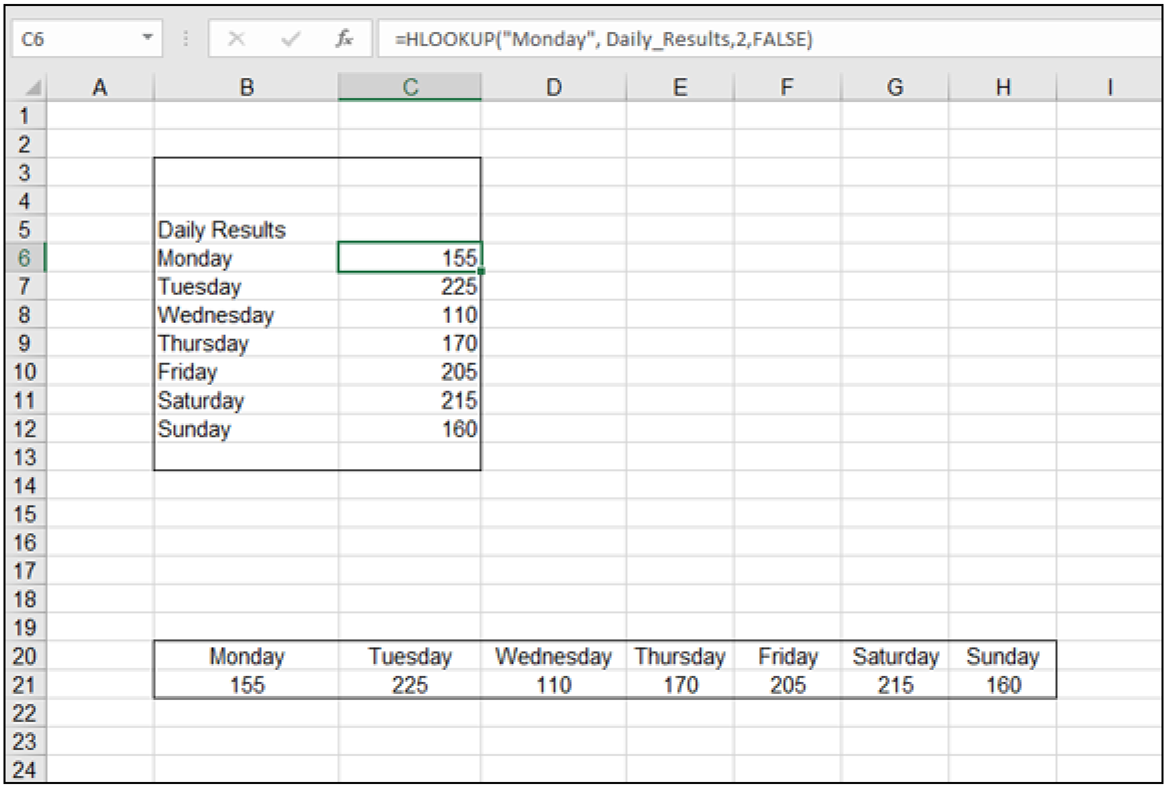


FIGURE 14-10: Using HLOOKUP to locate data in a table.

TIP

Why not just use a cell reference to the table cell that contains the desired data? A cell reference will not return the correct data if the table is moved or if one or more columns are added. With HLOOKUP and VLOOKUP, you know you'll always get data from the correct column or row.

In Figure 14-10, the table is the range B20:H21, which has been assigned the name Daily_Results. Each cell in the range C6:C12 uses HLOOKUP to locate a specific value in the table. For example, cell C6 has this formula:

```
HLOOKUP("Monday",Daily_Results,2,FALSE)
```

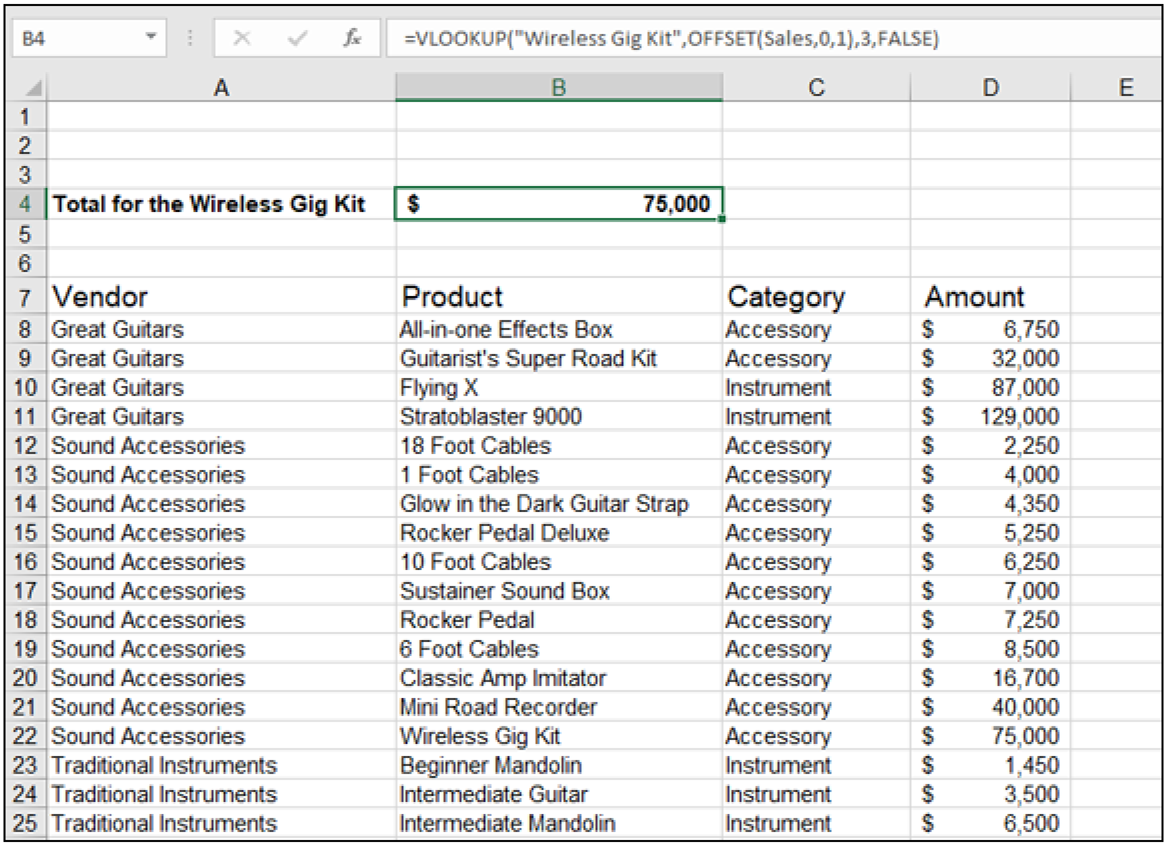
- **The first argument:** Tells the function to search for Monday in the first row of the table.

- **The second argument:** Specifies the table itself by its assigned name.
- **The third argument:** Tells the function to return the data in the second row of the specified column. This table has just two rows, but there is no effective size limit to the table you use with HLOOKUP.
- **The fourth argument:** Specifies that an exact match for Monday must be found. If you set this argument to true or omit it, HLOOKUP finds an approximate match. For approximate matching to work properly, the values in the row must be sorted, left to right, in ascending order.

VLOOKUP works in the same way, except that it finds a value in the first column of the table and then moves over a specified number of columns. The arguments follow:

- **The value to find in the leftmost column of the table.**
- **The address of the table itself:** This is either a range or a named area.
- **The column offset from the leftmost column:** This is not a fixed column number but rather the number of columns relative from the leftmost column.
- **A true or false value:** If true (or omitted), VLOOKUP finds an approximate match. If false, an exact match is required. For an approximate match, the column must be sorted in ascending order.

Figure 14-11 shows an example of using VLOOKUP. The worksheet displays products and annual revenue data for the fictitious guitar shop. The range A6:D27 has been named `Sales`.



B4 =VLOOKUP("Wireless Gig Kit",OFFSET(Sales,0,1),3,FALSE)

	A	B	C	D	E
1					
2					
3					
4	**Total for the Wireless Gig Kit**	**$ 75,000**			
5					
6					
7	Vendor	Product	Category	Amount	
8	Great Guitars	All-in-one Effects Box	Accessory	$ 6,750	
9	Great Guitars	Guitarist's Super Road Kit	Accessory	$ 32,000	
10	Great Guitars	Flying X	Instrument	$ 87,000	
11	Great Guitars	Stratoblaster 9000	Instrument	$ 129,000	
12	Sound Accessories	18 Foot Cables	Accessory	$ 2,250	
13	Sound Accessories	1 Foot Cables	Accessory	$ 4,000	
14	Sound Accessories	Glow in the Dark Guitar Strap	Accessory	$ 4,350	
15	Sound Accessories	Rocker Pedal Deluxe	Accessory	$ 5,250	
16	Sound Accessories	10 Foot Cables	Accessory	$ 6,250	
17	Sound Accessories	Sustainer Sound Box	Accessory	$ 7,000	
18	Sound Accessories	Rocker Pedal	Accessory	$ 7,250	
19	Sound Accessories	6 Foot Cables	Accessory	$ 8,500	
20	Sound Accessories	Classic Amp Imitator	Accessory	$ 16,700	
21	Sound Accessories	Mini Road Recorder	Accessory	$ 40,000	
22	Sound Accessories	Wireless Gig Kit	Accessory	$ 75,000	
23	Traditional Instruments	Beginner Mandolin	Instrument	$ 1,450	
24	Traditional Instruments	Intermediate Guitar	Instrument	$ 3,500	
25	Traditional Instruments	Intermediate Mandolin	Instrument	$ 6,500	

FIGURE 14-11: Using VLOOKUP to locate data in a table.

The goal is to use VLOOKUP to extract the sales amount for the Wireless Gig Kit. However, the product names are in the second column of the Sales range, and VLOOKUP normally searches in the first column. You can use OFFSET to force VLOOKUP to search for Wireless Gig Kit in the second column of the range. This is the formula in cell B3:

```
=VLOOKUP("Wireless Gig Kit",OFFSET(Sales,0,1),3,
FALSE)
```

Note that the offset specified as the third argument to VLOOKUP is `3`. That's because the sales figures are in the third column relative to the Products column, where VLOOKUP is performing its search.

Here's how to use either HLOOKUP or VLOOKUP:

1. **Click a cell where you want the result to appear.**

2. **Type either** =HLOOKUP(**or** =VLOOKUP(**to start the function.**
3. **If using**
   - HLOOKUP: Enter the value that you want to find in the top row of the table.
   - VLOOKUP: Enter the value that you want to find in the first column of the table.
4. **Type a comma (,).**
5. **Enter the range address that defines the table of data, or enter its name, if it has been assigned one.**
6. **Type a comma (,).**
7. **If using**
   - HLOOKUP: Enter a number to indicate the row of the value to return.
   - VLOOKUP: Enter a number to indicate the column of the value to return.
8. REMEMBER The number you enter here is relative to the range or area defined in the second argument.
9. **(Optional) Type a comma (,) and then type** FALSE**.**
   This forces the function to find an exact match for the value entered in the first argument.
10. **Type a**) **and press Enter.**

TIP Excel also provides the LOOKUP function, which is specialized for returning values from single-column or single-row ranges. See Excel Help for more information on this function.

MATCH AND INDEX

The MATCH function returns the relative row number or column number of a value in a table. The key point here is that MATCH returns the relative location but does not return the value itself.

This function is useful when you need an item's position. You are not often interested in this information by itself but may use it in a more complex formula. I show you how shortly.

MATCH takes three arguments:

- **The value to search for:** This can be a number, text, or a logical value.
- **Where to look:** This is a range spanning a single row or column, or a named area that comprises a single row or column.
- **How the match is to be applied:** This argument is optional.

The third argument can be one of three values. They work as follows:

- `1` tells MATCH to find the largest value that is less than or equal to the lookup value. The array must be sorted in ascending order. This is the default value if the argument is omitted.
- `-1` tells MATCH to find the smallest value that is greater than or equal to the lookup value. The array must be sorted in descending order.
- `0` tells MATCH to find the first value that is an exact match. The array need not be sorted.

Figure 14-12 shows the products and revenue for the guitar shop. Note that the information has been sorted in ascending order according to the Amount column. The goal is to get a count of how many products have sales less than $10,000. MATCH makes this easy, as shown in Figure 14-12. This formula is in cell B4:

```
=MATCH(10000,OFFSET(Sales,0,3,ROWS(Sales),1))-1
```

B4 =MATCH(10000,OFFSET(Sales,0,3,ROWS(Sales),1))-1

	A	B	C	D	E
1					
2					
3					
4	Products with sales < 10,000	13			
5					
6	Vendor	Product	Category	Amount	
7	Traditional Instruments	Beginner Mandolin	Instrument	$ 1,450	
8	Sound Accessories	18 Foot Cables	Accessory	$ 2,250	
9	Traditional Instruments	Intermediate Guitar	Instrument	$ 3,500	
10	Sound Accessories	1 Foot Cables	Accessory	$ 4,000	
11	Sound Accessories	Glow in the Dark Guitar Strap	Accessory	$ 4,350	
12	Sound Accessories	Rocker Pedal Deluxe	Accessory	$ 5,250	
13	Sound Accessories	10 Foot Cables	Accessory	$ 6,250	
14	Traditional Instruments	Intermediate Mandolin	Instrument	$ 6,500	
15	Great Guitars	All-in-one Effects Box	Accessory	$ 6,750	
16	Sound Accessories	Sustainer Sound Box	Accessory	$ 7,000	
17	Sound Accessories	Rocker Pedal	Accessory	$ 7,250	
18	Traditional Instruments	Beginner Guitar	Instrument	$ 7,500	
19	Sound Accessories	6 Foot Cables	Accessory	$ 8,500	
20	Traditional Instruments	Intermediate Banjo	Instrument	$ 14,750	
21	Sound Accessories	Classic Amp Imitator	Accessory	$ 16,700	
22	Traditional Instruments	Beginner Banjo	Accessory	$ 27,000	
23	Great Guitars	Guitarist's Super Road Kit	Accessory	$ 32,000	
24	Sound Accessories	Mini Road Recorder	Accessory	$ 40,000	

FIGURE 14-12: Making a match.

Take this formula apart from the inside out. First, you know that MATCH needs a reference to the column where it is to search — in this case, the Amount column in the Sales range. Sounds like a job for OFFSET! Type the following:

```
OFFSET(Sales,0,3,ROWS(Sales),1)
```

This returns a range that has the following characteristics:

- Offset by no rows and three columns from the Sales range
- Has a height equal to the number of rows in Sales
- Has a width of one column

TIP Now that you have this range, you can tell MATCH to look for the largest value that is less than or equal to `10000`. Because the data is sorted, the relative position of this value in the range is one more than the number of products with sales less than $10,000. Why one more? The heading row at the top of the range is counted, too — so you subtract 1 to get the final answer.

Here's how to use the MATCH function:

1. **Click a cell where you want the result to appear.**
2. **Type** =MATCH(**to start the function.**
3. **Enter a value to match.**
   This can be a numeric, text, or logic value. You can enter a cell address provided that the referenced cell has a usable value.
4. **Type a comma (**,**).**
5. **Enter the range in which to look for a match.**
   This can be a range reference or a named area.
6. **(Optional) Enter a comma (**,**) and enter a** –1, 0**, or** 1 **to tell the function how to make a match.**
   The default is 1. A 0 forces an exact match.
7. **Type a**) **and press Enter.**

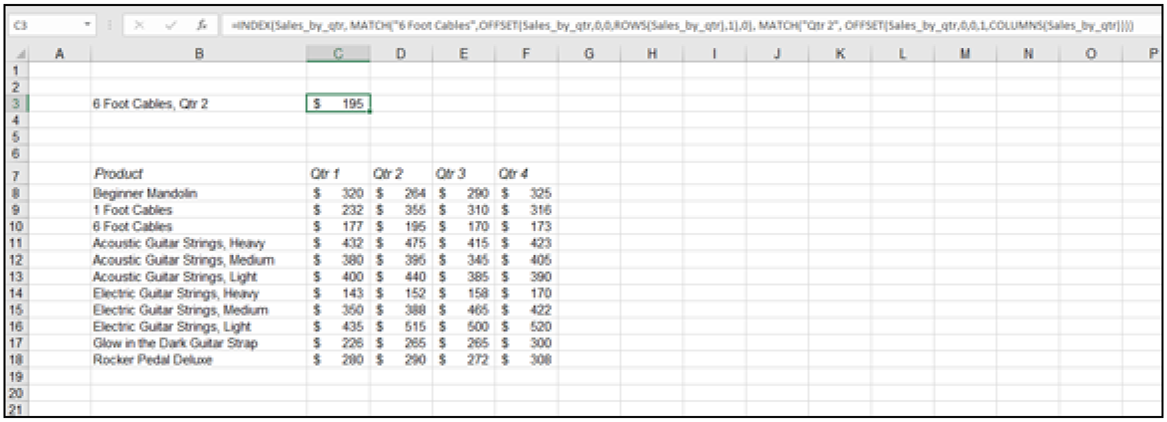
The information returned by MATCH can be helpful when you use it with the INDEX function. INDEX returns the value found at a specified row-and-column intersection within a table. You can use MATCH to find the row and find the column and then use INDEX to get the actual data.

INDEX takes three arguments:

- The table to look in as a range address or range name
- The row number relative to the table's first row
- The column number relative to the table's leftmost column

The return value is the value of the cell where the row and column intersect.

Figure 14-13 shows an example in which INDEX retrieves a value from a table that summarizes some guitar-shop sales by product and quarter. The table range in this example has been named Sales_by_qtr.



C3 =INDEX(Sales_by_qtr, MATCH("6 Foot Cables",OFFSET(Sales_by_qtr,0,0,ROWS(Sales_by_qtr),1),0), MATCH("Qtr 2", OFFSET(Sales_by_qtr,0,0,1,COLUMNS(Sales_by_qtr))))

6 Foot Cables, Qtr 2	$ 195			
Product	*Qtr 1*	*Qtr 2*	*Qtr 3*	*Qtr 4*
Beginner Mandolin	$ 320	$ 264	$ 290	$ 325
1 Foot Cables	$ 232	$ 355	$ 310	$ 316
6 Foot Cables	$ 177	$ 195	$ 170	$ 173
Acoustic Guitar Strings, Heavy	$ 432	$ 475	$ 415	$ 423
Acoustic Guitar Strings, Medium	$ 380	$ 395	$ 345	$ 405
Acoustic Guitar Strings, Light	$ 400	$ 440	$ 385	$ 390
Electric Guitar Strings, Heavy	$ 143	$ 152	$ 158	$ 170
Electric Guitar Strings, Medium	$ 350	$ 388	$ 465	$ 422
Electric Guitar Strings, Light	$ 435	$ 515	$ 500	$ 520
Glow in the Dark Guitar Strap	$ 226	$ 265	$ 265	$ 300
Rocker Pedal Deluxe	$ 280	$ 290	$ 272	$ 308

FIGURE 14-13: Using INDEX to extract data from a table.

The following formula, in cell C2, extracts the sales for 6 Foot Cables for Qtr 2:

```
=INDEX(Sales_by_qtr, MATCH("6 Foot Cables",
OFFSET(Sales_by_qtr,0,0,ROWS(Sales_by_qtr),1),0),
MATCH("Qtr 2", OFFSET(Sales_by_qtr,
0,0,1,COLUMNS(Sales_by_qtr))))
```

Wow, that's quite a cell-full of formula! But you already know everything you need to understand it. The first argument of INDEX is no mystery; it is simply the name assigned to the table. The second and third arguments, which tell INDEX what cell to look in, are complicated. Look at the first one, for the row argument:

```
MATCH("6 Foot Cables", OFFSET(Sales_by_qtr,
0,0,ROWS(Sales_by_qtr),1),0)
```

You want to look down the table’s first column, where the product names are listed, and find the row that contains `6 Foot Cables`. You also know that the MATCH function is just right for this job and that the function needs to know where to look. In other words, you must tell it the address of the table's first column. Here is where OFFSET comes into play:

```
OFFSET(Sales_by_qtr, 0, 0, ROWS(Sales_by_qtr), 1)
```

This call to OFFSET returns a range address that has the following characteristics:

- Is located with reference to the range Sales_by_qtr
- Is offset from Sales_by_qtr by zero rows and zero columns (in other words, starts at cell B7)
- Contains the same number of rows as Sales_by_qtr
- Contains one column

The result is that this call to OFFSET returns the range B7:B18. The MATCH function becomes this, in effect:

```
MATCH("6 Foot Cables", B7:B18, 0)
```

Because an exact match is requested, the data does not have to be sorted. MATCH finds the search text in the fourth row relative to the top of the table. This is the value that INDEX uses for its row argument. The column argument to INDEX is handled in the same way.

Here's how to use the INDEX function:

1. **Click a cell where you want the result to appear.**
2. **Type** =INDEX(**to start the function.**
3. **Enter a reference to the table.**
   You can drag the mouse over the range or enter its address. If the table has been named, you can enter the name.
4. **Type a comma (,).**
5. **Enter the row number relative to the table's first row.**
   This number can be the result of a calculation or the value returned from a function.
6. **Type a comma (,).**

7. **Enter the column number relative to the table’s leftmost column.** This number can be the result of a calculation or the value returned from a function.
8. **Type a) and press Enter.**

FORMULATEXT

FORMULATEXT displays the syntax of a formula. It's simple and yet serves a great feature. Think about it. You might have a workbook full of formulas; however, all you see is the result of the calculations. What if you need to see the formulas themselves? It's great to have the answer given by a calculation, but often, you need to know how the calculation works!

FORMULATEXT to the rescue! This function references a cell that has a formula and displays the formula without having it calculate the answer. Figure 14-14 shows how this works. Columns A and B contain numbers. Column C contains formulas that return calculated values using the numbers in columns A and B. Column D uses FORMULATEXT to display the formulas in column C.

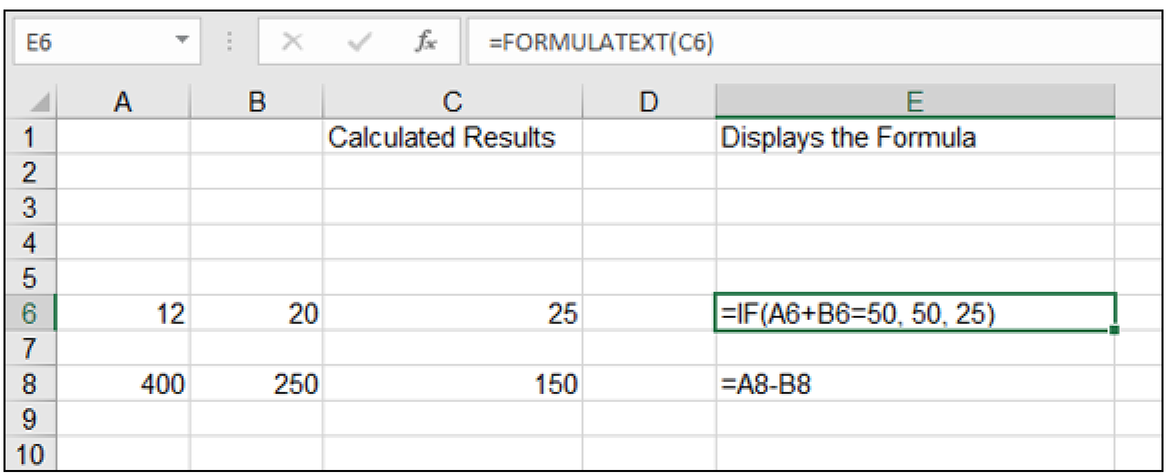


FIGURE 14-14: Using FORMULATEXT to see the syntax of formulas.

Here’s how to use the FORMULATEXT function:

1. **Click a cell where you want the result to appear.**
2. **Type** =FORMULATEXT(**to start the function.**

3. **Click a cell that has a formula.**
4. **Type a) and press Enter.**

TIP There is a setting in Excel's options to always display formulas as syntax instead of showing the calculated results. Look in the Display options for this worksheet on the Advanced tab in Excel Options to see where this option is set. This setting converts all formulas to text, and you can see them all. The caveat in using this approach is no calculations occur! If you need to see all the calculations' inner workings, this is a good option. However, if you want to still have the formulas calculate answers and want to see how they are written, use FORMULATEXT.

NUMBERVALUE

NUMBERVALUE is used to format numbers that appear as text back to appearing as actual numbers. For example, your worksheet might display a value such as "14.25%". Excel will treat this correctly as a number if used in a formula, function, or calculation, but the percent sign is not part of the number.

REMEMBER A percentage is usually a decimal-based number.

The 14.25 percent, when not formatted, is .1425.

Perhaps you need to display such a nice-looking number for the raw value it really is. This is where NUMBERVALUE comes to the rescue.

Figure 14-15 shows a worksheet with two numbers in column A that, in one way or another, appear to be a bit non-numeric. Besides 14.25 percent, there is "1 2 3 4" (spaces between each digit).

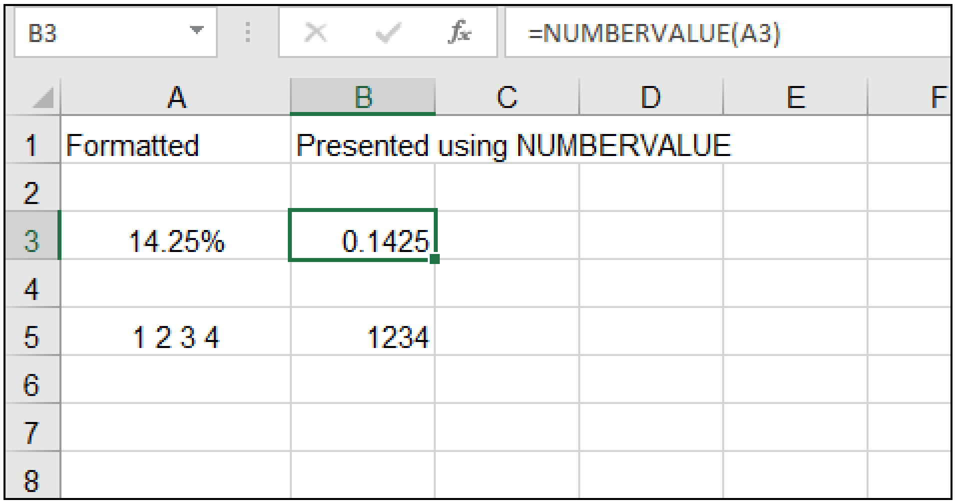


FIGURE 14-15: Using NUMBERVALUE to return the numeric presentation of a number.

Column B shows how the numbers look when you use NUMBERVALUE. For example, cell B3 contains this:

```
=NUMBERVALUE(A3)
```

Getting Informed with the CELL Function

The CELL function provides feedback about cells and ranges in a worksheet. You can find out what row and column a cell is in, what type of formatting it has, whether it's protected, and so on.

CELL takes two arguments:

- The first argument, which is enclosed in double quotes, tells the function what kind of information to return.
- The second argument tells the function which cell or range to evaluate. If you specify a range that contains more than one cell, the function returns information about the top-left cell in the range. The

second argument is optional; when it isn't provided, Excel reports back on the most recently changed cell.

Table 15-1 shows the list of possible entries for the first argument of the CELL function.

TABLE 15-1 Selecting the First Argument for the CELL Function

Argument	***Example***	***Comment***
address	=CELL("address")	Returns the address of the last changed cell.
col	=CELL("col",Sales)	Returns the column number of the first cell in the Sales range.

color	=CELL("color",B3)	Tells whether a particular cell (in this case, cell B3) is formatted in such a way that negative numbers are represented in color. The number, currency, and custom formats have selections for displaying negative numbers in red. If the cell is formatted for color-negative numbers, a 1 is returned; otherwise, a 0 is returned.

contents	=CELL("contents",B3)	Returns the contents of a particular cell (in this case, cell B3). If the cell contains a formula, returns the result of the formula and not the formula itself.
filename	=CELL("filename")	Returns the path, filename, and worksheet name of the workbook and worksheet that has the CELL function in it (for example, C:\Customers\[Acme Company]Sheet1). The function results in a blank answer in a new workbook that has not yet been saved.

format	=CELL("format",D12)	Returns a cell’s number format (in this case, cell D12). See Table 15-2 for a list of possible returned values.
parentheses	=CELL("parentheses",D12)	Returns 1 if a cell (in this case, D12) is formatted to have either positive values or all values displayed with parentheses. Otherwise, 0 is returned. A custom format is needed to make parentheses appear with positive values in the first place.

prefix	=CELL("prefix",R25)	Returns the type of text alignment in a cell (in this case, cell R25). There are a few possibilities: a single quotation mark (‘) if the cell is left-aligned; a double quotation mark (") if the cell is right-aligned; a caret (^) if the cell is set to centered; or a backslash (\) if the cell is fill-aligned. If the cell being evaluated is blank or has a number, the function returns nothing.

protect	=CELL("protect",D12)	Returns 1 if a cell’s protection (in this case, cell D12) is set to locked; otherwise, a 0 is returned. The returned value is not affected by whether the worksheet is currently protected.
row	=CELL("row",Sales)	Returns the row number of the first cell in the Sales range.

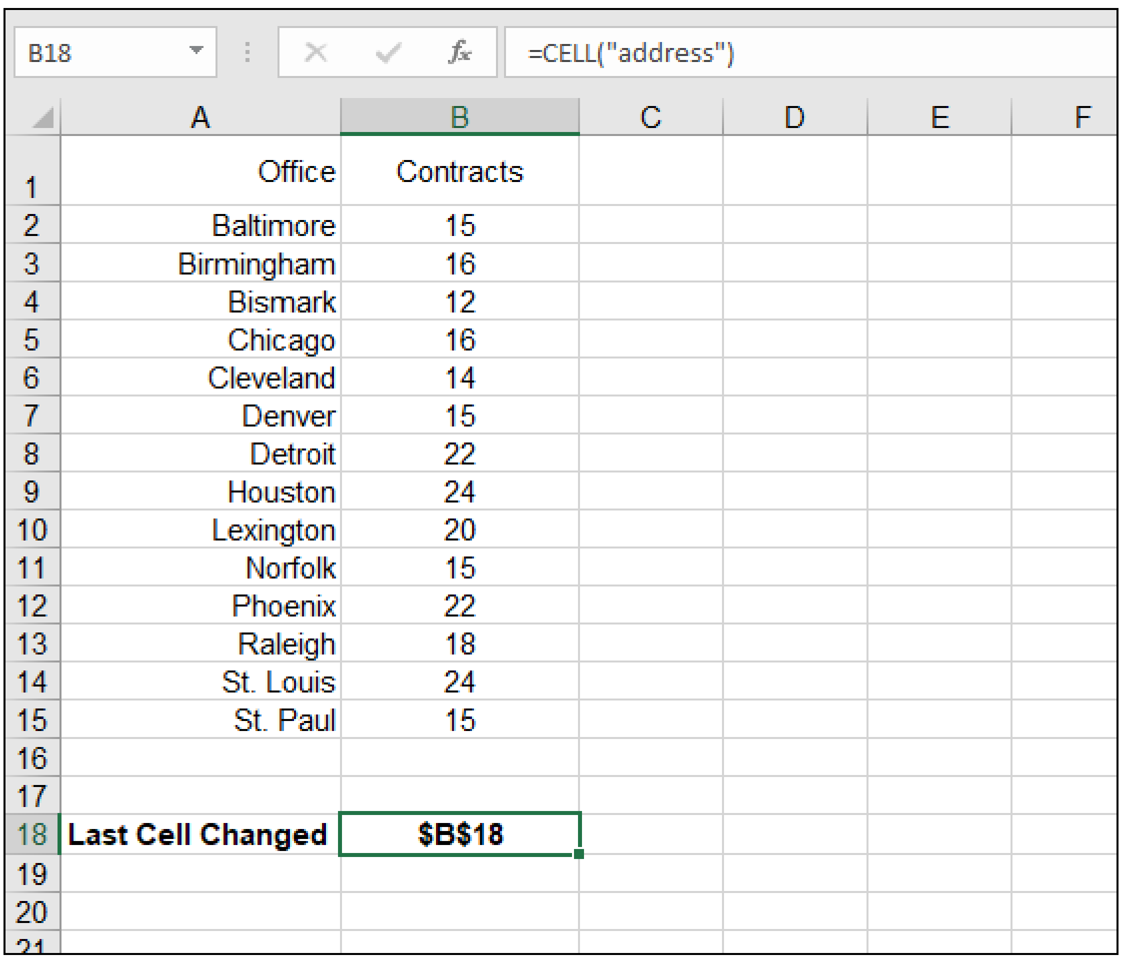
type	=CELL("type",D12)	Returns a value corresponding to the type of information in a cell (in this case, cell D12). There are three possible values: b if the cell is blank; l if the cell has alphanumeric data; and v for all other possible values, including numbers and errors.
width	=CELL("width")	Returns the width of the last changed cell, rounded to an integer. For example, a width of 18.3 is returned as 18.

The second argument, whether it's there or not, plays a key role in how the CELL function works. When it's included, the second argument is a cell address, such as B12, or a range name, such as Sales. Of course, you could have a range that is only one cell, but I won't confuse the issue!

WARNING If you enter a nonexistent range name for the second argument, Excel returns the #NAME? error. Excel can’t return information about something that doesn’t exist!

An interesting way to use CELL is to keep track of the last entry on a worksheet. Say you’re updating a list of values. The phone rings, and you’re tied up for a while on the call. When you get back to your list, you’ve forgotten where you left off. Yikes! What a time to think “If only I had used the CELL function!”

Figure 15-1 shows such a worksheet. Cell B18 displays the address of the last cell that was changed.


B18 =CELL("address")

	A	B	C	D	E	F
1	Office	Contracts				
2	Baltimore	15				
3	Birmingham	16				
4	Bismark	12				
5	Chicago	16				
6	Cleveland	14				
7	Denver	15				
8	Detroit	22				
9	Houston	24				
10	Lexington	20				
11	Norfolk	15				
12	Phoenix	22				
13	Raleigh	18				
14	St. Louis	24				
15	St. Paul	15				
16						
17						
18	**Last Cell Changed**	**$B$18**				
19						
20						

FIGURE 15-1: Keeping track of which cell had the latest entry.

Using CELL with the filename argument is great for displaying the workbook's path. This technique is common for printed worksheet reports. Being able to find the workbook file that a report was printed from 6 months ago is a real time-saver. Don't you just love it when the boss gives you an hour to create a report, doesn't look at it for 6 months, and *then* wants to make a change? Here's how you enter the CELL function to return the filename:

```
=CELL("filename")
```

You can format cells in many ways. When the first argument of CELL is `format`, a code is returned that corresponds to the formatting. The possible formats are those listed in the Format Cells dialog box. Table 15-2 shows the formats and the code that CELL returns.

TABLE 15-2 Returned Values for the `format` Argument

Format	*Returned Value from CELL Function*
General	G
0	F0
#,##0	,0
0.00	F2

#,##0.00	,2
$#,##0_);($#,##0)	C0
$#,##0_);[Red]($#,##0)	C0-
$#,##0.00_);($#,##0.00)	C2
$#,##0.00_);[Red]($#,##0.00)	C2-
0%	P0
0.00%	P2
0.00E+00	S2
# ?/? or ??/??	G

m/d/yy or m/d/yy h:mm or mm/dd/yy	D4
d-mmm-yy or dd-mmmm-yy	D1
d-mmm or dd-mmm	D2
mmm-yy	D3
mm/dd	D5
h:mm AM/PM	D7
h:mm:ss AM/PM	D6
h:mm	D9
h:mm:ss	D8

Using CELL with the `format` argument lets you add a bit of smarts to your worksheet. Figure 15-2 shows an example of CELL making sure information is correctly understood. The dates in column A are of the `d-mmm` format. The downside of this format is that the year is not known. So cell A1 has been given a formula that uses CELL to test the dates' format. If the `d-mmm` format is found in the first date (in cell A4), cell A1 displays a message that includes the year from cell A4. After all, cell A4 *has* a year; it's just formatted not to show it. This way, the year is always present — either in the dates themselves or at the top of the worksheet.

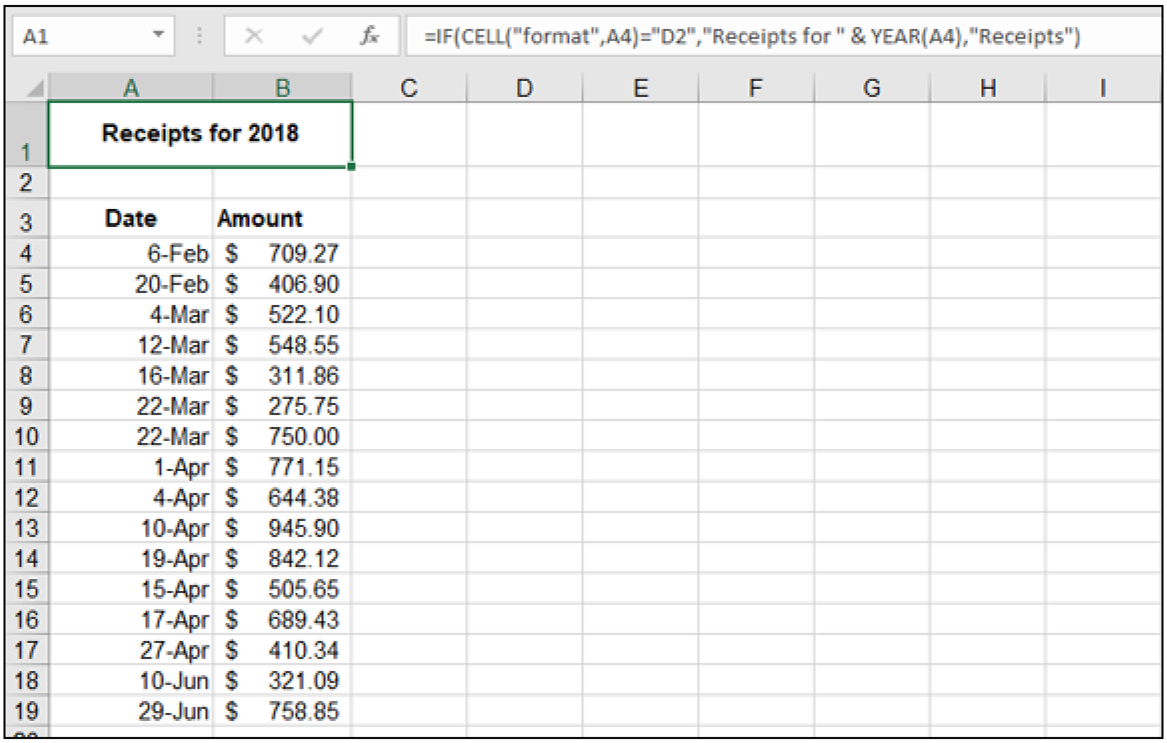


FIGURE 15-2: Using CELL and the `format` argument to display a useful message.

The formula in cell A1 — `=IF(CELL("format",A4)="D2","Receipts for "&YEAR(A4),"Receipts")` — says that if the formatting in A4 is `d-mmm` (according to the values in Table 15-2), display the message with the year; otherwise, just display `Receipts`.

Here's how to use the CELL function:

1. **Position the cursor in the cell where you want the results to appear.**
2. **Type** =CELL(**to begin the function entry.**
3. **Enter one of the first argument choices listed in Table 15-1.**
   Make sure to surround it with double quotes (" ").
4. **If you want to tell the function which cell or range to use, type a comma (,).**
5. **If you want, enter a cell address or the name of a range.**
6. **Type a) and press Enter.**

Getting Information About Excel and Your Computer System

Excel provides the INFO function to get information about your computer and about the program itself. INFO takes a single argument that tells the function what type of information to return. Table 15-3 shows how to use the INFO function.

TABLE 15-3 Using INFO to Find Out About Your Computer or Excel

Argument	*Example*	*Comment*

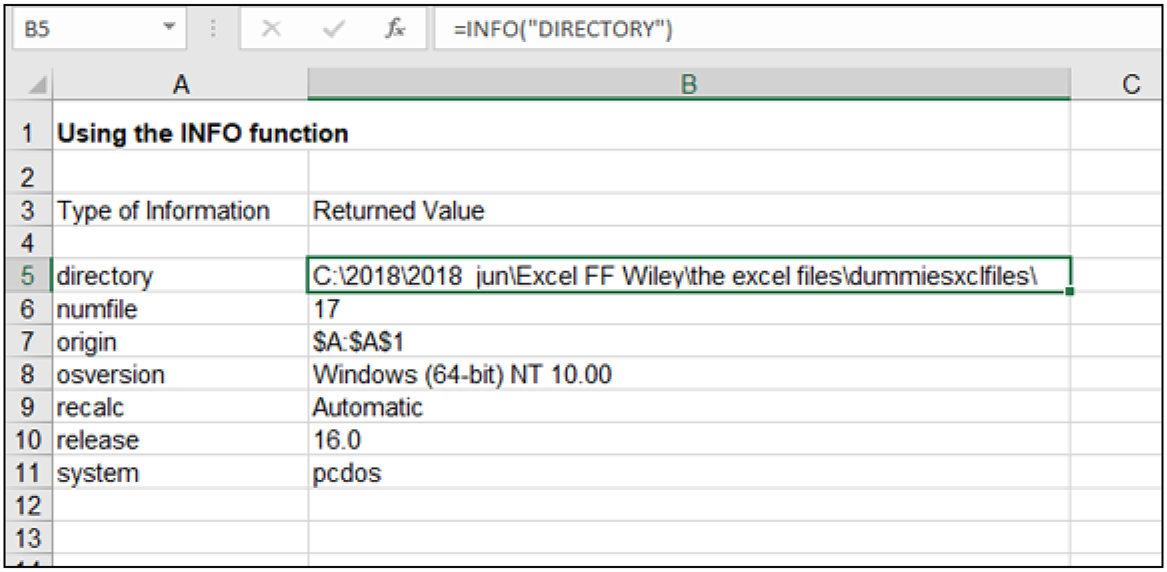
directory	=INFO("directory")	Returns the path of the current directory. Note that this is not necessarily the same path of the open workbook.
numfile	=INFO("numfile")	Returns the number of worksheets in all open workbooks. The function includes worksheets of add-ins, so the number could be misleading.
origin	=INFO("origin")	Returns the address of the cell at the top and to the left of the scrollable area. An A$ prefix in front of the cell address is for compatibility with Lotus 1-2-3.
osversion	=INFO("osversion")	Returns the name of the current operating system.

recalc	=INFO("recalc")	Returns the status of the recalculation mode: Automatic or Manual.
releas e	=INFO("release")	Returns the version number of Excel being run.
syste m	=INFO("system")	Returns the name of the operating environment: mac or pcdos.

One useful application of the INFO function is to use the returned Excel version number to determine whether the workbook can use a newer feature. For example, the ability to work with XML data has been available only in Excel 2002 and later. By testing the version number, you can be notified whether you can work with XML data. This formula uses the `release` choice as the argument:

```
=IF(INFO("release")>9,"This version can import
XML", "This version cannot import XML")
```

Figure 15-3 shows values returned with the INFO function.



B5 =INFO("DIRECTORY")

	A	B	C
1	**Using the INFO function**		
2			
3	Type of Information	Returned Value	
4			
5	directory	C:\2018\2018 jun\Excel FF Wiley\the excel files\dummiesxclfiles\	
6	numfile	17	
7	origin	$A:$A$1	
8	osversion	Windows (64-bit) NT 10.00	
9	recalc	Automatic	
10	release	16.0	
11	system	pcdos	
12			
13			

FIGURE 15-3: Getting facts about the computer with the INFO function.

Here's how to use the INFO function:

1. **Position the cursor in the cell where you want the results to appear.**
2. **Type** =INFO(**to begin the function entry.**
3. **Enter one of the argument choices listed in Table 15-3.**

   Make sure to surround it with double quotes (" ").
4. **Type a**) **and press Enter.**

Finding What IS and What IS Not

A handful of IS functions report back a true or false answer about certain cell characteristics. For example, is a cell blank, or does it contain text? These functions are often used in combination with other functions — typically, the IF function — to handle errors or other unexpected or undesirable results.

The errors Excel reports are not very friendly. What on earth does `#N/A` really tell you? The functions I describe in this section don't make the error any clearer, but they give you a way to instead display a friendly message like "Something is wrong, but I don't know what it is."

Table 15-4 shows the IS functions and how they're used. They all return either `True` or `False`, so the table just lists them.

TABLE 15-4 Using the IS Functions to See What Really Is

Function	*Comment*
=ISBLANK(value)	Tells whether a cell is blank.
=ISERR(value)	Tells whether a cell contains any error other than #N/A.
=ISERROR(value)	Tells whether a cell contains any error.
=ISEVEN(value)	Tells whether a number is even.
=ISFORMULA	Tells whether the cell contains a formula.
=ISLOGICAL(value)	Tells whether the value is logical.

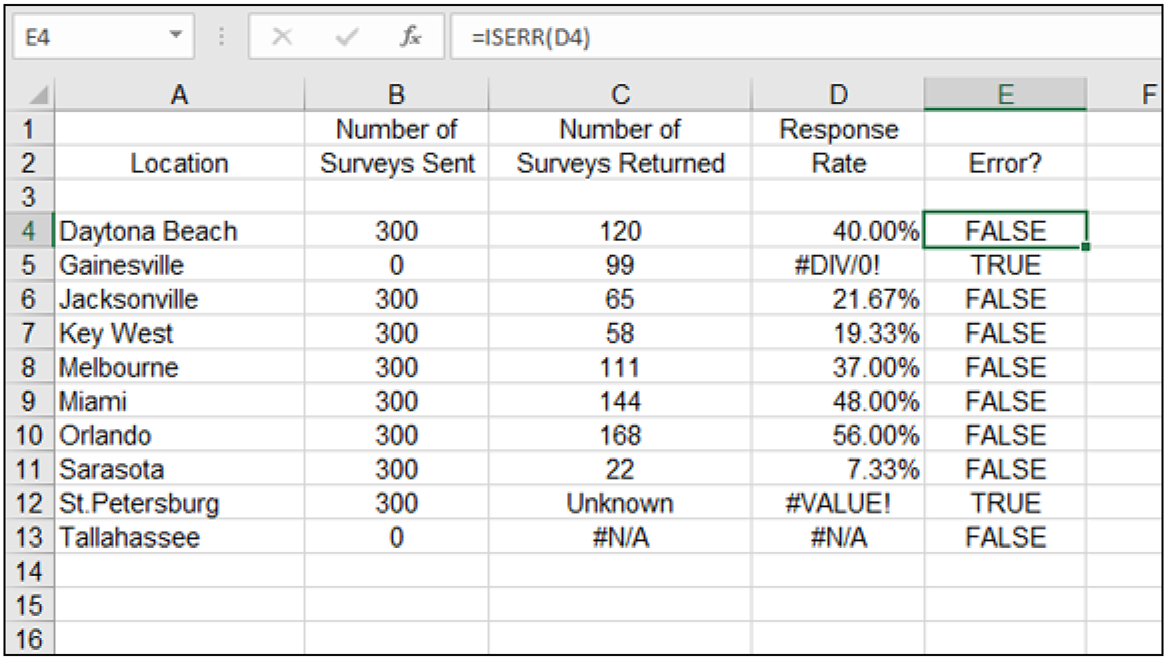
=ISNA(value)	Tells whether a cell contains the #N/A error.
=ISNONTEXT(value)	Tells whether a cell contains a number or error.
=ISNUMBER(value)	Tells whether a cell contains a number.
=ISODD(value)	Tells whether a number is odd.
=ISOWEEKNUM	Tells the ISO week number for the entered date. (ISO is the International Organization for Standardization, a standards-setting consortium.)
=ISREF(value)	Tells whether the value is a reference.
=ISTEXT(value)	Tells whether a cell contains text.

ISERR, ISNA, AND ISERROR

Three of the IS functions — ISERR, ISNA, and ISERROR — tell you about an error.

Error Function	*Comments*
ISERR	Returns true if the error is anything except the `#N/A` error. For example, the `#DIV/0!` error returns true.
ISNA	The opposite of ISERR. It returns true only if the error is `#N/A`.
ISERROR	Returns true for any type of error, including `#N/A`, `#VALUE!`, `#REF!`, `DIV/0!`, `#NUM!`, `#NAME?`, and `#NULL!`.

Why is `#N/A` treated separately? It is excluded from being handled with ISERR and has its own ISNA function. Actually, you can use `#N/A` to your advantage to avoid errors. How so? Figure 15-4 shows an example that calculates the percentage of surveys returned for some of Florida's larger cities. The calculation is simple: Just divide the returned number by the number sent.



E4 =ISERR(D4)

	A	B	C	D	E
1		Number of	Number of	Response	
2	Location	Surveys Sent	Surveys Returned	Rate	Error?
3					
4	Daytona Beach	300	120	40.00%	FALSE
5	Gainesville	0	99	#DIV/0!	TRUE
6	Jacksonville	300	65	21.67%	FALSE
7	Key West	300	58	19.33%	FALSE
8	Melbourne	300	111	37.00%	FALSE
9	Miami	300	144	48.00%	FALSE
10	Orlando	300	168	56.00%	FALSE
11	Sarasota	300	22	7.33%	FALSE
12	St.Petersburg	300	Unknown	#VALUE!	TRUE
13	Tallahassee	0	#N/A	#N/A	FALSE
14					
15					
16					

FIGURE 15-4: Using an error to your advantage.

However, errors do creep in. For example, no surveys were sent to Gainesville, yet 99 came back. Interesting! The calculation becomes a division by zero error, which makes sense. On the other hand, Tallahassee had no surveys sent, but here, the returned value is the `#N/A` error, purposely entered. Next, look at column E. In this column, `True` or `False` is returned to indicate whether the calculation, per city, should be considered an error: Gainesville true, Tallahassee false.

The result `true` or `false` appears in column E because all the cells in column E use the ISERR function. The formula in cell E13, which tests the calculation for Tallahassee, is `=ISERR(D13)`.

Simply put, D13 displays the `#N/A` error because its calculation (`=C13/B13`) uses a cell with an entered `#N/A`. The ISERR does not consider `#N/A` to be an error; therefore, E13 returns `False`. The upshot is that eyeballing column E makes it easy to distinguish entry and math errors from purposeful flagging of certain rows as having incomplete data.

ISBLANK, ISNONTEXT, ISTEXT, AND ISNUMBER

The ISBLANK, ISNONTEXT, ISTEXT, and ISNUMBER functions tell you what type of data is in a cell.

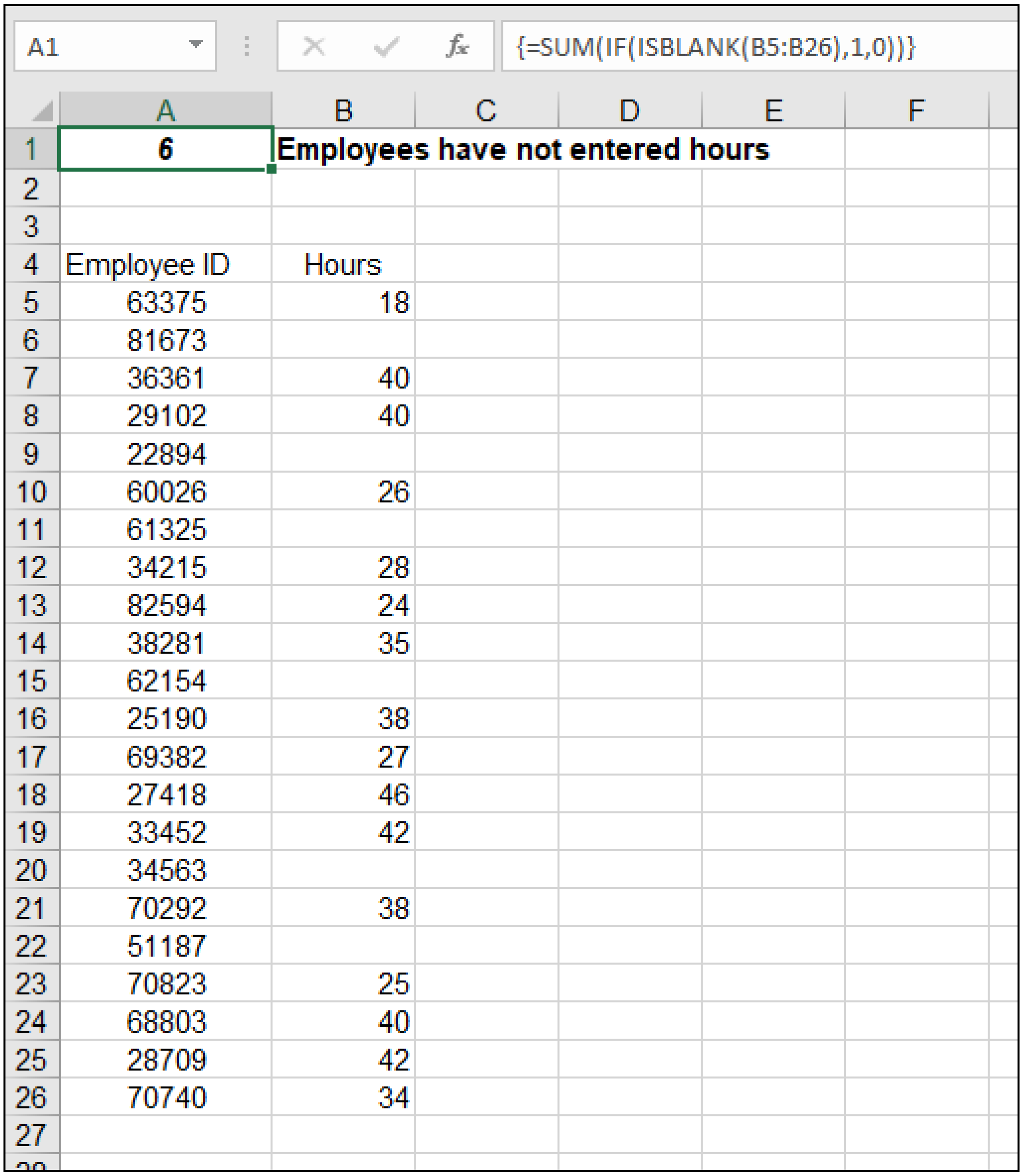
Error Function	***Comments***
ISBLANK	Returns true if the cell is empty; otherwise, returns false.
ISNONTEXT	Returns true if the cell contains anything that is not text: a number, a date/time, or an error. The function returns true if the cell is blank or false if the cell contains text or a formula whose result is text.
ISTEXT	The opposite of ISNONTEXT: Returns true if the cell contains text or a formula whose result is text; otherwise, returns false.
ISNUMBER	Returns true if the cell contains a number or a formula whose result is a number; otherwise, returns false.

ISBLANK returns true when nothing is in a cell. Using ISBLANK is useful for counting how many cells in a range are blank. Perhaps you're responsible for making sure that 200 employees get their time sheets in every week. You can use a formula that lets you know how many employees have not yet handed in their hours.

Such a formula uses ISBLANK along with the IF and SUM functions, like this:

```
{=SUM(IF(ISBLANK(B5:B26),1,0))}
```

This formula makes use of an array. See Chapter 3 for more information on using array formulas. Figure 15-5 shows how this formula works. In columns A and B are lists of employees and their hours. The formula in cell A1 reports how many employees are missing their hours.



A1 {=SUM(IF(ISBLANK(B5:B26),1,0))}

	A	B	C	D	E	F
1	**6**	**Employees have not entered hours**				
2						
3						
4	Employee ID	Hours				
5	63375	18				
6	81673					
7	36361	40				
8	29102	40				
9	22894					
10	60026	26				
11	61325					
12	34215	28				
13	82594	24				
14	38281	35				
15	62154					
16	25190	38				
17	69382	27				
18	27418	46				
19	33452	42				
20	34563					
21	70292	38				
22	51187					
23	70823	25				
24	68803	40				
25	28709	42				
26	70740	34				
27						

FIGURE 15-5: Calculating how many employees are missing an entry.

ISTEXT returns `True` when a cell contains any type of text. ISNONTEXT returns `True` when a cell contains anything that is not text, including numbers, dates, and times. The ISNONTEXT function also returns `True` if the cell contains an error.

The ISNUMBER function returns `True` when a cell contains a number, which can be an actual number or a number resulting from evaluation of a formula in the cell. You can use ISNUMBER as an aid to help data entry. Say you designed a worksheet that people fill out. One of the questions is age. Most people would enter a numeric value such as 18, 25, 70, and so on. But someone could type the age as text, such as eighteen, thirty-two, or “none of your business.” An adjacent cell could use ISNUMBER to return a message about entering the numeric age. The formula would look something like this:

```
=IF(ISNUMBER(B3),"","Please enter your age as a
number")
```

Here's how to use any of the IS functions:

1. **Position the cursor in the cell where you want the results to appear.**
2. **Enter one of the IS functions.**
   For example, type **=ISTEXT**(to begin the function entry.
3. **Enter a cell address.**
4. **Type a**) **and press Enter.**
   The result is always `True` or `False`.

Getting to Know Your Type

The TYPE function tells you what the type of the information is; for example:

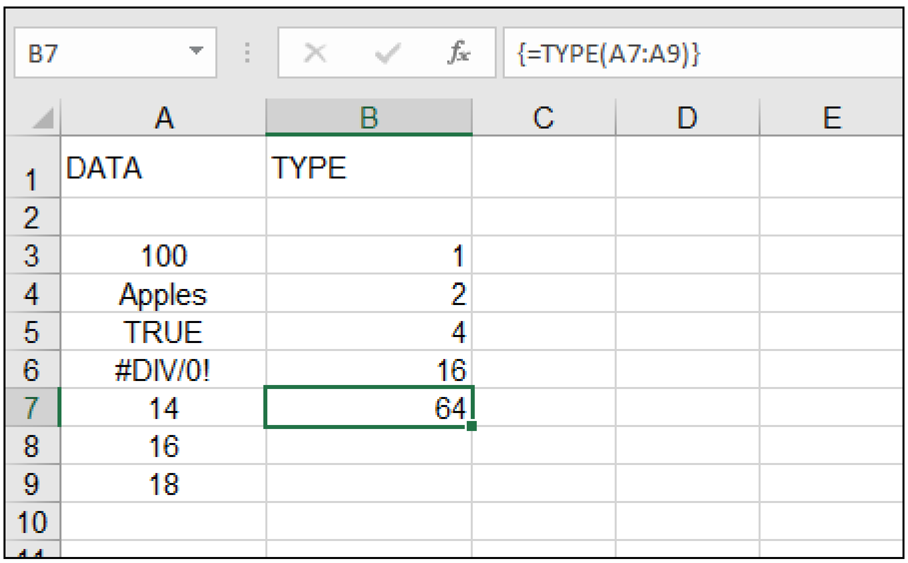
- Number
- Text
- A logical value
- An error
- An array

In all cases TYPE returns a number:

- `1` is returned for numbers.

- `2` is returned for text.
- `4` is returned for logical values.
- `16` is returned for errors.
- `64` is returned for arrays.

Figure 15-6 shows each of these values returned by the TYPE function. Cells B3:B7 contain the TYPE function, with each row looking at the adjacent cell in column A. The returned value of 64 in cell B7 is a little different. This indicates an array as the type. The formula in cell B7 is `=TYPE(A7:A9)`. This is an array of values from cells A7:A9.



	A	B	C	D	E
1	DATA	TYPE			
2					
3	100	1			
4	Apples	2			
5	TRUE	4			
6	#DIV/0!	16			
7	14	64			
8	16				
9	18				
10					

FIGURE 15-6: Getting the type of the data.

Here's how to use the TYPE function:

1. **Position the cursor in the cell where you want the results to appear.**
2. **Enter** =TYPE(**to begin the function entry.**
3. **Enter a cell address or click a cell.**
4. **Type a) and press Enter.**

The ERROR.TYPE function returns a number that corresponds to the particular error in a cell. Table 15-5 shows the error types and the returned numbers.

TABLE 15-5 Getting a Number of an Error

Error Type	***Returned Number***
`#NULL!`	1
`#DIV/0!`	2
`#VALUE!`	3
`#REF!`	4
`#NAME?`	5
`#NUM!`	6
`#N/A`	7

The best thing about the ERROR.TYPE function is that you can use it to change those pesky errors to something readable! To do this, use the CHOOSE function along with ERROR.TYPE, like this:

```
=CHOOSE(ERROR.TYPE(H14),"Nothing here!","You can't
divide by 0","A bad number has been entered", "The
formula is referencing a bad cell or range","There
is a problem with the entry","There is a problem
with the entered value","Something is seriously
wrong!")
```

See Chapter 14 for assistance on using the CHOOSE function. This is how you use the ERROR.TYPE function:

1. **Position the cursor in the cell where you want the results to appear.**
2. **Enter** =ERROR.TYPE(**to begin the function entry.**
3. **Enter a cell address or click a cell.**
4. **Type a**) **and press Enter.**

Putting Your Data into a Database Structure

REMEMBER To use the database functions, you need to put your data into a structured format. Excel is very flexible. Usually, you put data wherever you want. But to make the best of the database functions, you need to get your data into a contiguous area of rows and columns. Each row is a record, and each column is a field. The top row contains labels that identify the fields.

Figure 17-1 shows a database in a worksheet. This example is a list of students (by ID number) and their classes, teachers, and grades. Each student occupies a row — in other words, a record — in the database. Each of the four fields — Student ID, Class, Teacher, and Final Grade — is in one column and is identified by a label in the top row.

B2 | Calculus 101

	A	B	C	D
1	**Student ID**	**Class**	**Teacher**	**Final Grade**
2	VM4128	Calculus 101	Mr. Crasdale	77
3	CH8965	Ancient Greece	Mr. Young	81
4	SG9555	Accounting 101	Ms. Morley	76
5	AB5235	Calculus 101	Mr. Crasdale	86
6	KD0656	Accounting 101	Mr. Harris	98
7	JG1183	Accounting 101	Mr. Harris	98
8	HE7976	Masters of Philosophy	Mr. Crasdale	78
9	NR5090	Calculus 101	Mr. Porter	79
10	AJ3549	Accounting 101	Mr. Harris	85
11	CW8495	English Literature	Mr. Johnson	66
12	NR5090	Calculus 101	Mr. Porter	88
13	QT6233	Accounting 101	Ms. Morley	73
14	MP9632	English Literature	Ms. Rendson	84
15	DK7492	Accounting 101	Mr. Richards	71
16	RE3968	Ancient Greece	Mr. Young	75
17	AP8356	Accounting 101	Mr. Richards	88
18	FL1423	Masters of Philosophy	Ms. Untermeyer	93
19	BW2559	Calculus 101	Mr. Porter	86
20	QL9026	Calculus 101	Mr. Crasdale	77
21	FE4903	English Literature	Ms. Appleson	74
22	TY3987	Calculus 101	Mr. Porter	74
23	EE4688	English Literature	Mr. Johnson	73
24	WD0448	Ancient Greece	Mr. Young	76
25	GY7754	English Literature	Mr. Johnson	66

FIGURE 17-1: Using a database to store student information.

The data in the worksheet in Figure 17-1 is really just normal data. There is nothing special about it. However, the data sits in organized rows and columns, making it ready for working with Excel's database functions:

- Each column is a field that holds one particular item of data, such as Student ID or Class. It must contain no other data.
- Each row contains one record. In this example, a record is the data for one student.
- The top row of the database contains labels that identify the fields.

This sample data is used in this chapter to demonstrate the database functions. Of course, you can have a database in Excel and never use the database functions, but you have a lot more power at your fingertips if you do use them.

Working with Database Functions

The database functions all work in basically the same way. They perform some calculation on a specified field for those records that meet specified criteria. For example, you can use a database function to calculate the average final grade for all students in Accounting 101.

REMEMBER All database functions use the following three arguments:

- **The database range:** This argument tells the function where the database is. You enter it by using cell addresses (for example, A1:D200) or a named range (for example, Students). The range must include all records, including the top row of field names.
- **The field:** You must tell a database function which field to operate on. You can't expect it to figure this out by itself! You can enter either the column number or the field name. A column number, if used, is the number of the column offset from the first column of the database area. In other words, if a database starts in column C, and the field is

in column E, the column number is 3, not 5. If a heading is used, put it inside a set of double quotation marks. Database functions calculate a result based on the values in this field. Just how many values are used depends on the third argument: the criteria.

- **The criteria:** This tells the function where the criteria are located; it is not the criteria per se. The criteria tell the function which records to use in its calculation. You set up the criteria in a separate part of the worksheet, apart from the database area. This area's address is passed to the database function. Criteria are explained in detail throughout the chapter.

ESTABLISHING YOUR DATABASE

All database functions take a database reference as the first argument. The database area must include *headers* (field names) in the first row. In Figure 17-1 earlier in this chapter, the first row uses Student ID, Class, Teacher, and Final Grade as headers to the information in each respective column.

TIP A great way to work with the database functions is to name the database area and then enter the name, instead of the range address, in the function.

To set up a name, follow these steps:

1. **Select the entire database area.**
   Make sure the top row has headers and is included in the selection.
2. **Click the Formulas tab (at the top of the Excel window).**
3. **Click Define Name in the Defined Names area.**
   The New Name dialog box appears, with the range address set in the Refers To box.
4. **Type a name in the Name Box (or use the suggested name).**
5. **Click OK to close the dialog box.**

Later, if records are added to the bottom of the database, you have to redefine the named area's range to include the new rows. You can do this as follows:

1. **Click the Name Manager button on the Excel Formulas tab.**
   The Name Manager dialog box appears.
2. **Click the name in the list you want to redefine.**
3. **Click the Edit button in the dialog box.**
   Excel opens the Edit Name dialog box, shown in Figure 17-2, with information about the selected range.
4. **Change the reference in the Refers To box.**
   You can use the small square button to the right of the Refers To box to define the new reference by dragging the mouse pointer over it. Clicking the small square button reduces the size of the Edit Name dialog box and allows you access to the worksheet. When you are done dragging the mouse over the new worksheet area, press Enter to get back to the Edit Name dialog box.
5. **Click OK to save the reference change and close the dialog box.**
6. **Click Close.**

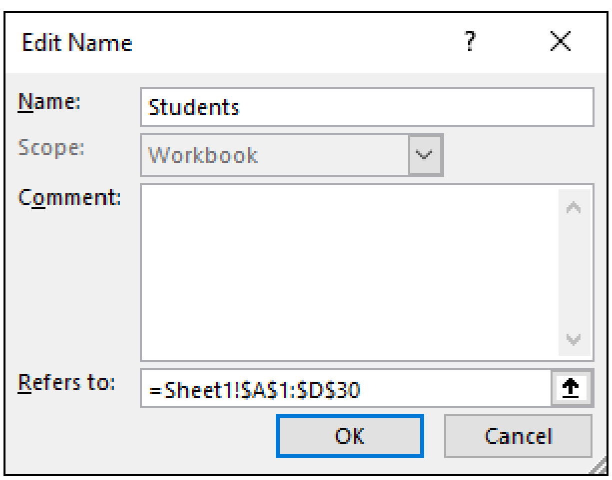


FIGURE 17-2: Updating the reference to a named area.

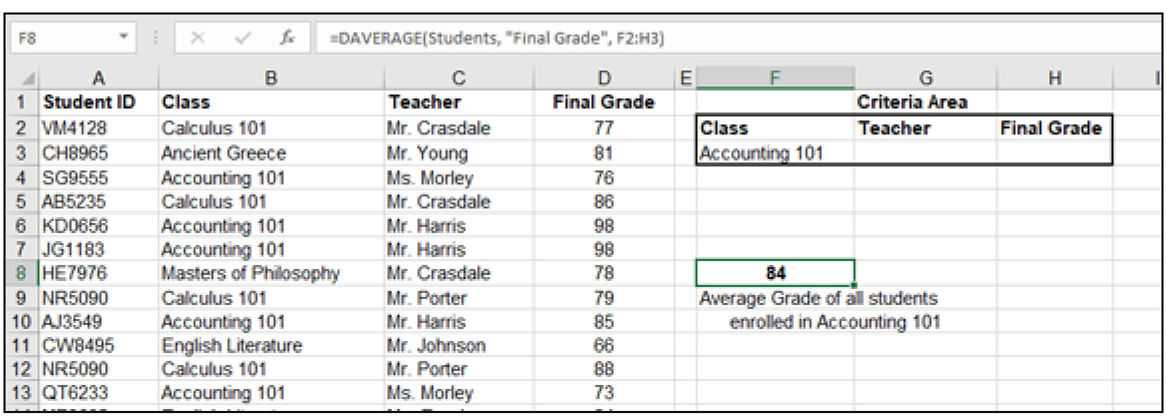
TIP If you add records to your database range by inserting new rows somewhere in the middle, rather than adding them on at the end, Excel automatically adjusts the reference to the named range.

ESTABLISHING THE CRITERIA AREA

As I mention earlier, the criteria are not part of the database function arguments but are somewhere in the worksheet and then referenced by the function. The criteria area can contain a single criterion, or it can contain two or more criteria. Each individual criterion is structured as follows:

- In one cell, enter the field name (header) of the database column that the criterion will apply to.
- In the cell below, enter the value that the field data must meet.

Figure 17-3 shows the student database with a criteria area to the right of the database. There are places to put criteria for the Class, Teacher, and Final Grade. In the example, a criterion has been set for the Class field. This criterion forces the database function to process only records (rows) where the Class is Accounting 101. Note, though, that a criterion can be set for more than one field. In this example, the Teacher and Final Grade criteria have been left blank so they don't affect the results.


F8 =DAVERAGE(Students, "Final Grade", F2:H3)

	A	B	C	D	E	F	G	H
1	Student ID	Class	Teacher	Final Grade			Criteria Area	
2	VM4128	Calculus 101	Mr. Crasdale	77		Class	Teacher	Final Grade
3	CH8965	Ancient Greece	Mr. Young	81		Accounting 101		
4	SG9555	Accounting 101	Ms. Morley	76				
5	AB5235	Calculus 101	Mr. Crasdale	86				
6	KD0656	Accounting 101	Mr. Harris	98				
7	JG1183	Accounting 101	Mr. Harris	98				
8	HE7976	Masters of Philosophy	Mr. Crasdale	78		84		
9	NR5090	Calculus 101	Mr. Porter	79		Average Grade of all students		
10	AJ3549	Accounting 101	Mr. Harris	85		enrolled in Accounting 101		
11	CW8495	English Literature	Mr. Johnson	66				
12	NR5090	Calculus 101	Mr. Porter	88				
13	QT6233	Accounting 101	Ms. Morley	73				

FIGURE 17-3: Selecting criteria to use with a database function.

The DAVERAGE function has been entered into cell F8 and uses this criteria range. The three arguments are in place. The name Students tells the function where the database is, the Final Grade field (column) is where the function finds values to calculate the average, and the criteria are set to the worksheet range that has criteria that tell the function to use only records where the Class is Accounting 101 — in other words, F2:H3. The entry in cell F8 looks like this:

```
=DAVERAGE(Students,"Final Grade",F2:H3)
```

Why does this function refer to F2:H3 as the criteria range when the only defined criterion is located in the range F2:F3? It's a matter of convenience. Because cells G3 and H3 in the criteria range are blank, the Teacher and Final Grade fields are ignored by a database function that uses this criteria range. However, if you want to enter a criterion for one of those fields, just enter it in the appropriate cell; there is no need to edit the database function arguments. What about assigning a name to the criteria area and then using the name as the third argument to the database function? That works perfectly well, too.

WARNING Whether you use a named area for your criteria or simply type the range address, you must be careful to specify an area that includes all the criteria but does *not* include any blank rows or columns. If you do, the database function's results will be incorrect.

Here's how you enter any of the database functions. This example uses the DSUM function, but the instructions are the same for all the database functions; just use the one that performs the desired calculation. Follow these steps:

1. **Import or create a database of information in a worksheet.**
   The information should be in contiguous rows and columns. Be sure to use field headers.

2. **Optionally, use the New Name dialog box to give the database a name.** To name your database, see the section "Establishing your database" earlier in this chapter.
3. **Select a portion of the worksheet to be the criteria area and then add headers to this area that match the database headers.** You have to provide criteria headers only for database fields that criteria are applied to. For example, your database area may have ten fields, but you need to define criteria to three fields. Therefore, the criteria area can be three columns wide.
4. **Position the cursor in the cell where you want the results to appear.** This cell must not be in the database area or the criteria area.
5. **Type** =DSUM(**to begin the function entry.**
6. **Enter the database range or a name, if one is set.**
7. **Type a comma (,).**
8. **Enter either of the following:**
   - The header name, in quotation marks, of the database field that the function should process
   - The column number
9. **Type a comma (,).**
10. **Enter the range of the criteria area.**
11. **Type a**) **and press Enter.**

Fine-Tuning Criteria with AND and OR

Excel's database functions would not be of much use if you could not create fairly sophisticated queries. A few common types of queries follow:

- Records that match two or more individual criteria
- Records that match any one of several criteria
- Values that fall within a specified range of values

To find records that match two or more criteria, place the criteria in adjacent columns in the criteria area. Continuing with the student-grade database, the criteria area shown in Figure 17-4 matches records where the Class field contains `Accounting 101` and the Teacher field contains `Mr. Harris`. This is called an AND criterion.

E	F	G	H
		Criteria Area	
	Class	**Teacher**	**Final Grade**
	Accounting 101	Mr. Harris	

FIGURE 17-4: Finding records that match two criteria.

To match records that meet any one of several criteria, place the individual criteria in two or more rows below the field name. Figure 17-5 shows a criteria range that matches all records where the Class field contains either `Accounting 101` or `English Literature`. This is called an OR criterion.

E	F	G	H
		Criteria Area	
	Class	**Teacher**	**Final Grade**
	Accounting 101		
	English Literature		

FIGURE 17-5: Finding records that match any one of two or more criteria.

To combine AND with OR in a criteria range, use two or more columns and two or more rows. Figure 17-6 shows a criteria range that finds all records where Class is `Accounting 101` and Teacher is either `Mr. Harris` or `Mr. Richards`.

E	F	G	H
		Criteria Area	
	Class	**Teacher**	**Final Grade**
	Accounting 101	Mr. Harris	
	English Literature	Mr. Richards	

FIGURE 17-6: Combining AND and OR criteria.

To define a criterion that uses ranges of values, use these numerical comparison operators:

- < for less than
- > for greater than
- <= for less than or equal to
- >= for greater than or equal to

Of course, you can apply these to fields with numerical values. Figure 17-7 shows two criteria areas. The upper one matches all records in which Final Grade is `90` or higher. The lower one matches all records in which Final Grade is equal to or greater than `80` and less than `90`.

Final Grade
>=90

Final Grade
<90
>=80

FIGURE 17-7: Defining numerical range criteria.

Adding Only What Matters with DSUM

The DSUM function lets you sum numbers in a database column for just those rows that match the criteria you specify. For example, take a database that contains data on individual sale amounts for sales people. The database range is named Sales. You want to calculate total sales for each of the three sales representatives. Figure 17-8 shows how this is done. Three criteria areas are defined in D2:D3, E2:E3, and F2:F3. The DSUM function is entered in cells E8:E10. The formula in cell E8 is

```
=DSUM(SALES, "Sale Amount", D2:D3)
```

E8 =DSUM(Sales,"Sale Amount",D2:D3)

	A	B	C	D	E	F
1	**Sales Person**	**Sale Amount**				
2	Jack Bennet	$ 43,234		**Sales Person**	**Sales Person**	**Sales Person**
3	Amy Wilson	$ 22,314		Jack Bennet	Amy Wilson	Wendell Jones
4	Wendell Jones	$ 22,355				
5	Amy Wilson	$ 12,500				
6	Wendell Jones	$ 8,900				
7	Jack Bennet	$ 12,450			**Total Sales**	
8	Amy Wilson	$ 22,334		Jack Bennet	**$ 79,134**	
9	Amy Wilson	$ 34,223		Amy Wilson	**$ 123,741**	
10	Wendell Jones	$ 12,332		Wendell Jones	**$ 70,387**	
11	Amy Wilson	$ 6,700				
12	Wendell Jones	$ 12,400				
13	Jack Bennet	$ 23,450				
14	Wendell Jones	$ 10,900				
15	Amy Wilson	$ 25,670				
16	Wendell Jones	$ 3,500				
17						

FIGURE 17-8: Calculating the sum of sales with the DSUM function.

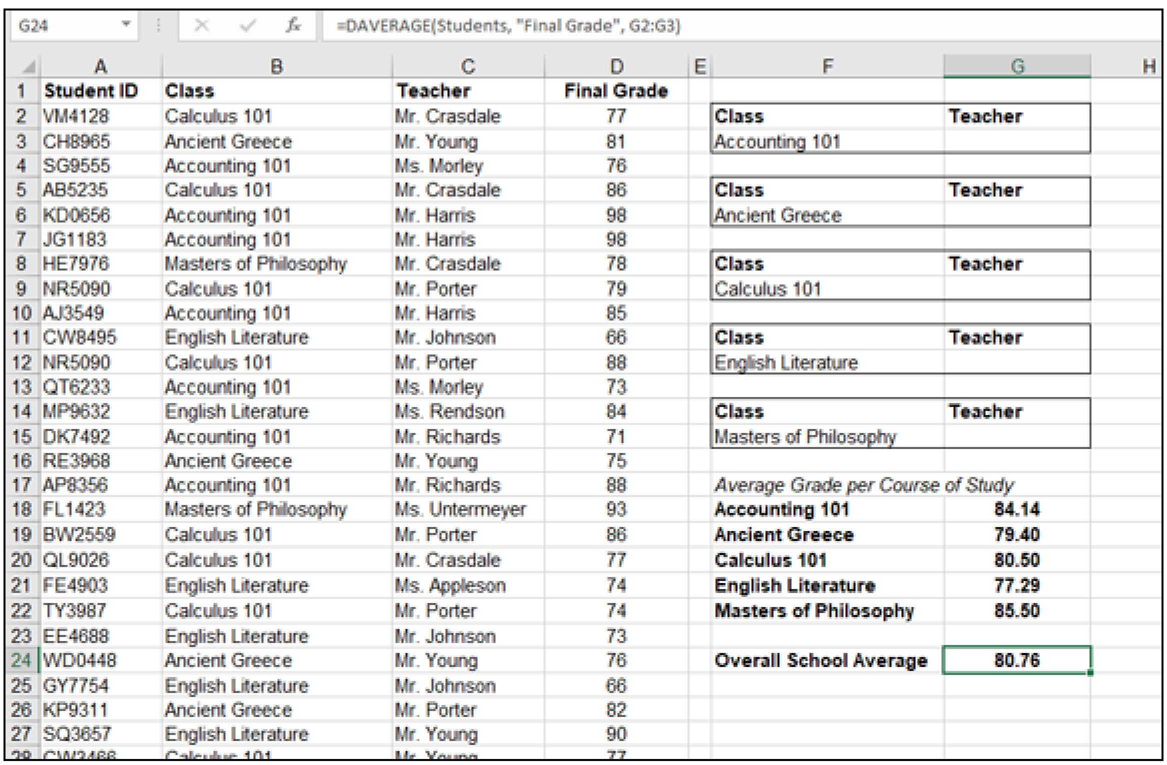
The functions entered in E9 and E10 are identical except for referencing a different criteria range. The results show clearly that Amy is the sales leader.

Going for the Middle with DAVERAGE

The DAVERAGE function lets you find the average, or *mean*, of a field for just the rows that match the criteria. For this example, you return to the student database.

Figure 17-9 shows a worksheet in which the average grade for each course has been calculated by DAVERAGE. For example, cell G22 shows the average grade for Masters of Philosophy. Here is the formula:

```
=DAVERAGE(Students,"Final Grade",F14:G15)
```



G24 =DAVERAGE(Students, "Final Grade", G2:G3)

	A	B	C	D	E	F	G	H
1	**Student ID**	**Class**	**Teacher**	**Final Grade**				
2	VM4128	Calculus 101	Mr. Crasdale	77		**Class**	**Teacher**	
3	CH8965	Ancient Greece	Mr. Young	81		Accounting 101		
4	SG9555	Accounting 101	Ms. Morley	76				
5	AB5235	Calculus 101	Mr. Crasdale	86		**Class**	**Teacher**	
6	KD0656	Accounting 101	Mr. Harris	98		Ancient Greece		
7	JG1183	Accounting 101	Mr. Harris	98				
8	HE7976	Masters of Philosophy	Mr. Crasdale	78		**Class**	**Teacher**	
9	NR5090	Calculus 101	Mr. Porter	79		Calculus 101		
10	AJ3549	Accounting 101	Mr. Harris	85				
11	CW8495	English Literature	Mr. Johnson	66		**Class**	**Teacher**	
12	NR5090	Calculus 101	Mr. Porter	88		English Literature		
13	QT6233	Accounting 101	Ms. Morley	73				
14	MP9632	English Literature	Ms. Rendson	84		**Class**	**Teacher**	
15	DK7492	Accounting 101	Mr. Richards	71		Masters of Philosophy		
16	RE3968	Ancient Greece	Mr. Young	75				
17	AP8356	Accounting 101	Mr. Richards	88		*Average Grade per Course of Study*		
18	FL1423	Masters of Philosophy	Ms. Untermeyer	93		**Accounting 101**	**84.14**	
19	BW2559	Calculus 101	Mr. Porter	86		**Ancient Greece**	**79.40**	
20	QL9026	Calculus 101	Mr. Crasdale	77		**Calculus 101**	**80.50**	
21	FE4903	English Literature	Ms. Appleson	74		**English Literature**	**77.29**	
22	TY3987	Calculus 101	Mr. Porter	74		**Masters of Philosophy**	**85.50**	
23	EE4688	English Literature	Mr. Johnson	73				
24	WD0448	Ancient Greece	Mr. Young	76		**Overall School Average**	**80.76**	
25	GY7754	English Literature	Mr. Johnson	66				
26	KP9311	Ancient Greece	Mr. Porter	82				
27	SQ3657	English Literature	Mr. Young	90				

FIGURE 17-9: Calculating the average grade for each course.

Each calculated average uses a different criteria area. Each area filters the result by a particular course. In all cases, the criteria area for the Teacher is left blank and, therefore, has no effect on the results.

For the sake of comparison, DAVERAGE is also used in cell G24 to show the overall average for all courses. Because a criterion is a required function argument, the calculation in cell G24 is set to look at an empty cell. None of the Class criteria cells is free, so the function looks to the Teacher criterion in cell G3. Because this cell has no particular teacher entered as a criterion, all of the records in the database are used to create this average — just what you want. Here is the formula in cell G24:

```
=DAVERAGE(Students,"Final Grade",G2:G3)
```

It doesn't matter which field header you use in the criterion when you're getting a result based on all records in a database. What *does* matter is that there is no actual criterion below the header.

Counting Only What Matters with DCOUNT

The DCOUNT function lets you determine how many records in the database match the criteria.

Figure 17-10 shows how DCOUNT can determine how many students took each course. Cells G18:G22 contain formulas that count records based on the criterion (the Class) in the associated criteria sections. Here is the formula used in cell G20, which counts the number of students in Calculus 101:

```
=DCOUNT(Students,"Final Grade",F8:G9)
```

G20 =DCOUNT(Students,"Final Grade",F8:G9)

	A	B	C	D	E	F	G
1	**Student ID**	**Class**	**Teacher**	**Final Grade**			
2	VM4128	Calculus 101	Mr. Crasdale	77		**Class**	**Teacher**
3	CH8965	Ancient Greece	Mr. Young	81		Accounting 101	
4	SG9555	Accounting 101	Ms. Morley	76			
5	AB5235	Calculus 101	Mr. Crasdale	86		**Class**	**Teacher**
6	KD0656	Accounting 101	Mr. Harris	98		Ancient Greece	
7	JG1183	Accounting 101	Mr. Harris	98			
8	HE7976	Masters of Philosophy	Mr. Crasdale	78		**Class**	**Teacher**
9	NR5090	Calculus 101	Mr. Porter	79		Calculus 101	
10	AJ3549	Accounting 101	Mr. Harris	85			
11	CW8495	English Literature	Mr. Johnson	66		**Class**	**Teacher**
12	NR5090	Calculus 101	Mr. Porter	88		English Literature	
13	QT6233	Accounting 101	Ms. Morley	73			
14	MP9632	English Literature	Ms. Rendson	84		**Class**	**Teacher**
15	DK7492	Accounting 101	Mr. Richards	71		Masters of Philosophy	
16	RE3968	Ancient Greece	Mr. Young	75			
17	AP8356	Accounting 101	Mr. Richards	88		*Average Grade per Course of Study*	
18	FL1423	Masters of Philosophy	Ms. Untermeyer	93		**Accounting 101**	**7**
19	BW2559	Calculus 101	Mr. Porter	86		**Ancient Greece**	**5**
20	QL9026	Calculus 101	Mr. Crasdale	77		**Calculus 101**	**8**
21	FE4903	English Literature	Ms. Appleson	74		**English Literature**	**7**
22	TY3987	Calculus 101	Mr. Porter	74		**Masters of Philosophy**	**2**
23	EE4688	English Literature	Mr. Johnson	73			
24	WD0448	Ancient Greece	Mr. Young	76			

FIGURE 17-10: Calculating the number of students in each course.

Note that DCOUNT requires a column of numbers to count. Therefore, the Final Grade heading is put in the function. Counting on Class or Teacher would result in zero. Using a column that specifically has numbers may seem a little odd. The function is not summing the numbers; it just counts the number of records. But what the heck? It works.

Now take this a step further. How about counting the number of students who got a grade of 90 or better in any class? How can this be done? This calculation requires a different criterion — one that selects all records where Final Grade is `90` or greater. Figure 17-11 shows a worksheet with this criterion and the calculated result shown.

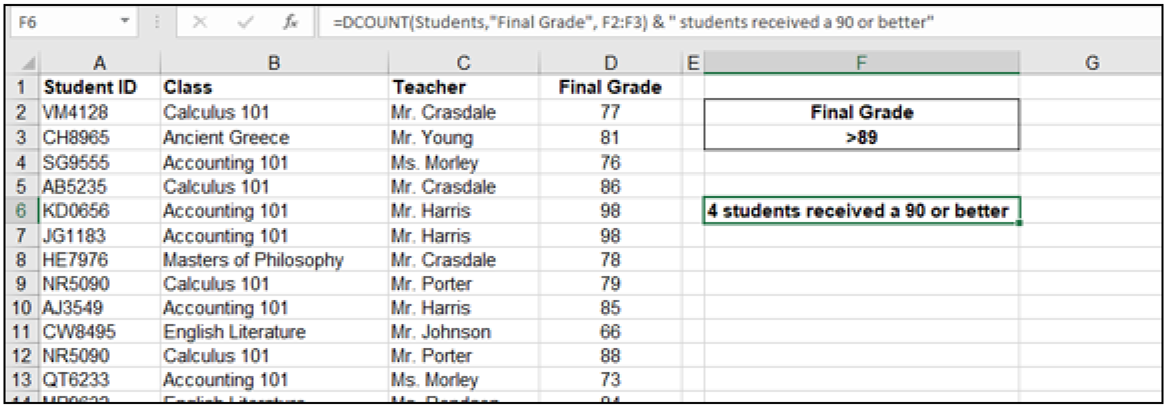


FIGURE 17-11: Calculating the number of students who earned a grade of 90 or better.

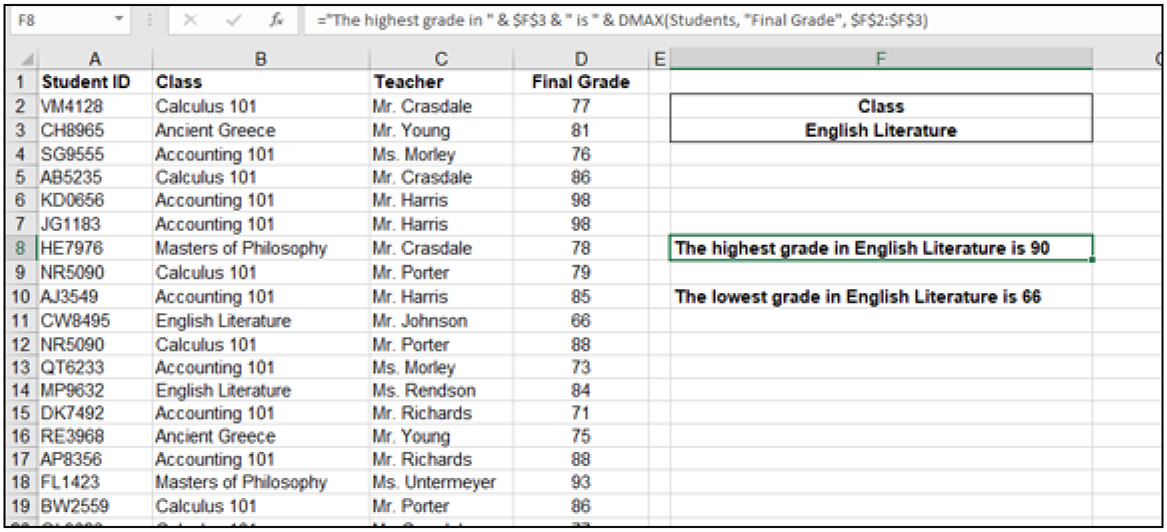
The result in cell F6 *concatenates* — that is, combines but does not add — the answer from the DCOUNT function with some text. The formula looks like this:

```
=DCOUNT(Students,"Final Grade",F2:F3) & " students
received a 90 or better."
```

The criterion specifically states to use all records where the Final Grade is greater than `89` (>89). You can specify >=90 with the exact same result.

Finding Highest and Lowest with DMIN and DMAX

The DMIN and DMAX functions find the minimum or maximum value, respectively, in a database column, for just the rows that match the criteria. Figure 17-12 shows how these two functions can find the highest and lowest grades for English Literature.



F8 =“The highest grade in ” & $F$3 & ” is ” & DMAX(Students, “Final Grade”, $F$2:$F$3)

	A	B	C	D	E	F
1	Student ID	Class	Teacher	Final Grade		
2	VM4128	Calculus 101	Mr. Crasdale	77		Class
3	CH8965	Ancient Greece	Mr. Young	81		English Literature
4	SG9555	Accounting 101	Ms. Morley	76		
5	AB5235	Calculus 101	Mr. Crasdale	86		
6	KD0656	Accounting 101	Mr. Harris	98		
7	JG1183	Accounting 101	Mr. Harris	98		
8	HE7976	Masters of Philosophy	Mr. Crasdale	78		The highest grade in English Literature is 90
9	NR5090	Calculus 101	Mr. Porter	79		
10	AJ3549	Accounting 101	Mr. Harris	85		The lowest grade in English Literature is 66
11	CW8495	English Literature	Mr. Johnson	66		
12	NR5090	Calculus 101	Mr. Porter	88		
13	QT6233	Accounting 101	Ms. Morley	73		
14	MP9632	English Literature	Ms. Rendson	84		
15	DK7492	Accounting 101	Mr. Richards	71		
16	RE3968	Ancient Greece	Mr. Young	75		
17	AP8356	Accounting 101	Mr. Richards	88		
18	FL1423	Masters of Philosophy	Ms. Untermeyer	93		
19	BW2559	Calculus 101	Mr. Porter	86		

FIGURE 17-12: Calculating the highest and lowest grades for a specified class.

The formulas in cells F8 and F10 are practically identical. Here is the formula in cell F8:

```
="The highest grade in " & $F$3 & " is " &
DMAX(Students,"Final Grade",$F$2:$F$3)
```

Finding Duplicate Values with DGET

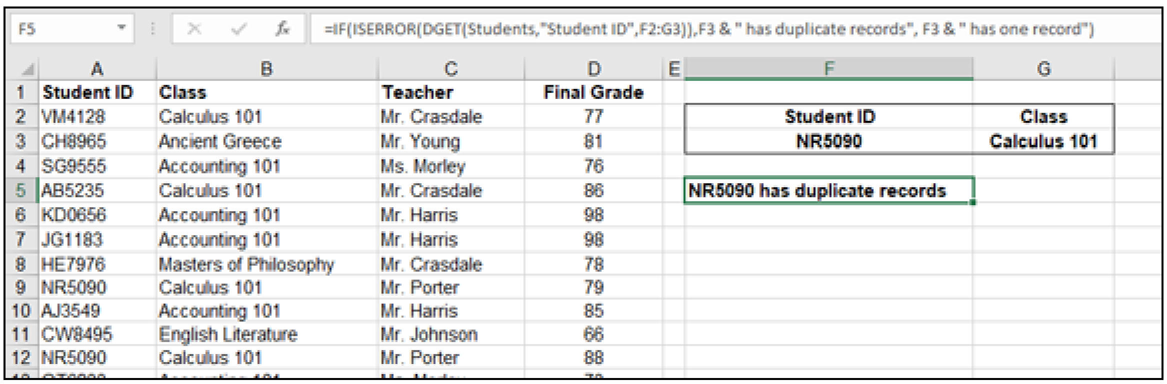
DGET is a unique database function. It does not perform a calculation but checks for duplicate entries. The function returns one of three values:

- If one record matches the criterion, DGET returns the criterion.
- If no records match the criterion, DGET returns the `#VALUE!` error.
- If more than one record matches the criterion, DGET returns the `#NUM!` error.

By testing to see whether DGET returns an error, you can discover problems with your data. Perhaps you suspect that a student has registered twice for a specific class. If this is true, two records will have the same Student ID and Class.

Figure 17-13 shows how to check whether student NR5090 is entered more than once for Calculus 101. If there is more than one record, DGET returns an error. Cell F5 contains a formula that nests the DGET function inside the ISERROR function; all that is inside the IF function. If DGET returns an error, return one message; if DGET does not return an error, return a different message. Here is the formula:

```
=IF(ISERROR(DGET(Students,"Student ID",F2:G3)),F3 &
" has duplicate records", F3 & " has one record")
```



F5 =IF(ISERROR(DGET(Students,"Student ID",F2:G3)),F3 & " has duplicate records", F3 & " has one record")

	A	B	C	D	E	F	G
1	Student ID	Class	Teacher	Final Grade			
2	VM4128	Calculus 101	Mr. Crasdale	77		Student ID	Class
3	CH8965	Ancient Greece	Mr. Young	81		NR5090	Calculus 101
4	SG9555	Accounting 101	Ms. Morley	76			
5	AB5235	Calculus 101	Mr. Crasdale	86		NR5090 has duplicate records	
6	KD0656	Accounting 101	Mr. Harris	98			
7	JG1183	Accounting 101	Mr. Harris	98			
8	HE7976	Masters of Philosophy	Mr. Crasdale	78			
9	NR5090	Calculus 101	Mr. Porter	79			
10	AJ3549	Accounting 101	Mr. Harris	85			
11	CW8495	English Literature	Mr. Johnson	66			
12	NR5090	Calculus 101	Mr. Porter	88			

FIGURE 17-13: Using DGET to test for duplicate records in a database.

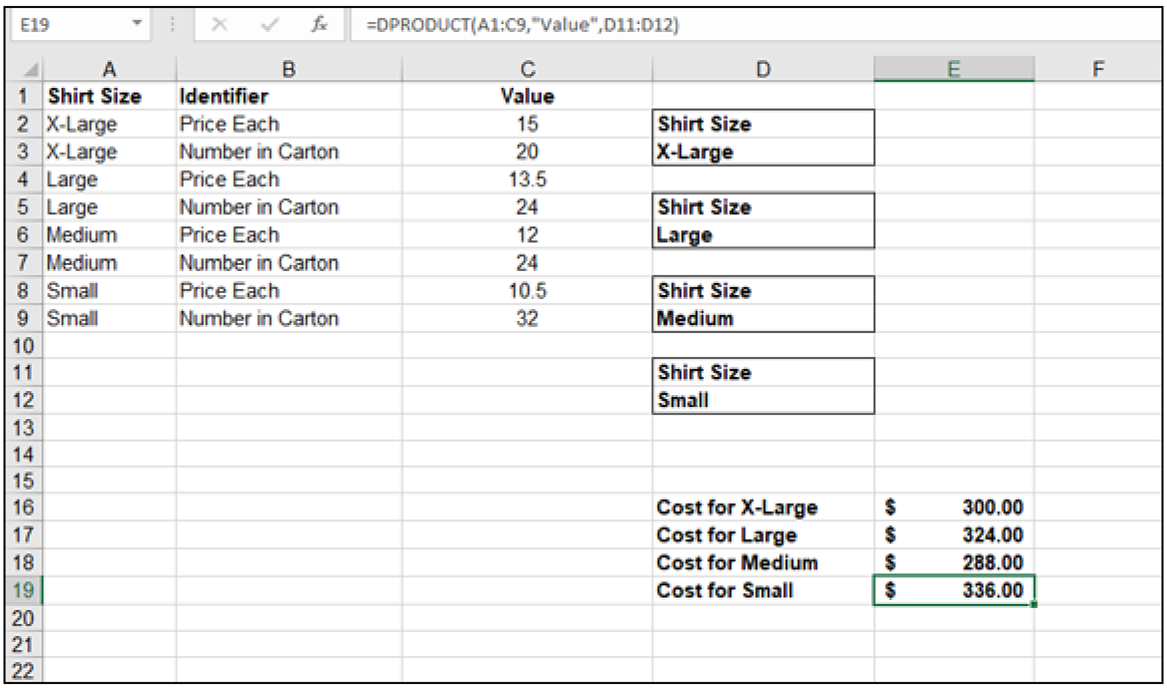
Being Productive with DPRODUCT

DPRODUCT multiplies values that match the criterion in a database. This is powerful but also able to produce results that are not the intention. In other words, it's one to thing to add and derive a sum. That is a common operation on a set of data. Looking back at Figure 17-8, you can see that the total sales for Jack Bennet, $79,134, are the sum of three amounts: $43,234, $12,450, and

$23,450. If multiplication were applied to the three amounts, the answer (the product) would be $12,622,274,385,000. Oops! That's over 12 *trillion* dollars!

DPRODUCT multiplies and, therefore, is not likely to be used as often as a function like DSUM, but when you need to multiply items in a database, DPRODUCT is a tool of choice.

Figure 17-14 shows a situation in which DPRODUCT is productive. The database area contains shirts. For each shirt size, there are two rows: the price per shirt and the number of shirts that are packed in a carton. The cost for a carton of shirts is, therefore, the product of the price per shirt times the number of shirts. There are four shirt sizes, each with its own price and carton count.



E19 =DPRODUCT(A1:C9,"Value",D11:D12)

	A	B	C	D	E	F
1	**Shirt Size**	**Identifier**	**Value**			
2	X-Large	Price Each	15	**Shirt Size**		
3	X-Large	Number in Carton	20	**X-Large**		
4	Large	Price Each	13.5			
5	Large	Number in Carton	24	**Shirt Size**		
6	Medium	Price Each	12	**Large**		
7	Medium	Number in Carton	24			
8	Small	Price Each	10.5	**Shirt Size**		
9	Small	Number in Carton	32	**Medium**		
10						
11				**Shirt Size**		
12				**Small**		
13						
14						
15						
16				**Cost for X-Large**	**$ 300.00**	
17				**Cost for Large**	**$ 324.00**	
18				**Cost for Medium**	**$ 288.00**	
19				**Cost for Small**	**$ 336.00**	
20						
21						
22						

FIGURE 17-14: Calculating the total costs of cartons filled with shirts.

To make sure you work with just one size per use of DPRODUCT, four criteria areas are set up — one for each size. Any single criteria area has the Shirt Size heading and the actual shirt size, such as Medium. For example, D8:D9 contains the criteria for medium-size shirts.

Four cells each contain DPRODUCT, and within each cell, the particular criteria area is used. For example, cell E18 has this formula:

```
=DPRODUCT(A1:C9, "Value", D8:D9)
```

The database range is A1:C9. Value is the field the function looks in for values to multiply, and the multiplication occurs on values for which the shirt size matches the criteria.

A worksheet set up like the one shown in Figure 17-14 is especially useful when new data are occasionally pasted into the database area. The set of DPRODUCT functions will always provide the products based on whatever data are placed in the database area. This particular example of DPRODUCT shows how to work with data in which more than one row pertains to an item. In this case, each shirt size has a row showing the price per shirt and a second row showing the number of shirts that fit in a carton.

Financial Formulas

Financial Functions 101

The key to using any of Excel's financial functions is to understand the terminology used by their arguments. Many of the most common financial functions, such as PV (Present Value), NPV (Net Present Value), FV (Future Value), PMT (Payment), and IPMT (Interest Payment) take similar arguments:

- **PV** is the present value that is the principal amount of the annuity.
- **FV** is the future value that represents the principal plus interest on the annuity.
- **PMT** is the payment made each period in the annuity. Normally, the payment is set over the life of the annuity and includes principal plus interest without any other fees.
- **RATE** is the interest rate per period. Normally, the rate is expressed as an annual percentage.

- **NPER** is the total number of payment periods in the life of the annuity. You calculate this number by taking the Term (the amount of time that interest is paid) and multiplying it by the Period (the point in time when interest is paid or earned) so that a loan with a 3-year term with 12 monthly interest payments has 3 x 12, or 36 payment periods.

REMEMBER When using financial functions, keep in mind that the *fv, pv,* and *pmt* arguments can be positive or negative, depending on whether you're receiving the money (as in the case of an investment) or paying out the money (as in the case of a loan). Also keep in mind that you want to express the *rate* argument in the same units as the *nper* argument, so that if you make monthly payments on a loan and you express the *nper* as the total number of monthly payments, as in 360 (30 x 12) for a 30-year mortgage, you need to express the annual interest rate in monthly terms as well. For example, if you pay an annual interest rate of 7.5 percent on the loan, you express the *rate* argument as 0.075/12 so that it is monthly as well.

The PV, NPV, and FV Functions

The PV (Present Value), NPV (Net Present Value), and FV (Future Value) functions all found on the Financial button's drop-down menu on the Ribbon's Formulas tab (Alt+MI) enable you to determine the profitability of an investment.

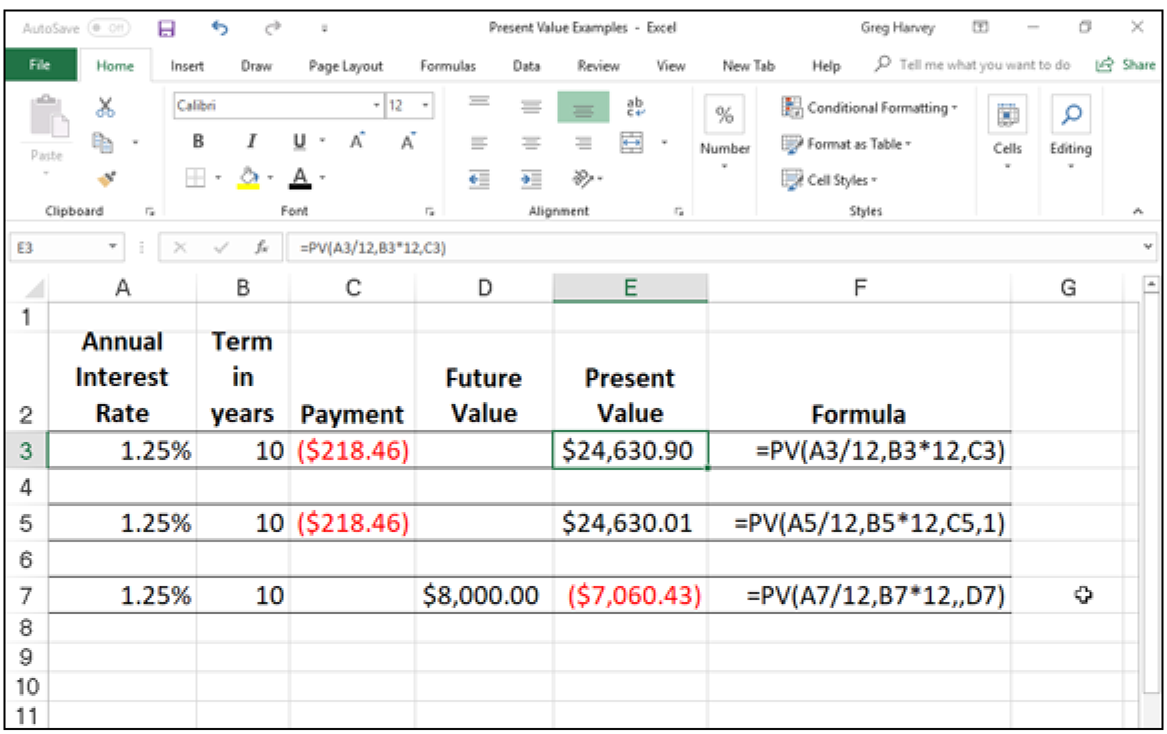
CALCULATING THE PRESENT VALUE

The PV, or Present Value, function returns the present value of an investment, which is the total amount that a series of future payments is worth presently. The syntax of the PV function is as follows:

```
=PV(rate,nper,pmt,[fv],[type])
```

The *fv* and *type* arguments are optional arguments in the function (indicated by the square brackets). The *fv* argument is the future value or cash balance that you want to have after making your last payment. If you omit the *fv* argument, Excel assumes a future value of zero (0). The *type* argument indicates whether the payment is made at the beginning or end of the period: Enter 0 (or omit the *type* argument) when the payment is made at the end of the period, and use 1 when it is made at the beginning of the period.

Figure 4-1 contains several examples using the PV function. All three PV functions use the same annual percentage rate of 1.25 percent and term of 10 years. Because payments are made monthly, each function converts these annual figures into monthly ones. For example, in the PV function in cell E3, the annual interest rate in cell A3 is converted into a monthly rate by dividing by 12 (A3/12). The annual term in cell B3 is converted into equivalent monthly periods by multiplying by 12 (B3 x 12).



	A	B	C	D	E	F
2	Annual Interest Rate	Term in years	Payment	Future Value	Present Value	Formula
3	1.25%	10	($218.46)		$24,630.90	=PV(A3/12,B3*12,C3)
5	1.25%	10	($218.46)		$24,630.01	=PV(A5/12,B5*12,C5,1)
7	1.25%	10		$8,000.00	($7,060.43)	=PV(A7/12,B7*12,,D7)

FIGURE 4-1: Using the PV function to calculate the present value of various investments.

Note that although the PV functions in cells E3 and E5 use the *rate, nper,* and *pmt* ($218.46) arguments, their results are slightly different. This is caused by the difference in the *type* argument in the two functions: the PV function in cell E3 assumes that each payment is made at the end of the period (the *type* argument is 0 whenever it is omitted), whereas the PV function in cell E5 assumes that each payment is made at the beginning of the period (indicated by a *type* argument of 1). When the payment is made at the beginning of the period, the present value of this investment is $0.89 higher than when the payment is made at the end of the period, reflecting the interest accrued during the last period.

The third example in cell E7 (shown in Figure 4-1) uses the PV function with an *fv* argument instead of the *pmt* argument. In this example, the PV function states that you would have to make monthly payments of $7,060.43 for a 10-year period to realize a cash balance of $8,000, assuming that the investment returned a constant annual interest rate of 1 1/4 percent. Note that when you use the PV function with the *fv* argument instead of the *pmt* argument, you must still indicate the position of the *pmt* argument in the function with a comma (thus the two commas in a row in the function) so that Excel doesn't mistake your *fv* argument for the *pmt* argument.

CALCULATING THE NET PRESENT VALUE

The NPV function calculates the net present value based on a series of cash flows. The syntax of this function is

```
=NPV(rate,value1,[value2],[…])
```

where *value1*, *value2,* and so on are between 1 and 13 value arguments representing a series of payments (negative values) and income (positive values), each of which is equally spaced in time and occurs at the end of the period. The NPV investment begins one period before the period of the *value1* cash flow and ends with the last cash flow in the argument list. If your first cash flow occurs at the beginning of the period, you must add it to the result of the NPV function rather than include it as one of the arguments.

Figure 4-2 illustrates the use of the NPV function to evaluate the attractiveness of a five-year investment that requires an initial investment of $30,000 (the value in cell G3). The first year, you expect a loss of $22,000 (cell B3); the second year, a profit of $15,000 (cell C3); the third year, a profit of $25,000 (cell D3); the fourth year, a profit of $32,000 (cell E3); and the fifth year, a profit of $38,000 (cell F3). Note that these cell references are used as the *value* arguments of the NPV function.

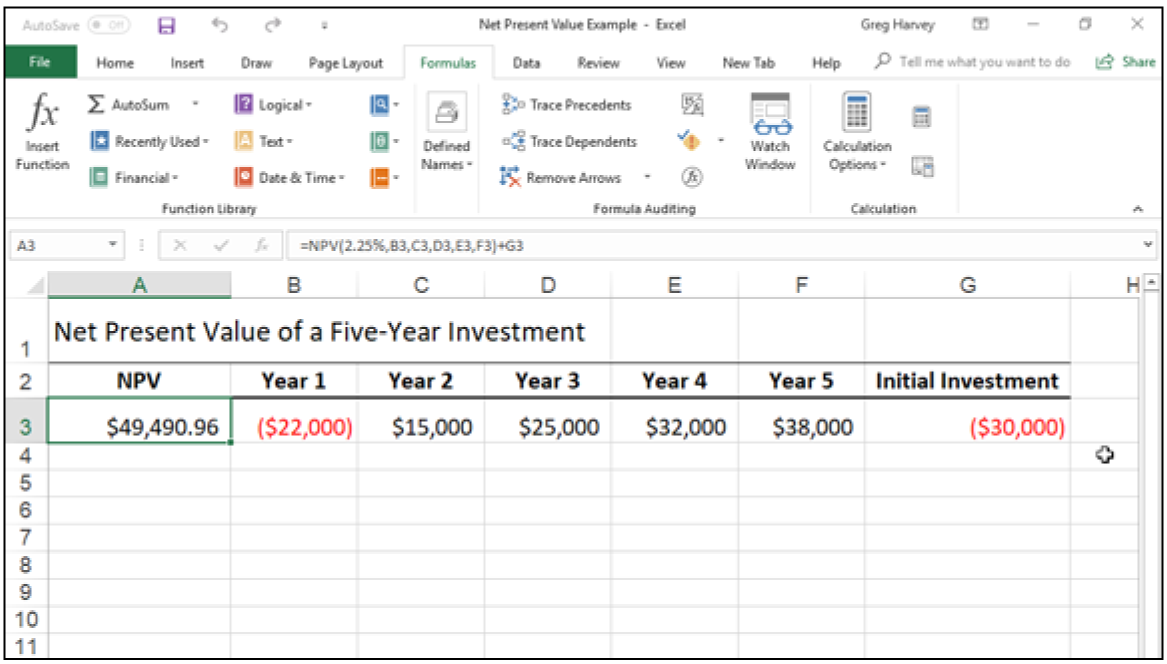


FIGURE 4-2: Using the NPV function to calculate the net present value of an investment.

Unlike when using the PV function, the NPV function doesn’t require an even stream of cash flows. The *rate* argument in the function is set at 2.25 percent. In this example, this represents the *discount rate* of the investment — that is, the interest rate that you may expect to get during the five-year period if you put your money into some other type of investment, such as a high-yield money-market account. This NPV function in cell A3 returns a net present value of $49,490.96, indicating that you can expect to realize a great deal more from investing your $30,000 in this investment than you possibly could from investing the money in a money-market account at the interest rate of 2.25 percent.

CALCULATING THE FUTURE VALUE

The FV function calculates the future value of an investment. The syntax of this function is

```
=FV(rate,nper,pmt,[pv],[type])
```

The *rate*, *nper*, *pmt*, and *type* arguments are the same as those used by the PV function. The *pv* argument is the present value or lump-sum amount for which you want to calculate the future value. As with the *fv* and *type* arguments in the PV function, both the *pv* and *type* arguments are optional in the FV function. If you omit these arguments, Excel assumes their values to be zero (0) in the function.

You can use the FV function to calculate the future value of an investment, such as an IRA (Individual Retirement Account). For example, suppose that you establish an IRA at age 43 and will retire 22 years from now at age 65 and that you plan to make annual payments into the IRA at the beginning of each year. If you assume a rate of return of 2.5 percent a year, you would enter the following FV function in your worksheet:

```
=FV(2.5%,22,-1500,,1)
```

Excel then indicates that you can expect a future value of $44,376.64 for your IRA when you retire at age 65. If you had established the IRA a year prior and the account already has a present value of $1,538, you would amend the FV function as follows:

```
=FV(2.5%,22,-1500,-1538,1)
```

In this case, Excel indicates that you can expect a future value of $47,024.42 for your IRA at retirement.

The PMT Function

The PMT function on the Financial button's drop-down menu on the Formulas tab of the Ribbon calculates the periodic payment for an annuity, assuming a stream of equal payments and a constant rate of interest. The PMT function uses the following syntax:

```
=PMT(rate,nper,pv,[fv],[type])
```

As with the other common financial functions, *rate* is the interest rate per period, *nper* is the number of periods, *pv* is the present value or the amount the future payments are worth presently, *fv* is the future value or cash balance that you want after the last payment is made (Excel assumes a future value of zero when you omit this optional argument as you would when calculating loan payments), and *type* is the value 0 for payments made at the end of the period or the value 1 for payments made at the beginning of the period. (If you omit the optional *type* argument, Excel assumes that the payment is made at the end of the period.)

The PMT function is often used to calculate the payment for mortgage loans that have a fixed rate of interest. Figure 4-3 shows you a sample worksheet that contains a table using the PMT function to calculate loan payments for a range of interest rates (from 2.75 percent to 4.00 percent) and principals ($150,000 to $159,000). The table uses the initial principal that you enter in cell B2, copies it to cell A7, and then increases it by $1,000 in the range A8:A16. The table uses the initial interest rate that you enter in cell B3, copies to cell B6, and then increases this initial rate by 1/4 of a percent in the range C6:G6. The term in years in cell B4 is a constant factor that is used in the entire loan payment table.

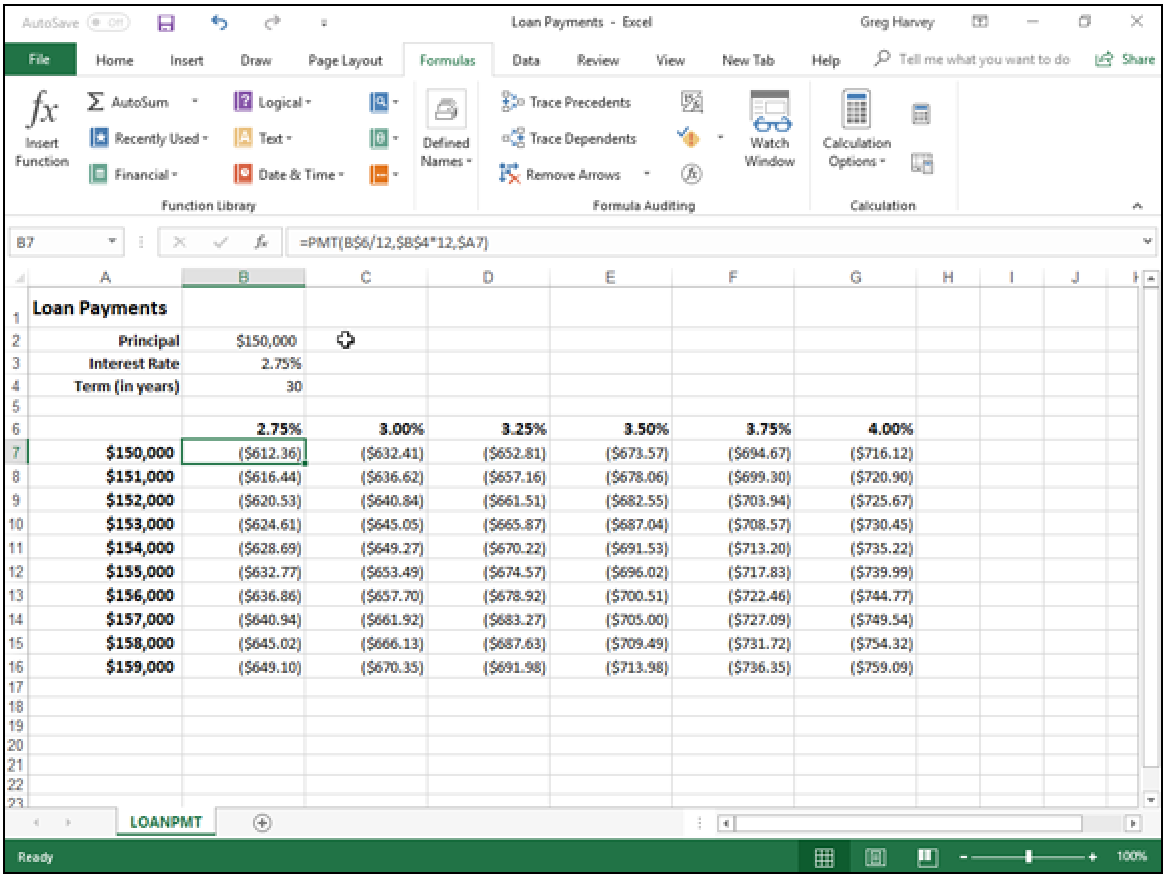


FIGURE 4-3: Loan Payments table using the PMT function to calculate various loan payments.

To get an idea of how easy it is to build this type of loan payment table with the PMT function, follow these steps for creating it in a new worksheet:

1. **Enter the titles** Loan Payments **in cell A1,** Principal **in cell A2,** Interest Rate **in cell A3,** and Term (in years) **in cell A4.**
2. **Enter** $150,000 **in cell B2, enter** 2.75% **in cell B3, and enter** 30 **in cell B4.**

   These are the starting values with which you build the Loan Payments table.
3. **Position the cell pointer in B6 and then build the formula =B3.**

   By creating a linking formula that brings forward the starting interest rate value in B3 with the formula, you ensure that the interest rate value in B6 will immediately reflect any change that you make in cell B3.

4. **Position the cell pointer in cell C6 and then build the formula =B6+.25%.**
   By adding 1/4 of a percent to the interest rate to the value in B6 with the formula =B6+0.25% in C6 rather than creating a series with the AutoFill handle, you ensure that the interest rate value in cell C6 will always be 1/4 of a percent larger than any interest rate value entered in cell B6.
5. **Drag the Fill handle in cell C6 to extend the selection to the right to cell G6.**
6. **Position the cell pointer in cell A7 and then build the formula =B2.**
   Again, by using the formula =B2 to bring the initial principal forward to cell A7, you ensure that cell A7 always has the same value as cell B2.
7. **Position the cell pointer in A8 active and then build the formula =A7+1000.**
   Here too, you use the formula =A7+1000 rather than create a series with the AutoFill feature so that the principal value in A8 will always be $1,000 greater than any value placed in cell A7.
8. **Drag the Fill handle in cell A8 down until you extend the selection to cell A16 and then release the mouse button.**
9. **In cell B7, click the Insert Function button on the Formula bar, select Financial from the Or Select a Category drop-down list, and then double-click the PMT function in the Select a Function list box.**
   The Function Arguments dialog box that opens allows you to specify the *rate*, *nper*, and *pv* arguments. Be sure to move the Function Arguments dialog box to the right so that no part of it obscures the data in columns A and B of your worksheet before proceeding with the following steps for filling in the arguments.
10. **Click cell B6 to insert B6 in the Rate text box and then press F4 twice to convert the relative reference B6 to the mixed reference B$6 (column relative, row absolute) before you type** /12**.**
    You convert the relative cell reference B6 to the mixed reference B$6 so that Excel does *not* adjust the row number when you copy the PMT formula down each row of the table, but it *does* adjust the column letter when you copy the formula across its columns. Because the initial interest rate entered in B3 (and then brought forward to cell B6) is an *annual* interest rate, but you want to know the *monthly* loan payment, you need to

convert the annual rate to a monthly rate by dividing the value in cell B6 by 12.

11. **Click the Nper text box, click cell B4 to insert this cell reference in this text box, and then press F4 once to convert the relative reference B4 to the absolute reference $B$4 before you type** *12**.**
    You need to convert the relative cell reference B4 to the absolute reference $B$4 so that Excel adjusts neither the row number nor the column letter when you copy the PMT formula down the rows and across the columns of the table. Because the term is an *annual* period, but you want to know the *monthly* loan payment, you need to convert the yearly periods to monthly periods by multiplying the value in cell B4 by 12.
12. **Click the Pv text box, click A7 to insert this cell reference in this text box, and then press F4 three times to convert the relative reference A7 to the mixed reference $A7 (column absolute, row relative).**
    You need to convert the relative cell reference A7 to the mixed reference $A7 so that Excel won't adjust the column letter when you copy the PMT formula across each column of the table, but will adjust the row number when you copy the formula down across its rows.
13. **Click OK to insert the formula =PMT(B$6/12,$B$4*12,$A7) in cell B7.**
    Now you're ready to copy this original PMT formula down and then over to fill in the entire Loan Payments table.
14. **Drag the Fill handle on cell B7 down until you extend the fill range to cell B16 and then release the mouse button.**
    After you've copied the original PMT formula down to cell B16, you're ready to copy it to the right to G16.
15. **Drag the Fill handle to the right until you extend the fill range B7:B16 to cell G16 and then release the mouse button.**

After copying the original formula with the Fill handle, be sure to widen columns B through G sufficiently to display their results. (You can do this in one step by dragging through the headers of these columns and then double-clicking the right border of column G.)

After you've created a loan table like this, you can then change the beginning principal or interest rate, as well as the term to see what the payments would be under various other scenarios. You can also turn on the Manual Recalculation so that you can control when the Loan Payments table is recalculated.

For information on how to switch to manual recalculation and use this mode to control when formulas are recalculated, see Book 3, Chapter 1. For information on how to protect the worksheet so that users can input new values only into the three input cells (B2, B3, and B4) to change the starting loan amount, interest rate, and the term of the loan, see Book 4, Chapter 1.

Depreciation Functions

Excel lets you choose from four different depreciation functions, each of which uses a slightly different method for depreciating an asset over time. These built-in depreciation functions found on the Financial button's drop-down menu on the Formulas tab of the Ribbon include the following:

- SLN(*cost,salvage,life*) to calculate straight-line depreciation
- SYD(*cost,salvage,life,per*) to calculate sum-of-years-digits depreciation
- DB(*cost,salvage,life,period*,[*month*]) to calculate declining balance depreciation
- DDB(*cost,salvage,life,period*,[*factor*]) to calculate double-declining balance depreciation

As you can see, with the exception of the optional *month* argument in the DB function and the optional *factor* argument in the DDB function, all the depreciation functions require the *cost, salvage,* and *life* arguments, and all but the SLN function require a *period* argument as well:

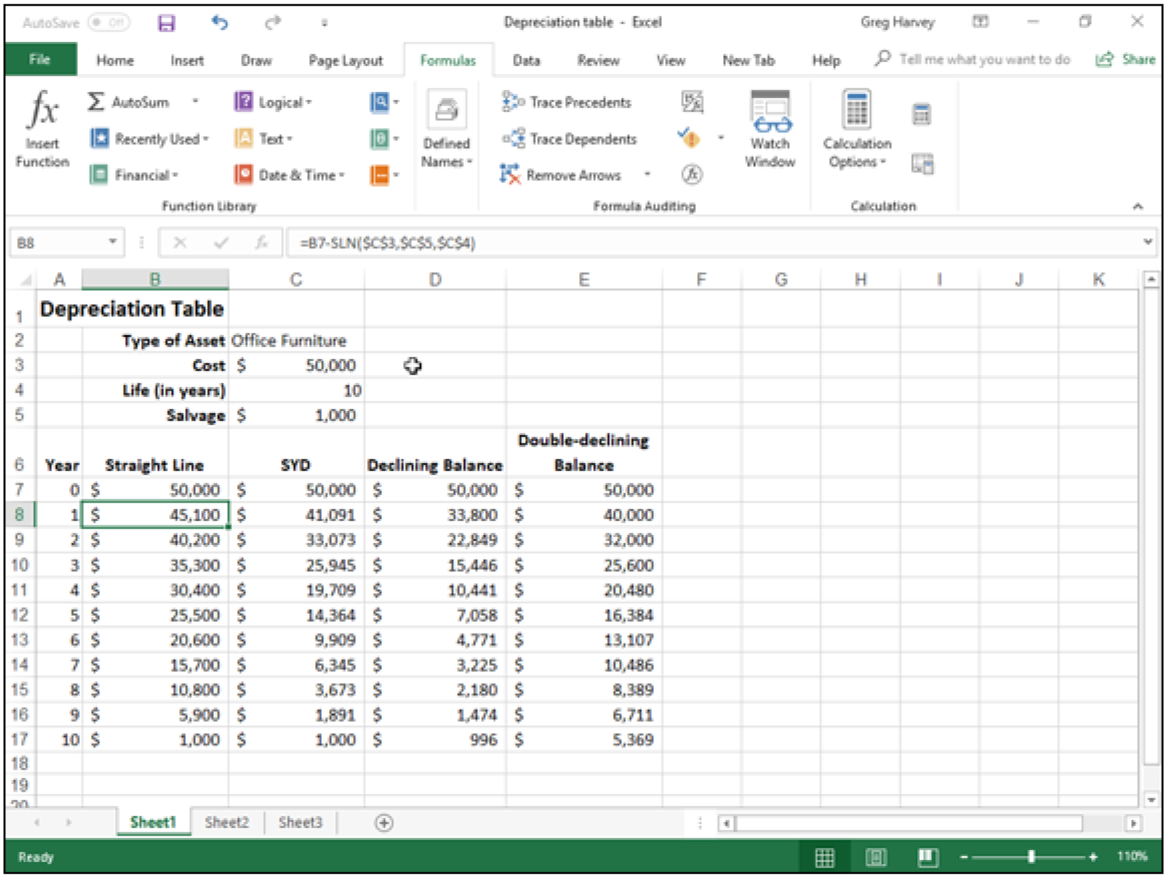
- *Cost* is the initial cost of the asset that you're depreciating.

- *Salvage* is the value of the asset at the end of the depreciation (also known as the salvage value of the asset).
- *Life* is the number of periods over which the asset is depreciating (also known as the useful life of the asset).
- *Per* or *period* is the period over which the asset is being depreciated. The units that you use in the *period* argument must be the same as those used in the *life* argument of the depreciation function so that if you express the life argument in years, you must also express the *period* argument in years.

Note that the DB function accepts an optional *month* argument. This argument is the number of months that the asset is in use in the first year. If you omit the *month* argument from your DB function, Excel assumes the number of months of service to be 12.

When using the DDB function to calculate the double-declining balance method of depreciation, you can add an optional *factor* argument. This argument is the rate at which the balance declines in the depreciation schedule. If you omit this optional *factor* argument, Excel assumes the rate to be 2 (thus, the name *double-declining balance*).

Figure 4-4 contains a Depreciation table that uses all four depreciation methods to calculate the depreciation of office furniture originally costing $50,000 to be depreciated over a 10-year period, assuming a salvage value of $1,000 at the end of this depreciation period.



	A	B	C	D	E
1	Depreciation Table				
2		Type of Asset	Office Furniture		
3		Cost	$ 50,000		
4		Life (in years)	10		
5		Salvage	$ 1,000		
6	Year	Straight Line	SYD	Declining Balance	Double-declining Balance
7	0	$ 50,000	$ 50,000	$ 50,000	$ 50,000
8	1	$ 45,100	$ 41,091	$ 33,800	$ 40,000
9	2	$ 40,200	$ 33,073	$ 22,849	$ 32,000
10	3	$ 35,300	$ 25,945	$ 15,446	$ 25,600
11	4	$ 30,400	$ 19,709	$ 10,441	$ 20,480
12	5	$ 25,500	$ 14,364	$ 7,058	$ 16,384
13	6	$ 20,600	$ 9,909	$ 4,771	$ 13,107
14	7	$ 15,700	$ 6,345	$ 3,225	$ 10,486
15	8	$ 10,800	$ 3,673	$ 2,180	$ 8,389
16	9	$ 5,900	$ 1,891	$ 1,474	$ 6,711
17	10	$ 1,000	$ 1,000	$ 996	$ 5,369

FIGURE 4-4: A Depreciation table showing 10-year depreciation of an asset using various methods.

The Formula bar shown in Figure 4-4 shows the SLN formula that I entered into cell B8:

```
=B7-SLN($C$3,$C$5,$C$4)
```

This formula subtracts the amount of straight-line depreciation to be taken in the first year of service from the original cost of $50,000. (This value is brought forward from cell C3 by the formula =C3.) After creating this original formula in cell B8, I then used the Fill handle to copy it down to cell B17, which contains the final salvage value of the asset in the 10th year of service.

Cell C8 contains a similar formula for calculating the sum-of-years-digits depreciation for the office furniture. This cell contains the following formula:

```
=C7-SYD($C$3,$C$5,$C$4,$A8)
```

This formula subtracts the amount of sum-of-years-digits depreciation to be taken at the end of the first year from the original cost of $50,000 in cell C7 (also brought forward from cell C3 by the formula =C3). After creating this original formula in cell C8, I again used the Fill handle to copy it down to cell C17, which also contains the final salvage value of the asset in the 10th year of service.

I used the same basic procedure to create the formulas using the DB and DDB depreciation methods in the cell ranges D8:D17 and E8:E17, respectively. Cell D8 contains the following DB formula:

```
=D7-DB($C$3,$C$5,$C$4,$A8)
```

Cell E8 contains the following DDB formula:

```
=E7-DDB($C$3,$C$5,$C$4,$A8)
```

Note that, like the SYD function, both of these depreciation functions require the use of a *period* argument, which is supplied by the list of years in the cell range A8:A17. Note also that the value in cell C4, which supplies the *life* argument to the SYD, DB, and DDB functions, matches the year units used in this cell range.

Analysis ToolPak Financial Functions

By activating the Analysis ToolPak add-in (see Book 1, Chapter 2), you add a whole bunch of powerful financial functions to the Financial button's drop-down menu on the Formulas tab of the Ribbon. Table 4-1 shows all the financial functions that are added to the Insert Function dialog box when the Analysis ToolPak is activated. As you can see from this table, the Analysis ToolPak financial functions are varied and quite sophisticated.

TABLE 4-1 Financial Functions in the Analysis ToolPak

Function	***What It Calculates***
ACCRINT(issue,first_interest,settlement,rate,[par],frequency,[basis],[calc_methd])	Calculates the accrued interest for a security that pays periodic interest.
ACCRINTM(issue,maturity,rate,[par],[basis])	Calculates the accrued interest for a security that pays interest at

maturit
y.

AMORDEGRC(cost,date_purchased,first_period,salvage, period,rate,[basis]) and AMORLINC(cost, date_purchased,first_period,salvage,period,rate,[basis])	Used in French accoun ting system s for calcula ting deprec iation. AMOR DEGR C and AMOR LINC return the deprec iation for each accoun ting period. AMOR DEGR C works like AMOR LINC except

that it applies a depreciation coefficient in the calculation that depends upon the life of the assets.

COUPDAYBS(settlement,maturity,frequency,[basis])	Calculates the number of days from the beginning of a coupon period to the settlement date.
COUPDAYS(settlement,maturity,frequency,[basis])	Calculates the number of days in the coupon period.

COUPDAYSNC(settlement,maturity,frequency,[basis])	Calculates the number of days from the settlement date to the next coupon date.
COUPNCD(settlement,maturity,frequency,[basis])	Calculates a number that represents the next coupon date after a settlement date.

COUPNUM(settlement,maturity,frequency,[basis])	Calculates the number of coupons payable between the settlement date and maturity date, rounded up to the nearest whole coupon.

COUPPCD(settlement,maturity,frequency,[basis])	Calculates a number that represents the previous coupon date before the settlement date.

CUMIPMT(rate,nper,pv,start_period,end_period,type)	Calculates the cumulative interest paid on a loan between the start_period and end_period. The type argument is 0 when the payment is made at the end of the period and 1 when it's made

at the beginning of the period.

CUMPRINC(rate,nper,pv,start_period,end_period,type)	Calculates the cumulative principal paid on a loan between the start_period and end_period. The type argument is 0 when the payment is made at the end of the period and 1 when it's made

	at the beginning of the period.
DISC(settlement,maturity,pr,redemption,[basis])	Calculates the discount rate for a

	security.
DOLLARDE(fractional_dollar,fraction)	Converts a dollar price expressed as a fraction into a dollar price expressed as a decimal number.

DOLLARFR(decimal_dollar,fraction)	Converts a dollar price expressed as a decimal number into a dollar price expressed as a fraction.

DURATION(settlement,maturity,coupon,yld,frequency,[basis])	Calculates the Macauley duration for an assumed par value of $100. (Duration is defined as the weighted average of the present value of the cash flows and is used as a measure of

the respon se of a bond price to chang es in yield.)

EFFECT(nominal_rate,npery)	Calculates the effective annual interest rate given the nominal interest rate and the number of compounding periods per year.
INTRATE(settlement,maturity,investment,redemption,[basis])	Calculates the interest rate for a fully investe

	d security.
MDURATION(settlement,maturity,coupon,yld,frequency,[basis])	Calculates the modified Macauley duration for a security with an assumed part value of $100.

NOMINAL(effect_rate,npery)	Calculates the nominal annual interest rate given the effect rate and the number of compounding periods per year.
ODDFPRICE(settlement,maturity,issue,first_coupon,rate,yld,redemption,frequency,[basis])	Calculates the price per $100 face value of a securit

	y having an odd (short or long) first period.
ODDFYIELD(settlement,maturity,issue,first_coupon,rate,pr,redemption,frequency,[basis])	Calculates the yield of a security that has an odd (short or long) first period.

ODDLPRICE(settlement,maturity, last_interest,rate,yld,redemption,frequency,[basis])	Calculates the price per $100 face value of a security having an odd (short or long) last coupon period.
ODDLYIELD(settlement,maturity,last_interest,rate,pr,redemption,frequency,[basis])	Calculates the yield of a security that has an odd (short or

	long) last period.
PRICE(settlement,maturity,rate,yld,redemption,frequency,[basis])	Calculates the price per $100 face value of a security that pays periodic interest.

PRICEDISC(settlement,maturity,discount,redemption,[basis])	Calculates the price per $100 face value of a discounted security.
PRICEMAT(settlement,maturity,issue,rate,yld,[basis])	Calculates the price per $100 face value of a security that pays interest at maturity.

RECEIVED(settlement,maturity,investment,discount,[basis])	Calculates the amount received at maturity for a fully invested security.
TBILLEQ(settlement,maturity,discount)	Calculates the bond-equivalent yield for a Treasury bill.

TBILLPRICE(settlement,maturity,discount)	Calculates the price per $100 face value for a Treasury bill.
TBILLYIELD(settlement,maturity,pr)	Calculates the yield for a Treasury bill.
XIRR(values,dates,[guess])	Calculates the internal rate of return for a schedule of cash

	flows that are not periodic.
XNPV(rate,values,dates)	Calculates the net present value for a schedule of cash flows that are not periodic.

YIELD(settlement,maturity,rate,pr,redemption,frequency,[basis])	Calculates the yield on a security that pays periodic interest (used to calculate bond yield).
YIELDDISC(settlement,maturity,pr,redemption,[basis])	Calculates the annual yield for a discounted security.

YIELDMAT(settlement,maturity,issue,rate,pr,[basis])	Calculates the annual yield of a security that pays interest at maturity.

You may note in Table 4-1 that many of the Analysis ToolPak financial functions make use of an optional *basis* argument. This optional *basis* argument is a number between 0 and 4 that determines the day count basis to use in determining the fractional part of the year:

- 0 (or omitted) to base it on the U.S. (NASD) method of 30/360 (see the coverage on the DAYS360 function in Book 3, Chapter 3 for details on the U.S. method)
- 1 to base the fraction on actual days/actual days
- 2 to base the fraction on actual days/360
- 3 to base the fraction on actual days/365
- 4 to base the fraction on the European method of 30/360 (see the DAYS360 coverage in Book 3, Chapter 3 for details on the European method)

REMEMBER For detailed information on the other required arguments in the Analysis ToolPak financial functions shown in this table, select the function from the Financial button’s drop-down list and then click the Help on This Function link in the lower-left corner of its Function Arguments dialog box.

Ten Tips for Working with Formulas

Master Operator Precedence

One of the most important factors in writing formulas is getting the operators correct, and I do not mean telephone-company operators. This has to do with mathematical operators — you know, little details such as plus signs, and multiplication signs, and where the parentheses go. *Operator precedence* — the order in which operations are performed — can make a big difference in the result. You have an easy way to keep your operator precedence in order. All you have to remember is “Please excuse my dear Aunt Sally.”

No, I have not lost my mind! This phrase is a mnemonic for the following:

- Parentheses

- Exponents
- Multiplication
- Division
- Addition
- Subtraction

Thus, parentheses have the first (highest) precedence, and subtraction has the last precedence. Well, to be honest, multiplication has the same precedence as division and addition has the same precedence as subtraction, but you get the idea!

For example, the formula `=1 + 2 × 15` equals 31. If you think it should equal 45, you'd better go visit your aunt! The answer equals 45 if you include parentheses, such as this: `=(1 + 2) × 15`.

WARNING Getting the order of the operators correct is critical to the well-being of your worksheet. Excel generates an error when the numbers of open and closed parentheses do not match, but if you mean to add two numbers before the multiplication, Excel does not know that you simply left the parentheses out!

A few minutes of refreshing your memory on operator order can save you a lot of headaches down the road.

Display Formulas

In case you haven’t noticed, it’s kind of hard to view your formulas without accidentally editing them. That’s because any time you are in “edit” mode and the active cell has a formula, the formula may incorporate the address of any other cell you click. This totally messes things up.

Wouldn’t it be easy if you could just *look* at all your formulas? There is a way! It’s simple. Click File at the top left of the Excel workspace, click Options, click the Advanced tab, and then scroll down to the Display options for this worksheet section (see Figure 18-1).

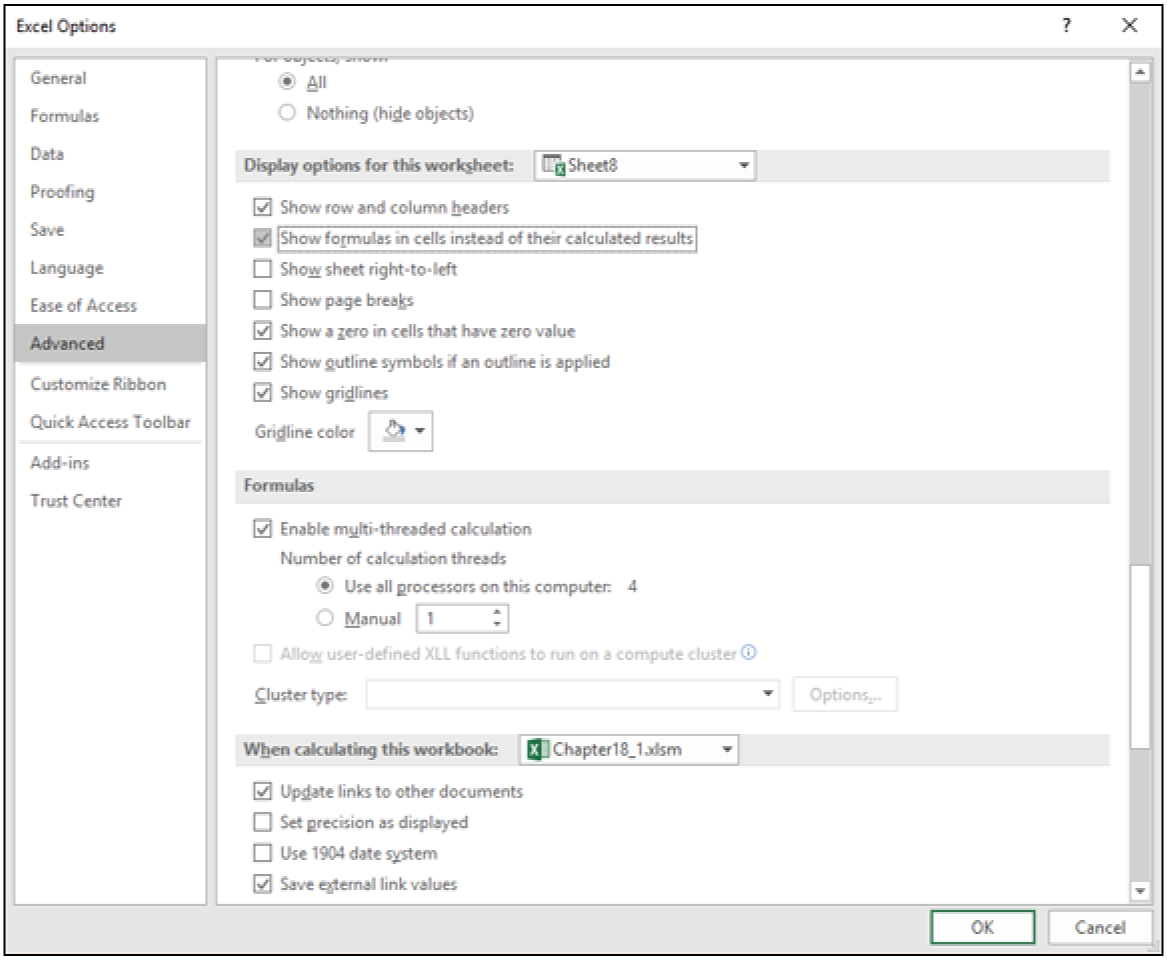
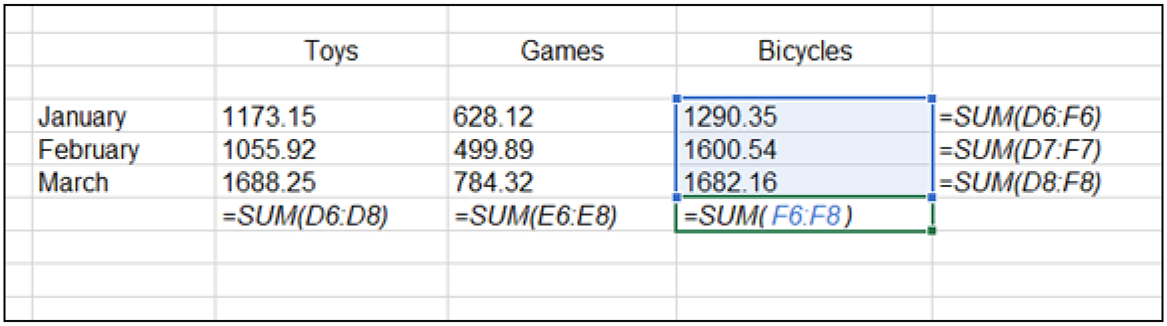


FIGURE 18-1: Setting options.

Notice the Show Formulas in Cells Instead of Their Calculated Results check box. This box tells Excel that for any cells that have formulas to display the formula itself instead of the calculated result. Figure 18-2 shows a worksheet that displays the formulas. To return to normal view, repeat these steps and deselect the option. This option makes it easy to see what all the formulas are!



	Toys	Games	Bicycles	
January	1173.15	628.12	1290.35	=SUM(D6:F6)
February	1055.92	499.89	1600.54	=SUM(D7:F7)
March	1688.25	784.32	1682.16	=SUM(D8:F8)
	=SUM(D6:D8)	=SUM(E6:E8)	=SUM(F6:F8)	

FIGURE 18-2: Viewing formulas the easy way.

WARNING You can accidentally edit functions even when you have selected the Show Formulas option. Be careful clicking around the worksheet.

Fix Formulas

Suppose that your worksheet has some errors. Don't panic! It happens to even the savviest users, and Excel can help you figure out what's going wrong. On the Formulas tab in the Formula Auditing section is the Error Checking button. Clicking the button displays the Error Checking dialog box, shown in Figure 18-3. That is, the dialog box appears if your worksheet has any errors. Otherwise, it just pops up a message that the error check is complete. It's that smart!

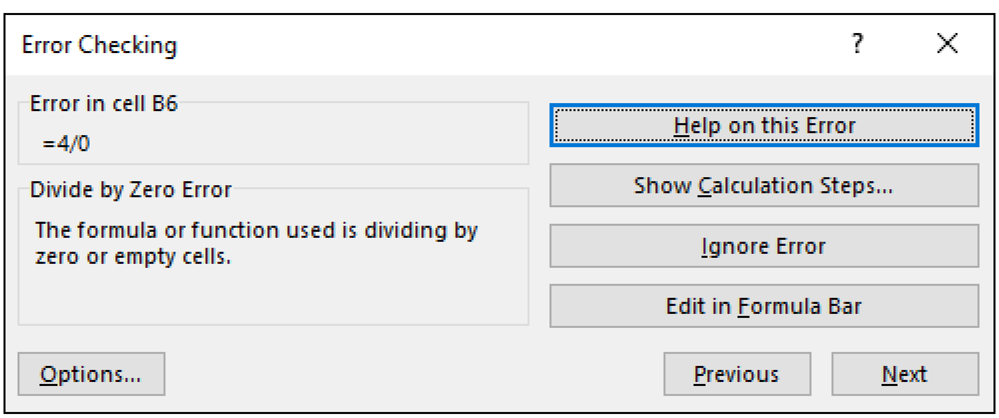


FIGURE 18-3: Checking for errors.

When there *are* errors, the dialog box appears and sticks around while you work on each error. The Next and Previous buttons let you cycle through all the errors before the dialog box closes. For each error it finds, you choose what action to take:

- **Help on This Error:** This leads to the Help system and displays the topic for the particular type of error.
- **Show Calculation Steps:** The Evaluate Formula dialog box opens, and you can watch step by step how the formula is calculated. This lets you identify the particular step that caused the error.
- **Ignore Error:** Maybe Excel is wrong. Ignore the apparent error.
- **Edit in Formula Bar:** This is a quick way to fix the formula yourself if you don't need any other help.

The Error Checking dialog box also has an Options button. Clicking the button opens the Formulas tab of the Excel Options dialog box. On the Formulas tab, you can select settings and rules for how errors are recognized and triggered.

Use Absolute References

If you are going to use the same formula for a bunch of cells, such as those going down a column, the best method is to write the formula once and then drag it down to the other cells by using the fill handle. The problem is that when you drag the formula to new locations, any relative references change.

Often, this *is* the intention. When there is one column of data and an adjacent column of formulas, typically, each cell in the formula column refers to its neighbor in the data column. But if the formulas all reference a cell that is not adjacent, the intention usually is for all the formula cells to reference an unchanging cell reference. Get this to work correctly by using an absolute reference to the cell.

TIP To use an absolute reference to a cell, use the dollar sign ($) before the row number, before the column letter, or before both. Do this when you write the first formula, before dragging it to other cells, or you will have to update all the formulas.

For example, don't write this:

```
=A4 × (B4 + A2)
```

Write it this way instead:

```
=A4 × (B4 + $A$2)
```

This way, all the formulas reference A2 no matter where you copy them, instead of that reference turning into A3, and A4, and so on.

You can cycle through relative, missed, and absolute references. Press F4 to use this shortcut.

Turn Calc On/Turn Calc Off

The Excel default is to calculate your formulas automatically as they are entered or when you change the worksheet. In some situations, you may want to set the calculation to manual. Leaving the setting on automatic is usually not an issue, but if you are working on a hefty workbook with lots of calculations, you may need to rethink this one.

Imagine this: You have a cell that innocently does nothing but display the date. But dozens of calculations throughout the workbook reference that cell. Then dozens more calculations reference the first batch of cells that reference the cell with the data. Get the picture? In a complex workbook, there could be a lot of calculating going on, and the time it takes can be noticeable.

Turning the calculation setting to manual lets you decide when to calculate. Do this in the Excel Options dialog box; click the File tab on the Ribbon and then click Options. In the dialog box, click the Formulas tab, in which calculation options are selected, as shown in Figure 18-4. You can select one of the automatic calculation settings or manual calculation.

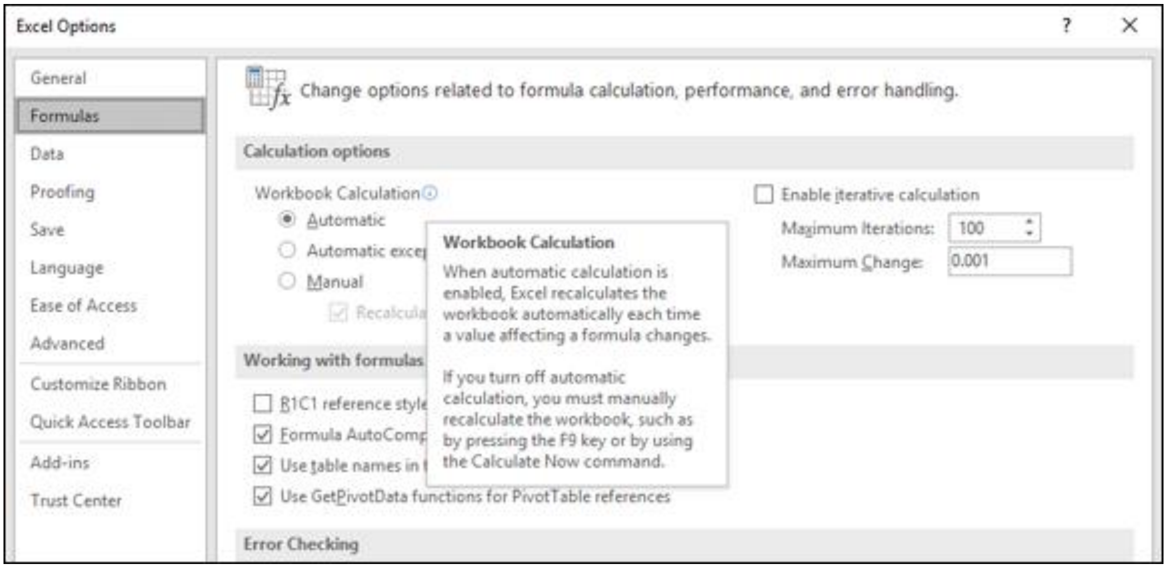


FIGURE 18-4: Setting the calculation method.

Pressing F9 calculates the workbook. Use it when the calculation is set to Manual. Here are some further options:

What You Press	***What You Get***
F9	Calculates formulas that have changed since the last calculation, in all open workbooks.
Shift+F9	Calculates formulas that have changed since the last calculation, just in the active worksheet.

Ctrl+Alt +F9	Calculates all formulas in all open workbooks, regardless of when they were last calculated.
Calculate Now	This is a button in the Calculation group in the Formulas tab. It calculates formulas that have changed since the last calculation, in all open workbooks.

Use Named Areas

Heck, maybe it's just me, but I think it is easier to remember a word such as *Customers* or *Inventory* or *December* than it is to remember B14:E26 or AF220:AR680. So I create names for the ranges that I know I'll reference in my formulas and functions.

Naming areas is easy to do, and in fact, you can do it a few ways. The first is to use the New Name dialog box. You can get to this by clicking the Define Name button on the Formulas tab of the Ribbon. In the dialog box, you set a range, give it a name, enter an optional comment, and then click OK (see Figure 18-5). The comment feature is useful for further notes about the range. For example, the name of the range might be "July Sales," and in the comment area you can enter any further related information.

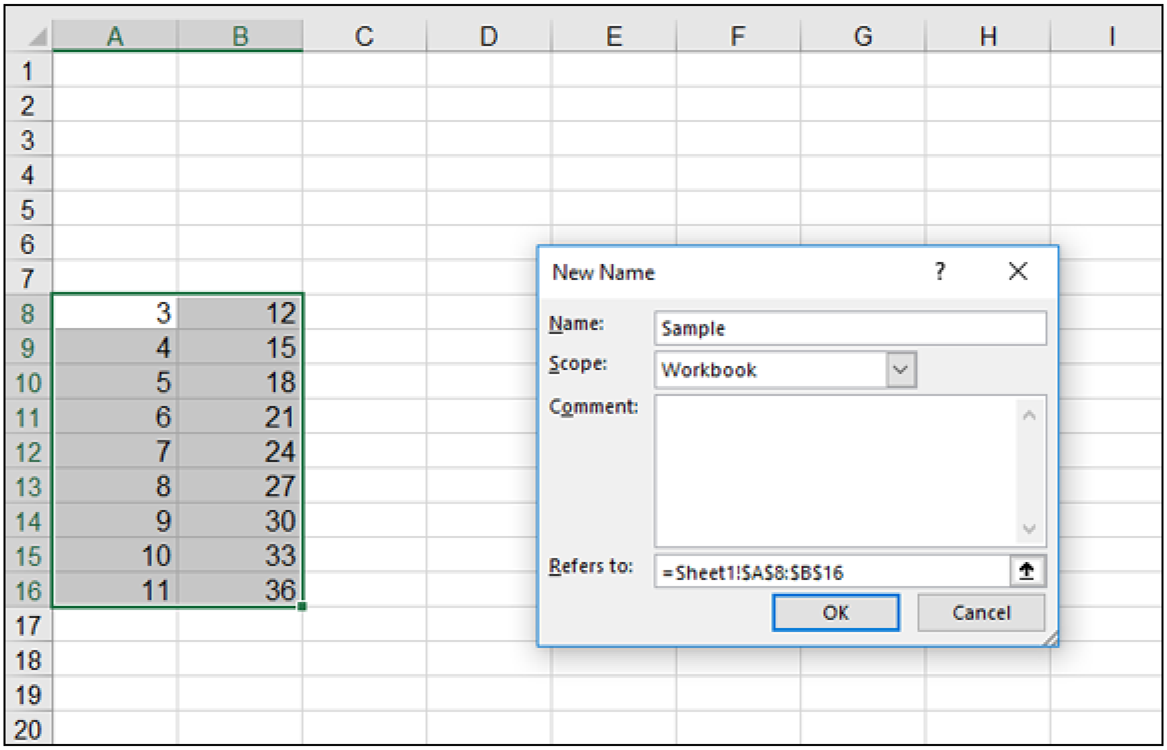


FIGURE 18-5: Defining a named area.

The Name Manager is another dialog box that you can display by clicking its button on the Formulas tab. This dialog box lets you add, update, and delete named areas. A really quick way to just add them (but not update or delete) is to follow these steps:

1. **Select an area on the worksheet.**
2. **Click the Name Box and enter the name.**

   The Name Box is part of the Formula Bar and sits to the left of where formulas are entered.
3. **Press Enter.**

Done! Now you can use the name as you please. Figure 18-6 shows a name being entered in the Name Box. Of course, you can use a particular name only once in a workbook. After the defined name is entered, you can find it in the Name Box by clicking the down arrow in the right of the box.

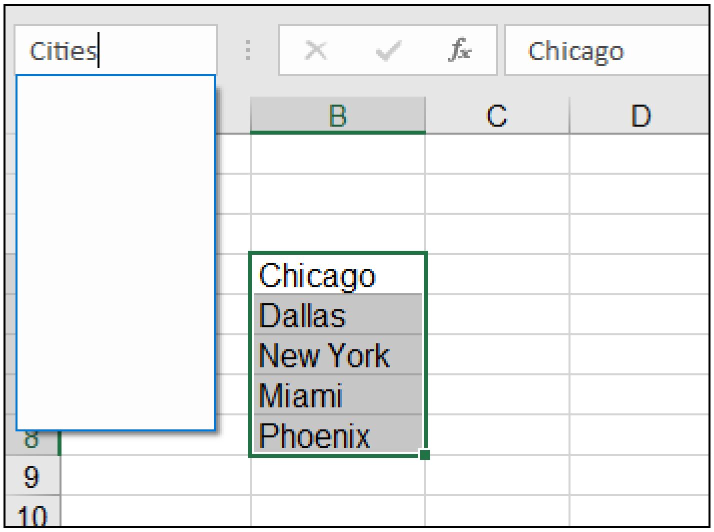


FIGURE 18-6: Defining a named area the easy way.

Use Formula Auditing

There are precedents and dependents. There are external references. There is interaction everywhere. How can you track where the formula references are coming from and going to?

Use the formula auditing tools, that's how! On the Formulas tab is the Formula Auditing section. In the section are various buttons that control the visibility of auditing trace arrows (see Figure 18-7).

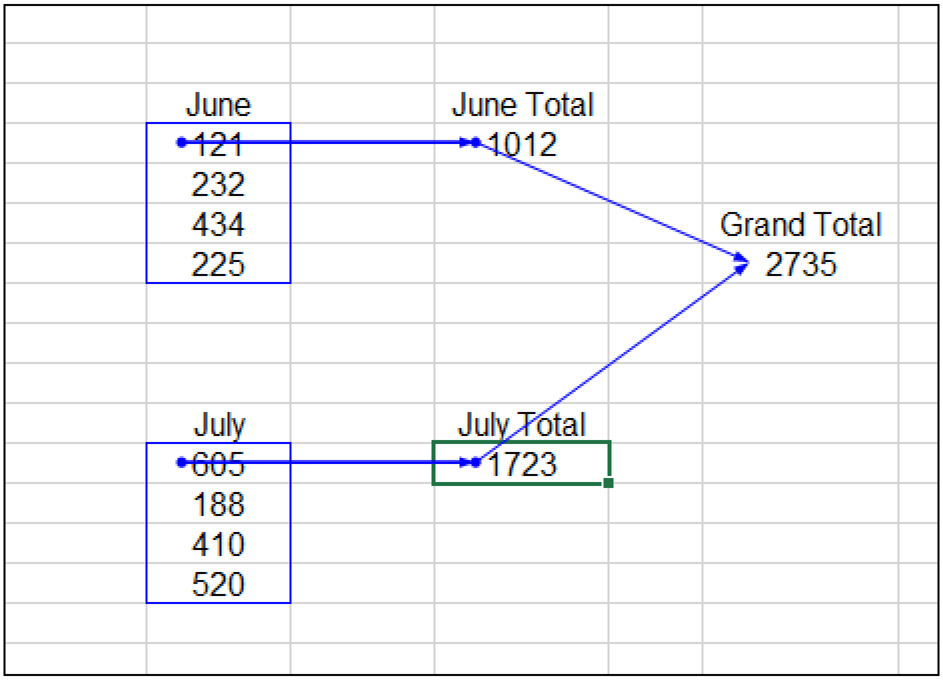


FIGURE 18-7: Auditing formulas.

The formula auditing toolbar has several features that let you wade through your formulas. Besides showing tracing arrows, the toolbar also lets you check errors, evaluate formulas, check for invalid data, and add comments to worksheets.

Use Conditional Formatting

Just as the IF function returns a certain value when the first argument condition is true and another value when it's false, conditional formatting lets you apply a certain format to a cell when a condition is true. On the Home tab in the Styles section is a drop-down menu with many conditional formatting options. Figure 18-8 shows some values that have been treated with conditional formatting. Conditional formatting lets you set the condition and select the format that is applied when the condition is met. For example, you could specify that the cell be displayed in bold italic when the value it contains is greater than 100.

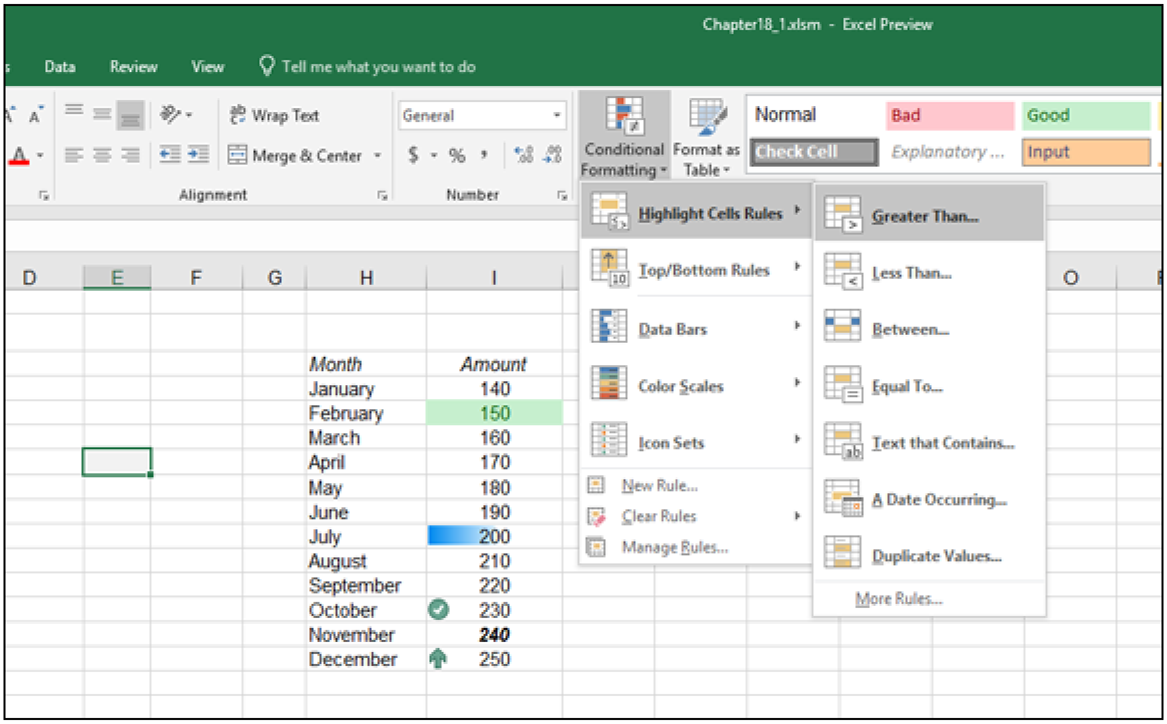


FIGURE 18-8: Applying a format when a condition is met.

Conditions are set as rules. The Rule Types are

- Format all cells based on their values.
- Format only cells that contain… .
- Format only top or bottom ranked values.
- Format only values that are above or below average.
- Format only unique or duplicate values.
- Use a formula to determine which cells to format.

When the condition is true, formatting can control the following:

- Borders
- Number formatting
- Font settings (style, color, bold, italic, and so on)
- Fill (a cell’s background color or pattern)

Cells can also be formatted with color schemes or icon images placed in the cell.

Use Data Validation

On the Data tab, in the Data Tools section, is Data Validation. Data Validation lets you apply a rule to a cell (or a range of cells) that the entry must adhere to. For example, a cell can be set to accept only an integer entry between 50 and 100 (see Figure 18-9).

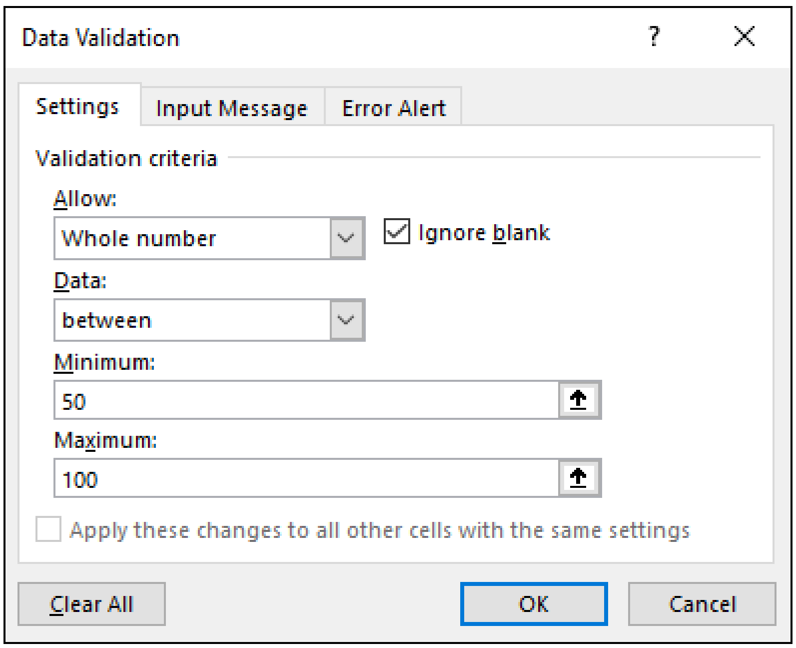


FIGURE 18-9: Setting data validation.

When entry does not pass the rule, a message is displayed (see Figure 18-10).

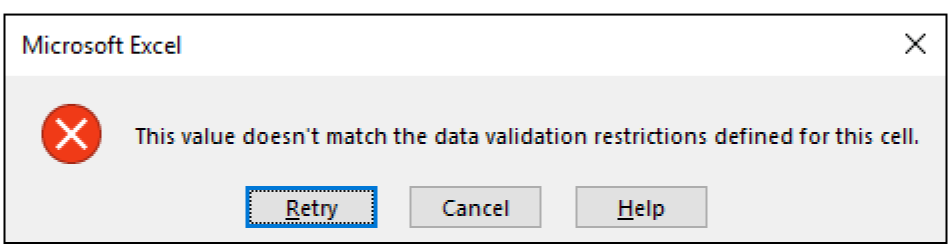


FIGURE 18-10: Caught making a bad entry.

TIP The error message can be customized. For example, if someone enters the wrong number, the displayed error message can say `Noodlehead — learn how to count!` Just don't let the boss see that.

Create Your Own Functions

Despite all the functions provided by Excel, you may need one that you just don’t see offered. Excel lets you create your own functions by using VBA programming code; your functions show up in the Insert Function dialog box.

Okay, I know what you’re thinking: Me, write VBA code? No way! It’s true — this is not for everyone. But nonetheless, here is a short-and-sweet example. If you can conquer this, you may want to find out more about programming VBA. Who knows — maybe one day you’ll be churning out sophisticated functions of your own! Make sure you are working in a macro-enabled workbook (one of the Excel file types).

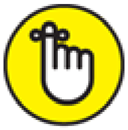
Follow along to create custom functions:

1. **Press Alt + F11.**
   This gets you to the Visual Basic Editor, where VBA is written.
   You can also click the Visual Basic button on the Developer tab of the Ribbon. The Developer tab is visible only if the Developer check box is selected on the Customize Ribbon tab of the Excel Options dialog box.
2. **Choose Insert⇒ Module in the editor.**

   You have an empty code module sitting in front of you. Now it’s time to create your very own function!
3. **Type this programming code, shown in Figure 18-11:**
   ```
   Public Function Add(number1 As Double, number2 As
   Double)
   ```

```
  Add = number1 + number2
End Function
```

4. **Save the function.**



REMEMBER Macros and VBA programming can be saved only in a macro-enabled workbook.

After you type the first line and press Enter, the last one appears automatically. This example function adds two numbers, and the word `Public` lists the function in the Insert Function dialog box. You may have to find the Excel workbook on the Windows taskbar because the Visual Basic Editor runs as a separate program. Or press Alt+ F11 to toggle back to the Workbook.

5. **Return to Excel.**
6. **Click the Insert Function button on the Formulas tab to display the Insert Function dialog box (see Figure 18-12).**
7. **Click OK.**

The Function Arguments dialog box opens, ready to receive the arguments (see Figure 18-13). Isn't this incredible? It's as though you are creating an extension to Excel, and in essence, you are.

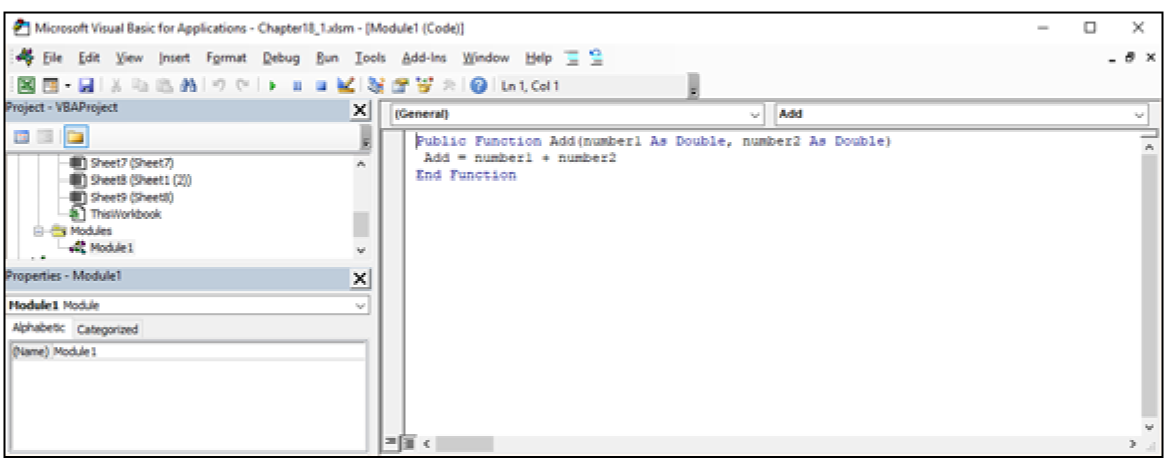


FIGURE 18-11: Writing your own function.

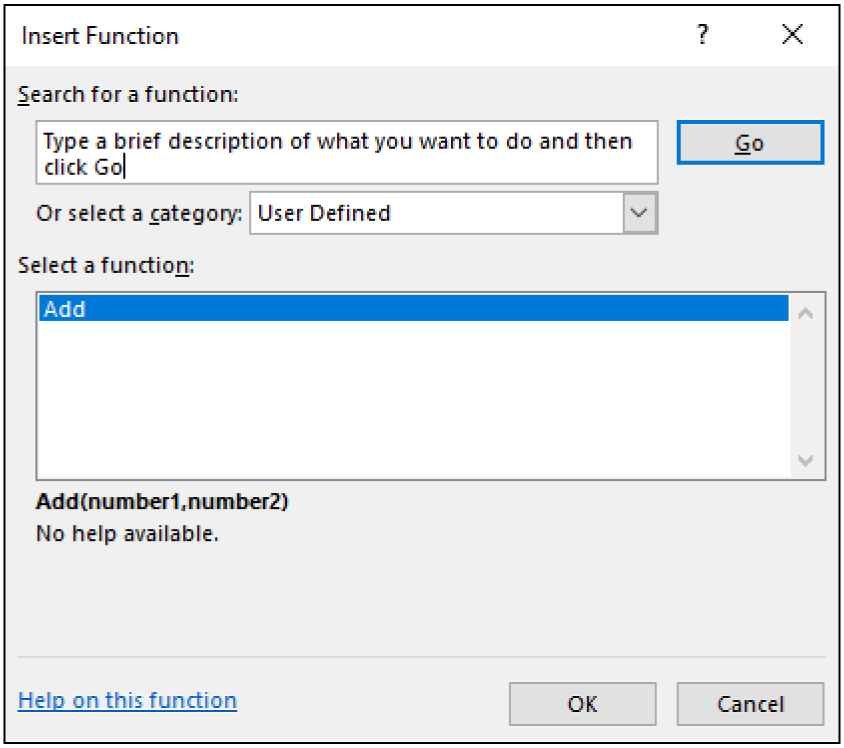


FIGURE 18-12: Finding the function in the User Defined category.

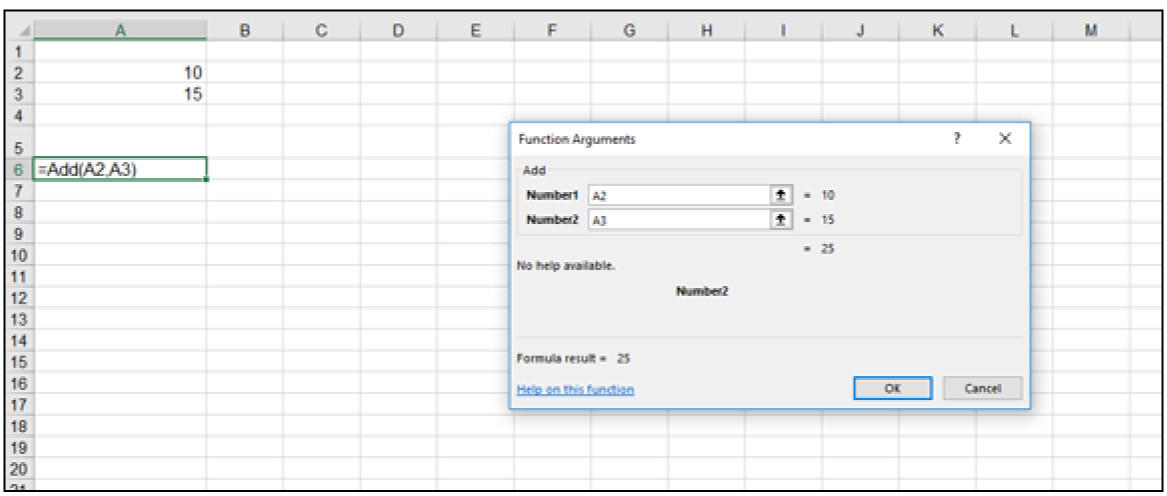


FIGURE 18-13: Using the custom Add function.

This is a very basic example of what you can do by writing your own function. The possibilities are endless, but of course, you need to know how to program VBA. I suggest reading *Excel VBA Programming For Dummies*, 5th Edition, by Michael Alexander (John Wiley & Sons, Inc.).

TIP Macro-enabled workbooks have the file extension `.xlsm`.

Ten Functions You Really Should Know

SUM

Adding numbers is one of the most basic mathematical operations, and so there is the SUM function, dedicated to doing just that. SUM takes up to 255 arguments.

Each argument can be a single number or a range containing multiple numbers. That means SUM can add up a whole bunch of numbers! The syntax follows:

```
=SUM(number 1, number 2, …)
```

You can also use SUM with a range, as shown here:

```
=SUM(A1:A12)
```

You can also use SUM with more than one range, such as this:

```
=SUM(A1:A12, B1:B12)
```

AVERAGE

Although technically a statistical function, AVERAGE is used so often that it deserves a place in the top ten functions. Everyone is interested in averages. What's the average score? What's the average salary? What's the average height? What's the average number of hours we watch TV? (That's a sore spot in my home!)

AVERAGE can take up to 255 arguments. Each argument can be a number or a range that contains numbers. The syntax follows:

```
=AVERAGE(number 1 ,number 2 ,…)
```

You can also use AVERAGE with a range, as shown here:

```
=AVERAGE(A1:A12)
```

You can also use AVERAGE with more than one range, such as this:

```
=AVERAGE(A1:A12, B1:B12)
```

COUNT

COUNT counts the number of cells in a range that contain numbers. It does not provide any sum — just the count. For a list with ten numbers, for example, COUNT returns 10, regardless of what the numbers are.

COUNT takes up to 255 arguments, which can be cell references, range references, or numbers themselves. COUNT ignores non-numeric values. If an argument to COUNT is the ten-cell range A1:A10 but only two cells contain a number, COUNT returns 2. The syntax follows:

```
=COUNT(cell reference 1, cell reference 2,…)
```

You can also use COUNT with a range, as shown here:

```
=COUNT(A1:A12)
```

You can also use COUNT with more than one range, such as this:

```
=COUNT(A1:A12, B1:B12)
```

INT and ROUND

The INT and ROUND functions both work by removing or reducing a number's decimal portion. They differ in exactly *how* they remove it.

INT

INT simply drops the decimal portion without rounding — that is, without regard to whether the number is closer to the next higher integer or the next lower integer. Be aware that INT always truncates to the next lower integer. For example, INT changes 12.05 to 12, but it also changes 12.95 to 12. Also, INT changes both –5.1 and –5.9 to –6, not to –5, because –6 is the next lower integer. INT takes a single number argument (as an actual number or a cell reference). The syntax follows:

```
=INT(number or cell reference)
```

ROUND

On the other hand, the ROUND function lets you control how the decimal portion is handled. ROUND takes two arguments: the number to be manipulated and the number of decimal places to round to. This gives you more control. A number such as 5.6284 can become 5.628, 5.63, 5.6, or just 6. ROUND always rounds up or down to the nearest number of the next significant digit, so 5.628 becomes 5.63, not 5.62.

ROUND turns 12.95 into either 12.9 or 13, depending on the setting of the second argument. Note that two functions — ROUNDUP and ROUNDDOWN — round in one direction only. The syntax for ROUND follows:

```
=ROUND(number, number of decimal places to round
to)
```

The syntax for ROUNDUP and ROUNDDOWN is the same as ROUND:

```
=ROUNDUP(number, number of decimal places to round to)
=ROUNDDOWN(number, number of decimal places to round to)
```

IF

IF is a very handy function. It tests a condition and returns one of two results, depending on the outcome of the test. The test must return a true or false answer. For example, a test may be `B25 > C30`. If true, IF returns its second argument. If false, IF returns its third argument.

IF is often used as a validation step to prevent unwanted errors. The most common use of this is to test whether a denominator is 0 before doing a division operation. By testing for 0 first, you can avoid the `#DIV/0!` error.

One of the great things about IF is that the result can be a blank. This function is great when you want to return a result if the test comes out one way but not if the result is otherwise. The syntax follows:

```
=IF(logical test, value if true, value if false)
```

NOW and TODAY

The NOW function returns the current date and time according to your computer's internal clock. TODAY returns just the date. If the date or time is wrong, it can't help you with that.

A common use of NOW is to return the date and time for a printed report. You know, so a message such as `Printed on 12/20/2019 0:15` can be put on the printed paper.

A common use for TODAY is to calculate the elapsed time between a past date and today. For example, you may be tracking a project's duration. A cell on the worksheet has the start date. Another cell has a formula that subtracts that date from TODAY. The answer is the number of days that have gone by.

NOW and TODAY take no arguments. The syntax for each follows:

```
=NOW()
=TODAY()
```

HLOOKUP and VLOOKUP

HLOOKUP and VLOOKUP both find a value in a table. A *table* is an area of rows and columns that you define. Both of these functions work by using a search value for the first argument that, when found in the table, helps return a different value.

In particular, you use HLOOKUP to return a value in a row that is in the same column as the search value. You use VLOOKUP to return a value in a column that is in the same row as the search value. The syntax for these functions follows:

```
=HLOOKUP(lookup value, table area, row, match type)
=VLOOKUP(lookup value, table area, column, match
type)
```

ISNUMBER

A rose is a rose and by any other name would smell as sweet, but numbers don't get off that easy. For example, 15 is a digit, but *fifteen* is a word. The ISNUMBER function tells you, flat-out true or false, if a value in a cell is a number (including the results of formulas). The syntax follows:

```
=ISNUMBER(value)
```

MIN and MAX

MIN and MAX find the respective lowest or highest numeric value in a range of values. These functions take up to 255 arguments, and an argument can be a range. Therefore, you can test a large list of numbers simply by entering the list as a range. The syntax for these functions follows:

```
=MAX(number1,number2,…)
=MIN(number1,number2,…)
```

You can also use MIN and MAX with a range, as shown here:

```
=MAX(A1:A12)
```

or with more than one range, such as this:

```
=MAX(A1:A12, B1:B12)
```

SUMIF and COUNTIF

SUMIF and COUNTIF sum or count values, respectively, if a supplied criterion is met. This makes for some robust calculations. With these functions, it's easy to return answers for a question such as "How many shipments went out in October?" or "How many times did the Dow Jones Industrial Average close over 18,000 last year?"

SUMIF takes three arguments:

- A range in which to apply the criteria
- The actual criteria
- The range from which to sum values

A key point here is that the first argument may or may not be the same range from which values are summed. Therefore, you can use SUMIF to answer a question such as "How many shipments went out in October?" but also one such as “What is the sum of the numbers over 100 in this list?” The syntax of SUMIF follows:

```
=SUMIF(range,criteria,sum_range)
```

Note, too, that the third argument in SUMIF can be left out. When this happens, SUMIF uses the first argument as the range in which to apply the criteria and also as the range from which to sum.

COUNTIF counts the number of items in a range that match criteria. This is just a count. The value of the items that match the criteria doesn’t matter past the fact that it matches the criteria. But after a cell’s value matches the criteria, the count of that cell is 1. COUNTIF takes just two arguments:

- The range from which to count the number of values
- The criteria to apply

The syntax for COUNTIF follows:

```
COUNTIF(range,criteria)
```

- Copy
- Add Highlight
- Add Note