EXCEL FUNCTIONS

Functions are pre-programmed tools that will manipulate data for you.

=SUM()

The first function anyone should know tells Excel to carry out basic arithmetic. Let's say you wanted the numbers in cell A2 and cell B2 to be added together in cell B3. All you'd do is enter =SUM into B3, click the =SUM entry in the pop-up that appears, *Control + Click* in cell A2 and B2 to highlight both, then hit *Enter.* In an instant you'd see sum of the two cells appear.

You can use this function, and most of the functions listed here, to encompass as many cells as you require—just select all the ones you need included.

=AVERAGE

Average does exactly what it says and works similarly to SUM. If you have a worksheet containing your monthly earnings over the past year in columns A2 through A13, type =AVERAGE(A2:A13) into A14 to get the average.

Again, you can also use your mouse cursor to click-and-drag highlight a range of cells, or use *Control + Click* to handpick individual cells.

=MIN()

If you need to find the smallest number in a range of cells, MIN can help you do that. Using =MIN(B3:B39) will give you the smallest number contained in those cells.

=MAX()

Max is the counterpart to MIN and does the opposite showing you the largest number contained in a range of cells.

=TRIM()

If you copy text from another program into Excel you can often end up with excess whitespace that turns your spreadsheet into a visual horror. TRIM can help you clean it up.

TRIM can only deal with text from a single cell. So start by cleaning up the text in cell B1 by typing =TRIM(B1) into cell C1, for example, and then repeat the function for any other cells you need tidied up.

If you're looking to clean up line breaks, try CLEAN instead of TRIM.

=COUNT()

If you need to know how many cells in a given range contain numbers, don't bother counting by hand—just use the COUNT function. If you have a mix of numbers and text in cells A1-A20, for example, type into cell A21 =COUNT(A1:20) and you'll have the answer in no time.

=COUNTA()

Similar to the above example, you can use =COUNTA() to count the number of cells in a given range that contain characters such as numbers, text, or symbols.

=LEN()

If you want to count the number of characters in a single cell, including white spaces, check out LEN. Want to know how many characters are in cell A1? Just type =LEN(A1) into a different cell and you'll find out.

=CONCATENATE()

This takes data from two cells and turns it into one. Check out our previous look at Excel functions to see how this can be useful.

=DAYS()

Want to know the number of days between two dates in a spreadsheet? If you had for example, September 5, 2015 in cell A4 and December 27, 2015 in A5 just use =DAYS(A5, A4) to get the answer. (Reversing the order of cells would give you a positive number).

=NETWORKDAYS()

Knowing the number of days is great, but if you need to know how many work days that range encompasses, turn to NETWORKDAYS (that's 'net work days') instead. This function uses the same format as DAYS, but you need to use an ascending cell order to get a non-negative number. So =NETWORKDAYS(A4, A5) will give you 80 and not -80.

=SQRT()

Need to know the square root of 1764? Type =SQRT(1764) into a cell, hit *Enter*, and you'll find the answer to life, the universe, and everything—including the example formula.

=NOW()

Want to see the current date and time whenever you open a particular worksheet? Type =NOW() into the cell where you want the date and you're done. If you want to see a future date from right now, you can use something like =NOW()+32. The now function does not take any arguments so don't put anything in between the brackets.

=ROUND()

As its name suggests, this function lets you round off numbers. ROUND requires two arguments: a number or cell, and the number of digits to round to. If you have the number 231.852645 in A1, for example, =ROUND(A1, 0) gives you 232, =ROUND(A1, 1) gives you 232.9, and =ROUND(A1, -1) returns 230.

=ROUNDUP(), =ROUNDDOWN()

If you want more direct control over rounding up or down there are functions for that too. ROUNDUP and ROUNDDOWN use exactly the same argument format as ROUND.

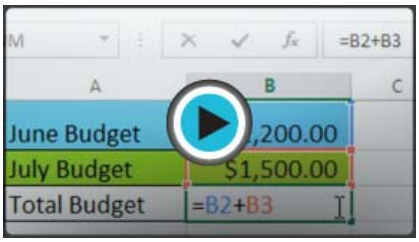
Excel 2013

Simple Formulas

Introduction

One of the most powerful features in Excel is the ability to **calculate** numerical information using **formulas**. Just like a calculator, Excel can add, subtract, multiply, and divide. In this lesson, we'll show you how to use **cell references** to create simple formulas.

Video: Simple Formulas



Optional: Download our **practice workbook**.

Mathematical operators

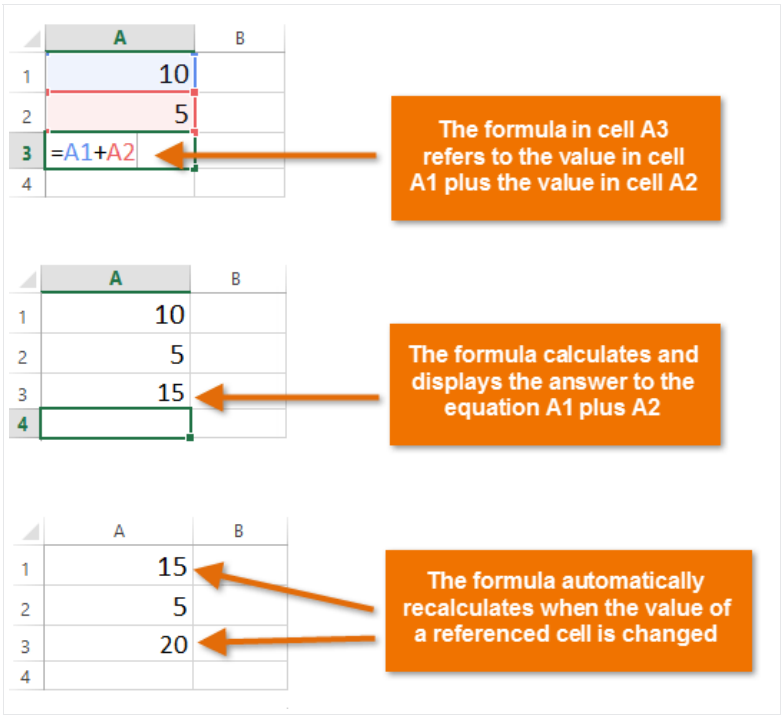
Excel uses standard operators for formulas, such as a **plus sign** for addition (**+**), a **minus sign** for subtraction (**-**), an **asterisk** for multiplication (*****), a **forward slash** for division (**/**), and a **caret** (**^**) for exponents.

Addition	+
Subtraction	-
Multiplication	*
Division	/
Exponents	^

All formulas in Excel must begin with an **equals sign** (**=**). This is because the cell contains, or is equal to, the formula and the value it calculates.

Understanding cell references

While you can create simple formulas in Excel manually (for example, **=2+2** or **=5*5**), most of the time you will use **cell addresses** to create a formula. This is known as making a **cell reference**. Using cell references will ensure that your formulas are always accurate because you can change the value of referenced cells without having to rewrite the formula.



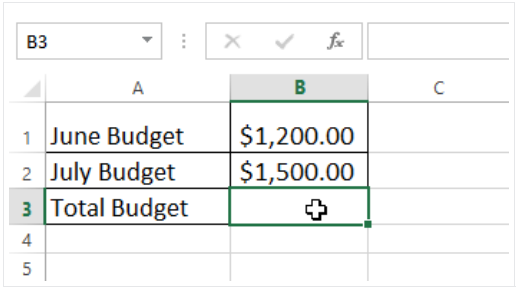
By combining a mathematical operator with cell references, you can create a variety of simple formulas in Excel. Formulas can also include a combination of cell references and numbers, as in the examples below:

=A1+A2	Adds cells A1 and A2
=C4-3	Subtracts 3 from cell C4
=E7/J4	Divides cell E7 by J4
=N10*1.05	Multiplies cell N10 by 1.05
=R5^2	Finds the square of cell R5

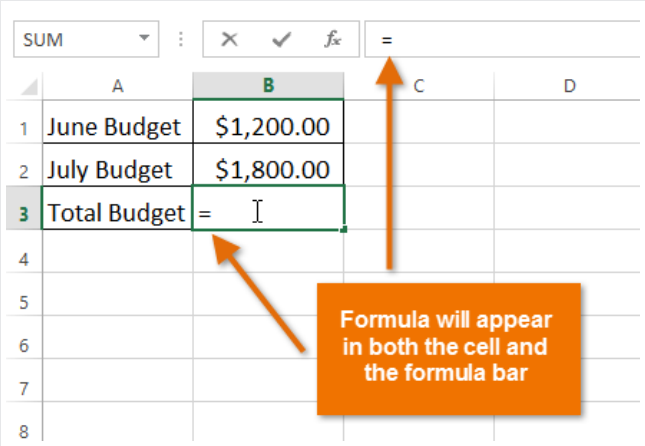
To create a formula:

In our example below, we'll use a simple formula and cell references to calculate a budget.

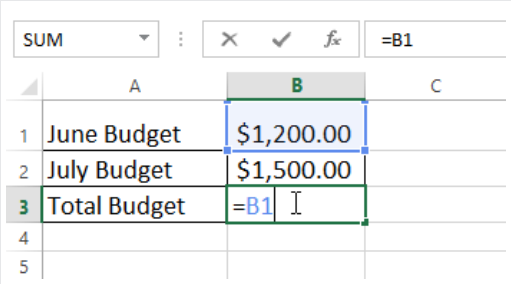
1. Select the **cell** that will contain the formula. In our example, we'll select cell **B3**.



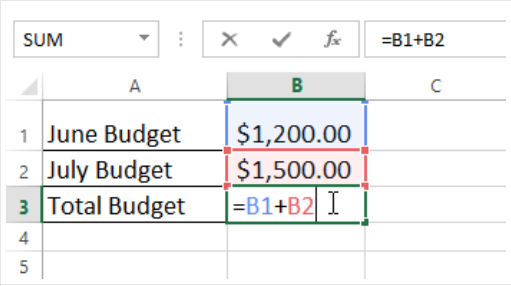
2. Type the **equals sign (=)**. Notice how it appears in both the **cell** and the **formula bar**.



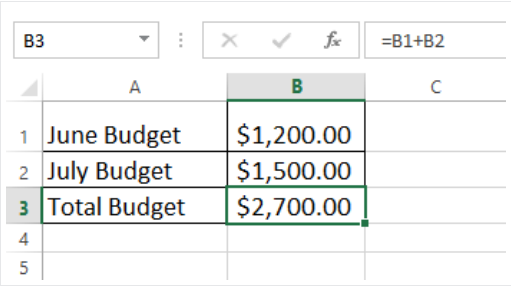
3. Type the **cell address** of the cell you want to reference first in the formula: cell **B1** in our example. A **blue border** will appear around the referenced cell.



4. Type the **mathematical operator** you want to use. In our example, we'll type the **addition sign** (**+**).
5. Type the **cell address** of the cell you want to reference second in the formula: cell **B2** in our example. A **red border** will appear around the referenced cell.



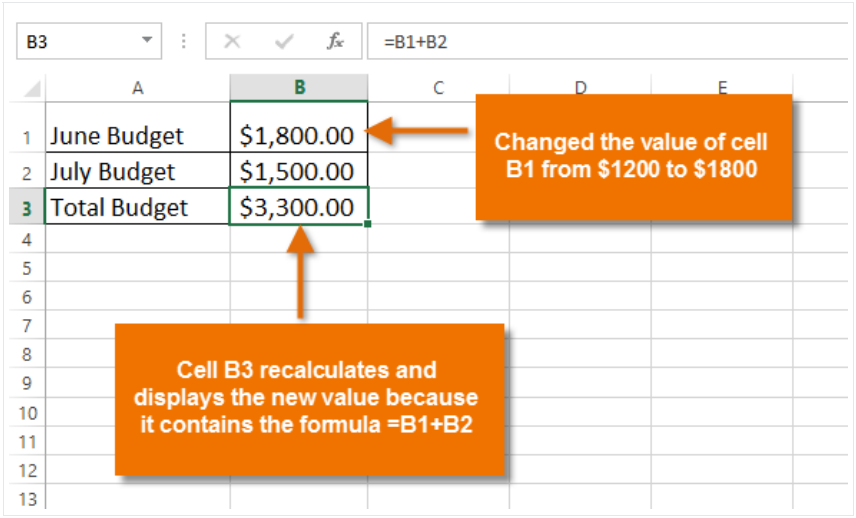
6. Press **Enter** on your keyboard. The formula will be **calculated**, and the **value** will be displayed in the cell.



If the result of a formula is too large to be displayed in a cell, it may appear as **pound signs** (#######) instead of a value. This means the column is not wide enough to display the cell content. Simply **increase the column width** to show the cell content.

Modifying values with cell references

The true advantage of cell references is that they allow you to **update data** in your worksheet without having to rewrite formulas. In the example below, we've modified the value of cell B1 from $1,200 to $1,800. The formula in B3 will automatically recalculate and display the new value in cell B3.

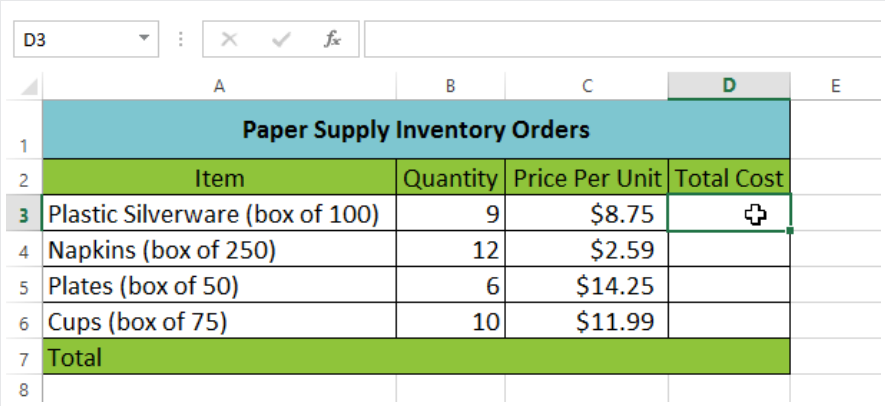


Excel **will not always tell you** if your formula contains an error, so it's up to you to check all of your formulas. To learn how to do this, you can read the **Double-Check Your Formulas** lesson from our **Excel Formulas** tutorial.

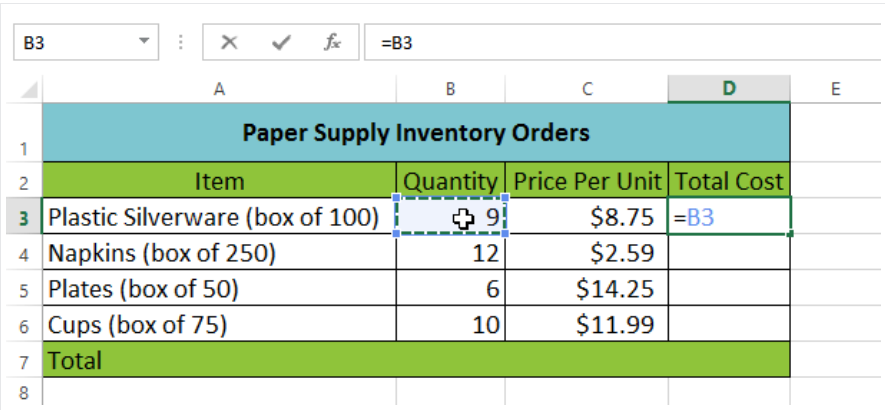
To create a formula using the point-and-click method:

Rather than typing cell addresses manually, you can **point and click** on the cells you want to include in your formula. This method can save a lot of time and effort when creating formulas. In our example below, we'll create a formula to calculate the cost of ordering several boxes of plastic silverware.

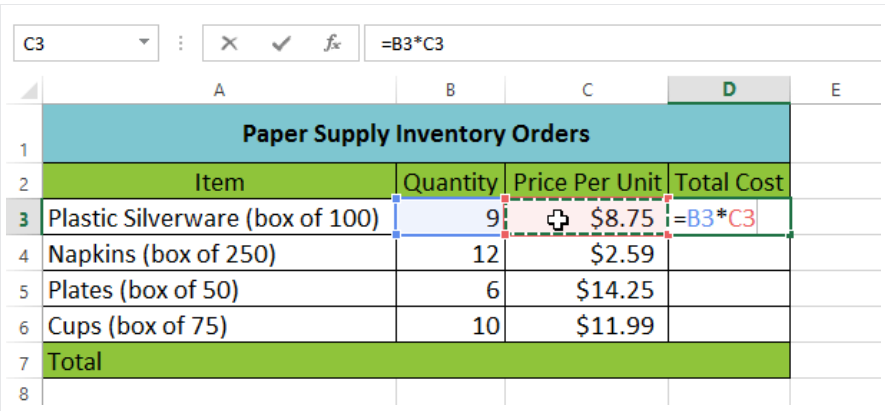
1. Select the **cell** that will contain the formula. In our example, we'll select cell **D3**.



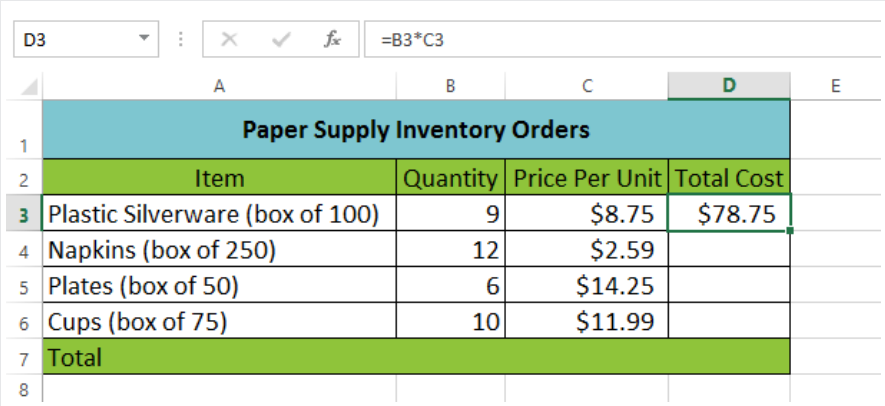
2. Type the **equals sign (=)**.
3. Select the **cell** you want to reference first in the formula: cell **B3** in our example. The **cell address** will appear in the formula, and a **dashed blue line** will appear around the referenced cell.



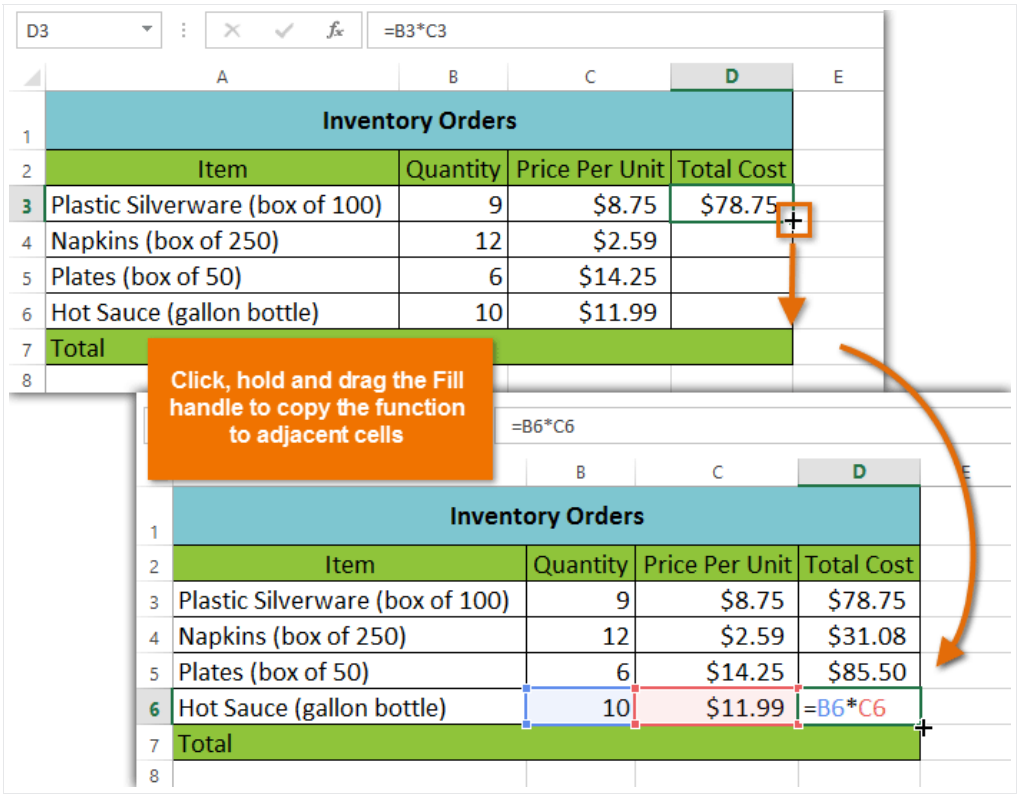
4. Type the **mathematical operator** you want to use. In our example, we'll type the **multiplication sign (*)**.
5. Select the **cell** you want to reference second in the formula: cell **C3** in our example. The **cell address** will appear in the formula, and a **dashed red line** will appear around the referenced cell.



6. Press **Enter** on your keyboard. The formula will be **calculated**, and the **value** will be displayed in the cell.



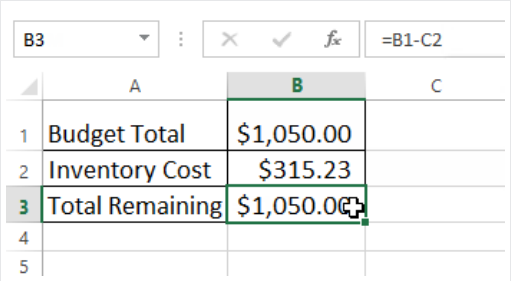
Formulas can also be **copied** to adjacent cells with the **fill handle**, which can save a lot of time and effort if you need to perform the **same calculation** multiple times in a worksheet. Review our lesson on **Relative and Absolute Cell References** to learn more.



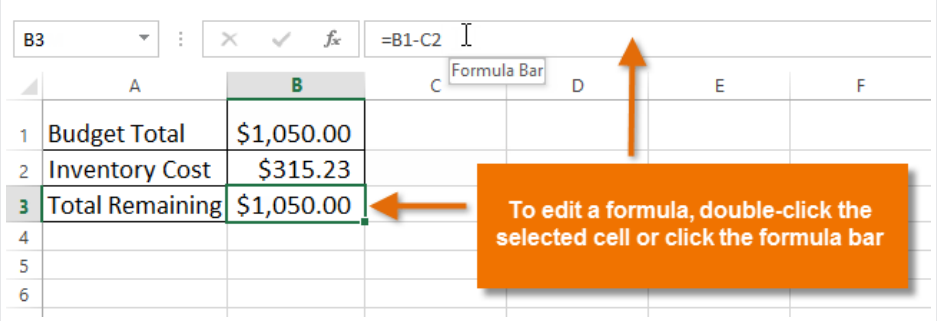
To edit a formula:

Sometimes you may want to modify an existing formula. In the example below, we've entered an incorrect cell address in our formula, so we'll need to correct it.

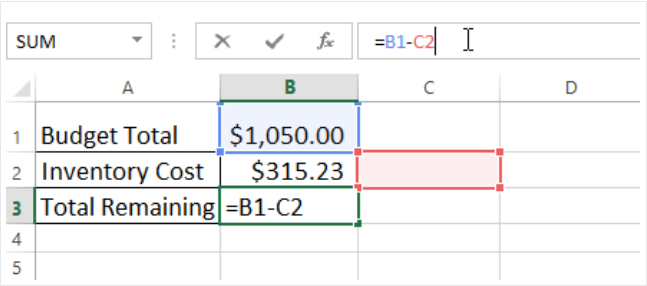
1. Select the **cell** containing the formula you want to edit. In our example, we'll select cell **B3**.



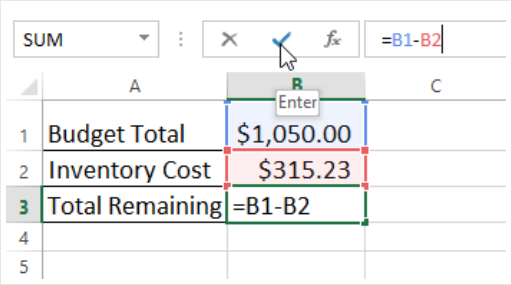
2. Click the **formula bar** to edit the formula. You can also **double-click** the cell to view and edit the formula directly within the cell.



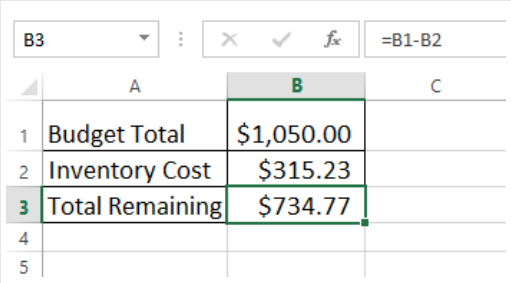
3. A **border** will appear around any referenced cells. In our example, we'll change the second part of the formula to reference cell **B2** instead of cell **C2**.



4. When you're finished, press **Enter** on your keyboard or click the **checkmark** in the formula bar.



5. The formula will be **updated**, and the **new value** will be displayed in the cell.



If you change your mind, you can press the **Esc** key on your keyboard to avoid accidentally making changes to your formula.

To show all of the formulas in a spreadsheet, you can hold the **Ctrl** key and press ` (grave accent). The grave accent key is usually located in the top-left corner of the keyboard. You can press **Ctrl+`** again to switch back to the normal view.

Challenge!

1. Open an existing Excel workbook. If you want, you can use our **practice workbook**.
2. Create a simple addition formula using **cell references**. If you are using the example, create the formula in cell **B4** to calculate the total budget.
3. Try modifying the **value** of a cell referenced in a formula. If you are using the example, change the value of cell **B2** to $2,000. Notice how the formula in cell B4 recalculates the total.
4. Try using the **point-and-click method** to create a formula. If you are using the example, create a formula in cell **G5** that multiplies the cost of **napkins** by the **quantity** needed to calculate the **total cost**.
5. **Edit** a formula using the formula bar. If you are using the example, edit the formula in cell **B9** to change the **division sign** (**/**) to a **minus sign** (**-**).

Excel 2013

Complex Formulas

Introduction

A simple formula is a mathematical expression with one operator, such as **7+9**. A **complex formula** has more than one mathematical operator, such as **5+2*8**. When there is more than one operation in a formula, the **order of operations** tells Excel which operation to calculate first. In order to use Excel to calculate complex formulas, you will need to understand the order of operations.

Video: Complex Formulas

Optional: Download our **practice workbook**.

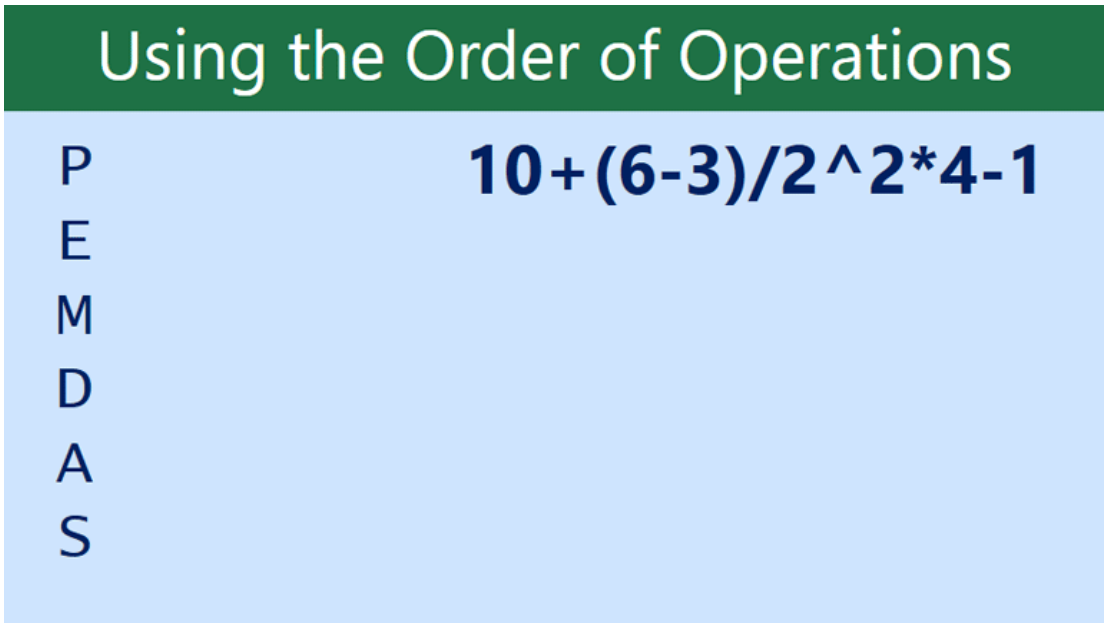
The order of operations

Excel calculates formulas based on the following **order of operations**:

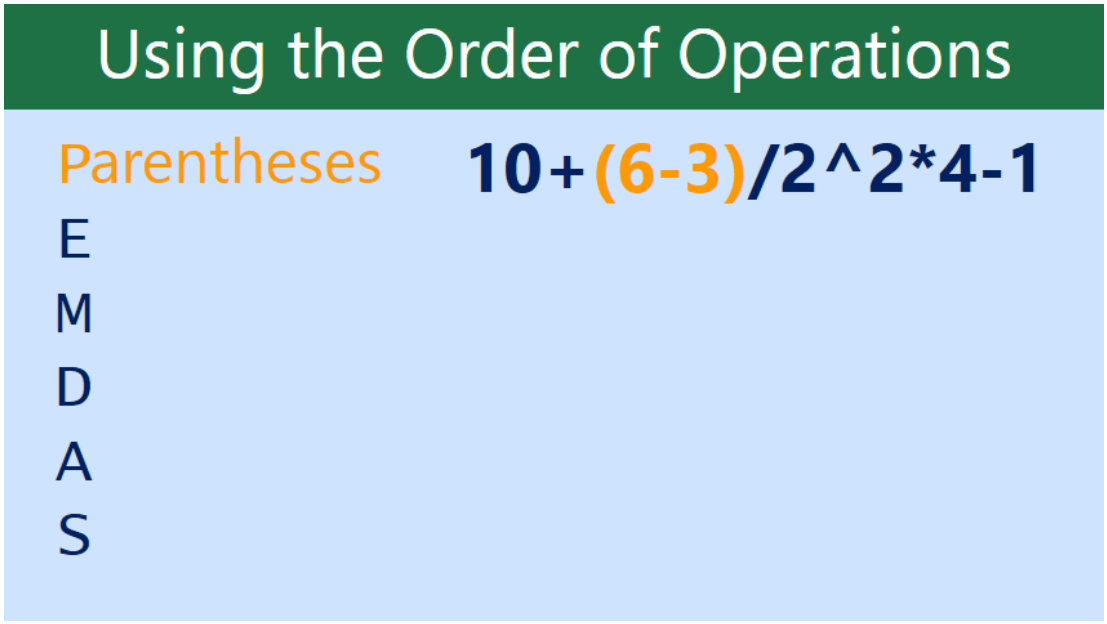
1. Operations enclosed in **parentheses**
2. **Exponential** calculations (3^2, for example)
3. **Multiplication** and **division**, whichever comes first
4. **Addition** and **subtraction**, whichever comes first

A mnemonic that can help you remember the order is **PEMDAS**, or **P**lease **E**xcuse **M**y **D**ear **A**unt **S**ally.

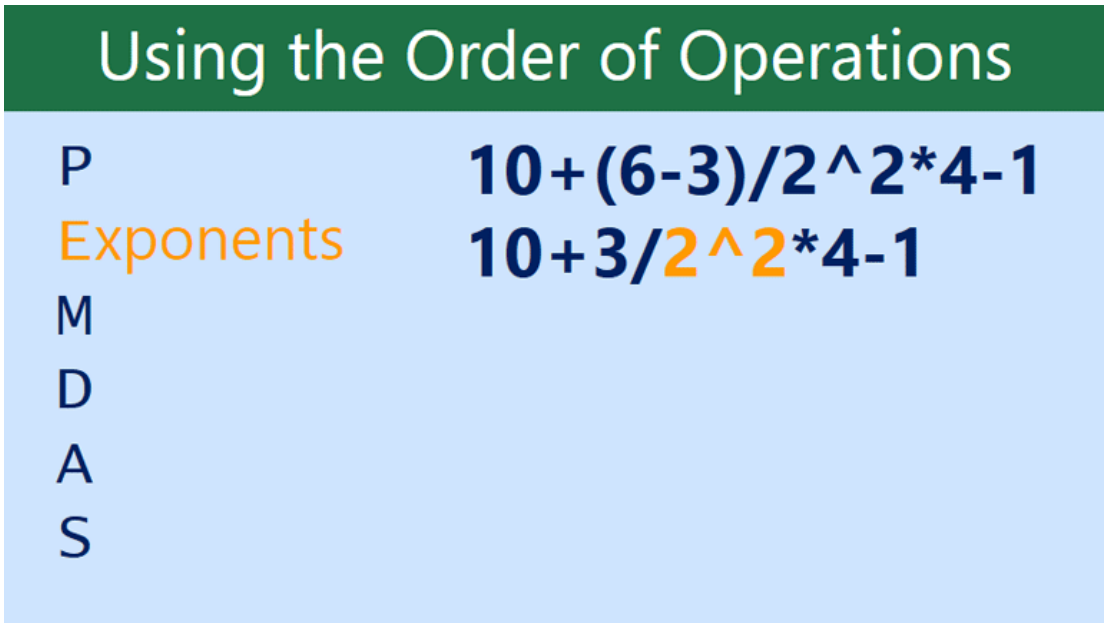
Click the arrows in the slideshow below to learn more about how the order of operations is used to calculate formulas in Excel.

- 

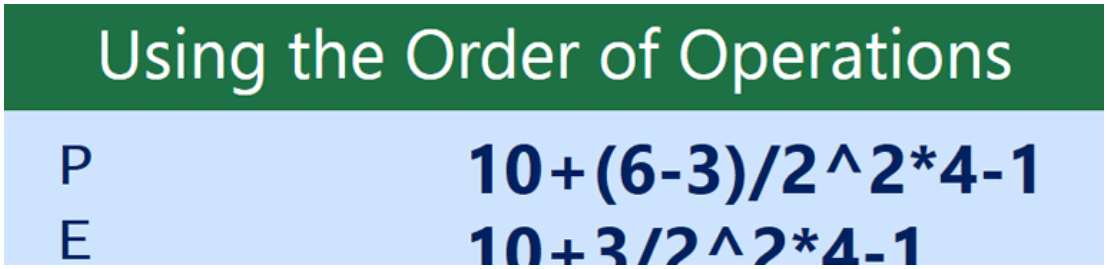
While this formula may look complicated, we can use the order of operations step by step to find the right answer.

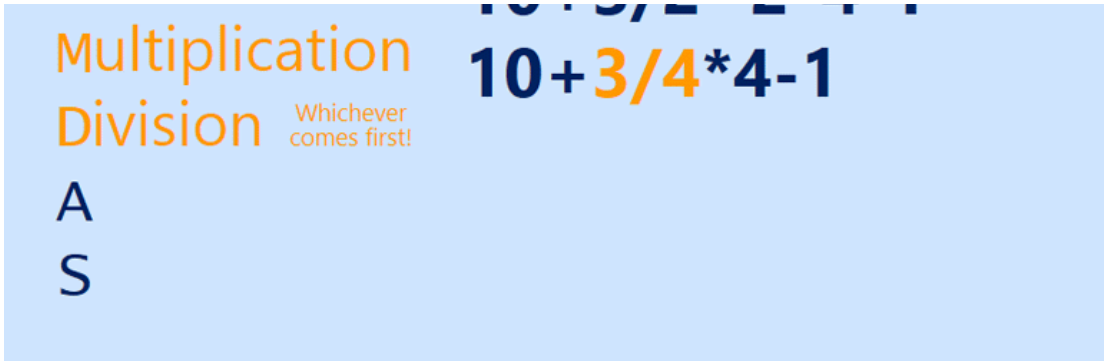


- First, we'll start by calculating anything inside the parentheses. In this case, there's only one thing we need to calculate: 6-3=3.

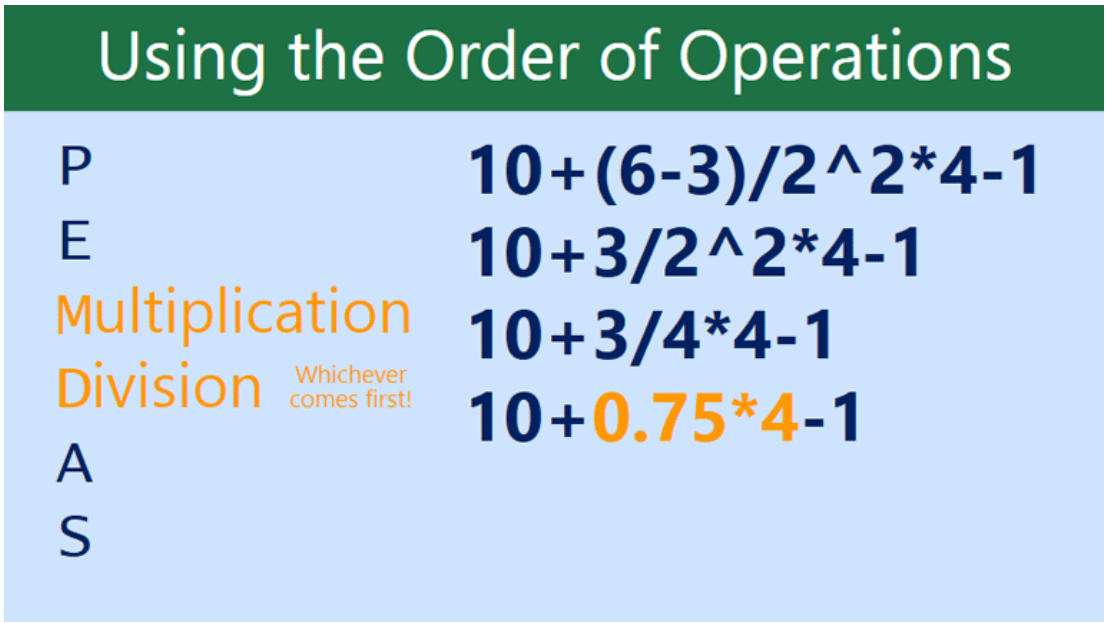


- As you can see, the formula already looks simpler. Next, we'll look to see if there are any exponents. There's one: 2^2=4.

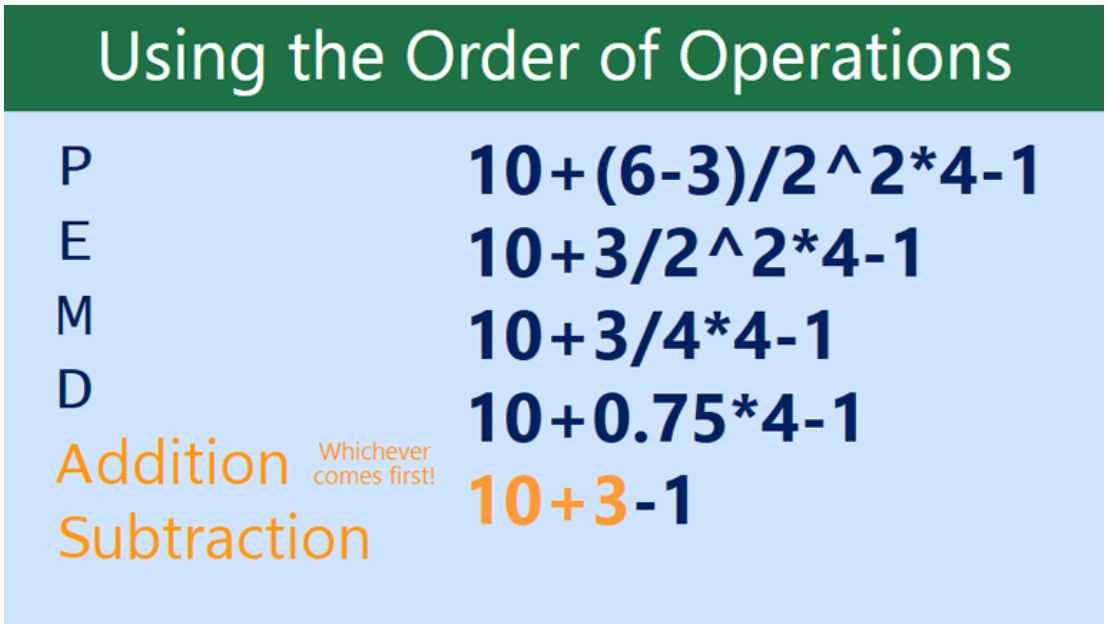




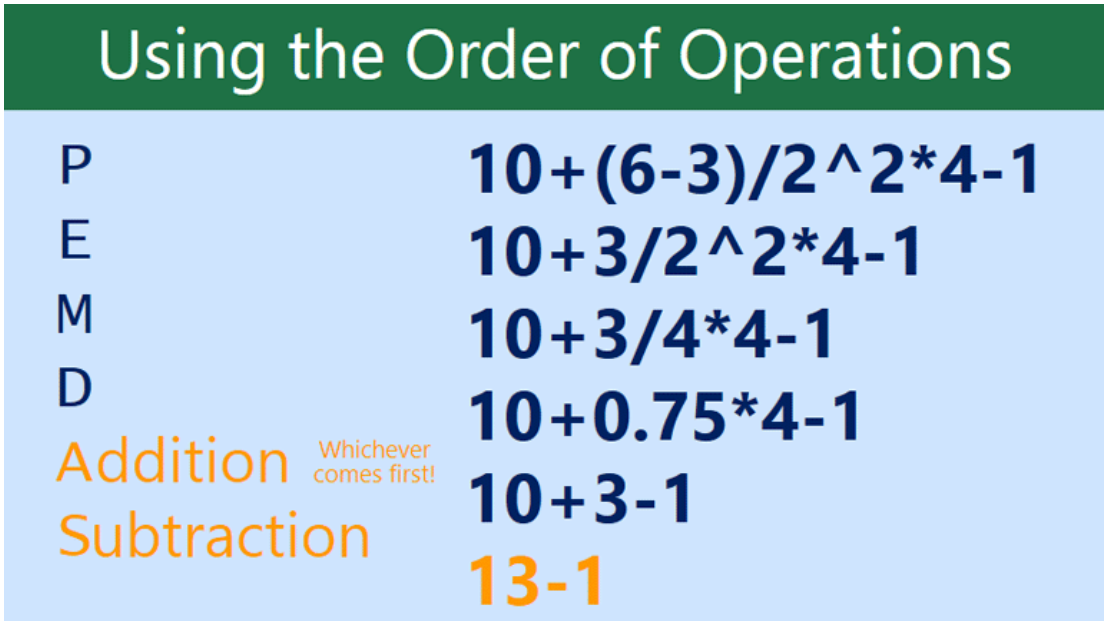
Next, we'll solve any multiplication and division, working from left to right. Because the division operation comes before the multiplication, it is calculated first: 3/4=0.75.



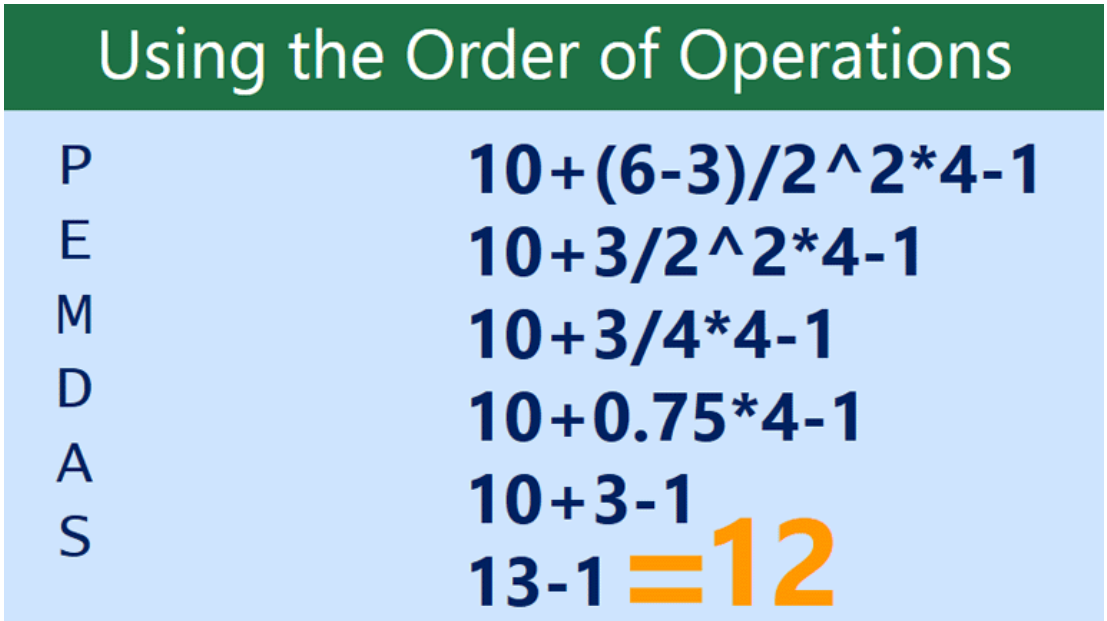
- Now, we'll solve our remaining multiplication operation: 0.75*4=3.



Next, we'll calculate any addition or subtraction, again working from left to right. Addition comes first: 10+3=13.



- Finally, we have one remaining subtraction operation: 13-1=12.

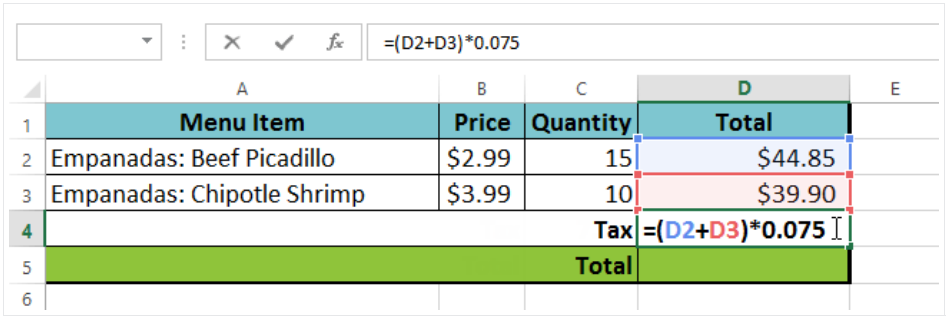


- Now we have our answer: 12. This is the exact same result you would get if you entered the formula into Excel.

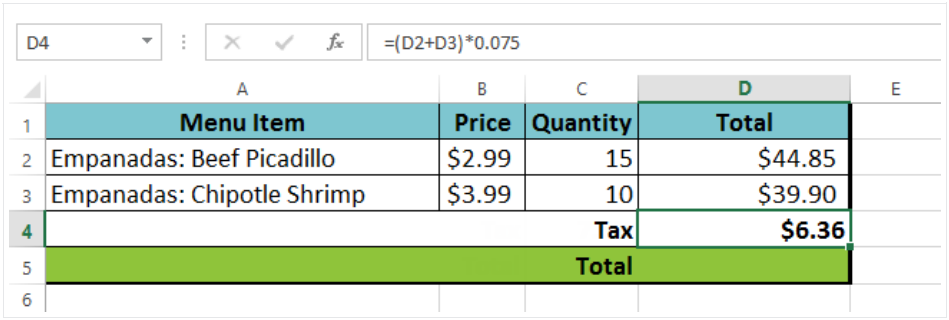
-

Creating complex formulas

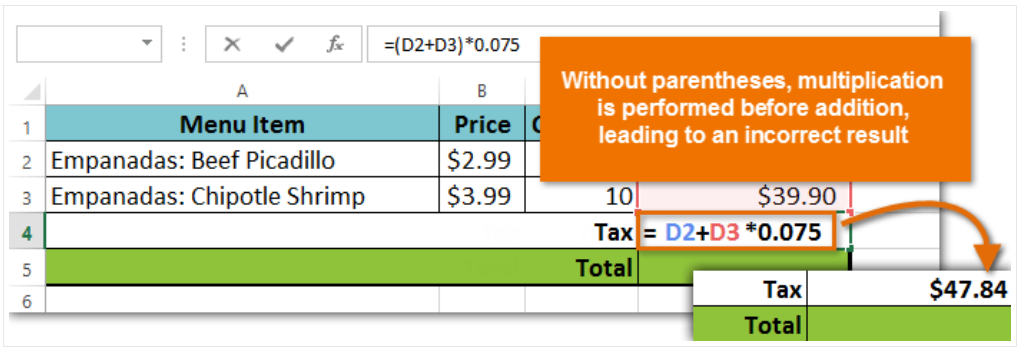
In the example below, we will demonstrate how Excel solves a complex formula using the order of operations. Here, we want to calculate the cost of **sales tax** for a catering invoice. To do this, we'll write our formula as **=(D2+D3)*0.075** in cell **D4**. This formula will add the prices of our items together and then multiply that value by the 7.5% tax rate (which is written as 0.075) to calculate the cost of sales tax.



Excel follows the order of operations and first adds the values inside the parentheses: **(44.85+39.90) = $84.75**. It then multiplies that value by the tax rate: **$84.75*0.075**. The result will show that the sales tax is **$6.36**.



It is especially important to enter complex formulas with the correct order of operations. Otherwise, Excel will not calculate the results accurately. In our example, if the **parentheses** are not included, the multiplication is calculated first and the result is incorrect. Parentheses are the best way to define which calculations will be performed first in Excel.



If-Then Formulas

Excel functions, or formulas, lie at the heart of the application's deep well of capabilities. Today we'll tackle IF statements, a string of commands that determine whether a condition is met or not. Just like a yes-no question, if the specified condition is true, Excel returns one user-determined value and, if false, it returns another.

The IF statement is also known as a logical formula: IF, then, else. **If** something is true, **then** do this, **else**/otherwise do that. For example, **if** it's raining, **then** close the windows, **else**/otherwise leave the windows open.

The syntax (or sentence structure; that is, the way the commands are organized in the formula) of an Excel IF statement is: =IF(logic_test, value_if true, value_if_false). IF statements are used in all programming languages and, although the syntax may vary slightly, the function provides the same results.

Remember: Learning Excel functions/formulas and how they work are the first steps toward using Visual Basic, Microsoft's event-driven programming language. Here are five easy IF statements to get you started.

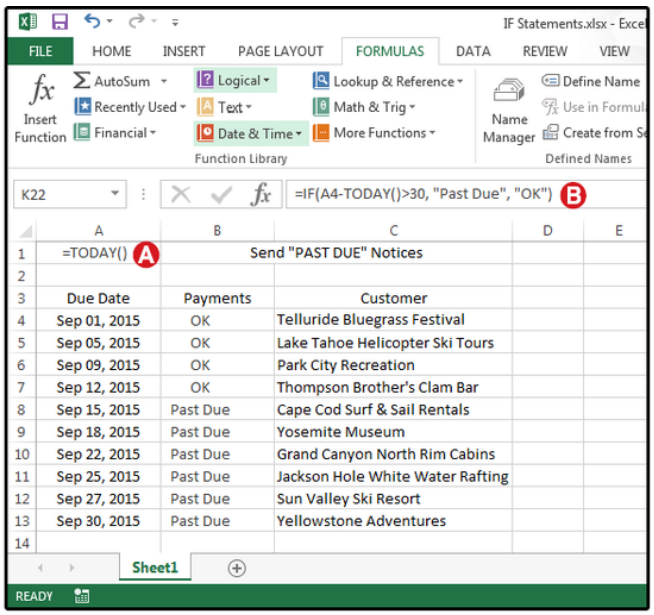
Past-due notices

In this spreadsheet, the customer's payment due date is listed in column A, the payment status is shown in column B, and the customer's company name is in column C. The company accountant enters the date that each payment arrives, which generates this Excel spreadsheet. The bookkeeper enters a formula in column B that calculates which customers are more than 30 days past due, then sends late notices accordingly.

A. Enter the formula: =TODAY() in cell A1, which displays as the current date.

B. Enter the formula: =IF(A4-TODAY()>30, "Past Due", "OK") in cell B4.

In English, this formula means: **If** the date in cell A4 minus today's date is greater than 30 days, **then** enter the words 'Past Due' in cell B4, **else**/otherwise enter the word 'OK.' Copy this formula from B4 to B5 through B13.

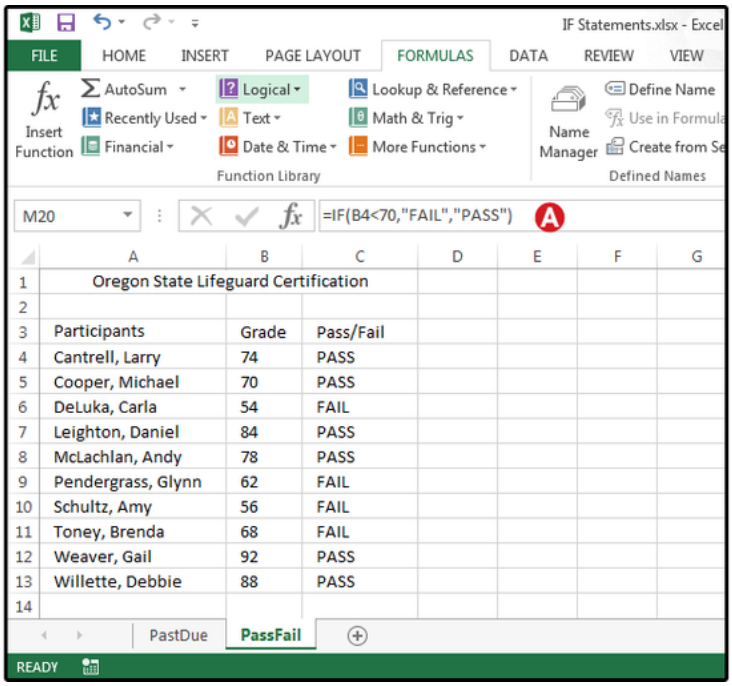


Pass/Fail lifeguard test

The Oregon Lifeguard Certification is a Pass/Fail test that requires participants to meet a minimum number of qualifications to pass. Scores of less than 70 percent fail, and those scores greater than that, pass. Column A lists the participants' names; column B shows their scores; and column C displays whether they passed or failed the course. The information in column C is attained by using an IF statement.

Once the formulas are entered, you can continue to reuse this spreadsheet forever. Just change the names at the beginning of each quarter, enter the new grades at the end of each quarter, and Excel calculates the results.

A. Enter this formula in cell C4: =IF(B4<70,"FAIL","PASS"). This means if the score in B4 is less than 70, **then** enter the word *FAIL* in cell B4, **else**/otherwise enter the word *PASS*. Copy this formula from C4 to C5 through C13.



Use an IF statement to convert numeric scores to a pass-fail status.

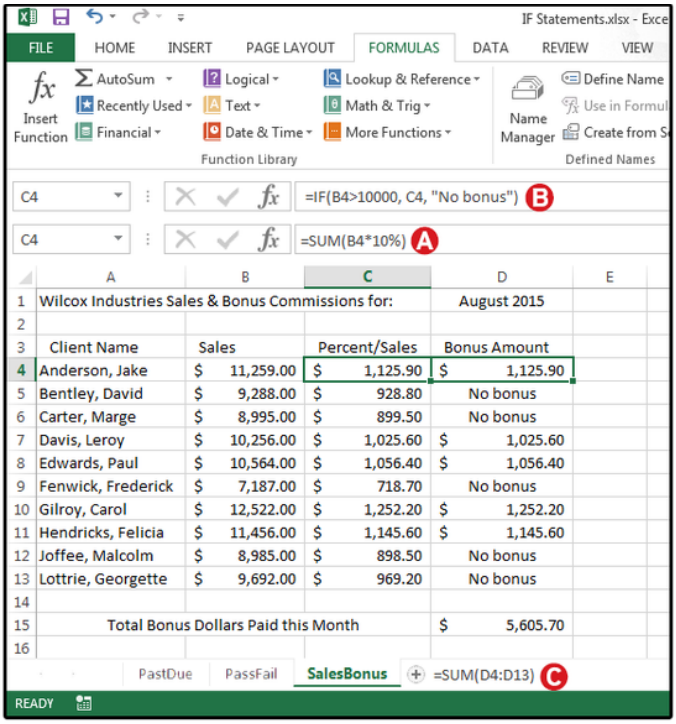
Sales & bonus commissions

Wilcox Industries pays its sales staff a 10-percent commission for all sales greater than $10,000. Sales below this amount do not receive a bonus. The names of the sales staff are listed in column A. Enter each person's total monthly sales in column B. Column C multiples the total sales by 10 percent, and column D displays the commission amount or the words 'No Bonus.' Now you can add the eligible commissions in column D and find out how much money was paid out in bonuses for the month of August.

A. Enter this formula in cell C4: =SUM(B4*10%), then copy from C4 to C5 through C13. This formula calculates 10 percent of each person's sales.

B. Enter this formula in cell D4: =IF(B4>10000, C4, "No bonus"), then copy from D4 to D5 through D13. This formula copies the percentage from column C for sales greater than $10,000 or the words 'No Bonus' for sales less than $10,000 into column D.

C. Enter this formula in cell D15: =SUM(D4:D13). This formula sums the total bonus dollars for the current month.



Use an IF statement to calculate sales bonus commissions.

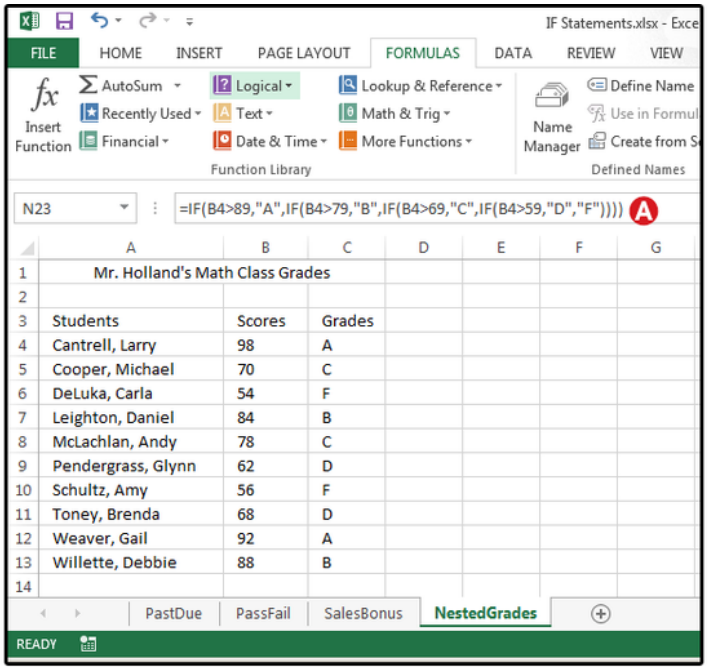
Convert scores to grades with nested IF statements

This example uses a "nested" IF statement to convert the numerical Math scores to letter grades. The syntax for a nested IF statement is this: IF data is true, then do this; IF data is true, then do this; IF data is true, then do this; IF data is true, then do this; else/otherwise do that. You can nest up to seven IF functions.

The student's names are listed in column A; numerical scores in Column B; and the letter grades in column C, which are calculated by a nested IF statement.

A. Enter this formula in cell C4: =IF(B4>89,"A",IF(B4>79,"B",IF(B4>69,"C",IF(B4>59,"D","F")))), then copy from C4 to C5 through C13.

Note: Every open, left parenthesis in a formula must have a matching closed, right parenthesis. If your formula returns an error, count your parentheses.



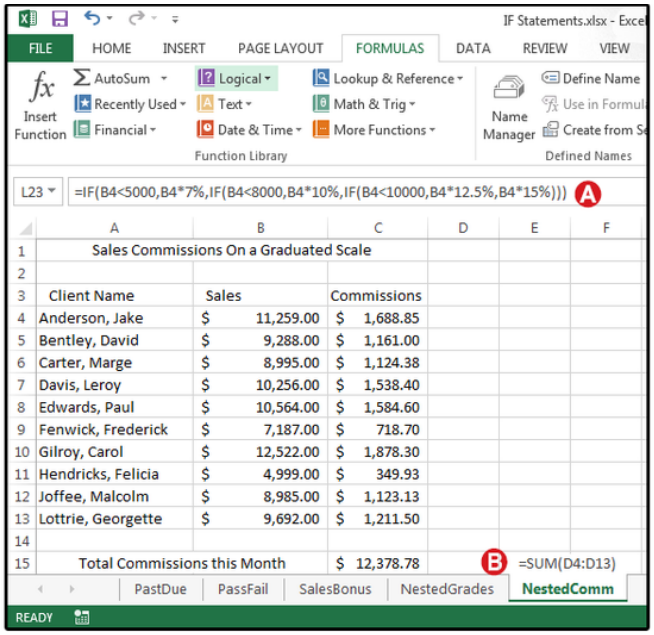
Use a nested IF statement to convert numeric scores to letter grades.

Determine sliding scale sales commissions with nested IF statements

This last example uses another nested IF statement to calculate multiple commission percentages based on a sliding scale, then totals the commissions for the month. The syntax for a nested IF statement is this: IF data is true, then do this; IF data is true, then do this; IF data is true, then do this; else/otherwise do that. The names of the sales staff are listed in column A; each person's total monthly sales are in column B; and the commissions are in column C, which are calculated by a nested IF statement, then totaled at the bottom of that column (in cell C15).

A. Enter this formula in cell C4: `=IF(B4<5000,B4*7%,IF(B4<8000,B4*10%,IF(B4<10000,B4*12.5%,B4*15%)))`, then copy from C4 to C5 through C13.

B. Enter this formula in cell C15: `=SUM(C4:C13)`. This formula sums the total commission dollars for the current month.



Use a nested IF statement to calculate different commissions based on different percentages.

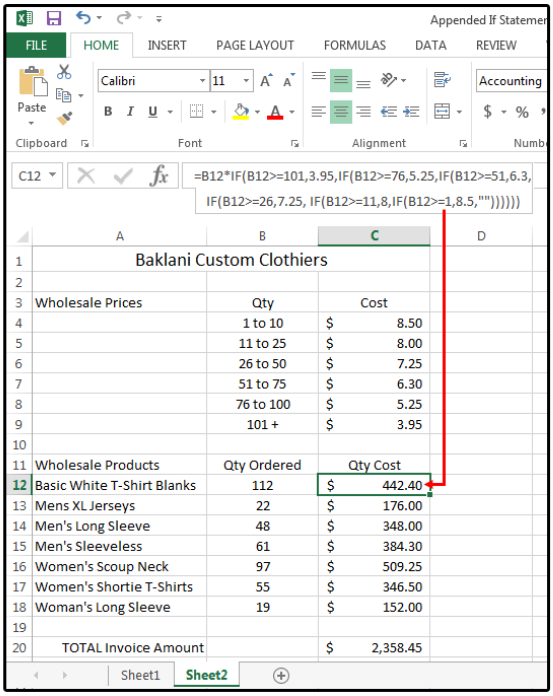
Calculate product price based on quantity

Many retailers and most wholesalers provide customers a price break based on quantity. For example, a wholesaler may sell 1 to 10 T-shirts for $10.50 each, and 11 to 25 T-shirts for $8.75 each. Use the following IF statement formula to determine how much money you can save by purchasing wholesale products for your company in bulk.

1. Enter the header **Wholesale Prices** in cell A3. Enter the headers **Qty** and **Cost** in cells B3 and C3.

2. Enter the quantity breaks in cells B4 through B9.

3. Enter the price breaks in cells C4 through C9.



Calculate product prices based on quantity.

4. Next, enter the headers **Qty Ordered** and **Qty Cost** in cells B11 and C11.

5. Enter the header **Wholesale Products** in cell A11, then enter as many products as you like below.

6. Enter the quantity ordered for each product in column B.

7. And last, enter the following formula in the adjacent cells of column C:

=B12*IF(B12>=101,3.95,IF(B12>=76,5.25,IF(B12>=51,6.3,IF(B12>=26,7.25, IF(B12>=11,8,IF(B12>=1,8.5,””))))))

Note there are no spaces in this formula. Copy down the formula as far as needed in column C to apply to all the wholesale products and quantities in columns A and B.

8. Once the order is complete, create a total line: In column A (in this case, A20), type **TOTAL Invoice Amount**. In column C (in this case, C20), use the SUM function to total the column.

Now, that all the formulas are in place, all you have to do each week (or month) is enter the new quantities in column B, and Excel does the rest. If the prices change, just modify the matrix at the top of the spreadsheet, and all the other formulas and totals will automatically adjust.

Using other functions inside the IF statement

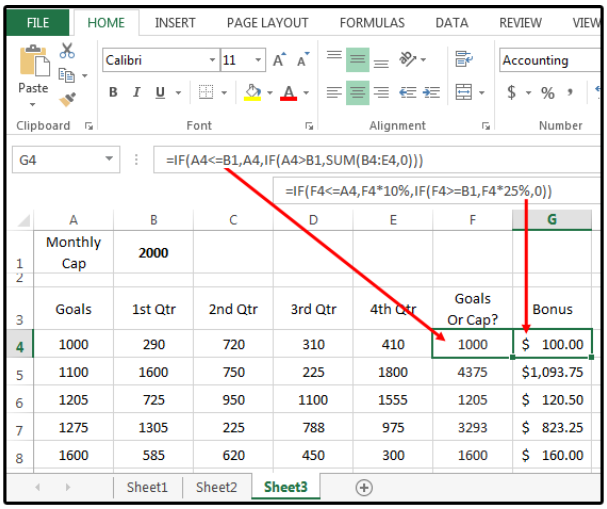
In this example, we'll nest two IF statements and insert the SUM function all into the same formula. The purpose is to determine whether the individual's goals are less than, equal to, or greater than the company's monthly cap.

1. Enter a monthly cap amount in B1, which can be changed every month to reflect the company's expectations.

2. Enter the following headers in cells A3 through G3: **Goals**, **1st Qtr**, **2nd Qtr**, **3d Qtr**, **4th Qtr**, **Goals or Cap?**, and **Bonus**.

3. Enter the top five marketing persons' goal amounts in cells A4 through A8.

4. Enter the same individuals' quarterly totals in cells B4 through E8.



Using other functions INSIDE the If Statement

For employees who achieved their goals, the bonus is 10% of the goal amount. For employees who exceeded the monthly cap, the bonus is 25% of their actual sales. Use these formulas to determine the bonus amounts.

5. Enter this formula in cells F4 through F8: =IF(A4<=B1,A4,IF(A4>B1,SUM(B4:E4,0)))

6. Enter this formula in cells G4 through G8: =IF(F4<=A4,F4*10%,IF(F4>=B1,F4*25%,0))

Use IF to avoid dividing by zero

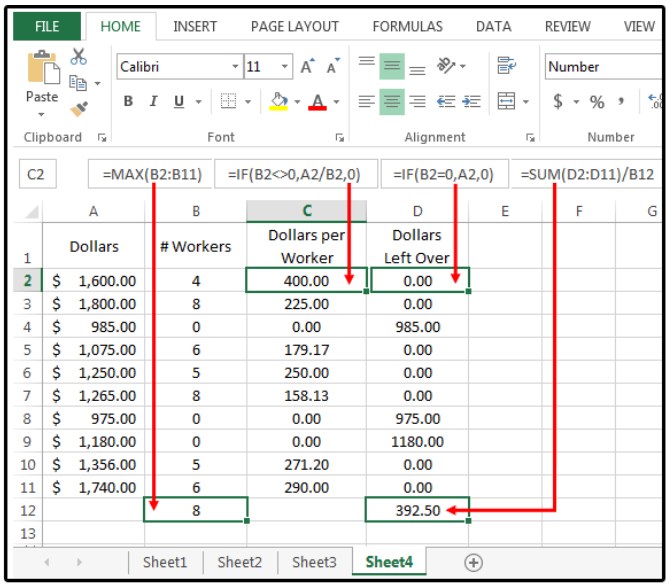
In Excel, if you try to divide any number by zero, the error message that Excel displays is this: #DIV/0!

So, what to do if you have a large database that requires a calculation based on division, and some of the cells contain zeros? Use the following IF statement formula to address this problem.

1. To track the money collected per month by the staff at Friar Tuck's Fundraisers, open a spreadsheet and create the following headers: **Dollars** in A1, **Dollars per Worker** in B1, **Dollars Left Over** in C1.

2. Enter some dollar amounts in column A and the number of staff working per day in column B. Column B will include some zeroes because, some days, nobody works (but donations still come in through the mail).

2. Enter this formula in C2 through C11: =IF(B2<>0,A2/B2,0). By using an IF statement to prevent Excel from dividing any of the funds collected by zero, only the valid numbers are calculated.



Use IF to avoid errors from dividing by zero

3. You could use another IF statement to locate the zeroes in column B, then total the corresponding dollars in column A, which would then be divided equally among all the staff. Copy this formula in D2 through D11: =IF(B4=0,A4,0).

4. Next, move your cursor to B12 and enter this formula =MAX(B2:B11) to locate the largest number of workers in column B, which is eight.

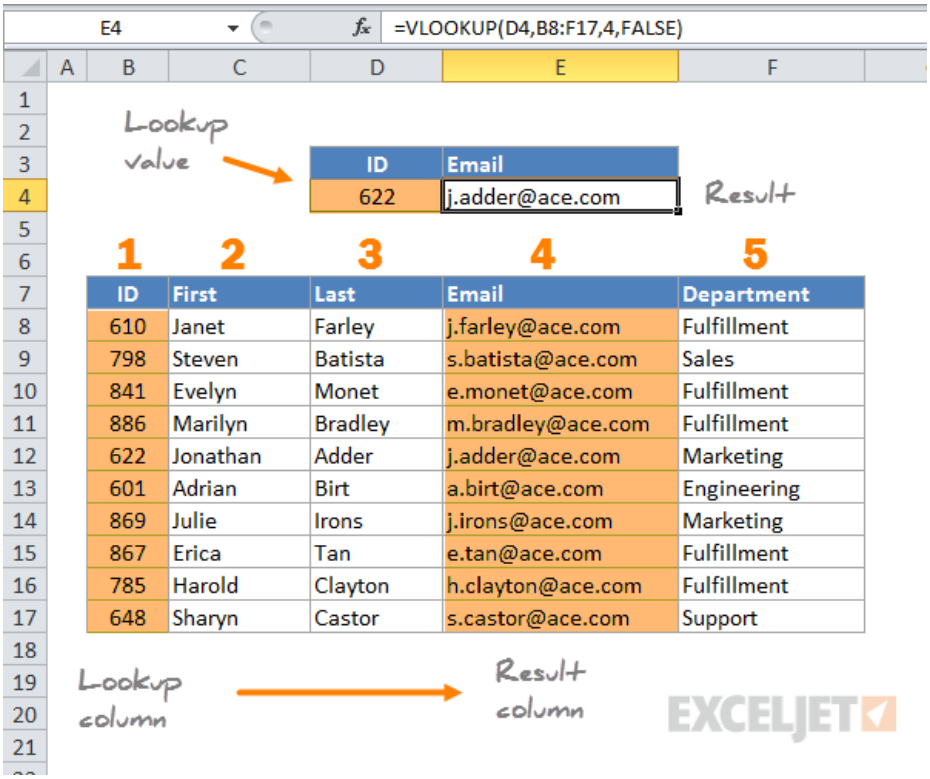
5. Move your cursor to D12 and enter this formula: =SUM(D2:D11)/B12. The answer is $392.50 for each of the eight staff members working at Friar Tuck's Fundraisers.

VLOOKUPS

VLOOKUP is a function to lookup up and retrieve data in a table. The "V" in VLOOKUP stands for vertical, which means the data in the table must be arranged vertically, with data in rows. (For horizontally structured data, see HLOOKUP).

If you have a well structured table, with information arranged vertically, and a column on the left which you can use to match a row, you can probably use VLOOKUP.

VLOOKUP requires that the table be structured so that lookup values appear in the left-most column. The data you want to retrieve (result values) can appear in any column to the right. When you use VLOOKUP, imagine that every column in the table is numbered, starting from the left. To get a value from a particular column, simply supply the appropriate number as the "column index". In the example below, we want to look up the email address, so we are using the number 4 for column index:



In the above table, the employee IDs are in column 1 on the left and the email addresses are in column 4 to the right.

To use VLOOKUP, you supply 4 pieces of information, or "arguments":

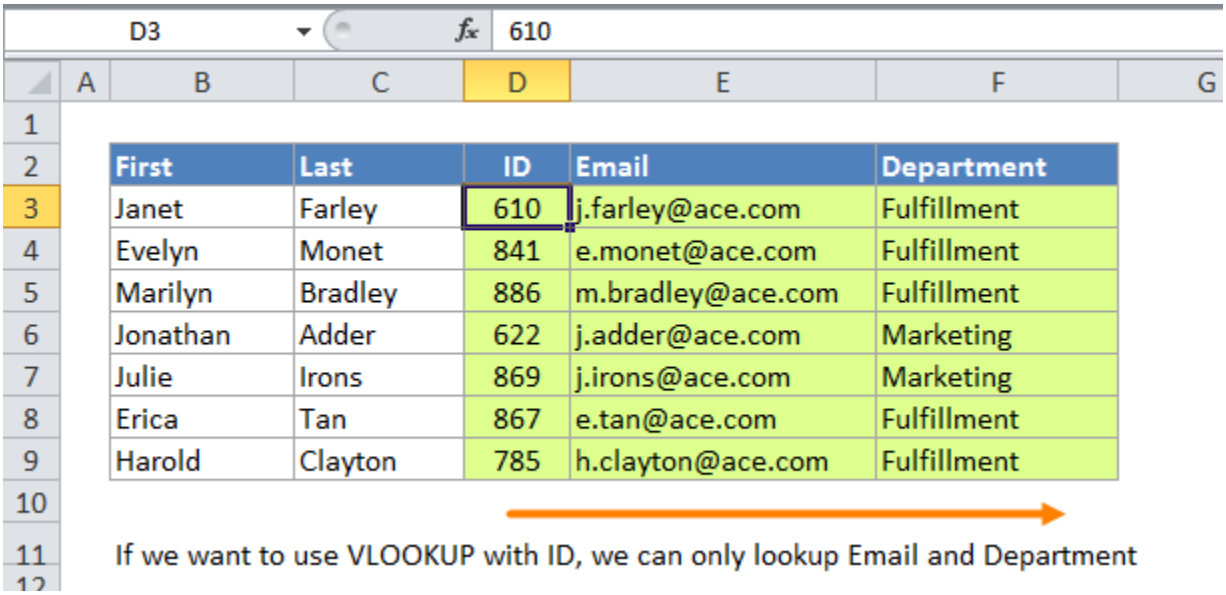
1. The value you are looking for (**lookup_value**)
2. The range of cells that make up the table (**table_array**)
3. The number of the column from which to retrieve a result (**column_index**)
4. The match mode (**range_lookup**, TRUE = approximate, FALSE = exact)

Video: How to use VLOOKUP (3 min)

2. VLOOKUP only looks right

Perhaps the biggest limitation of VLOOKUP is that it can only look to the right to retrieve data.

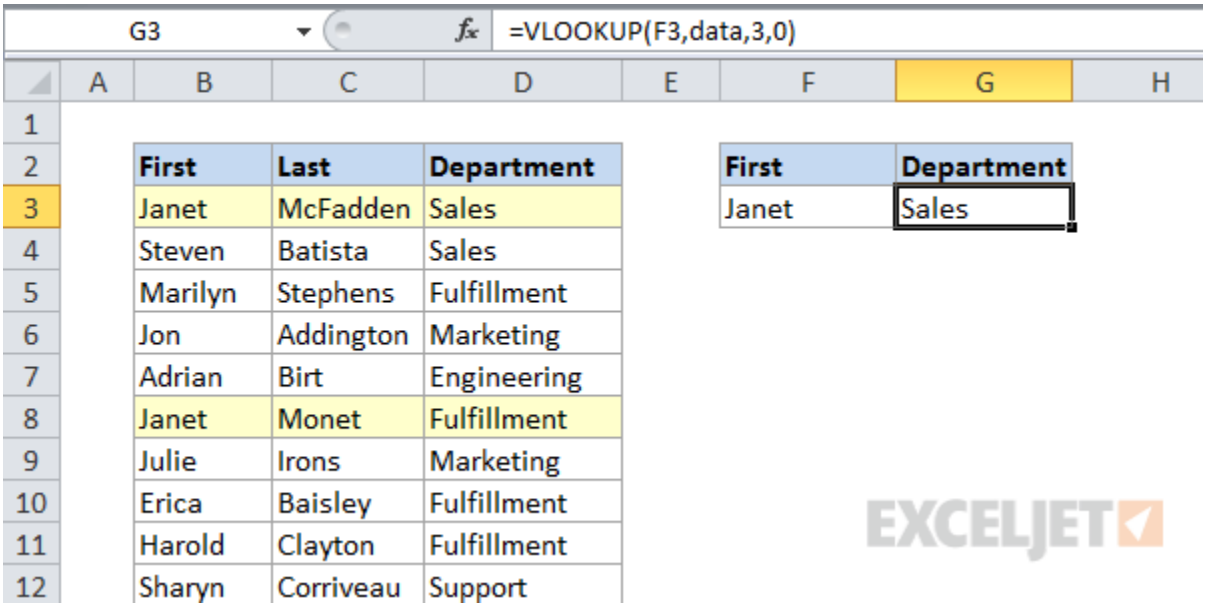
This means that VLOOKUP can only get data from columns to the right of first column in the table. When lookup values appear in the first (leftmost) column, this limitation doesn't mean much, since all other columns are already to the right. However, if the lookup column appears inside the table somewhere, you'll only be able to lookup values from columns to the right of that column. You'll also have to supply a smaller table to VLOOKUP that starts with the lookup column.



You can overcome this limitation by using INDEX and MATCH instead of VLOOKUP.

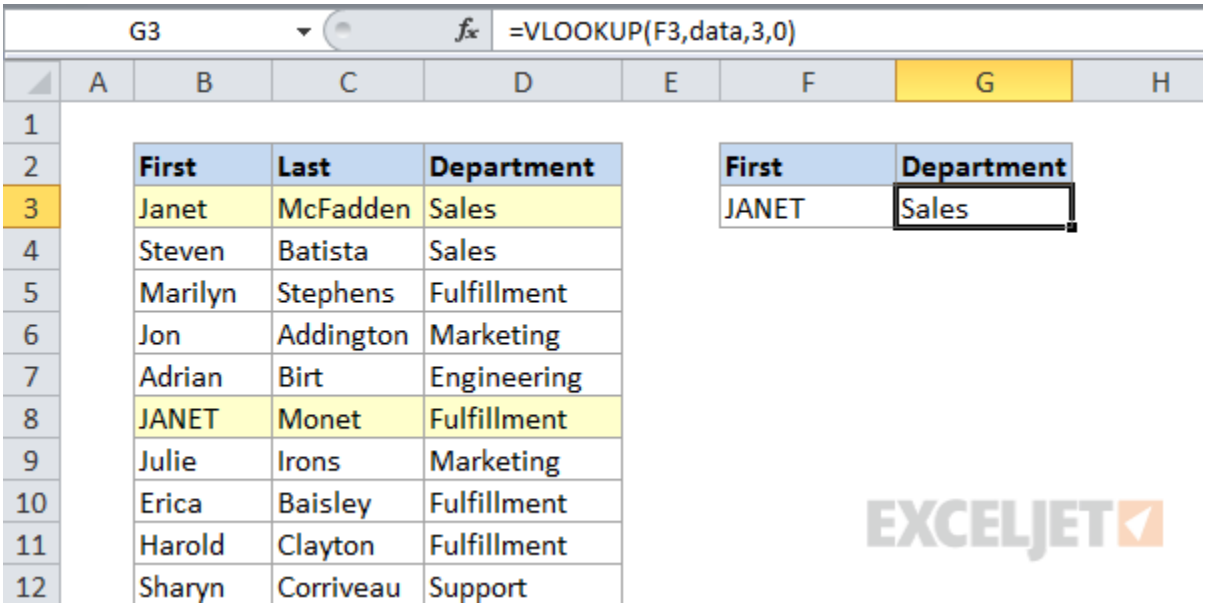
3. VLOOKUP always finds the first match

If the lookup column contains duplicate values, VLOOKUP will match the first value only. If the first column in the table contains no duplicates, this obviously doesn't matter. However, if the first column does contain duplicate values, VLOOKUP will only match the first instance. In this example, we are using VLOOKUP to find a first name. Although there are two "Janet"s in the list, VLOOKUP matches only the first:



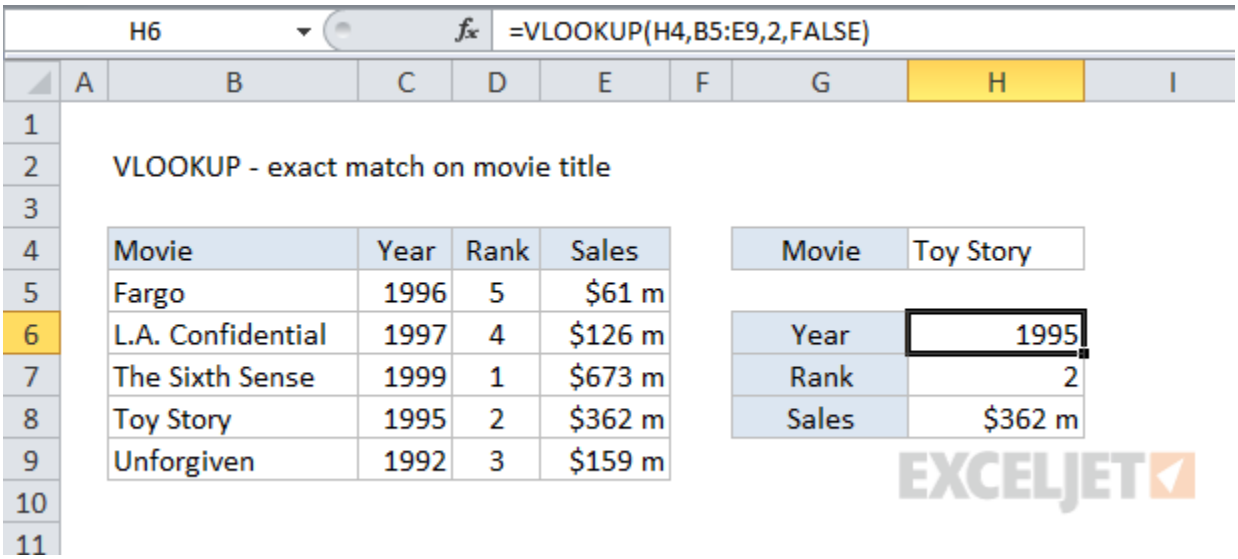
4. VLOOKUP is not case-sensitive

When looking up a value, VLOOKUP does not process upper and lower case text differently. To VLOOKUP, a product code like "PQRF" is identical to "pqrf". In the example below, we are looking for uppercase "JANET" but VLOOKUP does not distinguish case so it simply matches "Janet", since that's the first match it finds:



5. VLOOKUP has two matching modes

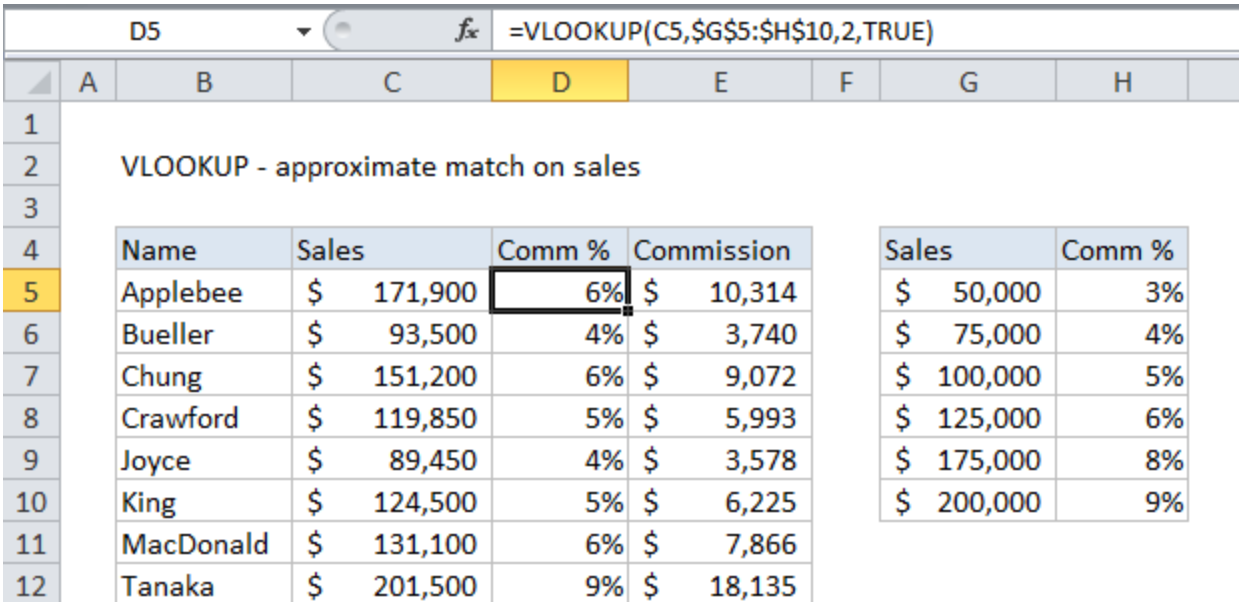
VLOOKUP has two modes of operation: exact match and approximate match. In most cases, you'll probably want to use VLOOKUP in exact match mode. This makes sense when you want to lookup information based on a unique key of some kind, for example, product information based on a product code, or movie data based on a movie title:



The formula in H6 to lookup year based on an exact match of movie title is:

=VLOOKUP(H4,B5:E9,2,FALSE) // FALSE = exact match

However, you'll want to use approximate mode in cases where you're not matching on a unique id, but rather you're looking up the "best match" or the "best category". For example, perhaps you're looking up postage based on weight, looking up tax rate based on income, or looking up a commission rate based on a monthly sales number. In these cases, you likely won't find the exact lookup value in the table. Instead, you want VLOOKUP to get you the best match for a given lookup value.



The formula in D5 does an approximate match to retrieve the correct commission:

=VLOOKUP(C5,$G$5:$H$10,2,TRUE) // TRUE = approximate match

6. Caution: VLOOKUP uses approximate match by default

Exact and approximate matching in VLOOKUP is controlled by the 4th argument, called "range lookup". This name is not intuitive, so you'll just have to memorize how it works.

For exact match, use FALSE or 0. For approximate match, set range_lookup to TRUE or 1:

=VLOOKUP(value,table,column,TRUE) // approximate match
=VLOOKUP(value,table,column,FALSE) // exact match

Unfortunately, the 4th argument, range_lookup, is optional and defaults to TRUE, which means VLOOKUP will do an approximate match by default. When doing an approximate match, VLOOKUP assumes the table is sorted and performs a binary search. During a binary search, if VLOOKUP finds an exact match value, it returns a value from that row. If however, VLOOKUP encounters a value greater than the lookup value, it will return a value from the previous row.

This is a dangerous default because many people unwittingly leave VLOOKUP in it's default mode, which can cause an incorrect result when the table is not sorted.

To avoid this problem, make sure to use FALSE or zero as the 4th argument when you want an exact match.

7. You can force VLOOKUP to do an exact match

To force VLOOKUP to find an exact match, make sure to set the 4 argument (range_lookup) to FALSE or zero. These two formulas are equivalent:

=VLOOKUP(value, data, column, FALSE)
=VLOOKUP(value, data, column, 0)

In exact match mode, when VLOOKUP can't find a value, it will return #N/A. This a clear indication that the value isn't found in the table.

8. You can tell VLOOKUP to do an approximate match

To use VLOOKUP in approximate match mode, either omit the 4th argument (range_lookup) or supply it as TRUE or 1. These 3 formulas are equivalent:

=VLOOKUP(value, data, column)
=VLOOKUP(value, data, column, 1)
=VLOOKUP(value, data, column, TRUE)

We recommend that you always set range_lookup argument explicitly, even though VLOOKUP doesn't require it. That way, you always have a visual reminder of the match mode you expect.

Video: How to use VLOOKUP for approximate matches

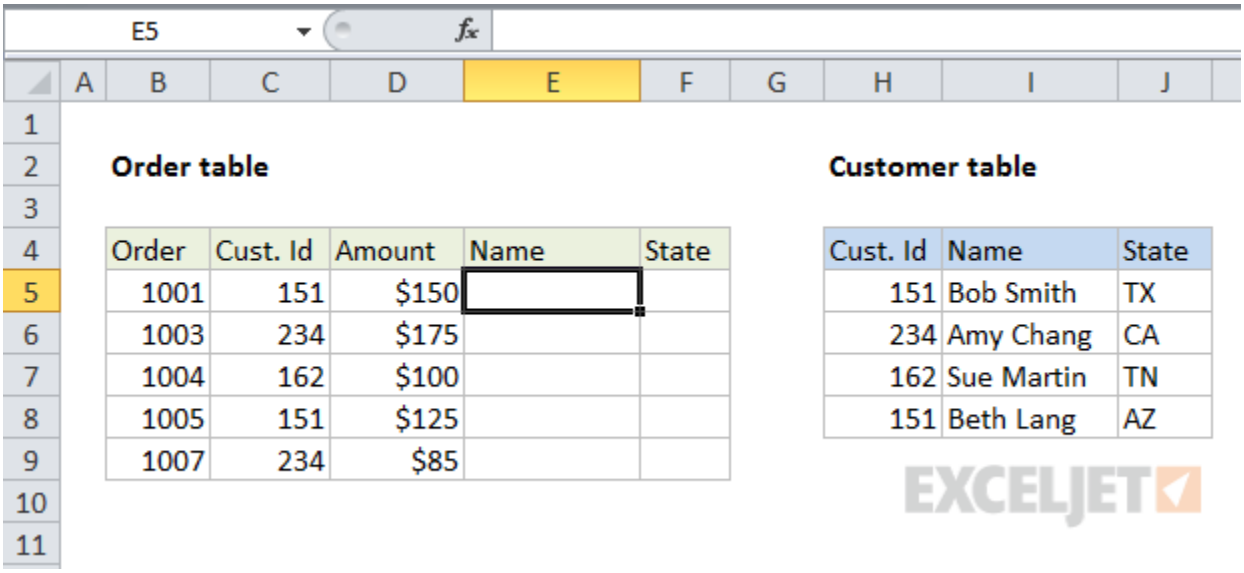
9. For approximate matches, data must be sorted

If you are using approximate mode matching, your data *must be sorted* ascending order by lookup value. Otherwise, you may get an incorrect results. Also be aware that sometimes text data may *look* sorted, even though it's not.

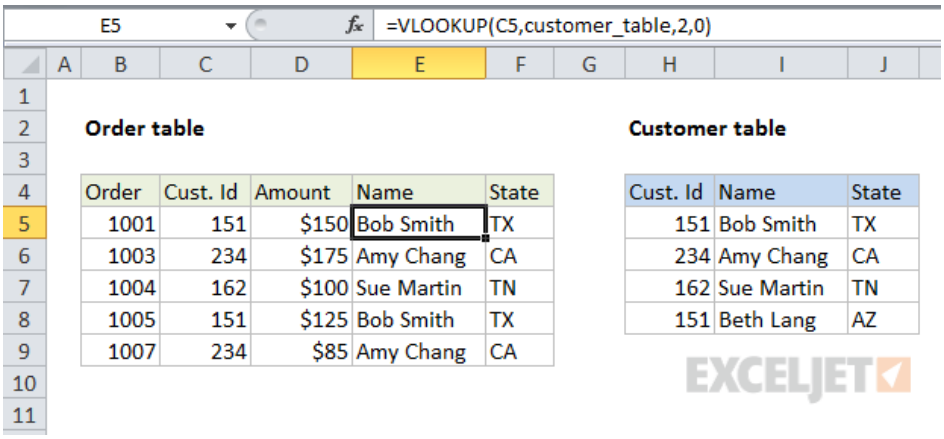
Felienne Hermans has a great example of this problem here, from a cool analysis she did on actual Enron spreadsheets!

10. VLOOKUP can merge data in different tables

A common use case for VLOOKUP is to join data from two or more tables. For example, perhaps you have order data in one table, and customer data in another and you want to bring some customer data into the order table for analysis:



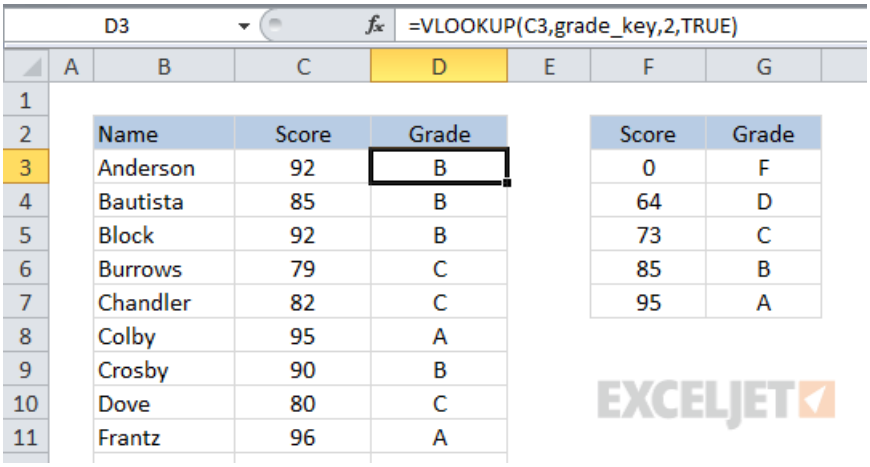
Because the customer id exists in both tables, you can use this value to pull in the data you want with VLOOKUP. Just configure VLOOKUP to use the id value in table one, and the data in table 2, with the required column index. In the example below, we are using two VLOOKUP formulas. One to pull in the customer name, and the other to pull in the customer state.



11. VLOOKUP can classify or categorize data

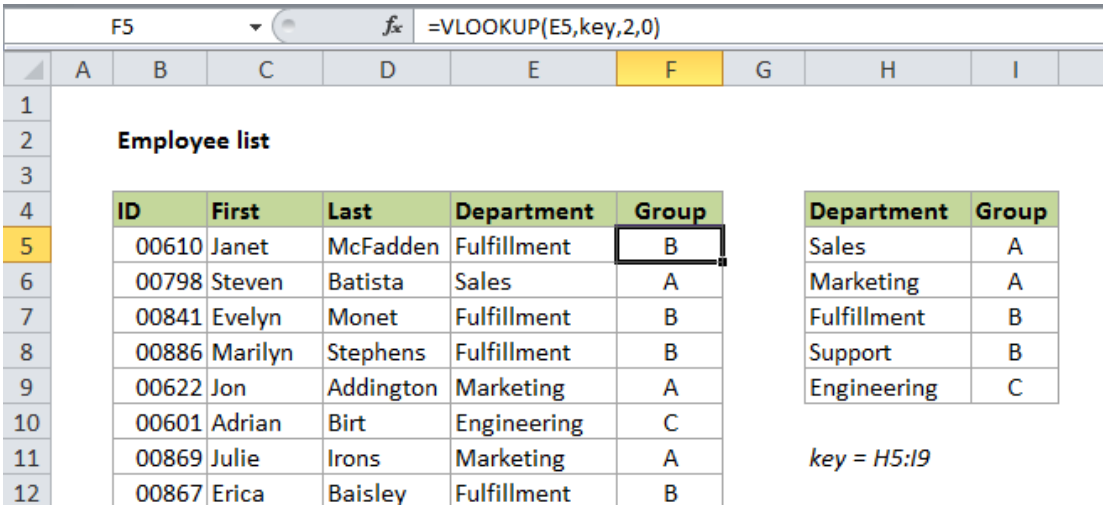
If you ever need to apply arbitrary categories to data records, you can easily do so with VLOOKUP, by using a table that acts as the "key" to assign categories.

A classic example is grades, where you need to assign a grade based on a score:



In this case, VLOOKUP is configured for approximate match, so it's important that the table be sorted in ascending order.

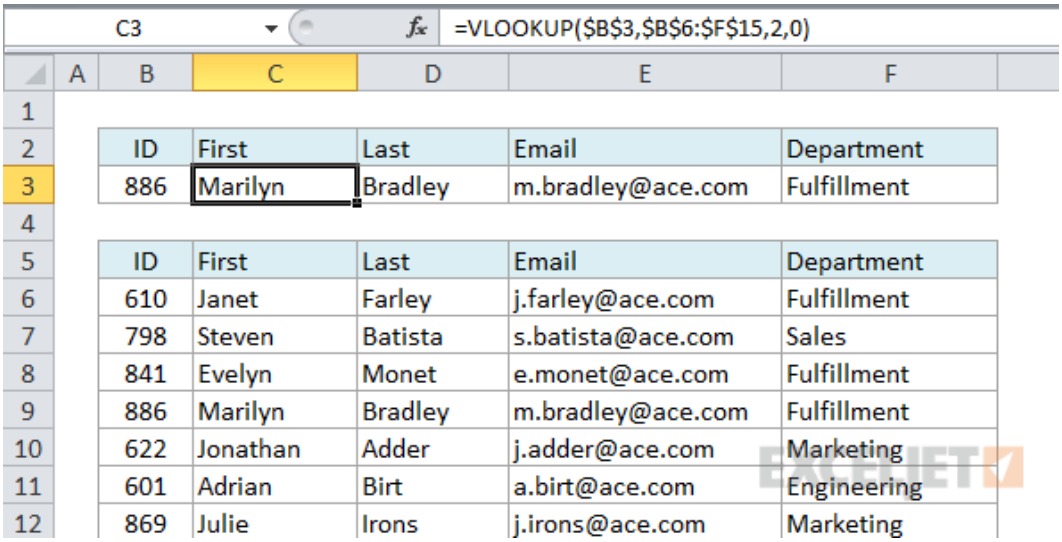
But you can also use VLOOKUP to assign arbitrary categories. In the example below, we are using VLOOKUP to to calculate a group for each department using a small table (named "key") that defines the grouping.



12. Absolute references make VLOOKUP more portable

In situations where you plan to retrieve information from more than one column in a table, or if you need to copy and paste VLOOKUP, you can save time and aggravation by using absolute references for the lookup value and table array. This lets you copy the formula, and then change only the column index number to use the same lookup to get a value from a different column.

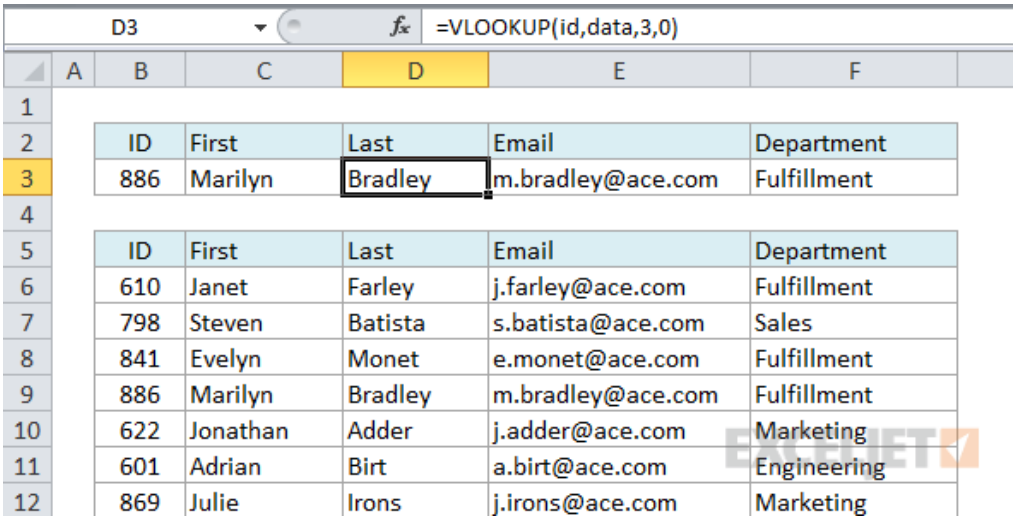
For example, because the lookup value and table array are absolute, we can copy the formula across the columns, then come back and change the column index as needed.



13. Named ranges make VLOOKUP easier to read (and more portable)

Absolute ranges are pretty ugly looking, so can make your VLOOKUP formulas a lot cleaner and easier to read by replacing absolute references with named ranges, which are automatically absolute.

For example, in the employee data example above, you can name the input cell "id" and then name the data in the table "data", you can write your formula as follows:



ID	First	Last	Email	Department
886	Marilyn	Bradley	m.bradley@ace.com	Fulfillment

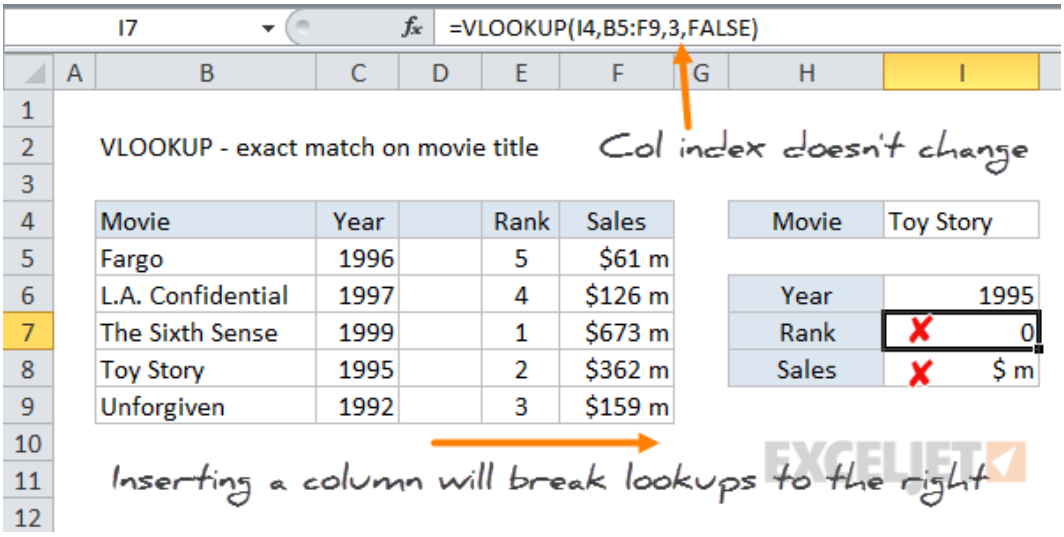
ID	First	Last	Email	Department
610	Janet	Farley	j.farley@ace.com	Fulfillment
798	Steven	Batista	s.batista@ace.com	Sales
841	Evelyn	Monet	e.monet@ace.com	Fulfillment
886	Marilyn	Bradley	m.bradley@ace.com	Fulfillment
622	Jonathan	Adder	j.adder@ace.com	Marketing
601	Adrian	Birt	a.birt@ace.com	Engineering
869	Julie	Irons	j.irons@ace.com	Marketing

Not only is this formula easier to read, but it's also more portable, since named ranges are automatically absolute.

14. Inserting a column may break existing VLOOKUP formulas

If you have existing VLOOKUP formulas in a worksheet, formulas may break if you insert a column in the table. This is because hard-coded column index values don't change automatically when columns are inserted or deleted.

In this example, the lookups for Rank and Sales were broken when a new column was inserted between Year and Rank. Year continues to work because it is on the left of the inserted column:

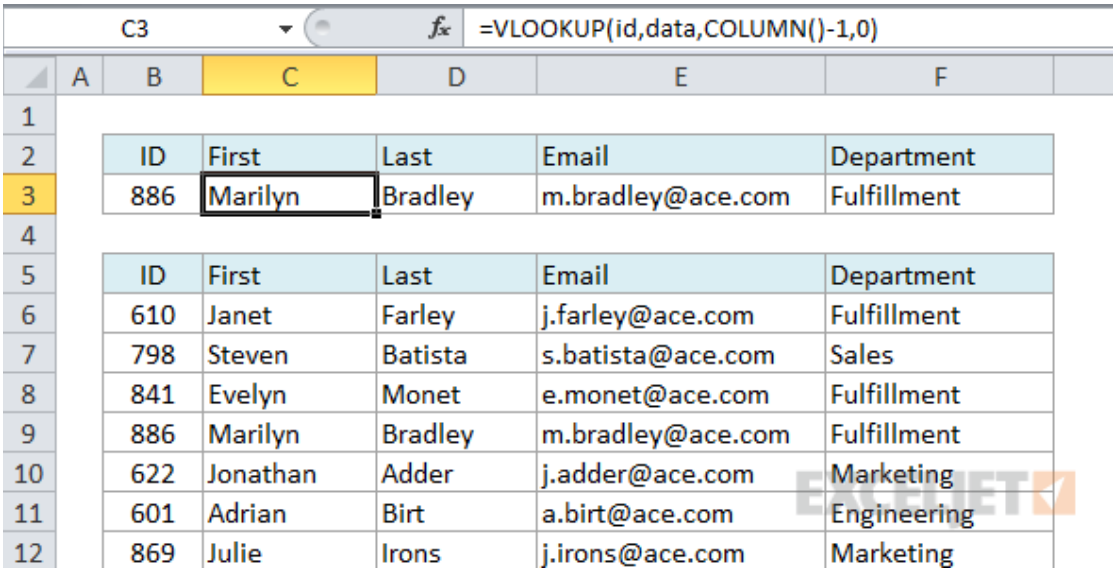


To avoid this problem, you can calculate a column index as described in the next two tips.

15. You can use ROW or COLUMN to calculate a column index

If you're the type who is bothered by any amount of editing after copying a formula, you can use either ROW or COLUMN to generate dynamic column indexes. If you're getting data from consecutive columns, this trick lets you set up one VLOOKUP formula, then copy it across with no changes required.

For example, with the employee data below, we can use the COLUMN function to generate a dynamic column index. For the first formula in cell C3, COLUMN by itself will return 3 (because column C is third in the worksheet) so we simply need to subtract one, and copy the formula across:



ID	First	Last	Email	Department
886	Marilyn	Bradley	m.bradley@ace.com	Fulfillment

ID	First	Last	Email	Department
610	Janet	Farley	j.farley@ace.com	Fulfillment
798	Steven	Batista	s.batista@ace.com	Sales
841	Evelyn	Monet	e.monet@ace.com	Fulfillment
886	Marilyn	Bradley	m.bradley@ace.com	Fulfillment
622	Jonathan	Adder	j.adder@ace.com	Marketing
601	Adrian	Birt	a.birt@ace.com	Engineering
869	Julie	Irons	j.irons@ace.com	Marketing

All formulas are identical with no post-editing required.

The formula we are using is this:

=VLOOKUP(id,data,COLUMN()-1,0)

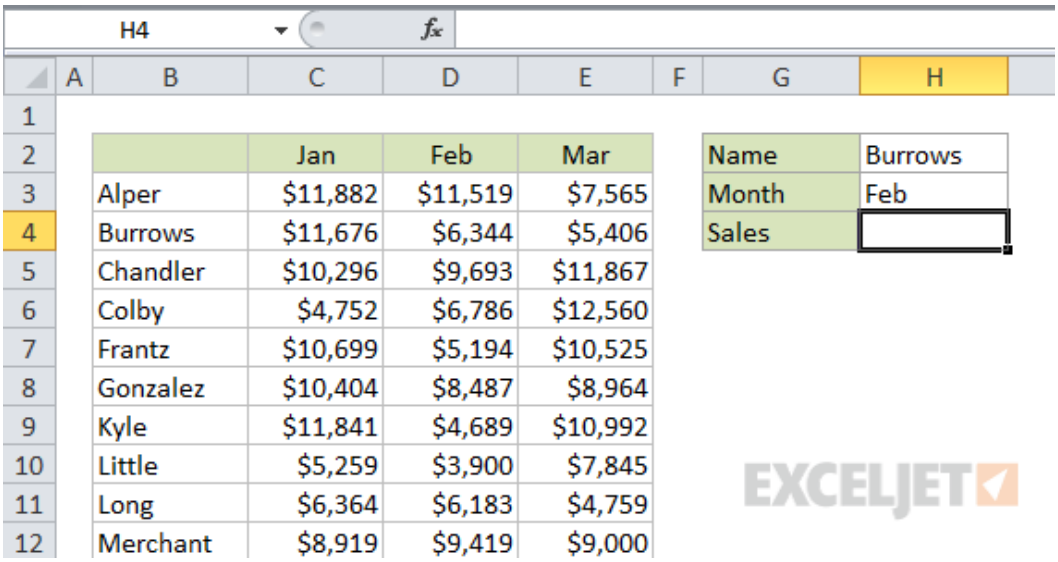
16. Use VLOOKUP + MATCH for a fully dynamic column index

Taking the above tip one step further, you can use MATCH to look up the position of a column in a table and return a fully dynamic column index.

This is sometimes called a two-way lookup since you are looking up both the row and the column.

An example would be looking up sales for a salesperson in a particular month, or looking up the price for a particular product from a particular supplier.

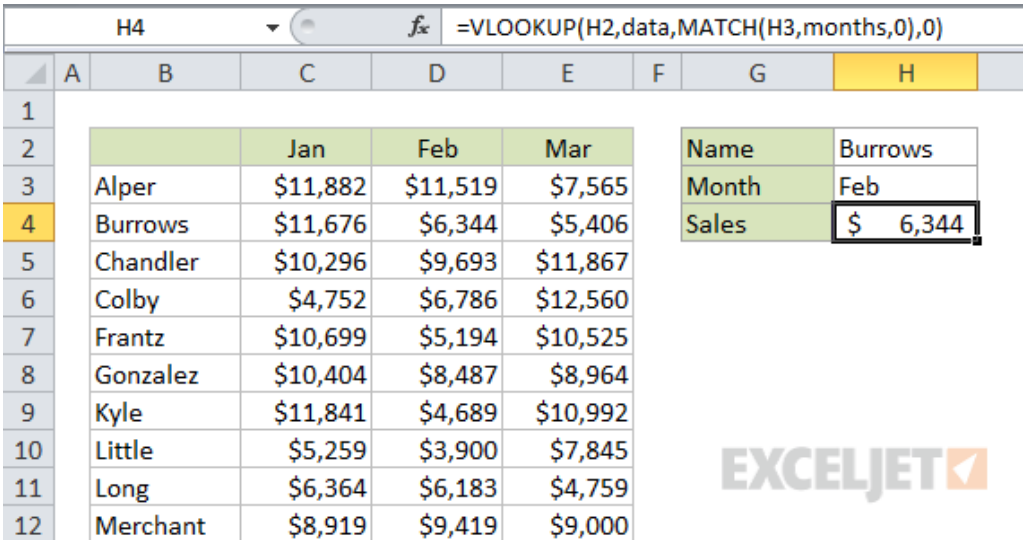
For example, suppose you have sales per month, broken out by salesperson:



	Jan	Feb	Mar
Alper	$11,882	$11,519	$7,565
Burrows	$11,676	$6,344	$5,406
Chandler	$10,296	$9,693	$11,867
Colby	$4,752	$6,786	$12,560
Frantz	$10,699	$5,194	$10,525
Gonzalez	$10,404	$8,487	$8,964
Kyle	$11,841	$4,689	$10,992
Little	$5,259	$3,900	$7,845
Long	$6,364	$6,183	$4,759
Merchant	$8,919	$9,419	$9,000

Name	Burrows
Month	Feb
Sales	

VLOOKUP can easily find the sales person, but it has no way to handle the month name automatically. The trick is to to use the MATCH function in place of a static column index.



Notice that we give match a range that includes all columns in the table in order to "sync up" the the column numbers used by VLOOKUP.

=VLOOKUP(H2,data,MATCH(H3,months,0),0)

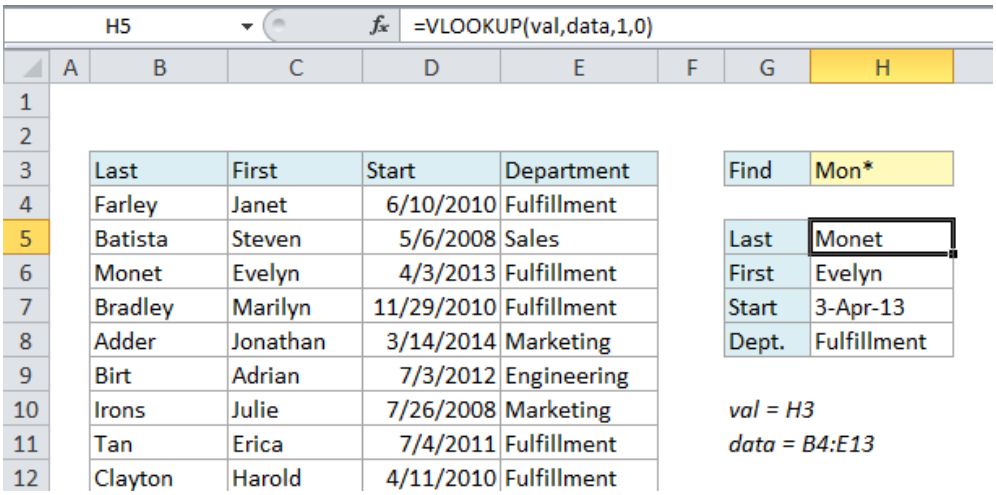
Note: you'll often see two way lookups done with INDEX and MATCH, an approach that offers more flexibility and better performance on big data sets. See how in this quick video: How to do a two-way lookup with INDEX and MATCH.

17. VLOOKUP allows wildcards for partial matching

Any time you're using VLOOKUP in exact match mode, you have the option of using wildcards in the lookup value. It may seem counterintuitive, but wildcards let you do an exact match based on a partial match :)

Excel provides two wildcard characters: an asterisk (*) matches one or more characters, and a question mark (?) matches one character.

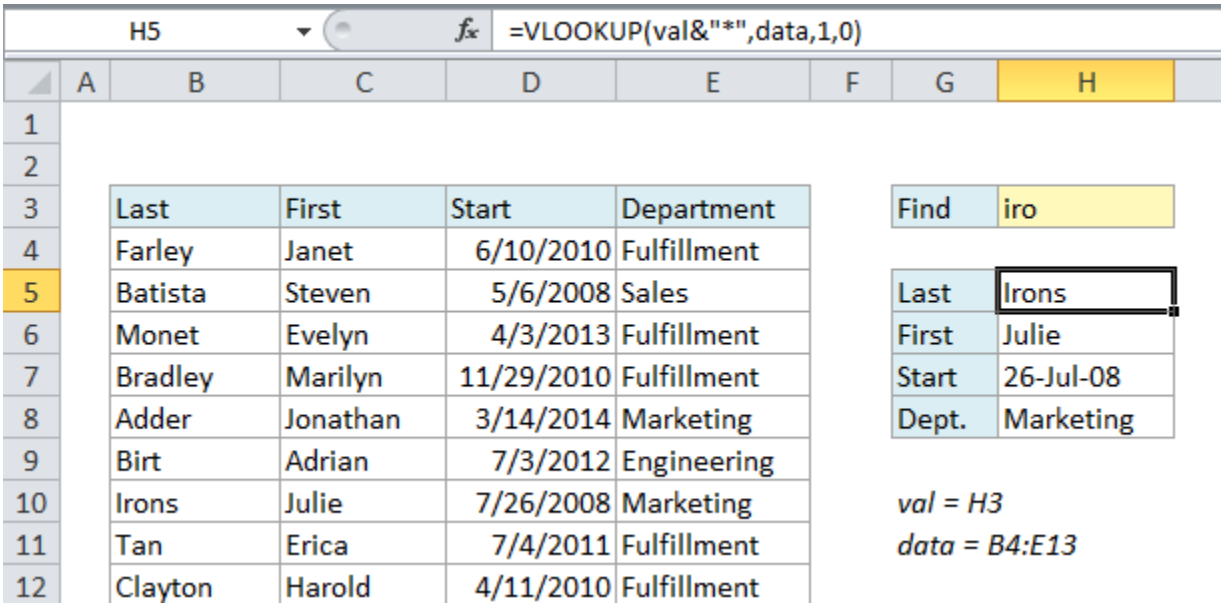
For example, you can type an asterisk directly into a cell and refer to it as a lookup value with VLOOKUP. In the screen below, we have entered "Mon*" into H3, which is a named range called "val". This causes VLOOKUP to match the name "Monet".



The formula in this case is simple:

=VLOOKUP(val,data,1,0)

If you like, you can adjust the VLOOKUP formula to use a built-in wildcard, like the example below, where we simply concatenate the value in H3 with an asterisk.



	Last	First	Start	Department
4	Farley	Janet	6/10/2010	Fulfillment
5	Batista	Steven	5/6/2008	Sales
6	Monet	Evelyn	4/3/2013	Fulfillment
7	Bradley	Marilyn	11/29/2010	Fulfillment
8	Adder	Jonathan	3/14/2014	Marketing
9	Birt	Adrian	7/3/2012	Engineering
10	Irons	Julie	7/26/2008	Marketing
11	Tan	Erica	7/4/2011	Fulfillment
12	Clayton	Harold	4/11/2010	Fulfillment

Find	iro
Last	Irons
First	Julie
Start	26-Jul-08
Dept.	Marketing

val = H3
data = B4:E13

In this case, we are concatenating the asterisk to the lookup value inside the VLOOKUP function:

=VLOOKUP(val&"*",data,1,0)

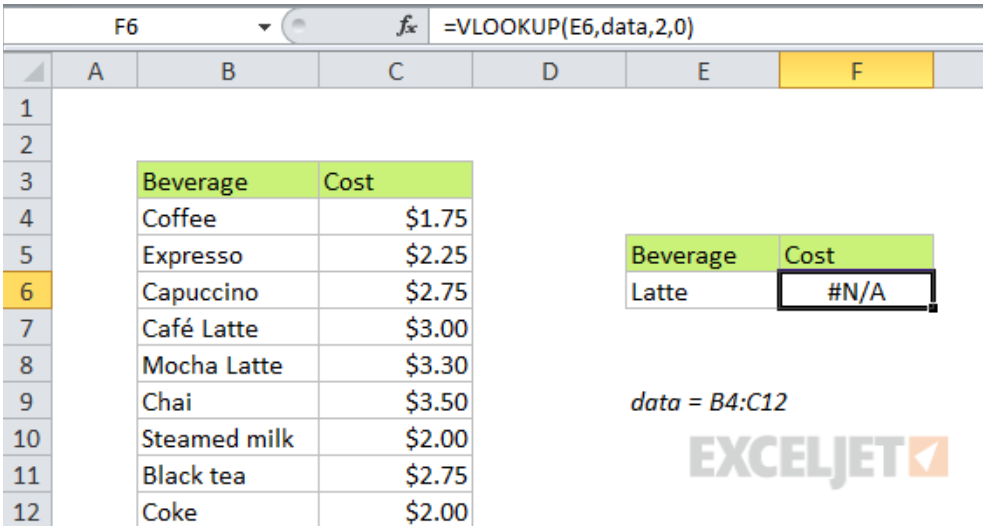
Note: Be careful with wildcards and VLOOKUP. They give you an easy way to create a "lazy match", but they also make it easy to find the wrong match.

18. You can trap #N/A errors and display a friendly message

In exact match mode, VLOOKUP will display the #N/A error when no match is found. In one way, this is useful because it tell you definitively that there is no match in the lookup table. However, #N/A errors aren't very fun to look at, so there are several ways you can trap this error and display something else instead.

Once you start using VLOOKUP, you're bound to run into the #N/A error, which occurs when VLOOKUP isn't able to find a match.

This is a useful error, because VLOOKUP is telling you clearly that it can't find the lookup value. In this example, "Latte" doesn't exist as a beverage in the table, so VLOOKUP throws an #N/A error



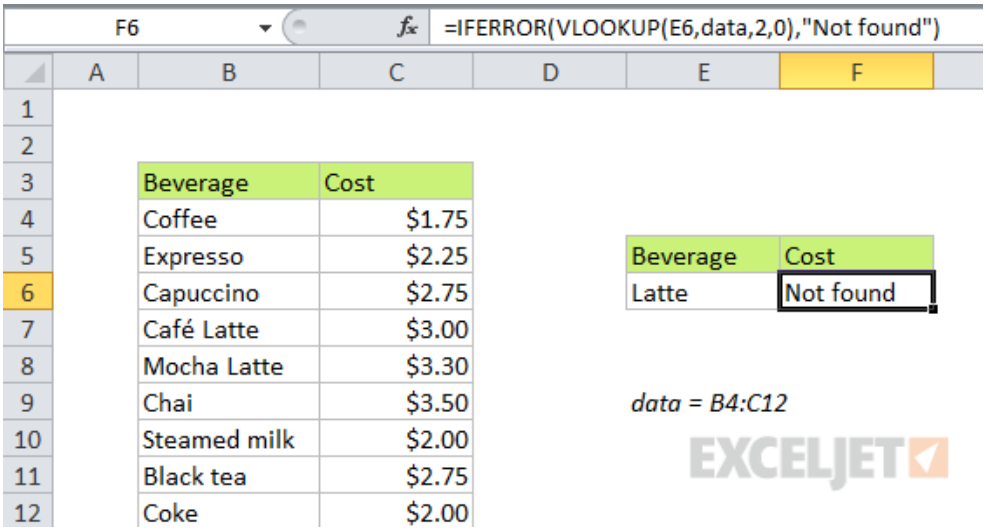
The formula in this case is a completely standard exact match:

=VLOOKUP(E6,data,2,0)

However, #N/A errors aren't very fun to look at, so you might want to catch this error and display a more friendly message.

The easiest way to trap errors with VLOOKUP is to wrap VLOOKUP in the IFERROR function. IFERROR allows you to "catch" any error and return a result of your choosing.

To trap this error and display a "not found" message instead of the error, you can simply wrap the orignal formula inside of IFERROR and set the result you want:



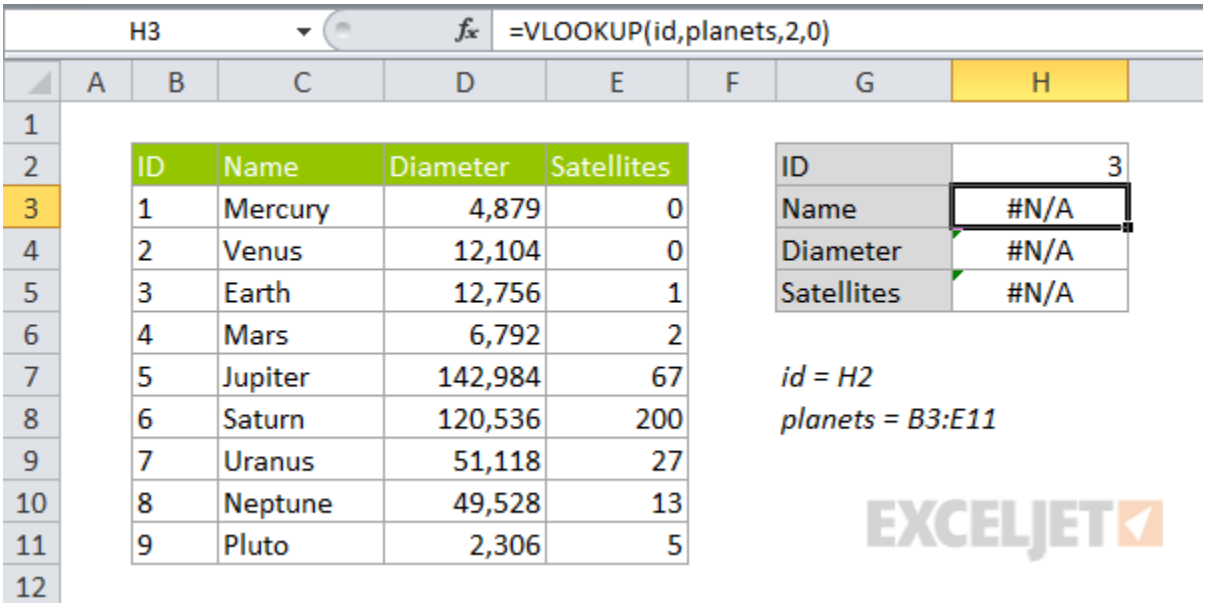
If the lookup value is found, no error occurs and VLOOKUP function returns a normal result. Here is the formula:

=IFERROR(VLOOKUP(E6,data,2,0),"Not found")

19. Numbers as text can cause a match error

Sometimes, the table you are working with in VLOOKUP might contain numbers entered as text. If you are simply retrieving numbers as text from a column in a table, it doesn't matter. But if the first column of the table contains numbers entered as text, you will get an #N/A error if the lookup value is not also text.

In the following example, the ids for the planet table are numbers *entered as text*, which causes VLOOKUP to return an error since the lookup value is the *number* 3:

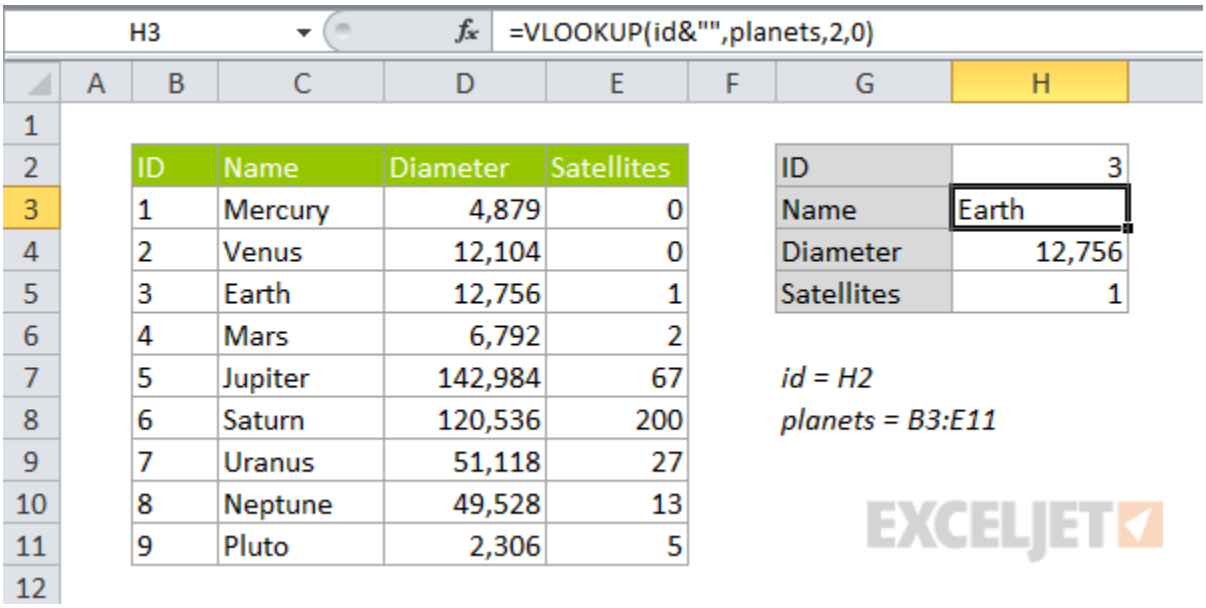


To solve this problem, you need to make sure the lookup value and the first column of the table are both the same data type (either both numbers or both text).

One way to do this is to convert the values in the lookup column to numbers. An easy way to do this is to add zero using paste special.

If you don't have easy control over the source table, you can also adjust the VLOOKUP formula to convert the lookup value to text by concatenating "" to the value like so:

=VLOOKUP(id&"",planets,2,0)



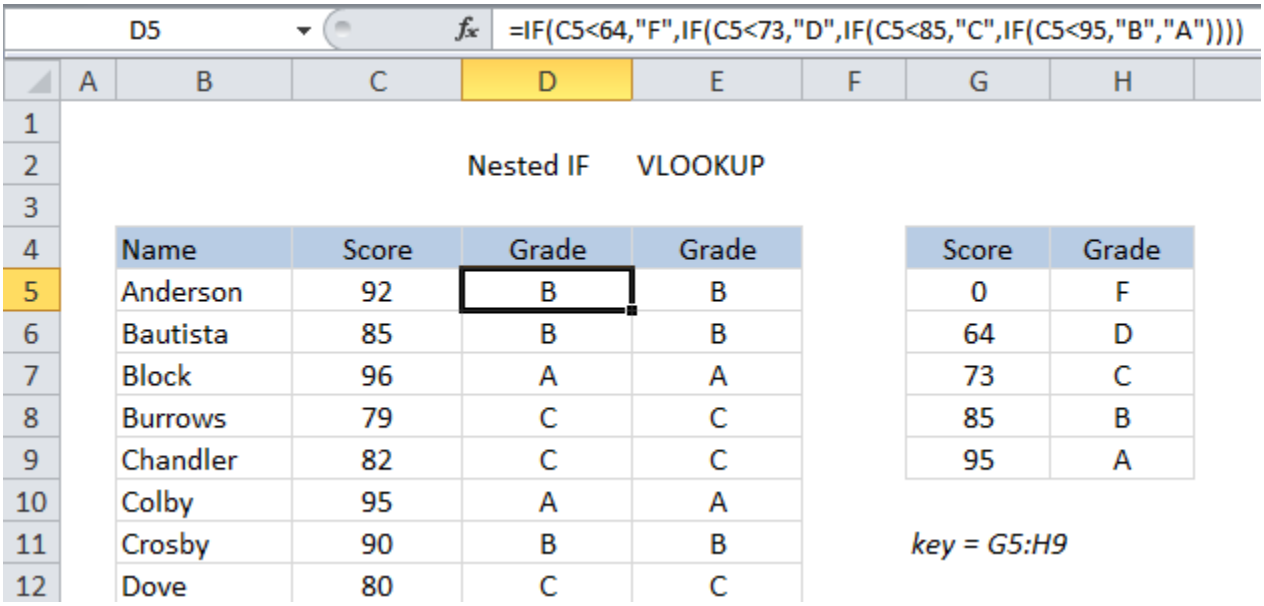
If you can't be certain when you'll have numbers and when you'll have text, you can cater to both options by wrapping VLOOKUP in IFERROR and writing a formula to handle both cases:

=IFERROR(VLOOKUP(id,planets,3,0),VLOOKUP(id&"",planets,3,0))

20. You can use VLOOKUP to replace nested IF statements

One of the more interesting uses of VLOOKUP is to replace nested IF statements. If you've ever built a series of nested IFs, you know that they work fine, but they require a bit of parentheses wrangling. You also have to be careful about the order you work in, so as not to introduce a logical error.

For example, a common use of nested IFs is to assign grades based on a score of some kind. In the example below, you can see a formula has been build with nested IFs to do just that, using the grade key at the right as the guide.



Name	Score	Grade	Grade
Anderson	92	B	B
Bautista	85	B	B
Block	96	A	A
Burrows	79	C	C
Chandler	82	C	C
Colby	95	A	A
Crosby	90	B	B
Dove	80	C	C

Score	Grade
0	F
64	D
73	C
85	B
95	A

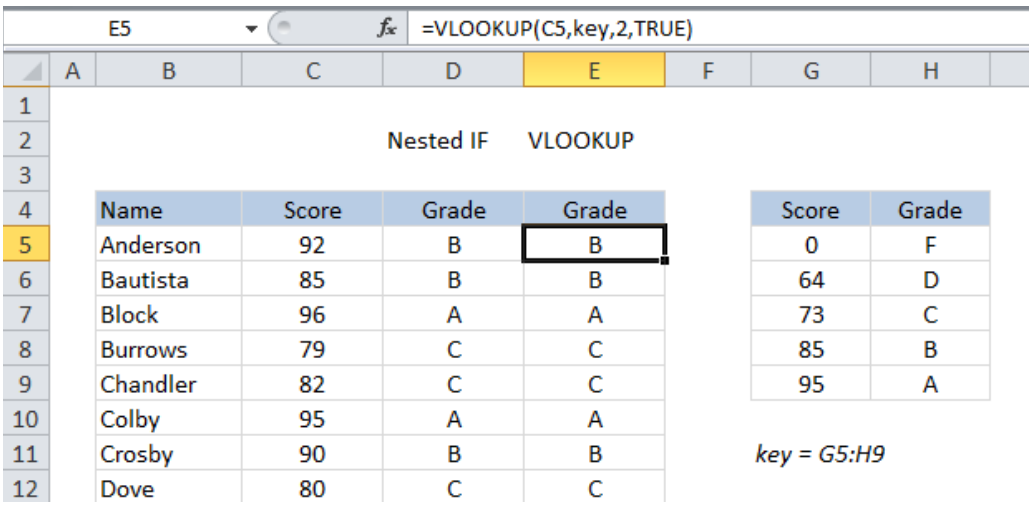
The full nested IF formula looks like this:

=IF(C5<64,"F",IF(C5<73,"D",IF(C5<85,"C",IF(C5<95,"B","A"))))

This works fine, but note that both the logic and the actual scores are baked right into the formula. If the scoring changes for any reason, you'll need to carefully update one formula then copy it down the entire table.

By contrast VLOOKUP can assign the same grades with a simple formula. All you need to do is make sure the grade key table is set up for VLOOKUP (i.e. it most be sorted by score, and contain brackets to handle all scores).

After defining a named range "key" for the grade key table, the VLOOKUP formula is very simple and generated the same grades as the original nested IFs formula:



With the grade key table named "key" we have a very simple VLOOKUP formula:

=VLOOKUP(C5,key,2,TRUE)

A nice bonus of this approach is that both the logic and the scores are built right into the grade key table. If anything changes, you can simply update the table directly and the VLOOKUP formulas will update automatically - no editing required.

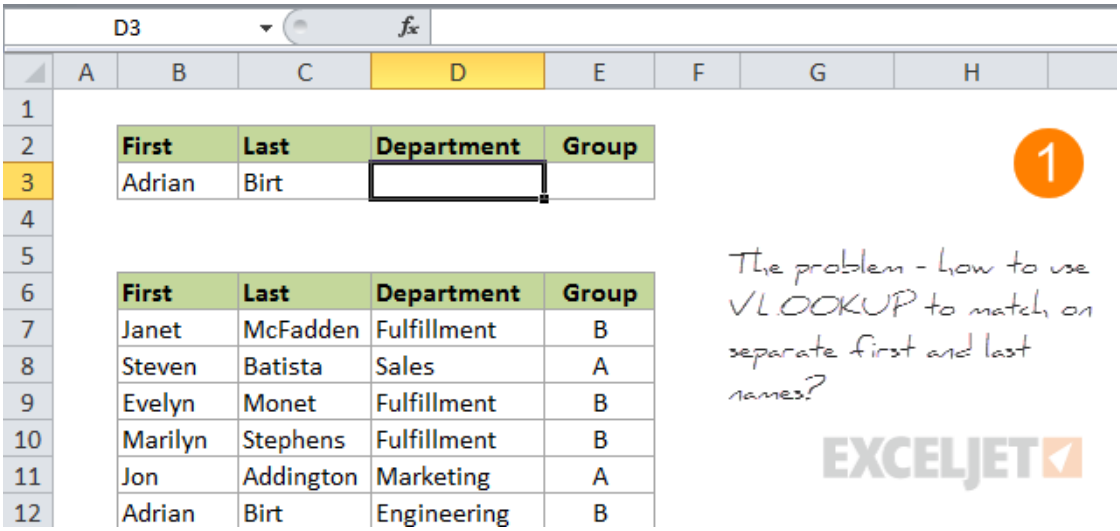
Video: How to replace nested IFs with VLOOKUP

21. VLOOKUP can only handle a single criteria

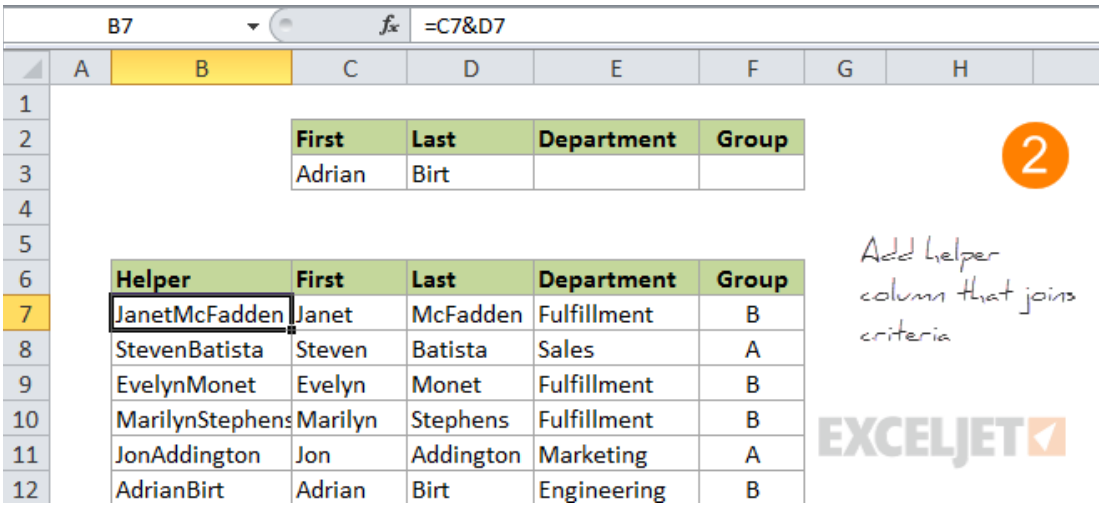
By design, VLOOKUP can only find values based on a single criteria, which is supplied as a lookup value to find in the first column of the table (the lookup column).

This means you can't easily do things like look up an employee with the last name of "Smith" in "Accounting", or look up an employee based on first and last names in separate columns.

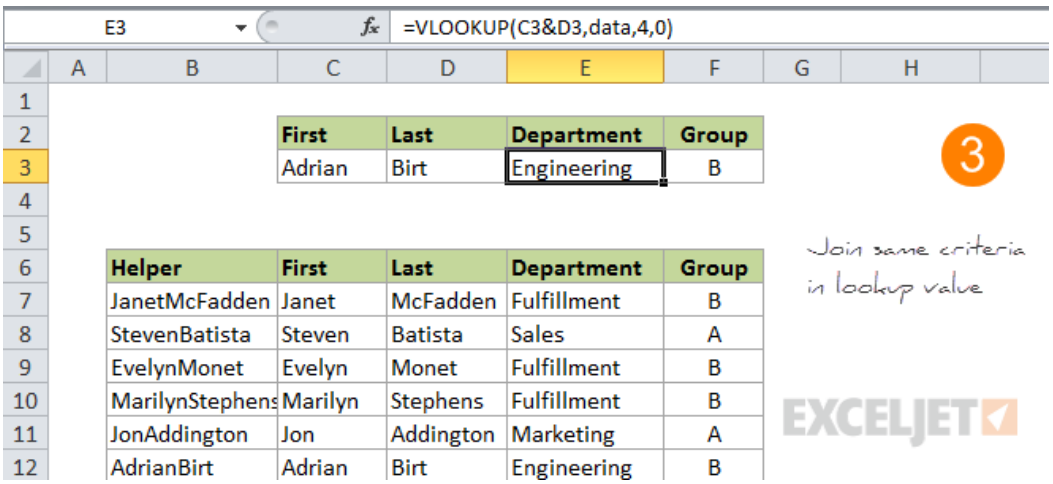
However, there are ways overcome this limitation. One workaround is to create a helper column that concatenates values from different columns to create lookup values that behave like multiple conditions. For example, here we want find the department and group for an employee, but the first name and last name appear in separate columns. How can we lookup both at once?



First, add a helper column that simply concatenates first and last names together:



Then configure VLOOKUP to use a table that includes this new column, and join first and last names for the lookup value:



The final VLOOKUP formula looks up first and last names together using the helper column as the key:

=VLOOKUP(C3&D3,data,4,0)

22. Two VLOOKUPS are faster than one VLOOKUP

It may seem completely crazy, but when you have a big set of data and need to do an exact match, you can speed up VLOOKUP a lot by adding another VLOOKUP to the formula!

The background: imagine that you have a lot of order data, say, more than 10,000 records and you are using VLOOKUP to lookup the order total based on the order id. So, you are using something like this:

=VLOOKUP(order_id,order_data, 5, FALSE)

The FALSE at the end forces VLOOKUP to do an exact match. You want an exact match because there's a chance that an order number won't be found. In this case, the exact match setting will cause VLOOKUP to return #N/A error.

The problem is that exact matches are really slow, because Excel must proceed in a linear fashion through all values until it finds a match or not.

Conversely, approximate matches are lightning fast because Excel is able to do what's called a binary search.

The problem with binary searches however (i.e. VLOOKUP in approximate match mode) is that VLOOKUP can return the wrong result when a value isn't found. Worse, the result might look completely normal, so it can be very difficult to spot.

The solution is to use VLOOKUP twice, both times in approximate match mode. The first instance simply checks that the value really exists. If so, another VLOOKUP is run (again, in approximate match mode) to fetch the data you want. If not, you can return any value you want to indicate that a result was not found.

The final formula looks like this:

=IF(VLOOKUP(order_id,order_data,1,TRUE)=order_id, VLOOKUP(order_id,order_data,5,TRUE), "Missing")

I learned this approach from Charles Williams of FastExcel, who has a fantastic, detailed article here: Why 2 VLOOKUPS are better than 1 VLOOKUP.