Commonly Used Excel Formulas

Look Up Values in a List of Data:

Let's say you want to look up an employee's phone extension by using their badge number or the correct rate of a commission for a sales amount. You look up data to quickly and efficiently find specific data in a list and to automatically verify that you are using correct data. After you look up the data, you can perform calculations or display results with the values returned. There are several ways to look up values in a list of data and to display the results. There are two Lookup functions: VLOOKUP and HLOOKUP.

When to use what:

- Use VLOOKUP when your comparison values are located in a column to the left of the data you want to find.
- Use HLOOKUP when your comparison values are located in a row across the top of a table of data, and you want to look down a specified number of rows.

1. **VLOOKUP** (Look up values vertically in a list. It searches for a value in the first column of a table array and returns a value in the same row from another column in the table array.)

   **Syntax** - VLOOKUP(lookup_value,table_array,col_index_num,range_lookup)

   **Sample** - =VLOOKUP(1,A2:C10,2,True) OR =VLOOKUP(1,A2:C10,2,False)

   **Lookup_value (Required):** The value to search in the first column of the table. Lookup_value can be a value or a reference. If lookup_value is smaller than the smallest value in the first column of table_array, VLOOKUP returns the #N/A error value.

   **Table_array (Required):** Two or more columns of data. Use a reference to a range or a range name. The values in the first column of table_array are the values searched by lookup_value. These values can be text, numbers, or logical values. Uppercase and lowercase texts are equivalent. You cannot have duplicate values in the leftmost column of the lookup range.
**Col_index_num (Required):** The column number in table_array from which the matching value must be returned. A col_index_num of 1 returns the value in the first column in table_array (in the same row); a col_index_num of 2 returns the value in the second column in table_array (in the same row), and so on. If col_index_num is:

- Less than 1, VLOOKUP returns the #VALUE! error value.
- Greater than the number of columns in table_array, VLOOKUP returns the #REF! error value.

**Range_lookup (Optional):** A logical value that specifies whether you want VLOOKUP to find an exact match or an approximate match:

- If TRUE or omitted, an exact or approximate match is returned. If an exact match is not found, the next largest value that is less than lookup_value is returned. The values in the first column of table_array must be placed in ascending sort order; otherwise, VLOOKUP may not give the correct value. You can put the values in ascending order by choosing the Sort command from the Data menu and selecting Ascending.

- If FALSE, VLOOKUP will only find an exact match. In this case, the values in the first column of table_array do not need to be sorted. If there are two or more values in the first column of table_array that match the lookup_value, the first value found is used. If an exact match is not found, the error value #N/A is returned.

**Remarks**

- When searching text values in the first column of table_array, ensure that the data in the first column of table_array does not have leading spaces, trailing spaces, inconsistent use of straight (’ or “) and curly (‘ or “) quotation marks, or nonprinting characters. In these cases, VLOOKUP may give an incorrect or unexpected value.

- When searching number or date values, ensure that the data in the first column of table_array is not stored as text values. In this case, VLOOKUP may give an incorrect or unexpected value.

- If range_lookup is FALSE and lookup_value is text, then you can use the wildcard characters, question mark (?) and asterisk (*), in lookup_value. A question mark matches any single character; an asterisk matches any sequence of characters. If you want to find an actual question mark or asterisk, type a tilde (~) preceding the character.

*See examples in worksheets “VLOOKUPValue”, “VLOOKUPMatch”, and “VLOOKUPCalculate” under the workbook named “LookUps.xlsx”.*

2. **HLOOKUP** (Look up values horizontally in a list.) Searches for a value in the top row of a table or an array of values, and then returns a value in the same column from a row you specify in the table or array.
Syntax

HLOOKUP(lookup_value, table_array, row_index_num, range_lookup)

Lookup_value (Required): The value to be found in the first row of the table. Lookup_value can be a value, a reference, or a text string.

Table_array (Required): A table of information in which data is looked up. Use a reference to a range or a range name.

- The values in the first row of table_array can be text, numbers, or logical values.
- If range_lookup is TRUE, the values in the first row of table_array must be placed in ascending order: ...-2, -1, 0, 1, 2,..., A-Z, FALSE, TRUE; otherwise, HLOOKUP may not give the correct value. If range_lookup is FALSE, table_array does not need to be sorted.
- Uppercase and lowercase texts are equivalent.
- Sort the values in ascending order, left to right.

Row_index_num (Required): The row number in table_array from which the matching value will be returned. A row_index_num of 1 returns the first row value in table_array, a row_index_num of 2 returns the second row value in table_array, and so on. If row_index_num is less than 1, HLOOKUP returns the #VALUE! error value; if row_index_num is greater than the number of rows on table_array, HLOOKUP returns the #REF! error value.

Range_lookup (Optional): A logical value that specifies whether you want HLOOKUP to find an exact match or an approximate match. If TRUE or omitted, an approximate match is returned. In other words, if an exact match is not found, the next largest value that is less than lookup_value is returned. If FALSE, HLOOKUP will find an exact match. If one is not found, the error value #N/A is returned.

Remarks

- If HLOOKUP can't find lookup_value, and range_lookup is TRUE, it uses the largest value that is less than lookup_value.
- If lookup_value is smaller than the smallest value in the first row of table_array, HLOOKUP returns the #N/A error value.
- If range_lookup is FALSE and lookup_value is text, you can use the wildcard characters, question mark (?) and asterisk (*), in lookup_value. A question mark matches any single character; an asterisk matches any sequence of characters. If you want to find an actual question mark or asterisk, type a tilde (~) before the character.

*See examples under the worksheet named “HLOOKUP”.*
Conditional Formulas:

*Examples in Conditional Formulas.xlsx workbook.*

1. Create conditional formulas
   AND, OR, NOT, and IF functions can be used to create conditional formulas to test whether conditions are true or false and making logical comparisons between expressions.

   **AND Function:** Returns TRUE if **all** its arguments are TRUE; returns FALSE if **one or more** argument is FALSE.
   **Syntax**
   \[
   \text{AND}(\text{logical1}, \text{logical2}, ...) \]
   Logical1, logical2, ... are 1 to 30 conditions you want to test that can be either TRUE or FALSE.

   **OR Function:** Returns TRUE if **any** argument is TRUE; returns FALSE if **all** arguments are FALSE.
   **Syntax**
   \[
   \text{OR}(\text{logical1}, \text{logical2}, ...) \]
   Logical1, logical2, ... are 1 to 30 conditions you want to test that can be either TRUE or FALSE.

   **NOT Function:** Reverses the value of its argument. Use NOT when you want to make sure a value is **not equal to one particular value**.
   **Syntax**
   \[
   \text{NOT}(\text{logical}) \]
   Logical is a value or expression that can be evaluated to TRUE or FALSE.
   **Remark**
   If logical is FALSE, NOT returns TRUE; if logical is TRUE, NOT returns FALSE.

   **IF Function:** Returns one value if a condition you specify evaluates to TRUE and another value if it evaluates to FALSE. Use IF to conduct conditional tests on values and formulas.
   **Syntax**
   \[
   \text{IF}(\text{logical-test}, \text{value-if-true}, \text{value-if-false}) \]
   Formula with the IF function
   1. **logical-test:** The condition that you want to check. This argument can use any comparison calculation operator (\(=\), \(>\), \(<\), \(\geq\), \(\leq\), \(<>\)).
   2. **value-if-true:** The value to return if the condition is true. Value_if_true can be another formula.
   3. **value-if-false:** The value to return if the condition is false. Value_if_false can be another formula.
When to use what Functions:

a) Create a conditional formula that results in a logical value (TRUE or FALSE)
   To do this task, use the AND, OR, and NOT functions, and operators.

b) Create a conditional formula that results in another calculation or in values other than TRUE or FALSE
   To do this task, use the IF, AND, and OR functions.

See examples under the worksheet named “Conditional”.

Note: Up to 64 IF functions can be nested as value_if_true and value_if_false arguments to construct more elaborate tests.

See examples under the worksheet named “NestedConditional”.

1. Check if a number is greater than or less than another number
   • Let's say you want to determine whether a salesperson sold more this year over last year so that you can calculate an appropriate bonus, or one warehouse contains more of an item than another warehouse in order to keep inventory levels balanced. You can simply use one of the comparison calculation operators (=, >, <, >=, =<, <>).
   • To check if a number is greater than or less than another number, use the IF function.

See examples under the worksheet named “Compare#s”.

2. Display or hide zero values
   a) Display or hide all zero values on a worksheet
      • Click the Microsoft Office Button , click Excel Options, and then click the Advanced category.
      • Under Display options for this worksheet, select a worksheet, and then do one of the following:
        o To display zero (0) values in cells, select the Show a zero in cells that have zero value check box.
        o To display zero values as blank cells, clear the Show a zero in cells that have zero value check box.

See examples under the worksheet named “ZeroValue1”.

b) Use a number format to hide zero values in selected cells
   Caution: Follow this procedure to hide zero values in selected cells. If the value in one of these cells changes to a nonzero value, the format of the value will be similar to the general number format.
   • Select the cells that contain the zero (0) values that you want to hide.
• On the Home tab, in the Cells group, click Format, and then click Format Cells.

• In the Category list, click Custom. In the Type box, type 0;-0;;@

See examples under the worksheet named “ZeroValue2”.

Notes: The hidden values appear only in the formula bar — or in the cell if you edit within the cell — and are not printed. To display hidden values again, select the cells, and then on the Home tab, in the Cells group, point to Format, and click Format Cells. In the Category list, click General to apply the default number format.

c) Use a conditional format to hide zero values returned by a formula

• Select the cell that contains the zero (0) value.
• On the Home tab, in the Styles group, click the arrow next to Conditional Formatting, point to Highlight Cells Rules, and then click Equal To.
• In the box on the left, type 0.
• In the box on the right, select Custom Format.
• In the Format Cells dialog box, click the Font tab.
• In the Color box, select white.

See examples under the worksheet named “ZeroValue3”.

d) Use a formula to display zeros as blanks or dashes

To do this task, use the IF function.

See examples under the worksheet named “ZeroValue4”.

e) Hide zero values in a PivotTable report

• Click the PivotTable report.
• On the Options tab, in the PivotTable Options group, click the arrow next to Options, and then click Options.
• Click the Layout & Format tab, then select the For empty cells, show check box. In the box, type the value that you want to display in empty cells. To display blank cells, delete any characters.

See examples under the worksheet named “PivotZeroValue” and “PivotTable”.

MORE ON COUNT FUNCTION:
COUNTIF

The COUNTIF function counts the number of cells within a range that meet a single criterion that you specify. For example, you can find out how many times a particular text or a number value occurs.

Syntax: =COUNTIF(range, criteria)

Range: Required. One or more cells to count, including numbers or names, arrays, or references that contain numbers. Blank and text values are ignored.

Criteria: Required. A number, expression, cell reference, or text string that defines which cells will be counted. For example, criteria can be expressed as 32, ">32", B4, "apples", or "32".

See examples under the worksheet named “Countif” under COUNTIF.XLSX workbook.

On the other hand, if you have more than one criterion you want to specify there are two ways to do:

1. COUNTIFS function can be used.

Syntax: COUNTIFS(criteria_range1, criteria1, [criteria_range2, criteria2]...)

criteria_range1: Required. The first range in which to evaluate the associated criteria.

criteria1: Required. The criteria in the form of a number, expression, cell reference, or text that define which cells will be counted. For example, criteria can be expressed as 32, ">32", B4, "apples", or "32".

criteria_range2, criteria2, ...: Optional. Additional ranges and their associated criteria. Up to 127 range/criteria pairs are allowed.

Important! Each additional range must have the same number of rows and columns as the criteria_range1 argument. The ranges do not have to be adjacent to each other.

2. You can use combination of COUNT and IF functions as long as you enter the formula in Array format. Array formula is formula that performs multiple calculations on one or more sets of values, and then returns either a single result or multiple results. Array formulas are enclosed between braces { } and are entered by pressing CTRL+SHIFT+ENTER. After entering your formula in a cell, press CTRL+SHIFT+ENTER. You will see your entire formula is enclosed between braces { }. If the formula is not entered as an array formula, the error #VALUE! is returned. For these formulas to work, the second argument to the IF function must be a number.

Note: Use can use wildcard search with both COUNTIF and COUNTIFS function such as “B*” to search all the cells begin with “B”. However, you CANNOT use wildcard search in the combination of COUNT and IF functions.

See examples under the worksheet named “COUNTIFS” under COUNTIF.XLSX workbook.
Financial Functions:

Getting familiar with financial terms:

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate</td>
<td>is the interest rate for the loan. Note: Divide the interest rate by 12 to get a monthly rate.</td>
</tr>
<tr>
<td>Nper</td>
<td>is the total number of payments for the loan. Note: The number of years the money is paid out is multiplied by 12 to get the total number of payments.</td>
</tr>
<tr>
<td>Pv</td>
<td>is the present value, or the total amount that a series of future payments is worth now; also known as the principal.</td>
</tr>
<tr>
<td>Fv</td>
<td>is the future value, or a cash balance you want to attain after the last payment is made. If Fv is omitted, it is assumed to be 0 (zero), that is, the future value of a loan is 0.</td>
</tr>
<tr>
<td>Type</td>
<td>is the number 0 (zero) or 1 and indicates when payments are due. 0 or omitted = at the end of the period; 1 = at the beginning of the period</td>
</tr>
<tr>
<td>PMT</td>
<td>is the payment amount (usually monthly). Shows as negative for money you are paying out (such as loan amount, amount to a savings account or investment).</td>
</tr>
</tbody>
</table>

See examples in *Financial Functions.xlsx* workbook.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Function</th>
</tr>
</thead>
</table>
| PMT(rate,nper,pv,fv,type)  
A2 (annual interest rate)  
A3 (number of months of payments)  
A4 (loan amount)  
0 (future value of the loan such as 0 when you paid off)  
1 (payment at the beginning of the period)  
=PMT(A2/12, A3, A4) OR  
=PMT(A2/12, A3, A4, 0, 1) | Calculates the payment for a loan based on constant payments and a constant interest rate.  
Example: *PMT-Loan* worksheet |

=PMT(A2/12, A3*12, 0, A4)  
A2 (annual interest rate)  
A3 (number of months for monthly savings)  
0 (present value, such as zero as of now)  
A4 (future value, such as amount to be saved in certain periods)  
=PMT(A2/12, A3*12, 0, A4) | You can also use PMT to determine payments to annuities other than loans.  
Example: *PMT-Annuity* worksheet |

| Note: | For all the arguments, cash you pay out, such as deposits to savings, is represented by negative numbers; cash you receive, such as dividend checks, is represented by positive numbers. |

CUMIPMT(rate,nper,pv,start_period,end_period,type)  
A2 (annual interest rate)  
A3 (years of the loan)  
A4 (present value)  
13 (start period, such as 13th payment)  
24 (end period, such as 24th payment)  
0 (payment at the end of the period)  
=CUMIPMT(A2/12,A3*12,A4,13,24,0) | Returns the cumulative interest paid on a loan between start period and end period.  
Example: *CUMIPMT* worksheet  
Note: You must enter either pmt type 0 or 1. |
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FV(rate,nper,pmt,pv,type)</td>
<td>Returns the future value of an investment based on periodic, constant payments and a constant interest rate.</td>
</tr>
<tr>
<td>NPER(rate, pmt, pv, fv, type)</td>
<td>Returns the number of periods for an investment based on periodic, constant payments and a constant interest rate.</td>
</tr>
<tr>
<td>PV(rate,nper,pmt,fv,type)</td>
<td>Returns the present value of an investment. The present value is the total amount that a series of future payments is worth now. For example, when you borrow money, the loan amount is the present value to the lender.</td>
</tr>
</tbody>
</table>

**Goal Seek**

Goal Seek function lets you set a formula cell to achieve a desired result by changing the value of one of the variables in the equation. Goal Seek works only with one variable input value.

**Steps:**
- Set up the formula as usual by using separate cells for each variable.
- On the Data tab, in the Data Tools group, click What-If Analysis, and then click Goal Seek.
- In the Set cell box, enter the reference for the cell that contains the formula that you want to resolve.
- In the To value box, type the formula result that you want.
- In the By changing cell box, enter the reference for the cell that contains the value that you want to adjust. Click OK.

**Note:** The cell that Goal Seek changes must be referenced by the formula in the cell that you specified in the Set cell box. See example in GoalSeek worksheet under Financial Functions.xlsx.
Scenarios

Examples in Financial Functions.xlsx workbook under Scenarios and WorstScenario worksheets.

A scenario is a set of values that Excel saves and can substitute automatically in cells on a worksheet. You can create and save different groups of values on a worksheet and then switch to any of these new scenarios to view different results.

How many scenarios can you create for a workbook? - Limited by available memory; a summary report shows only the first 251 scenarios.

For example, you want to create several different scenarios on your household budget since you see the possibility of working hours being cut back. Create a workbook that contains all the data labels, formulas and even charts you intend to use. Steps to create Scenarios:

1. Click anywhere on the worksheet.
2. On Data tab, under Data Tools Group, click the arrow next to the What-If-Analysis.
3. Select Scenario Manager.
4. Click on Add button.
5. In Add Scenario Window, give a descriptive name to your new scenario.
6. In Changing Cells box, click on red arrow to narrow down the window and select all the cells that you intend to change for your different scenarios. (For example, you may not intend to change some expense cells in your original scenario with full pay but those cells are your target cells to be changed if you take a pay cut. You want to include those cell addresses even in your original scenario though you do not need to change the values of those cells.) If changing cells are not adjacent, hold down the Control button and click those individual cells. Use Shift and Click for adjacent cells. Note: Limit to maximum numbers of 32 changing cells in a scenario. Also it is a good idea to name those changing cells to make a scenario summary report easier to read when you create one.
7. Put any explanation in Comment box why you create a particular scenario. Click OK.
8. Next, the Scenario Values box appears, showing the values for each of the cells you selected to change. Click OK to save the first scenario, with your original values, and return to the Scenario Manager. Remember you are comparing your original scenario (before the pay cut) with future scenario what-if you need to take a pay cut.
9. Go through steps 1 through 8 above to create scenarios and click OK when finished.

To display your different scenarios that you just created:

1. Open Scenario Manager by following the same steps 1 through 3 above.
2. Click on the name of the scenario that you want to display.
3. Click Show. Note: After you close the Scenario Manager dialog box, the values from the last scenario that you displayed remain on the worksheet. If you saved your initial values as a scenario, you can display those values before you close the Scenario Manager dialog box.

To merge scenarios:

1. Select the worksheet in which to store the merged scenarios results.
2. Open Scenario Manager by following the same steps 1 through 3 above under creating scenarios.
3. Click Merge.
4. In the **Merge Scenarios** dialog box, click the arrow next to Book and select a workbook that contains scenarios that you want to merge in your results.

5. In the **Sheet** box, click the name of the worksheet that contains scenarios that you want to merge.

6. Click **OK** to merge the scenarios from the selected worksheet into the current worksheet.

7. The **Merge Scenarios** dialog box closes, and the scenarios that you merged now appear in the **Scenario Manager** dialog box.

8. Repeat the preceding four steps as needed until you have merged all the scenarios that you want.

9. When you are finished, the scenarios that you merged are all part of the current worksheet. You can close the Scenario Manager Dialog box, or leave it open to continue your analysis.

To create a Scenario Summary Report:

1. On the **Data** tab, in the **Data Tools** group, click **What-If Analysis**, and then click **Scenario Manager**.

2. Click **Summary**.

3. Click **Scenario summary** or **Scenario PivotTable report**.

4. In the **Result** cells box, enter the references for the cells that refer to cells whose values are changed by the scenarios. Separate multiple references with commas.

   **Note:** Scenario reports do not automatically recalculate. If you change the values of a scenario, those changes will not show up in an existing summary report, but will show up if you create a new summary report. You don't need result cells to generate a scenario summary report, but you do need them for a scenario PivotTable report.
Date and Time Functions:

For examples, open *Date & Time Functions.xlsx* workbook.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>=NOW()</td>
<td>Returns the serial number of the current date and time</td>
</tr>
<tr>
<td>=TODAY()</td>
<td>Returns the serial number of today’s date</td>
</tr>
<tr>
<td>=DAYS360(A4,B4)</td>
<td>Calculates the number of dates between two dates based on a 360-day year (Twelve 30-days month)</td>
</tr>
<tr>
<td>A4 (start date)</td>
<td>When to use: Use this function to help compute payments if your accounting system is based on twelve 30-day months.</td>
</tr>
<tr>
<td>B4 (end date)</td>
<td></td>
</tr>
<tr>
<td>=WORKDAY(A4,B4,C4:E4)</td>
<td>Returns the serial number of the date before or after a specified number of workdays (Working days excludes weekends and any dates identified as holidays)</td>
</tr>
<tr>
<td>A4 (start date)</td>
<td>When to use: Use WORKDAY to exclude weekends or holidays when you calculate invoice due dates, expected delivery times, or the number of days of work performed.</td>
</tr>
<tr>
<td>B4 (days to complete work)</td>
<td></td>
</tr>
<tr>
<td>C4:E4 (range of cells for holidays if any)</td>
<td></td>
</tr>
<tr>
<td>=NETWORKDAYS(A4,B4,C4:E4)</td>
<td>Returns the number of whole workdays between two dates (Working days excludes weekends and any dates identified as holidays)</td>
</tr>
<tr>
<td>A4 (start date)</td>
<td>When to use: Use NETWORKDAYS to calculate employee benefits that accrue based on the number of days worked during a specific term.</td>
</tr>
<tr>
<td>B4 (end date)</td>
<td></td>
</tr>
<tr>
<td>C4:E4 (range of cells for holidays if any)</td>
<td></td>
</tr>
</tbody>
</table>

Calculating Dates & Times:

- You may compute future and past dates or count days by simply adding or subtracting numbers from current date. For example, see *ComputeDates* worksheet in *Date & Time Functions.xlsx*. Examples in *Adding Days workbook* further illustrates how to incorporate IF function to avoid Excel’s default Date setting, 1/1/1900, if formulas get copied into other cells which are set to be calculated on empty cells.

- You may also compute elapsed time in minutes and seconds between two different time periods. In *Date & Time Functions.xlsx* workbook, *Calculate Elapsed Time* worksheet explains the formatting and a simple formula on how to calculate elapsed time. Formatting the column where you want to see the calculation by using “Custom” Category is utmost important in correctly showing the elapsed time in minutes and seconds only.
## Math Functions

See examples in *Math Functions.xlsx* workbook.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>=ROUND(number,num_digits)</td>
<td>Rounds a number to a specified number of digits.</td>
<td>=ROUND(1.987654,2)</td>
<td>1.987654 (a real number you want to round) 2 (number of digits to which you want to round number) 1.99 (Answer)</td>
</tr>
<tr>
<td>=ROUNDUP(number,num_digits)</td>
<td>Rounds a number up, away from 0 (zero) and just like Round function, it is often used to reduce the number of decimal places of data and to calculate total costs when charging as a whole on partial use involved. What differs from Round function is that it always rounds the last digit up to the next highest number.</td>
<td>=ROUNDUP(76.4,0) OR =ROUNDUP(3.14159,3)</td>
<td>76.4 OR 3.14159 (real numbers you want to round up) 0 OR 3 (number of digits to which you want to round number) Answers = 77 &amp; 3.142</td>
</tr>
<tr>
<td>=ROUNDDOWN(number,num_digits)</td>
<td>ROUNDDOWN behaves like ROUND, except that it always rounds a number down. You can use this function when other ways of calculating (regular round or roundup functions) will not benefit you such as a small business owner trying to track the number of vacation hours earned by employees based on working hours.</td>
<td>=ROUNDDOWN(3.14159,3)</td>
<td>3.14159 (a real number you want to round down) 3 (number of digits to which you want to round number) Answer = 3.141</td>
</tr>
</tbody>
</table>

In Excel Number Format, using increase or decrease decimal values on the Ribbon actually just changes the 'display' value but NOT the 'real' value. To change your Excel formulas so that the displayed answers match the values behind the results, add the ROUND function to your formulas. Using the ROUND function, you can calculate your values using the same number of decimal places that you want to display in your worksheet.

Example: *Round worksheet*

Example: *RoundUpBasic worksheet and RoundComparison worksheet*

Example: *RoundDownBasic worksheet and RoundComparison worksheet*
Custom Display on numbers:

You can format fractions or numbers that contain decimal points by including digit placeholders, decimal points, and thousand separators.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (zero)</td>
<td>This digit placeholder displays insignificant zeros if a number has fewer digits than there are zeros in the format. For example, if you type 8.9, and you want it to be displayed as 8.90, use the format #.00.</td>
</tr>
<tr>
<td>#</td>
<td>This digit placeholder follows the same rules as the 0 (zero). However, Excel does not display extra zeros when the number that you type has fewer digits on either side of the decimal than there are # symbols in the format. For example, if the custom format is #.##, and you type 8.9 in the cell, the number 8.9 is displayed.</td>
</tr>
<tr>
<td>?</td>
<td>This digit placeholder follows the same rules as the 0 (zero). However, Excel adds a space for insignificant zeros on either side of the decimal point so that decimal points are aligned in the column. For example, the custom format 0.0? aligns the decimal points for the numbers 8.9 and 88.99 in a column.</td>
</tr>
<tr>
<td>. (period)</td>
<td>This digit placeholder displays the decimal point in a number.</td>
</tr>
</tbody>
</table>

If a number has more digits to the right of the decimal point than there are placeholders in the format, the number rounds to as many decimal places as there are placeholders (such as 1234.59 was displayed as 1234.6 with the code ####.#). If there are more digits to the left of the decimal point than there are placeholders, the extra digits are displayed (such as 567.9 to display as 567.900 with the code #.000). If the format contains only number signs (#) to the left of the decimal point, numbers less than 1 begin with a decimal point (such as to display .56 as .560 with the code #.000).

<table>
<thead>
<tr>
<th>To display</th>
<th>As</th>
<th>Use this code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234.59</td>
<td>1234.6</td>
<td>####.#</td>
</tr>
<tr>
<td>8.9</td>
<td>8.900</td>
<td>.000</td>
</tr>
<tr>
<td>.631</td>
<td>0.6</td>
<td>0.#</td>
</tr>
<tr>
<td>12</td>
<td>1234.568</td>
<td>#.0#</td>
</tr>
<tr>
<td>44.398</td>
<td>44.398</td>
<td>??.??</td>
</tr>
<tr>
<td>102.65</td>
<td>102.65</td>
<td>(with aligned decimals)</td>
</tr>
<tr>
<td>2.8</td>
<td>2.8</td>
<td>(with aligned decimals)</td>
</tr>
<tr>
<td>5.25</td>
<td>5 1/4</td>
<td># ??/??</td>
</tr>
<tr>
<td>5.3</td>
<td>5 3/10</td>
<td>Or you can use Fraction category under custom format</td>
</tr>
</tbody>
</table>
**Display a thousands separator:** To display a comma as a thousands separator or to scale a number by a multiple of 1,000, include the following separator in the number format.

, (comma)  Displays the thousands separator in a number. Excel separates thousands by commas if the format contains a comma that is enclosed by number signs (#) or by zeros. A comma that follows a digit placeholder scales the number by 1,000. For example, if the format is “#.0,” and you type 12,200,000 in the cell, the number 12200.0 is displayed.

<table>
<thead>
<tr>
<th>To display</th>
<th>As</th>
<th>Use this code</th>
</tr>
</thead>
<tbody>
<tr>
<td>12000</td>
<td>12,000</td>
<td>#,###</td>
</tr>
<tr>
<td>12000</td>
<td>12</td>
<td>#,</td>
</tr>
<tr>
<td>12200000</td>
<td>12.2</td>
<td>0.0,,</td>
</tr>
</tbody>
</table>

See examples in *Math Functions.xlsx* workbook under *Custom worksheet*. 