Guidelines and examples of array formulas

To become an Excel power user, you need to know how to use array formulas, which can perform calculations that you can't do by using non-array formulas. The following article is based on a series of Excel Power User columns written by Colin Wilcox and adapted from chapters 14 and 15 of Excel 2002 Formulas, a book written by John Walkenbach, an Excel MVP. To learn more about John's other books, see his book page.

In this article

- Learn about array formulas
- Learn about array constants
- Putting basic array formulas to work
- Putting advanced array formulas to work

Learn about array formulas

This section introduces array formulas and explains how to enter, edit, and troubleshoot them.

WHY USE ARRAY FORMULAS?

If you have experience using formulas in Excel, you know that you can perform some fairly sophisticated operations. For example, you can calculate the total cost of a loan over any given number of years. However, if you really want to master formulas in Excel, you need to know how to use array formulas. You can use array formulas to do complex tasks, such as:

- Count the number of characters that are contained in a range of cells.
- Sum only numbers that meet certain conditions, such as the lowest values in a range or numbers that fall between an upper and lower boundary.
• Sum every \( n \)th value in a range of values.

NOTE You may see array formulas referred to as “CSE formulas,” because you press CTRL+SHIFT+ENTER to enter them into your workbooks.

A QUICK INTRODUCTION TO ARRAYS AND ARRAY FORMULAS

If you’ve done even a little programming, you’ve probably run across the term array. For our purposes, an array is a collection of items. In Excel, those items can reside in a single row (called a one-dimensional horizontal array), a column (a one-dimensional vertical array), or multiple rows and columns (a two-dimensional array). You cannot create three-dimensional arrays or array formulas in Excel.

An array formula is a formula that can perform multiple calculations on one or more of the items in an array. Array formulas can return either multiple results or a single result. For example, you can place an array formula in a range of cells and use the array formula to calculate a column or row of subtotals. You can also place an array formula in a single cell and then calculate a single amount. An array formula that resides in multiple cells is called a multi-cell formula, and an array formula that resides in a single cell is called a single-cell formula.

The examples in the next section show you how to create multi-cell and single-cell array formulas.

TRY IT!

This exercise shows you how to use multi-cell and single-cell array formulas to calculate a set of sales figures. The first set of steps uses a multi-cell formula to calculate a set of subtotals. The second set uses a single-cell formula to calculate a grand total.

Create a multi-cell array formula
1. Open a new, blank workbook.
2. Copy the example worksheet data, and then paste it into the new workbook starting at cell A1.

   +How to copy the example worksheet data

   • Create a blank workbook or worksheet.
   • Select the example in the Help topic.
   • Press CTRL+C.
   • In the worksheet, select cell A1, and press CTRL+V.
3. Use the Paste Options button that appears nearby to match the destination formatting.

4. To multiply the values in the array (the cell range C2 through D11), select cells E2 through E11, and then enter the following formula in the formula bar:

   \[=C2:C11*D2:D11\]

5. Press CTRL+SHIFT+ENTER.

Excel surrounds the formula with braces (\{\}) and places an instance of the formula in each cell of the selected range. This happens very quickly, so what you see in column E is the total sales amount for each car type for each salesperson.
Create a single-cell array formula

1. In cell A13 of the workbook, type **Total Sales**.

2. In cell B13, type the following formula, and then press CTRL+SHIFT+ENTER:

   \[=\text{SUM(C2:C11*D2:D11)}\]

In this case, Excel multiplies the values in the array (the cell range C2 through D11) and then uses the `SUM` function to add the totals together. The result is a grand total of $111,800 in sales. This example shows how powerful this type of formula can be. For example, suppose you have 15,000 rows of data. You can sum part or all of that data by creating an array formula in a single cell.

Also, notice that the single-cell formula (in cell B13) is completely independent of the multi-cell formula (the formula in cells E2 through E11). This points to another advantage of using array formulas — flexibility. You can take any number of actions, such as changing the formulas in column E or deleting that column altogether, without affecting the single-cell formula.

Array formulas also offer these advantages:

- **Consistency**  If you click any of the cells from E2 downward, you see the same formula. That consistency can help ensure greater accuracy.

- **Safety**  You cannot overwrite a component of a multi-cell array formula. For example, click cell E3 and press DELETE. You have to either select the entire range of cells (E2 through E11) and change the formula for the entire array, or leave the array as is. As an added safety measure, you must press CTRL+SHIFT+ENTER to confirm the change to the formula.

- **Smaller file sizes**  You can often use a single array formula instead of several intermediate formulas. For example, the workbook you created for this exercise uses one array formula to calculate the results in column E. If you had used standard formulas (such as \(=\text{C2*D2}\)), you would have used 11 different formulas to calculate the same results.

A LOOK AT ARRAY FORMULA SYNTAX

For the most part, array formulas use standard formula syntax. They all begin with an equal sign, and you can use any of the built-in Excel functions in your array formulas. The key difference is that when using an array formula, you must press CTRL+SHIFT+ENTER to enter your formula. When you do this, Excel surrounds your array formula with braces — if you type the braces manually, your formula will be converted to a text string, and it will not work.
The next thing you need to understand is that array functions are a form of shorthand. For example, the multi-cell function that you used earlier is the equivalent of:

\[ \begin{align*}
C2 \times D2 \\
C3 \times D3 \\
\end{align*} \]

and so on. The single-cell formula in cell B13 condenses all of those multiplication operations, plus the arithmetic required to add those subtotals: \[ E2 + E3 + E4 \], and so on.

**RULES FOR ENTERING AND CHANGING ARRAY FORMULAS**

The primary rule for creating an array formula is worth repeating: Press CTRL+SHIFT+ENTER whenever you need to enter or edit an array formula. That rule applies to both single-cell and multi-cell formulas.

Whenever you work with multi-cell formulas, you also need to follow these rules:

- You must select the range of cells to hold your results before you enter the formula. You did this in step 3 of the multi-cell array formula exercise when you selected cells E2 through E11.
- You cannot change the contents of an individual cell in an array formula. To try this, select cell E3 in the sample workbook and press DELETE.
- You can move or delete an entire array formula, but you cannot move or delete part of it. In other words, to shrink an array formula, you first delete the existing formula and then start over.

  **TIP** To delete an array formula, select the entire formula (for example, \[ C2:C11 \times D2:D11 \]), press DELETE, and then press CTRL+SHIFT+ENTER.

- You cannot insert blank cells into or delete cells from a multi-cell array formula.

**Expanding an array formula**

At times, you may need to expand an array formula. (Remember that you cannot shrink an array formula.) The process is not complicated, but you must remember the rules listed in the previous section.

1. In the sample workbook, clear any text and single-cell formulas that are located below the main table.
2. Paste these additional lines of data into the workbook starting at cell A12. Use the Paste Options button that appears nearby to match the destination formatting.
3. Select the range of cells that contains the current array formula (E2:E11), plus the empty cells (E12:E17) that are next to the new data. In other words, select cells E2:E17.

4. Press F2 to switch to edit mode.

5. In the formula bar, change $C11$ to $C17$, change $D11$ to $D17$, and then press CTRL+SHIFT+ENTER. Excel updates the formula in cells E2 through E11 and places an instance of the formula in the new cells, E12 through E17.

---

### DISADVANTAGES OF USING ARRAY FORMULAS

Array formulas can seem magical, but they also have some disadvantages:

- You may occasionally forget to press CTRL+SHIFT+ENTER. Remember to press this key combination whenever you enter or edit an array formula.
- Other users may not understand your formulas. Array formulas are relatively undocumented, so if other people need to modify your workbooks, you should either avoid array formulas or make sure those users understand how to change them.
Depending on the processing speed and memory of your computer, large array formulas can slow down calculations.

Learn about array constants

This section introduces array constants and explains how to enter, edit, and troubleshoot them.

A BRIEF INTRODUCTION TO ARRAY CONSTANTS

Array constants are a component of array formulas. You create array constants by entering a list of items and then manually surrounding the list with braces ({}), like so:

=\{1,2,3,4,5\}

Earlier in this article, we emphasized the need to press CTRL+SHIFT+ENTER when you create array formulas. Because array constants are a component of array formulas, you surround the constants with braces manually by typing them. You then use CTRL+SHIFT+ENTER to enter the entire formula.

If you delimit (separate) the items by using commas, you create a horizontal array (a row). If you delimit the items by using semicolons, you create a vertical array (a column). To create a two-dimensional array, you delimit the items in each row by using commas, and you delimit each row by using semicolons.

As with array formulas, you can use array constants with any of the built-in functions that Excel provides. The following sections explain how to create each kind of constant and how to use these constants with functions in Excel.

Create one-dimensional and two-dimensional constants

The following procedure will give you some practice in creating horizontal, vertical, and two-dimensional constants.

Create a horizontal constant
1. Use the workbook from the previous column, or start a new workbook.
2. Select cells A1 through E1.
3. In the formula bar, enter the following formula, and then press CTRL+SHIFT+ENTER:

=\{1,2,3,4,5\}
NOTE In this case, you should type the opening and closing braces ({}).

You see the following result.

Create a vertical constant
1. In your workbook, select a column of five cells.
2. In the formula bar, enter the following formula and press CTRL+SHIFT+ENTER:

   `{1;2;3;4;5}`

You see the following result.

Create a two-dimensional constant
1. In your workbook, select a block of cells four columns wide by three rows high.
2. In the formula bar, enter the following formula, and then press CTRL+SHIFT+ENTER:

   `{1;2;3;4;5}`
You see the following result:

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>

**USE CONSTANTS IN FORMULAS**

Now that you are familiar with entering array constants, here is a simple example that uses what we've discussed:

1. Open a blank worksheet.
2. Copy the following table starting at cell A1. Use the Paste Options button that appears nearby to match the destination formatting.

   3 4 5 6 7

3. In cell A3, enter the following formula, and then press CTRL+SHIFT+ENTER:

   \[
   =\text{SUM}(A1:E1*\{1,2,3,4,5\})
   \]

Notice that Excel surrounds the constant with another set of braces, because you entered it as an array formula.
The value 85 appears in cell A3. The next section explains how the formula works.

A LOOK AT THE ARRAY CONSTANT SYNTAX

The formula you just used contains several parts.

1 Function
2 Stored array
3 Operator
4 Array constant

=SUM( A1:E1 * {1,2,3,4,5} )

The last element inside the parentheses is the array constant: \{1,2,3,4,5\}. Remember that Excel does not surround array constants with braces; you must do this. Also remember that after you add a constant to an array formula, you press CTRL+SHIFT+ENTER to enter the formula.

Because Excel performs operations on expressions enclosed in parentheses first, the next two elements that come into play are the values stored in the workbook (A1:E1) and the operator. At this point, the formula multiplies the values in the stored array by the corresponding values in the constant. It's the equivalent of:

=SUM(A1*1,B1*2,C1*3,D1*4,E1*5)

Finally, the SUM function adds the values, and the sum 85 appears in cell A3:

To avoid using the stored array and to just keep the operation entirely in memory, replace the stored array with another array constant:

=SUM((3,4,5,6,7)*\{1,2,3,4,5\})

To try this, copy the function, select a blank cell in your workbook, paste the formula into the formula bar, and then press CTRL+SHIFT+ENTER. You see the same result as you did in the earlier exercise that used the array formula =SUM(A1:E1*\{1,2,3,4,5\}).
ELEMENTS THAT YOU CAN USE IN CONSTANTS

Array constants can contain numbers, text, logical values (such as TRUE and FALSE), and error values (such as #N/A). You can use numbers in the integer, decimal, and scientific formats. If you include text, you must surround that text with double quotation marks ("').

Array constants cannot contain additional arrays, formulas, or functions. In other words, they can contain only text or numbers that are separated by commas or semicolons. Excel displays a warning message when you enter a formula such as {1,2,A1:D4} or {1,2,SUM(Q2:Z8)}. Also, numeric values cannot contain percent signs, dollar signs, commas, or parentheses.

NAMING ARRAY CONSTANTS

Possibly the best way to use array constants is to name them. Named constants can be much easier to use, and they can hide some of the complexity of your array formulas from beginning users. To name an array constant and use it in a formula, do the following:

1. On the Formulas tab, in the Defined Names group, click Define Name.

   The Define Name dialog box appears.

2. In the Name box, type Quarter1.

3. In the Refers to box, enter the following constant (remember to type the braces manually):

   ={"January","February","March"}

   The contents of the dialog box should look like this:

   ![New Name Dialog Box]

4. Click OK.
5. On the worksheet, select a row of three blank cells.

6. Type the following formula, and then press CTRL+SHIFT+ENTER.

\[=\text{Quarter1}\]

You see the following result.

When you use a named constant as an array formula, remember to enter the equal sign. If you don't, Excel interprets the array as a string of text. Finally, keep in mind that you can use combinations of text and numbers.

**TROUBLESHOOTING ARRAY CONSTANTS**

Look for the following problems when your array constants don't work:

- Some elements might not be separated with the proper character. If you omit a comma or semicolon, or if you put one in the wrong place, the array constant may not be created correctly or you may see a warning message.
- You may have selected a range of cells that doesn't match the number of elements in your constant. For example, if you select a column of six cells for use with a five-cell constant, the #N/A error value appears in the empty cell. Conversely, if you select too few cells, Excel omits the values that don't have a corresponding cell.

**ARRAY CONSTANTS IN ACTION**

The following examples demonstrate a few of the ways in which you can put array constants to use in array formulas. Some of the examples use the **TRANSPOSE** function to convert rows to columns and vice versa.

**Multiply each item in an array**

1. Select a block of empty cells four columns wide by three rows high.
2. Type the following formula, and then press CTRL+SHIFT+ENTER.
Square the items in an array

- Select a block of empty cells four columns wide by three rows high.
- Type the following array formula, and then press CTRL+SHIFT+ENTER.

\[
\begin{bmatrix}
{1,2,3,4;5,6,7,8;9,10,11,12}^2
\end{bmatrix}
\]

Alternatively, enter this array formula, which uses the caret operator (\(^\)):

\[
\begin{bmatrix}
{1,2,3,4;5,6,7,8;9,10,11,12}^2
\end{bmatrix}
\]

**Transpose a one-dimensional row**

1. Select a column of five blank cells.
2. Type the following formula, and then press CTRL+SHIFT+ENTER:

\[
\text{=TRANSPOSE}((1,2,3,4,5))
\]

Even though you entered a horizontal array constant, the \text{TRANSPOSE} function converts the array constant into a column.

**Transpose a one-dimensional column**

1. Select a row of five blank cells.
2. Enter the following formula, and then press CTRL+SHIFT+ENTER:

\[
\text{=TRANSPOSE}((1;2;3;4;5))
\]

Even though you entered a vertical array constant, the \text{TRANSPOSE} function converts the constant into a row.

**Transpose a two-dimensional constant**

1. Select a block of cells three columns wide by four rows high.
2. Enter the following constant, and press CTRL+SHIFT+ENTER.

\[
\text{=TRANSPOSE}((1,2,3,4;5,6,7,8;9,10,11,12))
\]

The \text{TRANSPOSE} function converts each row into a series of columns.
Putting basic array formulas to work

This section provides examples of basic array formulas.

GET STARTED

Use the data in this section to create two sample worksheets.

1. Open an existing workbook or create a new workbook, and make sure it contains two blank worksheets.

2. Copy the data in the following table, and paste it into the worksheet starting at cell A1.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>the quick</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1200</td>
<td>brown fox</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>jumped over</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>475</td>
<td>the lazy</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>power user</td>
<td>2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>600</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. Your finished worksheet should look like this.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>400</td>
<td>the quick</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1200</td>
<td>brown fox</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3200</td>
<td>jumped over</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>475</td>
<td>the lazy</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>500</td>
<td>power user</td>
<td>2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>600</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>800</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. Name the first worksheet **Data**, and name a second blank worksheet **Arrays**.

CREATE ARRAYS AND ARRAY CONSTANTS FROM EXISTING VALUES
Create an array from existing values
1. In your sample workbook, select the Arrays worksheet.
2. Select the cell range C1 through E3.
3. Enter the following formula in the formula bar, and then press CTRL+SHIFT+ENTER:

\[=\text{Data!E1:G3}\]

You see the following result.

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>

The formula links to the values stored in cells E1 through G3 on the Data worksheet. The alternative to this multi-cell array formula is to place a unique formula in each cell of the Arrays worksheet, as follows.

\[=\text{Data!E1}\]
\[=\text{Data!F1}\]
\[=\text{Data!G1}\]
\[=\text{Data!E2}\]
\[=\text{Data!F2}\]
\[=\text{Data!G2}\]
\[=\text{Data!E3}\]
\[=\text{Data!F3}\]
\[=\text{Data!G3}\]

If you change some of the values on the Data worksheet, those changes appear on the Arrays worksheet. Remember that to change any values on the Data worksheet, you have to follow the rules for editing array formulas. For more information about those rules, see the section Learn about array formulas.

Create an array constant from existing values
1. On the Arrays worksheet, select cells C1 through E3.
2. Press F2 to switch to edit mode.
3. Press F9 to convert the cell references to values. Excel converts the values into an array constant.
4. Press CTRL+SHIFT+ENTER to enter the array constant as an array formula.

Excel replaces the \[=\text{Data!E1:G3}\] array formula with the following array constant:
The link has been broken between the Data and Arrays worksheets, and the array formula has been replaced by an array constant.

COUNT CHARACTERS IN A RANGE OF CELLS

The following example shows you how to count the number of characters, including spaces, in a range of cells.

- On the Data worksheet, enter the following formula in cell C7, and then press CTRL+SHIFT+ENTER:

  =SUM(LEN(C1:C5))

The value 47 appears in cell C7.

In this case, the LEN function returns the length of each text string in each of the cells in the range. The SUM function then adds those values together and displays the result in the cell that contains the formula, C7.

FIND THE N SMALLEST VALUES IN A RANGE

This example shows how to find the three smallest values in a range of cells.

1. On the Data worksheet, select cells A12 through A14.

   This set of cells will hold the results returned by the array formula.

2. In the formula bar, enter the following formula, and then press CTRL+SHIFT+ENTER:

   =SMALL(A1:A10,{1;2;3})

The values 400, 475, and 500 appear in cells A12 through A14, respectively.

This formula uses an array constant to evaluate the SMALL function three times and return the smallest (1), second smallest (2), and third smallest (3) members in the array that is contained in cells A1:A10. To find more values, you add more arguments to the constant and an equivalent number of result cells to the A12:A14 range. You can also use additional functions with this formula, such as SUM or AVERAGE. For example:

  =SUM(SMALL(A1:A10,{1;2;3}))

  =AVERAGE(SMALL(A1:A10,{1;2;3}))
FIND THE N LARGEST VALUES IN A RANGE

To find the largest values in a range, you can replace the SMALL function with the LARGE function. In addition, the following example uses the ROW and INDIRECT functions.

1. On the Data worksheet, select cells A12 through A14.
2. Press DELETE to clear the existing formula but leave the cells selected.
3. In the formula bar, enter this formula, and then press CTRL+SHIFT+ENTER:

   =LARGE(A1:A10,ROW(INDIRECT("1:3")))

The values 3200, 2700, and 2000 appear in cells A12 through A14, respectively.

At this point, it may help to know a bit about the ROW and INDIRECT functions. You can use the ROW function to create an array of consecutive integers. For example, select an empty column of 10 cells in your practice workbook, enter this array formula in cells A1:A10, and then press CTRL+SHIFT+ENTER:

   =ROW(1:10)

The formula creates a column of 10 consecutive integers. To see a potential problem, insert a row above the range that contains the array formula (that is, above row 1). Excel adjusts the row references, and the formula generates integers from 2 to 11. To fix that problem, you add the INDIRECT function to the formula:

   =ROW(INDIRECT("1:10"))

The INDIRECT function uses text strings as its arguments (which is why the range 1:10 is surrounded by double quotation marks). Excel does not adjust text values when you insert rows or otherwise move the array formula. As a result, the ROW function always generates the array of integers that you want.

Let us examine the formula that you used earlier — =LARGE(A1:A10,ROW(INDIRECT("1:3"))) — starting from the inner parentheses and working outward: The INDIRECT function returns a set of text values, in this case the values 1 through 3. The ROW function in turn generates a three-cell columnar array. The LARGE function uses the values in the cell range A1:A10, and it is evaluated three times, once for each reference returned by the ROW function. The values 3200, 2700, and 2000 are returned to the three-cell columnar array. If you want to find more values, you add a greater cell range to the INDIRECT function.

Finally, you can use this formula with other functions, such as SUM and AVERAGE.

FIND THE LONGEST TEXT STRING IN A RANGE OF CELLS

This example finds the longest string of text in a range of cells. This formula works only when a data range contains a single column of cells.
On the Data worksheet, clear the existing formula from cell C7, enter the following formula in that cell, and then press CTRL+SHIFT+ENTER:

```
=INDEX(C1:C5,MATCH(MAX(LEN(C1:C5)),LEN(C1:C5),0),1)
```

The value jumped over appears in cell C7.

Let us examine the formula, starting from the inner elements and working outward. The LEN function returns the length of each of the items in the cell range C1:C5. The MAX function calculates the largest value among those items, which corresponds to the longest text string, which is in cell C3.

Here’s where things get a little complex. The MATCH function calculates the offset (the relative position) of the cell that contains the longest text string. To do that, it requires three arguments: a lookup value, a lookup array, and a match type. The MATCH function searches the lookup array for the specified lookup value. In this case, the lookup value is the longest text string:

```
(MAX(LEN(C1:C5)))
```

and that string resides in this array:

```
LEN(C1:C5)
```

The match type argument is 0. The match type can consist of a 1, 0, or -1 value. If you specify 1, MATCH returns the largest value that is less than or equal to the lookup value. If you specify 0, MATCH returns the first value exactly equal to the lookup value. If you specify -1, MATCH finds the smallest value that is greater than or equal to the specified lookup value. If you omit a match type, Excel assumes 1.

Finally, the INDEX function takes these arguments: an array, and a row and column number within that array. The cell range C1:C5 provides the array, the MATCH function provides the cell address, and the final argument (1) specifies that the value comes from the first column in the array.

For more information about the functions discussed here, see Help in Excel.

Putting advanced array formulas to work

This section provides examples of advanced array formulas.

**SUM A RANGE THAT CONTAINS ERROR VALUES**
The `SUM` function in Excel does not work when you try to sum a range that contains an error value, such as #N/A. This example shows you how to sum the values in a range named `Data` that contains errors.

```excel
=SUM(IF(ISERROR(Data),"",Data))
```

The formula creates a new array that contains the original values minus any error values. Starting from the inner functions and working outward, the `ISERROR` function searches the cell range (`Data`) for errors. The `IF` function returns a specific value if a condition you specify evaluates to TRUE and another value if it evaluates to FALSE. In this case, it returns empty strings (""`) for all error values because they evaluate to TRUE, and it returns the remaining values from the range (`Data`) because they evaluate to FALSE, meaning that they don't contain error values. The `SUM` function then calculates the total for the filtered array.

**COUNT THE NUMBER OF ERROR VALUES IN A RANGE**

This example is similar to the previous formula, but it returns the number of error values in a range named `Data` instead of filtering them out:

```excel
=SUM(IF(ISERROR(Data),1,0))
```

This formula creates an array that contains the value 1 for the cells that contain errors and the value 0 for the cells that don’t contain errors. You can simplify the formula and achieve the same result by removing the third argument for the `IF` function, like so:

```excel
=SUM(IF(ISERROR(Data),1))
```

If you don't specify the argument, the `IF` function returns FALSE if a cell does not contain an error value. You can simplify the formula even more:

```excel
=SUM(IF(ISERROR(Data)*1))
```

This version works because TRUE*1=1 and FALSE*1=0.

**SUM VALUES BASED ON CONDITIONS**

You may need to sum values based on conditions. For example, this array formula sums just the positive integers in a range named `Sales`:

```excel
=SUM(IF(Sales>0,Sales))
```

The `IF` function creates an array of positive values and false values. The `SUM` function essentially ignores the false values because 0+0=0. The cell range that you use in this formula can consist of any number of rows and columns.
You can also sum values that meet more than one condition. For example, this array formula calculates values greater than 0 and less than or equal to 5:

=SUM((Sales>0)*(Sales<=5)*(Sales))

Keep in mind that this formula returns an error if the range contains one or more non-numeric cells.

You can also create array formulas that use a type of OR condition. For example, you can sum values that are less than 5 and greater than 15:

=SUM(IF((Sales<5)+(Sales>15),Sales))

The IF function finds all values smaller than 5 and greater than 15 and then passes those values to the SUM function.

IMPORTANT You cannot use the AND and OR functions in array formulas directly because those functions return a single result, either TRUE or FALSE, and array functions require arrays of results. You can work around the problem by using the logic shown in the previous formula. In other words, you perform math operations, such as addition or multiplication, on values that meet the OR or AND condition.

COMPUTE AN AVERAGE THAT EXCLUDES ZEROS

This example shows you how to remove zeros from a range when you need to average the values in that range. The formula uses a data range named Sales:

=AVERAGE(IF(Sales<>0,Sales))

The IF function creates an array of values that do not equal 0 and then passes those values to the AVERAGE function.

COUNT THE NUMBER OF DIFFERENCES BETWEEN TWO RANGES OF CELLS

This array formula compares the values in two ranges of cells named MyData and YourData and returns the number of differences between the two. If the contents of the two ranges are identical, the formula returns 0. To use this formula, the cell ranges must be the same size and of the same dimension:

=SUM(IF(MyData=YourData,0,1))

The formula creates a new array of the same size as the ranges that you are comparing. The IF function fills the array with the value 0 and the value 1 (0 for mismatches and 1 for identical cells). The SUM function then returns the sum of the values in the array.

You can simplify the formula like this:
=SUM(1*(MyData<>YourData))

Like the formula that counts error values in a range, this formula works because TRUE*1=1, and FALSE*1=0.

FIND THE LOCATION OF THE MAXIMUM VALUE IN A RANGE

This array formula returns the row number of the maximum value in a single-column range named Data:

=MIN(IF(Data=MAX(Data),ROW(Data),""))

The IF function creates a new array that corresponds to the Data range. If a corresponding cell contains the maximum value in the range, the array contains the row number. Otherwise, the array contains an empty string (""). The MIN function uses the new array as its second argument and returns the smallest value, which corresponds to the row number of the maximum value in Data. If the Data range contains identical maximum values, the formula returns the row of the first value.

If you want to return the actual cell address of a maximum value, use this formula:

=ADDRESS(MIN(IF(Data=MAX(Data),ROW(Data),"")),COLUMN(Data))