

WELCOME TO



Data Analytics with Excel

Formulas Tab/Auditing Tools

AUDITING TOOLS:

Trace Precedents/Dependents shows which cells **affect** or **are affected by** the value of the selected cell

Show Formulas displays all of the formulas in the sheet as text

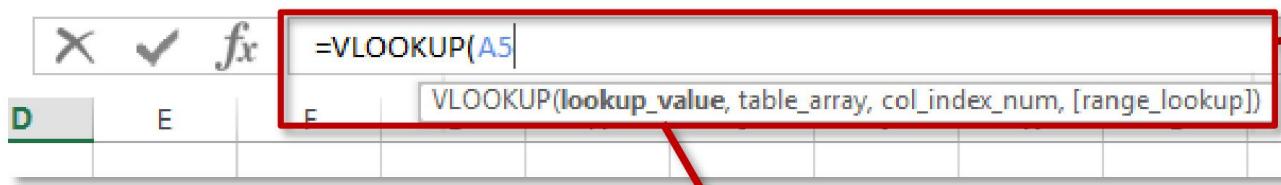
Evaluate Formula allows you to step into a formula and determine the output of each component

FORMULA LIBRARY:

Includes a list of all common formulas, component parts, and brief descriptions of how each formula works



Formula Syntax



As you begin to type a formula, a pop-up will appear to guide you through each step, shown in bold

All Excel formulas start with a “=“ and can either be selected from the formula library or typed directly into the formula bar

A1

Single-cell references describe a cell’s location within a worksheet, in terms of the intersection between a column (A through XFD), and a row (1 through 1,048,576)

A	B	C
1		
2		
3		
4		
5		

A1:C4

Array references describe a contiguous group of cells based on the location of the top-left (A1) and the bottom-right (C4) cells, separated by a “:”

A	B	C
1		
2		
3		
4		

A1,C4

Non-contiguous references describe selections of individual cells that do not share a common border, separated by a “,”

A	B	C
1		
2		
3		
4		

Hold the phone, how come some cell references include a “\$”?

These are used to create **Fixed, Relative, or Mixed References**; the \$ basically locks a specific cell range or reference so that it does not change if you apply the formula to other cells

For Example:

- \$A\$1 = Fixed column, Fixed row
- A\$1 = Relative column, Fixed row
- \$A1 = Fixed column, Relative row
- A1 = Relative column, Relative row

PRO TIP:

Select part of your formula with the cursor and use “F4” to quickly scroll through reference types. ALWAYS THINK ABOUT YOUR REFERENCES



	A	B	C
1	\$A\$1		
2			
3			
4			\$A\$1

	A	B	C
1	A\$1		
2			
3			
4			C\$1

	A	B	C
1	\$A1		
2			
3			
4			\$A4

	A	B	C
1	A1		
2			
3			
4			C4

Common Excel Errors

Error Type	What it means	How to fix it
#####	<i>Column isn't wide enough to display values</i>	<i>Drag or double-click column border to increase width, or right-click to set custom column width</i>
#NAME?	<i>Excel does not recognize text in a formula</i>	<i>Make sure that function names are correct, references are valid and spelled properly, and quotation marks and colons are in place</i>
#VALUE!	<i>Formula has the wrong type of argument</i>	<i>Check that your formula isn't trying to perform an arithmetic operation on text strings or cells formatted as text</i>
#DIV/0!	<i>Formula is dividing by zero or an empty cell</i>	<i>Check the value of your divisor; if 0 is correct, use an IF statement to display an alternate value if you choose</i>
#REF!	<i>Formula refers to a cell that is not valid</i>	<i>Make sure that you didn't move, delete, or replace cells that are referenced in your formula</i>



IFERROR Statement

The **IFERROR** statement is an excellent tool to eliminate annoying error messages (#N/A, #DIV/0!, #REF!, etc.), which is particularly useful for front-end formatting

=IFERROR(value, value_if_error)

Formula or value (which may or may not result in an error)

Value returned in the case of an error

In this case we're replacing an error caused by the A1/B1 formula with "Invalid Formula", and an error caused by a VLOOKUP function with "-"

{ =IFERROR(A1/B1,"Invalid Formula")
=IFERROR(VLOOKUP(A1,D1:E4,2,0),"")

PRO TIP:

If you're writing a formula that may trigger an error (i.e. a VLOOKUP where not all values have a match), WRITE THE FULL FORMULA FIRST then wrap it in an IFERROR statement



The **F4** function is used for two helpful shortcuts:

1) Adding or modifying cell reference types

*With your cursor selecting any cell reference or array within a formula, the **F4** key will cycle through fixed, relative, and absolute reference types*



2) Repeating your last command or action

F4 will also repeat the last user action, such as inserting/deleting rows or columns, changing cell format or style, etc. (Note: F4 will not repeat entered values or formulas)

The **F2** function displays the cell ranges that are tied to a given formula

PRO TIP:

*Use **F2** to help diagnose formula errors or make quick adjustments to cell references and arrays*



The **CTRL** function can be combined with a variety of keys, such as:

1) CTRL-ARROW

*Jumps to the left, right, top, or bottom edge
(i.e. last non-blank cell) of a contiguous data array*

2) CTRL-SHIFT-ARROW

Extends a selection to the left, right, top, or bottom edge (i.e. last non-blank cell) of a data array

3) CTRL-PAGE UP/DOWN

Jumps between tabs of a workbook

The screenshot shows three separate Excel spreadsheets demonstrating the use of the **CTRL-SHIFT-RIGHT ARROW** and **CTRL-SHIFT-DOWN ARROW** keys.

- CTRL-SHIFT-RIGHT ARROW:** The first spreadsheet shows a 12x8 grid of data. A selection is made from cell A1 to cell A12. Using **CTRL-SHIFT-RIGHT ARROW** extends the selection to cover all columns from A to H, resulting in a 12x8 selection.
- CTRL-SHIFT-DOWN ARROW:** The second spreadsheet shows a 12x8 grid of data. A selection is made from cell A1 to cell A11. Using **CTRL-SHIFT-DOWN ARROW** extends the selection to cover all rows from 1 to 12, resulting in an 11x8 selection.
- CTRL-PAGE UP/DOWN:** The third spreadsheet shows a 12x8 grid of data. A selection is made from cell A1 to cell A11. Using **CTRL-PAGE DOWN** extends the selection to cover all rows from 1 to 12, resulting in an 11x8 selection.

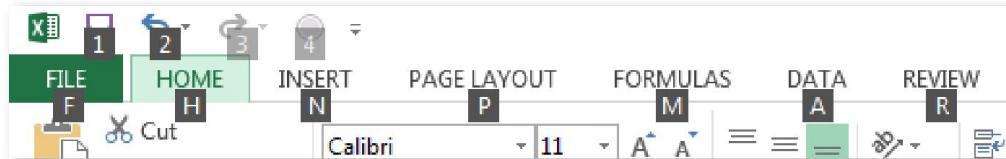
FULL LIST:

<http://office.microsoft.com/en-us/excel-help/excel-shortcut-and-function-keys-HP010073848.aspx>

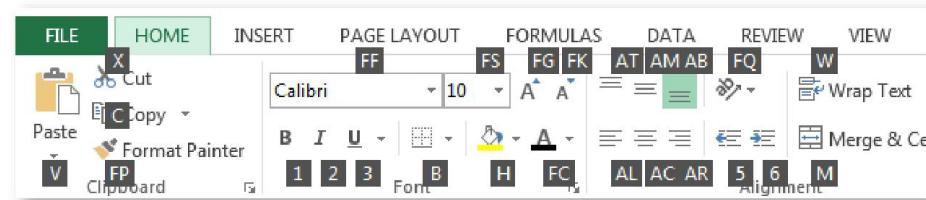


The **ALT** function enables **Key Tips**, which allow you to access any function in the ribbon using keyboard shortcuts (*Note: you do not need to hold down ALT*)

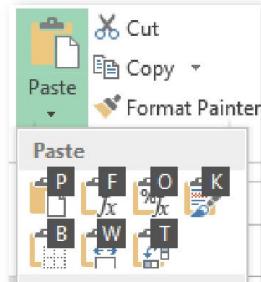
1) Press ALT to reveal tab-level shortcuts:



2) Press the key for the tab you want (i.e. H) to reveal additional shortcuts:



3) Continue to press shortcut keys (i.e. V) to drill into specific functions:



PRO TIP:

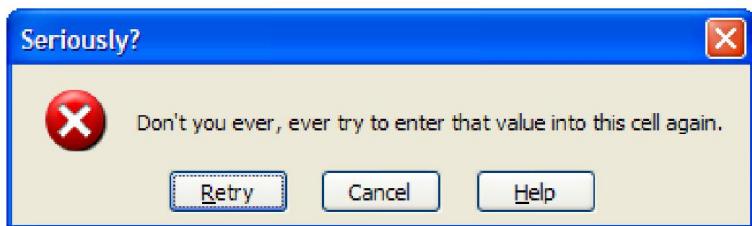
Use **ALT-H-V-V** to paste as values
or **ALT-H-V-F** to paste as formulas

Data Validation

Data Validation allows you to specify exactly what types of values a cell can contain (i.e. whole numbers, positive integers, values from a list, etc.)

One of the most useful forms of data validation is **LIST**, which creates a drop-down menu of options based on a source list that you specify:

(but the best part is that you can write your own hilarious error messages) See, Excel can be fun!



The 'Data Validation' dialog box shows the 'Settings' tab selected. Under 'Validation criteria', the 'Allow' dropdown is set to 'List'. The 'Source' field contains the formula '=L\$4:\$L\$9'. The 'In-cell dropdown' checkbox is checked. A red box highlights the 'Allow' section. To the right, a screenshot shows a dropdown menu with items A, B, C, D, E, F, where 'D' is selected. A red box highlights the dropdown and another red arrow points to the list of items.

Fixed vs. Volatile Functions

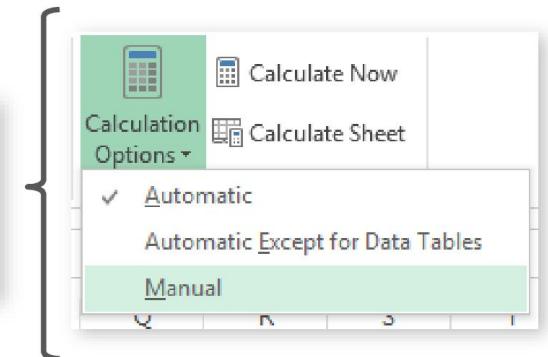
Volatile Functions are functions or formulas in Excel that change every time the workbook recalculates (i.e. any time you enter data anywhere in any open workbook)

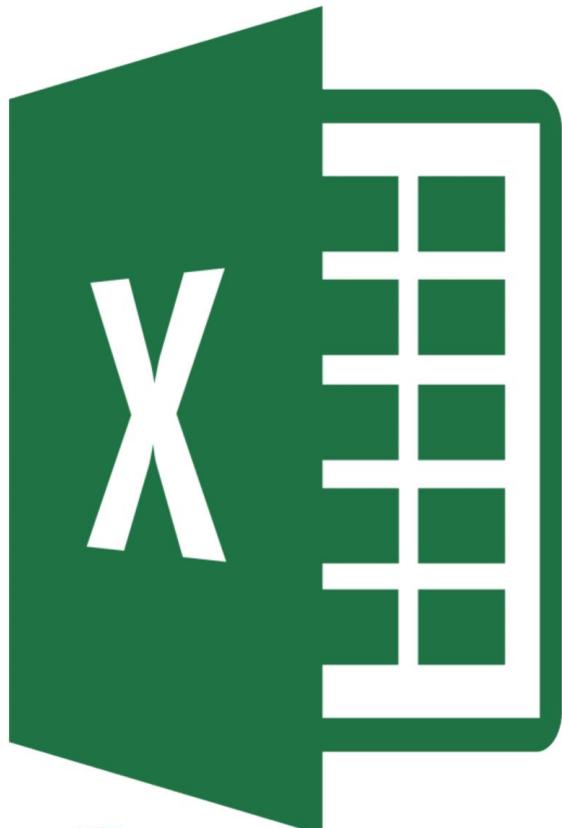


Handle with Care: Common volatile functions include **NOW()**, **TODAY()**, **RAND()**, **OFFSET()** & **INDIRECT()**

PRO TIP:

To control when Excel recalculates, change the Calculation Options to “Manual” in the Formulas tab (just don’t forget you changed it!)





All **Logical Operators** in Excel are based on simple “IF/THEN” statements:

- IF it's raining, THEN bring an umbrella*
- IF it's sunny, THEN bring sunglasses*
- IF it's sunny AND it's summer, skip work and go to the beach*

Basically it just says “*Hey Excel, if this statement is true, return this value. Otherwise, return something else.*”

=IF(logical_test, [Value if True], [Value if False])

Any test that results in either
TRUE or **FALSE**
(i.e. A1="Google", B2<100, etc)

Value returned if logical
test is **TRUE**

Value returned if logical
test is **FALSE**

	A	B	C	D
1	Location	Temp (F)	Precip (mm)	Freeze
2	A	75	0	No
3	B	18	0	Yes
4	C	86	0	No
5	D	80	2.3	No
6	E	28	1.2	Yes
7	F	68	0.5	No
8	G	26	0	Yes

= IF(B2<=0, "Yes", "No")

*In this case we're categorizing the Freeze column
as "Yes" if the temperature is equal to or below 32,
otherwise "No"*

Nested IF Statements

By using **Nested IF Statements**, you can include multiple logical tests within a single formula:

	A	B	C	D	E
1	Location	Temp (F)	Precip (mm)	Freeze	Climate
2	A	75	0	No	Mild
3	B	18	0	Yes	Cold
4	C	86	0	No	Hot
5	D	80	2.3	No	Mild
6	E	28	1.2	Yes	Cold
7	F	68	0.5	No	Mild
8	G	26	0	Yes	Cold

= IF(B2<40,"COLD",IF(B2>80,"HOT","MILD"))

If temp<40, climate = "Cold", if temp>80, climate = "Hot", otherwise climate = "Mild"

AND / OR Statements

Excel's **AND** and **OR** statements allow you to include multiple logical tests at once:

A	B	C	D	E	F	G	
1	Location	Temp (F)	Precip (mm)	Freeze	Climate	Precip Type	Conditions
2	A	75	0	No	Mild	None	Dry
3	B	18	0	Yes	Cold	None	Dry
4	C	86	0	No	Hot	None	Dry
5	D	80	2.3	No	Mild	Rain	Wet
6	E	28	1.2	Yes	Cold	Snow	Wet
7	F	68	0.5	No	Mild	Rain	Wet
8	G	26	0	Yes	Cold	None	Dry

=IF(OR(F2="Rain", F2="Snow"), "Wet", "Dry")

Here we're categorizing conditions as "Wet" if the precipitation type equals "rain" OR "snow", otherwise Conditions = "Dry"

=IF(AND(D2="Yes", C2>0), "Snow", IF(AND(D2="No", C2>0), "Rain", "None"))

If the temp is below freezing AND the amount of precipitation > 0, then Precip Type = "Snow", if the temp is above freezing AND the amount of precipitation >0, then Precip Type = "Rain", otherwise Precip Type = "None"

PRO TIP:

When writing nested functions, copy/paste repetitive pieces and tweak individual elements to save time (rather than starting from scratch)



NOT / <> Operators

If you want to evaluate a case where a logical statement is *not* true, you can use either the **NOT** statement or a “<>” operator

	A	B	C	D	E	F	G
1	Location	Temp (F)	Precip (mm)	Freeze	Climate	Precip Type	Conditions
2	A	75	0	No	Mild	None	Dry
3	B	18	0	Yes	Cold	None	Dry
4	C	86	0	No	Hot	None	Dry
5	D	80	2.3	No	Mild	Rain	Wet
6	E	28	1.2	Yes	Cold	Snow	Wet
7	F	68	0.5	No	Mild	Rain	Wet
8	G	26	0	Yes	Cold	None	Dry

=IF(NOT(C2=0),“Wet”,“Dry”)

=IF(C2<>0,“Wet”,“Dry”)

In both of these examples, we’re defining **Conditions = “Wet” if the amount of precipitation is NOT equal to 0**

The **IFERROR** statement is an excellent tool to eliminate annoying error messages (#N/A, #DIV/0!, #REF!, etc.), which is particularly useful for front-end formatting

=IFERROR(value, value_if_error)



Formula or value that may or may not result in an error



Value returned in the case of an error

PRO TIP:

If you're writing a formula that may trigger an error (i.e. a VLOOKUP where not all values have a match), WRITE THE FULL FORMULA FIRST then wrap it in an IFERROR statement

Excel offers a number of different **IS** formulas, each of which checks whether a certain condition is true:

ISBLANK = Checks whether the reference cell or value is blank

ISNUMBER = Checks whether the reference cell or value is numerical

ISTEXT = Checks whether the reference cell or value is a text string

ISERROR = Checks whether the reference cell or value returns an error

ISEVEN = Checks whether the reference cell or value is even

ISODD = Checks whether the reference cell or value is odd

ISLOGICAL = Checks whether the reference cell or value is a logical operator

ISFORMULA = Checks whether the reference cell or value is a formula

The Count, Average, Median, Mode, Max/Min, Percentile and Standard Deviation/Variance functions are used to perform basic calculations on a data array

	A	B	C	D	
1		Value			
2	90	Sample Size	19		=COUNT(A2:A20)
3	13				
4	22	Average:	51.47		=AVERAGE(A2:A20)
5	98				
6	61	Median:	54		=MEDIAN(A2:A20)
7	68				
8	50	Mode:	22		=MODE(A2:A20)
9	91				
10	16	Max:	98		=MAX(A2:A20)
11	23				
12	60	Min:	13		=MIN(A2:A20)
13	22				
14	56	25th Percentile	23		=PERCENTILE(A2:A20,.25)
15	54				
16	87	75th Percentile	68		=PERCENTILE(A2:A20,.75)
17	33				
18	68	Standard Deviation	28		=STDEV(A2:A20)
19	45				
20	21	Variance	767		=VAR(A2:A20)
21					

RANK / SMALL / LARGE

	A
1	Value
2	90
3	13
4	22
5	98
6	61
7	68
8	50

→ RANK(A2,A2:A8) = 2
→ RANK(A3,A2:A8) = 7 (lowest)
→ RANK(A4,A2:A8) = 6
→ RANK(A5,A2:A8) = 1 (highest)
→ RANK(A6,A2:A8) = 4
→ RANK(A7,A2:A8) = 3
→ RANK(A8,A2:A8) = 5

The **SMALL/LARGE** functions return the n^{th} smallest/largest values within an array

The **RANK** function returns the rank of a particular number among a list of values

	A
1	Value
2	90
3	13
4	22
5	98
6	61
7	68
8	50

LARGE(A2:A8,2) = 90
(the 2nd largest number in the array is 90)

SMALL(A2:A8,3) = 50
(the 3rd smallest number in the array is 50)

PERCENTRANK

	A	B
1	Value	Percent Rank
2	2,717	18%
3	3,485	24%
4	5,202	76%
5	3,612	29%
6	4,432	59%
7	2,699	12%
8	4,585	65%
9	6,003	94%
10	4,820	71%
11	2,550	6%
12	5,795	88%
13	4,240	41%
14	6,827	100%
15	4,359	53%
16	2,320	0%
17	5,775	82%
18	4,241	47%
19	3,966	35%

PERCENTRANK returns the rank of a value as a percentage of a given array or dataset

=PERCENTRANK(array, x)

What range of data
are you looking at?

Which **value** within the
range are you looking at?

PERCENTRANK(\$A\$2:\$A\$19, A14) = 100% (highest)

PERCENTRANK(\$A\$2:\$A\$19, A16) = 0% (lowest)

RAND() and **RANDBETWEEN** act like random number generators in Excel:

	A	B	C	D	E
1	0.5173	0.4091	0.7560	0.9012	0.2167
2	0.0906	0.2317	0.0906	0.5856	0.8646
3	0.1544	0.8240	0.4279	0.8782	0.7795
4	0.0097	0.0872	0.7740	0.9137	0.7815
5	0.2089	0.7028	0.0449	0.8173	0.9983
6	0.0761	0.4388	0.4056	0.5639	0.0668

The **RANDBETWEEN** function returns an integer between two values that you specify



The **RAND()** function returns a random value between 0 and 1 (to 15 digits)

	A	B	C	D	E
1	83	23	64	62	92
2	59	45	40	50	91
3	24	37	70	30	32
4	54	85	69	55	3
5	73	12	36	53	2
6	29	72	68	59	99

=RANDBETWEEN(0,100)

SUM PRODUCT

The **SUMPRODUCT** formula multiplies corresponding cells from multiple arrays and returns the sum of the products (*Note: all arrays must have the same dimensions*)

=SUMPRODUCT(array1, array2 ... array_N)

Example: Total Revenue

	A	B	C	D
1	Product	Quantity	Price	Revenue
2	Apple	2	\$0.50	\$1.00
3	Banana	4	\$1.00	\$4.00
4	Orange	3	\$0.80	\$2.40
5	Total			\$7.40

	A	B	C	D
1	Product	Quantity	Price	Revenue
2	Apple	2	\$0.50	
3	Banana	4	\$1.00	
4	Orange	3	\$0.80	
5	Total			\$7.40

Without using SUMPRODUCT, you could multiply quantity*price in each row and sum the products

SUMPRODUCT(B2:B4,C2:C4) = \$7.40

SUMPRODUCT is often used with filters to calculate products *only* for rows that meet certain criteria:

	A	B	C	D
1	Store	Product	Quantity	Price
2	Stop & Shop	Apple	2	\$0.50
3	Shaws	Banana	4	\$1.00
4	Market Basket	Banana	3	\$1.00
5	Trader Joe's	Pineapple	8	\$2.50
6	Stop & Shop	Orange	2	\$0.80
7	Shaws	Apple	1	\$0.50
8	Market Basket	Apple	5	\$0.50
9	Trader Joe's	Banana	6	\$1.00
10	Market Basket	Pineapple	3	\$2.50
11	Trader Joe's	Orange	8	\$0.80
12	Stop & Shop	Pineapple	3	\$2.50
13	Shaws	Pineapple	5	\$2.50
14	Stop & Shop	Banana	2	\$1.00
15	Shaws	Orange	6	\$0.80
16	Market Basket	Orange	7	\$0.80
17	Trader Joe's	Apple	3	\$0.50

Quantity of goods sold at Shaws:

SUMPRODUCT((A2:A17="Shaws")*C2:C17) = 16

Total revenue from Shaws:

SUMPRODUCT((A2:A17="Shaws")*C2:C17*D2:D17) = \$21.80

Revenue from apples sold at Shaws:

SUMPRODUCT((A2:A17="Shaws")*(B2:B17="Apple")*C2:C17*D2:D17) = \$0.50

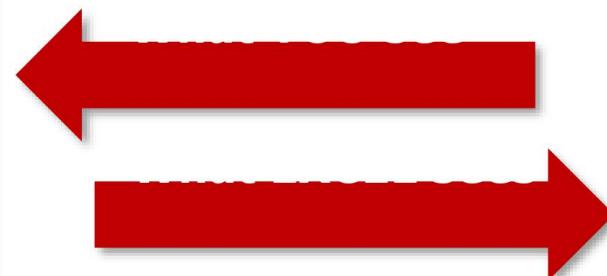
PRO TIP:

When you add filters to a SUMPRODUCT, you need to change the commas to multiplication signs

Great, but how does it *really* work?

SUMPRODUCT((A2:A17="Shaws")*(B2:B17="Apple")*C2:C17*D2:D17) = \$0.50

	A	B	C	D
1	Store	Product	Quantity	Price
2	Stop & Shop	Apple	2	\$0.50
3	Shaws	Banana	4	\$1.00
4	Market Basket	Banana	3	\$1.00
5	Trader Joe's	Pineapple	8	\$2.50
6	Stop & Shop	Orange	2	\$0.80
7	Shaws	Apple	1	\$0.50
8	Market Basket	Apple	5	\$0.50
9	Trader Joe's	Banana	6	\$1.00
10	Market Basket	Pineapple	3	\$2.50
11	Trader Joe's	Orange	8	\$0.80
12	Stop & Shop	Pineapple	3	\$2.50
13	Shaws	Pineapple	5	\$2.50
14	Stop & Shop	Banana	2	\$1.00
15	Shaws	Orange	6	\$0.80
16	Market Basket	Orange	7	\$0.80
17	Trader Joe's	Apple	3	\$0.50



When you apply a condition or filter to a column, Excel translates those cells as 0's (if false) and 1's (if true)

	A	B	C	D
1	Store	Product	Quantity	Price
2	0	1	2	\$0.50
3	1	0	4	\$1.00
4	0	0	3	\$1.00
5	0	0	8	\$2.50
6	0	0	2	\$0.80
7	1	1	1	\$0.50
8	0	1	5	\$0.50
9	0	0	6	\$1.00
10	0	0	3	\$2.50
11	0	0	8	\$0.80
12	0	0	3	\$2.50
13	1	0	5	\$2.50
14	0	0	2	\$1.00
15	1	0	6	\$0.80
16	0	0	7	\$0.80
17	0	1	3	\$0.50

If you multiply all four columns, ONLY ROWS THAT SATISFY ALL CONDITIONS WILL PRODUCE A NON-ZERO SUM

The **COUNTIF**, **SUMIF**, and **AVERAGEIF** formulas calculate a sum, count, or average based on specific criteria

	A	B
1	Name	Age
2	George	90
3	Maria	13
4	Ryan	22
5	Tim	98
6	George	61
7	Tim	68
8	Tim	50
9	Maria	91
10	George	16
11	Maria	23
12	Tim	60
13	Ryan	22
14	Maria	56
15	George	54
16	George	87
17	Ryan	33
18	Ryan	68
19	Ryan	45
20	George	21

=COUNTIF(range, criteria)

=SUMIF(range, criteria, sum_range)

=AVERAGEIF(range, criteria, average_range)

Which cells need to
match your criteria?

Under what condition
do I want to sum, count,
or aver age?

Where are the values
that I want to sum
or average?

COUNTIF(B2:B20,22) = 2

SUMIF(A2:A20,"Ryan",B2:B20) = 190

SUMIF(A2:A20,"<>Tim",B2:B20) = 702

AVERAGEIF(A2:A20,"Maria",B2:B20) = 45.75



COUNTIFS / SUMIFS / AVERAGEIFS

COUNTIFS, SUMIFS, and AVERAGEIFS are used when you want to evaluate a count, sum, or average based on *multiple* conditions or criteria

=COUNTIFS(criteria_range1, criteria1, criteria_range2 , criteria2...)

=SUMIFS(sum_range, criteria_range1, criteria1, criteria_range2 , criteria2...)

=AVERAGEIFS(average_range, criteria_range1, criteria1, criteria_range2 , criteria2...)

	A	B	C	D
1	Month	Tactic	Campaign	Clicks
2	Jan	Search	Google	166
3	Jan	Search	MSN	263
4	Jan	Display	Contextual	289
5	Jan	Display	Retargeting	137
6	Feb	Search	Google	124
7	Feb	Search	MSN	311
8	Feb	Display	Contextual	350
9	Feb	Display	Retargeting	384
10	Mar	Search	Google	168
11	Mar	Search	MSN	358
12	Mar	Display	Contextual	347
13	Mar	Display	Retargeting	390

} COUNTIFS(B2:B13,"Search", D2:D13,>200) =
3 SUMIFS(D2:D13,
A2:A13,"Feb",B2:B13,"Display") = 734
AVERAGEIFS(D2:D13,
A2:A13,"Jan",C2:C13,"MSN") = 263

PRO TIP:

If you use <or >, you need to add quotation marks as you would with text (i.e. ">200")

Using **Named Arrays** can simplify a lookup function if you use the same data array in multiple formulas

For example, if you name the array from A1:D6 “Apparel”...

...you can write your vlookup formula in either of the following ways:

```
=VLOOKUP(A1,$A$1:$D$6,2)  
=VLOOKUP(A1,Apparel,2)
```

Let's take a look at one of Excel's most common reference functions – **VLOOKUP**:

=VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])

This is the **value** that you are trying to match in the table array

This is **where** you are looking for the lookup value

Which column contains the data you're looking for?

Are you trying to match the **exact** lookup value (0), or something similar (1)?

	A	B	C	D
1	Product	Quantity	Product ID	Price
2	T-shirt	26	93754	\$14.99
3	Sweater	14	24783	\$49.99
4	Shorts	22	23984	\$24.50
5	Socks	36	58394	\$9.99
6	Spandex Unitard	2	27838	\$79.99

D2=VLOOKUP(A2, \$G\$1:\$H\$5, 2, 0)

G	H
Product	Price
Shorts	\$24.50
Sweater	\$49.99
Spandex Unitard	\$79.99
T-shirt	\$14.99
Socks	\$9.99

To populate the Price in column D, we look up the name of the product in the data array from G1:H5 and return the value from the 2nd column over

HLOOKUP

Use **HLOOKUP** if your table array is transposed (variables headers listed in rows)

=HLOOKUP(lookup_value, table_array, row_index_num, [range_lookup])

This is the **value** that you are trying to match in the table array

This is **where** you are looking for the lookup value

Which column contains the data you're looking for?

Are you trying to match the **exact** lookup value (0), or something similar (1)?

	A	B	C	D
1	Product	Quantity	Product ID	Price
2	T-shirt	26	93754	\$14.99
3	Sweater	14	24783	\$49.99
4	Shorts	22	23984	\$24.50
5	Socks	36	58394	\$9.99
6	Spandex Unitard	2	27888	\$79.99

D2=HLOOKUP(A2, \$H\$1:\$L\$2, 2, 0)

With an HLOOKUP, we search for the product name in H1:L2 and return the value from the 2nd row down

G	H	I	J	K	L
Product	Shorts	T-shirt	Sweater	Spandex Unitard	Socks
Price	\$24.50	\$14.99	\$49.99	\$79.99	\$9.99

There are **two key rules** that constrain **VLOOKUP** and **HLOOKUP** formulas:

1. The lookup value must be in the **first column** of a VLOOKUP table array or the **first row** of a HLOOKUP table array
2. Excel will always return the value from the **top most row** or **left most column** of a table array when multiple instances of the lookup value are present



PRO TIP:



Avoid breaking Law #2 by identifying a “Key” that is common to both datasets and is unique for every row (NOTE: Keys often take the form of a concatenation of multiple fields)

The **ROW** function returns the row number of a given *reference*, while the **ROWS** function returns the number of rows in a given *array or array formula*

=ROW([reference])

=ROWS(array)

This example uses an array, which is why it includes the fancy {} signs – more on that in the ARRAY functions section

ROW(C10) = 10

ROWS(A10:D15) = 6

ROWS({1,2,3;4,5,6}) = 2

The **COLUMN** function returns the column number of a given *reference*, while the **COLUMNS** function returns the number of columns in a given *array* or *array formula*

=COLUMN([reference])

=COLUMNS(array)

PRO TIP:

Leave the cell reference out and just write ROW() or COLUMN() to return the row or column number of the cell in which the formula is written

COLUMN(C10) = 3

COLUMNS(A10:D15) = 4

COLUMNS({1,2,3;4,5,6}) = 3

The **INDEX** function returns the *value* of a specific cell within an array

=INDEX(array, row_num, column_num)



What range of cells
are you looking at?



How many rows down
is the value you want?



How many columns over
is the value you want?

	A	B	C
1	Tools	Price	Inventory
2	Hammer	\$5.00	55
3	Screw Driver	\$2.50	66
4	Pliers	\$3.34	333
5	Wrench set	\$10.00	234
6	Chain Saw	\$55.48	23
7	Tool Box	\$19.99	5
8	Level	\$2.25	7

INDEX(\$A\$1:\$C\$5, 5, 3) = 234

In this case we're telling Excel to find the value of a cell somewhere within the array of A1:C5. Starting from the upper left, we move down to the 5th row and right to the 3rd column, to return the value of 234

The **MATCH** function returns the *position* of a specific value within a column or row

=MATCH(lookup_value, lookup_array, [match_type])

What value are you trying to find the position of?

In which row or column are you looking? (**must be a 1-dimensional array**)

Are you looking for the exact value (0), or anything close?

	A	B
1	Tools	Price
2	Hammer	\$5.00
3	Screw Driver	\$2.50
4	Pliers	\$3.34
5	Wrench set	\$10.00

MATCH("Pliers",\$A\$1:\$A\$5, 0) = 4

	A	B	C
1	Tools	Price	Inventory
2	Hammer	\$5.00	55
3	Screw Driver	\$2.50	66
4	Pliers	\$3.34	333

MATCH(66,\$A\$3:\$C\$3, 0) = 3

Matching the word "Pliers" in column A, we find it in the **4th row**. Matching the number 66 in row 3, we find it in the **3rd column**

INDEX / MATCH

INDEX and **MATCH** are commonly used in tandem to act like a **LOOKUP** function; the only difference is that **INDEX/MATCH** can find values in any column or row in an array

Example: Price Checker

	A	B	C	D
1		Small	Medium	Large
2	Sweater	\$10	\$12	\$15
3	Jacket	\$30	\$35	\$40
4	Pants	\$25	\$30	\$35
5				
6	Product:	Pants		
8	Size:	Medium		
10	PRICE:	?		
11				

In this example, we want to populate the price of a given product and size in cell B10 by returning a particular value within the array B2:D4

B10=INDEX(B2:D4, MATCH(B6,A2:A4,0), MATCH(B8,B1:D1,0))



The number of rows down to index depends on what product I'm looking for, so we use a **MATCH** function and search for the value in cell B6 (in this case "Pants")

The number of columns over to index depends on what size I'm looking for, so we use a **MATCH** function and search for the value in cell B8 (in this case, "Medium")

Considering the output of each **MATCH** function, the formula is just a simple **INDEX**:

B10 = INDEX(B2:D4, 3, 2) = \$30



The **OFFSET** function is similar to **INDEX**, but can return either the value of a cell within an array (like **INDEX**) or a specific *range* of cells

=OFFSET(**reference**, **rows**, **columns**, [**height**], [**width**])

What's your
starting point?

How many rows
down should you
move?

How many
columns over
should you move?

If you want to return a
multidimensional array, how
tall and wide should it be?

An **OFFSET** formula where **[height]**=1 and **[width]**=1 will operate exactly like an **INDEX**. A more common use of **OFFSET** is to create dynamic arrays (like the Scroll Chart example in the appendix)



PRO TIP:

Don't use OFFSET or INDEX/MATCH when a simple VLOOKUP will do the trick

TRIM/UPPER/LOWER/PROPER

Text functions can be used to standardize formatting, particularly the **TRIM**, **UPPER**, **LOWER**, and **PROPER** functions:

	A	B	C	D
1	Sample Text String	Formula	Output	Notes
2	SAMPLE sentence	=TRIM(A2)	SAMPLE sentence	Removes any leading or trailing spaces from a text string
3	SAMPLE sentence	=LOWER(A3)	sample sentence	Converts all characters in a text string to lower case
4	SAMPLE sentence	=UPPER(A4)	SAMPLE SENTENCE	Converts all characters in a text string to upper case
5	SAMPLE sentence	=PROPER(A5)	Sample Sentence	Converts all characters in a text string to proper case (first letter capitalized)
6				

PRO TIP:

If two text strings are identical except one has a trailing space, they will look exactly the same but Excel will treat them as completely different values; TRIM will make them equivalent

CONCATENATE

CONCATENATE allows you to combine text, cell values, or formula outputs into a single text string

Note: Rather than typing “**=CONCATENATE(Text1, Text2...)**”, you can simply separate each piece of the resulting text string with an ampersand (“**&**”)

	A	B	C	D
1	First Name	Last Name	Formula	Output
2	Daniel	Wright	=A2&B2	DanielWright
3	Daniel	Wright	=A3&" "&B3	Daniel Wright
4	Daniel	Wright	=LEFT(A4,3)&" "&B4	Dan Wright
5	Daniel	Wright	=LEFT(A5,3)&" "&LEFT(B5,1)&"."	Dan W.

LEFT/MID/RIGHT/LEN

The **LEFT**, **MID**, and **RIGHT** functions return a specific number of characters from a location within a text string, and **LEN** returns the total number of characters

=LEFT(text, [num_chars])

=RIGHT(text, [num_chars])

=MID(text, start_num, num_chars)

A	B	C	D
1	Sample Text String	Formula	Output
2	MA-02215%AAA%_100	=LEFT(A3,2)	MA
3	MA-02215%AAA%_100	=MID(A5,4,5)	02215
4	MA-02215%AAA%_100	=RIGHT(A7,3)	100
5	MA-02215%AAA%_100	=LEN(A9)	17



The **TEXT** function converts a numeric value to text and assigns a particular format

=TEXT(value, format_text)

Numeric value, formula that evaluates to a numeric value, or reference to a cell containing a numeric value

Numeric format as a text string enclosed in quotes (i.e. “m/d/yyyy”, “\$0.00” or “#,##0.00”)

	A	B
1	Name	Earnings
2	Tim	\$4,500
3	George	\$3,250
4	Lisa	\$3,725

=“Lisa earned ”&B4 *returns “Lisa earned 3725”*

=“Lisa earned ”&TEXT(B4“#,###”) *returns “Lisa earned \$3,725”*



PRO TIP:

Use **VALUE** to convert a text string that represents a number into a value

The **SEARCH** function returns the number of the character at which a specific character or text string is first found (otherwise returns #VALUE! error)

=SEARCH(find_text, within_text, [start_num])

What character or string
are you searching for?

Where is the text that
you're searching through?

Search from the beginning (default) or
after a certain number of characters?

	A	B	C	D
11	MA-02215%AAA%_100	=SEARCH("%",A11)	9	Searches the string for "%" and returns the position
13	MA-02215%AAA%_100	=SEARCH("%",A13,10)	13	Searches for "%", starting with the 10th character, and returns the position
15	MA-02215%AAA%_100	=MID(A13,SEARCH("%",A13),5)	%AAA%	Returns 5 chars from the middle of the string, beginning where it finds "%"
17	MA-02215%AAA%_100	=MID(A13,SEARCH("%",A15)+1,3)	AAA	Returns 3 chars from the middle of the string, beginning 1 position after "%"



PRO TIP:

The **FIND** function works exactly the same way, but is case-sensitive

IF(ISNUMBER(SEARCH

IF(ISNUMBER(SEARCH is powerful combination of functions that can be used to classify data based on cells that contain specific strings of text

=IF(ISNUMBER(SEARCH(**find_text, within_text**)),**value_if_true, value_if_false**)



Searches for a specific string of text within a given cell



Returns one value if that string is found (TRUE), and another if it is not found (FALSE)

	A	B
1	Placement	Media
2	12983-Aff-160x90_small	Other
3	982308-Disp-160x90_large	Display
4	23124-Aff-160x90_small	Other
5	463-Disp-160x90_small	Display
6	390238-Agg-160x90_large	Other

=IF(ISNUMBER(SEARCH("Disp",A2)),"Display","Other")

Search the cells in column A for the text string "Disp" and classify column B as "Display" if you find it, "Other" if you don't



Every date in Excel has an associated **date value, which is how Excel calculates the passage of time (using midnight on 1/1/1900 as the starting point)**

Excel recognizes most typed dates and automatically applies a common format (i.e. m/d/yyyy), along with an associated date value (cell format → General)

*Note: If you type a date in a format that Excel does NOT recognize, it will be treated as text and there will be no associated date value; however, you can use a **DATEVALUE** or **TIMEVALUE** function to convert unformatted dates or times into serial values*

Date	Date Value
1/1/1900	1
1/11/1900	11
2/6/2015	42041
2/6/15 12:00 PM	42041.5
2/6/15 6:00 PM	42041.75



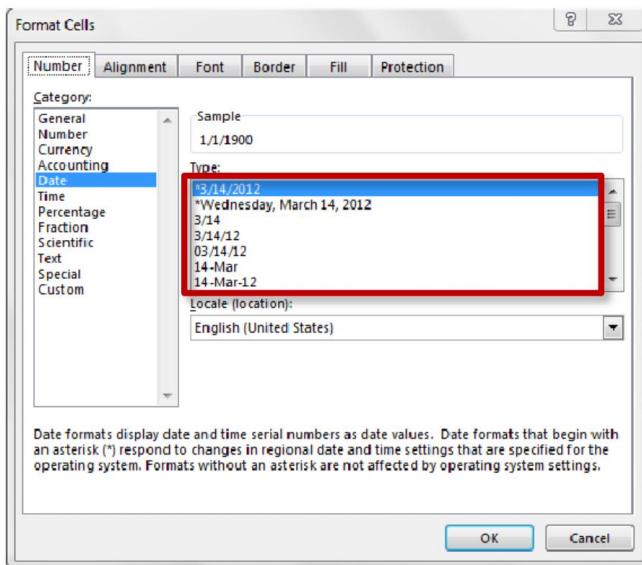
Jan 1, 1900 is the first date with an assigned date value (1). Feb 6, 2015 is the 42,041st day since 1/1/1900, so its date value = 42041

Date values can also indicate fractions of days: 42041.5 translates to noon on 2/6/2015 (50% through the day), and 42041.75 translates to 6:00pm on 2/6/2015 (75% through the day)

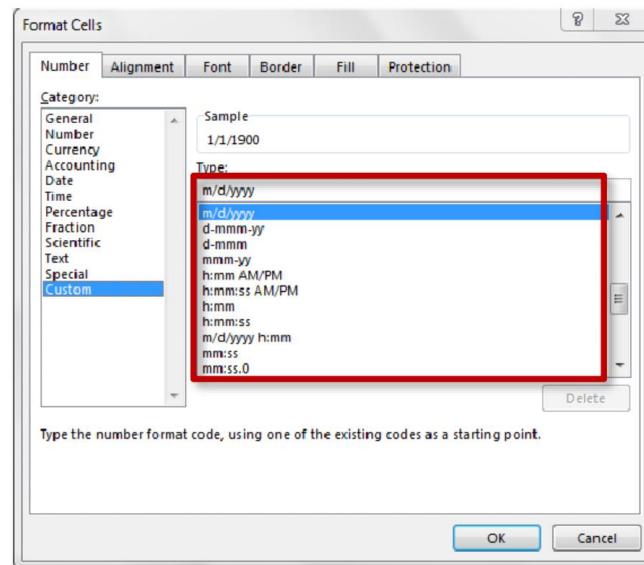
Date Formatting

To format dates in Excel, you can either select a preset option from the “Date” category of the “Format Cells” dialog box, OR create your own **custom format**

Preset Formats:



Custom Format:

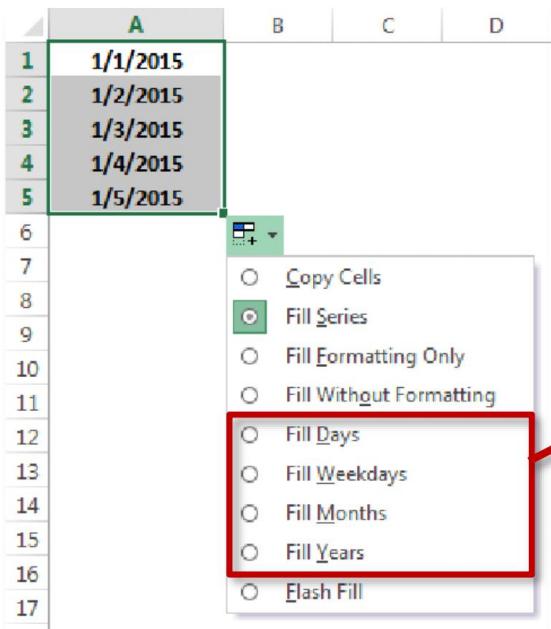


You can build your own custom formats using combinations of date/time codes. For example:

d = day w/out leading zero (1-31)
dd = day w/ leading zero (01-31)
ddd = day-of-week (Sat)
dddd = day-of-week (Saturday)
m = month w/out leading zero (1-15)
mm = month w/ leading zero (01-15)
mmm = month abbreviation (Jan)
mmmm = full month (January)
yy = last 2 digits of year (15)
yyyy = full year (2015)

(full list available at support.office.com)

When you drag the corner of a cell containing a date, Excel automatically applies subsequent values automatically using **Fill Series** options:



Click the **Auto Fill Options** button to determine exactly which values your subsequent cells should take:

Copy Cells = Repeats the same value in all cells

Fill Days = Increases the date by 1 day per cell

Fill Weekdays = Increases the date by 1 day per cell (excluding weekends)

Fill Months = Increases the date by 1 month per cell

Fill Years = Increases the date by 1 year per cell

The TODAY() and NOW() functions return the current date or exact time

*Note: These are **volatile** functions, meaning that they change with every worksheet calculation*

TODAY()
NOW()

2/6/2015
2/6/2015 17:15



This is what the TODAY() and NOW() functions return at 5:15pm on February 6, 2015. Note that these values will automatically update with every change made to the workbook



PRO TIP:

Make sure to enter TODAY() and NOW() functions with both parentheses included – these functions don't refer to other cells

Serialization Formulas

Excel will always calculate dates and times based on their *precise* underlying serial values, but what if you need to work with less-specific values, like months instead of days, or hours instead of seconds?

The **YEAR**, **MONTH**, **DAY**, **HOUR**, **MINUTE**, and **SECOND** functions extract individual components of a given date:

	A	B	C	D	E	F	G
1		YEAR	MONTH	DAY	HOUR	MINUTE	SECOND
2	2/6/2015 17:57	2015	2	6	17	57	16
3		=YEAR(A2)	=MONTH(A2)	=DAY(A2)	=HOUR(A2)	=MINUTE(A2)	=SECOND(A2)
4							

Use the **EOMONTH** function to calculate the last day of a given month, or to calculate the start/end dates of previous or future months

=EOMONTH(*start_date*, *months*)

Reference to the cell containing
the start/current date

Number of months before or after the start/current date (positive number
yields a date in the future, negative number yields a date in the past)

	A	B	C
1			
2		<i>Current Date:</i>	8/3/2015
3			
4		<i>End of month:</i>	8/31/2015
5		<i>Start of Month:</i>	8/1/2015
6		<i>Start of Next Month:</i>	9/1/2015

→ =EOMONTH(C2, 0)
→ =EOMONTH(C2, -1)+1
→ =EOMONTH(C2, 0)+1

YEARFRAC calculates the fraction of a year represented by the number of whole days between two dates

=YEARFRAC(start_date, end_date, [basis])

Reference to the cell containing the start date

Reference to the cell containing the end date

Option specify the type of day count to use:

0 (default) = US (NASD) 30/360

1 = actual/actual (**RECOMMENDED**)

2 = actual/360

3 = actual/365

4 = European 30/360

	A	B
1		
2	Start Date:	1/1/2015
3	End Date:	2/28/2015

=YEARFRAC(B2, B3, 1) = 15.9%

=YEARFRAC(B2, B3, 2) = 16.1%



PRO TIP:
YEARFRAC is a great tool for pacing and projection calculations

If you want to know which day of the week a given date falls on, there are two ways to do it:

- 1) Use a custom cell format of either “ddd” (Sat) or “ddddd” (Saturday)
-Note that this doesn't change the underlying value, only how that value is displayed
- 2) Use the **WEEKDAY** function to return a serial value corresponding to a particular day of the week (either 1-7 or 0-6)

=WEEKDAY(serial_number, [return type])



This refers to a cell containing a **date** or **time**



0 (default) = Sunday (1) to Saturday (7)
1 = Monday (1) to Sunday (7)
3 = Monday (0) to Sunday (6)

WORKDAY / NETWORKDAYS

WORKDAY returns a date that is a specified number of days before or after a given start date, excluding weekends and (optionally) holidays; **NETWORKDAYS** counts the number of workdays between two dates:

=WORKDAY(start_date, days, [holidays])

This refers to the cell containing the start date

Number of days before or after start date

Optional reference to a list of holiday dates

=NETWORKDAYS(start_date, end_date, [holidays])

This refers to the cell containing the start date

This refers to the cell containing the end date

Optional reference to a list of holiday dates

A	B
1	
2	Start Date: 1/1/2015
3	End Date: 2/28/2015

=WORKDAY(B2, 20) = 1/29/2015

=NETWORKDAYS(B2, B3) = 42

DATEDIF calculates the number of days, months, or years between two dates

=DATEDIF(start_date, end_date, unit)

Reference to the cell containing the start date

Reference to the cell containing the end date

How do you want to calculate the difference?

“D” = # of days between dates

“M” = # of months between dates

“Y” = # of years between dates

“MD” = # of days between dates, ignoring months and years

“YD” = # of days between dates, ignoring years

“YM” = # of months between dates, ignoring days and years

	A	B
1		
2	Start Date:	1/1/2015
3	End Date:	2/28/2015
4		

=DATEDIF(B2, B3, “D”) = 58
=DATEDIF(B2, B3, “MD”) = 27



PRO TIP:

If you only need to calculate the # of days between dates, just use subtraction

Formula-Based Formatting

If you want to go rogue, you can adjust the style of existing conditional formats or create your own **formula-based rules**

The image shows the Microsoft Excel ribbon on the left, specifically the Home tab under Styles. A red box highlights the 'Conditional Formatting' section, which includes options like 'Highlight Cells Rules', 'Top/Bottom Rules', 'Data Bars', 'Color Scales', 'Icon Sets', 'New Rule...', 'Clear Rules', and 'Manage Rules...'. An arrow points from the 'New Rule...' option in the ribbon to the first 'New Formatting Rule' dialog box. This dialog is titled 'New Formatting Rule' and has a 'Select a Rule Type:' section with several options. The first option, 'Format all cells based on their values', is selected. Below it is an 'Edit the Rule Description:' section with a 'Format Style:' dropdown set to '2-Color Scale'. It shows settings for 'Minimum' (Type: Lowest Value, Value: (Lowest value), Color: orange) and 'Maximum' (Type: Highest Value, Value: (Highest value), Color: yellow). A preview bar at the bottom shows a gradient from orange to yellow. Buttons for 'OK' and 'Cancel' are at the bottom right.

The second 'New Formatting Rule' dialog box is shown to the right of the first. It has the same structure, with 'Select a Rule Type:' showing 'Format all cells based on their values' as selected. In the 'Edit the Rule Description:' section, the 'Format values where this formula is true:' field contains the formula '=OR(\$G\$5=\$D\$12,\$G\$5=\$F\$12)'. A preview bar below shows text in different colors (blue, green, red) corresponding to the formula's output. Buttons for 'OK' and 'Cancel' are at the bottom right.

This is where you can add, clear, and manage your conditional formatting rules



Formula-Based

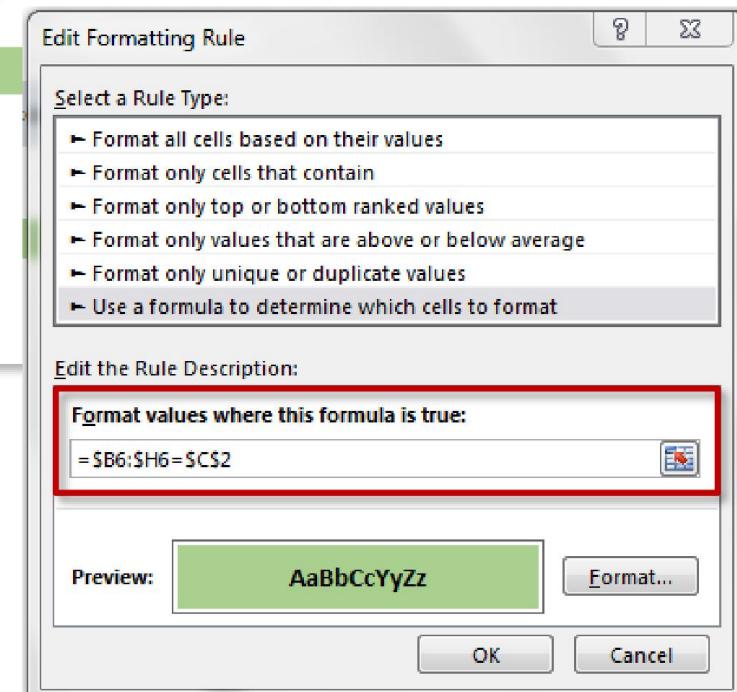
Formatting

State: **Arizona**

State	Population	Student Pop.	SAT Participation Rate	Mean Verbal Score	Mean Math Score
Alabama	4447100	177884	9%	559	554
Alaska	626932	12539	51%		
Arizona	5130632	102613	34%		
Arkansas	2001	2673400	53468	6%	
California	2001	33871648	1016149	51%	
Colorado	2001	4301261	172050	31%	
Connecticut	2001	3405565	102167	82%	
D.C.	2001	783600	23508	56%	
Delaware	2001	572059	17162	67%	

In this example we're formatting the cells in columns B through H with a green fill and bold text, but only when the state name is equal to the value in cell \$C\$2

Note that the row label is relative (no "\$"), which allows us to apply this formatting to other rows without losing functionality



Array Functions - Intro

Array functions perform multiple calculations on one or more items in an array, and can take the form of either a *single-cell* formula (which exists within one cell) or a *multi-cell* formula (which can be applied to a number of cells and return multiple results)

You must press **CTRL-SHIFT-ENTER** to enter, edit, or delete an array formula; this automatically adds brackets “{ }” to indicate that the function applies to an array

	A	B	C	D
1	Name	Earnings	Units	
2	Tim	\$4,500	4	\$18,000
3	George	\$3,250	2	
4	Lisa	\$3,725	3	
5	Zach	\$4,150	5	

If you select D2:D5, type “=B2:B5*C2:C5” and hit ENTER, the formula will only be applied to cell D2

	A	B	C	D
1	Name	Earnings	Units	
2	Tim	\$4,500	4	\$18,000
3	George	\$3,250	2	\$6,500
4	Lisa	\$3,725	3	\$11,175
5	Zach	\$4,150	5	\$20,750

If you select D2:D5, type “=B2:B5 * C2:C5” and hit CTRL-SHIFT-ENTER, you have created an array formula applied to all cells in the range

When you work with **array functions**, you must obey the following rules:



1. You *must* press **CTRL-SHIFT-ENTER (C-S-E)** to edit or enter an array formula
2. For multi-cell array functions, you must select the range of cells *before* entering the formula
3. You cannot change the contents of any individual cell which is part an array formula
4. You can move or delete an *entire* array formula, but not a piece of it (so you often have to delete and rebuild)
5. You cannot insert blank cells into or delete cells from a multi-cell array formula

Array functions can be incredibly powerful, but also a total buzzkill to work with; here are some of the key pros and cons of using them:

PROS	CONS
<ul style="list-style-type: none">-Condenses multiple calculations into one formula, often reducing file size-Can perform some complex functions that non-array formulas cannot-Reduces the risk of human error such as accidentally deleting parts of arrays or mistyping formulas	<ul style="list-style-type: none">-Can be very difficult to modify or delete existing array formulas-Limited visibility into the formula's function, especially for users who are not familiar with arrays-Eliminates the option to modify cells contained within arrays-May reduce processing speed if multiple array functions are used

Array constants are created by manually entering a list of items directly into the formula bar and manually surrounding the list with brackets ({})

	A	B	C	D
1	1	2	3	4
2				

Horizontal array constants create an array contained within a single row, and are delimited by commas (i.e. Select A1:D1, type “={1,2,3,4}” then hit C-S-E)

	A
1	1
2	2
3	3
4	4

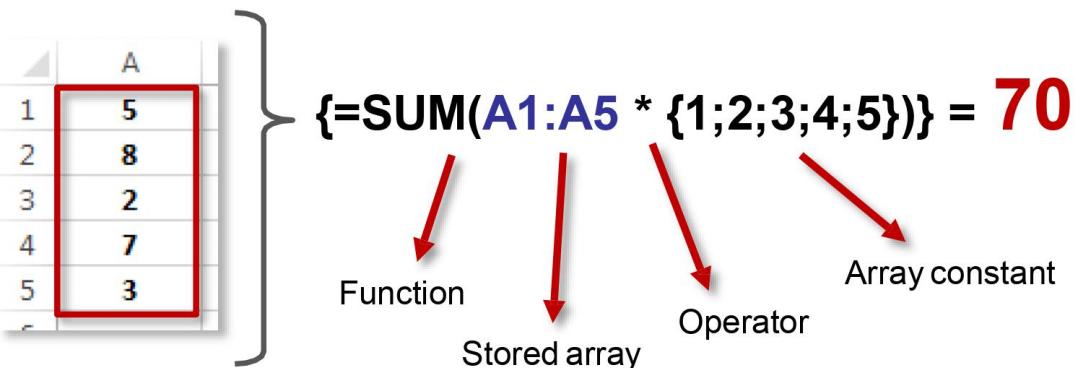
Vertical array constants create an array contained within a single column, and are delimited by semicolons (i.e. Select A1:A4, type “={1;2;3;4}” then hit C-S-E)

	A	B	C	D
1	1	2	3	4
2	5	6	7	8

Two-dimensional array constants create an array contained across multiple rows and columns (i.e. Select A1:D2, type “={1,2,3,4;5,6,7,8}” then hit C-S-E)

Array Constants in Formulas

Array constants can contain values, text (surrounded by “ ”), logical values (TRUE, FALSE), or error values (#N/A), and can be used as part of an array formula



*This function takes each value in the array A1:A5 and multiplies it against the corresponding value in the array constant {1;2;3;4;5}, which essentially translates into the following formula: =SUM(A1*1, A2*2, A3*3, A4*4, A5*5)*

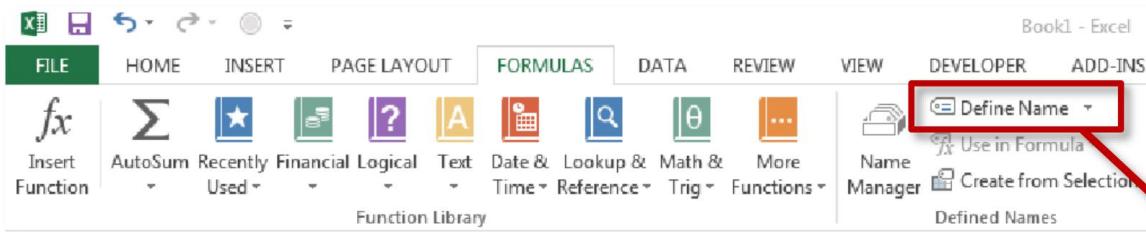
PRO TIP:



You manually add the brackets when you type array constants, but the additional brackets surrounding the entire formula are automatically added once you press C-S-E

Named Arrays

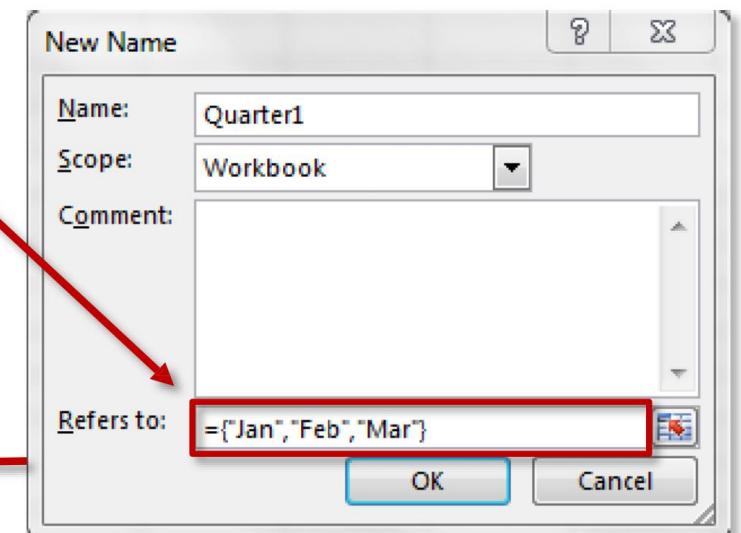
Just like normal cell ranges, **array constants** can be assigned a name using Excel's name manager, which can make them much easier to work with



Select "Define Name" (or Name Manager → New) from the **Formulas** tab

	A	B	C
1	Jan	Feb	Mar

Now if you select A1:C1, type "**=Quarter1**" and press CTRL-SHIFT-ENTER, the saved array will populate



In the **New Name** dialog box, enter the array constant (remembering to manually include the brackets), give it a name, and select OK

TRANSPOSE

The **TRANSPOSE** function allows you to change the orientation of a given data array
(i.e. from 5 rows x 2 columns to 2 rows x 5 columns)

NOTE: The range in which you enter a **TRANSPOSE** function must be the exact dimensions of the transposed data

{=TRANSPOSE(array)}

	A	B	C	D	E
3	20	125			
4	15	150			
5	25	120			
6	20	115			
7	15	140			
8					
9	20	15	25	20	15
10	125	150	120	115	140
..					

Select A9:E10, type “=TRANSPOSE(A3:B7)” and press
CTRL-SHIFT-ENTER to copy the transposed data



PRO TIP:

To transpose a data set that you may want to later edit, just use Paste Special → Transpose (ALT-H-V-T)

The **INDIRECT** function returns the reference specified by a text string, and can be used to change a cell reference within a formula without changing the formula itself

=INDIRECT(**ref_text**, [a1])



Which cell includes the text
that you are evaluating?

Is your text string in **A1** format (1) or **R1C1** format (0)?

	A	B
1		5
2		
3		B1
4		R1C2

ROW(B3) = 3

ROW(INDIRECT(B3)) = 1

ROW(INDIRECT(B4,0)) = 1

*In the first **ROW** function, Excel returns the row number of cell B3, regardless of what value it contains.*

*When you add **INDIRECT**, Excel sees that cell B3 contains a reference (B1) and returns the row of the reference*

Let's be real, the **INDIRECT** function is pretty confusing at first. Here are a few more examples that should give you a sense of how it works and why it can be useful:

	A	B	C	D
1	2014 Data			
2	Product	Sales	B3:B5	
3	A	5	B3:B5	
4	B	8	A3:B5	
5	C	3	A9:B11	
6				
7	2015 Data			
8	Product	Sales		
9	A	12		
10	B	17		
11	C	8		



$$\text{SUM(D2)} = 0$$

$$\text{SUM(INDIRECT(D2))} = 16$$

*The sum of “B3:B5” as a value doesn’t make sense, but the sum of B3:B5 as a reference is valid – **INDIRECT** tells Excel to recognize that the cell you’re referring to is a reference, not a value*

$$\text{VLOOKUP("A", D4, 2, 0)} = \#N/A$$

$$\text{VLOOKUP("A", INDIRECT(D4), 2, 0)} = 5$$

***INDIRECT** will tell a **VLOOKUP** formula to use an array contained within a cell, rather than treat the cell itself as the array (which returns #N/A)*

HYPERNLINK creates a shortcut that links users to a document or location within a document (which can exist on a network server, within a workbook, or via a web address)

=HYPERLINK(link_location,[friendly_name])



Where will people go if they click?



How do you want the link to read?

=HYPERLINK(http://www.example.com/report.xlsx, “Click Here”)

=HYPERLINK(“[C:\My Documents\Report.xlsx], “Open Report”)

=HYPERLINK("#Sheet2!A1")

