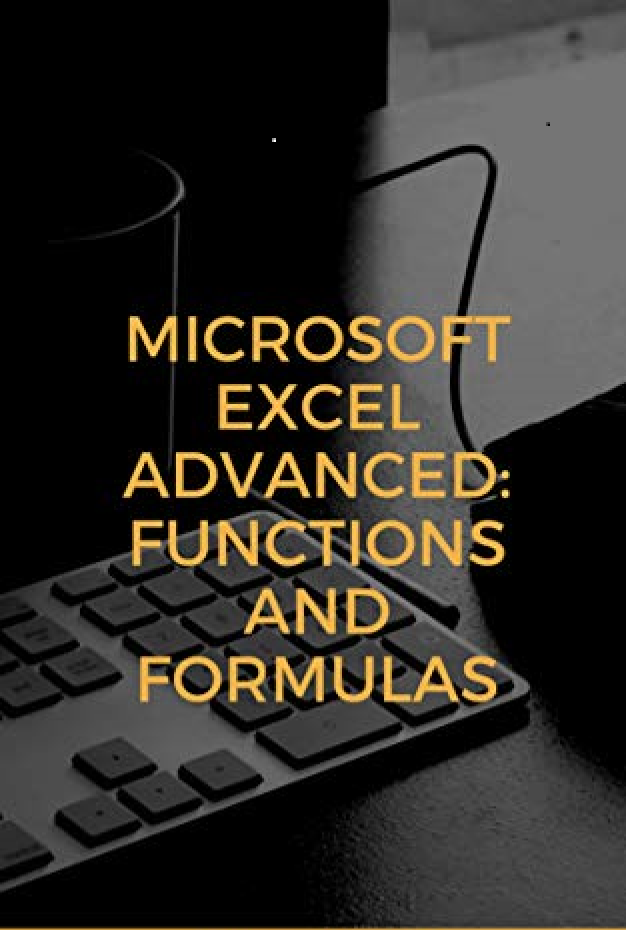


MICROSOFT EXCEL ADVANCED: FUNCTIONS AND FORMULAS

AHMED SHEIKH

Excel Advanced

Functions and Formulas

***Ahmed Sheikh,* M.Sc. (USA)**

TABLE OF CONTENTS

Unit 1: Building complex formulas

In this unit, you will learn how to:

- Work with nesting functions
- Compare results between using a formula with nested IF vs. a VLOOKUP
- Use the VLOOKUP with the COLUMN function
- Source table structure information using CHOOSE function
- Use INDEX and MATCH to search for information

Formulas and worksheet functions are essential to manipulating data and obtaining useful information from your Excel workbooks. This section includes formulas containing functions such as IF, AND, OR, NOT, CHOOSE, VLOOKUP, COLUMN, INDEX and MATCH and to make it more challenging the examples that will be used include combinations of these functions in the same formula.

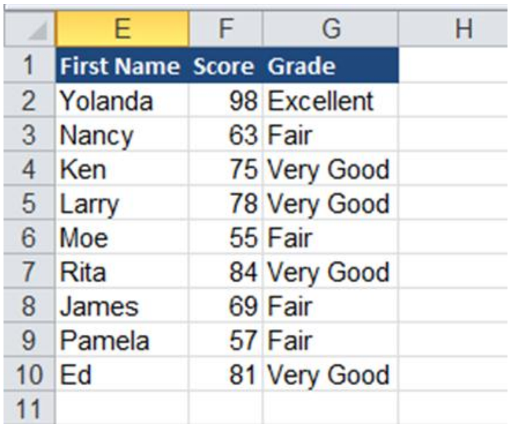
Using nested IF statements

In Excel the 'IF' function is commonly used as it provides solutions for many complex and varying situations in Excel. Let's begin with a formula containing multiple IF functions, also known as a nested IF formula. It is necessary to understand the syntax of this function which is as follows:

IF(Logical Test, Value if True, IF(Logical Test, Value if True, IF(Logical Test, Value if True, Value if False)))

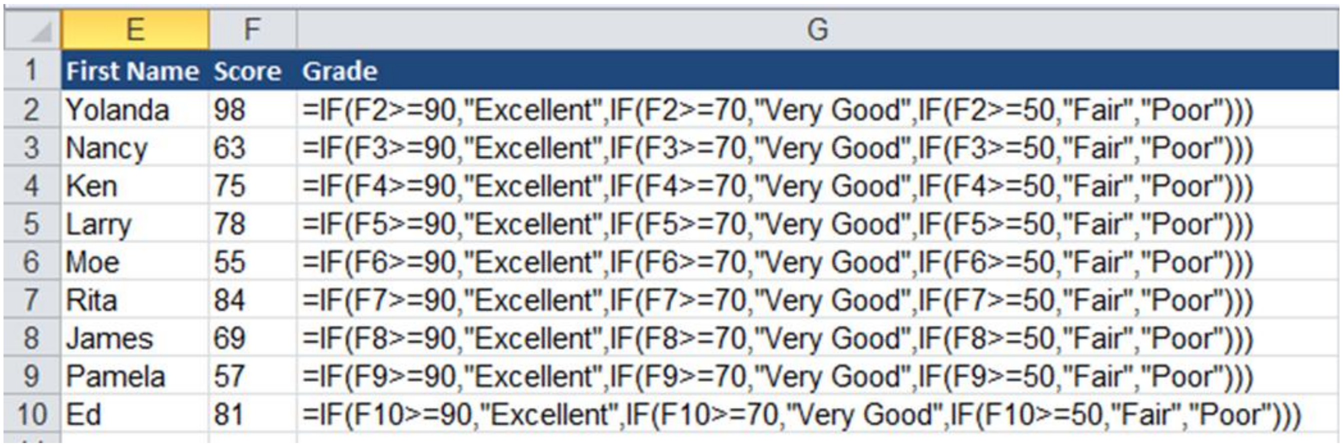
In the above example there are three logical tests which produce a value if true followed by a final result if none of the IF statements are true. The following is an example showing grading comments based on score results.

=IF(F2>=90,"Excellent", IF(F2>=70,"Very Good", IF(F2>=50,"Fair","Poor")))



	E	F	G	H
1	First Name	Score	Grade	
2	Yolanda	98	Excellent	
3	Nancy	63	Fair	
4	Ken	75	Very Good	
5	Larry	78	Very Good	
6	Moe	55	Fair	
7	Rita	84	Very Good	
8	James	69	Fair	
9	Pamela	57	Fair	
10	Ed	81	Very Good	
11				

To examine the formulas in more detail take a look at the following screenshot showing the breakdown of the nested IF formula which produces the required results.

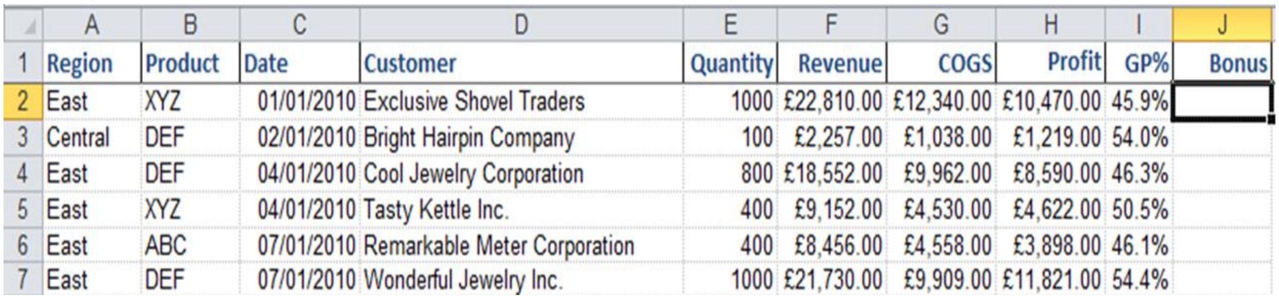


	E	F	G
1	First Name	Score	Grade
2	Yolanda	98	=IF(F2>=90,"Excellent",IF(F2>=70,"Very Good",IF(F2>=50,"Fair","Poor")))
3	Nancy	63	=IF(F3>=90,"Excellent",IF(F3>=70,"Very Good",IF(F3>=50,"Fair","Poor")))
4	Ken	75	=IF(F4>=90,"Excellent",IF(F4>=70,"Very Good",IF(F4>=50,"Fair","Poor")))
5	Larry	78	=IF(F5>=90,"Excellent",IF(F5>=70,"Very Good",IF(F5>=50,"Fair","Poor")))
6	Moe	55	=IF(F6>=90,"Excellent",IF(F6>=70,"Very Good",IF(F6>=50,"Fair","Poor")))
7	Rita	84	=IF(F7>=90,"Excellent",IF(F7>=70,"Very Good",IF(F7>=50,"Fair","Poor")))
8	James	69	=IF(F8>=90,"Excellent",IF(F8>=70,"Very Good",IF(F8>=50,"Fair","Poor")))
9	Pamela	57	=IF(F9>=90,"Excellent",IF(F9>=70,"Very Good",IF(F9>=50,"Fair","Poor")))
10	Ed	81	=IF(F10>=90,"Excellent",IF(F10>=70,"Very Good",IF(F10>=50,"Fair","Poor")))

Creating compound logical tests using AND, OR, NOT functions with IF statements

Using an IF function by itself in a formula works when there are no special conditions to be added. When the results are based on conditions then you have to nest the IF function with either the AND function or with the OR function. The AND function can hold up to 255 logical tests, however, to arrive at a result which is TRUE all the logical tests must be true. This makes the AND function quite restrictive in its use. On the other hand the OR function, which can also hold up to 255 logical tests, requires that only one of its logical tests be true in order to arrive at a result which is TRUE.

In the following screenshot the Excel spreadsheet represents sales data and we will be using the IF function together with AND, NOT and VLOOKUP to work out bonus calculations.



	A	B	C	D	E	F	G	H	I	J
1	Region	Product	Date	Customer	Quantity	Revenue	COGS	Profit	GP%	Bonus
2	East	XYZ	01/01/2010	Exclusive Shovel Traders	1000	£22,810.00	£12,340.00	£10,470.00	45.9%	
3	Central	DEF	02/01/2010	Bright Hairpin Company	100	£2,257.00	£1,038.00	£1,219.00	54.0%	
4	East	DEF	04/01/2010	Cool Jewelry Corporation	800	£18,552.00	£9,962.00	£8,590.00	46.3%	
5	East	XYZ	04/01/2010	Tasty Kettle Inc.	400	£9,152.00	£4,530.00	£4,622.00	50.5%	
6	East	ABC	07/01/2010	Remarkable Meter Corporation	400	£8,456.00	£4,558.00	£3,898.00	46.1%	
7	East	DEF	07/01/2010	Wonderful Jewelry Inc.	1000	£21,730.00	£9,909.00	£11,821.00	54.4%	

To pay a 2% bonus for sales greater than £20000 we would use a formula with a simple IF function that would be as follows: -

=IF(F2>20000,0.02*F2,0)

But what if we decided that the bonus would only be paid if the GP% is greater than 50% in addition to the first condition that sales must be greater than £20000? We could then use two IF functions one to deal with the first logical test and then the second to deal with the new logical test. The formula to produce the desired result would look like this: -

=IF(F2>20000,IF(I2>0.5,0.02*F2,0),0)

The exact same result can be achieved by using the following formula which incorporates both the IF and the AND functions.

=IF(AND(F2>20000,I2>0.5),0.02*F2,0)

If you decide to try the calculation without using any functions give the following formula a try.

=F2*0.02*(F2>20000)*(I2>0.5)

This formula starts out by calculating a 2% bonus for everyone: F2*0.02. But then the formula continues with two additional terms both of which must be in parentheses. Excel treats (F2>20000) as a logical test and will evaluate that expression to either TRUE or FALSE and will do the same with (I2>0.5). So, for row 2 the intermediate step will appear as follows: -

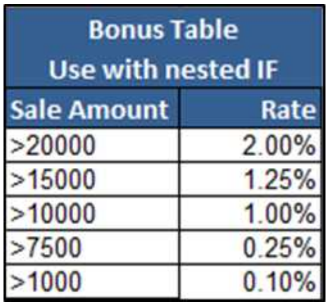
=22810*0.02*TRUE*FALSE

When Excel has to use TRUE or FALSE in a calculation, the TRUE is treated as a one. The FALSE is treated as a zero. Since any number times zero is zero, the logical tests at the end of the formula will wipe out the bonus if any one of the conditions is not true. The final step of the calculation will be as follows: -

=22810*0.02*1*0 which will equal zero. In row 7, the calculation will be =21730*0.02*TRUE*TRUE which becomes =21730*0.02*1*1 giving a result of £434.60.

Comparing a nested IF formula with VLOOKUP

What if your bonus rates were based on a sliding scale? You could use a nested IF formula which would evaluate each level starting with the largest category first. The bonus table would look as follows: -

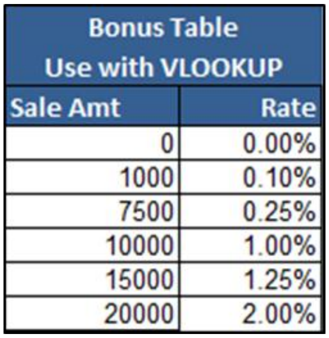


Bonus Table Use with nested IF	
Sale Amount	Rate
>20000	2.00%
>15000	1.25%
>10000	1.00%
>7500	0.25%
>1000	0.10%

The formula would be: -

=IF(F3>20000,$N$6,IF(F3>15000,$N$7,IF(F3>10000,$N$8,IF(F3>7500,$N$9,IF(F3>1000,$N$10,0)))))*F3

To produce the same result with a VLOOKUP function the bonus table must be modified to the following: -



Bonus Table Use with VLOOKUP	
Sale Amt	Rate
0	0.00%
1000	0.10%
7500	0.25%
10000	1.00%
15000	1.25%
20000	2.00%

The formula would be as follows: -

=VLOOKUP(F3,CommTable,2)*F3 where the range name 'CommTable' refers to M16:N21.

Using the NOT function

The NOT function can also be used in formulas to reverse the TRUE or FALSE result. We will use the NOT function in a formula which uses the IF function, together with the AND function. The formula will be as follows: -

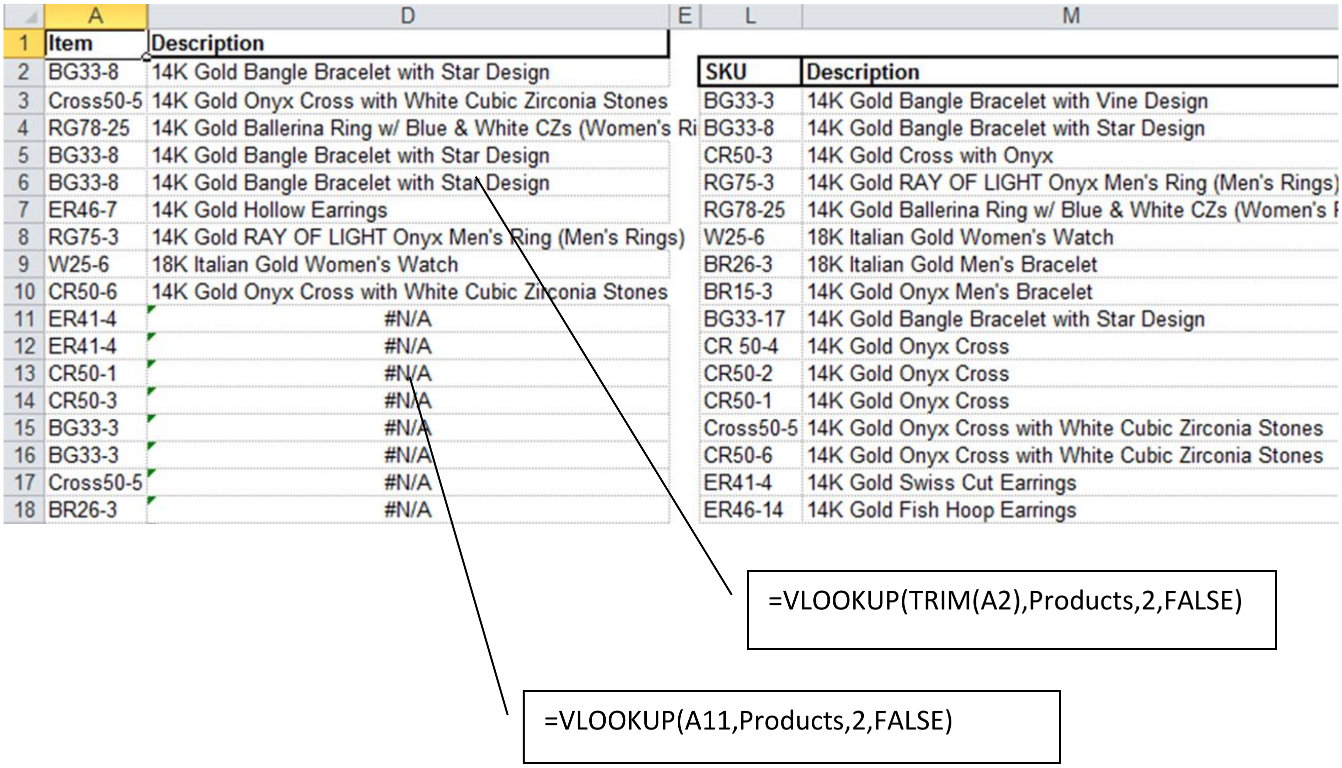
=IF(NOT(AND(F3>20000,J3=$O$6)),F3*0.02,0)

Cell F3 contains the sales value and cell O6 contains the rep's name. Normally, the AND function will produce a result if both logical tests are true, but with the NOT function in the picture, the results are reversed.

Nesting VLOOKUP functions

A common practice in Excel is to download data from other data sources. This often presents a problem when the data contains leading, trailing or extra unwanted spaces. So, when using a VLOOKUP function a lookup column containing trailing/extra spaces can cause #N/A errors to occur. The way around this is to add the TRIM function to the formula which immediately eliminates the spaces so that the lookup column data will match with the table array data.

Should the table array have any unwanted spaces, this can be dealt with in a similar fashion but with an added twist to the formula. The first result using TRIM before the table array will produce a #VALUE! error. This can be resolved by entering the formula by holding down the CTRL + SHIFT buttons and then pressing the enter key. This is known as an array formula and will be covered in more detail in a later unit called Array Functions.

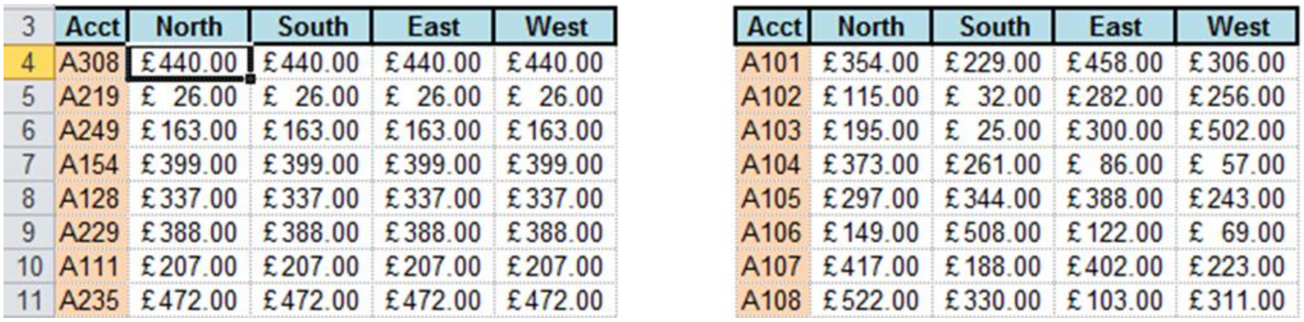


	A	D	E	L	M
1	Item	Description			
2	BG33-8	14K Gold Bangle Bracelet with Star Design		SKU	Description
3	Cross50-5	14K Gold Onyx Cross with White Cubic Zirconia Stones		BG33-3	14K Gold Bangle Bracelet with Vine Design
4	RG78-25	14K Gold Ballerina Ring w/ Blue & White CZs (Women's Ri		BG33-8	14K Gold Bangle Bracelet with Star Design
5	BG33-8	14K Gold Bangle Bracelet with Star Design		CR50-3	14K Gold Cross with Onyx
6	BG33-8	14K Gold Bangle Bracelet with Star Design		RG75-3	14K Gold RAY OF LIGHT Onyx Men's Ring (Men's Rings)
7	ER46-7	14K Gold Hollow Earrings		RG78-25	14K Gold Ballerina Ring w/ Blue & White CZs (Women's
8	RG75-3	14K Gold RAY OF LIGHT Onyx Men's Ring (Men's Rings)		W25-6	18K Italian Gold Women's Watch
9	W25-6	18K Italian Gold Women's Watch		BR26-3	18K Italian Gold Men's Bracelet
10	CR50-6	14K Gold Onyx Cross with White Cubic Zirconia Stones		BR15-3	14K Gold Onyx Men's Bracelet
11	ER41-4	#N/A		BG33-17	14K Gold Bangle Bracelet with Star Design
12	ER41-4	#N/A		CR 50-4	14K Gold Onyx Cross
13	CR50-1	#N/A		CR50-2	14K Gold Onyx Cross
14	CR50-3	#N/A		CR50-1	14K Gold Onyx Cross
15	BG33-3	#N/A		Cross50-5	14K Gold Onyx Cross with White Cubic Zirconia Stones
16	BG33-3	#N/A		CR50-6	14K Gold Onyx Cross with White Cubic Zirconia Stones
17	Cross50-5	#N/A		ER41-4	14K Gold Swiss Cut Earrings
18	BR26-3	#N/A		ER46-14	14K Gold Fish Hoop Earrings

=VLOOKUP(TRIM(A2),Products,2,FALSE)

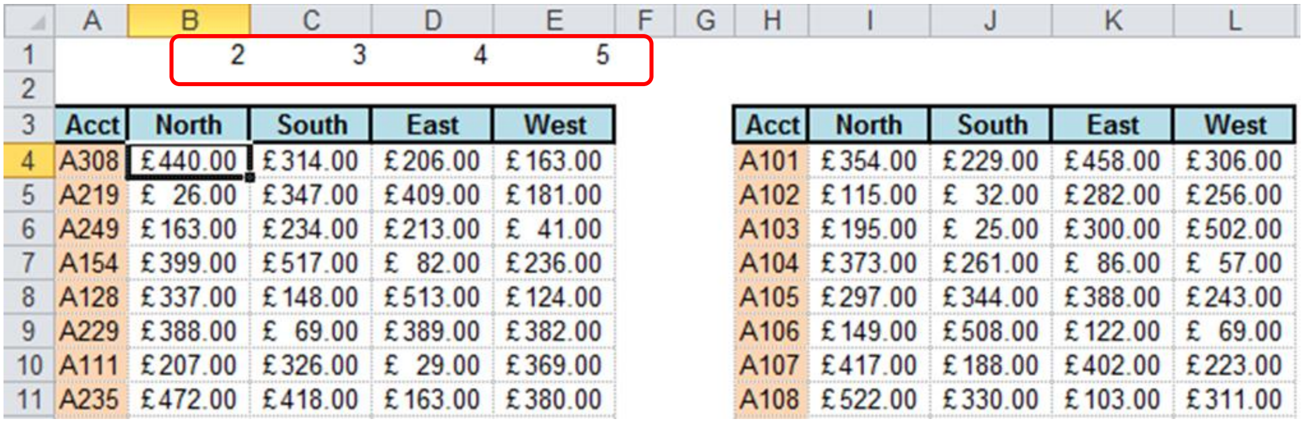
=VLOOKUP(A11,Products,2,FALSE)

Using VLOOKUP with the COLUMN function



3	Acct	North	South	East	West		Acct	North	South	East	West
4	A308	£ 440.00	£ 440.00	£ 440.00	£ 440.00		A101	£ 354.00	£ 229.00	£ 458.00	£ 306.00
5	A219	£ 26.00	£ 26.00	£ 26.00	£ 26.00		A102	£ 115.00	£ 32.00	£ 282.00	£ 256.00
6	A249	£ 163.00	£ 163.00	£ 163.00	£ 163.00		A103	£ 195.00	£ 25.00	£ 300.00	£ 502.00
7	A154	£ 399.00	£ 399.00	£ 399.00	£ 399.00		A104	£ 373.00	£ 261.00	£ 86.00	£ 57.00
8	A128	£ 337.00	£ 337.00	£ 337.00	£ 337.00		A105	£ 297.00	£ 344.00	£ 388.00	£ 243.00
9	A229	£ 388.00	£ 388.00	£ 388.00	£ 388.00		A106	£ 149.00	£ 508.00	£ 122.00	£ 69.00
10	A111	£ 207.00	£ 207.00	£ 207.00	£ 207.00		A107	£ 417.00	£ 188.00	£ 402.00	£ 223.00
11	A235	£ 472.00	£ 472.00	£ 472.00	£ 472.00		A108	£ 522.00	£ 330.00	£ 103.00	£ 311.00

The formula =VLOOKUP($A4,$H$4:$L$227,2,FALSE) is in cell B4 in the above example. When this is copied across the row number will not change and therefore the results are all the same. This can be overcome by numbering B1 to 2, C1 to 3, D1 to 4 and E1 to 5 then modifying the formula to the following =VLOOKUP($A4,$H$4:$L$227,B$1,FALSE). When this is copied across the results will now be correct. The following screenshot illustrates the result.



	A	B	C	D	E	F	G	H	I	J	K	L
1		2	3	4	5							
2												
3	Acct	North	South	East	West			Acct	North	South	East	West
4	A308	£ 440.00	£ 314.00	£ 206.00	£ 163.00			A101	£ 354.00	£ 229.00	£ 458.00	£ 306.00
5	A219	£ 26.00	£ 347.00	£ 409.00	£ 181.00			A102	£ 115.00	£ 32.00	£ 282.00	£ 256.00
6	A249	£ 163.00	£ 234.00	£ 213.00	£ 41.00			A103	£ 195.00	£ 25.00	£ 300.00	£ 502.00
7	A154	£ 399.00	£ 517.00	£ 82.00	£ 236.00			A104	£ 373.00	£ 261.00	£ 86.00	£ 57.00
8	A128	£ 337.00	£ 148.00	£ 513.00	£ 124.00			A105	£ 297.00	£ 344.00	£ 388.00	£ 243.00
9	A229	£ 388.00	£ 69.00	£ 389.00	£ 382.00			A106	£ 149.00	£ 508.00	£ 122.00	£ 69.00
10	A111	£ 207.00	£ 326.00	£ 29.00	£ 369.00			A107	£ 417.00	£ 188.00	£ 402.00	£ 223.00
11	A235	£ 472.00	£ 418.00	£ 163.00	£ 380.00			A108	£ 522.00	£ 330.00	£ 103.00	£ 311.00

There is one drawback to this because someone who is not familiar with the worksheet may inadvertently delete the numbers in the first row adversely affecting the VLOOKUP formula. To deal with this possible problem there is an alternative method which may be used and that is to add the COLUMN function together with the VLOOKUP function into the formula.

The formula to achieve this would be as follows: -

=VLOOKUP($A4,Accounts,COLUMN(B1),FALSE) where Accounts is the name given to the Lookup_Array $H$4:$L$227.

Source table structure information using CHOOSE function

The CHOOSE function picks from a list of options which are based upon an Index value given by the user.

CHOOSE(index_num, value1, [value2], ...)

The CHOOSE function syntax has the following arguments:

Index_num - This is required. Specifies which value argument is selected. Index_num must be a number between 1 and 254, or a formula or reference to a cell containing a number between 1 and 254.

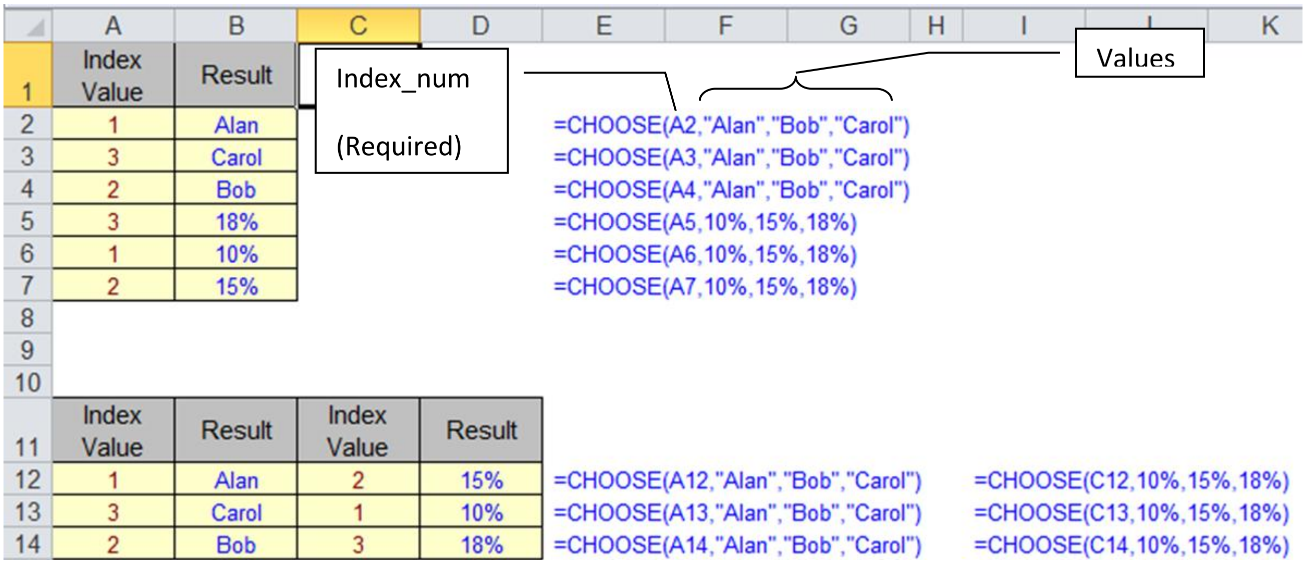
If index_num is 1, CHOOSE returns value1; if it is 2, CHOOSE returns value2; and so on.

If index_num is less than 1 or greater than the number of the last value in the list, CHOOSE returns the #VALUE! error.

If index_num is a fraction, it is truncated to the lowest integer before being used.

Value1, [value2], ... Value 1 is required, subsequent values are optional. 1 to 254 value arguments from which CHOOSE selects a value or an action to perform based on index_num. The arguments can be numbers, cell references, defined names, formulas, functions, or text.

The following example illustrates how to produce results which are based on an index number. In the case below, the CHOOSE function only uses three values to choose from.



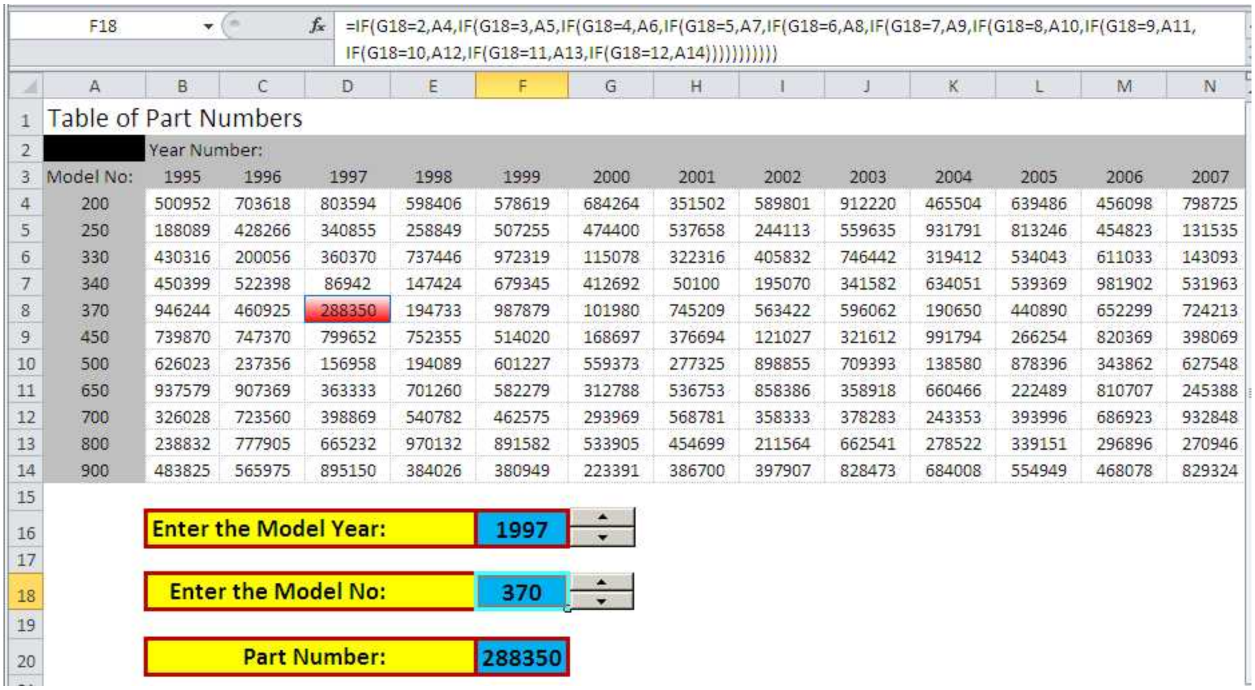
	A	B	C	D	E	F	G	H	I	J	K
1	Index Value	Result									
2	1	Alan			=CHOOSE(A2,"Alan","Bob","Carol")						
3	3	Carol			=CHOOSE(A3,"Alan","Bob","Carol")						
4	2	Bob			=CHOOSE(A4,"Alan","Bob","Carol")						
5	3	18%			=CHOOSE(A5,10%,15%,18%)						
6	1	10%			=CHOOSE(A6,10%,15%,18%)						
7	2	15%			=CHOOSE(A7,10%,15%,18%)						
8											
9											
10											
11	Index Value	Result	Index Value	Result							
12	1	Alan	2	15%	=CHOOSE(A12,"Alan","Bob","Carol")				=CHOOSE(C12,10%,15%,18%)		
13	3	Carol	1	10%	=CHOOSE(A13,"Alan","Bob","Carol")				=CHOOSE(C13,10%,15%,18%)		
14	2	Bob	3	18%	=CHOOSE(A14,"Alan","Bob","Carol")				=CHOOSE(C14,10%,15%,18%)		

Compare the CHOOSE function with Nested IF with INDEX

The following example shows how there is sometimes a choice of which reference functions to use to produce the same result. Some functions achieve the result more efficiently than others.

The aim is for a particular cell, F18, to display a different value of Model No whenever a spinner control is incremented from 2 to 12 according to the table below:

Spinner value	Model No
2	200
3	250
4	330
5	340
6	370
7	450
8	500
9	650
10	700
11	800
12	900



F18

=IF(G18=2,A4,IF(G18=3,A5,IF(G18=4,A6,IF(G18=5,A7,IF(G18=6,A8,IF(G18=7,A9,IF(G18=8,A10,IF(G18=9,A11,IF(G18=10,A12,IF(G18=11,A13,IF(G18=12,A14)))))))))))

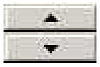
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Table of Part Numbers													
2		Year Number:												
3	Model No:	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007
4	200	500952	703618	803594	598406	578619	684264	351502	589801	912220	465504	639486	456098	798725
5	250	188089	428266	340855	258849	507255	474400	537658	244113	559635	931791	813246	454823	131535
6	330	430316	200056	360370	737446	972319	115078	322316	405832	746442	319412	534043	611033	143093
7	340	450399	522398	86942	147424	679345	412692	50100	195070	341582	634051	539369	981902	531963
8	370	946244	460925	288350	194733	987879	101980	745209	563422	596062	190650	440890	652299	724213
9	450	739870	747370	799652	752355	514020	168697	376694	121027	321612	991794	266254	820369	398069
10	500	626023	237356	156958	194089	601227	559373	277325	898855	709393	138580	878396	343862	627548
11	650	937579	907369	363333	701260	582279	312788	536753	858386	358918	660466	222489	810707	245388
12	700	326028	723560	398869	540782	462575	293969	568781	358333	378283	243353	393996	686923	932848
13	800	238832	777905	665232	970132	891582	533905	454699	211564	662541	278522	339151	296896	270946
14	900	483825	565975	895150	384026	380949	223391	386700	397907	828473	684008	554949	468078	829324

Enter the Model Year: 1997

Enter the Model No: 370

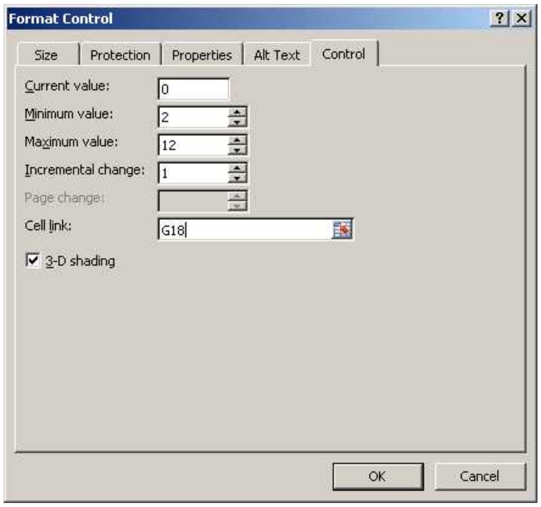
Part Number: 288350

Creating the Spinner control



To create the Spinner control for cell G18

- First turn on the Developer tab
- 2010 File, Options, Customise Ribbon, tick Developer tab in the right pane
- 2007 Office Button, Popular, tick Developer tab
- Select Developer tab, Insert in the Controls group
- Choose the Spin Button (Form Control)
- Click on the spreadsheet to place the spin button
- Size and place the spin button over cell G18



Current value: 0

Minimum value: 2

Maximum value: 12

Incremental change: 1

Page change:

Cell link: G18

☑ 3-D shading

- Right click the control, Format Control
- Set the minimum value to 2, maximum value to 12
- These values represent the HLOOKUP row numbers which return the Part Numbers
- Incremental change should be 1
- Cell link to G18
- Click OK

A nested IF formula in cell F18 has been used to produce the value of the Model Number. The formula takes a fair amount of time to construct. Is there an easier way? Trying using the CHOOSE function! The following formula will produce the same result as the nested IF statement.

```
=CHOOSE(G18-1,200,250,330,340,370,450,500,650,700,800,900)
```

This formula is also fairly long and time consuming to create. Could there be an easier way? Try using the INDEX function! The following formula will produce matching results of both nested IF and CHOOSE.

```
=INDEX(A3:A14,G18)
```

Using the MATCH function to locate data position

The MATCH function searches for a specified item in a range of cells, and then returns the relative position of that item in the range. For example, if the range A1:A3 contains the values 5, 25, and 38, then the formula

=MATCH(25,A1:A3,0) returns the number 2, because 25 is the second item in the range.

Use MATCH instead of one of the LOOKUP functions when you need the position of an item in a range instead of the item itself. For example, you might use the MATCH function to provide a value for the row_num argument of the INDEX function.

Syntax

MATCH(lookup_value, lookup_array, [match_type])

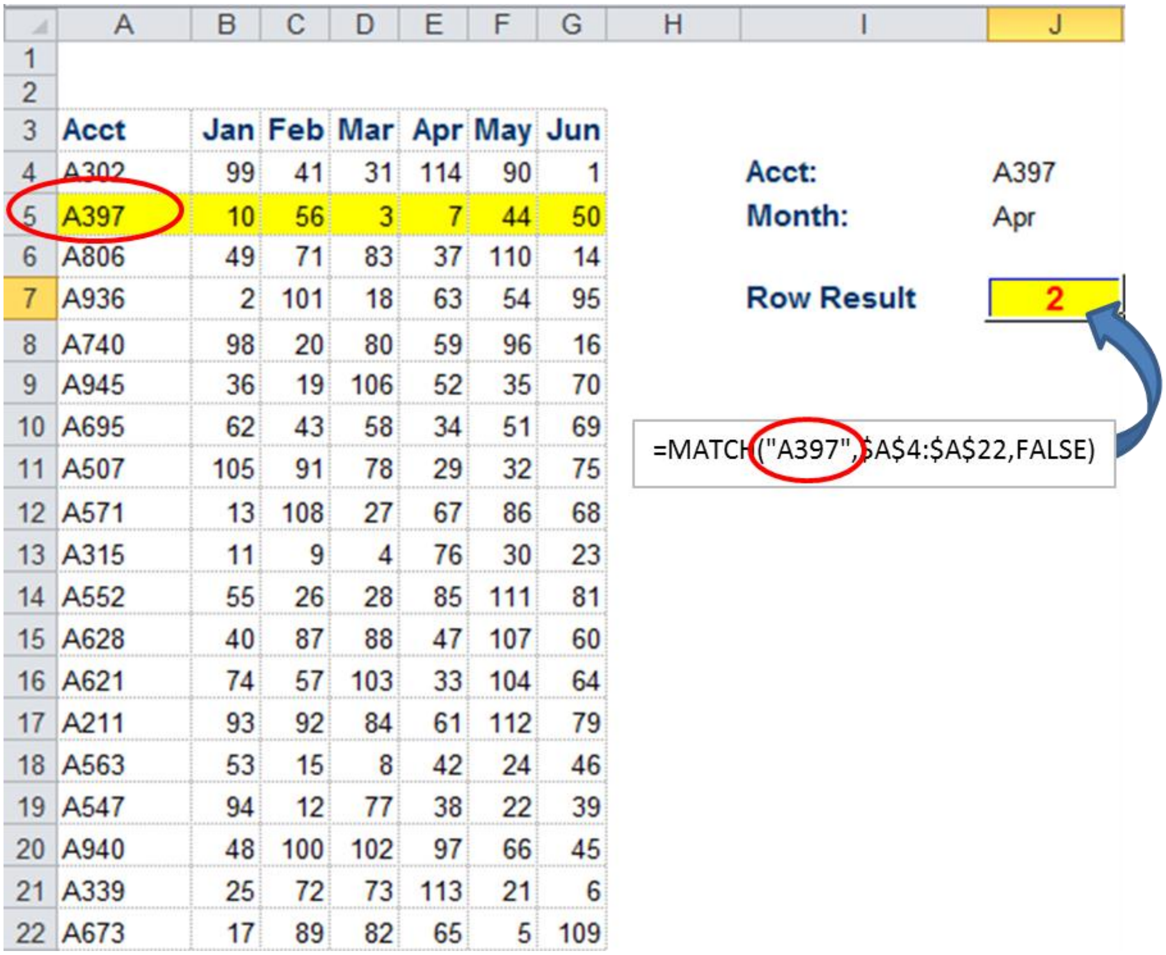
Match Types

Match_type 1 or omitted	**MATCH** finds the largest value that is less than or equal to ***lookup_value***. The values in the ***lookup_array*** argument must be placed in ascending order, for example: ...-2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE.
Match_type 0	**MATCH** finds the first value that is exactly equal to ***lookup_value***. The values in the ***lookup_array*** argument can be in any order.
Match_type -1	**MATCH** finds the smallest value that is greater than or equal to ***lookup_value***. The values in the ***lookup_array*** argument must be placed in descending order, for example: TRUE, FALSE, Z-A, ...2, 1, 0, -1, -2, ..., and so on.

If ***match_type*** is 0 and ***lookup_value*** is a text string, you can use the wildcard characters — the question mark (**?**) and asterisk (*****) — in the ***lookup_value*** argument.

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.

The following example shows how the MATCH function finds the row number in which a particular item is located.



	A	B	C	D	E	F	G
1							
2							
3	**Acct**	**Jan**	**Feb**	**Mar**	**Apr**	**May**	**Jun**
4	A302	99	41	31	114	90	1
5	A397	10	56	3	7	44	50
6	A806	49	71	83	37	110	14
7	A936	2	101	18	63	54	95
8	A740	98	20	80	59	96	16
9	A945	36	19	106	52	35	70
10	A695	62	43	58	34	51	69
11	A507	105	91	78	29	32	75
12	A571	13	108	27	67	86	68
13	A315	11	9	4	76	30	23
14	A552	55	26	28	85	111	81
15	A628	40	87	88	47	107	60
16	A621	74	57	103	33	104	64
17	A211	93	92	84	61	112	79
18	A563	53	15	8	42	24	46
19	A547	94	12	77	38	22	39
20	A940	48	100	102	97	66	45
21	A339	25	72	73	113	21	6
22	A673	17	89	82	65	5	109

I	J
Acct:	A397
Month:	Apr
Row Result	2

```
=MATCH("A397",$A$4:$A$22,FALSE)
```

Use the INDEX function for retrieving information by location

The INDEX function returns a value or the reference to a value from within a table or range. There are two forms of the INDEX function: the arrayform and the reference form. Definition of Array: Used to build single formulas that produce multiple results or that operate on a group of arguments that are arranged in rows and columns. An array range shares a common formula; an array constant is a group of constants used as an argument.

The INDEX Array form

The INDEX array form returns the value of an element in a table or an array, selected by the row and column number indexes.

Use the array form if the first argument to INDEX is an array constant.

The Syntax for the INDEX function is as follows: -

INDEX(array, row_num, [column_num])

The INDEX function syntax has the following arguments: -

Array (Required) is a range of cells or an array constant.

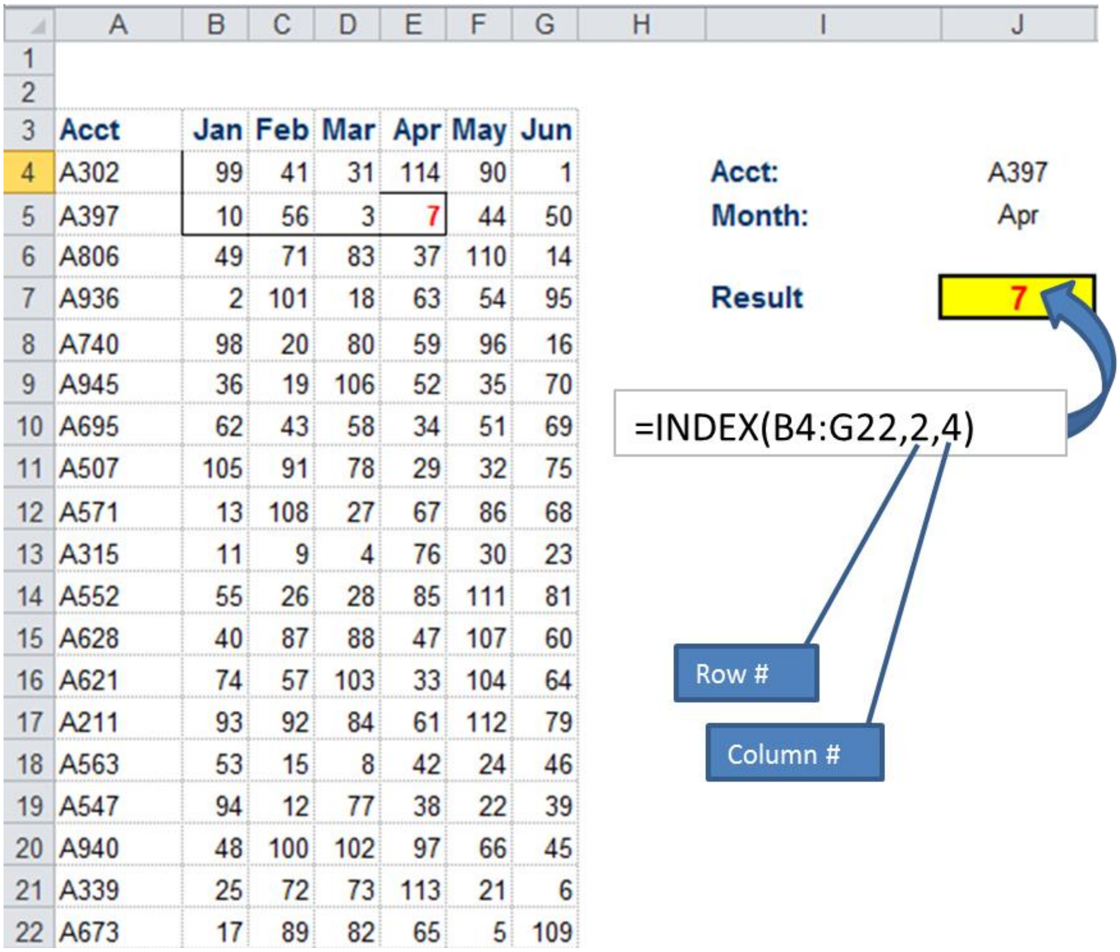
If array contains only one row or column, the corresponding row_num or column_num argument is optional.

If array has more than one row and more than one column, and only row_num or column_num is used, INDEX returns an array of the entire row or column in array.

Row_num Required. Selects the row in array from which to return a value. If row_num is omitted, column_num is required.

Column_num Optional. Selects the column in array from which to return a value. If column_num is omitted, row_num is required.

The next example shows how a value is retrieved from a table array using the INDEX function incorporating the row number and column number options.



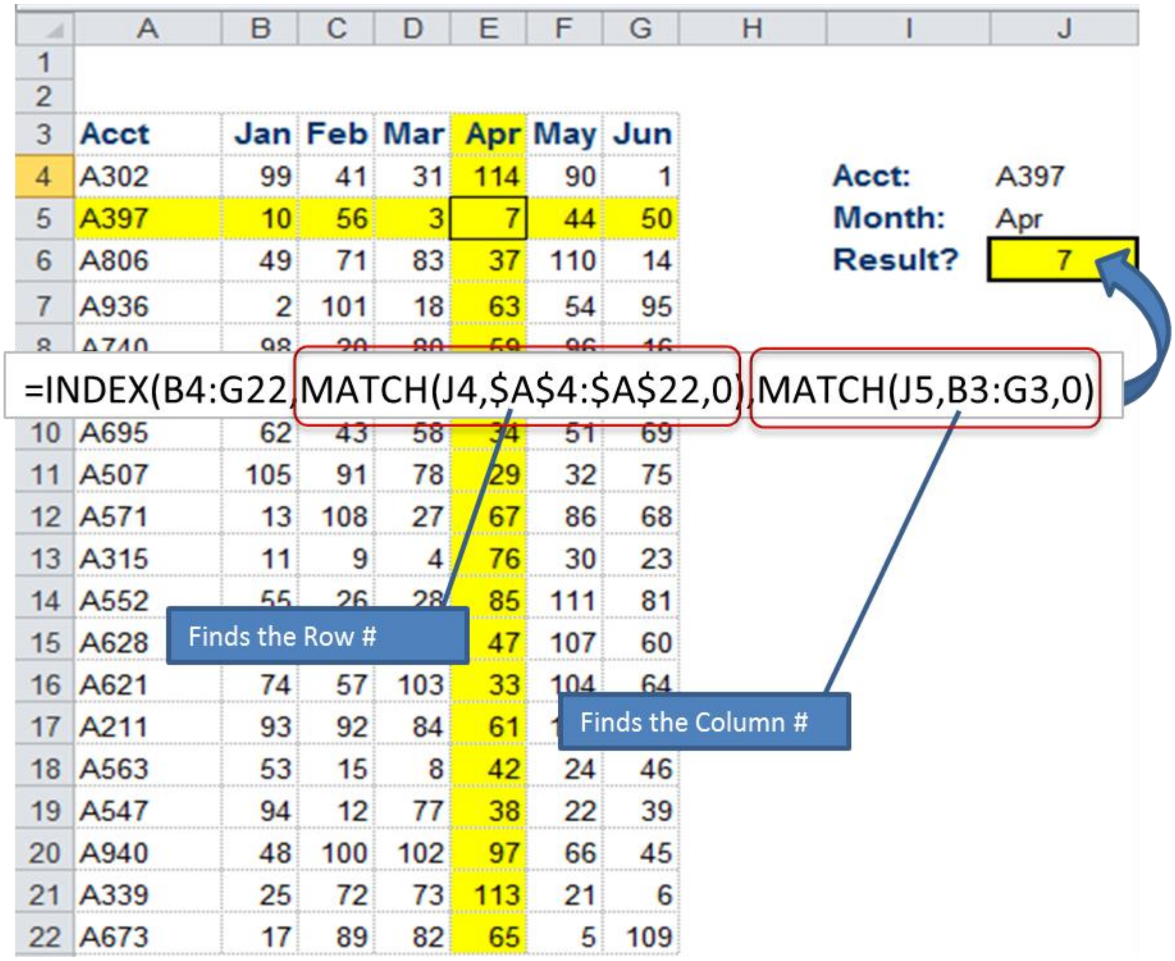
Acct	Jan	Feb	Mar	Apr	May	Jun
A302	99	41	31	114	90	1
A397	10	56	3	7	44	50
A806	49	71	83	37	110	14
A936	2	101	18	63	54	95
A740	98	20	80	59	96	16
A945	36	19	106	52	35	70
A695	62	43	58	34	51	69
A507	105	91	78	29	32	75
A571	13	108	27	67	86	68
A315	11	9	4	76	30	23
A552	55	26	28	85	111	81
A628	40	87	88	47	107	60
A621	74	57	103	33	104	64
A211	93	92	84	61	112	79
A563	53	15	8	42	24	46
A547	94	12	77	38	22	39
A940	48	100	102	97	66	45
A339	25	72	73	113	21	6
A673	17	89	82	65	5	109

Acct:	A397
Month:	Apr
Result	7

=INDEX(B4:G22,2,4)

Using a nested formula containing INDEX, MATCH and MATCH (two-way lookup)

Both the INDEX and MATCH functions can be used independently or can be used in the same formula. To combine the INDEX function with MATCH to find both row and column positions, the MATCH function must be used twice as in the following example.



	A	B	C	D	E	F	G	H	I	J
1										
2										
3	Acct	Jan	Feb	Mar	Apr	May	Jun			
4	A302	99	41	31	114	90	1		Acct:	A397
5	A397	10	56	3	7	44	50		Month:	Apr
6	A806	49	71	83	37	110	14		Result?	7
7	A936	2	101	18	63	54	95			
8	A740	98	20	80	59	96	16			
10	A695	62	43	58	34	51	69			
11	A507	105	91	78	29	32	75			
12	A571	13	108	27	67	86	68			
13	A315	11	9	4	76	30	23			
14	A552	55	26	28	85	111	81			
15	A628	[illegible]	[illegible]	[illegible]	47	107	60			
16	A621	74	57	103	33	104	64			
17	A211	93	92	84	61	[illegible]	[illegible]			
18	A563	53	15	8	42	24	46			
19	A547	94	12	77	38	22	39			
20	A940	48	100	102	97	66	45			
21	A339	25	72	73	113	21	6			
22	A673	17	89	82	65	5	109			

=INDEX(B4:G22,MATCH(J4,$A$4:$A$22,0),MATCH(J5,B3:G3,0))

Unit 2: Using Advanced Functions

In this unit you will learn how to:

- Use COUNTIFS, SUMIFS and AVERAGEIFS to tabulate data based on single/multiple criteria
- Use Statistical functions: MEDIAN, RANK, LARGE, SMALL
- Use Maths functions: Round and related functions, the Mod function
- Use the AGGREGATE function to sum data in ranges with errors
- Use a variety of Financial functions such as PMT, FV, IRR

Use COUNTIFS and SUMIFS

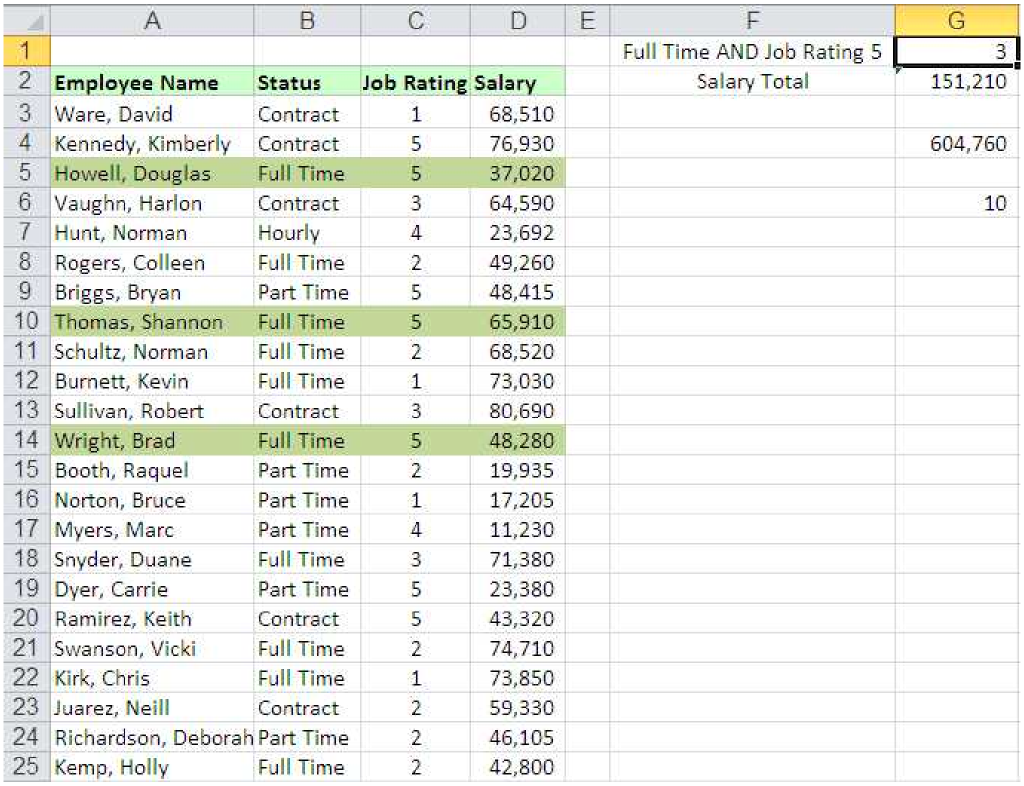
Excel has the useful functions COUNTIF and SUMIF which are able to count the number of records or sum values of a field based on a criteria. In the list below for example theses function calculate there are 10 'Full Time' employees with a total salary of £604,760. Here are the formulas:

```
=COUNTIF(B3:B25,"Full Time")
=SUMIF(B3:B25,"Full Time",D3:D25)
```

Similarly the AVERAGEIF function would calculate the average salary for the Full Time employees:

```
=AVERAGEIF(B3:B25,"Full Time",D3:D25)
```

Since Excel 2007 there has been a corresponding set of functions ending with the letter S. COUNTIFS, SUMIFS and AVERAGEIFS. These functions allow for **multiple criteria**. For example, the number of Full Time employees with a job rating of 5.



	A	B	C	D	E	F	G
1						Full Time AND Job Rating 5	3
2	**Employee Name**	**Status**	**Job Rating**	**Salary**		Salary Total	151,210
3	Ware, David	Contract	1	68,510			
4	Kennedy, Kimberly	Contract	5	76,930			604,760
5	Howell, Douglas	Full Time	5	37,020			
6	Vaughn, Harlon	Contract	3	64,590			10
7	Hunt, Norman	Hourly	4	23,692			
8	Rogers, Colleen	Full Time	2	49,260			
9	Briggs, Bryan	Part Time	5	48,415			
10	Thomas, Shannon	Full Time	5	65,910			
11	Schultz, Norman	Full Time	2	68,520			
12	Burnett, Kevin	Full Time	1	73,030			
13	Sullivan, Robert	Contract	3	80,690			
14	Wright, Brad	Full Time	5	48,280			
15	Booth, Raquel	Part Time	2	19,935			
16	Norton, Bruce	Part Time	1	17,205			
17	Myers, Marc	Part Time	4	11,230			
18	Snyder, Duane	Full Time	3	71,380			
19	Dyer, Carrie	Part Time	5	23,380			
20	Ramirez, Keith	Contract	5	43,320			
21	Swanson, Vicki	Full Time	2	74,710			
22	Kirk, Chris	Full Time	1	73,850			
23	Juarez, Neill	Contract	2	59,330			
24	Richardson, Deborah	Part Time	2	46,105			
25	Kemp, Holly	Full Time	2	42,800			

The COUNTIFS function prompts for the first criteria range and first criteria (Status range B3:B25 and “Full Time”) followed by the second criteria range and second criteria (Job rating range and a rating of 5) Here is the function:

=COUNTIFS(B3:B25,"full time",C3:C25,5)

Similarly the SUMIF function calculates the total salary for the same two criteria.

=SUMIFS(D3:D25,B3:B25,"full time",C3:C25,5)

In this example there are 3 Full Time employees with a total salary of £151,210.

Excel allows a maximum of 127 range/criteria pairs.

Using COUNTIF with OR logic

When using COUNTIF S the criteria combine with AND logic. The more criteria used the fewer the records included. reducing the number of records being counted.

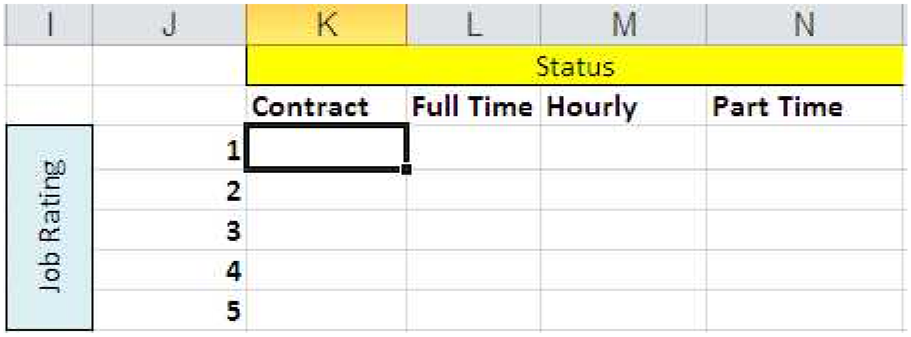
To combine criteria with OR logic conditions the simll add the COUNTIF functions together. For example to count Full Timers or Part Timers enter the formula as follows:

=COUNTIF(B3:B25,"full time",C3:C25,5) + COUNTIF(B3:B25,"part time ",C3:C25,5)

This results in finding 17 employees who are working Full time or Part Time.

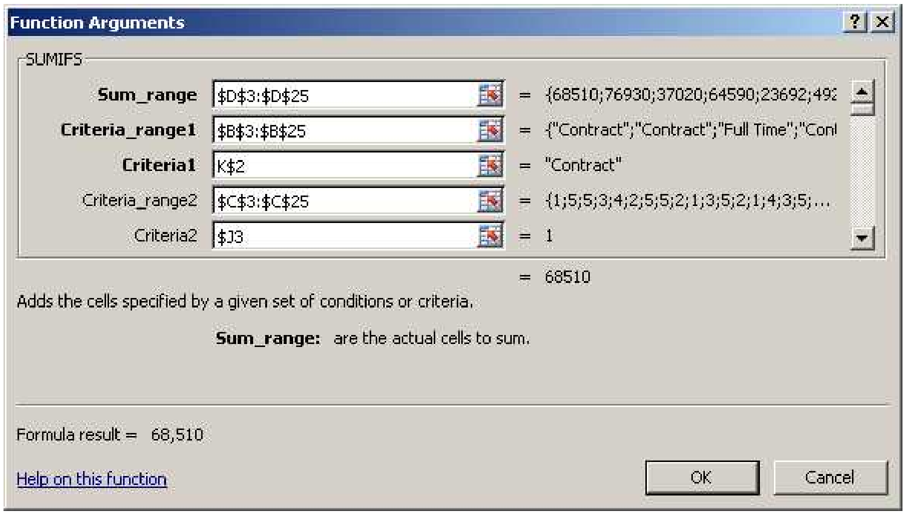
Creating Tabulated data using SUMIFS

Rather than just calculating one result from a SumIfs it is possible to create tabulated data that allows a comparison to be made between all the Job Ratings and Status types.



I	J	K	L	M	N
		Status			
		Contract	Full Time	Hourly	Part Time
Job Rating	1				
	2				
	3				
	4				
	5				

To create tabulated data using the SUMIFS function first type all the different values as Row and column labels. Then click at the intersection point K3.and create the SUMIFS function:



Function Arguments

SUMIFS

Sum_range	$D$3:$D$25	= {68510;76930;37020;64590;23692;49:
Criteria_range1	$B$3:$B$25	= {"Contract";"Contract";"Full Time";"Con
Criteria1	K$2	= "Contract"
Criteria_range2	$C$3:$C$25	= {1;5;5;3;4;2;5;5;2;1;3;5;2;1;4;3;5;...
Criteria2	$J3	= 1

= 68510

Adds the cells specified by a given set of conditions or criteria.

Sum_range: are the actual cells to sum.

Formula result = 68,510

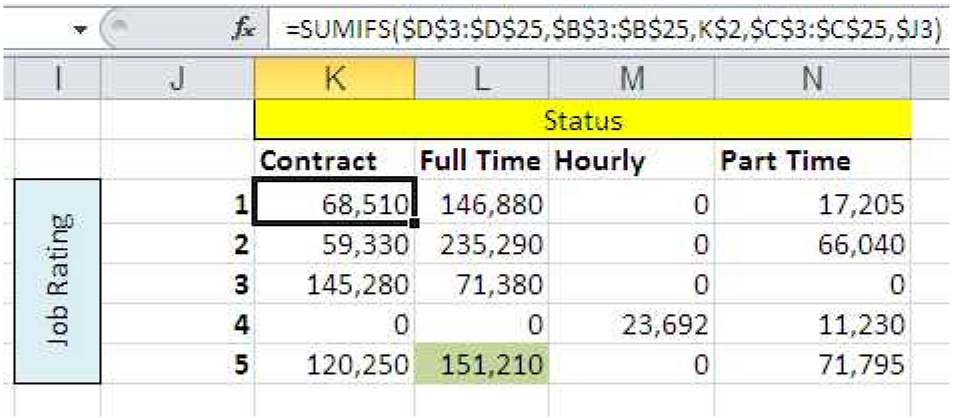
Note about Partial Absolute Referencing

All the criteria ranges have Absolute references.
The Status criteria is partially Absolute where the **row** is fixed (K$2).
The Job rating criteria is partially Absolute where the **column** is fixed ($J3).

The full formula is:

=SUMIFS($D$3:$D$25,$B$3:$B$25,K$2,$C$3:$C$25,$J3)

It can be Autofilled or copied down and across to fill the table as follows.

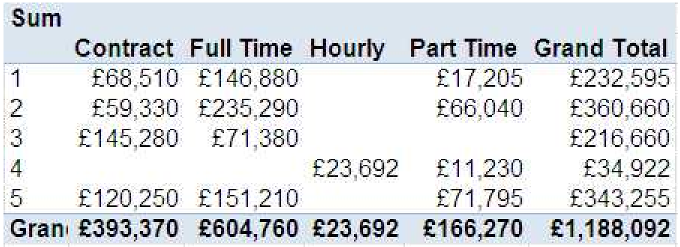


=SUMIFS($D$3:$D$25,$B$3:$B$25,K$2,$C$3:$C$25,$J3)

I	J	K	L	M	N
		Status			
		Contract	Full Time	Hourly	Part Time
Job Rating	1	68,510	146,880	0	17,205
	2	59,330	235,290	0	66,040
	3	145,280	71,380	0	0
	4	0	0	23,692	11,230
	5	120,250	151,210	0	71,795

Creating Tabulated data with a Pivot Table

The same tabulated data created using Sumifs functions can be created with a Pivot Table.



Sum					
	Contract	Full Time	Hourly	Part Time	Grand Total
1	£68,510	£146,880		£17,205	£232,595
2	£59,330	£235,290		£66,040	£360,660
3	£145,280	£71,380			£216,660
4			£23,692	£11,230	£34,922
5	£120,250	£151,210		£71,795	£343,255
Gran	**£393,370**	**£604,760**	**£23,692**	**£166,270**	**£1,188,092**

Row Labels as the Job Rating
Column labels as Status
Value as Salary (with Currency Number format)

Whereas the Pivot Table needs to be refreshed if there is a change in the data, the Sumifs table will update automatically.

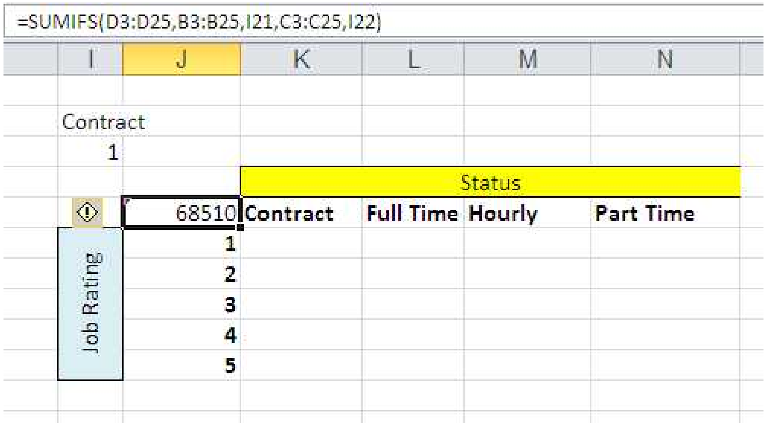
Creating Tabulated data with a Data Table

A third method to create the same tabulated data is via a Data Table. This method uses the same Sumifs formula but avoids the need for Absolute and Partial Absolute Referencing.

First type the Sumifs formula at the top left

To creating the Data Table

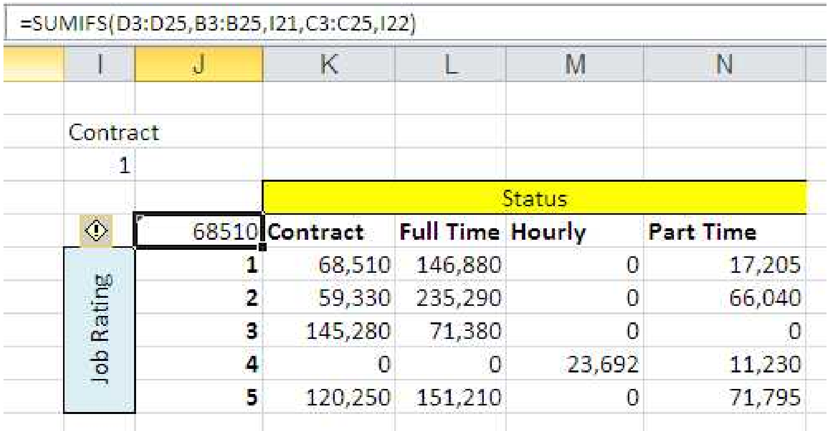
1. Create the border labels for the table.
2. Create a Row and Column input cells. Type the word Contract into the Row input cell and 1 into the Column input cell.
3. Type the SUMIFS formula at the top left blank cell that intersects the borders referring to the input cells for the two criteria.



=SUMIFS(D3:D25,B3:B25,I21,C3:C25,I22)

I	J	K	L	M	N
Contract					
1					
		Status			
	68510	Contract	Full Time	Hourly	Part Time
Job Rating	1				
	2				
	3				
	4				
	5				

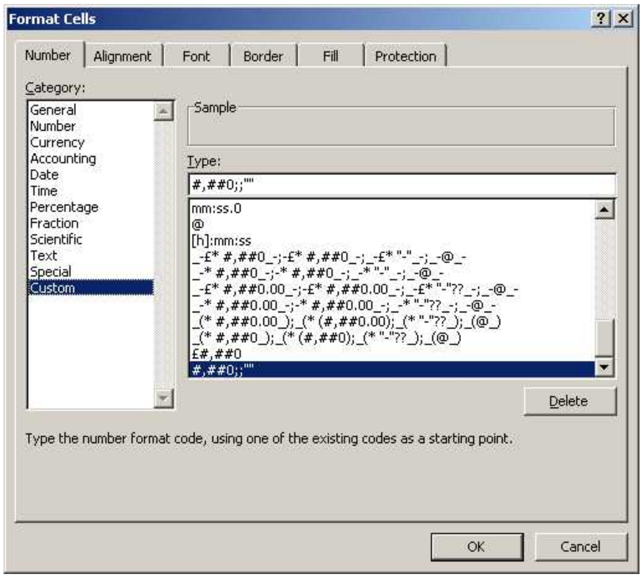
4. Highlight the table including the formula and borders.
5. Select Data, What-IF Analysis, Data Table
6. For the Row Input click on the word Contract (I21) and for the Column Input click on the Job Rating 1 (I22)



=SUMIFS(D3:D25,B3:B25,I21,C3:C25,I22)

I	J	K	L	M	N
Contract					
1					
		Status			
	68510	Contract	Full Time	Hourly	Part Time
Job Rating	1	68,510	146,880	0	17,205
	2	59,330	235,290	0	66,040
	3	145,280	71,380	0	0
	4	0	0	23,692	11,230
	5	120,250	151,210	0	71,795

Finally the input cell values can be cleared and the zeros supressed with the Custom format: #,##0;;””

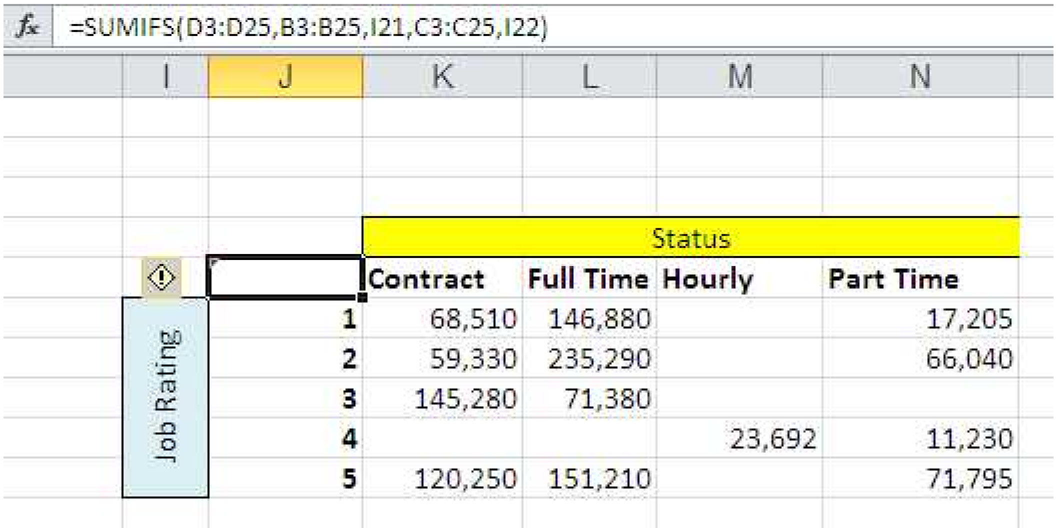


Category: Custom

Type: #,##0;;""

Type the number format code, using one of the existing codes as a starting point.

This custom format displays numbers with embedded commas, no decimal places and zero values as blank.



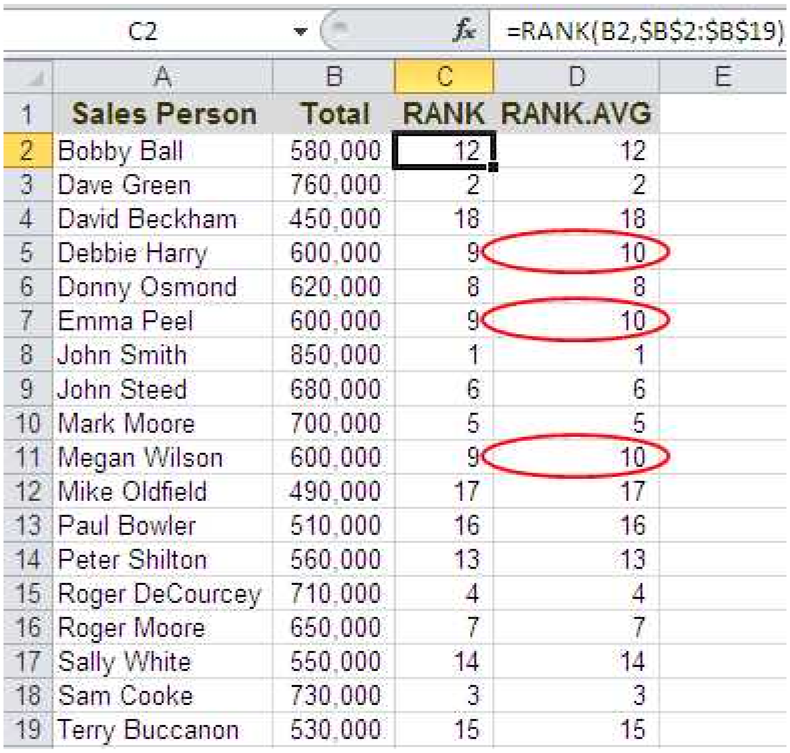
```
=SUMIFS(D3:D25,B3:B25,I21,C3:C25,I22)
```

I	J	K	L	M	N
		Status			
		Contract	Full Time	Hourly	Part Time
Job Rating	1	68,510	146,880		17,205
	2	59,330	235,290		66,040
	3	145,280	71,380		
	4			23,692	11,230
	5	120,250	151,210		71,795

Using Statistical functions MEDIAN, RANK, MODE, LARGE, SMALL

RANK and RANK.AVG Function

Calculating the ranking order for values in a range can be helpful. For example if you have a worksheet containing annual sales figures for salespeople you can rank them in order. If there are duplicate values there are two different ways of handling the ranking.



C2 =RANK(B2,$B$2:$B$19)

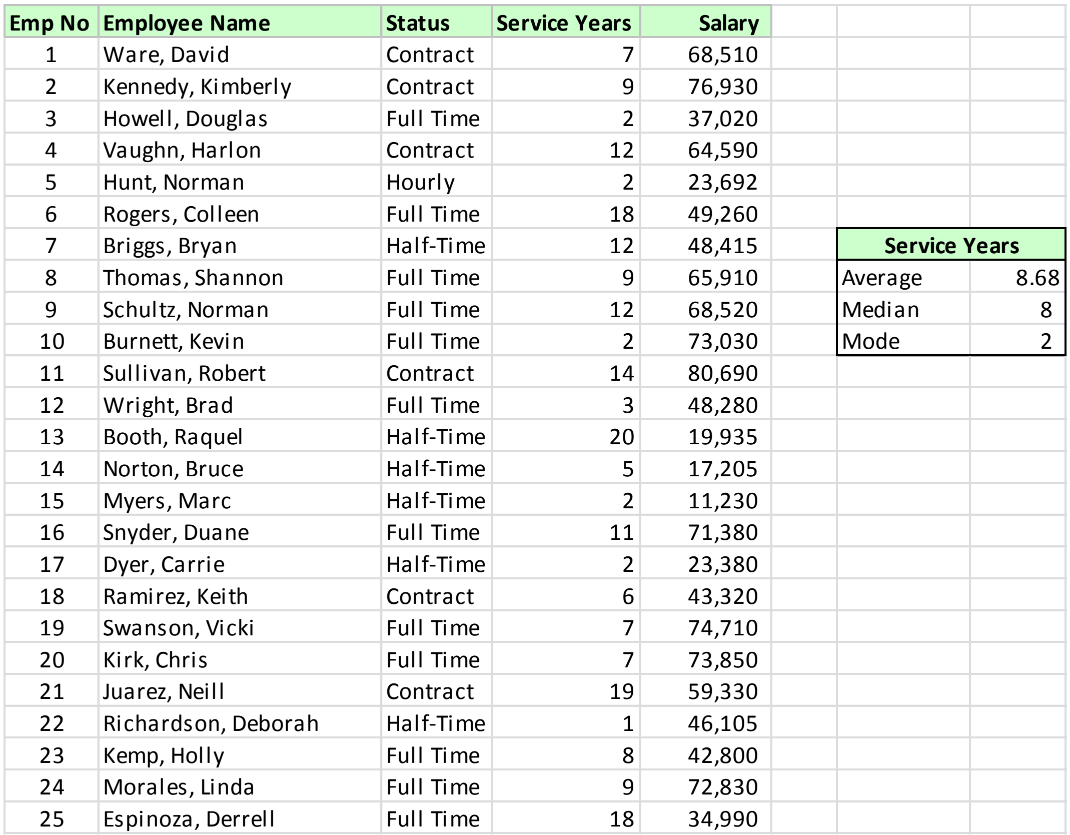
	A	B	C	D	E
1	Sales Person	Total	RANK	RANK.AVG	
2	Bobby Ball	580,000	12	12	
3	Dave Green	760,000	2	2	
4	David Beckham	450,000	18	18	
5	Debbie Harry	600,000	9	10	
6	Donny Osmond	620,000	8	8	
7	Emma Peel	600,000	9	10	
8	John Smith	850,000	1	1	
9	John Steed	680,000	6	6	
10	Mark Moore	700,000	5	5	
11	Megan Wilson	600,000	9	10	
12	Mike Oldfield	490,000	17	17	
13	Paul Bowler	510,000	16	16	
14	Peter Shilton	560,000	13	13	
15	Roger DeCourcey	710,000	4	4	
16	Roger Moore	650,000	7	7	
17	Sally White	550,000	14	14	
18	Sam Cooke	730,000	3	3	
19	Terry Buccanon	530,000	15	15	

In the above example the RANK function calculates there is a 3 way tie for 9th place. The next in rank will be in 12th place.

Using the RANK.AVG function the average rank is calculates the tie to have a rank of 10 (the average of 9,10 and 11).

MEDIAN function

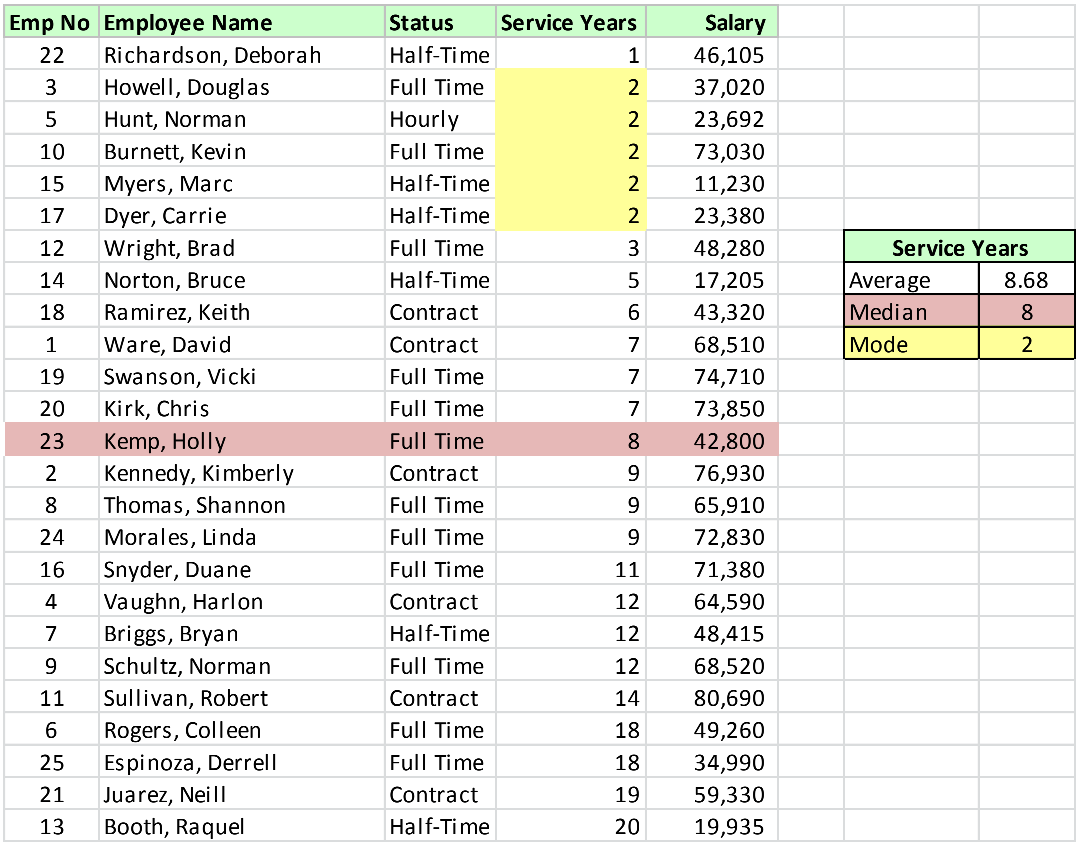
In the example below of 25 employees their Average year’s service is compared with the Median service. The average (also referred to as arithmetic mean) is 8.68 after formatting to 2 decimal places whereas the Median is 8.



Emp No	Employee Name	Status	Service Years	Salary
1	Ware, David	Contract	7	68,510
2	Kennedy, Kimberly	Contract	9	76,930
3	Howell, Douglas	Full Time	2	37,020
4	Vaughn, Harlon	Contract	12	64,590
5	Hunt, Norman	Hourly	2	23,692
6	Rogers, Colleen	Full Time	18	49,260
7	Briggs, Bryan	Half-Time	12	48,415
8	Thomas, Shannon	Full Time	9	65,910
9	Schultz, Norman	Full Time	12	68,520
10	Burnett, Kevin	Full Time	2	73,030
11	Sullivan, Robert	Contract	14	80,690
12	Wright, Brad	Full Time	3	48,280
13	Booth, Raquel	Half-Time	20	19,935
14	Norton, Bruce	Half-Time	5	17,205
15	Myers, Marc	Half-Time	2	11,230
16	Snyder, Duane	Full Time	11	71,380
17	Dyer, Carrie	Half-Time	2	23,380
18	Ramirez, Keith	Contract	6	43,320
19	Swanson, Vicki	Full Time	7	74,710
20	Kirk, Chris	Full Time	7	73,850
21	Juarez, Neill	Contract	19	59,330
22	Richardson, Deborah	Half-Time	1	46,105
23	Kemp, Holly	Full Time	8	42,800
24	Morales, Linda	Full Time	9	72,830
25	Espinoza, Derrell	Full Time	18	34,990

Service Years	
Average	8.68
Median	8
Mode	2

To see how the Median is calculated it helps to sort the employees by the Service column.



Emp No	Employee Name	Status	Service Years	Salary
22	Richardson, Deborah	Half-Time	1	46,105
3	Howell, Douglas	Full Time	2	37,020
5	Hunt, Norman	Hourly	2	23,692
10	Burnett, Kevin	Full Time	2	73,030
15	Myers, Marc	Half-Time	2	11,230
17	Dyer, Carrie	Half-Time	2	23,380
12	Wright, Brad	Full Time	3	48,280
14	Norton, Bruce	Half-Time	5	17,205
18	Ramirez, Keith	Contract	6	43,320
1	Ware, David	Contract	7	68,510
19	Swanson, Vicki	Full Time	7	74,710
20	Kirk, Chris	Full Time	7	73,850
23	Kemp, Holly	Full Time	8	42,800
2	Kennedy, Kimberly	Contract	9	76,930
8	Thomas, Shannon	Full Time	9	65,910
24	Morales, Linda	Full Time	9	72,830
16	Snyder, Duane	Full Time	11	71,380
4	Vaughn, Harlon	Contract	12	64,590
7	Briggs, Bryan	Half-Time	12	48,415
9	Schultz, Norman	Full Time	12	68,520
11	Sullivan, Robert	Contract	14	80,690
6	Rogers, Colleen	Full Time	18	49,260
25	Espinoza, Derrell	Full Time	18	34,990
21	Juarez, Neill	Contract	19	59,330
13	Booth, Raquel	Half-Time	20	19,935

Service Years	
Average	8.68
Median	8
Mode	2

After sorting by Service Years you can see that the Median is the number in the middle of a set of numbers. There are 12 people with a service lower than 8 and 12 people with a service greater than 8 years.

The MODE function

Another statistical function is MODE. In the same example the Mode of the Service Years is 2. This is the most common service for the 25 employees. If there were other years which are equally common then MODE returns the first one searching from top to bottom. There is also new function to deal with multiple mode values called MODE.MULT. This is an Array function which will be mentioned in a later module on Array functions.

The LARGE and SMALL functions.

In addition to the MAX or MIN functions that calculate the highest and lowest value in a range, the functions LARGE and SMALL find the second, third or any given rank within a range of values. In the example above:

=LARGE(E2:E26,2) finds 76930 as the second highest salary.

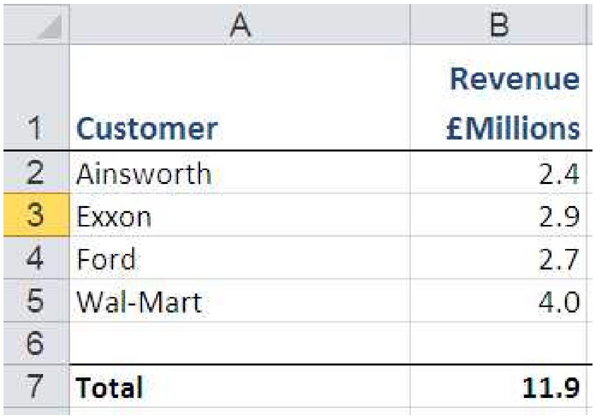
=SMALL(E2:E26,2) finds 17205, the second smallest salary.

These functions may be useful when you want to find out information without having to sort a list.

Use Maths functions: Round and related functions, the Mod function

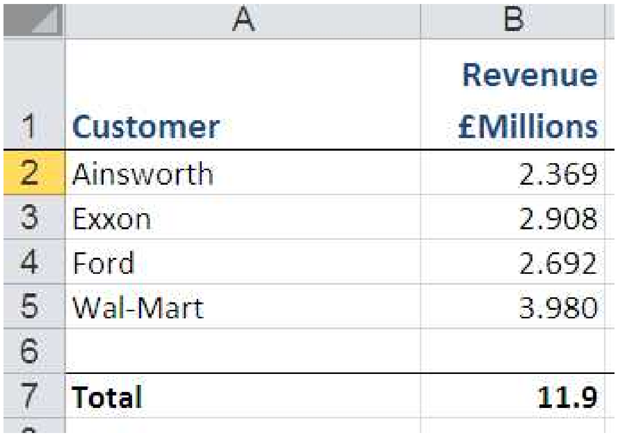
ROUND and related functions

In some situations, formatting may cause Excel to display inaccurate results such as when totaling numbers with decimal places. Because Excel uses the full precision of figures rather than the displayed figures the sum total may appear to be incorrect.



	A	B
1	Customer	Revenue £Millions
2	Ainsworth	2.4
3	Exxon	2.9
4	Ford	2.7
5	Wal-Mart	4.0
6		
7	**Total**	**11.9**

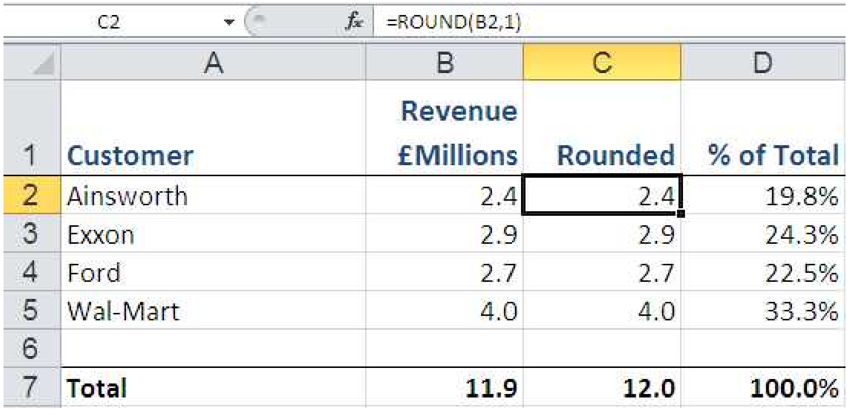
In the example someone might notice the figures should add up to 12.0 rather than 11.9 and wonder where the missing amount went. You could explain that 11.9 is the more accurate answer and show the precise figures.



	A	B
1	Customer	Revenue £Millions
2	Ainsworth	2.369
3	Exxon	2.908
4	Ford	2.692
5	Wal-Mart	3.980
6		
7	**Total**	**11.9**

But if you want the figures to actually add up precisely as displayed on screen you can use a ROUND function. In cell C2 enter the function =ROUND(B2,1)

The result is now exactly 2.4. After copying the formula down for the other customers the total now adds up to 12.0.



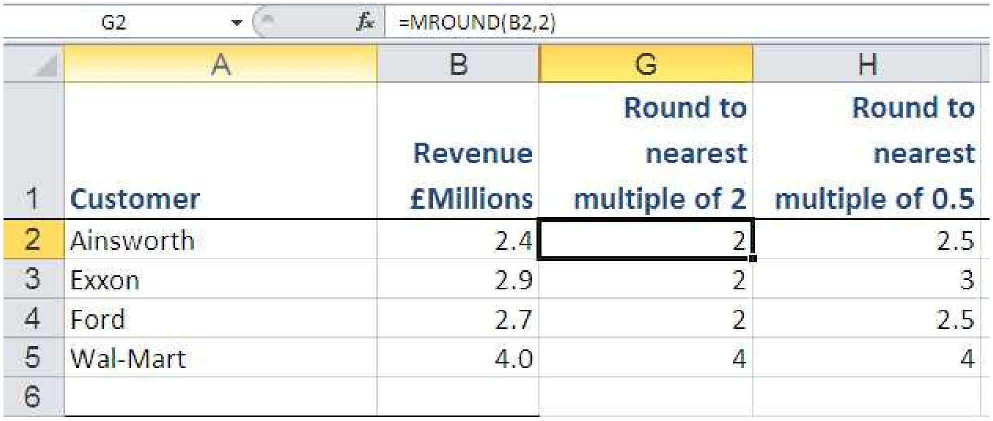
C2 =ROUND(B2,1)

	A	B	C	D
1	**Customer**	**Revenue £Millions**	**Rounded**	**% of Total**
2	Ainsworth	2.4	2.4	19.8%
3	Exxon	2.9	2.9	24.3%
4	Ford	2.7	2.7	22.5%
5	Wal-Mart	4.0	4.0	33.3%
6				
7	**Total**	**11.9**	**12.0**	**100.0%**

The ROUND function rounds up or down depending on the precise figure. For example when rounding to 1 decimal place 2.35 rounds up to 2.4, 2.34 rounds down to 2.3. To round all numbers up another function can be used called ROUNDUP. The function ROUNDDOWN is used to always round figures down.

Note in the above example the % values don't quite add up to 100.0%. This can be corrected by changing the formula from =B2/B$7 to =C2/B$7 so it then refers to the rounded values.

There is yet another similar function called MROUND which rounds number to a particular multiple. For example

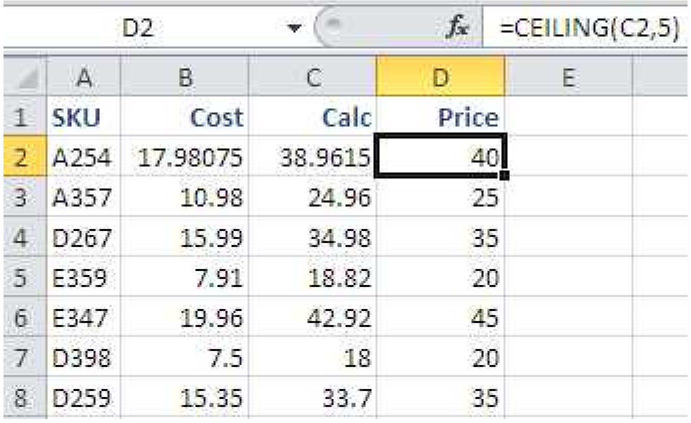


G2 =MROUND(B2,2)

	A	B	G	H
1	**Customer**	**Revenue £Millions**	**Round to nearest multiple of 2**	**Round to nearest multiple of 0.5**
2	Ainsworth	2.4	2	2.5
3	Exxon	2.9	2	3
4	Ford	2.7	2	2.5
5	Wal-Mart	4.0	4	4
6				

The revenues here are rounded to the nearest 2 million and in the next column to the nearest half million pounds.

Another related function is called CEILING. It can be used, for example, to round a price up to the nearest 5p amount.



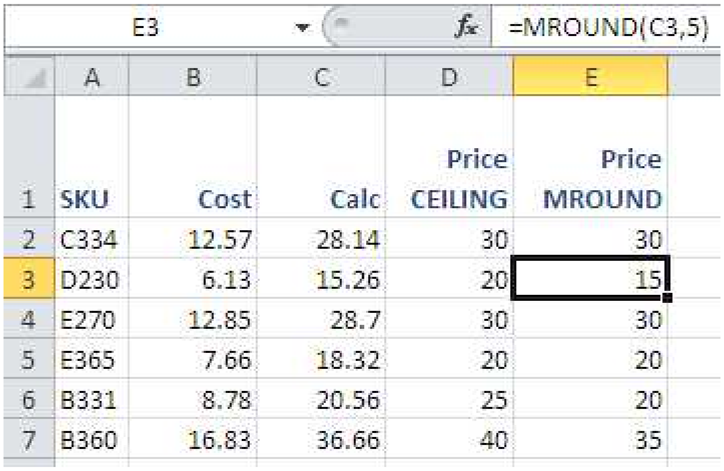
	A	B	C	D	E
1	SKU	Cost	Calc	Price	
2	A254	17.98075	38.9615	40	
3	A357	10.98	24.96	25	
4	D267	15.99	34.98	35	
5	E359	7.91	18.82	20	
6	E347	19.96	42.92	45	
7	D398	7.5	18	20	
8	D259	15.35	33.7	35	

The function in the Price column is

=CEILING(C2,5)

Where the level of significance is 5

The difference between this and using MROUND is that CEILING always rounds up whereas MROUND rounds to the nearest multiple. For example, 15.26 rounds to 20 and 15 respectively with CEILING and MROUND.



=MROUND(C3,5)

	A	B	C	D	E
1	SKU	Cost	Calc	Price CEILING	Price MROUND
2	C334	12.57	28.14	30	30
3	D230	6.13	15.26	20	15
4	E270	12.85	28.7	30	30
5	E365	7.66	18.32	20	20
6	B331	8.78	20.56	25	20
7	B360	16.83	36.66	40	35

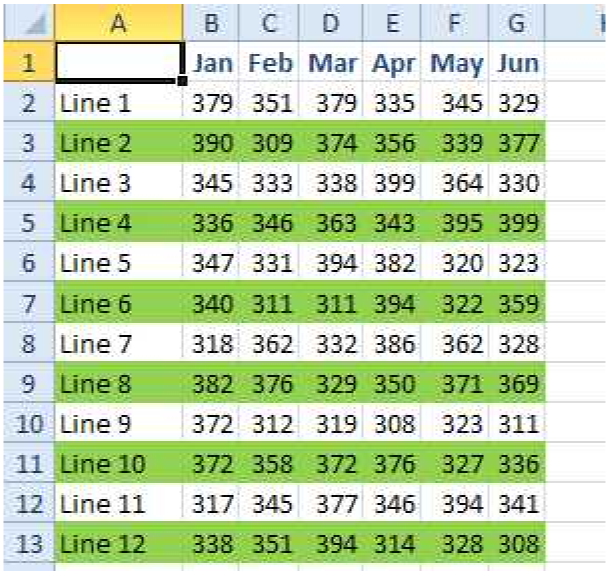
MOD function

At first there doesn’t seem to be a business need for the Mod function. What it does is return the remainder part of a number which is divided by another number. For example:

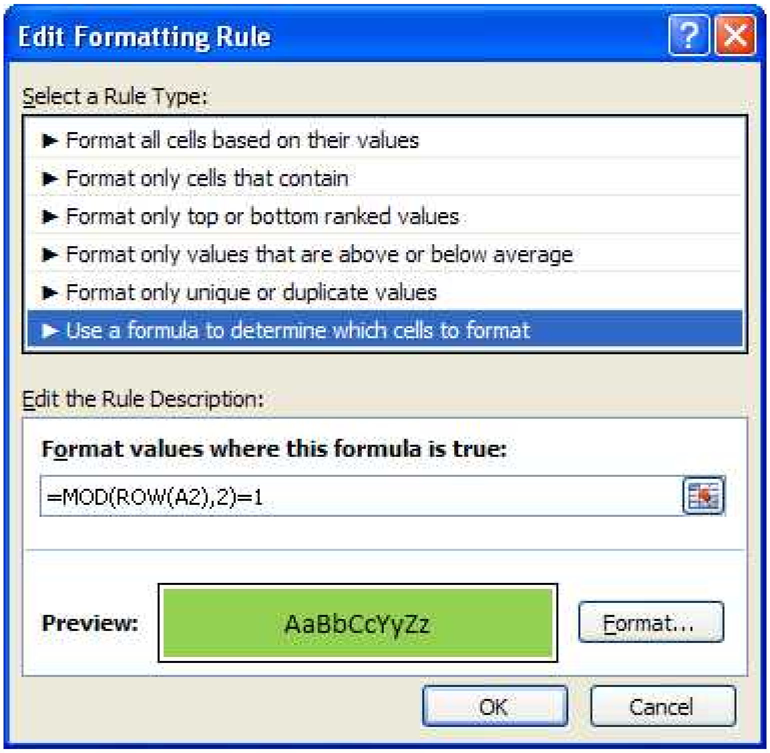
=Mod(8,3) returns 2

As 8 divided by 2 is 6 with a remainder of 2.

The function can be employed as a way of formatting worksheet tables in a banded style using Conditional Formatting.



	A	B	C	D	E	F	G
1		Jan	Feb	Mar	Apr	May	Jun
2	Line 1	379	351	379	335	345	329
3	Line 2	390	309	374	356	339	377
4	Line 3	345	333	338	399	364	330
5	Line 4	336	346	363	343	395	399
6	Line 5	347	331	394	382	320	323
7	Line 6	340	311	311	394	322	359
8	Line 7	318	362	332	386	362	328
9	Line 8	382	376	329	350	371	369
10	Line 9	372	312	319	308	323	311
11	Line 10	372	358	372	376	327	336
12	Line 11	317	345	377	346	394	341
13	Line 12	338	351	394	314	328	308



Edit Formatting Rule

Select a Rule Type:

- ► Format all cells based on their values
- ► Format only cells that contain
- ► Format only top or bottom ranked values
- ► Format only values that are above or below average
- ► Format only unique or duplicate values
- ► Use a formula to determine which cells to format

Edit the Rule Description:

Format values where this formula is true:

=MOD(ROW(A2),2)=1

Preview: AaBbCcYyZz

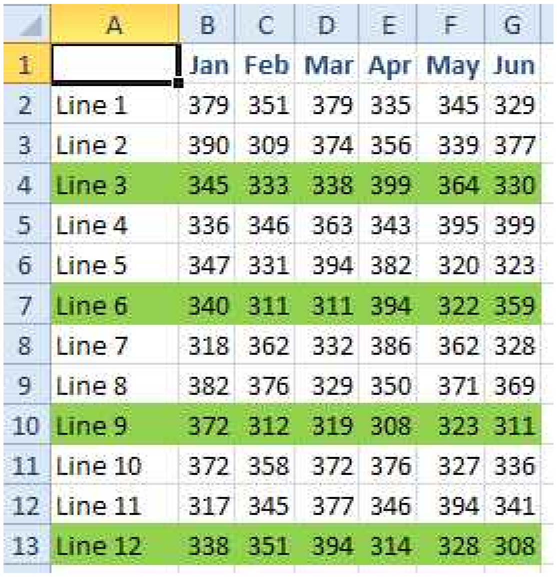
- Select the range of cells to format.
- From the Home tab select Conditional Formatting, New Rule.
- Select ‘Use a formula to determine to which cells to format’ then enter the formula:
- Type =MOD(ROW(A2),2)=1
- Select Format and change the fill colour to green, OK.

The result is only odd rows change colour because when their ROW number is divided by 2 gives a remainder of 1.

The same effect can be achieved more directly by converting the range to a Table and clicking Banded Rows. But this method allows control over how many rows are banded.

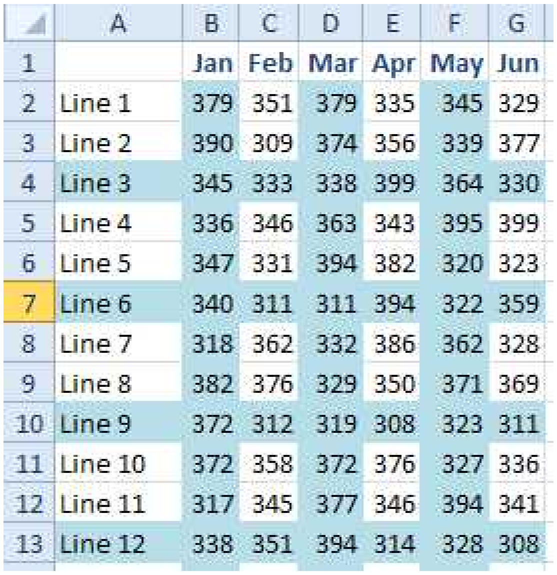
Try editing the formula to read

=MOD(ROW(A2),3)=1



	A	B	C	D	E	F	G
1		Jan	Feb	Mar	Apr	May	Jun
2	Line 1	379	351	379	335	345	329
3	Line 2	390	309	374	356	339	377
4	Line 3	345	333	338	399	364	330
5	Line 4	336	346	363	343	395	399
6	Line 5	347	331	394	382	320	323
7	Line 6	340	311	311	394	322	359
8	Line 7	318	362	332	386	362	328
9	Line 8	382	376	329	350	371	369
10	Line 9	372	312	319	308	323	311
11	Line 10	372	358	372	376	327	336
12	Line 11	317	345	377	346	394	341
13	Line 12	338	351	394	314	328	308

Now every third row becomes banded.



	A	B	C	D	E	F	G
1		Jan	Feb	Mar	Apr	May	Jun
2	Line 1	379	351	379	335	345	329
3	Line 2	390	309	374	356	339	377
4	Line 3	345	333	338	399	364	330
5	Line 4	336	346	363	343	395	399
6	Line 5	347	331	394	382	320	323
7	Line 6	340	311	311	394	322	359
8	Line 7	318	362	332	386	362	328
9	Line 8	382	376	329	350	371	369
10	Line 9	372	312	319	308	323	311
11	Line 10	372	358	372	376	327	336
12	Line 11	317	345	377	346	394	341
13	Line 12	338	351	394	314	328	308

A similar effect can be achieved for columns using the COLUMN function.

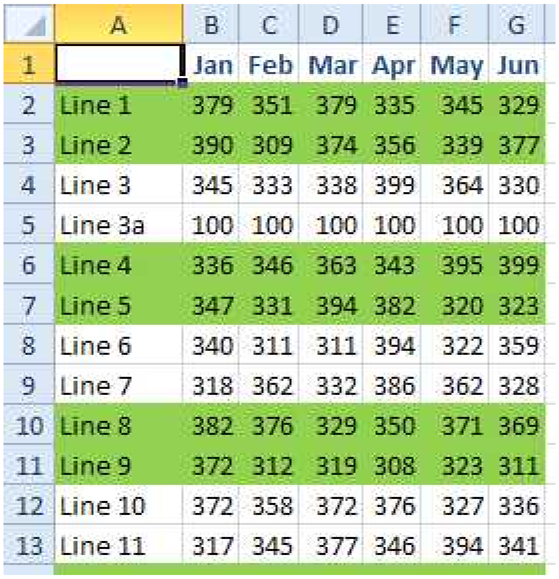
This time the formula is

=MOD(COLUMN(B2),2)=1

Other banding effects can be created

=MOD(ROW(A2),4)>1

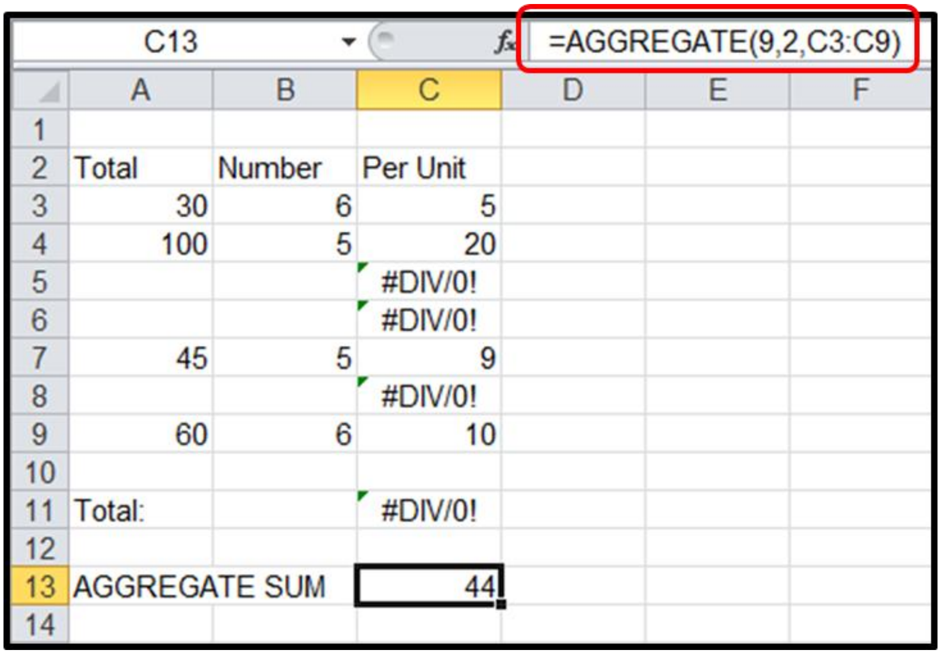
Produces bands every 3rd and 4th row where the remainder of the row divided by 4 is greater than 1.



	A	B	C	D	E	F	G
1		Jan	Feb	Mar	Apr	May	Jun
2	Line 1	379	351	379	335	345	329
3	Line 2	390	309	374	356	339	377
4	Line 3	345	333	338	399	364	330
5	Line 3a	100	100	100	100	100	100
6	Line 4	336	346	363	343	395	399
7	Line 5	347	331	394	382	320	323
8	Line 6	340	311	311	394	322	359
9	Line 7	318	362	332	386	362	328
10	Line 8	382	376	329	350	371	369
11	Line 9	372	312	319	308	323	311
12	Line 10	372	358	372	376	327	336
13	Line 11	317	345	377	346	394	341

Use the AGGREGATE function to sum data in ranges with errors

This function is a cousin of the SUBTOTAL function and only available in Excel 2010. One of the benefits of using the AGGREGATE function is that it ignores errors. For example:



C13 =AGGREGATE(9,2,C3:C9)

	A	B	C	D	E	F
1						
2	Total	Number	Per Unit			
3	30	6	5			
4	100	5	20			
5			#DIV/0!			
6			#DIV/0!			
7	45	5	9			
8			#DIV/0!			
9	60	6	10			
10						
11	Total:		#DIV/0!			
12						
13	AGGREGATE SUM		44			
14						

The formula in C11, SUM(C3:C9) returns an error because there are divide by zero errors in some of the cells in the sum range.

With the AGGREGATE function

=AGGREGATE(9,2,C3:C9)

The 9 refers to function number taken from the list shown below

Function_num	Function
1	AVERAGE
2	COUNT
3	COUNTA
4	MAX
5	MIN
6	PRODUCT
7	STDEV.S
8	STDEV.P
9	SUM
10	VAR.S

Function_num	Function
11	VAR.P
12	MEDIAN
13	MODE.SNGL
14	LARGE
15	SMALL
16	PERCENTILE.INC
17	QUARTILE.INC
18	PERCENTILE.EXC
19	QUARTILE.EXC

The 2 in the second argument is used here to ignore errors as well as subtotal functions. Other options from 0 – 7 allow hidden rows to be ignored and other variations. Selecting the hyperlink ‘Help on this function’ displays a full list of all options.

Use a variety of Financial functions such as PMT, FV, IRR

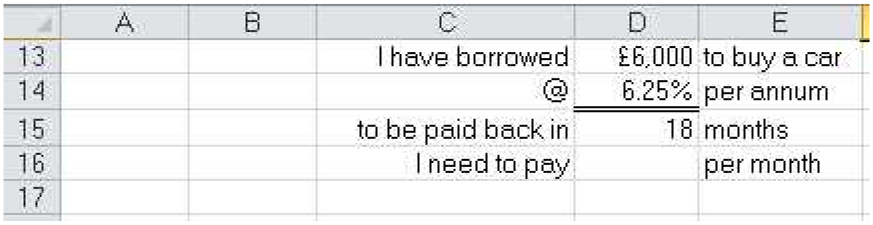
The most common use of Excel is to perform calculations involving money. Every day people make thousands of financial decisions based on figures in a spreadsheet. These decisions range from simple to complex. Can I afford to buy a new car in the next 18 months? Will a business result in a positive cash flow after 5 years?

PMT

The PMT (Payment Monthly Term) function returns the loan payment (principal plus interest) per period assuming constant payment amounts and a fixed interest rate.

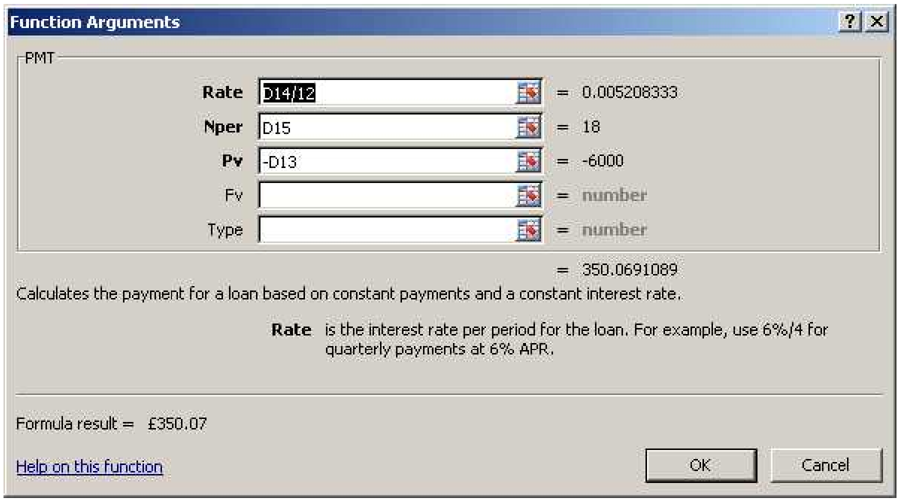
Syntax

PMT(rate,nper,pv,[fv],[type])



	A	B	C	D	E
13			I have borrowed	£6,000	to buy a car
14			@	6.25%	per annum
15			to be paid back in	18	months
16			I need to pay		per month
17					

For example suppose you borrow £6,000 to buy a car and plan to pay it off on monthly installments at an interest rate of 6.25% over 18 months. The PMT function calculates that the repayment to be made is £350.07 at the end of every month.



Function Arguments

PMT

Rate D14/12 = 0.005208333

Nper D15 = 18

Pv -D13 = -6000

Fv = number

Type = number

= 350.0691089

Calculates the payment for a loan based on constant payments and a constant interest rate.

Rate is the interest rate per period for the loan. For example, use 6%/4 for quarterly payments at 6% APR.

Formula result = £350.07

Note that Rate is the monthly interest rate so is divided by 12.
Nper is the number of monthly payments to be made.
Pv is the monthly payment amount and is entered as a negative because it is a payment.
Fv is a cash balance you may wish to attain after the last payment is made (assumed to be 0 if omitted).
Type determines when the monthly payment is to be paid (0 or omitted for end of month, 1 for beginning of month).

Note that £350.07 x 18 = £6,301.24 so the amount of interest to be paid in this example is £301.24.

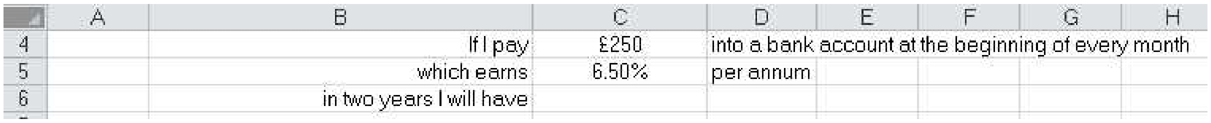
FV

The FV or Future Value function calculates the future value of an investment based on constant payments and a constant interest rate.

Syntax

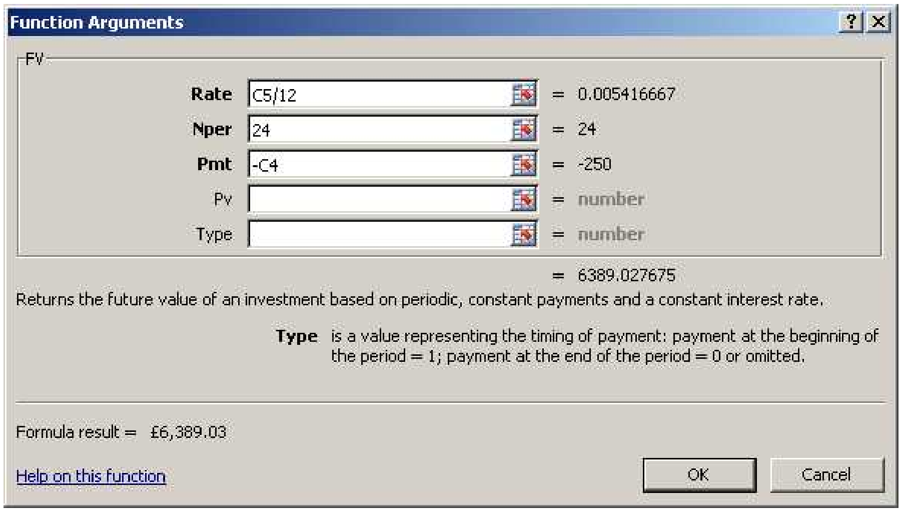
FV(rate,nper,pmt,[pv],[type])

For example if you pay £250 into the bank every month at an annual interest rate of 6.5% for 2 years what will be the future value of your investment?



	A	B	C	D	E	F	G	H
4		If I pay	£250	into a bank account at the beginning of every month				
5		which earns	6.50%	per annum				
6		in two years I will have						

In C6 create the following FV function by selecting Formulas, Financial, FV



FV

Rate	C5/12	= 0.005416667
Nper	24	= 24
Pmt	-C4	= -250
Pv		= number
Type		= number

= 6389.027675

Returns the future value of an investment based on periodic, constant payments and a constant interest rate.

Type is a value representing the timing of payment: payment at the beginning of the period = 1; payment at the end of the period = 0 or omitted.

Formula result = £6,389.03

As before

Rate is the monthly interest rate

Nper is the number of months 24/12

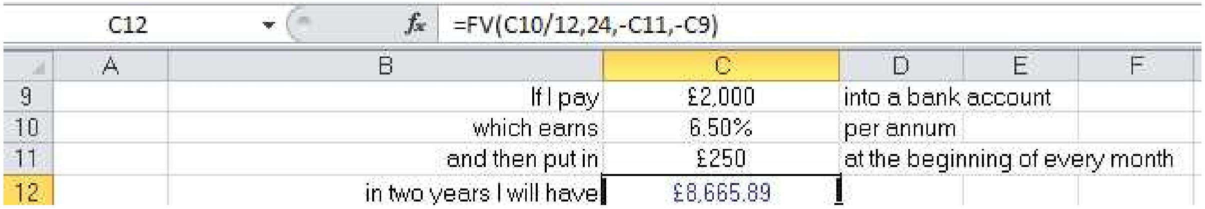
Pmt is -£250 as it is a payment

PV is an optional lump-sum amount that is paid at the beginning of the investment. In this case it is 0.

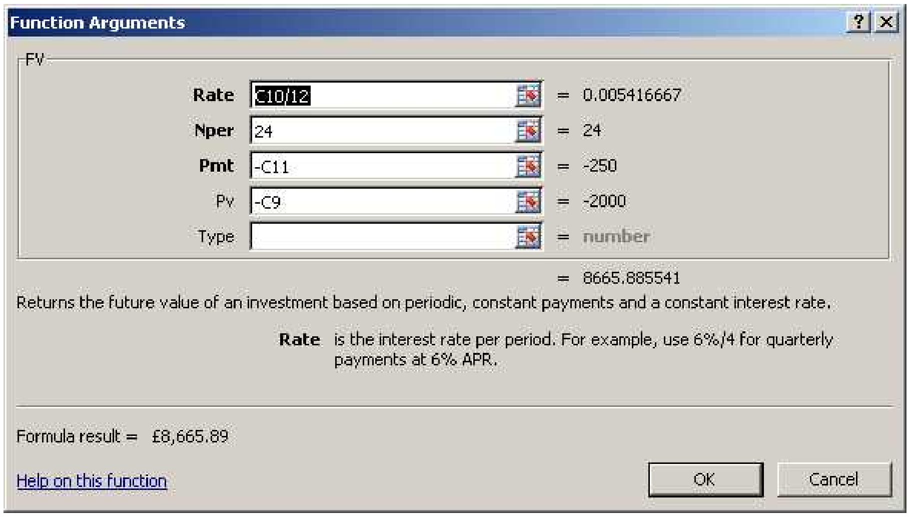
Type is again for when the monthly payment is paid (End of the month if omitted).

This function =FV(C5/12,24,-C4) returns £6,389.03 as the future value of the investment.

Now suppose you invest £2,000 as a lump-sum as well as making regular payments at the end of each month for 2 years.


C12 =FV(C10/12,24,-C11,-C9)

	A	B	C	D	E	F
9		If I pay	£2,000	into a bank account		
10		which earns	6.50%	per annum		
11		and then put in	£250	at the beginning of every month		
12		in two years I will have	£8,665.89			



Rate C10/12 = 0.005416667

Nper 24 = 24

Pmt -C11 = -250

Pv -C9 = -2000

Type = number

= 8665.885541

Returns the future value of an investment based on periodic, constant payments and a constant interest rate.

Rate is the interest rate per period. For example, use 6%/4 for quarterly payments at 6% APR.

Formula result = £8,665.89

Note again that the *Pv* and *Pmt* are entered as negative values because they are both payments.

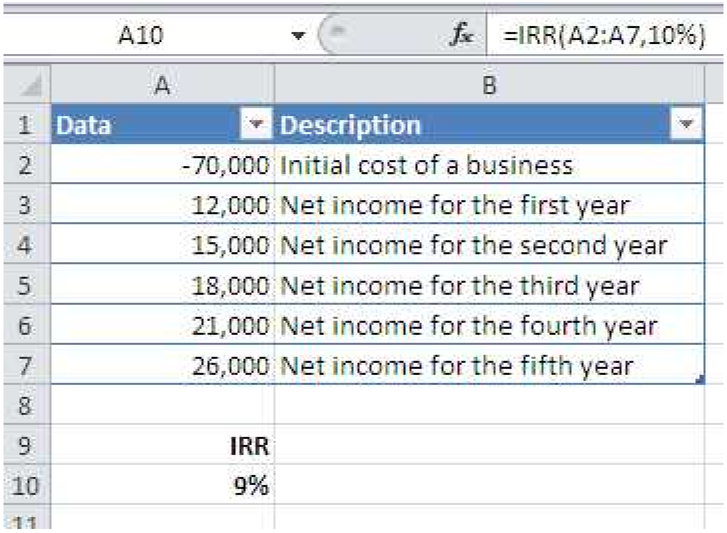
This time the FV function returns £8,665.89 which now includes the initial lump-sum with interest as well as the monthly payments and their interest.

IRR

Another useful financial function for businesses is the Internal Rate of Return IRR. This function calculates the rate of return on a series of regular cash flow income (positive values) and payments (negative values).

Syntax

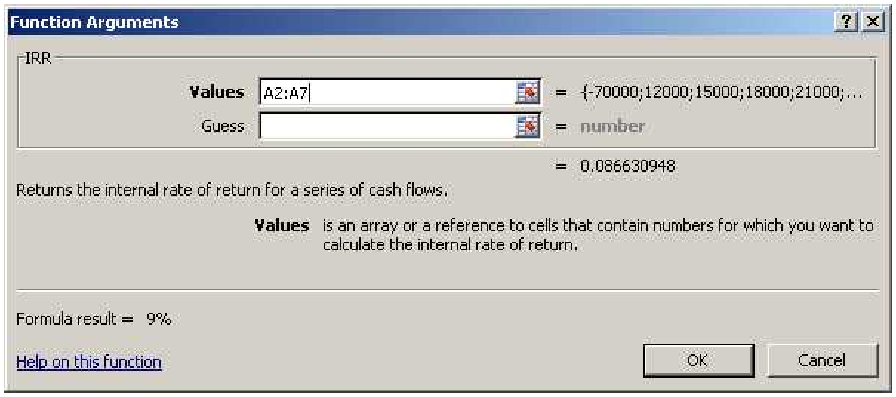
IRR(values, [guess])



A10 =IRR(A2:A7,10%)

	A	B
1	Data	Description
2	-70,000	Initial cost of a business
3	12,000	Net income for the first year
4	15,000	Net income for the second year
5	18,000	Net income for the third year
6	21,000	Net income for the fourth year
7	26,000	Net income for the fifth year
8		
9	IRR	
10	9%	

In the example above the IRR is 9% for the income amounts shown and the initial cost payment of 70,000.



Function Arguments

IRR

Values A2:A7 = {-70000;12000;15000;18000;21000;...

Guess = number

= 0.086630948

Returns the internal rate of return for a series of cash flows.

Values is an array or a reference to cells that contain numbers for which you want to calculate the internal rate of return.

Formula result = 9%

Help on this function

Values: the range of payment and income values.

Guess: is optional and set to 10% if omitted.

Note: A #NUM! error message appears if an answer could not be found after 20 cycle calculations. In that case enter a guess for the IRR %.

Unit 3: Date & Text Functions

In this unit you will learn how to:

- Find smarter ways to calculate dates and times using TODAY, NETWORKDAYS, WORKDAY and DATEDIF
- Use Text functions UPPER, PROPER, LEFT, RIGHT, LEN, MID, SEARCH and FIND
- Use TYPE to identify the data type of cell contents
- Use TRIM to remove excess spaces in cells

Calculating Dates and Times using TODAY, NETWORKDAYS, WORKDAY and DATEDIF

TODAY

The TODAY function displays the current date. Because it updates to show the new current date it is often used for date calculations that change with time such as a person’s age.

Syntax

=TODAY()

NETWORKDAYS

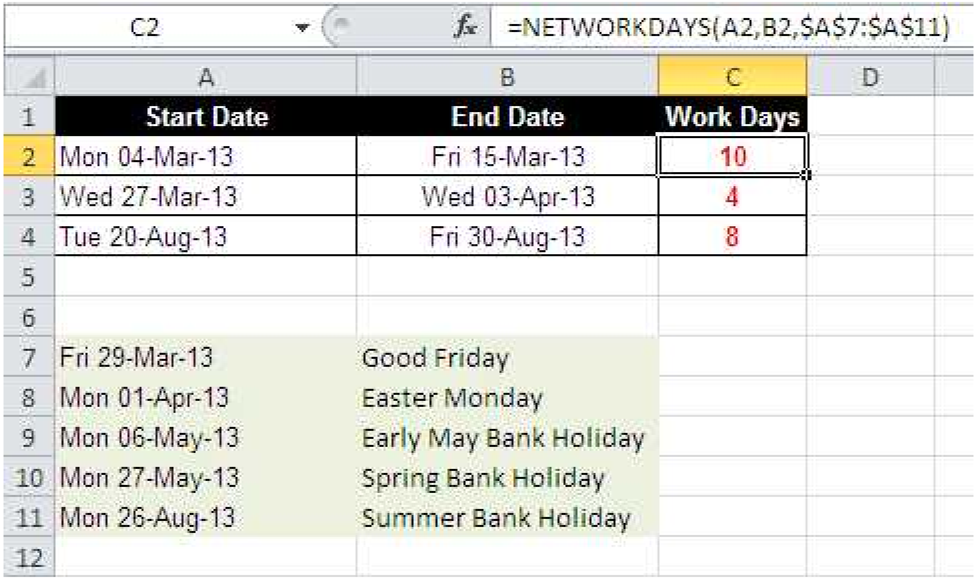
This function calculates the number of working days (not weekend dates) between two dates.

Syntax

=NETWORKDAYS(StartDate,EndDate,Holidays)

Holidays is a list of dates which will be excluded from the calculation such as public holidays. They can be entered as range of cells or as a list “25/12/12”,”01/”01/13”etc.

This example calculates the net working days between two dates for employees taking annual leave.



C2 =NETWORKDAYS(A2,B2,$A$7:$A$11)

	A	B	C	D
1	Start Date	End Date	Work Days	
2	Mon 04-Mar-13	Fri 15-Mar-13	10	
3	Wed 27-Mar-13	Wed 03-Apr-13	4	
4	Tue 20-Aug-13	Fri 30-Aug-13	8	
5				
6				
7	Fri 29-Mar-13	Good Friday		
8	Mon 01-Apr-13	Easter Monday		
9	Mon 06-May-13	Early May Bank Holiday		
10	Mon 27-May-13	Spring Bank Holiday		
11	Mon 26-Aug-13	Summer Bank Holiday		
12				

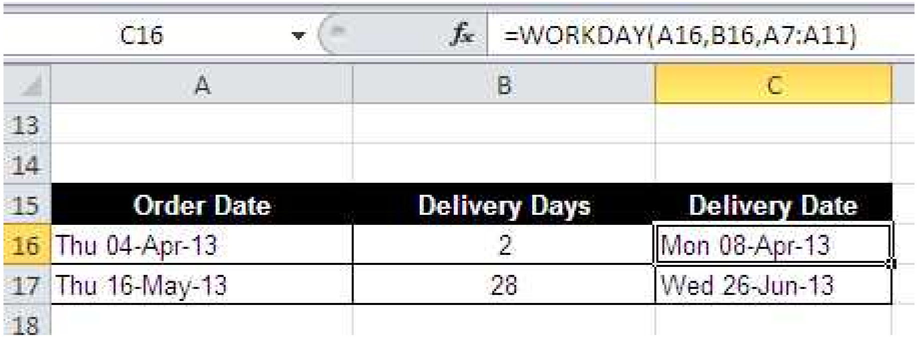
WORKDAY

This function calculates a future or past date based on a starting date and a specified number of working days. One of its uses could be to calculate invoice due dates or delivery dates from an order date.

Syntax

=WORKDAY(StartDate,Days,Holidays)

In this example the function calculates a delivery date based on an order date and an estimate of the number of days till delivery.



C16 =WORKDAY(A16,B16,A7:A11)

	A	B	C
13			
14			
15	Order Date	Delivery Days	Delivery Date
16	Thu 04-Apr-13	2	Mon 08-Apr-13
17	Thu 16-May-13	28	Wed 26-Jun-13
18			

DATEDIF

Another useful function for calculating intervals between dates is DATEDIF. It is one of Excel’s mysteries why DATEDIF does not appear in the drop-down list of functions. You must always enter it manually. The function originates from Lotus 1-2-3 and is no longer documented in Excel Help.

DATEDIF is useful for calculating the number of days, months and years between two dates.

Syntax

=DATEDIF(FirstDate,SecondDate,"Interval")

FirstDate must be earlier than *SecondDate* or an error is returned.

Interval is as follows:

"d"	Days between the two dates.
"m"	Months between the two dates.
"y"	Years between the two dates.
"yd"	Days between the dates, as if the dates were in the same year.
"ym"	Months between the dates, as if the dates were in the same year.
"md"	Days between the two dates, as if the dates were in the same month and year.

For example suppose you want to know someone’s age in years. The formula using DateDif would be:

=DATEDIF(C8,TODAY(),"y")

To include the number of months type:

=DATEDIF(C8,TODAY(),"ym")

And to include te number of days:

=DATEDIF(C8,TODAY(),"yd")

Putting these together the age in years, months and days would be:

```
=DATEDIF(C8,TODAY(),"y")&" Years, "&DATEDIF(C8,TODAY(),"ym")&" Months and "&DATEDIF(C8,TODAY(),"md")&" Days"
```

Time Calculations

Time values can be entered into Excel worksheets by typing them in the format:

hh:mm

For example 12:00 would mean mid day.

The time is based on a 24 hour clock so 1:00 would be 1 AM.

If you wish to type in 12 hour format then add the suffix AM or PM.

It is possible to include seconds by typing a second colon for example 12:00:10.

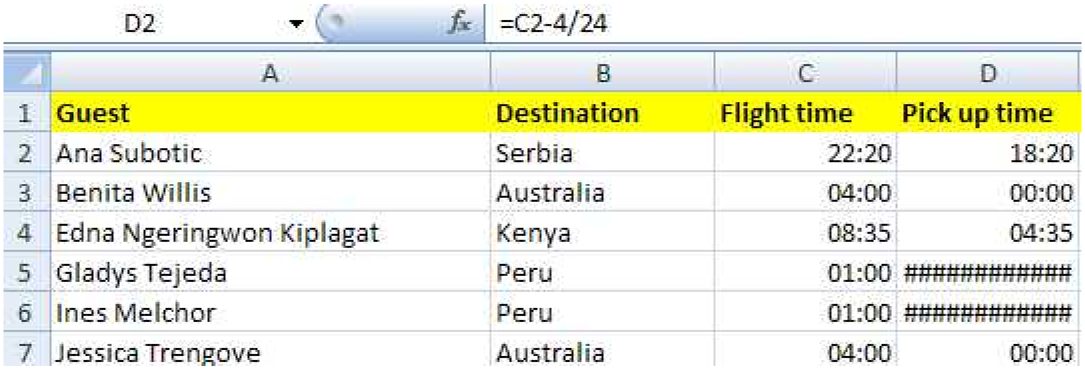
Time values can also be formatted afterwards in the usual way using

Format, Cells, Number, Time. Here are a list of some of the possible ways of formatting times:

Time format	**How it displays**
hh:mm	09:00
h:mm	9:00
hh:mm AM/PM	09:00 AM, 04:00 PM
hh:mm:ss	09:00:00
mm:ss.0	27:30.4 (27 mins, 30.4 seconds)
[hh]:mm	Allows for hour values greater than 24

When a time is entered into a cell with a valid time format Excel actually stores the value as a number between 0 and 1. Midnight would be stored as 0 and midday as 0.5. This can be seen by choosing General from the Format, Cells number format category, So if a time is formatted to a date format it would appear as 1/1/1900. As with dates, negative values are not allowed and fill the cell with ############## marks. This causes a problem when subtracting two times if the result is before midnight. For example: 03:00 – 4:00

In the example below the pick up time for the airport is 4 hours before the flight time.



D2 =C2-4/24

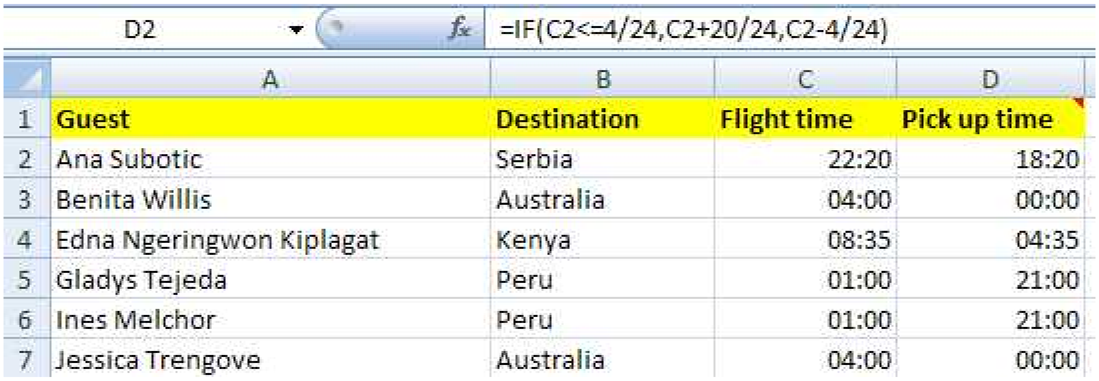
	A	B	C	D
1	**Guest**	**Destination**	**Flight time**	**Pick up time**
2	Ana Subotic	Serbia	22:20	18:20
3	Benita Willis	Australia	04:00	00:00
4	Edna Ngeringwon Kiplagat	Kenya	08:35	04:35
5	Gladys Tejeda	Peru	01:00	############
6	Ines Melchor	Peru	01:00	############
7	Jessica Trengove	Australia	04:00	00:00

As 4 hours is 4/24th part of a day the formula = C2-4/24 calculates the pick up time for all guests apart from those catching flight less than 4 hours after midnight.

To prevent an error occurring an IF statement can be used to make the adjustment.

=IF(C2<=4/24,C2+20/24,C2-4/24)

For flights in the early hours (12-4am) instead of subtracting 4 hours the IF statement adds 20 hours.



=IF(C2<=4/24,C2+20/24,C2-4/24)

	A	B	C	D
1	**Guest**	**Destination**	**Flight time**	**Pick up time**
2	Ana Subotic	Serbia	22:20	18:20
3	Benita Willis	Australia	04:00	00:00
4	Edna Ngeringwon Kiplagat	Kenya	08:35	04:35
5	Gladys Tejeda	Peru	01:00	21:00
6	Ines Melchor	Peru	01:00	21:00
7	Jessica Trengove	Australia	04:00	00:00

There is an alternative and shorter formula which does away with the need for the logical function. Simply add 1 day to the time of the flight. The formula would be:

=(C2+1)-4/24

Excel now has no problem with subtracting 4 hours as the resulting time is a positive number.

Creating Timesheets

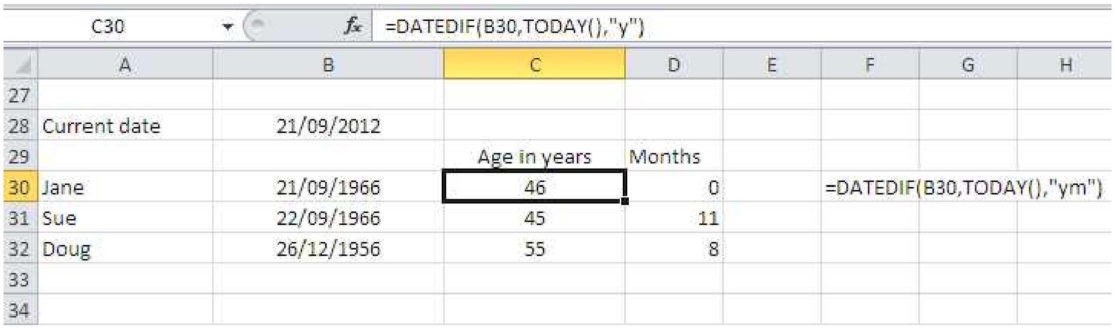
Times can be added up with a Sum function. This goes well until the total goes above 24 hours.

Start	End	Duration
09:00	18:30	09:30
09:00	17:00	08:00
08:00	18:00	10:00
	Total	**03:30**

To display the absolute number of hours you need to apply a custom format:

[hh]:mm

The correct result of 27:30 is then displayed for the timesheet.



C30 =DATEDIF(B30,TODAY(),"y")

	A	B	C	D	E	F	G	H
27								
28	Current date	21/09/2012						
29			Age in years	Months				
30	Jane	21/09/1966	46	0		=DATEDIF(B30,TODAY(),"ym")		
31	Sue	22/09/1966	45	11				
32	Doug	26/12/1956	55	8				
33								
34								

Use Text Functions

When data is copied from another source such as a Website you may need to manipulate or transform it in some way. The text functions mentioned in this unit can are designed to transform data. For example the UPPER function transforms text to upper case. The LEFT and RIGHT function truncates text to a specific number of characters to the left or right. Other functions such as FIND are used in combination with other text functions to locate the position of spaces or other characters within text. Other functions such as TRIM are used to clean out spaces from the beginning or end of text.

Often you will want to replace the original data with the transformed data. To do that you will need to create the transformed data containing the text function in a temporary column. The new data containing the function or functions can then be copied but then pasted back as **Values** onto the original data.

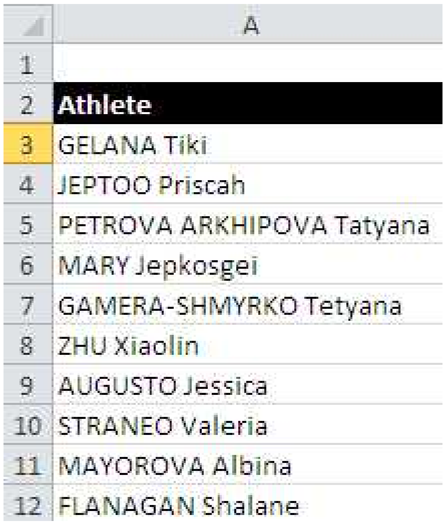
LEFT

The LEFT function displays a specific number of characters from the left hand side of a piece of text.

Syntax

=LEFT(Text,NumberOfCharacters)

Example:



	A
1	
2	**Athlete**
3	GELANA Tiki
4	JEPTOO Priscah
5	PETROVA ARKHIPOVA Tatyana
6	MARY Jepkosgei
7	GAMERA-SHMYRKO Tetyana
8	ZHU Xiaolin
9	AUGUSTO Jessica
10	STRANEO Valeria
11	MAYOROVA Albina
12	FLANAGAN Shalane

Suppose you import a list of athletes from the 2012 Olympic Website and find the Last name is in capitals and before the first name. You would rather display the names as First name then Last name in Proper case but there are 107 runners so you don't feel like retyping the names.

Use of the LEFT function wouldn't help because the Last names are all different lengths. You could display the Last name initials by typing in B3:

=LEFT(A3,1)

Copying the function down will display all athlete initials. But to display the Last names you need first to find the position of the space and use that to determine the length of Number of characters to display.

In Cell B3 you would type:

=LEFT(A3,FIND(" ",A3)-1)

This formula uses the text function FIND and will work for all the athletes with a single Last names.

FIND

The FIND function looks for a specific character or characters in the text within a cell and returns its character position counting from the left.

For example for GELANA tiki FIND(" ",A3) returns 7

Subtracting 1 results in the number of charaters for her Last name.

You can optionally start part way through the text by typing a startnumber. That might be useful if there is more than one space in a full name.

Syntax

=FIND(FindText,WithinText,[startnumber]))

So combing the LEFT and FIND functions results in a list of Last names only. There are two exceptions, PETROVA ARKHIPOVA Tatyana and DA SILVA Adriana Aparecida because there are effectively two Last names without a hyphens between them. They will have to be edited manually but 105 names are separated automatically with these text functions.

SEARCH

The SEARCH function is very similar to FIND..The main difference is that SEARCH is not case sensitive and it allows ‘wildcards’ to be used. A **?** stands for single character and a ***** stands for any number of characters.

For example

=SEARCH("* * ",A3)

returns the position 1if the text happens to contain 2 spaces, (3 names rather than 2) otherwise it returns #VALUE.

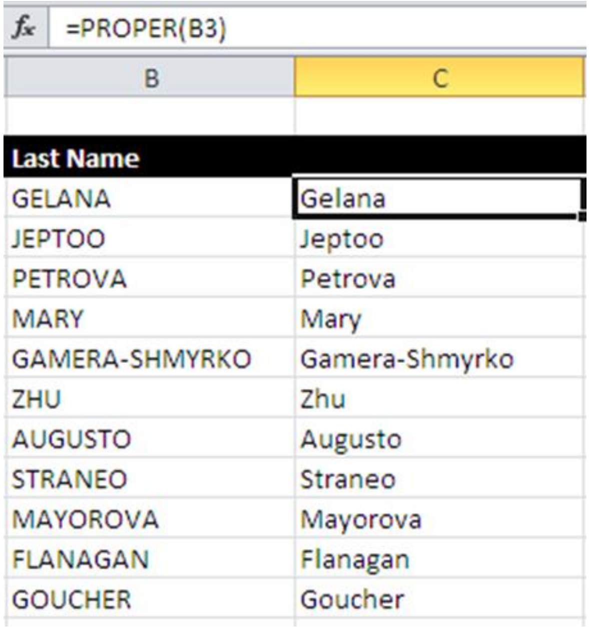
PROPER

This function capitalizes the first character of each work of the text within a cell.

Syntax

=PROPER(TextToConvert)

To convert text to Proper case simply type =PROPER(A3) in a blank column and copy down.



=PROPER(B3)

B	C
Last Name	
GELANA	Gelana
JEPTOO	Jeptoo
PETROVA	Petrova
MARY	Mary
GAMERA-SHMYRKO	Gamera-Shmyrko
ZHU	Zhu
AUGUSTO	Augusto
STRANEO	Straneo
MAYOROVA	Mayorova
FLANAGAN	Flanagan
GOUCHER	Goucher

UPPER and LOWER

These text functions convert all text in cell to upper or lower case.

Syntax

=UPPER(Text)

=LOWER(Text)

They may be useful when there is a mixture of case. They can also be wrapped around other functions, for example,

=PROPER(LEFT(A3,FIND(" ",A3)-1))

LEN

This function counts the number of characters in a piece of text including spaces and numbers.

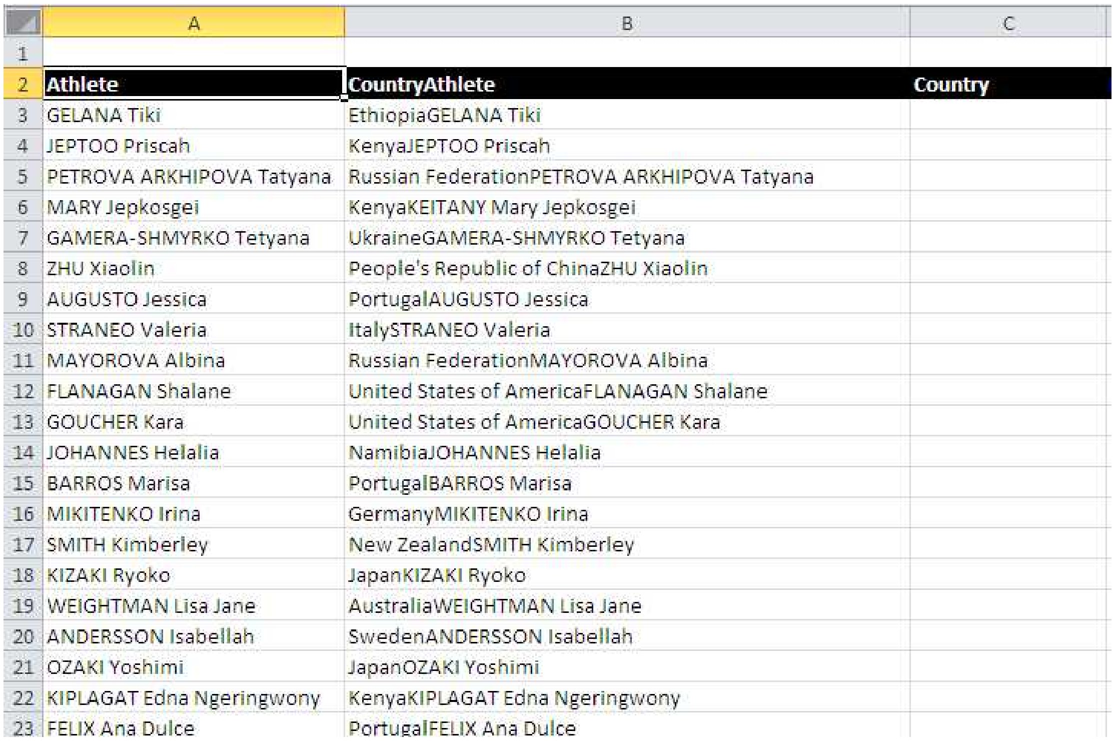
Syntax

=LEN(Text)

At first there may not seem to be a business need for counting thre number of characters in piece of text. But it can be useful for transforming data when used in combination with other text functions.

In the example below, suppose you wish to separate the country from the athlete name and display it in column C. Unfortunately there is no space or other delimiter after the country name to help you. This is because the data came originally from a flag on the Website next to each athlete name. So you the SEARCH function or even Excel’s powerful Text to Columns feature on the Data tab will not help you.

But if you have the athlete name already it should be possible to use a combination of the LEN and LEFT function to subtract extract the country name.



	A	B	C
1			
2	**Athlete**	**CountryAthlete**	**Country**
3	GELANA Tiki	EthiopiaGELANA Tiki	
4	JEPTOO Priscah	KenyaJEPTOO Priscah	
5	PETROVA ARKHIPOVA Tatyana	Russian FederationPETROVA ARKHIPOVA Tatyana	
6	MARY Jepkosgei	KenyaKEITANY Mary Jepkosgei	
7	GAMERA-SHMYRKO Tetyana	UkraineGAMERA-SHMYRKO Tetyana	
8	ZHU Xiaolin	People's Republic of ChinaZHU Xiaolin	
9	AUGUSTO Jessica	PortugalAUGUSTO Jessica	
10	STRANEO Valeria	ItalySTRANEO Valeria	
11	MAYOROVA Albina	Russian FederationMAYOROVA Albina	
12	FLANAGAN Shalane	United States of AmericaFLANAGAN Shalane	
13	GOUCHER Kara	United States of AmericaGOUCHER Kara	
14	JOHANNES Helalia	NamibiaJOHANNES Helalia	
15	BARROS Marisa	PortugalBARROS Marisa	
16	MIKITENKO Irina	GermanyMIKITENKO Irina	
17	SMITH Kimberley	New ZealandSMITH Kimberley	
18	KIZAKI Ryoko	JapanKIZAKI Ryoko	
19	WEIGHTMAN Lisa Jane	AustraliaWEIGHTMAN Lisa Jane	
20	ANDERSSON Isabellah	SwedenANDERSSON Isabellah	
21	OZAKI Yoshimi	JapanOZAKI Yoshimi	
22	KIPLAGAT Edna Ngeringwony	KenyaKIPLAGAT Edna Ngeringwony	
23	FELIX Ana Dulce	PortugalFELIX Ana Dulce	

Exercise

Use a combination of LEN and LEFT to create a formula to display Ethiopia in cell **C3**. Then copy down for the others.

Notice something went wrong for Valeria Straneo. Can you see why her country displays as Ital instead of Italy?

TYPE

Other information about data in a cell can be found using the TYPE function.

This function returns an integer value depending on the data type in the cell. The TYPE function returns one of the following:

1 = number

2 = text

4 = logical

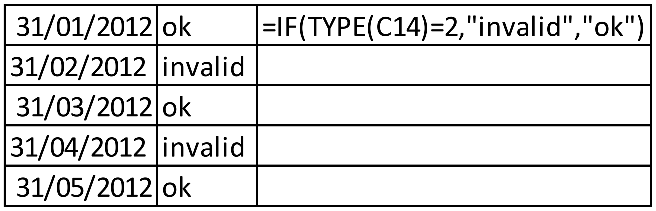
16 = error

64 = array

Syntax

=TYPE(value)

The function is usually used in combination with other functions. For example, here it is used to check for invalid dates. If a date has too many days Excel will convert it to text rather than a number.



31/01/2012	ok	=IF(TYPE(C14)=2,"invalid","ok")
31/02/2012	invalid	
31/03/2012	ok	
31/04/2012	invalid	
31/05/2012	ok	

TRIM

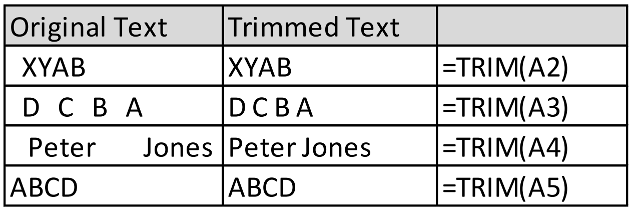
Sometimes a space is accidentally entered at the end of text and they later become unnoticed causing problems down the line when manipulating or pivoting data down the line. It only happens for text entries as trailing spaces are automatically removed when numbers or dates are typed into a cell.

Leading and trailing spaces are sometimes included also when importing data from other sources. To overcome these problems the TRIM function is used to remove the unwanted spaces.

Syntax

TRIM(Text)

While leading and trailing spaces are removed by the TRIM function those within text are not. However, multiple spaces are automatically reduced to a single space.



Original Text	Trimmed Text	
XYAB	XYAB	=TRIM(A2)
D C B A	D C B A	=TRIM(A3)
Peter Jones	Peter Jones	=TRIM(A4)
ABCD	ABCD	=TRIM(A5)

Exercise

Include the TRIM function into your formula to extract the country name so that trailing spaces in the data are removed. The function should now read:

=LEFT(B3,LEN(TRIM(B3))-LEN(TRIM(A3)))

CONCATENATE

This function is used to join separate pieces of text into one item.

Syntax

CONCATENATE(text1,text2,text3...)

As many as 255 pieces text can be joined using CONCATENATE.

Name 1	Name 2	Concatenated Text	
Peter	Rabbit	Peter Rabbit	=CONCATENATE(C2," ",D2)
Donald	Duck	Donald Duck	=CONCATENATE(C3," ",D3)
Micky	Mouse	Micky Mouse	=C4&" "&D4

Note that the same result can be achieved using the **&** operator to combine text from different cells.

Unit 4: Array formulas

In this unit, you will learn how to:

- Work with arrays in formulas
- Use the SUMPRODUCT function
- Create elegant formulas that appear to perform spreadsheet magic
- Compare SUMPRODUCT, SUMIFS with SUM arrays
- Use the TRANSPOSE function to switch row and column data

Understanding Array Formulas

If you do any computer programming, you've probably been exposed to the concept of an array. An *array* is simply a collection of items operated on collectively or individually. In Excel, an array can be one dimensional or two dimensional. These dimensions correspond to rows and columns. For example, a *one-dimensional array* can be stored in a range that consists of one row (a horizontal array) or one column (a vertical array). A *two-dimensional array* can be stored in a rectangular range of cells.

As you'll see, arrays need not be stored in cells. You can also work with arrays that exist only in Excel's memory which are known as array constants. You can then use an *array formula* to manipulate this information and return a result. An array formula can occupy multiple cells or reside in a single cell.

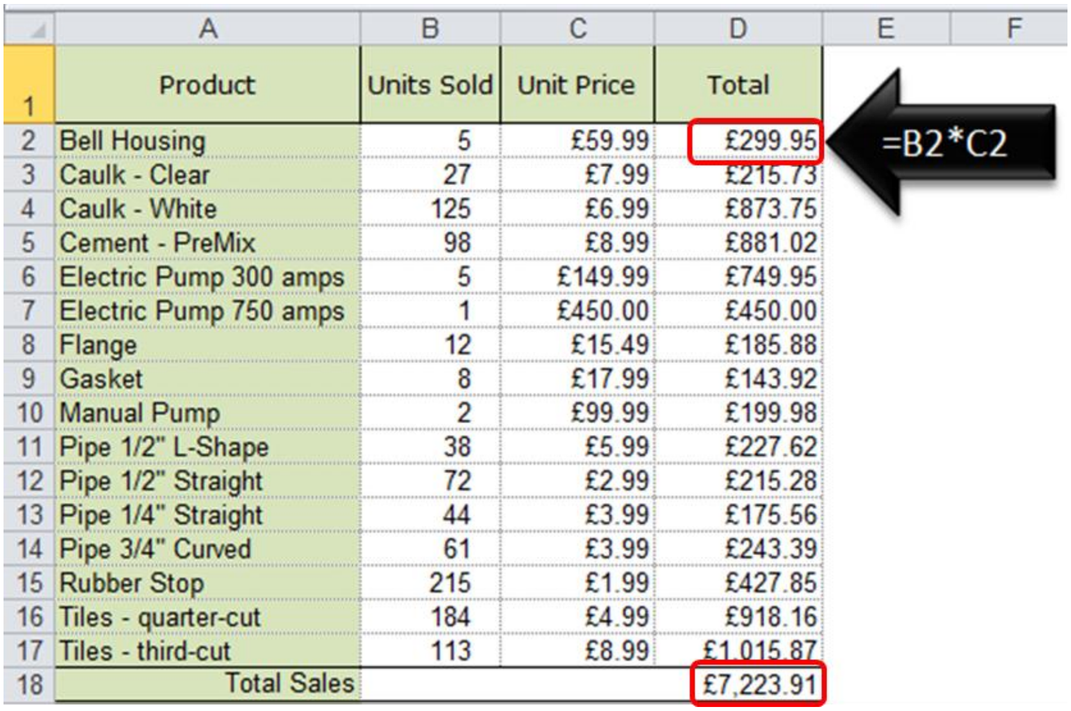
This section presents two array formula examples: an array formula that occupies multiple cells and another array formula that occupies only one cell.

A multicell array formula

The following table shows a simple worksheet set up to calculate product sales. Normally, you'd calculate the value in column D (total sales per product) with a formula such as the one that follows, and then you'd copy this formula down the column.

=B2*C2

After copying the formula down, the worksheet contains sixteen formulas in column D with a sales total formula in D18.



=B2*C2

	A	B	C	D	E	F
1	Product	Units Sold	Unit Price	Total		
2	Bell Housing	5	£59.99	£299.95		
3	Caulk - Clear	27	£7.99	£215.73		
4	Caulk - White	125	£6.99	£873.75		
5	Cement - PreMix	98	£8.99	£881.02		
6	Electric Pump 300 amps	5	£149.99	£749.95		
7	Electric Pump 750 amps	1	£450.00	£450.00		
8	Flange	12	£15.49	£185.88		
9	Gasket	8	£17.99	£143.92		
10	Manual Pump	2	£99.99	£199.98		
11	Pipe 1/2" L-Shape	38	£5.99	£227.62		
12	Pipe 1/2" Straight	72	£2.99	£215.28		
13	Pipe 1/4" Straight	44	£3.99	£175.56		
14	Pipe 3/4" Curved	61	£3.99	£243.39		
15	Rubber Stop	215	£1.99	£427.85		
16	Tiles - quarter-cut	184	£4.99	£918.16		
17	Tiles - third-cut	113	£8.99	£1,015.87		
18	Total Sales			£7,223.91		

An alternative method uses one formula (an array formula) to calculate all sixteen values in D2:D17. This single formula occupies sixteen cells and returns an array of sixteen values. This is also known as a *two dimensional array.*

To create a multicell array formula to perform the calculations, follow these steps:

1. Select a range to hold the results. In this case, the range is D2:D17. Because you can’t display more than one value in a single cell, sixteen cells are required to display the resulting array — so you select sixteen cells to make this array work.

2. Type the following formula:

=B2:B17*C2:C17

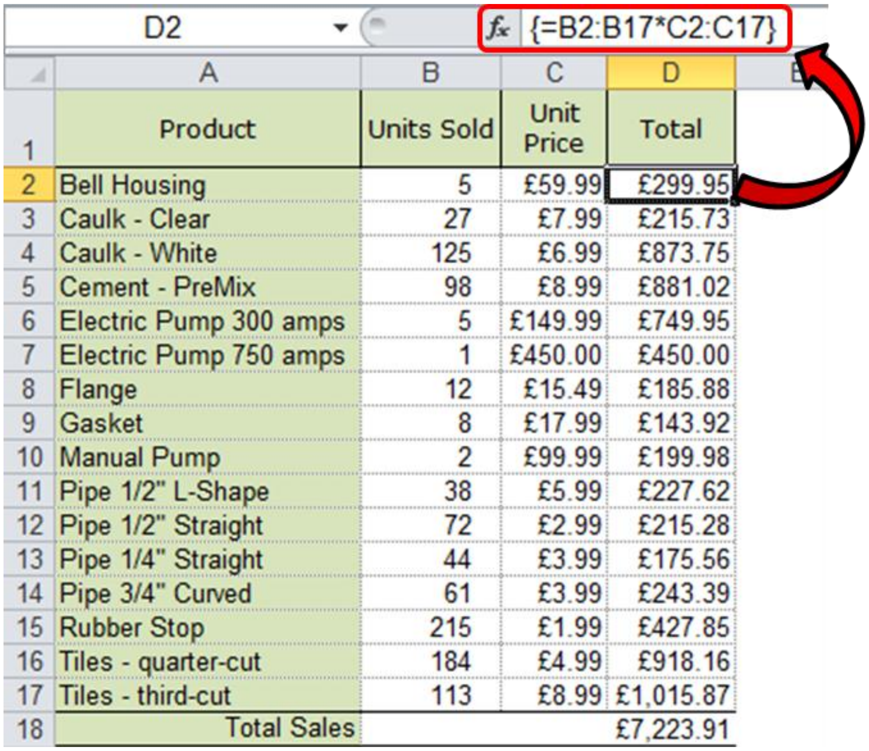
3. At this point you would normally press 'Enter' however, because this is an array formula you must press CTRL+SHIFT and then press Enter to activate the formula. This type of formula is also known as a CSE formula (**C**TRL+**S**HIFT+**E**NTER).

> Note: You can’t insert a multicell array formula into a range that has been designated a table (using Insert > Tables > Table). In addition, you can’t convert a range that contains a multicell array formula to a table.

The formula is entered into all sixteen selected cells. If you examine the Formula bar, you see the following:

{=B2:B17*C2:C17}

Excel places curly brackets around the formula to indicate that it’s an array formula.



	A	B	C	D
1	Product	Units Sold	Unit Price	Total
2	Bell Housing	5	£59.99	£299.95
3	Caulk - Clear	27	£7.99	£215.73
4	Caulk - White	125	£6.99	£873.75
5	Cement - PreMix	98	£8.99	£881.02
6	Electric Pump 300 amps	5	£149.99	£749.95
7	Electric Pump 750 amps	1	£450.00	£450.00
8	Flange	12	£15.49	£185.88
9	Gasket	8	£17.99	£143.92
10	Manual Pump	2	£99.99	£199.98
11	Pipe 1/2" L-Shape	38	£5.99	£227.62
12	Pipe 1/2" Straight	72	£2.99	£215.28
13	Pipe 1/4" Straight	44	£3.99	£175.56
14	Pipe 3/4" Curved	61	£3.99	£243.39
15	Rubber Stop	215	£1.99	£427.85
16	Tiles - quarter-cut	184	£4.99	£918.16
17	Tiles - third-cut	113	£8.99	£1,015.87
18	Total Sales			£7,223.91

This formula performs its calculations and returns a sixteen-item array. The array formula actually works with two other arrays, both of which happen to be stored in ranges. The values for the first array are stored in B2:B17 and the values for the second array are stored in C2:C17.

This array formula returns exactly the same values as these sixteen normal formulas entered into individual cells in D2:D17:

```
=B2*C2
=B3*C3
=B4*C4
=B5*C5
=B6*C6
=B7*C7
=B8*C8
=B9*C9
=B10*C10
=B11*C11
=B12*C12
=B13*C13
=B14*C14
=B15*C15
=B16*C16
=B17*C17
```

Using a single array formula rather than individual formulas does offer a few advantages:

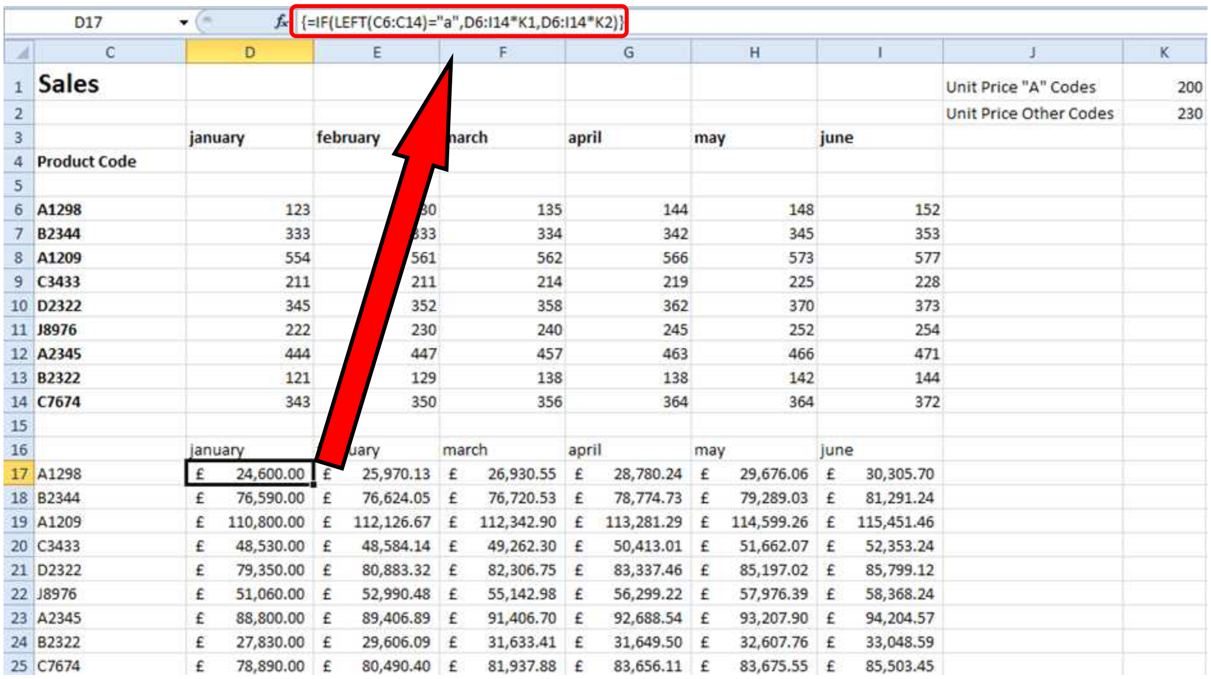
- It's a good way to ensure that all formulas in a range are identical.
- Using a multicell array formula makes it less likely that you'll overwrite a formula accidentally. You can't change one cell in a multicell array formula. Excel displays an error message if you attempt to do so.
- Using a multicell array formula will almost certainly prevent novices from tampering with your formulas.

Using a multicell array formula as described in the preceding list also has some potential disadvantages:

- It's impossible to insert a new row into the range. *But in some cases, the inability to insert a row is a positive feature.* For example, you might not want users to add rows because it would affect other parts of the worksheet.
- If you add new data to the bottom of the range, you need to modify the array formula to accommodate the new data.

Multi-cell array block formula

Here is another example of a muti-cell array that creates the array in a block rather than a single column. The array formula in range D17:I25 calculates the quantity values in range D6:I14 multiplied by a Unit Price of 200 for Product Codes starting with the letter A. For the other Product Codes the array multiplies by 230.



D17 {=IF(LEFT(C6:C14)="a",D6:I14*K1,D6:I14*K2)}

	C	D	E	F	G	H	I	J	K
1	**Sales**							Unit Price "A" Codes	200
2								Unit Price Other Codes	230
3		**january**	**february**	**arch**	**april**	**may**	**june**		
4	**Product Code**								
5									
6	**A1298**	123	30	135	144	148	152		
7	**B2344**	333	333	334	342	345	353		
8	**A1209**	554	561	562	566	573	577		
9	**C3433**	211	211	214	219	225	228		
10	**D2322**	345	352	358	362	370	373		
11	**J8976**	222	230	240	245	252	254		
12	**A2345**	444	447	457	463	466	471		
13	**B2322**	121	129	138	138	142	144		
14	**C7674**	343	350	356	364	364	372		
15									
16		january	uary	march	april	may	june		
17	A1298	£ 24,600.00	£ 25,970.13	£ 26,930.55	£ 28,780.24	£ 29,676.06	£ 30,305.70		
18	B2344	£ 76,590.00	£ 76,624.05	£ 76,720.53	£ 78,774.73	£ 79,289.03	£ 81,291.24		
19	A1209	£ 110,800.00	£ 112,126.67	£ 112,342.90	£ 113,281.29	£ 114,599.26	£ 115,451.46		
20	C3433	£ 48,530.00	£ 48,584.14	£ 49,262.30	£ 50,413.01	£ 51,662.07	£ 52,353.24		
21	D2322	£ 79,350.00	£ 80,883.32	£ 82,306.75	£ 83,337.46	£ 85,197.02	£ 85,799.12		
22	J8976	£ 51,060.00	£ 52,990.48	£ 55,142.98	£ 56,299.22	£ 57,976.39	£ 58,368.24		
23	A2345	£ 88,800.00	£ 89,406.89	£ 91,406.70	£ 92,688.54	£ 93,207.90	£ 94,204.57		
24	B2322	£ 27,830.00	£ 29,606.09	£ 31,633.41	£ 31,649.50	£ 32,607.76	£ 33,048.59		
25	C7674	£ 78,890.00	£ 80,490.40	£ 81,937.88	£ 83,656.11	£ 83,675.55	£ 85,503.45		

To create the array formula first highlight the range D17:I25.

Then type the formula:

=IF(LEFT(C6:C14)="a",D6:I14*K1,D6:I14*K2)

Remember to press CSE after typing the formula and make it into an array:

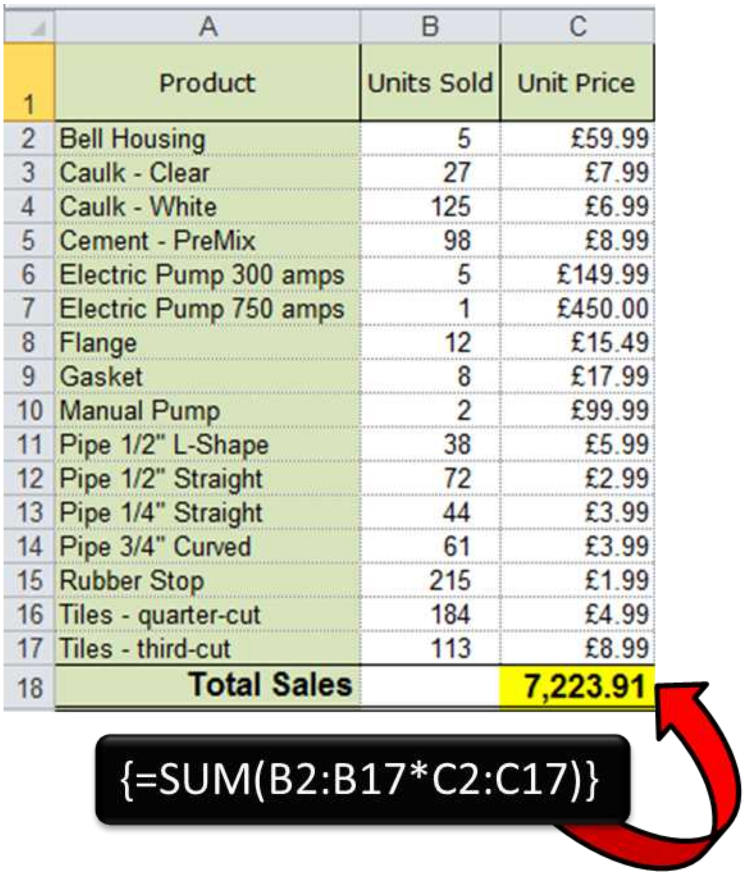
{=IF(LEFT(C6:C14)="a",D6:I14*K1,D6:I14*K2)}

The advantage of using an array here is that there is no need for any absolute referencing of cells because there is no copying involved. To achieve the same calculation without using arrays the formula would read:

=IF(LEFT($C6)="a",D6*$K$1,D6*$K$2)

A single-cell array formula

Now let's take a look at a single-cell array formula, also known as a *one dimensional array*. The following example which is similar to the previous one does not use column D. The objective is to calculate the sum of the total product sales without using the individual calculations that were in column D.



	A	B	C
1	Product	Units Sold	Unit Price
2	Bell Housing	5	£59.99
3	Caulk - Clear	27	£7.99
4	Caulk - White	125	£6.99
5	Cement - PreMix	98	£8.99
6	Electric Pump 300 amps	5	£149.99
7	Electric Pump 750 amps	1	£450.00
8	Flange	12	£15.49
9	Gasket	8	£17.99
10	Manual Pump	2	£99.99
11	Pipe 1/2" L-Shape	38	£5.99
12	Pipe 1/2" Straight	72	£2.99
13	Pipe 1/4" Straight	44	£3.99
14	Pipe 3/4" Curved	61	£3.99
15	Rubber Stop	215	£1.99
16	Tiles - quarter-cut	184	£4.99
17	Tiles - third-cut	113	£8.99
18	**Total Sales**		**7,223.91**

{=SUM(B2:B17*C2:C17)}

When you enter this formula, make sure that you use Ctrl+Shift+Enter (and don't type the curly brackets because Excel automatically adds them for you).

This formula works with two arrays, both of which are stored in cells. The first array is stored in B2:B7 and the second array is stored in C2:C7. The formula multiplies the corresponding values in these two arrays and creates a new array (which exists only in memory). The SUM function then operates on this new array and returns the sum of its values.

> Note: In this example, you can use the SUMPRODUCT function to obtain the same result without using an array formula: =SUMPRODUCT(B2:B17,C2:C17)

As you can see, array formulas allow many other types of calculations that are otherwise not possible.

Creating an array constant

The examples in the preceding section used arrays stored in worksheet ranges. The examples in this section demonstrate an important concept: An array need not be stored in a range of cells. This type of array, which is stored in memory, is referred to as an *array constant.*

To create an array constant, list its items (separated by commas) into a cell and surround the items with curly brackets as per the following example of a five-item horizontal array constant:

{2,5,0,0,7}

The following formula uses the SUM function, with the preceding array constant as its argument.

The formula returns the sum of the values in the array (which is 14):

=SUM({2,5,0,0,7})

> Note: This formula uses an array, but the formula itself isn't an array formula. Therefore, you don't use Ctrl+Shift+Enter to enter the formula. Although, entering the formula as an array formula will also work producing the same result.

> Note: When you specify an array directly (as shown above), you must type in the curly brackets around the array elements. When you enter an array formula, on the other hand, you do not supply the brackets.

Array constant elements

An array constant can contain numbers, text, logical values (TRUE or FALSE), and even error values, such as #N/A. Numbers can be in integer, decimal, or scientific format. You must enclose text in double quotation marks. You can use different types of values in the same array constant, as in this example:

{1,2,3,TRUE,FALSE,TRUE,"Bob","John","Sam"}

An array constant can't contain formulas, functions, or other arrays. Numeric values can't contain dollar signs, commas, parentheses, or percent signs. For example, the following is an invalid array constant:

{SQRT(32),$56.32,12.5%}

Understanding the Dimensions of an Array

As stated previously, an array can be one dimensional or two dimensional. A one-dimensional array's orientation can be horizontal (corresponding to a single row) or vertical (corresponding to a single column).

One-dimensional horizontal arrays

The elements in a one-dimensional horizontal array are separated by commas, and the array can be displayed in a row of cells. The following example is a one-dimensional horizontal array constant:

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

Displaying this array in a range requires five consecutive cells in a row. To enter this array into a range, select a range of cells that consists of one row and five columns. Then enter **={1,2,3,4,5}** and press Ctrl+Shift+Enter.

> Note: If you enter this array into a horizontal range that consists of more than five cells, the extra cells will contain #N/A (which denotes unavailable values). If you enter this array into a vertical range of cells, only the first item (1) will appear in each cell.

The following example is another horizontal array; it has seven elements and is made up of text strings:

```
{"Sun","Mon","Tue","Wed","Thu","Fri","Sat"}
```

To enter this array, select seven cells in a row and type the following (followed by Ctrl+Shift+Enter):

```
={"Sun","Mon","Tue","Wed","Thu","Fri","Sat"}
```

One-dimensional vertical arrays

The elements in a one-dimensional vertical array are separated by *semicolons*, and the array can be displayed in a column of cells. The following is a seven-element vertical array constant:

```
{10;20;30;40;50;60;70}
```

Displaying this array in a range requires seven cells in a column. To enter this array into a range, select a range of cells that consists of seven rows and one column. Then enter the following formula, followed by Ctrl+Shift+Enter:

```
={10;20;30;40;50;60;70}
```

The following is another example of a vertical array; this one has four elements:

```
{"Bell Housing";"Flange";"Gasket";"Electric Pump 750 amps"}
```

Two-dimensional arrays

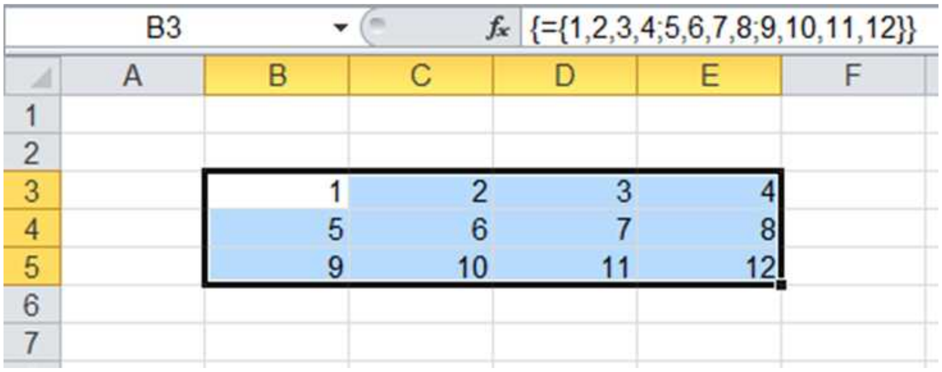
A two-dimensional array uses *commas* to separate its horizontal elements and *semicolons* to separate its vertical elements. The following example shows a 3 × 4 array constant:

{1,2,3,4;5,6,7,8;9,10,11,12}

Displaying this array in a range requires 12 cells. To enter this array into a range, select a range of cells that consists of three rows and four columns. Then type the following formula, followed by Ctrl+Shift+Enter:

={1,2,3,4;5,6,7,8;9,10,11,12}

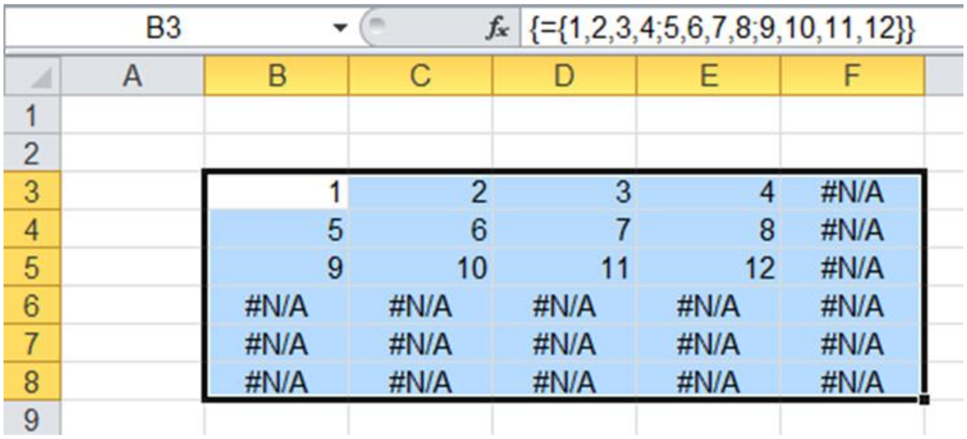
The following figure shows how this array appears when entered into a range (in this case, B3:E5).



B3 {={1,2,3,4;5,6,7,8;9,10,11,12}}

	A	B	C	D	E	F
1						
2						
3		1	2	3	4	
4		5	6	7	8	
5		9	10	11	12	
6						
7						

If you enter an array into a range that has more cells than array elements, Excel displays #N/A in the extra cells. The next figure shows a 3 × 4 array entered into a 6 × 5 cell range.



B3 {={1,2,3,4;5,6,7,8;9,10,11,12}}

	A	B	C	D	E	F
1						
2						
3		1	2	3	4	#N/A
4		5	6	7	8	#N/A
5		9	10	11	12	#N/A
6		#N/A	#N/A	#N/A	#N/A	#N/A
7		#N/A	#N/A	#N/A	#N/A	#N/A
8		#N/A	#N/A	#N/A	#N/A	#N/A
9						

Each row of a two-dimensional array must contain the same number of items. The array that follows, for example, isn't valid, because the third row contains only three items:

{1,2,3,4;5,6,7,8;9,10,11}

Excel doesn't allow you to enter a formula that contains an invalid array.

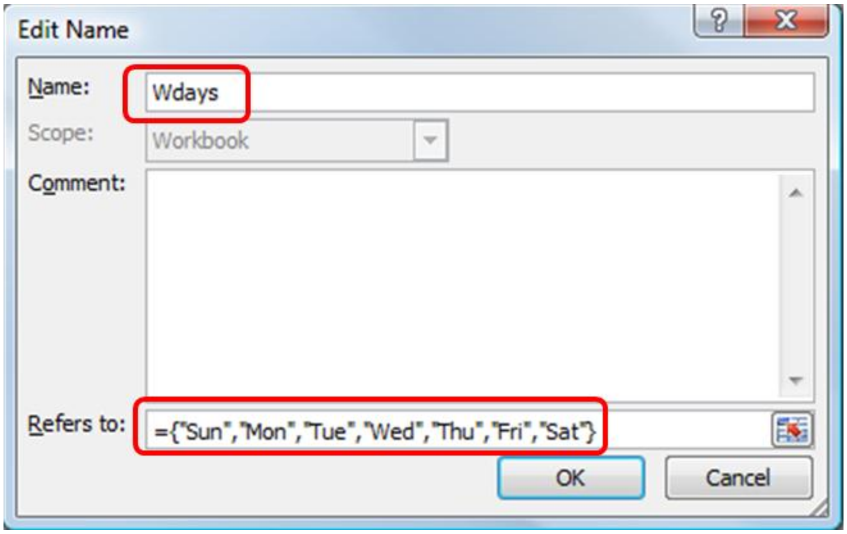
Naming Array Constants

You can create an array constant, give it a name, and then use this named array in a formula. Technically, a named array is a named formula.

The following figure shows a named array being created from the New Name dialog box. (Access this dialog box by choosing Formulas > Defined Names > Define Name.) The name of the array is *Wdays*, and it refers to the following array constant:

{"Sun","Mon","Tue","Wed","Thu","Fri","Sat"}

Note: In the New Name dialog box, the array is defined (in the Refers To field) using a leading equal sign (=). Without this equal sign, the array is interpreted as a text string rather than an array. Also, you must type the curly brackets when defining a named array constant; Excel does not enter them for you.



Name: Wdays

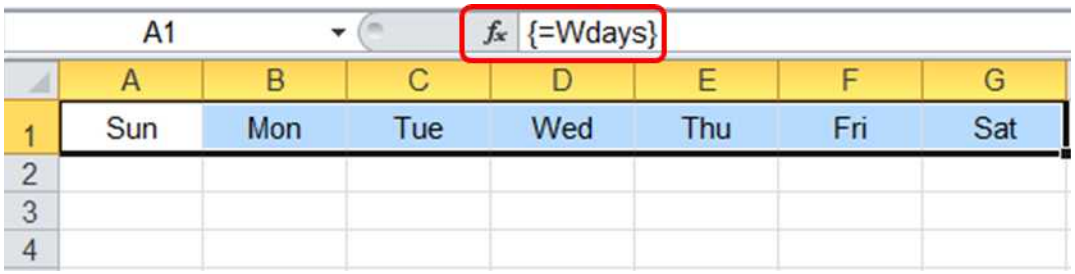
Scope: Workbook

Comment:

Refers to: ={"Sun","Mon","Tue","Wed","Thu","Fri","Sat"}

After creating this named array, you can use it in a formula. The next figure shows a worksheet that contains a single array formula entered into the range A1:G1. The formula is:

{=Wdays}



A1 *fx* {=Wdays}

	A	B	C	D	E	F	G
1	Sun	Mon	Tue	Wed	Thu	Fri	Sat
2							
3							
4							

Using TRANPOSE to insert a horizontal array into a vertical range of cells

Because commas separate the array elements, the array has a horizontal orientation. Use semicolons to create a vertical array. Or you can use the Excel TRANSPOSE function to insert a horizontal array into a vertical range of cells. The following array formula, which is entered into a seven-cell vertical range, uses the TRANSPOSE function:

```
{=TRANSPOSE(Wdays)}
```

Using INDEX to access individual elements from an array

You also can access individual elements from the array by using the Excel INDEX function. The following formula, for example, returns Wed, the fourth item in the Wdays array:

```
=INDEX(Wdays,4)
```

Working with Array Formulas

This section deals with the mechanics of selecting cells that contain arrays and entering and editing array formulas. These procedures differ a bit from working with ordinary ranges and formulas.

Entering an array formula

When you enter an array formula into a cell or range, you must follow a special procedure so that Excel knows that you want an array formula rather than a normal formula. You enter a normal formula into a cell by pressing Enter. You enter an array formula into one or more cells by pressing Ctrl+Shift+Enter.

Don't enter the curly brackets when you create an array formula; Excel inserts them for you. If the result of an array formula consists of more than one value, you must select all the cells in the results range before you enter the formula. If you fail to do so, only the first element of the result is returned.

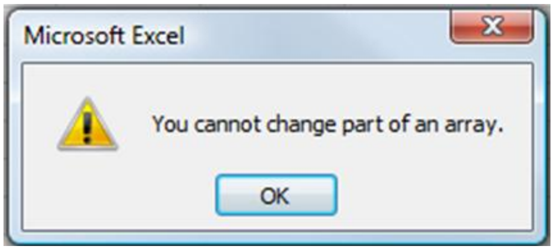
Selecting an array formula range

You can select the cells that contain a multicell array formula manually by using the normal cell selection procedures. Or you can use either of the following methods:

- Activate any cell in the array formula range. Display the Go To dialog box (choose Home > Editing > Find & Select > Go To, or just press F5). In the Go To dialog box, click the Special button and then choose the Current Array option. Click OK to close the dialog box.
- Activate any cell in the array formula range and press Ctrl+/ to select the entire array.

Editing an array formula

If an array formula occupies multiple cells, you must edit the entire range as though it were a single cell. The key point to remember is that you can't change just one element of a multicell array formula. If you attempt to do so, Excel displays the message shown in the following figure.



You cannot change part of an array.

The following rules apply to multicell array formulas. If you try to do any of these things, Excel lets you know about it:

- You can’t change the contents of any individual cell that makes up an array formula.
- You can’t move cells that make up part of an array formula (but you can move an entire array formula).
- You can’t delete cells that form part of an array formula (but you can delete an entire array).
- You can’t insert new cells into an array range. This rule includes inserting rows or columns that would add new cells to an array range.
- You can’t use multicell array formulas inside of a table that was created by choosing Insert > Tables > Table. Similarly, you can’t convert a range to a table if the range contains a multicell array formula.

To edit an array formula, select all the cells in the array range and activate the Formula bar as usual (click it or press F2). Excel removes the brackets from the formula while you edit it. Edit the formula and then press Ctrl+Shift+Enter to enter the changes. All the cells in the array now reflect your editing changes.

Note: If you accidentally press Ctrl+Enter (instead of Ctrl+Shift+Enter) after editing an array formula, the formula will be entered into each selected cell, but it will no longer be an array formula. And it will probably return an incorrect result. Just reselect the cells, press F2, and then press Ctrl+Shift+Enter.

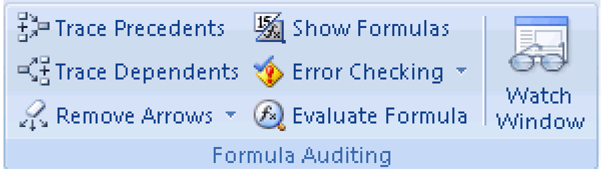
Unit 5: Auditing Formulas

In this unit, you will learn how to:

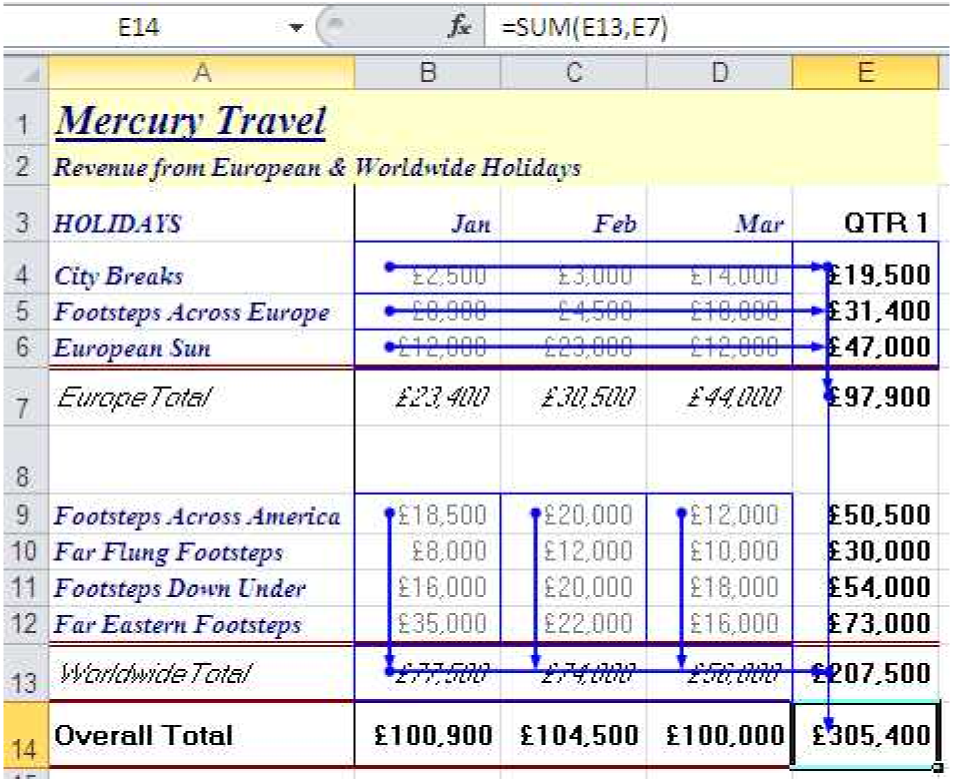
- Trace formula precedents, dependents and errors
- Correct errors in formulas
- Combine IF with VLOOKUP to suppress error messages
- Use the IS information function
- Use error checking functions; ISERR, ISERROR, IFERROR

Trace Formula Precedents, Dependents and Errors

On the Formulas, **Formula Auditing** group there are buttons that can help you trace errors and show the cells that are referenced in a formula.



The **Trace Precedents** button will display blue arrows to cells that supply data to the formula. Arrows may be red if a supplying cell has an error.



E14 f_x =SUM(E13,E7)

	A	B	C	D	E
1	***Mercury Travel***				
2	*Revenue from European & Worldwide Holidays*				
3	*HOLIDAYS*	*Jan*	*Feb*	*Mar*	QTR 1
4	*City Breaks*	£2,500	£3,000	£14,000	**£19,500**
5	*Footsteps Across Europe*	£8,900	£4,500	£18,000	**£31,400**
6	*European Sun*	£12,000	£23,000	£12,000	**£47,000**
7	*Europe Total*	*£23,400*	*£30,500*	*£44,000*	**£97,900**
8					
9	*Footsteps Across America*	£18,500	£20,000	£12,000	**£50,500**
10	*Far Flung Footsteps*	£8,000	£12,000	£10,000	**£30,000**
11	*Footsteps Down Under*	£16,000	£20,000	£18,000	**£54,000**
12	*Far Eastern Footsteps*	£35,000	£22,000	£16,000	**£73,000**
13	*Worldwide Total*	*£77,500*	*£74,000*	*£56,000*	**£207,500**
14	**Overall Total**	**£100,900**	**£104,500**	**£100,000**	**£305,400**

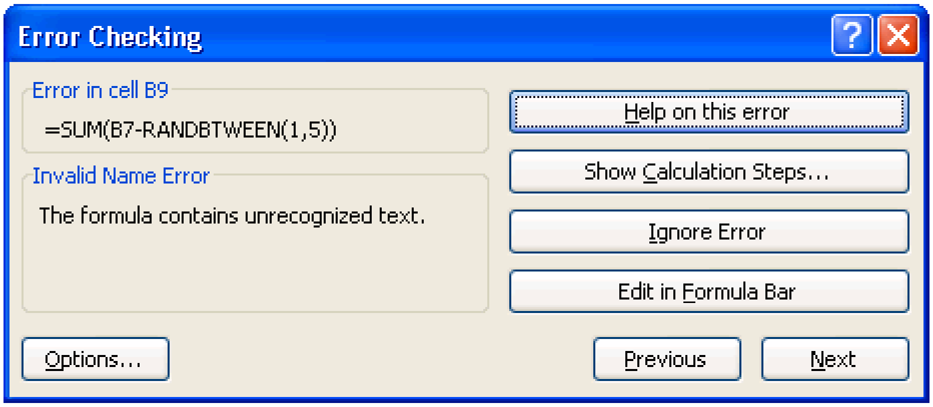
The **Trace Dependents** button will display arrows to the other cells that depend on a given cell's data.

The **Remove Arrows** button will remove both precedence and dependence arrows from your Excel screen.

The **Show Formulas** button will toggle between showing results and formulas in the worksheet. You can also use the **Ctrl + 'pipe' key** shortcut keys to do the same thing.

Correcting Errors in Formulas

The Error Checking option will check the entire worksheet for formula errors. If any are found, you will be alerted with an error checking dialogue (like the one below) that pertains to the specific error in question.



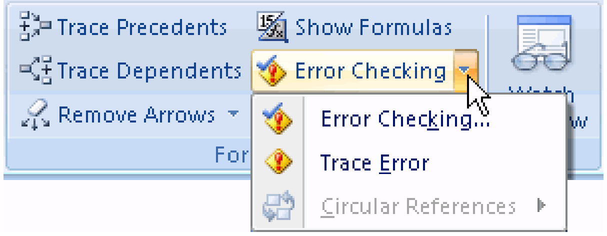
Error in cell B9

=SUM(B7-RANDBTWEEN(1,5))

Invalid Name Error

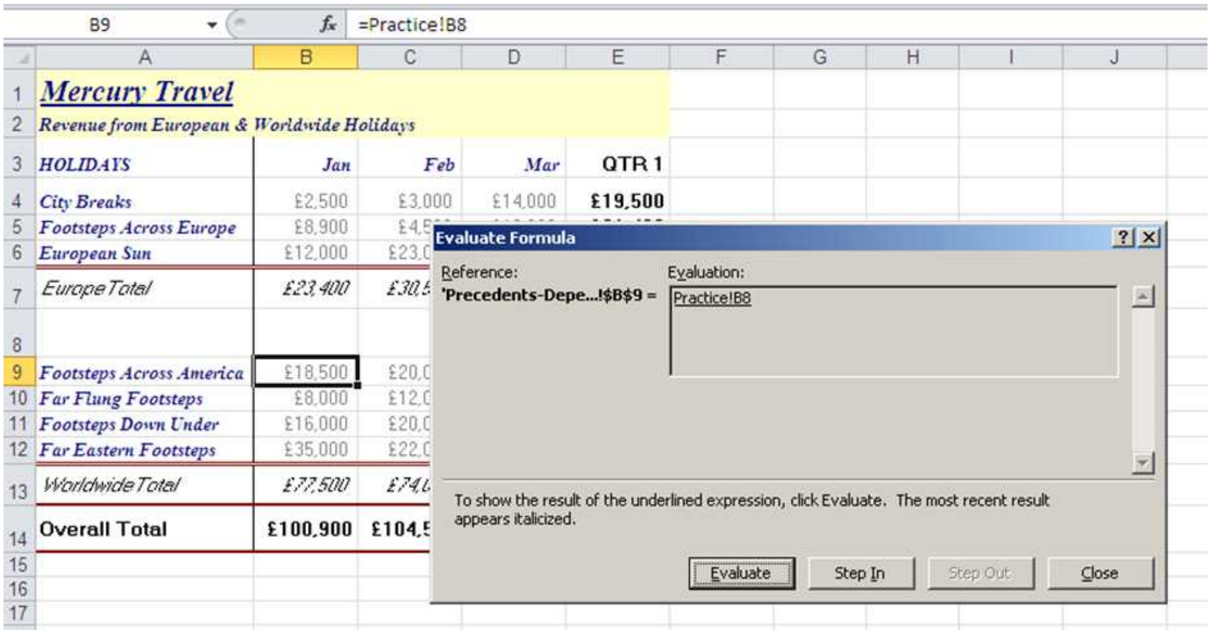
The formula contains unrecognized text.

You can also click the down arrow next to the Error Checking button to see additional options.



The first option, **Error Checking**, will perform the same action as if you clicked the Error Checking button directly. The **Trace Error** option will display arrows to the cells referenced in an incorrect formula. The **Circular References** menu will display cells that contain circular references, if there are any in the worksheet.

The **Evaluate Formula** dialog will help you analyse, interpret and correct formulas.



B9 =Practice!B8

	A	B	C	D	E
1	Mercury Travel				
2	Revenue from European & Worldwide Holidays				
3	HOLIDAYS	Jan	Feb	Mar	QTR 1
4	City Breaks	£2,500	£3,000	£14,000	£19,500
5	Footsteps Across Europe	£8,900	£4,5		
6	European Sun	£12,000	£23,0		
7	Europe Total	£23,400	£30,5		
8					
9	Footsteps Across America	£18,500	£20,0		
10	Far Flung Footsteps	£8,000	£12,0		
11	Footsteps Down Under	£16,000	£20,0		
12	Far Eastern Footsteps	£35,000	£22,0		
13	Worldwide Total	£77,500	£74,(		
14	Overall Total	£100,900	£104,5		

Evaluate Formula

Reference: 'Precedents-Depe...!$B$9 =

Evaluation: Practice!B8

To show the result of the underlined expression, click Evaluate. The most recent result appears italicized.

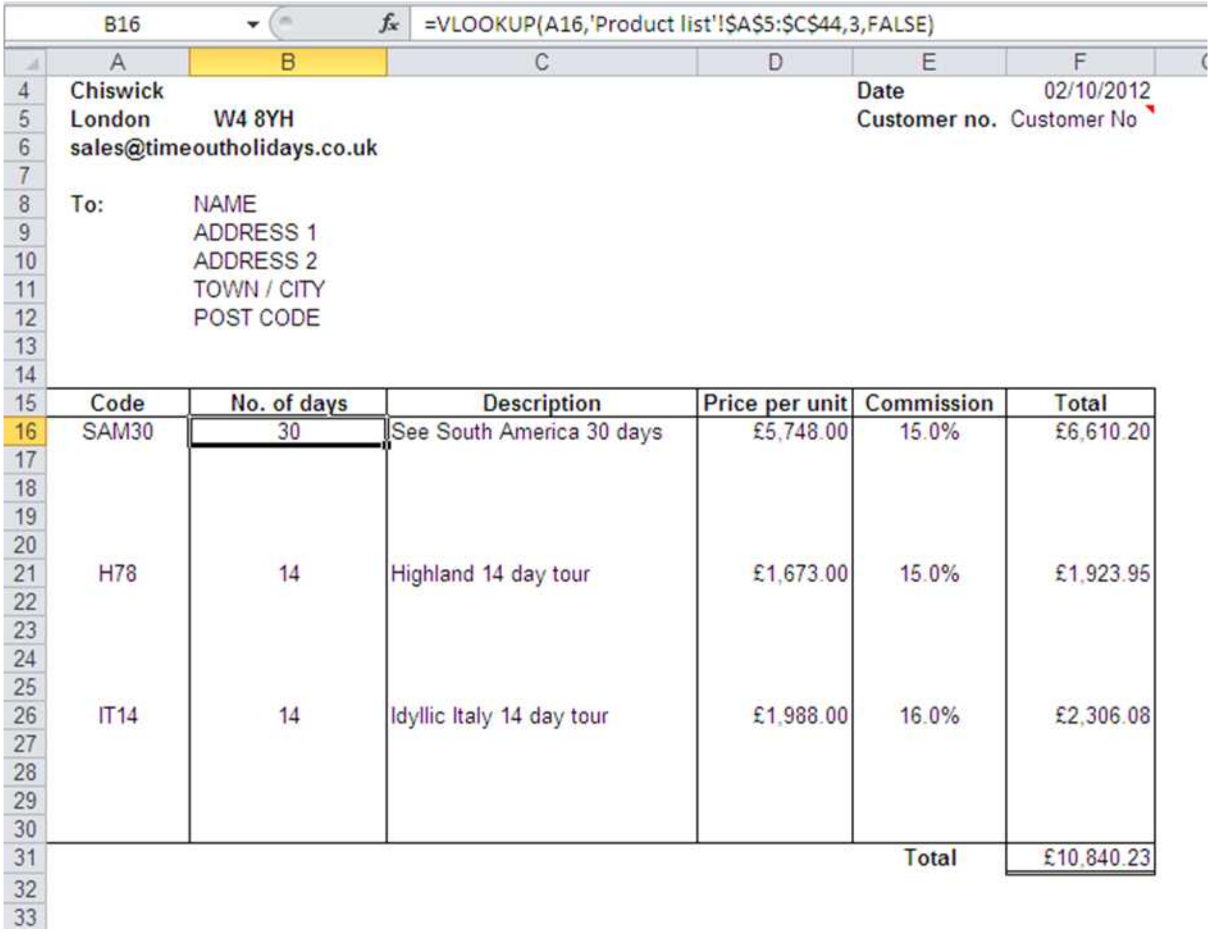
In this example cell B9 contains a formula linked to a cell in the worksheet called Practice.

Clicking **Step In** navigates to the Practice worksheet and **Step Out** returns back the worksheet containing the link.

Evaluate displays the value of the current cell rather than its formula.

Combine IF with VLOOKUP to suppress error messages

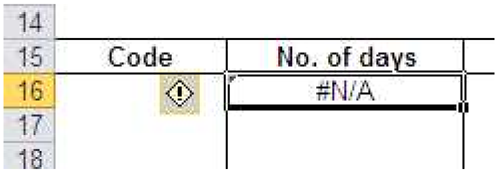
When working with VLOOKUP functions you may wish to suppress the #N/A error message that appears when an exact match is not found. For example, the invoice below uses a VLOOKUP functions to extract information from the Product List worksheet.



B16 =VLOOKUP(A16,'Product list'!$A$5:$C$44,3,FALSE)

	A	B	C	D	E	F
4	Chiswick				Date	02/10/2012
5	London	W4 8YH			Customer no.	Customer No
6	sales@timeoutholidays.co.uk					
7						
8	To:	NAME				
9		ADDRESS 1				
10		ADDRESS 2				
11		TOWN / CITY				
12		POST CODE				
13						
14						
15	Code	No. of days	Description	Price per unit	Commission	Total
16	SAM30	30	See South America 30 days	£5,748.00	15.0%	£6,610.20
17						
18						
19						
20						
21	H78	14	Highland 14 day tour	£1,673.00	15.0%	£1,923.95
22						
23						
24						
25						
26	IT14	14	Idyllic Italy 14 day tour	£1,988.00	16.0%	£2,306.08
27						
28						
29						
30						
31					Total	£10,840.23
32						
33						

If the Product Code is blank then the Vlookup returns #N/A



14		
15	Code	No. of days
16		#N/A
17		
18		

To suppress the error message an IF function can be used to test if the Code is blank as follows:

```
=IF(A16="","",VLOOKUP(A16,'Product list'!$A$5:$C$44,3,FALSE))
```

The IF statement specifically for a blank cell ("")

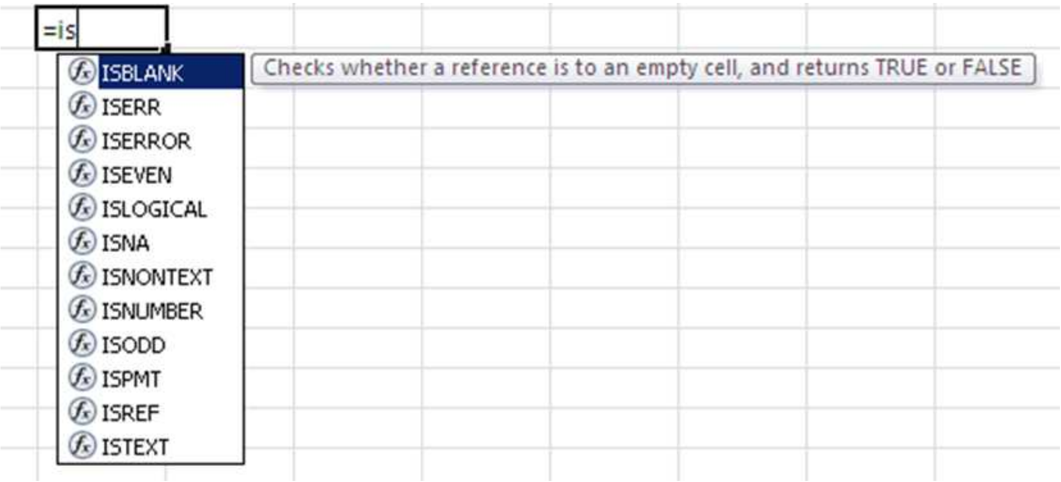
An alternative method is to use the ISNA function which displays a blank whenever a #N/A message appears.

```
=IF(ISNA(VLOOKUP(A16,'Product list'!$A$5:$C$44,3,FALSE)),"",VLOOKUP(A16,'Product list'!$A$5:$C$44,3,FALSE))
```

The IS Information Function

Type =IS into a cell and you will see there are many different functions beginning with IS. They are referred to collectively as **IS Functions**.

IS Functions return different types of information about a reference cell. They all return either TRUE or FALSE, for example ISBLANK returns TRUE if a cell is empty, ISNA returns TRUE if a cell contains a #N/A error message.



=is

Checks whether a reference is to an empty cell, and returns TRUE or FALSE

Here are the definitions of each IS Function.

ISBLANK	Value refers to an empty cell.
ISERR	Value refers to any error value except #N/A.
ISERROR	Value refers to any error value (#N/A, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, or #NULL!).
ISLOGICAL	Value refers to a logical value.
ISNA	Value refers to the #N/A (value not available) error value.
ISNONTEXT	Value refers to any item that is not text. (Note that this function returns TRUE if the value refers to a blank cell.)
ISNUMBER	Value refers to a number.
ISREF	Value refers to a reference.
ISTEXT	Value refers to text.

Error Checking Functions; ISERR, ISERROR and IFERROR

ISERR Function

This function tests a cell and returns TRUE if is an error in that cell.

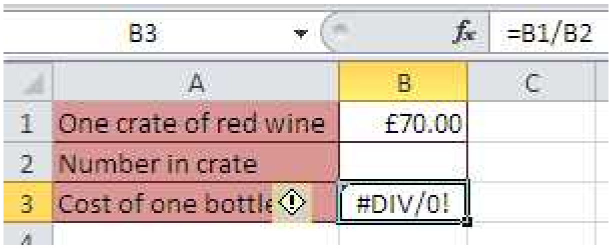
ISERR returns FALSE if the contents of the cell calculates without an error or if the error is the #NA message.

Cell to test	Result	
3	FALSE	=ISERR(A2)
#DIV/0!	TRUE	=ISERR(A3)
#NAME?	TRUE	=ISERR(A4)
#REF!	TRUE	=ISERR(A5)
#VALUE!	TRUE	=ISERR(A6)
#N/A	FALSE	=ISERR(A7)
#N/A	FALSE	=ISERR(A8)

Syntax

=ISERR(test cell)

The test cell can be a cell reference or a calculation.



B3 fx =B1/B2

	A	B	C
1	One crate of red wine	£70.00	
2	Number in crate		
3	Cost of one bottl	#DIV/0!	

In this example an error message appears because a cell B2 has been left blank. This created the divide by zero error (#DIV/0!) for the formula =B1/B2 in B3. While the error message disappears as soon as a value is typed into B2 it makes the spreadsheet look untidy.

The ISERR function can be used within an IF statement to suppress the error messages such as #DIV/0! or #VALUE!

=IF(ISERR(B1/B2),"",B1/B2)

Now any error message caused by 0, blank or a space being typed into B2 are suppressed and display as a blank cell. If there is no error then the calculation is performed (cost per bottle).

ISERROR Function

This function is similar to ISERR except it returns TRUE if there is a #NA error.

Syntax

=ISERROR(test cell)

IFERROR

This function not only traps for errors but also returns a value you specify if an error occurs. The IFERROR function avoids the need for an IF function and can also be created as an array formula.

Syntax

=IFERROR(test cell,value if error)

For example the

=IFERROR(B1/B2,"")

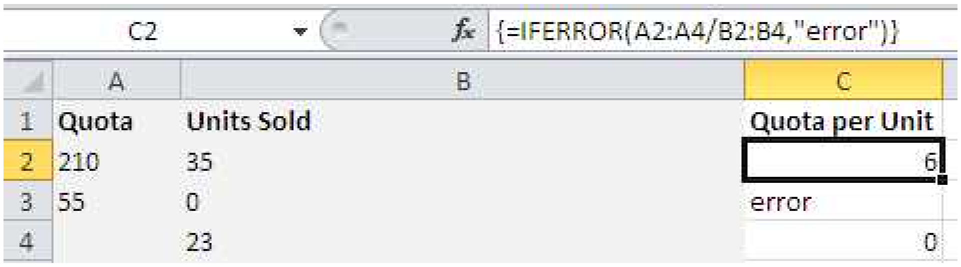
IFERROR as an Array Formula

The IFERROR function can be entered as an Array to perform a calculation on a range of cells rather than a single cell.

Syntax

{=IFERROR(test on each element in array, value if error in element)}

For example the Quota per Unit is entered as an array which calculates A2:A4/B2:B4 for each element and returns the word “error” if an error results for each calculation.



{=IFERROR(A2:A4/B2:B4,"error")}

	A	B	C
1	Quota	Units Sold	Quota per Unit
2	210	35	6
3	55	0	error
4		23	0

To create the formula

1. Select range C2:C4
2. Type =IFERROR(
3. Highlight A2:A4
4. Type /
5. Highlight B2:B4
6. Close)
7. Press Ctrl+Shift+Enter (CSE)

Pressing CSE at the end adds the array brackets to the formula:

={IFERROR(A2:A4/B2:B4,"error")}

Function syntax notes

IF

Syntax

=IF(Logical Test,Value if True,Value if False)

This function tests a condition.

If the condition is met it is considered to be TRUE.

If the condition is not met it is considered as FALSE.

Depending upon the result of the test, one of two actions will be carried out.

CHOOSE

Syntax =CHOOSE(UserValue, Item1, Item2, Item3 through to Item29)

This function picks from a list of options based upon an Index value given by the user.

INDEX

Synt*a*x 1

=INDEX(RangeToLookIn,Coordinate)

This is used when the RangeToLookIn is either a single column or row.

Syntax 2

=INDEX(RangeToLookIn,RowCoordinate,ColumnColumnCordinate)

This syntax is used when the range is made up of rows and columns.

This function picks a value from a range of data based on the row and column coordinates in the range.

Syntax 1 is for a single column or row of data.

Syntax 2 if for a block of data.

MATCH

Syntax

=MATCH(WhatToLookFor,WhereToLook,TypeOfMatch)

This function looks up an item in a list and finds it's **position**.

The Type of match is either 0, 1 or -1

0 looks for an exact match otherwise it dispays #NA

1 looks for an exact match or next lowest number

-1 looks for an exact match or next highest number.

INDEX

Syntax 1

=INDEX(RangeToLookIn,Coordinate)

This is used when the RangeToLookIn is either a single column or row.

Syntax 2

=INDEX(RangeToLookIn,RowCoordinate,ColumnColumnCordinate)

This syntax is used when the range is made up of rows and columns.

Syntax 3

=INDEX(NamedRangeToLookIn,RowCoordinate,ColumnColumnCordinate,AreaToPickFrom)

Using this syntax the range to look in can be made up of multiple areas.

The easiest way to refer to these areas is to select them and give them a single name.

MATCH

Syntax

=MATCH(WhatToLookFor,WhereToLook,TypeOfMatch)

This function looks for an item in a list and shows its position.

It can be used with text and numbers.

It can look for an exact match or an approximate match.

Match_type 1 or omitted

MATCH finds the largest value that is less than or equal to ***lookup_value***. The values in the ***lookup_array*** argument must be placed in ascending order, for example: ...-2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE.

Match_type 0

MATCH finds the first value that is exactly equal to ***lookup_value***. The values in the ***lookup_array*** argument can be in any order.

Match_type -1

MATCH finds the smallest value that is greater than or equal to ***lookup_value***. The values in the ***lookup_array*** argument must be placed in descending order, for example: TRUE, FALSE, Z-A, ...2, 1, 0, -1, -2, ..., and so on.

If ***match_type*** is 0 and ***lookup_value*** is a text string, you can use the wildcard characters — the question mark (**?**) and asterisk (*****) — in the ***lookup_value*** argument.

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.

RANK

Syntax

=RANK(NumberToRank,ListOfNumbers,RankOrder)

The RankOrder can be 0 zero or 1.

Using 0 will rank larger numbers at the top. (This is optional, leaving it out has the same effect).

Using 1 will rank small numbers at the top.

This function calculates the position of a value in a list relative to the other values in the list. A typical usage would be to rank the times of athletes in a race to find the winner.

The ranking can be done on an ascending (low to high) or descending (high to low) basis.

If there are duplicate values in the list, they will be assigned the same rank. Subsequent ranks would not follow on sequentially, but would take into account the fact that there were duplicates. If the numbers 30, 20, 20 and 10 were ranked, 30 is ranked as 1, both 20's are ranked as 2, and the 10 would be ranked as 4.

RANK.AVG

Syntax

=RANK.AVG(number,ref,[order])

Number is Required. The number whose rank you want to find.

Ref is Required. An array of, or a reference to, a list of numbers. Nonnumeric values in ref are ignored.

Order is Optional. A number specifying how to rank number.

If order is 0 (zero) or omitted, Microsoft Excel ranks number as if ref were a list sorted in descending order.

If order is any nonzero value, Microsoft Excel ranks number as if ref were a list sorted in ascending order.

This function returns the rank of a number in a list of numbers: its size relative to other values in the list; if more than one value has the same rank, the average rank is returned.

MEDIAN

Syntax

=MEDIAN(Range1,Range2,Range3…)

This function finds the median value of a group of values.

The median is not the average, it is the half way point where half the numbers in the group are larger than it and half the numbers are less than it. If there is no exact median number in the group, the two nearest the half way point are added and their average is used as the median.

MODE

Syntax

=MODE(Range1,Range2,Range3...)

This function displays the most frequently occurring number in a group of numbers.

For it to work correctly there must be at least two numbers which are the same. If all the values in the group are unique the function shows the error #N/A. When there is more than one set of duplicates, the number closest to the beginning of the group will be used. (Which is not really an accurate answer!)

ROUND

Syntax

=ROUND(NumberToRound,DecimalPlacesToUse)

This function rounds a number to a specified amount of decimal places.

If 0 is used the number is rounded to the nearest whole number.

If a negative amount of rounding is used the figures to the left of the decimal point are rounded.

ROUNDUP

Syntax

=ROUNDUP(NumberToRound,DecimalPlacesToUse)

This function rounds a number ***up*** to a specified amount of decimal places.

If 0 is used the number is rounded to the nearest whole number.

If a negative amount of rounding is used the figures to the left of the decimal point are rounded.

ROUNDDOWN

Syntax

=ROUNDDOWN(NumberToRound,DecimalPlacesToUse)

This function rounds a number ***down*** to a specified amount of decimal places.

The arguments are the same as for ROUNDUP

PMT

Syntax

=PMT(rate, nper, pv, [fv], [type])

Calculates the payment for a loan based on constant payments and a constant interest rate.

Rate is required. The interest rate for the loan.

Nper is required. The total number of payments for the loan.

Pv is required. The present value, or the total amount that a series of future payments is worth now; also known as the principal.

Fv is optional. 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.

Type is optional. The number 0 (zero) indicates payments are due in arrears, whereas, a 1 indicates payments are due in advance.

FV

Syntax

=FV(rate,nper,pmt,[pv],[type])

Returns the future value of an investment based on periodic, constant payments and a constant interest rate.

Rate is required. The interest rate per period.

Nper is required. The total number of payment periods in an annuity.

Pmt is required. The payment made each period; it cannot change over the life of the annuity. Typically, pmt contains principal and interest but no other fees or taxes. If pmt is omitted, you must include the pv argument.

Pv is optional. The present value, or the lump-sum amount that a series of future payments is worth right now. If pv is omitted, it is assumed to be 0 (zero), and you must include the pmt argument.

Type is optional. The number 0 or 1 and indicates when payments are due. If type is omitted, it is assumed to be 0 and payments are due at the end of the period. A type of 1 indicates payments are due at the beginning of the period.

TODAY

Syntax

=TODAY()

Use this to show the current date.

NETWORKDAYS

Syntax

=NETWORKDAYS(StartDate,EndDate,Holidays)

Holidays : This is a list of dates which will be excluded from the calculation, such as Xmas and Bank holidays.

This function will calculate the number of working days between two dates. It will exclude weekends and any holidays.

DATEDIF

Syntax

=DATEDIF(FirstDate,SecondDate,"Interval")

This function calculates the difference between two dates and can show the result in days, months or years.

WORKDAY

Syntax

=WORKDAY(StartDate,Days,Holidays)

Use this function to calculate a past or future date based on a starting date and a specified number of days. The function excludes weekends and holidays and can therefore be used to calculate delivery dates or invoice dates.

If the result is shown as a number this can be formatted to a normal date by using Home > Number > Long Date or Short Date.

LEFT

Syntax

=LEFT(text, [num_chars])

This function returns the first character or characters in a text string, based on the number of characters you specify.

RIGHT
Syntax

=RIGHT(text,[num_chars])

This function returns the last character or characters in a text string, based on the number of characters you specify.

SEARCH
Syntax

=SEARCH(find_text,within_text,[start_num])

The **SEARCH** function locates one text string within a second text string, and returns the number of the starting position of the first text string from the first character of the second text string. For example, to find the position of the letter "n" in the word "printer", you can use the following function:

=SEARCH("n","printer")

The SEARCH function is not case sensitive and allows wild card characters.

FIND
Syntax

=FIND(find_text, within_text, [start_num])

The **FIND** function locates one text string within a second text string, and returns the number of the starting position of the first text string from the first character of the second text string.

The FIND function is case sensitive and does not allow wild card characters.

LEN
Syntax

=LEN(text)

LEN returns the number of characters in a text string.

MID
Syntax

=MID(text, start_num, num_chars)

MID returns a specific number of characters from a text string, starting at the position you specify, based on the number of characters you specify.

ISERR
Syntax

=ISERR(CellToTest)

This function tests a cell and shows TRUE if there is an error value in the cell.

It will show FALSE if the contents of the cell calculate without an error, or if the error is the #NA message.

ISERROR

Syntax

=ISERROR(CellToTest)

This function tests a cell or calculation to determine whether an error has been generated.

It will show TRUE for any type of error and FALSE if no error is found.

IFERROR

Syntax

=IFERROR(value, value_if_error)

Returns a value you specify if a formula evaluates to an error; otherwise, returns the result of the formula. Use the IFERROR function to trap and handle errors in a formula .

The following error types are evaluated: #N/A, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, or #NULL!.

Excel Advanced Part 2 Functions

Excel Function Syntax (Required arguments shown in bold)	What it does
Logical	
=IF(logical_test,True,False)	Tests if a logical test is met and returns one true and one false result.
=AND(condition1,condition2,...)	Returns TRUE or FALSE if both or all conditions are met.
=OR(condition1,condition2,...)	Returns TRUE or FALSE if either conditions met.
=NOT(condition)	Returns TRUE if a condition is not met.
=IFERROR(value, value_if_error)	Displays a message if an error occurs.
Lookup & Reference	
=VLOOKUP(ItemToFind,RangeToLook,Column, SortedOrUnsorted)	Looks for an item in the leftmost column of a table range and returns a value from a specified column number.
=CHOOSE(Index_Value, item1, item2, item3)	Chooses a value or action from a list of items based on an index number.
=MATCH(lookup_value, lookup_array, match_type**)**	Returns the relative position of an item in range.
=INDEX(range, row_num, column_num**)**	Returns a value in a range based on a row and column position within the range.
Statistical	
=COUNTIFS(criteria_range1, criteria 1, criteria_range2, criteria2)	Counts the number of items that meet both or all the criteria in the criteria ranges.
=SUMIFS(Sum range, criteria_range1, criteria 1, criteria_range2, criteria2)	Sums a range of values that meet all of the criteria in the criteria ranges.
=RANK(NumberToRank,RangeOfNumbers, RankOrder**)**	Returns the rank position of a value in a list of numbers (highest rank for ties).
=RANK.AVG(number,ref,order**)**	Returns the rank position of a value in a list of numbers (average rank for ties).
=MEDIAN(Range1,Range2,Range3…)	Returns the middle rank value in a range.
=MODE(Range1,Range2,Range3...)	Returns the most common value in a range.
=ROUND(NumberToRound,DecimalPlaces)	Rounds a value to a specified number of decimal places.

Excel Function Syntax (Required arguments shown in bold)	What it does
=ROUNDUP(NumberToRound,DecimalPlaces)	Rounds up a value to specified number of decimal places.
=ROUNDDOWN(Cell,DecimalPlaces)	Rounds down a value to specified number of decimal places.
=CEILING(Cell,Decimal Places)	Rounds a number to a specified significance.
=MOD(Cell,Devisor)	Returns the remainder when a number is divided by a devisor.
=ROW()	Returns the row number of a reference.
=COLUMN()	Returns the column number of a reference.
Date and Time	
=TODAY()	Returns the current date
=NETWORKDAYS(StartDate,EndDate,Holidays)	Calculates the number of working days between two dates not including holidays.
=DATEDIF(FirstDate,SecondDate, Interval)	Subtracts the days between 2 dates in days, months or years.
=WORKDAY(StartDate,Days,Holidays)	Returns the date after a specified number of working days.
Date(year,month,day)	Returns the date a year, month and date value.
Day(date)	Returns the Day number of a a date.
Month(date)	Returns the Month number of a date.
Text	
=CONCATENATE(text1,text2,text3…)	Joins together text values.
=LEFT(text, num_chars)	Returns a number of characters to the right of a cell.
=RIGHT(text,num_chars)	Returns a number of characters to the right of a cell.
=SEARCH(find_text,within_text,start_num**)**	Searches for the character position of a text character within text.
=FIND(find_text, within_text, start_num**)**	Searches for the character position of a text character within text (case sensitive).
=LEN(text)	Returns the number of characters in a cell.
=MID(text, start_num, num_chars)	Returns text within cell starting from a given position and number.
Financial	

Excel Function Syntax (Required arguments shown in bold)	What it does
=PMT(rate, nper, pv, fv**,** type**)**	Calculates the payment on a loan based on constant payment and constant interest rates.
=FV(rate,nper,pmt,pv,type**)**	Calculates the future value of a regular number of payments.
=IRR(initial investment,cash flow1,cashflow2,guess)	Calculates the percentage rate of return on a series of payments of varying amount.
Information	
=ISNA(cell)	Checks whether a value is #NA and returns either TRUE or FALSE.
=ISBLANK(cell)	Returns TRUE if a cell is empty.
=ISTEXT(cell)	Returns TRUE if a cell is a text.
=ISNUMBER(cell)	Reyurns TRUE if a cell is a number.
=CELL("filename")	Returns the filename and path of the current file. Displays blank if a file is not yet saved. Press F9 to update after saving.
Array Functions **Press Ctrl+Shift+Enter after typing**	
{=TRANSPOSE(array)}	Converts a vertical range to horizontal or visa vice versa.
{=SUM(range1*range2)}	Returns the product of ranges or arrays.
=SUMPRODUCT(range1,range2)	Alternative way to return the product of ranges or arrays. No need to press CSE
{=FREQUENCY(data_range,bin_array)}	Calculates how often a value occurs in each element of the bin_array.