



101 EXCEL FUNCTIONS

PDF



PDF GUIDE

TABLE of Contents

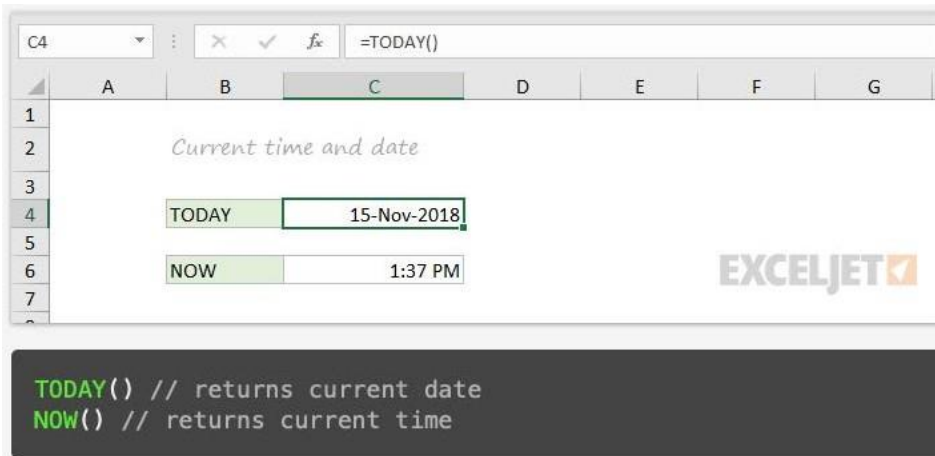
Date and Time Functions
Engineering
Information Functions
Logical Functions
Lookup and Reference Functions.....
STATISTICAL Functions.....
MATH Functions.....
TEXT Functions.....

Date and Time Functions

Excel provides many functions to work with [dates](#) and [times](#).

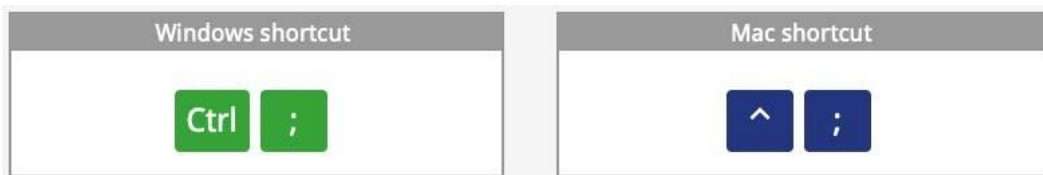
NOW and TODAY

You can get the current date with the [TODAY function](#) and the current date and time with the [NOW Function](#). Technically, the NOW function returns the current date and time, but you can format as time only, as seen below:



Note: these are [volatile functions](#) and will recalculate with every worksheet change. If you want a static value, use [date](#) and [time](#) shortcuts.

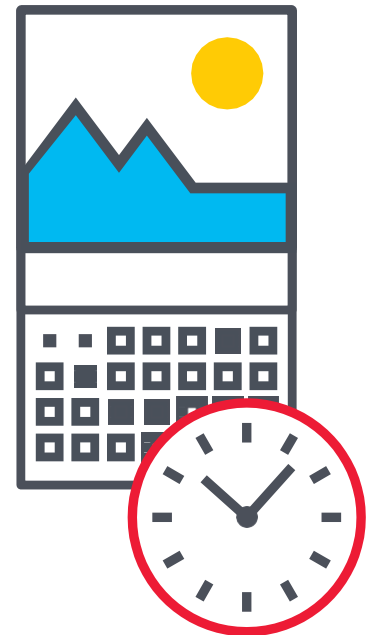
DATE SHORTCUT: This shortcut will insert the current date as a fixed value; it will not change.



TIME SHORTCUT: This shortcut will insert the current time as a fixed value; it will not change. [Note: In Mac 2016, Control Shift : stopped working to insert a time. Command ; now seems to work.]



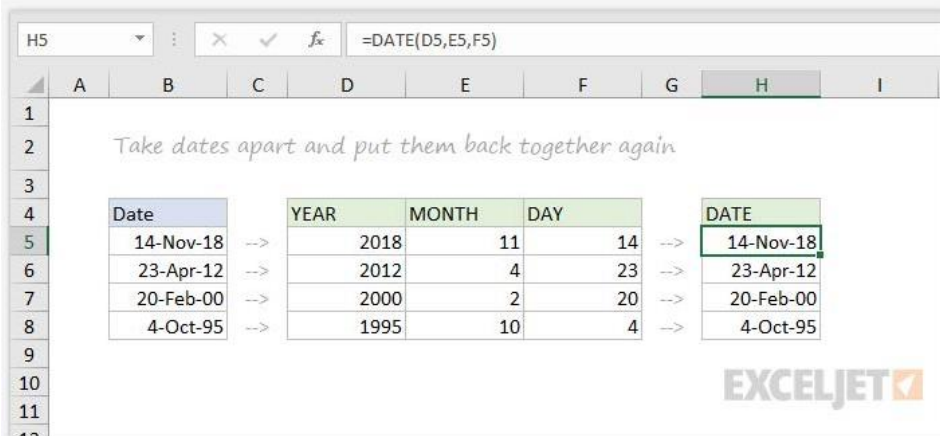
[More excel shortcuts.](#)



Easily add Date and Time to your Excel files using these functions

DAY, MONTH, YEAR, and DATE

You can use the [DAY](#), [MONTH](#), and [YEAR](#) functions to disassemble any date into its raw components, and the [DATE function](#) to put things back together again.



Take dates apart and put them back together again

Date		YEAR	MONTH	DAY		DATE
14-Nov-18	-->	2018	11	14	-->	14-Nov-18
23-Apr-12	-->	2012	4	23	-->	23-Apr-12
20-Feb-00	-->	2000	2	20	-->	20-Feb-00
4-Oct-95	-->	1995	10	4	-->	4-Oct-95

```
=DAY("14-Nov-2018") // returns 14
=MONTH("14-Nov-2018") // returns 11
=YEAR("14-Nov-2018") // returns 2018
=DATE(2018,11,14) // returns 14-Nov-2018
```

Did you know?

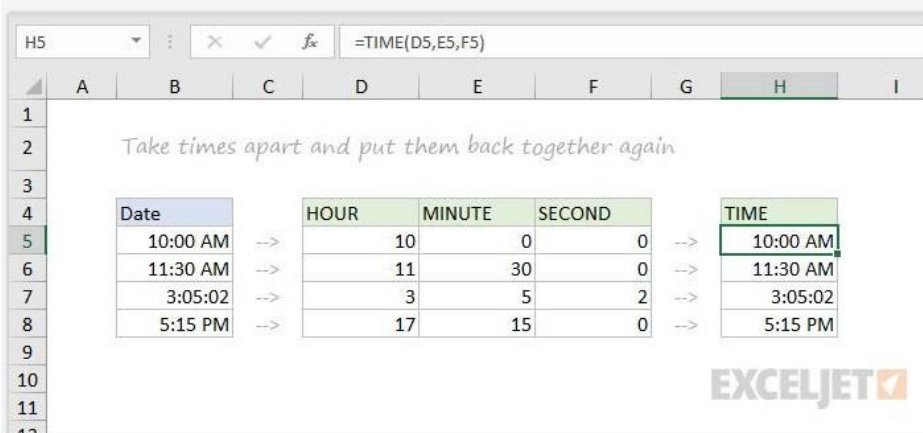
Excel dates are serial numbers that start in the year 1900.

Excel times are fractions of the number 1.

Both [dates](#) and [times](#) are numbers that can be used in math operations.

HOURL, MINUTE, SECOND, and TIME

Excel provides a set of parallel functions for times. You can use the [HOUR](#), [MINUTE](#), and [SECOND](#) functions to extract pieces of a time, and you can assemble a TIME from individual components with the [TIME function](#).



Take times apart and put them back together again

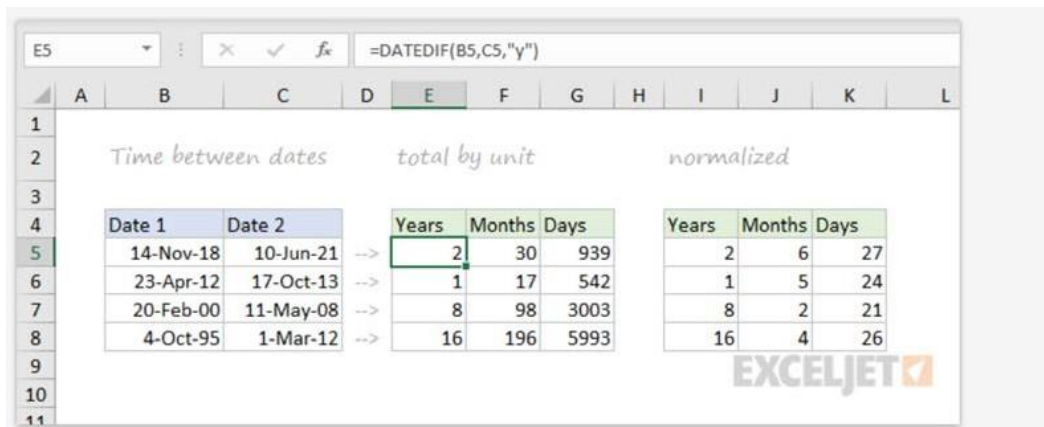
Date		HOUR	MINUTE	SECOND		TIME
10:00 AM	-->	10	0	0	-->	10:00 AM
11:30 AM	-->	11	30	0	-->	11:30 AM
3:05:02	-->	3	5	2	-->	3:05:02
5:15 PM	-->	17	15	0	-->	5:15 PM

```
=HOUR("10:30") // returns 10
=MINUTE("10:30") // returns 30
=SECOND("10:30") // returns 0
=TIME(10,30,0) // returns 10:30
```



DATEDIF and YEARFRAC

You can use the [DATEDIF function](#) to get time between dates in years, months, or days. DATEDIF can also be configured to get total time in “normalized” denominations, i.e. “2 years and 5 months and 27 days”.



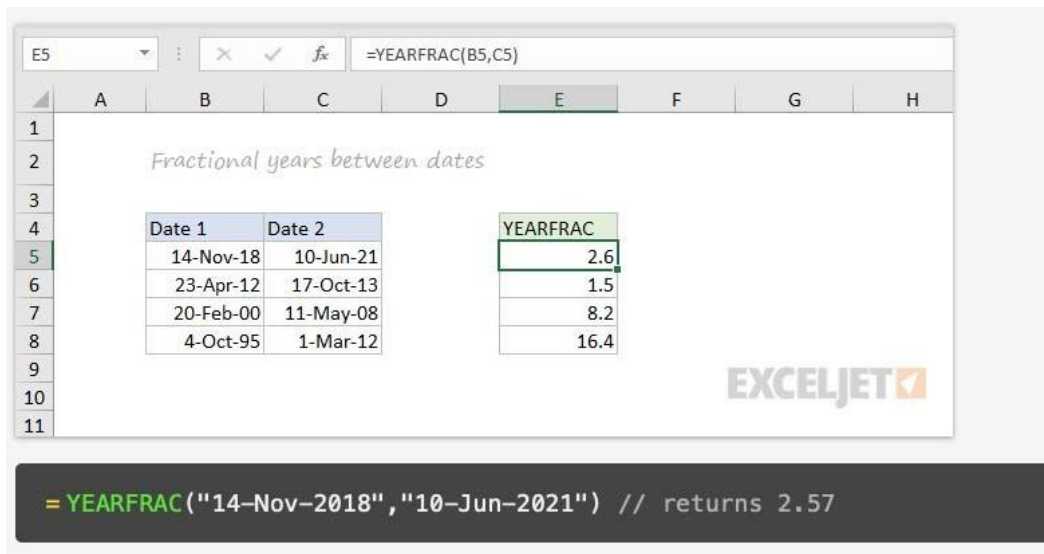
The screenshot shows an Excel spreadsheet with the formula bar displaying `=DATEDIF(B5,C5,"y")`. The spreadsheet is organized into three columns: "Time between dates", "total by unit", and "normalized". Each column contains a table of data.

Date 1	Date 2	Years	Months	Days
14-Nov-18	10-Jun-21	2	30	939
23-Apr-12	17-Oct-13	1	17	542
20-Feb-00	11-May-08	8	98	3003
4-Oct-95	1-Mar-12	16	196	5993

The "normalized" column shows the results in a "Years Months Days" format. The EXCELJET logo is visible in the bottom right corner.

The DATEDIF function is a good way to calculate age from a birthday. See [this example formula](#).

Use [YEARFRAC](#) to get fractional years:



The screenshot shows an Excel spreadsheet with the formula bar displaying `=YEARFRAC(B5,C5)`. The spreadsheet is organized into two columns: "Fractional years between dates" and "YEARFRAC". Each column contains a table of data.

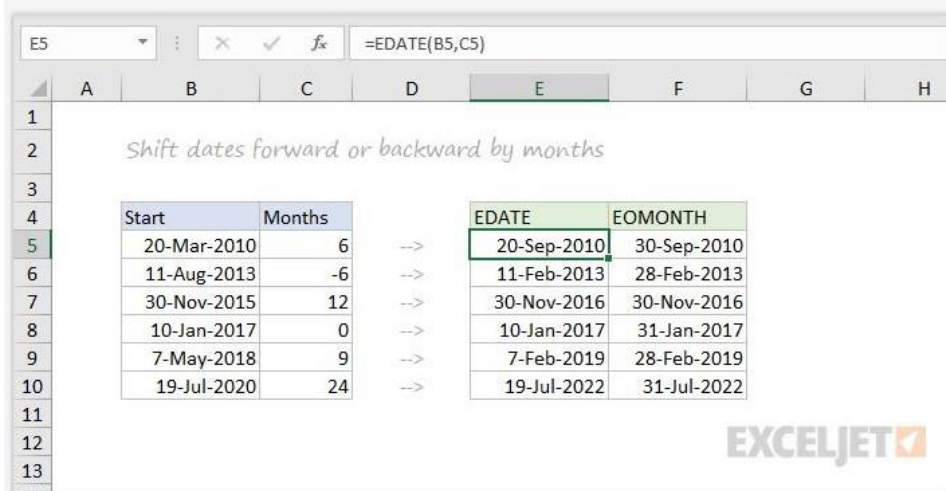
Date 1	Date 2	YEARFRAC
14-Nov-18	10-Jun-21	2.6
23-Apr-12	17-Oct-13	1.5
20-Feb-00	11-May-08	8.2
4-Oct-95	1-Mar-12	16.4

The EXCELJET logo is visible in the bottom right corner.

`= YEARFRAC("14-Nov-2018","10-Jun-2021") // returns 2.57`

EDATE and EOMONTH

A common task with dates is to shift a date forward (or backward) by a given number of months. You can use the [EDATE](#) and [EOMONTH](#) functions for this. EDATE moves by month and retains the day. EOMONTH works the same way, but always returns the last day of the month.



The image shows an Excel spreadsheet with the following data:

Start	Months		EDATE	EOMONTH
20-Mar-2010	6	→	20-Sep-2010	30-Sep-2010
11-Aug-2013	-6	→	11-Feb-2013	28-Feb-2013
30-Nov-2015	12	→	30-Nov-2016	30-Nov-2016
10-Jan-2017	0	→	10-Jan-2017	31-Jan-2017
7-May-2018	9	→	7-Feb-2019	28-Feb-2019
19-Jul-2020	24	→	19-Jul-2022	31-Jul-2022

The formula bar shows `=EDATE(B5,C5)`. The ExcelJET logo is visible in the bottom right corner of the spreadsheet area.

```
EDATE(date,6) // 6 months forward  
EOMONTH(date,6) // 6 months forward (end of month)
```



**Shift dates Forward
(or Backward)
using EDATE**

WORKDAY and NETWORKDAYS

To figure out a date n working days in the future, you can use the WORKDAY function. To calculate the number of [workdays](#) between two dates, you can use [NETWORKDAYS](#).

Start	Days	WORKDAY
Mon, 6-May-2019	5	Mon, 13-May-2019
Mon, 6-May-2019	10	Mon, 20-May-2019
Sat, 1-Jun-2019	30	Mon, 15-Jul-2019
Fri, 10-May-2019	15	Mon, 3-Jun-2019
Fri, 10-May-2019	-5	Fri, 3-May-2019

Holidays
27-May-2019
4-Jul-2019

holidays = G5:G6

```
WORKDAY(start,n,holidays) // date n workdays in future
```

Start	Finish	Workdays
Mon, 6-May-2019	Mon, 13-May-2019	6
Mon, 6-May-2019	Mon, 20-May-2019	11
Sat, 1-Jun-2019	Mon, 15-Jul-2019	30
Fri, 10-May-2019	Mon, 3-Jun-2019	16
Fri, 10-May-2019	Fri, 3-May-2019	-6

Holidays
27-May-2019
4-Jul-2019

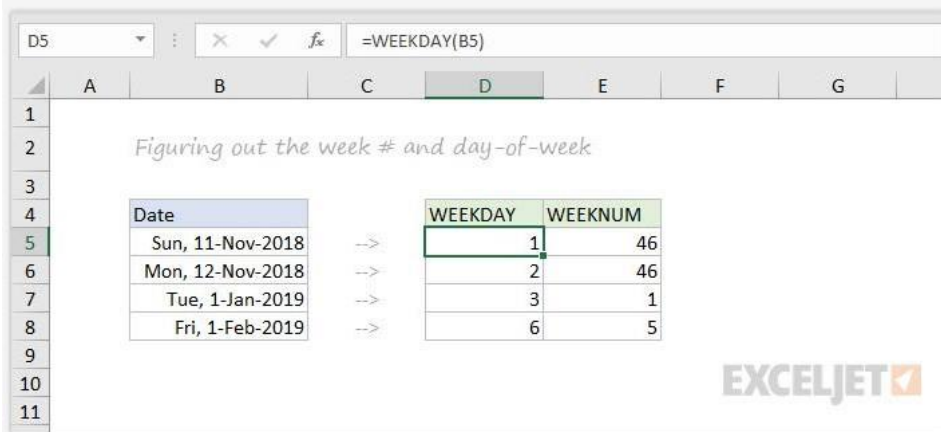
holidays = E5:E6

```
NETWORKDAYS(start,end,holidays) // number of workdays between dates
```

Note: Both functions automatically skip weekends (Saturday and Sunday) and will also skip holidays, if provided. If you need more flexibility on what days are considered weekends, see the [WORKDAY.INTL](#) function and [NETWORKDAYS.INTL](#) function.

WEEKDAY and WEEKNUM

To figure out the day of week from a date, Excel provides the [WEEKDAY function](#). WEEKDAY returns a number between 1-7 that indicates Sunday, Monday, Tuesday, etc. Use the [WEEKNUM function](#) to get the week number in a given year.



The image shows an Excel spreadsheet with the following data:

Date		WEEKDAY	WEEKNUM
Sun, 11-Nov-2018	-->	1	46
Mon, 12-Nov-2018	-->	2	46
Tue, 1-Jan-2019	-->	3	1
Fri, 1-Feb-2019	-->	6	5

The formula bar shows `=WEEKDAY(B5)`. The ExcelJET logo is visible in the bottom right corner of the spreadsheet area.

```
=WEEKDAY(date) // returns a number 1-7  
=WEEKNUM(date) // returns week number in year
```

See [this formula](#) to calculate sales per weekday.



MONDAY



TUESDAY



WEDNESDAY



THURSDAY



FRIDAY



SATURDAY

Engineering

CONVERT

Most Engineering functions are pretty technical... you'll find a lot of functions for complex numbers in this section. However, the [CONVERT](#) function is quite useful for everyday unit conversions. You can use CONVERT to change units for distance, weight, temperature, and much more.

Excel screenshot showing the CONVERT function formula bar: `=CONVERT(B5,C5,D5)`

Worksheet content:

Convert from one unit of measure to another

Input	From	To	Output
72	F	C	22.2
10	km	mi	6.2
175	lbm	kg	79.4
75	in	m	1.9
1	gal	l	3.8

EXCELJET

`= CONVERT(72,"F","C") // returns 22.2`

See [this formula](#) to calculate the BMI of an individual where the CONVERT function is used to convert between the metric and imperial unit systems.

Using the Versatile Convert Function

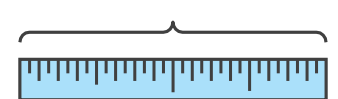


1 Gallon = 3.8 Litres

cm



in

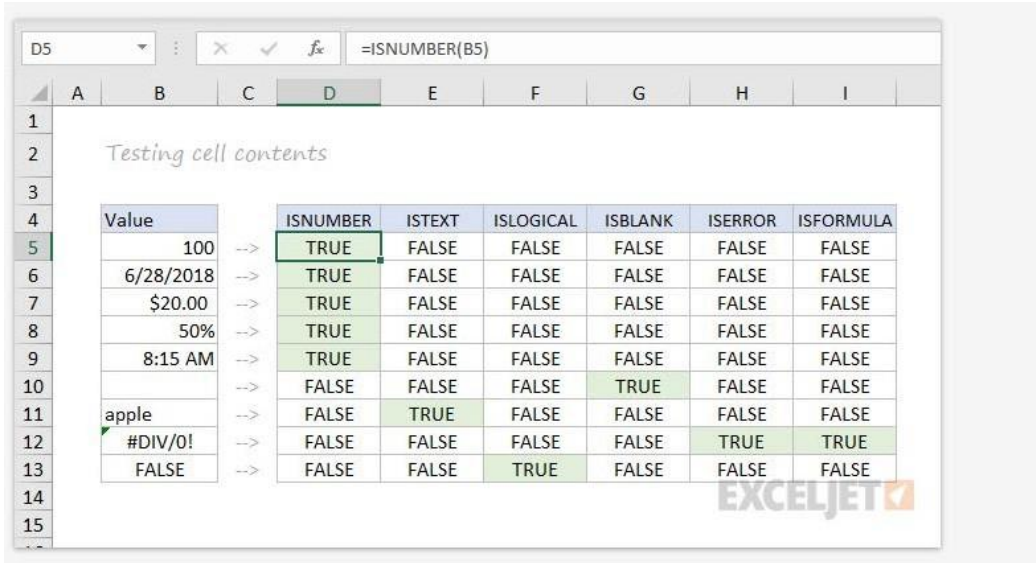


1 in = 2.54 cm

Information Functions

ISBLANK, ISERROR, ISNUMBER, and ISFORMULA

Excel provides many functions for checking the value in a cell, including [ISNUMBER](#), [ISTEXT](#), [ISLOGICAL](#), [ISBLANK](#), [ISERROR](#), and [ISFORMULA](#). These functions are sometimes called the “IS” functions, and they all return TRUE or FALSE based on a cell’s contents.



The screenshot shows an Excel spreadsheet with a table titled "Testing cell contents". The table has 7 columns: Value, ISNUMBER, ISTEXT, ISLOGICAL, ISBLANK, ISERROR, and ISFORMULA. The rows show the results of these functions for various values. The ISNUMBER column is highlighted in green for TRUE values and white for FALSE values. The ISERROR column is highlighted in green for TRUE values and white for FALSE values. The ISFORMULA column is highlighted in green for TRUE values and white for FALSE values.

Value	ISNUMBER	ISTEXT	ISLOGICAL	ISBLANK	ISERROR	ISFORMULA
100	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
6/28/2018	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
\$20.00	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
50%	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
8:15 AM	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
apple	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
#DIV/0!	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE

Excel also has [ISODD](#) and [ISEVEN](#) functions will test a number to see if it’s even or odd.

By the way, the green fill in the screenshot above is applied automatically with a [conditional formatting](#) formula.



**True or False
results using
Information
Functions**

Logical Functions

Excel's logical functions are a key building block of many advanced formulas. Logical functions return the boolean values TRUE or FALSE. If you need a primer on logical formulas, [this video goes through many examples](#).

AND, OR and NOT

The core of Excel's logical functions are the [AND function](#), the [OR function](#), and the [NOT function](#). In the screen below, each of these function is used to run a simple test on the values in column B:

The screenshot shows an Excel spreadsheet with the following data:

Value	AND	OR	NOT
2	FALSE	FALSE	FALSE
4	TRUE	FALSE	TRUE
3	FALSE	TRUE	TRUE
2	FALSE	FALSE	FALSE
7	TRUE	FALSE	TRUE
9	FALSE	TRUE	TRUE

Below the table, the formulas used are:

- =AND(B5 > 3, B5 < 9)
- =OR(B5 = 3, B5 = 9)
- =NOT(B5 = 2)

Is it Green?

	TRUE	FALSE
	✗	✓
	✓	✗
	✗	✓

IF and IFS functions

The [IF function](#) is one of the most used functions in Excel. In the screen below, IF checks test scores and assigns “pass” or “fail”:

The screenshot shows an Excel spreadsheet with the following data:

Name	Score	Result
Anderson	92	Pass
Bautista	85	Pass
Block	65	Fail
Burrows	79	Pass
Chandler	69	Fail
Colby	95	Pass

Passing score: 70

The logical functions above can be combined with the IF function to create more complex logical tests. Alternatively, multiple IF functions [can be nested together](#) to return more than two values as a result.

New in Excel 2019 and Office 365, the [IFS function](#) can run multiple logical tests without [nesting IFs](#).

The screenshot shows an Excel worksheet titled "Assigning grades with the IFS function". It contains two tables. The first table lists names and scores, and the second table shows the corresponding grades assigned by the IFS function.

Name	Score	Grade
Hannah	81.8	B
Edward	82.8	B
Miranda	91.3	A
William	76	C
Joanna	71.2	C
Collin	80.6	B
Mallory	85	B
Oscar	79.2	C

Score	Grade
0	F
60	D
70	C
80	B
90	A

The formula bar shows the IFS function: `=IFS(C5<60,"F",C5<70,"D",C5<80,"C",C5<90,"B",C5>=90,"A")`.

EXCELJET

`= IFS(C5 < 60, "F", C5 < 70, "D", C5 < 80, "C", C5 < 90, "B", C5 >= 90, "A")`

IFERROR and IFNA

The [IFERROR function](#) and [IFNA function](#) can be used as a simple way to trap and handle errors. In the screen below, [VLOOKUP](#) is used to retrieve cost from a menu item. Column F contains just a [VLOOKUP function](#), with no error handling. Column G shows how to use IFNA with VLOOKUP to display a custom message when an unrecognized item is entered.

The screenshot shows an Excel worksheet titled "Trapping VLOOKUP errors with IFNA". It contains two tables. The first table lists items and costs, and the second table shows the results of VLOOKUP and IFNA functions.

Item	Cost
Pizza	\$3.25
Hot Dog	\$1.75
Chicken	\$3.50
Sushi	\$5.00
Hamburger	\$3.25

Item	Cost	IFNA
Pizza	\$3.25	\$3.25
Sushi	\$5.00	\$5.00
Ice cream	#N/A	Not found

The formula bar shows the IFNA function: `=IFNA(VLOOKUP(E5,menu,2,0),"Not found")`.

EXCELJET

`= VLOOKUP(E5,menu,2,0) // no error trapping`
`= IFNA(VLOOKUP(E5,menu,2,0),"Not found") // catch errors`

Whereas IFNA only catches an #N/A error, the [IFERROR function](#) will catch any formula error.

Lookup and Reference Functions

VLOOKUP and HLOOKUP

Excel offers a number of functions to lookup and retrieve data. Most famous of all is [VLOOKUP](#):

D5 =VLOOKUP(C5,\$F\$5:\$G\$7,2,TRUE)

Basic lookup with VLOOKUP

Name	Sales	Commission
Ferris	\$71,900	3%
Bueller	\$93,500	4%
Chung	\$151,200	5%
Tanaka	\$124,600	5%
Irwin	\$82,500	4%
McNulty	\$60,400	3%

Sales	Commission
\$50,000	3%
\$75,000	4%
\$100,000	5%

EXCELJET

= VLOOKUP (C5 , \$F\$5 : \$G\$7 , 2 , TRUE)

More: [23 things to know about VLOOKUP](#).

[HLOOKUP](#) works like [VLOOKUP](#), but expects data arranged horizontally:

D5 =HLOOKUP(C5,\$G\$4:\$I\$5,2,TRUE)

Horizontal lookup with HLOOKUP

Name	Sales	Commission
Ferris	\$71,900	3%
Bueller	\$93,500	4%
Chung	\$151,200	5%
Tanaka	\$124,600	5%
Irwin	\$82,500	4%
McNulty	\$60,400	3%

Sales	\$50,000	\$75,000	\$100,000
Commission	3%	4%	5%

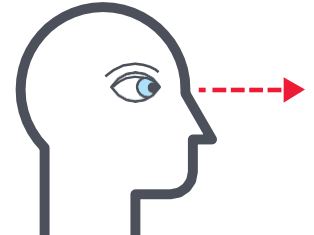
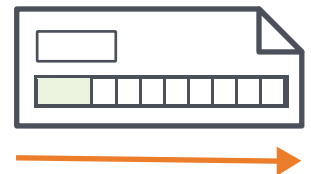
EXCELJET

= HLOOKUP (C5 , \$G\$4 : \$I\$5 , 2 , TRUE)

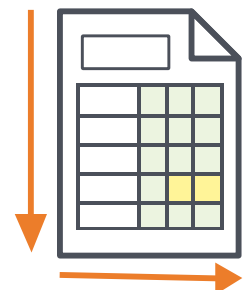
VLOOKUP is for vertical data



HLOOKUP is for horizontal data

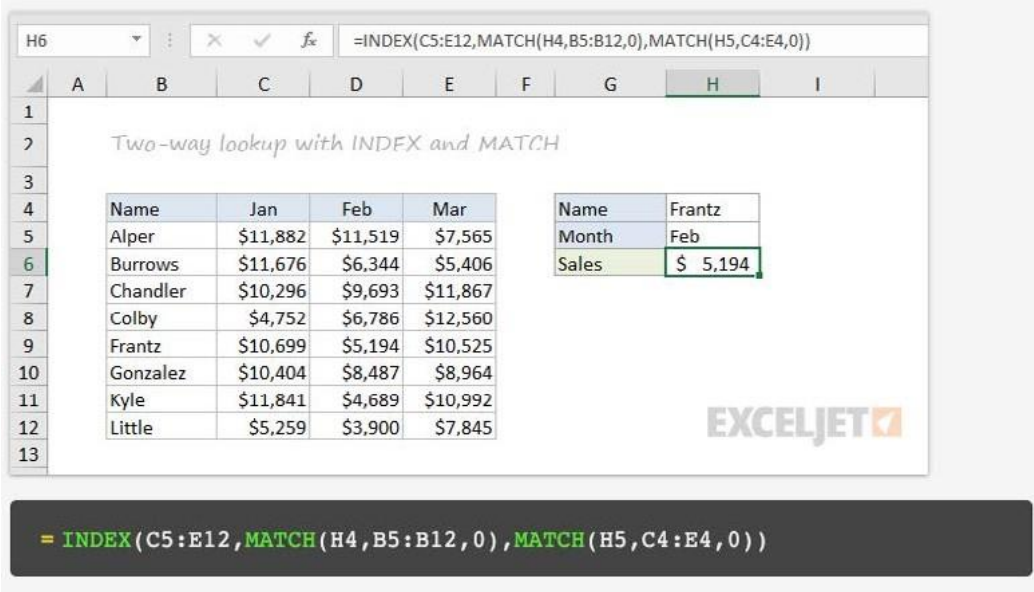


VLOOKUP only looks to the Right



INDEX and MATCH

For more complicated lookups, [INDEX](#) and [MATCH](#) offers more flexibility and power:



Two-way lookup with INDEX and MATCH

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

Name	Frantz
Month	Feb
Sales	\$ 5,194

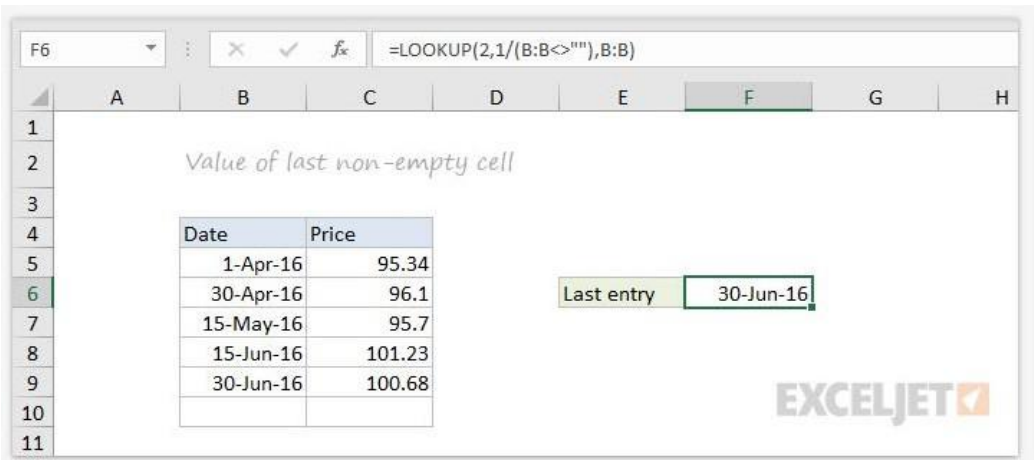
EXCELJET

= INDEX (C5:E12 , MATCH (H4 , B5:B12 , 0) , MATCH (H5 , C4:E4 , 0))

Both the [INDEX function](#) and the [MATCH function](#) are powerhouse functions that turn up in all kinds of formulas.

LOOKUP

The [LOOKUP function](#) has default behaviors that make it useful when solving certain problems. LOOKUP assumes values are sorted in ascending order and always performs an approximate match. When LOOKUP can't find a match, it will match the next smallest value. In the example below we are using LOOKUP to find the last entry in a column:



Value of last non-empty cell

Date	Price
1-Apr-16	95.34
30-Apr-16	96.1
15-May-16	95.7
15-Jun-16	101.23
30-Jun-16	100.68

Last entry	30-Jun-16
------------	-----------

EXCELJET

=LOOKUP(2,1/(B:B<>""),B:B)

[This page](#) explains this LOOKUP example in more depth.



The MATCH function is designed to find the position of an item in a range.

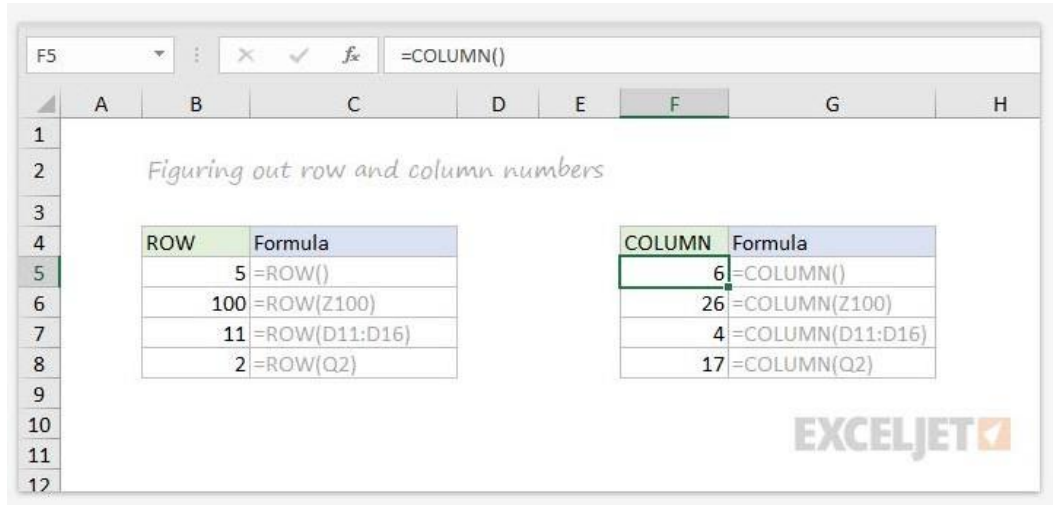


ARTICLE

[How to use INDEX and MATCH](#)

ROW and COLUMN

You can use the [ROW function](#) and [COLUMN function](#) to find row and column numbers on a worksheet. Notice both ROW and COLUMN return values for the current cell if no reference is supplied:



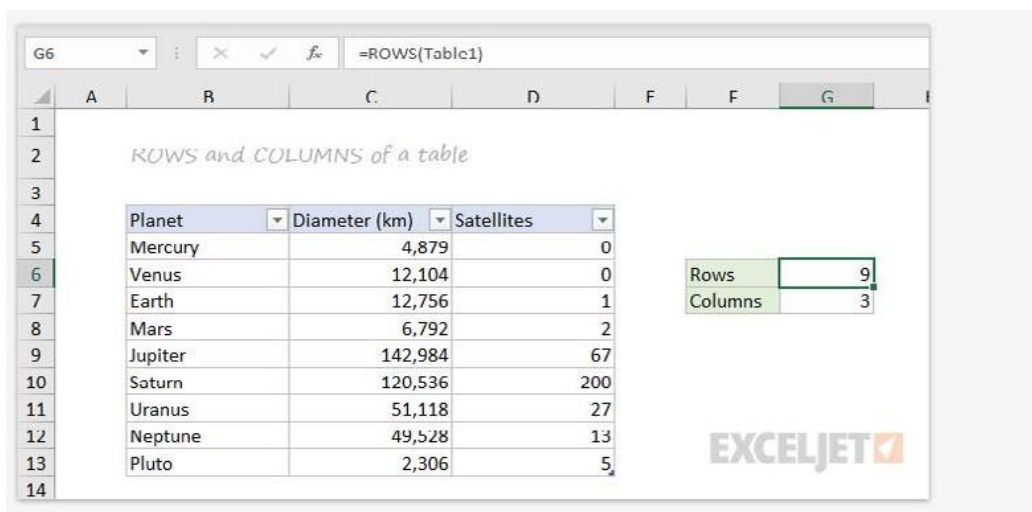
ROW	Formula
5	=ROW()
100	=ROW(Z100)
11	=ROW(D11:D16)
2	=ROW(Q2)

COLUMN	Formula
6	=COLUMN()
26	=COLUMN(Z100)
4	=COLUMN(D11:D16)
17	=COLUMN(Q2)

The row function also shows up often in advanced formulas that process data with [relative row numbers](#).

ROWS and COLUMNS

The [ROWS function](#) and [COLUMNS function](#) provide a count of rows in a reference. In the screen below, we are counting rows and columns in an [Excel Table](#) named “Table1”.



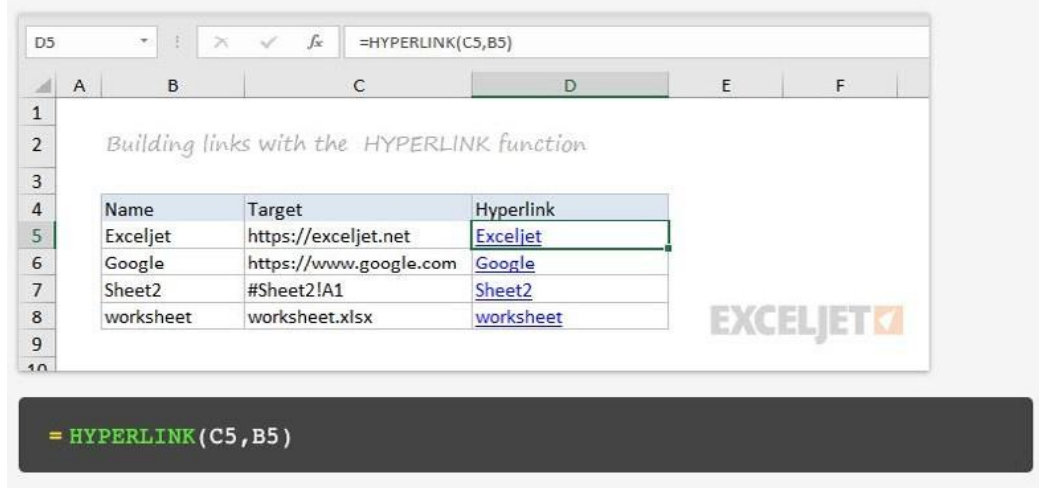
Planet	Diameter (km)	Satellites
Mercury	4,879	0
Venus	12,104	0
Earth	12,756	1
Mars	6,792	2
Jupiter	142,984	67
Saturn	120,536	200
Uranus	51,118	27
Neptune	49,528	13
Pluto	2,306	5

Rows	Columns
9	3

Note ROWS returns a count of data rows in a table, excluding the header row. By the way, here are [23 things to know about Excel Tables](#).

HYPERLINK

You can use the [HYPERLINK function](#) to construct a link with a formula. Note HYPERLINK lets you build both external links and internal links:



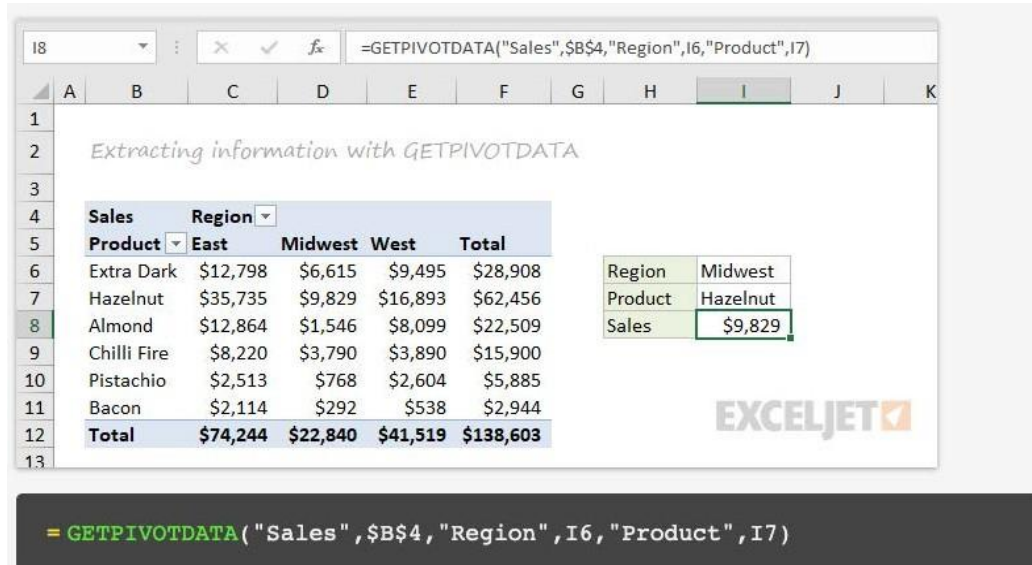
Name	Target	Hyperlink
Exceljet	https://exceljet.net	Exceljet
Google	https://www.google.com	Google
Sheet2	#Sheet2!A1	Sheet2
worksheet	worksheet.xlsx	worksheet

EXCELJET

= HYPERLINK (C5 ,B5)

GETPIVOTDATA

The [GETPIVOTDATA function](#) is useful for retrieving information from existing pivot tables.



Sales	Region	Product	East	Midwest	West	Total
Extra Dark	Midwest	Hazelnut	\$12,798	\$6,615	\$9,495	\$28,908
Hazelnut	Midwest	Hazelnut	\$35,735	\$9,829	\$16,893	\$62,456
Almond	Midwest	Hazelnut	\$12,864	\$1,546	\$8,099	\$22,509
Chilli Fire	Midwest	Hazelnut	\$8,220	\$3,790	\$3,890	\$15,900
Pistachio	Midwest	Hazelnut	\$2,513	\$768	\$2,604	\$5,885
Bacon	Midwest	Hazelnut	\$2,114	\$292	\$538	\$2,944
Total	Midwest	Hazelnut	\$74,244	\$22,840	\$41,519	\$138,603

EXCELJET

= GETPIVOTDATA ("Sales" , \$B\$4 , "Region" , I6 , "Product" , I7)

Build External and Internal Hyperlinks



CHOOSE

The [CHOOSE function](#) is handy any time you need to make a choice based on a number:

Simple lookups with the CHOOSE function

Input	Output
1	red
2	blue
3	green

`=CHOOSE(2,"red","blue","green") // returns "blue"`

TRANSPOSE

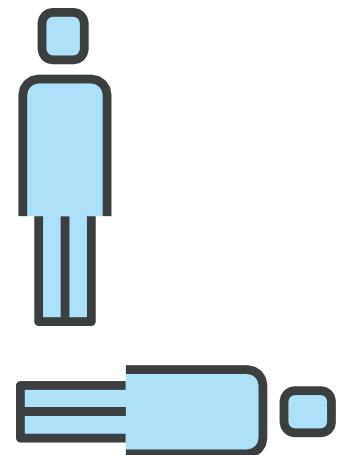
The [TRANSPOSE function](#) gives you an easy way to transpose vertical data to horizontal, and vice versa.

Vertical to horizontal with the TRANSPOSE function

Item	Cost
Pizza	\$3.25
Hot Dog	\$1.75
Chicken	\$3.50
Sushi	\$5.00
Falafel	\$3.25

`{=TRANSPOSE(B4:C9)}`

Note: TRANSPOSE is a formula and is therefore dynamic. If you just need to do a one-time transpose operation, use [Paste Special](#) instead.



**Transpose Vertical
Data to Horizontal**

OFFSET

The [OFFSET function](#) is useful for all kinds of dynamic ranges. From a starting location, it lets you specify row and column offsets, and also the final row and column size. The result is a range that can respond dynamically to changing conditions and inputs. You can feed this range to other functions, as in the screen below, where OFFSET builds a range that is fed to the SUM function:

The screenshot shows an Excel spreadsheet with the following data:

Region	Q1	Q2	Q3	Q4
East	8	4	3	8
West	8	6	10	7
South	9	7	4	4
North	3	10	1	7

Next to the table is a small table with user input:

Quarter	3
Sum	18

The formula bar shows: `=SUM(OFFSET(B4,1,I4,4,1))`

Below the spreadsheet, the formula is explained: `= SUM(OFFSET(B4,1,I4,4,1)) // sum of Q3`

INDIRECT

The [INDIRECT function](#) allows you to build references as text. This concept is a bit tricky to understand at first, but it can be useful in many situations. Below, we are using INDIRECT to get values from cell A1 in 5 different worksheets. Each reference is dynamic. If a sheet name changes, the reference will update.

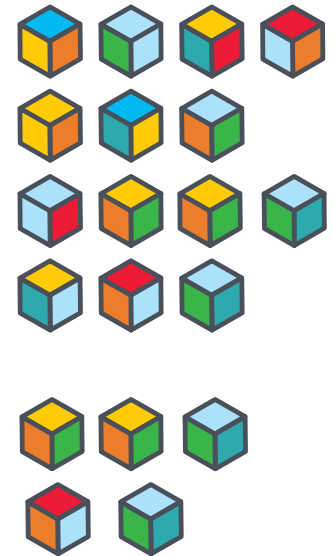
The screenshot shows an Excel spreadsheet with the following data:

Sheet	Value
Sheet1	100
Sheet2	200
Sheet3	300
Sheet4	400
Sheet5	500

The formula bar shows: `=INDIRECT(B5&"!A1")`

Below the spreadsheet, the formula is explained: `= INDIRECT(B5 & "!A1") // =Sheet1!A1`

The INDIRECT function is also used to “lock” references so they won’t change, when rows or columns are added or deleted. For more details, see linked examples at the bottom of the [INDIRECT function page](#).



The main purpose of OFFSET is to allow formulas to dynamically adjust to available data or to user input.

Caution: Both **OFFSET** and **INDIRECT** are volatile functions and can slow down large or complicated spreadsheets.

STATISTICAL Functions

COUNT and COUNTA

You can count numbers with the [COUNT function](#) and non-empty cells with [COUNTA](#). You can count blank cells with [COUNTBLANK](#), but in the screen below we are counting blank cells with [COUNTIF](#), which is more generally useful.



How Many?

Excel screenshot showing the formula bar with `=COUNT(B5:F5)` and the worksheet content.

Counting numbers and text with COUNT and COUNTA

	1	2	3	4	5		COUNT	COUNTA	Empty
5	25	50	75	100	125	-->	5	5	0
6	apple	pear	orange	peach	kiwi	-->	0	5	0
7	11-Mar	12-Mar	13-Mar	14-Mar	15-Mar	-->	5	5	0
8	apple	10	peach	12		-->	2	4	1
9	5%	10%	15%			-->	3	3	2

EXCELJET

```
= COUNT(B5:F5) // count numbers
= COUNTA(B5:F5) // count numbers and text
= COUNTIF(B5:F5,"") // count blanks
```

COUNTIF and COUNTIFS

For conditional counts, [the COUNTIF function](#) can apply one criteria. The [COUNTIFS function](#) can apply multiple criteria at the same time:

Excel screenshot showing the formula bar with `=COUNTIF(C5:C12,"red")` and the worksheet content.

Conditional counting with COUNTIF and COUNTIFS

Date	Color	State	Qty	Total
9-Jan	Red	TX	1	\$18.00
23-Jan	Blue	CO	2	\$34.00
3 Feb	Red	NM	2	\$36.00
18-Feb	Blue	TX	1	\$17.00
2-Mar	Blue	AZ	3	\$51.00
15-Mar	Red	AZ	1	\$17.00
25 Mar	Red	TX	2	\$36.00
2-Apr	Red	CO	4	\$72.00

Criteria	Count
Red	5
> 50	2
Red and TX	2
Blue > 50	1

EXCELJET

```
= COUNTIF(C5:C12,"red") // count red
= COUNTIF(F5:F12,">50") // count total > 50
= COUNTIFS(C5:C12,"red",D5:D12,"TX") // red and tx
= COUNTIFS(C5:C12,"blue",F5:F12,">50") // blue > 50
```

SUM, SUMIF, SUMIFS

To sum everything, use the [SUM function](#). To sum conditionally, use SUMIF or SUMIFS. Following the same pattern as the counting functions, the [SUMIF function](#) can apply only one criteria while the [SUMIFS function](#) can apply multiple criteria.

Formula bar: `=SUMIF(C5:C12,"red",F5:F12)`

Sums and conditional sums with SUM, SUMIF, and SUMIFS

Date	Color	State	Qty	Total
9-Jan	Red	TX	1	\$18.00
23-Jan	Blue	CO	2	\$34.00
3-Feb	Red	NM	2	\$36.00
18-Feb	Blue	TX	1	\$17.00
2-Mar	Blue	AZ	3	\$51.00
15-Mar	Red	AZ	1	\$17.00
25-Mar	Red	TX	2	\$36.00
2-Apr	Red	CO	4	\$72.00

Total	
Total	\$281.00
Red	\$179.00
> 50	\$123.00
Red and TX	\$54.00
Blue > 50	\$51.00

EXCELJET

```
=SUM(F5:F12) // everything
=SUMIF(C5:C12,"red",F5:F12) // red only
=SUMIF(F5:F12,">50") // over 50
=SUMIFS(F5:F12,C5:C12,"red",D5:D12,"tx") // red & tx
=SUMIFS(F5:F12,C5:C12,"blue",F5:F12,">50") // blue & >50
```

Averaging with AVERAGE, AVERAGEIF, and AVERAGEIFS

Date	Color	State	Qty	Total
9-Jan	Red	TX	1	\$18.00
23-Jan	Blue	CO	2	\$34.00
3-Feb	Red	NM	2	\$36.00
18-Feb	Blue	TX	1	\$17.00
2-Mar	Blue	AZ	3	\$51.00
15-Mar	Red	AZ	1	\$17.00
25-Mar	Red	TX	2	\$36.00
2-Apr	Red	CO	4	\$72.00

All	
All	\$35.13
Red	\$35.80
Red and TX	\$27.00

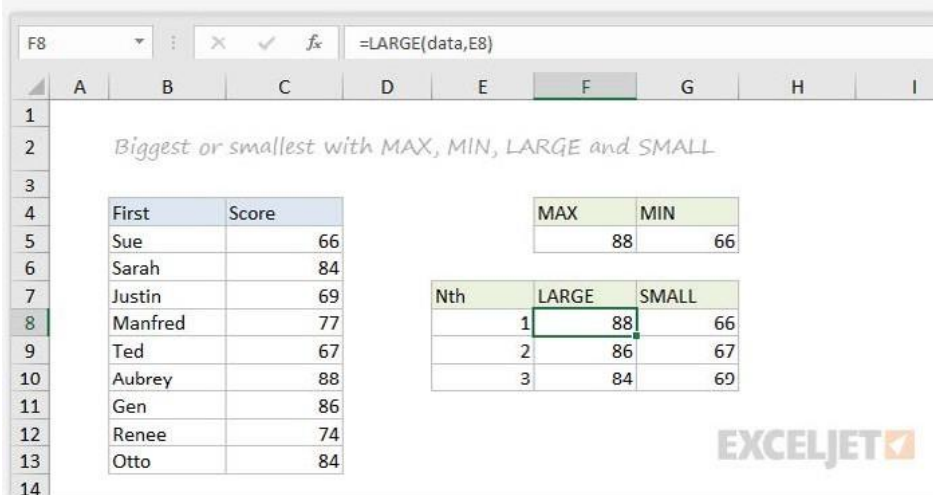
EXCELJET

```
=AVERAGE(F5:F12) // all
=AVERAGEIF(C5:C12,"red",F5:F12) // red only
=AVERAGEIFS(F5:F12,C5:C12,"red",D5:D12,"tx") // red and tx
```

5
9
7
—
23

MIN, MAX, LARGE, SMALL

You can find largest and smallest values with [MAX](#) and [MIN](#), and nth largest and smallest values with [LARGE](#) and [SMALL](#). In the screen below, “data” is the named range C5:C13, used in all formulas.



The screenshot shows an Excel worksheet with the following data:

First	Score
Sue	66
Sarah	84
Justin	69
Manfred	77
Ted	67
Aubrey	88
Gen	86
Renee	74
Otto	84

Summary statistics:

	MAX	MIN
	88	66

Nth	LARGE	SMALL
1	88	66
2	86	67
3	84	69

The formula bar shows `=LARGE(data,E8)`.



Find *LARGEST* and *SMALLEST* values

```
=MAX(data) // largest
=MIN(data) // smallest
=LARGE(data,1) // 1st largest
=LARGE(data,2) // 2nd largest
=LARGE(data,3) // 3rd largest
=SMALL(data,1) // 1st smallest
=SMALL(data,2) // 2nd smallest
=SMALL(data,3) // 3rd smallest
```

MINIFS, MAXIFS

The [MINIFS](#) and [MAXIFS](#). These functions let you find minimum and maximum values with conditions:

G5 `=MAXIFS(D5:D15,C5:C15,"female")`

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										

MINIFS and MAXIFS for min and max values with criteria

Name	Gender	Score
Hannah	Female	93
Edward	Male	79
Miranda	Female	85
William	Male	64
Joanna	Female	81
Collin	Male	83
Oscar	Male	64
Arturo	Male	76
Annie	Female	72
Weston	Male	64
Cassidy	Female	83

Gender	MAXIF	MINIF
Female	93	72
Male	83	64

EXCELJET

```
=MAXIFS(D5:D15,C5:C15,"female") // highest female  
=MAXIFS(D5:D15,C5:C15,"male") // highest male  
=MINIFS(D5:D15,C5:C15,"female") // lowest female  
=MINIFS(D5:D15,C5:C15,"male") // lowest male
```

Note: MINIFS and MAXIFS are new in Excel via Office 365 and Excel 2019.

MODE

The [MODE function](#) returns the most commonly occurring number in a range:

I5 `=MODE(B5:G5)`

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2												
3												
4												
5												
6												
7												
8												
9												

Find most frequently occurring number with MODE

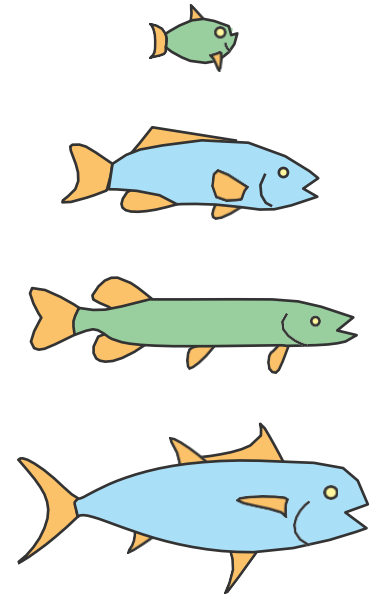
	1	2	3	4	5	6
1	1	2	2	1	1	2
5	5	10	15	15	10	5
69	69	70	70	71	71	70
95	95	115	125	115	95	115

MODE
1
5
70
115

EXCELJET

```
=MODE(B5:G5) // returns 1
```

Find the smallest blue fish



Most commonly occurring dog color



RANK

To rank values largest to smallest, or smallest to largest, use the [RANK function](#):

E5 =RANK(D5,\$D\$5:\$D\$12)

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								

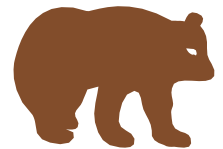
Assigning rank with the RANK function

City	State	Population	Rank
Houston	TX	2,100,263	4
Phoenix	AZ	1,445,632	6
New York	NY	8,175,133	1
Philadelphia	PE	1,526,006	5
Los Angeles	CA	3,792,621	2
San Antonio	TX	1,327,407	7
San Diego	CA	1,307,402	8
Chicago	IL	2,695,598	3

EXCELJET

See [this formula](#) which demonstrates how to use the RANK function to calculate race results.

Rank Largest to Smallest



MATH Functions

ABS

To change negative values to positive use the [ABS function](#).

Input	Output
-134.50	\$134.50
500.00	500
5.13	5.125
-0.13	\$0.13
-43.00	\$43.00

```
=ABS(-134.50) // returns 134.50
```

ABS – Negative to Positive

-100.00
↓
\$100.00

RAND and RANDBETWEEN

Both the [RAND function](#) and [RANDBETWEEN function](#) can generate random numbers on the fly. RAND creates long decimal numbers between zero and 1. RANDBETWEEN generates random integers between two given numbers.

RAND	RANDBETWEEN
0.351613613	58
0.301564967	22
0.683756914	4
0.673618677	66
0.749792539	27
0.770318131	58

```
=RAND() // between zero and 1  
=RANDBETWEEN(1,100) // between 1 and 100
```



To generate a random dice roll

RANDBETWEEN(1,6)

ROUND, ROUNDUP, ROUNDDOWN, INT

To round values up or down, use the [ROUND function](#). To force rounding up to a given number of digits, use [ROUNDUP](#). To force rounding down, use [ROUNDDOWN](#). To discard the decimal part of a number altogether, use the [INT function](#).

Number	Digits
11.777	1
15.11	1
13.85	1
9.91	1

ROUND	ROUNDUP	ROUNDDOWN	INT
11.8	11.8	11.7	11
15.1	15.2	15.1	15
13.9	13.9	13.8	13
9.9	10	9.9	9

```
=ROUND(11.777,1) // returns 11.8  
=ROUNDUP(11.777) // returns 11.8  
=ROUNDDOWN(11.777,1) // returns 11.7  
=INT(11.777) // returns 11
```

MROUND, CEILING, FLOOR

To round values to a the nearest multiple use the [MROUND function](#). The [FLOOR function](#) and [CEILING function](#) also round to a given multiple. FLOOR forces rounding down, and CEILING forces rounding up.

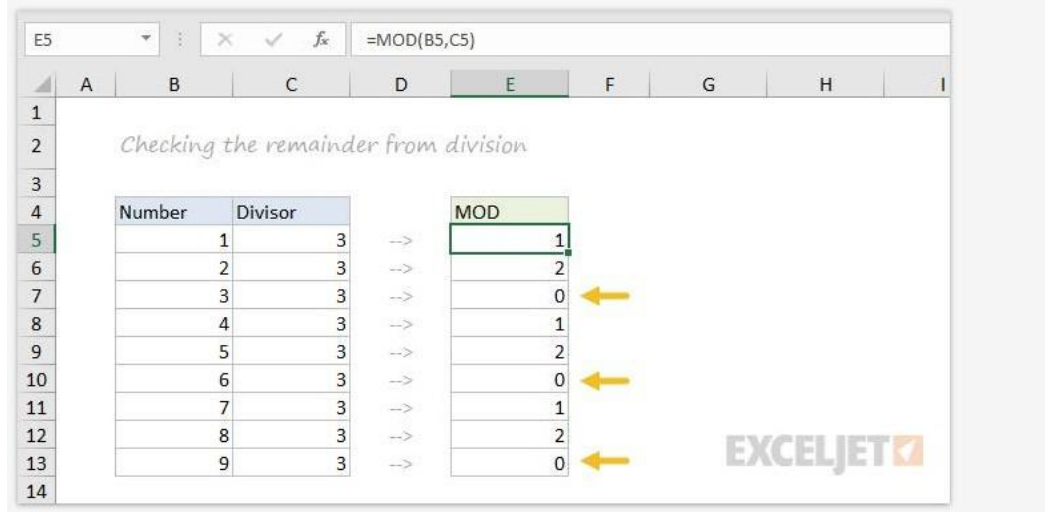
Number	Multiple
11.777	0.25
15.49	0.25
13.85	0.25
10.05	0.25

MROUND	CEILING	FLOOR
11.75	12	11.75
15.5	15.5	15.25
13.75	14	13.75
10	10.25	10

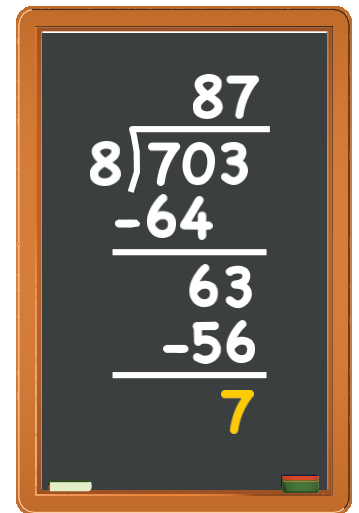
```
=MROUND(13.85,.25) // returns 13.75  
=CEILING(13.85,.25) // returns 14  
=FLOOR(13.85,.25) // returns 13.75
```

MOD

The [MOD function](#) returns the remainder after division. This sounds boring and geeky, but MOD turns up in all kinds of formulas, especially formulas that need to do something “every nth time”. In the screen below, you can see how MOD returns zero every third number when the divisor is 3:



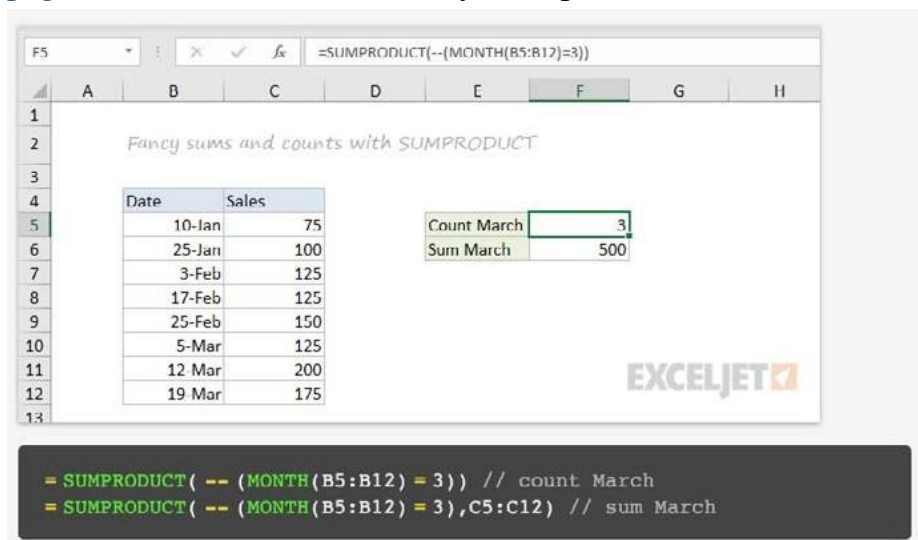
Number	Divisor	MOD
1	3	1
2	3	2
3	3	0
4	3	1
5	3	2
6	3	0
7	3	1
8	3	2
9	3	0



The [MOD function](#) returns the remainder

SUMPRODUCT

The [SUMPRODUCT](#) function is a powerful and versatile tool when dealing with all kinds of data. You can use SUMPRODUCT to easily count and sum based on criteria, and you can use it in elegant ways that just don’t work with COUNTIFS and SUMIFS. In the screen below, we are using SUMPRODUCT to count and sum orders in March. See the [SUMPRODUCT page](#) for details and links to many examples.



Date	Sales
10-Jan	75
25-Jan	100
3-Feb	125
17-Feb	125
25-Feb	150
5-Mar	125
12-Mar	200
19-Mar	175

Count March: 3
Sum March: 500

```
=SUMPRODUCT(--(MONTH(B5:B12)=3)) // count March  
=SUMPRODUCT(--(MONTH(B5:B12)=3),C5:C12) // sum March
```

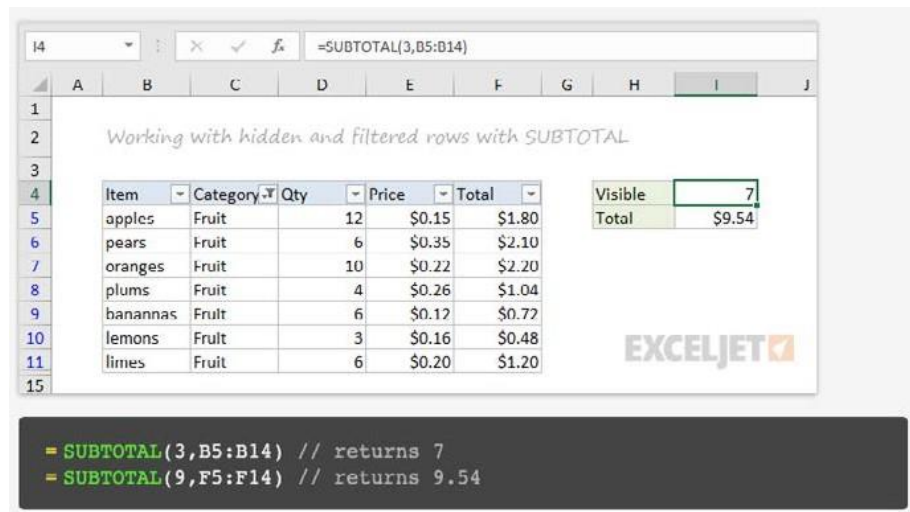


Sunglasses
sold in July

Sumproduct is a powerful and versatile tool that is easy to use.

SUBTOTAL

The [SUBTOTAL function](#) is an “aggregate function” that can perform a number of operations on a set of data. The key feature of SUBTOTAL is that it will ignore rows that have been “filtered out” of an [Excel Table](#), and, optionally, rows that have been manually hidden. In the screen below, SUBTOTAL is used to count and sum only the 7 visible rows in the table:



Working with hidden and filtered rows with SUBTOTAL

Item	Category	Qty	Price	Total
apples	Fruit	12	\$0.15	\$1.80
pears	Fruit	6	\$0.35	\$2.10
oranges	Fruit	10	\$0.22	\$2.20
plums	Fruit	4	\$0.26	\$1.04
banannas	Fruit	6	\$0.12	\$0.72
lemons	Fruit	3	\$0.16	\$0.48
limes	Fruit	6	\$0.20	\$1.20

Visible: 7
Total: \$9.54

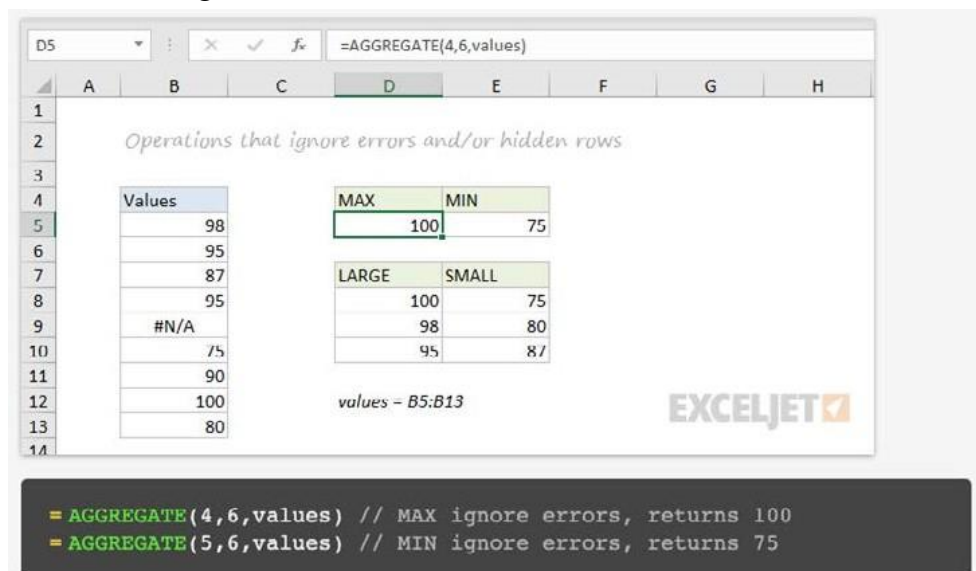
EXCELJET

```
= SUBTOTAL(3,B5:B14) // returns 7  
= SUBTOTAL(9,F5:F14) // returns 9.54
```

SUBTOTAL can perform 11 operations, including SUM, AVERAGE, COUNT, MAX, MIN, etc. (see [This Page](#) for the full list)

AGGREGATE

Like SUBTOTAL, the [AGGREGATE function](#) can run a number of aggregate operations on a set of data and can optionally ignore hidden rows. The key differences are that AGGREGATE can run more operations and can also ignore errors.



Operations that ignore errors and/or hidden rows

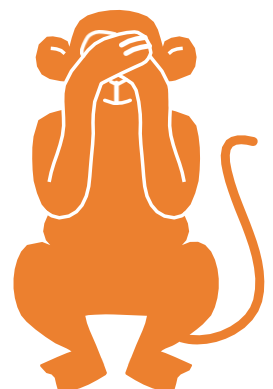
Values	MAX	MIN
98	100	75
95		
87		
95		
#N/A		
75		
90		
100		
80		

values = B5:B13

EXCELJET

```
= AGGREGATE(4,6,values) // MAX ignore errors, returns 100  
= AGGREGATE(5,6,values) // MIN ignore errors, returns 75
```

AGGREGATE can perform 19 operations and can also ignore errors. See [this page](#) for a full list of operations.

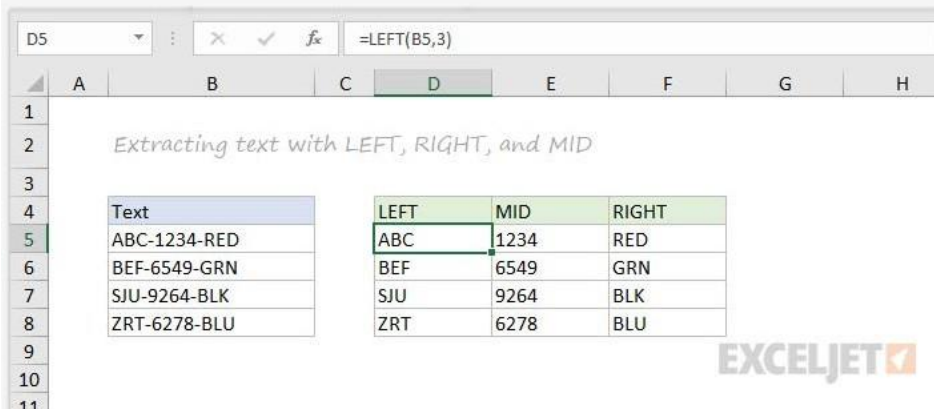


Above, AGGREGATE is used to perform MIN, MAX, LARGE and SMALL operations while ignoring errors. Normally, the error in cell B9 would prevent these functions from returning a result.

TEXT Functions

LEFT, RIGHT, MID

To extract characters from the left, right, or middle of text, use [LEFT](#), [RIGHT](#), and [MID](#) functions:



The screenshot shows an Excel spreadsheet with the following data:

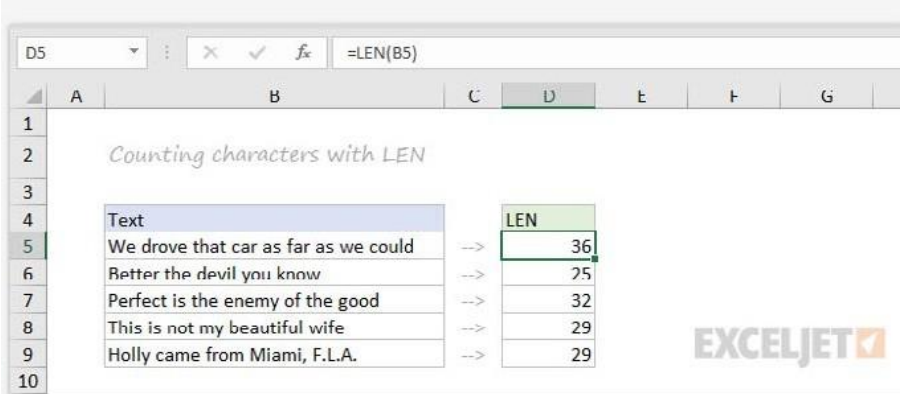
Text	LEFT	MID	RIGHT
ABC-1234-RED	ABC	1234	RED
BEF-6549-GRN	BEF	6549	GRN
SJU-9264-BLK	SJU	9264	BLK
ZRT-6278-BLU	ZRT	6278	BLU

The formula bar shows `=LEFT(B5,3)`. The text "Extracting text with LEFT, RIGHT, and MID" is written in the background.

```
=LEFT("ABC-1234-RED",3) // returns "ABC"  
=MID("ABC-1234-RED",5,4) // returns "1234"  
=RIGHT("ABC-1234-RED",3) // returns "RED"
```

LEN

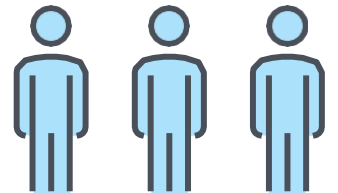
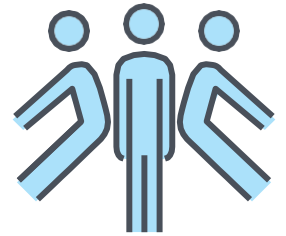
The [LEN function](#) will return the length of a text string. LEN shows up in a lot of formulas that count words or [characters](#).



The screenshot shows an Excel spreadsheet with the following data:

Text	LEN
We drove that car as far as we could	36
Better the devil you know	25
Perfect is the enemy of the good	32
This is not my beautiful wife	29
Holly came from Miami, F.L.A.	29

The formula bar shows `=LEN(B5)`. The text "Counting characters with LEN" is written in the background.



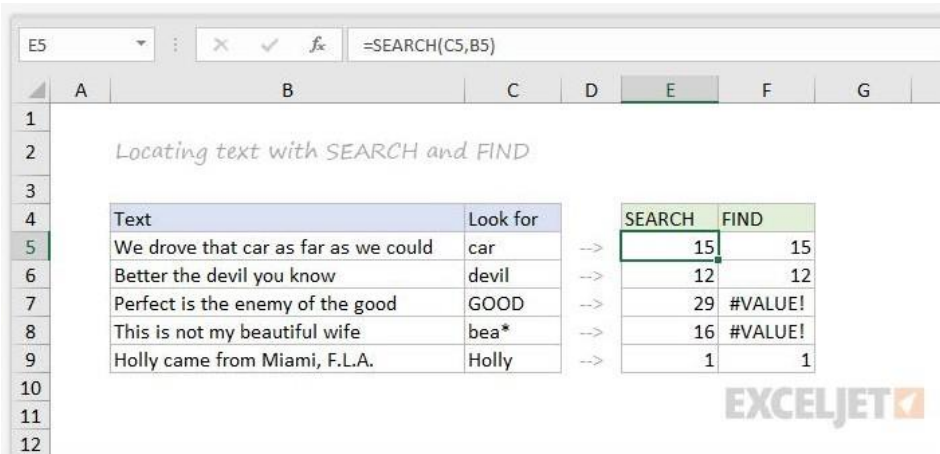
Separate text into columns

FORMULA

Get first name from full name

FIND, SEARCH

To look for specific text in a cell, use the [FIND function](#) or [SEARCH function](#). These functions return the numeric position of matching text, but SEARCH allows wildcards and FIND is case-sensitive. Both functions will throw an error when text is not found, so wrap in the [ISNUMBER function](#) to return TRUE or FALSE ([example here](#)).



Text	Look for	SEARCH	FIND
We drove that car as far as we could	car	15	15
Better the devil you know	devil	12	12
Perfect is the enemy of the good	GOOD	29	#VALUE!
This is not my beautiful wife	bea*	16	#VALUE!
Holly came from Miami, F.L.A.	Holly	1	1

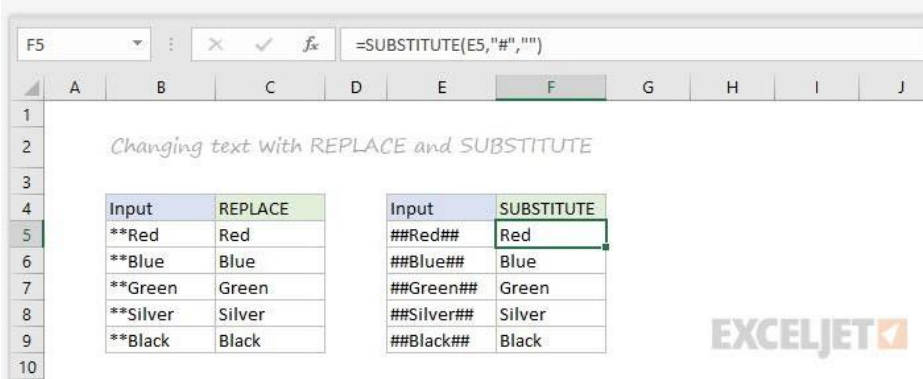
```
=FIND("Better the devil you know","devil") // returns 12  
=SEARCH("This is not my beautiful wife","bea*") // returns 12
```



**Text Functions –
Find, Replace and
Sustitute**

REPLACE, SUBSTITUTE

To replace text by position, use the [REPLACE function](#). To replace text by matching, use the [SUBSTITUTE function](#). In the first example, REPLACE removes the two asterisks (**) by replacing the first two characters with an empty string (""). In the second example, SUBSTITUTE removes all hash characters (#) by replacing "#" with "".



Input	REPLACE
**Red	Red
**Blue	Blue
**Green	Green
**Silver	Silver
**Black	Black

Input	SUBSTITUTE
##Red##	Red
##Blue##	Blue
##Green##	Green
##Silver##	Silver
##Black##	Black

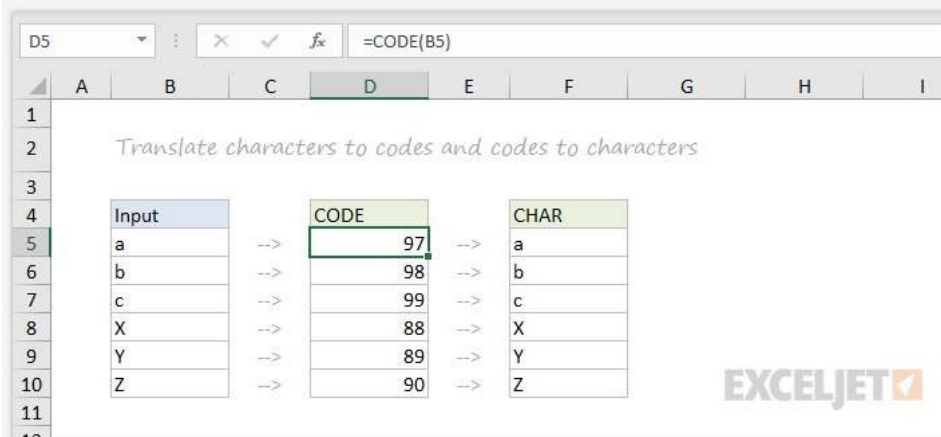
```
=REPLACE("**Red",1,2,"") // returns "Red"  
=SUBSTITUTE("##Red##","#","") // returns "Red"
```

FORMULA

**Cell contains one
of many things**

CODE, CHAR

To figure out the numeric code for a character, use the [CODE function](#). To translate the numeric code back to a character, use the [CHAR function](#). In the example below, CODE translates each character in column B to its corresponding code. In column F, CHAR translates the code back to a character.



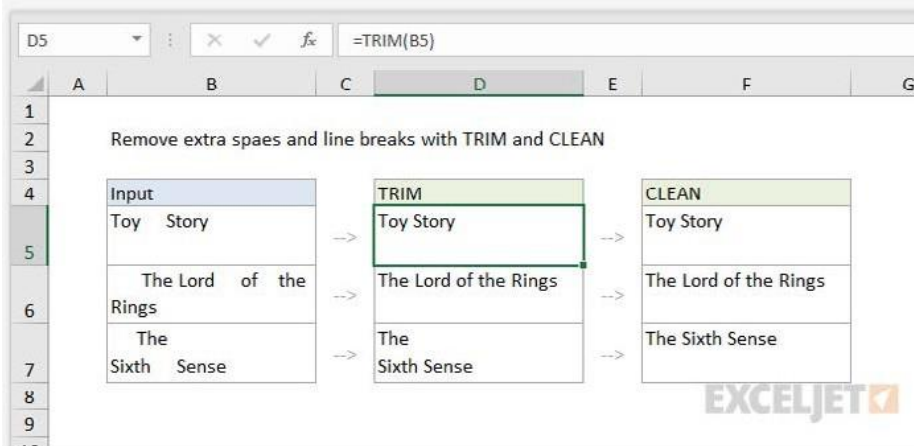
Input		CODE		CHAR
a	-->	97	-->	a
b	-->	98	-->	b
c	-->	99	-->	c
X	-->	88	-->	X
Y	-->	89	-->	Y
Z	-->	90	-->	Z



```
=CODE("a") // returns 97  
=CHAR(97) // returns "a"
```

TRIM, CLEAN

To get rid of extra space in text, use the [TRIM function](#). To remove line breaks and other non-printing characters, use [CLEAN](#).

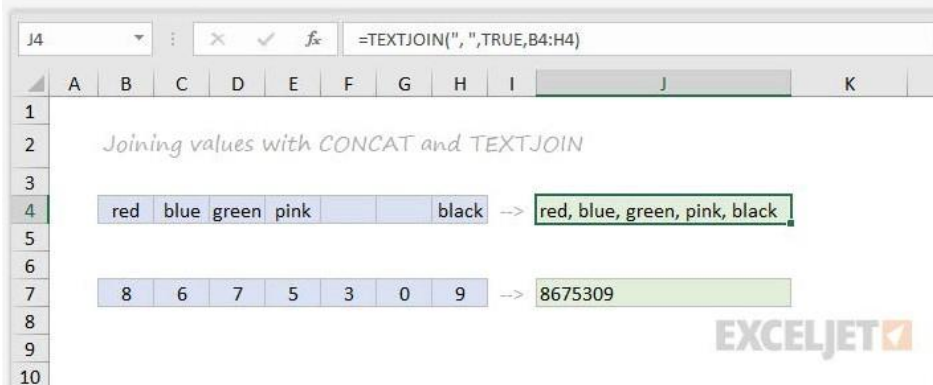


Input		TRIM		CLEAN
Toy Story	-->	Toy Story	-->	Toy Story
The Lord of the Rings	-->	The Lord of the Rings	-->	The Lord of the Rings
The Sixth Sense	-->	The Sixth Sense	-->	The Sixth Sense

```
=TRIM(A1) // remove extra space  
=CLEAN(A1) // remove line breaks
```


CONCAT, TEXTJOIN, CONCATENATE

New in Excel via Office 365 are CONCAT and TEXTJOIN. The [CONCAT function](#) lets you concatenate (join) multiple values, including a range of values without a delimiter. The [TEXTJOIN function](#) does the same thing, but allows you to specify a delimiter and can also ignore empty values.



The screenshot shows an Excel spreadsheet with the following content:

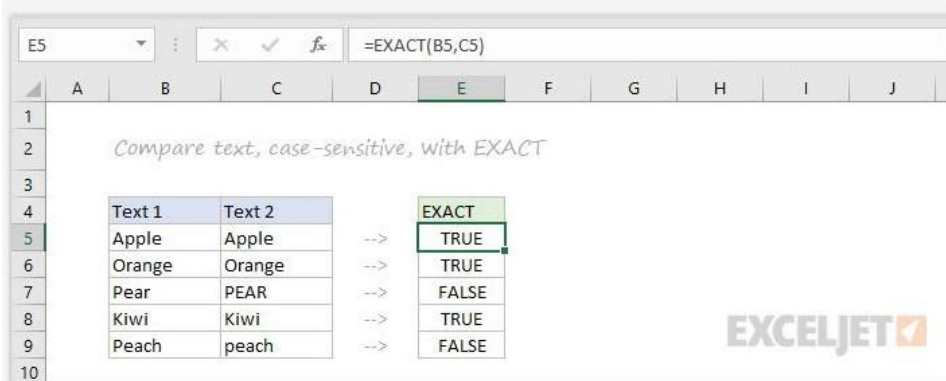
	A	B	C	D	E	F	G	H	I	J	K
1											
2										Joining values with CONCAT and TEXTJOIN	
3											
4		red	blue	green	pink			black	-->	red, blue, green, pink, black	
5											
6											
7		8	6	7	5	3	0	9	-->	8675309	
8											
9											
10											

```
=TEXTJOIN(", ",TRUE,B4:H4) // returns "red,blue,green,pink,black"  
=CONCAT(B7:H7) // returns "8675309"
```

Excel also provides the [CONCATENATE function](#), but it doesn't offer special features. I wouldn't bother with it and would instead [concatenate](#) directly with the ampersand (&) character in a formula.

EXACT

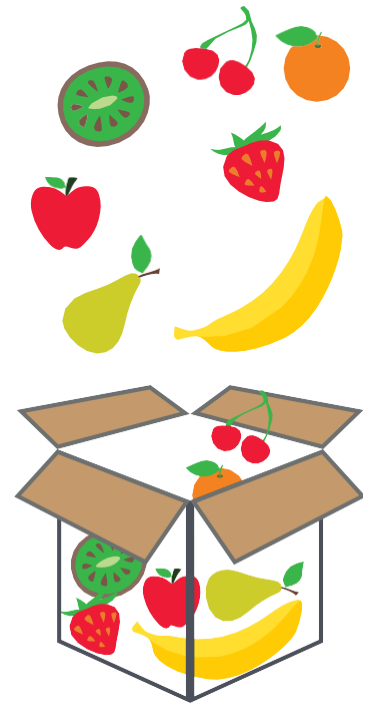
The [EXACT](#) function allows you to compare two text strings in a case-sensitive manner.



The screenshot shows an Excel spreadsheet with the following content:

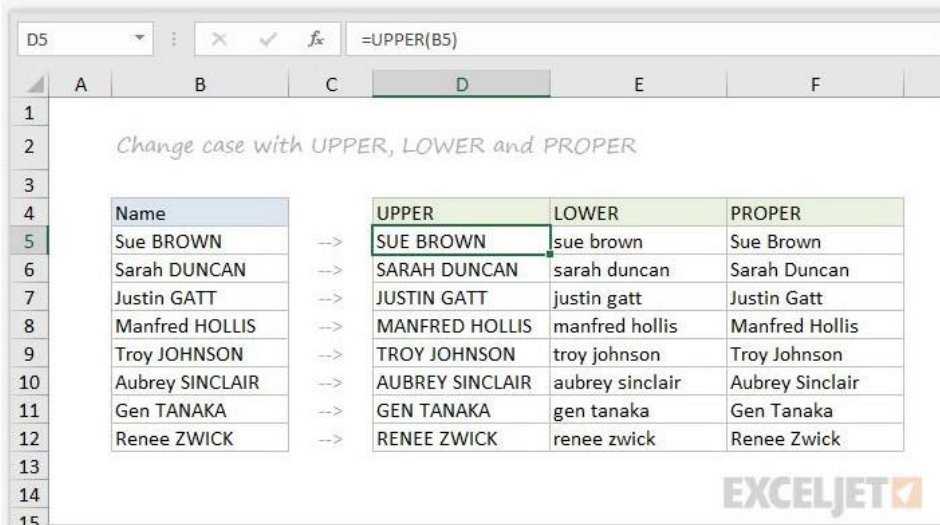
	A	B	C	D	E	F	G	H	I	J
1										
2										Compare text, case-sensitive, with EXACT
3										
4		Text 1	Text 2		EXACT					
5		Apple	Apple	-->	TRUE					
6		Orange	Orange	-->	TRUE					
7		Pear	PEAR	-->	FALSE					
8		Kiwi	Kiwi	-->	TRUE					
9		Peach	peach	-->	FALSE					
10										

Join cells together



UPPER, LOWER, PROPER

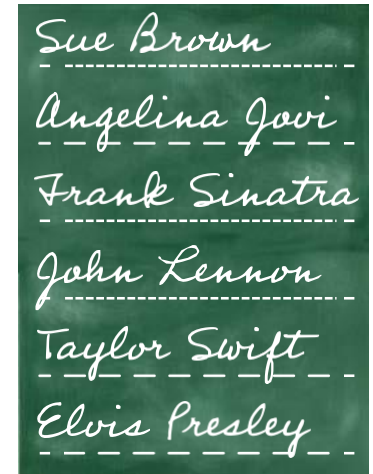
To change the case of text, use the [UPPER](#), [LOWER](#), and [PROPER](#) function



Name		UPPER	LOWER	PROPER
Sue BROWN	-->	SUE BROWN	sue brown	Sue Brown
Sarah DUNCAN	-->	SARAH DUNCAN	sarah duncan	Sarah Duncan
Justin GATT	-->	JUSTIN GATT	justin gatt	Justin Gatt
Manfred HOLLIS	-->	MANFRED HOLLIS	manfred hollis	Manfred Hollis
Troy JOHNSON	-->	TROY JOHNSON	troy johnson	Troy Johnson
Aubrey SINCLAIR	-->	AUBREY SINCLAIR	aubrey sinclair	Aubrey Sinclair
Gen TANAKA	-->	GEN TANAKA	gen tanaka	Gen Tanaka
Renee ZWICK	-->	RENEE ZWICK	renee zwick	Renee Zwick

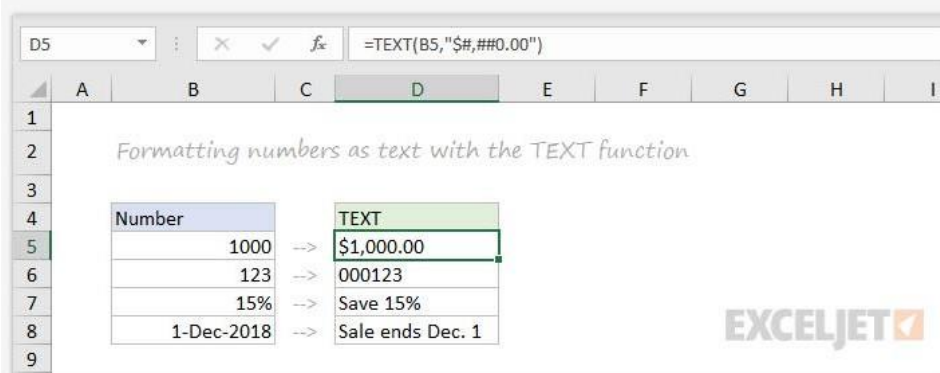
```
=UPPER("Sue BROWN") // returns "SUE BROWN"  
=LOWER("Sue BROWN") // returns "sue brown"  
=PROPER("Sue BROWN") // returns "Sue Brown"
```

Standardize a List of Names



TEXT

Last but definitely not least is the [TEXT function](#). The text function lets you apply number formatting to numbers (including dates, times, etc.) as text. This is especially useful when you need to embed a formatted number in a message, like “Sale ends on [date]”.



Number		TEXT
1000	-->	\$1,000.00
123	-->	000123
15%	-->	Save 15%
1-Dec-2018	-->	Sale ends Dec. 1

```
=TEXT(B5,"$#,##0.00")  
=TEXT(B6,"000000")  
="Save " & TEXT(B7,"0%")  
="Sale ends " & TEXT(B8,"mmm d")
```