FREE eBook

LEARNING excel

Free unaffiliated eBook created from
Stack Overflow contributors.

#excel

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: excel

It is an unofficial and free excel ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official excel.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with excel

Remarks

Microsoft Excel is a high-utility spreadsheet for use on calculation and macro programming. Excel also boasts compatibility with Visual Basic for Applications. Data is arranged and evaluated in a grid of cells, where it can be formulated or displayed in charts.

Examples

Examples of some basic Formulas available in Excel

Excel has many Formulas built right in. Here are a few examples of some of the basic Formulas that might be handy to know when getting started with Excel:

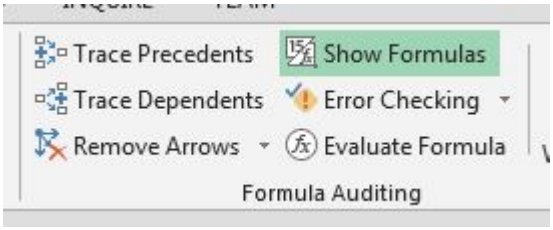
Important Note : The name and Syntax of these Formulas vary on the language of your Excel installation! For example this Function here :

in English : `=left("Hello World",5)`

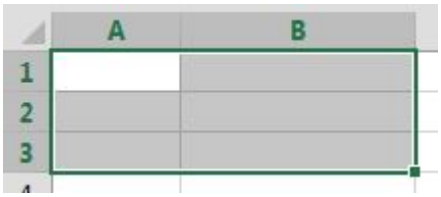
Same in German: `=links("Hello World";5)`

Not only the name is different but also some minor parts of the syntax! For example in the German version ; is used instead of , to separate the parameters.

All Formulas can be entered into any cell by first typing "=" then the name of the Formula. You can type the Formulas directly into the cell or you can select the cell and type the Formula into the formula bar. To show all the formulas on a sheet you can select the "Show Formulas" button in the "Formula Auditing" group of buttons on the Formulas tab:



A lot of Formulas use a range of cells. To give a range you reference the first cell in the top left corner such as `A1` and the bottom right corner such as `B3` and place a colon ":" between them like this `A1:B3`. This will give you a range of 6 cells in two columns and three rows:

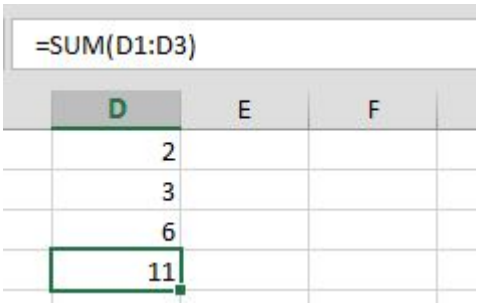


SUM()

In cells D1 - D3 are the values 2,3 and 6 In cell D4 is the following Formula: `=SUM(D1:D3)`, The Result is 11 which shows in the cell.

`Sum` Adds all numbers in a range of cells.

If you have that cell selected then the Formula will show in the Formula bar above the grid of cells:



COUNT()/COUNTA()/COUNTIF()/COUNTBLANK()

In cells E1 - E10 are the values 2,3,5,6, blank,8,9, blank,11 and 12

In cell F2 is the Formula: `=count(E1:E10)`, the Result is 8 which shows in the cell.

In cell F3 is the Formula: `=counta(E1:E10)`, the Result is 8 which shows in the cell.

In cell F4 is the Formula: `=countif(E1:E10,">5")`, the Result is 5 which shows in the cell.

In cell F5 is the Formula: `=countblank(E1:E10)`, the Result is 2 which shows in the cell.

`Count()` Counts the number of cells in a range that contain numbers.

`Counta()` Counts the number of cells in a range that are not empty.

`CountIF()` Counts the number of cells in a range that meet a given condition.

`CountBlank()` Counts the number of empty cells in a specified range of cells.

E	F
2	
3	=COUNT(E1:E10)
5	=COUNTA(E1:E10)
6	=COUNTIF(E1:E10,">5")
	=COUNTBLANK(E1:E10)
8	
9	
11	
12	

There are also many string Formulas.

LEFT()/RIGHT()/MID()

In cell D1 is "Hello World!".

In cell E1 is the Formula: `=left(D1,5)`, the Result is: "Hello"

In cell F1 is the Formula: `=right(D1,6)`, the Result is: "World!"

In cell G1 is the Formula: `=mid(D1,7,5)`, the Result is: "World"

`left` Returns the specified number of characters from the start of a text string. `=LEFT(String,Number of Characters)`

`right` Returns the specified number of characters from the end of a text string. `=RIGHT(String, Number of Characters)`

`mid` Returns the characters from the middle of a text string, given the starting position and length. `=MID(String, Start Position, Number of Characters)`

These three Formulas count the characters of the string starting with position 1 being the first character.

Read Getting started with excel online: https://riptutorial.com/excel/topic/906/getting-started-with-excel

Chapter 2: An easy process to convert Monthly to Quarterly data in Excel

Introduction

A frequent desire is to convert monthly data into quarterly data format. One simple method is to 1) sum the appropriate months into quarterly sums, then 2) identify those months as quarters, and finally 3) filter the quarterly data out of your monthly data. Here's a relatively quick and easy method illustrated through the following example:

Examples

An easy way to perform a sometimes tedious routine task:

If you have your data with each month of data arranged in rows like so:

	A	B	C
1	**Date**	**Product A**	**Service A**
2	2000-01-31	178.00	358.00
3	2000-02-29	619.00	553.00
4	2000-03-31	152.00	106.00
5	2000-04-30	662.00	102.00
6	2000-05-31	230.00	290.00
7	2000-06-30	248.00	583.00
8	2000-07-31	438.00	413.00
9	2000-08-31	428.00	401.00
10	2000-09-30	481.00	488.00
11	2000-10-31	631.00	433.00
12	2000-11-30	436.00	449.00
13	2000-12-31	378.00	300.00
14	2001-01-31	641.00	478.00
15	2001-02-28	560.00	550.00
16	2001-03-31	310.00	534.00
17	2001-04-30	301.00	402.00
18	2001-05-31	631.00	397.00
19	2001-06-30	490.00	268.00
20	2001-07-31	234.00	378.00
21	2001-08-31	516.00	220.00
22	2001-09-30	167.00	116.00

start by creating quarterly sums in the adjacent columns, D & E in our example.

Start with the third row in the new column or the first quarter you want to create, in this example we'll use March 31st (2000-03-31). Use the summation function over the prior two months and the quarter month, for March use:

=SUM(B2:B4)

Then use a similar function for the total in the next column E. It will have the following formula:

=SUM(C2:C4)

Don't concern yourself that the sums are created for months that are not quarters. This is done to make the copying easier for you. You'll not need those values anyway in the next few steps.

Finally, we need to identify quarters without a lot of work. If your Date column isn't in an Excel Date format, such as text, you may want to convert it. In a new column, F in this example, begin this formula in the first row of data. We will mark the rows using the following formula:

=IF(MOD(MONTH(A2),3)=0, "Quarter", "Month")

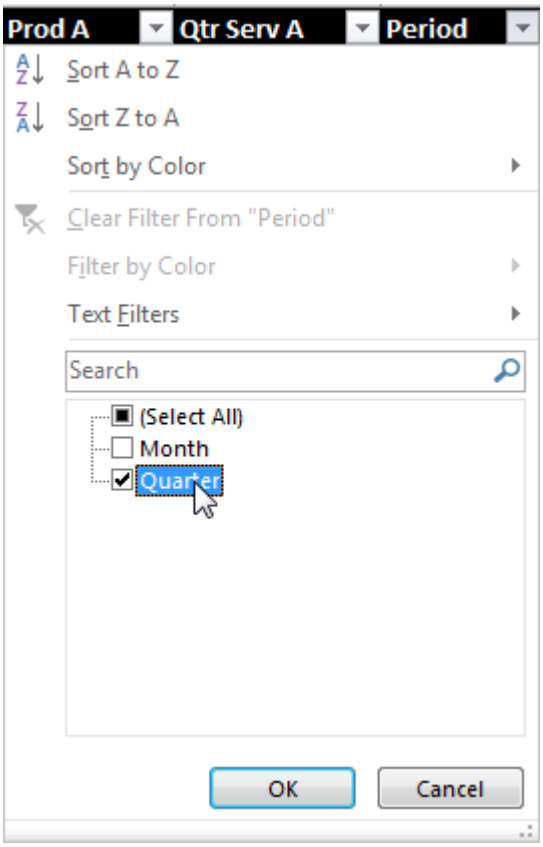
Now copy the three formulas in the new columns D,E & F into the remaining cells in your table. It should look like this:

	A	B	C	D	E	F
1	**Date**	**Product A**	**Service A**	**Qtr Prod A**	**Qtr Serv A**	**Period**
2	2000-01-31	330.00	621.00			Month
3	2000-02-29	245.00	664.00			Month
4	2000-03-31	370.00	230.00	945.00	1,515.00	Quarter
5	2000-04-30	437.00	467.00	1,052.00	1,361.00	Month
6	2000-05-31	629.00	260.00	1,436.00	957.00	Month
7	2000-06-30	218.00	198.00	1,284.00	925.00	Quarter
8	2000-07-31	214.00	661.00	1,061.00	1,119.00	Month
9	2000-08-31	309.00	136.00	741.00	995.00	Month
10	2000-09-30	239.00	566.00	762.00	1,363.00	Quarter
11	2000-10-31	126.00	103.00	674.00	805.00	Month
12	2000-11-30	201.00	375.00	566.00	1,044.00	Month
13	2000-12-31	587.00	672.00	914.00	1,150.00	Quarter
14	2001-01-31	341.00	376.00	1,129.00	1,423.00	Month
15	2001-02-28	484.00	186.00	1,412.00	1,234.00	Month
16	2001-03-31	535.00	141.00	1,360.00	703.00	Quarter
17	2001-04-30	594.00	210.00	1,613.00	537.00	Month
18	2001-05-31	309.00	246.00	1,438.00	597.00	Month
19	2001-06-30	514.00	145.00	1,417.00	601.00	Quarter
20	2001-07-31	665.00	568.00	1,488.00	959.00	Month
21	2001-08-31	408.00	666.00	1,587.00	1,379.00	Month
22	2001-09-30	456.00	233.00	1,529.00	1,467.00	Quarter

Finally, select this table, choose **Format As Table**, pick a simple format, and Excel will produce a result like so:

	A	B	C	D	E	F
1	Date	Product A	Service A	Qtr Prod A	Qtr Serv A	Period
2	2000-01-31	343.00	594.00			Month
3	2000-02-29	278.00	425.00			Month
4	2000-03-31	286.00	175.00	907.00	1,194.00	Quarter
5	2000-04-30	548.00	403.00	1,112.00	1,003.00	Month
6	2000-05-31	452.00	305.00	1,286.00	883.00	Month
7	2000-06-30	226.00	257.00	1,226.00	965.00	Quarter
8	2000-07-31	393.00	546.00	1,071.00	1,108.00	Month
9	2000-08-31	622.00	185.00	1,241.00	988.00	Month
10	2000-09-30	104.00	576.00	1,119.00	1,307.00	Quarter
11	2000-10-31	517.00	623.00	1,243.00	1,384.00	Month
12	2000-11-30	101.00	542.00	722.00	1,741.00	Month
13	2000-12-31	164.00	192.00	782.00	1,357.00	Quarter
14	2001-01-31	130.00	328.00	395.00	1,062.00	Month
15	2001-02-28	393.00	559.00	687.00	1,079.00	Month
16	2001-03-31	238.00	183.00	761.00	1,070.00	Quarter
17	2001-04-30	516.00	485.00	1,147.00	1,227.00	Month
18	2001-05-31	313.00	289.00	1,067.00	957.00	Month
19	2001-06-30	316.00	422.00	1,145.00	1,196.00	Quarter
20	2001-07-31	232.00	416.00	861.00	1,127.00	Month
21	2001-08-31	612.00	175.00	1,160.00	1,013.00	Month
22	2001-09-30	167.00	277.00	1,011.00	868.00	Quarter

The final step is to select the **filter button** over the new "Period" column in F and select only the "Quarters" values to filter on:



Your table should look like this now:

	A	B	C	D	E	F
1	Date	Product A	Service A	Qtr Prod A	Qtr Serv A	Period
4	2000-03-31	343.00	630.00	1,375.00	1,234.00	Quarter
7	2000-06-30	148.00	654.00	1,042.00	1,043.00	Quarter
10	2000-09-30	385.00	161.00	1,618.00	813.00	Quarter
13	2000-12-31	118.00	477.00	1,257.00	1,525.00	Quarter
16	2001-03-31	602.00	440.00	1,308.00	1,150.00	Quarter
19	2001-06-30	308.00	420.00	784.00	897.00	Quarter
22	2001-09-30	479.00	433.00	1,466.00	1,580.00	Quarter

At this point I would suggest selecting only the columns you need from your final table and then using **Paste->Special->Values** to relocate your data.

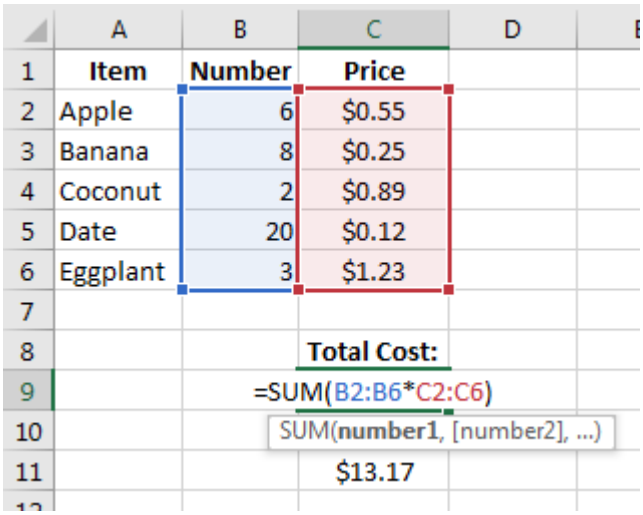
Read An easy process to convert Monthly to Quarterly data in Excel online: https://riptutorial.com/excel/topic/10837/an-easy-process-to-convert-monthly-to-quarterly-data-in-excel

Chapter 3: Array formulas

Examples

Sum of Product of Ranges

In this example, the total cost of buying food items is found by taking the number of each item and multiplying it by its cost and then adding all those values together.



Instead of creating a separate column for `Number` times `Price` and then summing the values in that new column, we can calculate the total price directly using an array formula:

```
=SUM(B2:B6*C2:C6)
```

Since this is an array formula, it must be entered using `Ctrl`+`Shift`+`Enter` in order for Excel to treat it as such (otherwise will return `#VALUE!`). Notice that f you see curly brackets `{...}` around the formula in the formula bar, then you know it is being evaluated as an array formula.

This is how this formula is evaluated step-by-step:

```
= SUM(B2:B6*C2:C6)
= SUM({6, 8, 2, 20, 3} * {0.55, 0.25, 0.89, 0.12, 1.23})
= SUM({6 * 0.55, 8 * 0.25, 2 * 0.89, 20 * 0.12, 3 * 1.23})
= SUM({3.30, 2.00, 1.78, 2.40, 3.69})
= 3.30 + 2.00 + 1.78 + 2.40 + 3.69
= 13.17
```

Another way to do this is to use the `SUMPRODUCT` function:

```
=SUMPRODUCT(B2:B6,C2:C6)
```

Note: In this case, using `Ctrl`+`Shift`+`Enter` is not necessary.

Read Array formulas online: https://riptutorial.com/excel/topic/5992/array-formulas

Chapter 4: Base Conversion

Introduction

Excel gives functions that will assist in converting from decimal to one of binary, octal and hexadecimal, and back again.

Remarks

Note that there are no leading `0` or `0x` in the functions.

Examples

Base Conversion

with the number `100` in cell A1, the results of these calculations

```
=DEC2BIN(A1)
=DEC2OCT(A1)
=DEC2HEX(A1)
=BIN2DEC(A1)
=OCT2DEC(A1)
=HEX2DEC(A1)
```

is

```
1100100
144
64
       4
      64
     256
```

note that the first 3 functions are left justified, as they are strings, and the last 3 are right justified as they are numeric.

Read Base Conversion online: https://riptutorial.com/excel/topic/10708/base-conversion

Chapter 5: Cell Formatting

Introduction

Each cell in an Excel spreadsheet can have unique formatting for things such as font, number formatting, color, borders, and alignment. Conditional formatting allows formatting for cells to vary based on data in the spreadsheet.

See also Excel rounding and precision.

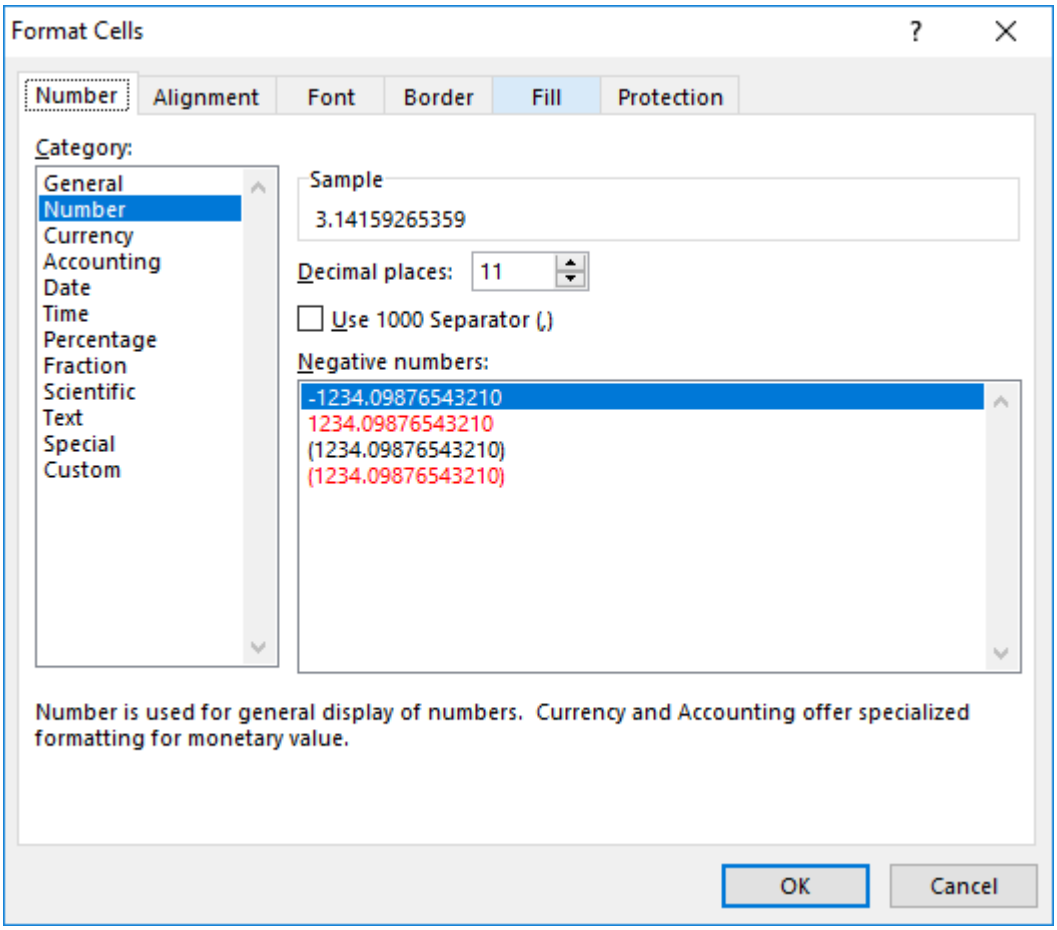
Remarks

Conditional Formatting allows the formatting of color, font effects, background color, etc. of cells on the spreadsheet to vary based on the cell's value, or other cell's values. A range of cells can have mini-charts or different icons in the cells based on their values.

Examples

Number Formatting

The most common use of formatting is to control display of numeric information contained in the cells in a consistent format, such as currency, a certain number of digits to the right of the decimal point, etc. There are several categories for numbers, such as Currency, Accounting, Percentage, and more. For each Category, there are various options available:



The cell formatting does *not* affect the internal value or formula results, just the displayed value. If a cell contains the formula `=4*ACOT(1)` Excel can display that as

- `3.14159265359` (Number with 11 decimal places)
- `$3.14` (US Currency format)
- `3 1/7` (Fraction with up to one digit)

You can create complex formats with different masks for positive, negative, and zero values. One shortcut is to use the existing Category closest to your needs, set the options, then click on the "Custom" Category, and you will see the formatting codes used, and you can alter them as needed.

The format code for "Fraction with up to one digit" is `# ?/?` and if you clicked Custom and entered `# "and" ?/?` for the format, the value displayed would be `3 and 1/7`

Read Cell Formatting online: https://riptutorial.com/excel/topic/9990/cell-formatting

Chapter 6: Counting unique cells

Examples

Using COUNTIF()

```
=SUMPRODUCT((A1:A100<>"")/COUNTIF(A1:A100,A1:A100&""))
```

counts *unique cell values* within A1:A100, *excluding* blank cells and ones with an empty string ("").

How does it do that? Example:

```
A1:A100 =    [1,   1,   2,   "apple", "peach", "apple", "", "", -,  -,  -,  ...]
```

Adding *&""* to the array is needed to turn blank cells (-) into empty strings (""). Result:

```
A1:A100&"" = ["1", "1", "2", "apple", "peach", "apple", "", "", "", "", "", ...]
```

After this trick, COUNTIF() can be applied. Both "" and - are counted as the same:

```
COUNTIF(A1:A100,A1:A100&"") = [2, 2, 1, 2, 1, 2, 94, 94, 94, 94, 94, ...]
```

To get the count of all unique cells, excluding blanks and "", we can divide

```
(A1:A100<>""), which is [1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, ...]
```

by our intermediate result, `COUNTIF(A1:A100,A1:A100&"")`, and sum up over the values.

```
SUMPRODUCT((A1:A100<>"")/COUNTIF(A1:A100,A1:A100&""))
= (1/2 + 1/2 + 1/1 + 1/2 + 1/1 + 1/2 + 0/94 + 0/94 + 0/94 + 0/94 + 0/94 + ...)
= 4
```

Using FREQUENCY() and MATCH()

```
=SUMPRODUCT(IF(FREQUENCY(MATCH(A1:A100,A1:A100,0),MATCH(A1:A100,A1:A100,0))>0,1))
```

Read Counting unique cells online: https://riptutorial.com/excel/topic/6263/counting-unique-cells

Chapter 7: Create Connection to other Excel Files

Examples

Creating an Connection to other Excel Files to query

There are many fine ways to get this done, which others have already suggested. Following along the "get Excel data via SQL track", here are some pointers.

Excel has the "Data Connection Wizard" which allows you to import or link from another data source or even within the very same Excel file. As part of Microsoft Office (and OS's) are two providers of interest: the old "Microsoft.Jet.OLEDB", and the latest "Microsoft.ACE.OLEDB". Look for them when setting up a connection (such as with the Data Connection Wizard). Once connected to an Excel workbook, a worksheet or range is the equivalent of a table or view. The table name of a worksheet is the name of the worksheet with a dollar sign ("$") appended to it, and surrounded with square brackets ("[" and "]"); of a range, it is simply the name of the range. To specify an unnamed range of cells as your recordsource, append standard Excel row/column notation to the end of the sheet name in the square brackets.

Naming the range of the data you wish to query using the name manager is very helpfull, as naming it "Database" can treat the data in a worksheet like a database table, allowiny you to run SQL statements on your data for easy access. It is also worth noting that the named range should not have blank or missing data, as it will cause the SQL to break.

The native SQL will (more or less be) the SQL of Microsoft Access. (In the past, it was called JET SQL; however Access SQL has evolved, and I believe JET is deprecated old tech.) Example, reading a worksheet: SELECT * FROM [Sheet1$] Example, reading a range: SELECT * FROM MyRange Example, reading an unnamed range of cells: SELECT * FROM [Sheet1$A1:B10] There are many many many books and web sites available to help you work through the particulars.

Read Create Connection to other Excel Files online: https://riptutorial.com/excel/topic/6328/create-connection-to-other-excel-files

Chapter 8: DATEDIF function

Syntax

- =DATEDIF(start_date,end_date,unit)

Parameters

Unit	Returns
"Y"	The number of complete years in the period
"M"	The number of complete months in the period
"D"	The number of days in the period
"MD"	The difference between the days in start_date and end_date. The months and years of the dates are ignored
"YM"	The difference between the months in start_date and end_date. The days and years of the dates are ignored
"YD"	The difference between the days of start_date and end_date. The years of the dates are ignored

Remarks

Be careful of Leap Year calculations when the units ignore years. For example:

```
=datedif("2010-01-01","2010-07-21","YD")
```

returns 201 days

```
=datedif("2016-01-01","2016-07-21","YD")
```

returns 202 days

Examples

Period count between dates

The `DATEDIF` function returns the difference between two date values, based on the interval specified. It is provided for compatibility with Lotus 1-2-3. The `DATEDIF` function cannot be found on the function list and autocomplete and screen tips are unavailable. *Note: It is pronounced "date*

diff" rather than "dated if".

```
=datedif("2010-01-01","2016-07-21","D")
```

returns the number of days (**2393**) between the two dates

```
=datedif("2010-01-01","2016-07-21","M")
```

returns the number of months (**78**) between the two dates

```
=datedif("2010-01-01","2016-07-21","Y")
```

returns the number of years (**6**) between the two dates

```
=datedif("2010-01-01","2016-07-21","MD")
```

returns the number of days (**20**) between the two dates-ignoring the months and years

```
=datedif("2010-01-01","2016-07-21","YM")
```

returns the number of months (**6**) between the two dates-ignoring the years

```
=datedif("2010-01-01","2016-07-21","YD")
```

returns the number of days (**201**) between the two dates-ignoring the years

Read DATEDIF function online: https://riptutorial.com/excel/topic/2786/datedif-function

Chapter 9: Excel best practice

Examples

Use Excel Tables

By selecting a matrix and choosing "Insert Table" from the menu, you create a table which allows you to pull and insert data in a structured way. Let's say you have named the table "SalesEvents" and given that the first (header) row reads "Salesperson" "Date" "Sales Amount", you can calculate sales amount like this: `=SUM(SalesEvents[Sales Amount])`. Entering data at the bottom of the table automatically adds new rows. This is a very good way of enhancing structure by encapsulating deltails inside your tables, leaving outside cells good posibilities to extract the high-level properties like column sums.

Use Pivot Tables

When you have an Excel Table, and not only then, it is easy to use data as input in a PivotTable which will provide most of the analysis you would need on it. Learn to use it you won't regret it! It could replaces tons of user designed cell Formulas and it is fast and much easier to document.

Pull data from PivotTables with GetPivotData

Sometimes it is hard to make all the data of your Pivot Table confirm to the reporting format you have to present your crunched data into. Then use GetPivotData! It has an automatic fill in of arguments that you easy can learn from and it lets you through its parameters flexibly choose and pick from all the visible fields in your Pivot Tables.

Best Practices

Here are some basic best practices for Excel:

- Flat File Database - Excel **IS** a Flat File application and should be treated as such
- Less Worksheet/Workbooks is more. Analyzing will be much faster with fewer worksheets/workbooks to go through. Try to ensure that all raw data is on one worksheet in one workbook
- The layout should consist of 1 sheet of raw data, your final data will come from this.
- Make headers **BOLD** this helps Excel recognize them as headers for things like sort
- When putting data into the data area of your spreadsheet try to avoid blank rows and columns. Excel might consider a blank row or column to be the end of your data. *It would be a good idea to leave the top 4 rows blank above your headers to use for totals instead of having them at the bottom*

- Sort your data whenever possible. This will help in speeding up some of the formulas and calculations that you have in the spreadsheet
- Use real dates for headings and then format them appropriately. By this I mean if you need month names for your headers use 1/1/2017, 1/2/2017, 1/3/2017 and then format them as "Mmmm". This is very simple nad will make life easier down the road when used in formulas.
- Don't put into 1 cell what can go into more than one. Meaning if you are making a list of users full names you should have the first name in one column and the last name in the next column (and possibly more columns for middle names and suffixes). It is easier to do this from the beginning than to try and do this with formulas later.
- Place your headings across the columns and your data in rows directly beneath. Excel has far more rows than it has columns. To make you spreadsheet future proof your raw data should have the data going down the rows so that it can continue well past the number of available columns.

Read Excel best practice online: https://riptutorial.com/excel/topic/6570/excel-best-practice

Chapter 10: Excel rounding and precision

Introduction

Several Excel formulas deal with rounding and precision of non-integer numbers. This is separate from using cell formatting that affects the display of numeric data. In some cases just using cell formatting is sufficient, but in complex calculations, strict rules for rounding and precision are required to obtain consistent and correct results.

Syntax

- =ROUND(number, num_digits)
- =ROUNDUP(number, num_digits)
- =ROUNDDOWN(number, num_digits)
- =MROUND(number, multiple)
- =TRUNC(number, [num_digits])
- =INT(number)
- =CEILING(number, significance)
- =FLOOR(number, significance)
- =EVEN(number)
- =ODD(number)
- =FIXED(number, [decimals], [no_commas])

Parameters

Parameters	Details
number	number to be rounded. Could be a cell like B2 or a constant like 3.14159
num_digits	which place to be rounded 2. Omitted or 0 means round to a whole number. 1 or 2 means round to tenths or hundredths. -1 or -3 means round to tens or thousands.
multiple	The multiple to which you want to round number.
significance	The multiple to which you want to round number.
decimals	The number of digits to the right of the decimal point. *(Optional - defaults to 2)*
no_commas	A logical value that, if `TRUE`, prevents `FIXED` from including commas in the returned text. *(Optional - defaults to `FALSE`)*
[...]	Parameters in [square brackets] are optional.

Remarks

The values displayed to the user can be presented with specific formatting that does not affect the actual data values. For example, displayed data could be formatted as a percentage. See Cell Formatting for details.

Examples

Using the ROUND function

The `ROUND` function rounds a value. The number of decimal places to round to is specified by a positive value in the `num_digits` parameter. A negative value for the `num_digits` will round the integer portion of the value left of the decimal point, e.g. to the nearest 10 (for -1) or to the nearest 1000 (for -3).

Here's a table showing how round may be used.

Starting with	ROUND(b,2)	ROUND(b,1)	ROUND(b)	ROUND(b,-1)
23.10651375	23.11	23.1	23	20
19.16818924	19.17	19.2	19	20
3.92748883	3.93	3.9	4	0
31.38208409	31.38	31.4	31	30
38.34235561	38.34	38.3	38	40
7.682632495	7.68	7.7	8	10
35.39315416	35.39	35.4	35	40
20.47004449	20.47	20.5	20	20
20.49775276	20.5	20.5	20	20
2.288822497	2.29	2.3	2	0

Additional similar functions are also available to control the direction of rounding:

- `ROUNDUP` - Always rounds a number up, away from zero.
- `ROUNDDOWN` - Always rounds a number down, towards zero.

Using the TRUNC & INT functions

The excel formula `TRUNC` is used to truncate a number to a given number of decimal places,

specified by the optional `num_digits` parameter. If this parameter is defined as a negative value it will truncate the integer portion of the value. If the parameter is omitted then the default value is `0` which removes the decimal portion of the number.

The `INT` function works in a smilar way to `TRUNC` in that it removes the decimal portion of a number by rounding it down to leave the integer portion. The difference between the two is when performing the operation on a negative number; `TRUNC` will strip the decimal, however `INT` will round the value down away from zero.

For example:

```
=TRUNC(123.456,2)
=TRUNC(123.4357,-1)
=TRUNC(-123.123)
=INT(567.89)
=INT(-567.89)
```

Will display:

```
123.45
120.00
-123.00
567.00
-568.00
```

Using the MROUND function

The Excel function `MROUND` is used to round a number to an interval other than a power of 10.

These examples show `MROUND` to the nearest quarter and to the nearest even number.

Starting with	MROUND(b,0.25)	MROUND(b,2)
23.93195211	24.00	24
2.793135388	2.75	2
21.93903064	22.00	22
13.74193739	13.75	14
16.77047412	16.75	16
13.03922302	13.00	14
17.06132896	17.00	18
16.11741694	16.00	16
33.48249592	33.50	34

Starting with	MROUND(b,0.25)	MROUND(b,2)
37.29656687	37.25	38

A similar result can be obtained using the `EVEN` and `ODD` functions which round a number **up** to the nearest even or odd number respectively.

Using the CEILING & FLOOR functions

The `CEILING` function rounds a number up, away from zero, to the nearest multiple of significance. The `FLOOR` function does the same by rounds the number down towards zero.

An example of when `CEILING` could be be used is if you want to avoid using pennies in your prices and your product is priced at $4.42, use the formula `=CEILING(4.42,0.05)` to round prices up to the nearest nickel.

For example:

```
=CEILING(2.2, 1)
=FLOOR(2.2, 1)
=CEILING(-4.8, 2)
=FLOOR(-4.8, 2)
=CEILING(0.456, 0.01)
=FLOOR(0.456, 0.01)
```

Would return:

```
3
2
-4
-6
0.46
0.45
```

Using the FIXED Function

The `FIXED` function rounds a number to the specified number of decimals defined by the `decimals` parameter, formats the number in decimal format using a comma as a separator unless specified as not required defined by the parameter `no_commas`, and returns the result as text. The `decimals` parameter is optional and defaults to two decimal places. The `no_commas` parameter is also optional and defaults to `FALSE`.

For example:

```
=FIXED(1234.567, 1)
=FIXED(1234.567, -1)
=FIXED(1234.567, 1, TRUE)
=FIXED(1234.567)
```

Would return:

```
1,234.6
1,230
1234.6
1234.57
```

Read Excel rounding and precision online: https://riptutorial.com/excel/topic/1871/excel-rounding-and-precision

Chapter 11: Excel specifications and limits

Remarks

Excel specifications and limits (Excel **2016-2013**, Excel **2010**, Excel **2007**)

Worksheet and workbook specifications and limits

Feature	Maximum limit
Open workbooks	Limited by available memory and system resources
Worksheet size	1,048,576 rows by 16,384 columns
Column width	255 characters
Row height	409 points
Page breaks	1,026 horizontal and vertical
Total number of characters that a cell can contain	32,767 characters
Characters in a header or footer	255
Maximum number of line feeds per cell	253
Sheets in a workbook	Limited by available memory (default is 1 sheet)
Colors in a workbook	16 million colors (32 bit with full access to 24 bit color spectrum)
Named views in a workbook	Limited by available memory
Unique cell formats/cell styles	64,000
Fill styles	256
Line weight and styles	256

Feature	Maximum limit
Unique font types	1,024 global fonts available for use; 512 per workbook
Number formats in a workbook	Between 200 and 250, depending on the language version of Excel that you have installed
Names in a workbook	Limited by available memory
Windows in a workbook	Limited by available memory
Hyperlinks in a worksheet	66,530 hyperlinks
Panes in a window	4
Linked sheets	Limited by available memory
Scenarios	Limited by available memory; a summary report shows only the first 251 scenarios
Changing cells in a scenario	32
Adjustable cells in Solver	200
Custom functions	Limited by available memory
Zoom range	10 percent to 400 percent
Reports	Limited by available memory
Sort references	64 in a single sort; unlimited when using sequential sorts
Undo levels	100
Fields in a data form	32
Workbook parameters	255 parameters per workbook
Items displayed in filter drop-down lists	10,000
Noncontiguous	2,147,483,648 cells

Feature	Maximum limit
cells that can be selected	
Maximum limits of memory storage and file size for Data Model workbooks	32-bit environment is subject to 2 gigabytes (GB) of virtual address space, shared by Excel, the workbook, and add-ins that run in the same process. A data model’s share of the address space might run up to 500 – 700 megabytes (MB), but could be less if other data models and add-ins are loaded.. 64-bit environment imposes no hard limits on file size. Workbook size is limited only by available memory and system resources.. Excel 2016 offers the Large Address Aware functionality that lets 32-bit Excel 2016 consume twice the memory when users work on a 64-bit Windows operating system. For more information, see Large Address Aware capability change for Excel.. Note: Adding tables to the Data Model increases the file size. If you don’t plan to create complex Data Model relationships using many data sources and data types in your workbook, uncheck the Add this data to the Data Model box when you import or create tables, pivot tables, or data connections.

Calculation specifications and limits

Feature	Maximum limit
Number precision	15 digits
Smallest allowed negative number	-2.23E-308
Smallest allowed positive number	2.23E-308
Largest allowed positive number	1.00E+308
Largest allowed negative number	-1.00E+308
Largest allowed positive number via formula	1.7976931348623158e+308
Largest allowed negative number via formula	-1.7976931348623158e+308
Length of formula contents	8,192 characters
Internal length of formula	16,384 bytes
Iterations	32,767
Worksheet arrays	Limited by available memory
Selected ranges	2,048

Feature	Maximum limit
Arguments in a function	255
Nested levels of functions	64
User defined function categories	255
Number of available worksheet functions	341
Size of the operand stack	1,024
Cross-worksheet dependency	64,000 worksheets that can refer to other sheets
Cross-worksheet array formula dependency	Limited by available memory
Area dependency	Limited by available memory
Area dependency per worksheet	Limited by available memory
Dependency on a single cell	4 billion formulas that can depend on a single cell
Linked cell content length from closed workbooks	32,767
Earliest date allowed for calculation	January 1, 1900 (January 1, 1904, if 1904 date system is used)
Latest date allowed for calculation	December 31, 9999
Largest amount of time that can be entered	9999:59:59

Charting specifications and limits

Feature	Maximum limit
Charts linked to a worksheet	Limited by available memory
Worksheets referred to by a chart	255
Data series in one chart	255
Data points in a data series for 2-D charts	Limited by available memory
Data points in a data series for 3-D charts	Limited by available memory
Data points for all data series in one chart	Limited by available memory

PivotTable and PivotChart report specifications and limits

Feature	Maximum limit
PivotTable reports on a sheet	Limited by available memory
Unique items per field	1,048,576
Row or column fields in a PivotTable report	Limited by available memory
Report filters in a PivotTable report	256 (may be limited by available memory)
Value fields in a PivotTable report	256
Calculated item formulas in a PivotTable report	Limited by available memory
Report filters in a PivotChart report	256 (may be limited by available memory)
Value fields in a PivotChart report	256
Calculated item formulas in a PivotChart report	Limited by available memory
Length of the MDX name for a PivotTable item	32,767
Length for a relational PivotTable string	32,767
Items displayed in filter drop-down lists	10,000

Workbooks with the "Allow changes by more than one user..." setting enabled If the Allow changes by more than one user... setting is on for a workbook, then the following information applies. This setting is accessible by clicking the Review tab > Share Workbook. Note that in newer versions of Excel, the Share Workbook button has been hidden. To unhide it, click File > Options > Quick Access Toolbar. Open the list under Choose commands from and select All Commands. Scroll down that list until you see Share Workbook (Legacy). Select that item and click Add. Click OK. The Share Workbook button is now at the top of the Excel window and looks like this: Share Workbook

Feature	Maximum limit
Users who can open the file at the same time	256
Personal views in the workbook	Limited by available memory
Days that change history is maintained	32,767 (default is 30 days)
Workbooks that can be merged at one time	Limited by available memory

Feature	Maximum limit
Cells that can be highlighted	32,767
Colors used to identify changes made by different users when change highlighting is turned on	32 (each user is identified by a separate color; changes made by the current user are highlighted with navy blue)
Excel tables in the workbook	0 (zero). Note: A workbook that contains one or more Excel tables cannot have the Allow changes by more than one user... setting enabled.

Excel specifications and limits (Excel **2016-2013**, Excel **2010**, Excel **2007**)

Examples

Excel specifications

Excel specifications and limits (Excel **2016-2013**, Excel **2010**, Excel **2007**)

Read Excel specifications and limits online: https://riptutorial.com/excel/topic/10833/excel-specifications-and-limits

Chapter 12: Index Match for Excel

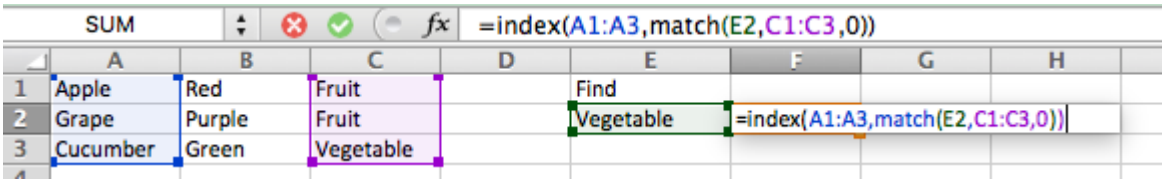
Introduction

A more versatile alternative to VLOOKUP. An Index Match packs the power of a Vlookup and Hlookup in one formula. You also do not need to know which number column/row the information is. Due to this, deleting columns/rows will not mess up the formula.

Examples

Vertical Index Match

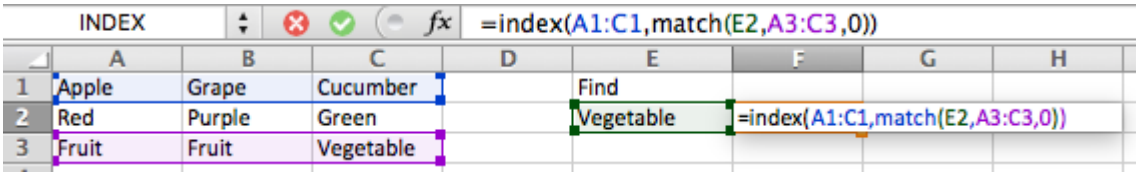
Vertical Index Match




```
=INDEX(A1:A3,MATCH(E2,C1:C3,0))
```

Horizontal Index Match

Horizontal Index Match




```
=INDEX(A1:C1,MATCH(E2,A3:C3,0))
```

Read Index Match for Excel online: https://riptutorial.com/excel/topic/9313/index-match-for-excel

Chapter 13: MATCH function

Introduction

(Optional) Every topic has a focus. Tell the readers what they will find here and let future contributors know what belongs.

Parameters

Parameter	Description
lookup_value	The value you want to match. Can be either a fixed value, cell reference or named range. Strings may not exceed 255 characters (*required*)
lookup_array	The cell reference (or named range) that you want to search, this can either be a row or a column sorted in ascending order for default type 1 matches; desceding order for -1 type matches; or any order for type 0 matches (*required*)
match_type	Controls the way the search works. Set to 0 if you only want exact matches, set to 1 if you want to match items less than or equal to your **lookup_value**, or -1 if you want to match items greater than or equal to your **lookup_value**. (*Optional* - defaults to 1)

Remarks

Purpose

Use the MATCH function to check if (and where) a value can be found in a list. Often seen as a parameter return for the row and/or column in INDEX(array, row, column) function. Allows negative row/column references allowing left or above lookups.

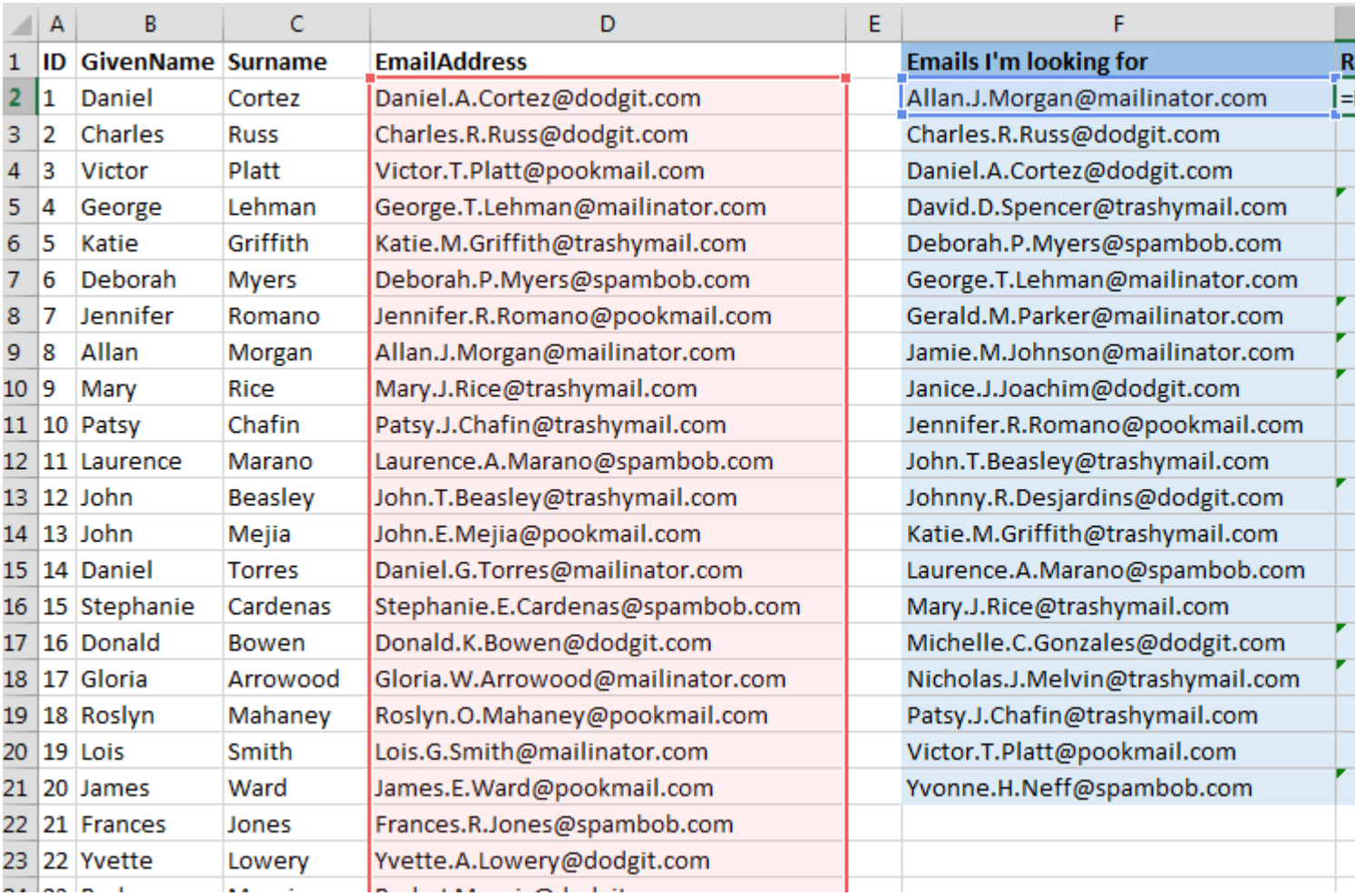
Similar functions:

- VLOOKUP - like MATCH but returns data from the table, rather than the row or column number. Can only search a table vertically and return values in or to the right of the found value.
- HLOOKUP - like MATCH but returns data from the table, rather than the row or column number. Can only search a table horizontally and return values in or below the found value.

Examples

Checking if an Email Address appears in a list of addresses

Let's say you need to check if an email address appears in a long list of email addresses.



	A	B	C	D	E	F
1	ID	GivenName	Surname	EmailAddress		Emails I'm looking for
2	1	Daniel	Cortez	Daniel.A.Cortez@dodgit.com		Allan.J.Morgan@mailinator.com
3	2	Charles	Russ	Charles.R.Russ@dodgit.com		Charles.R.Russ@dodgit.com
4	3	Victor	Platt	Victor.T.Platt@pookmail.com		Daniel.A.Cortez@dodgit.com
5	4	George	Lehman	George.T.Lehman@mailinator.com		David.D.Spencer@trashymail.com
6	5	Katie	Griffith	Katie.M.Griffith@trashymail.com		Deborah.P.Myers@spambob.com
7	6	Deborah	Myers	Deborah.P.Myers@spambob.com		George.T.Lehman@mailinator.com
8	7	Jennifer	Romano	Jennifer.R.Romano@pookmail.com		Gerald.M.Parker@mailinator.com
9	8	Allan	Morgan	Allan.J.Morgan@mailinator.com		Jamie.M.Johnson@mailinator.com
10	9	Mary	Rice	Mary.J.Rice@trashymail.com		Janice.J.Joachim@dodgit.com
11	10	Patsy	Chafin	Patsy.J.Chafin@trashymail.com		Jennifer.R.Romano@pookmail.com
12	11	Laurence	Marano	Laurence.A.Marano@spambob.com		John.T.Beasley@trashymail.com
13	12	John	Beasley	John.T.Beasley@trashymail.com		Johnny.R.Desjardins@dodgit.com
14	13	John	Mejia	John.E.Mejia@pookmail.com		Katie.M.Griffith@trashymail.com
15	14	Daniel	Torres	Daniel.G.Torres@mailinator.com		Laurence.A.Marano@spambob.com
16	15	Stephanie	Cardenas	Stephanie.E.Cardenas@spambob.com		Mary.J.Rice@trashymail.com
17	16	Donald	Bowen	Donald.K.Bowen@dodgit.com		Michelle.C.Gonzales@dodgit.com
18	17	Gloria	Arrowood	Gloria.W.Arrowood@mailinator.com		Nicholas.J.Melvin@trashymail.com
19	18	Roslyn	Mahaney	Roslyn.O.Mahaney@pookmail.com		Patsy.J.Chafin@trashymail.com
20	19	Lois	Smith	Lois.G.Smith@mailinator.com		Victor.T.Platt@pookmail.com
21	20	James	Ward	James.E.Ward@pookmail.com		Yvonne.H.Neff@spambob.com
22	21	Frances	Jones	Frances.R.Jones@spambob.com		
23	22	Yvette	Lowery	Yvette.A.Lowery@dodgit.com		

Use the MATCH function to return the row number on which the email address can be found. If there is no match, the function returns an **#N/A** error.

```
=MATCH(F2,$D$2:$D$200,0)
```

- The value you're retrieving data for is in cell **F2**
- The range you're searching is located in **$D$2:$D$200**
- You only want to know where there is an exact match (**0**)

But you may not care what row number the email address is on - you just want to know if it exists, so we can wrap the MATCH function to either return *Yes* or *Missing* instead:

```
=IFERROR(IF(MATCH(F2,$D$2:$D$200,0),"Yes"),"Missing")
```

Combining MATCH with INDEX

Say, you have a dataset consisting of names and email addresses. Now in another dataset, you just have the email address and wish to find the appropriate first name that belongs to that email address.

	A	B	C	D	E	F
1	ID	Name	Surname	Email		Email
2	1	oona	hietala	oona.hietala@example.com		alexander.baier@example.com
3	2	karlisa	almeida	karlisa.almeida@example.com		maria.fischer@example.com
4	3	alisa	lassila	alisa.lassila@example.com		morgan.francois@example.com
5	4	maria	fischer	maria.fischer@example.com		camille.guillot@example.com
6	5	halit	van geere	halit.vangeerenstein@example.com		louanne.mercier@example.com
7	6	morgan	francois	morgan.francois@example.com		alex.bell@example.com
8	7	lucas	walker	lucas.walker@example.com		oona.hietala@example.com
9	8	ross	watts	ross.watts@example.com		ross.watts@example.com
10	9	alex	bell	alex.bell@example.com		selma.johansen@example.com
11	10	laura	rasmusser	laura.rasmussen@example.com		alex.bell@example.com
12	11	gerardus	schelhaas	gerardus.schelhaas@example.com		
13	12	selma	johansen	selma.johansen@example.com		
14	13	vanessa	maier	vanessa.maier@example.com		
15	14	alexander	baier	alexander.baier@example.com		
16	15	louanne	mercier	louanne.mercier@example.com		
17	16	tiago	dupuis	tiago.dupuis@example.com		
18	17	ellen	ellis	ellen.ellis@example.com		
19	18	alicia	harris	alicia.harris@example.com		
20	19	camille	guillot	camille.guillot@example.com		
21	20	jason	cruz	jason.cruz@example.com		
22	21	lester	long	lester.long@example.com		
23	22	julia	burke	julia.burke@example.com		
24	23	arthur	nguyen	arthur.nguyen@example.com		
25	24	theo	bischoff	theo.bischoff@example.com		
26	25	meral	yılmazer	meral.yılmazer@example.com		
27						

The MATCH function returns the appropriate row the email is at, and the INDEX function selects it. Similarly, this can be done for columns as well. When a value cannot be found, it will return an **#N/A** error.

This is very similar behaviour to VLOOKUP OR HLOOKUP, but much faster and combines both previous functions in one.

- Search for cell **F2** value **(alexander.baier@example.com)**
- Within dataset **$D$2:$D$26**
- Use exact matching **(0)**
- Use the resulting relative row number **(14)** from a different dataset **$B$2:$B$26**

Read MATCH function online: https://riptutorial.com/excel/topic/4419/match-function

Chapter 14: SUMPRODUCT function

Introduction

The SUMPRODUCT function multiplies corresponding components in the given arrays, and returns the sum of those products

Syntax

- SUMPRODUCT(array1, [array2], [array3], ...)

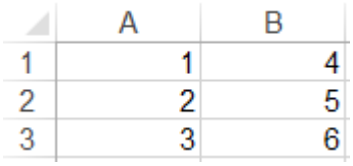
Remarks

- The array arguments must have the same dimensions. If they do not, SUMPRODUCT returns the #VALUE! error value.
- SUMPRODUCT treats array entries that are not numeric as if they were zeros.

Examples

Using SUMPRODUCT with numeric ranges

Consider the ranges `A1:A3` and `B1:B3` having the same size and only number values, as below



	A	B
1	1	4
2	2	5
3	3	6

```
=SUMPRODUCT(A1:A3,B1:B3)
```

This will loop through the ranges, taking the product of values in the same row and summing them, returning `32` in this example.

```
A1*B1 = 4
A2*B2 = 10
A3*B3 = 18
```

Using SUMPRODUCT with boolean arrays

Consider the following ranges `A1:A3` and `B1:B3` as below

	A	B
1	a	4
2	b	5
3	c	6

```
=SUMPRODUCT(--(A1:A3="c"),B1:B3)
```

This will first manipulate `(A1:A3="c")` into the following array

```
A1="c" = FALSE
A2="c" = FALSE
A3="c" = TRUE
```

Then apply the `--` operator which converts `TRUE` and `FALSE` into `1` and `0`, respectively. So the array becomes

```
--FALSE = 0
--FALSE = 0
--TRUE  = 1
```

Then the SUMPRODUCT formula completes as in the simple numeric case. Returning `6` in this example

```
0*4 = 0
0*5 = 0
1*6 = 6
```

Note: this is the equivalent of a SUMIF function

Read SUMPRODUCT function online: https://riptutorial.com/excel/topic/8096/sumproduct-function

Chapter 15: VLOOKUP

Introduction

Searches for a value in the first column of a table array and returns a value in the same row from another column in the table array.

The V in VLOOKUP stands for vertical. Use VLOOKUP instead of HLOOKUP when your comparison values are located in a column to the left of the data that you want to find.

Syntax

- VLOOKUP(lookup_value, table_array, col_index_num, range_lookup)

Parameters

Parameter	Description
lookup_value	The value you're searching for in the left column of the table. Can be either a fixed value, cell reference or named range (*required*)
table_array	The range of cells consisting of the column you want to search in on the left side with values in cells to the right that you want returned. Can be an Excel cell reference or a named range. (*required*)
col_index_num	The number of the column you want to return data from, counting from the left-most column in your table (*required*)
range_lookup	Controls the way the search works. If FALSE or 0, Excel will perform an exact search and return only where there is an exact match in the left most column. Attention to precision with rounding is very important for this search with numeric values. If TRUE or 1, Excel will perform and approximate search and return the last value it equals or exceeds. As such the first column must be sorted in ascending order for and an approximate search. **range_lookup**. (*Optional* - defaults to TRUE)

Remarks

Similar functions:

- HLOOKUP (the same as VLOOKUP but searches horizontally rather than vertically)
- MATCH (if your lookup_value has a match, returns the row number within the range)
- LOOKUP (similar to VLOOKUP and MATCH, and is provided for backward compatibility)

Common errors:

- Not setting the **range_lookup** parameter and getting the default, non-exact match behaviour
- Not fixing and absolute address range in the **table_array** - when copying a formula, the "lookup table" reference also moves

Examples

Using VLOOKUP to get a person's surname from their Employee ID

Vlookup finds some value in the *leftmost* column of a range and returns a value some number of columns to the right and in the same row.

Let's say you want to find the surname of Employee ID 2 from this table:

	A	B	C
1	**EmpID**	**First name**	**Surname**
2	1	Joe	Bloggs
3	2	Linda	Williams
4	3	John	Smith
5			

```
=VLOOKUP(2,$A$2:$C$4,3,0)
```

- The value you're retrieving data for is **2**
- The table you're searching is located in the range **$A$2:$C$4**
- The column you want to return data from is the **3**rd column from the left
- You only want to return results where there is an exact match (**0**)

Note that if there is no exact match on the employee ID, the `VLOOKUP` will return `#N/A`.

Using VLOOKUP to work out bonus percent (example with the "default" behaviour)

In most cases, the **range_lookup** is used as FALSE (an exact match). The default for this parameter is TRUE - it is less commonly used in this form, but this example shows one usecase. A supermarket gives a bonus based on the customers monthly spend.

	A	B
1	**Monthly spend (on or over)**	**Bonus**
2	0	0%
3	250	1%
4	500	2.5%
5	750	4%
6	1000	5%

If the customer spends 250 EUR or more in a month, they get 1% bonus; 500 EUR or more gives

2.5% etc. Of course, the customer won't always spend exactly one of the values in the table!

```
=VLOOKUP(261,$A$2:$B$6,2,TRUE)
```

In this mode, the VLOOKUP will find the first value in column `A` (following bottom-up steps) **less than or equal to** the value 261 -> i.e. we will get the `1%` value returned. **For this non-exact match, the table must be sorted in ascending order of the first column.**

- The value you're retrieving data for is **261**
- The table you're searching is located in the range **$A$2:$B$6**
- The column you want to return data from is the **2**nd column from the left
- You only want to return results where there is an non-exact match (**TRUE**) *we could leave off this TRUE as it is the default*

Using VLOOKUP with approximate matching.

When the **range_lookup** parameter is either omitted, TRUE, or 1, VLOOKUP will find an approximate match. By "approximate", we mean that VLOOKUP will match on the smallest value that's larger than your **lookup_value**. Note that your **table_array** *must* be sorted in ascending order by lookup values. Results will be unpredictable if your values are not sorted.

	A	B	C	D	E	F
1	**ID**	**Value**		**Search ID**	**Formula**	**Result**
2	1	B		4	=VLOOKUP(D2,A$2:B$7,2,TRUE)	C
3	3	C		4.5	=VLOOKUP(D3,A$2:B$7,2,TRUE)	C
4	5	D		5	=VLOOKUP(D4,A$2:B$7,2,TRUE)	D
5	7	E		5.5	=VLOOKUP(D5,A$2:B$7,2,TRUE)	D
6	9	F		6	=VLOOKUP(D6,A$2:B$7,2,TRUE)	D
7	11	G		6.5	=VLOOKUP(D7,A$2:B$7,2,TRUE)	D

Using VLOOKUP with exact matching

The core idea of `VLOOKUP` is to look up information in a spreadsheet table and place it in another.

For example, suppose this is the table in Sheet1:

```
John        12/25/1990
Jane        1/1/2000
```

In Sheet2, place `John`, `Andy`, and `Jane` in A1, A2, and A3.

In B1, to the right of `John`, I placed:

```
=VLOOKUP(A1,Sheet1!$A$1:$B$4,2,FALSE)
```

Here's a brief explanation of the parameters given to VLOOKUP. The A1 means I'm seeking `John` in A1 of Sheet2. The

```
Sheet1!$A$1:$B$4
```

tells the function to look at Sheet1, columns A through B (and rows 1 through 4). The dollar signs are necessary to tell Excel to use absolute (rather than relative) references. (Relative references would make the whole thing shift in undesirable ways when copying the formula down.

The `2` means to return the second column, which is the date.

The `FALSE` means that you are requiring an exact match.

I then copied down B1 to B2 and B3. (The easiest way to do this would be to click on B1 to highlight it. Then hold the shift key down and press the down arrow twice. Now B1, B2, and B3 are highlighted. Then press Ctrl-D to Fill Down the formula. If done correctly, one should have the same formula in B3 as in B1. *If the formulas for the lookup table are changing*, for instance from `Sheet1!A1:B3` to `Sheet1:A3:B5`, then you should use absolute references (with dollar signs) to prevent the change.)

Here are the results:

```
John    12/25/1990
Andy    #N/A
Jane    1/1/2000
```

It found John and Jane, and returned their birthdates. It did not find Andy, and so it displays an `#N/A`.

Read VLOOKUP online: https://riptutorial.com/excel/topic/4327/vlookup

Credits

S. No	Chapters	Contributors
1	Getting started with excel	4444, Andre Terra, Community, dot.Py, Mike, T.Furholzer, Washington Guedes
2	An easy process to convert Monthly to Quarterly data in Excel	Klatuu
3	Array formulas	Alexis Olson
4	Base Conversion	SeanC
5	Cell Formatting	Mark Stewart
6	Counting unique cells	Ulli Schmid
7	Create Connection to other Excel Files	petergensler
8	DATEDIF function	Andi Mohr, CallumDA, dav, Islam Tawfik
9	Excel best practice	Alon Eitan, Mats Lind, Mike
10	Excel rounding and precision	Mark Stewart, mike7mike, nekomatic, rajah9
11	Excel specifications and limits	paul bica
12	Index Match for Excel	Mukul215, Washington Guedes
13	MATCH function	Andi Mohr, japborst, Mark Fitzgerald
14	SUMPRODUCT function	CallumDA
15	VLOOKUP	Andi Mohr, Captain, EBH, Forward Ed, japborst, picobit, rajah9, Shrikant, Washington Guedes