SPREADSHEET PLANET

100 Essential Excel Formulas
Quick Reference



Put together by **Steve Scott** · Part of the Excel Speed Pack · **SpreadsheetPlanet.com**

THE SAMPLE SHEET USED IN EVERY LOOKUP AND COUNTING EXAMPLE

Build this once and you can test any formula on these pages as you read.

	A · Item	B · Region	C · Units
2	Apple	West	120
3	Banana	East	80
4	Cherry	West	150
5	Grape	East	90
6	Mango	West	42
7	Melon	East	200
8	Peach	West	60
9	Pear	East	110
10	Plum	West	75

Totals to know: **SUM = 927**, **AVERAGE = 103**, 9 rows, 5 West and 4 East.

MODERN REPLACEMENTS, USE THESE INSTEAD

- **XLOOKUP over VLOOKUP and INDEX/MATCH.** It looks left, returns whole rows, takes an if-not-found value, and survives inserted columns. `=XLOOKUP(key, lookup_col, return_col, "Not found")`
- **FILTER over nested IFs and manual filtering.** One formula returns every matching row and updates itself.
- **TEXTJOIN over CONCATENATE.** It takes a separator and skips blanks in one step.
- **TEXTBEFORE and TEXTAFTER over LEFT plus FIND.** Same result, no character arithmetic.
- **LET and LAMBDA when a formula gets long.** `LET` names a value once and reuses it, so it reads clearly and calculates faster. `LAMBDA` turns a working formula into your own named function in Name Manager.

MATH & TOTALS 12

SUM

Adds numbers or ranges.

```
=SUM(number1, [number2], ...)
```

`=SUM(C2:C10)` → **927**

SUMIF

Adds only the values that match one condition.

```
=SUMIF(range, criteria, [sum_range])
```

`=SUMIF(B2:B10,"West",C2:C10)` → **447**

SUMIFS

Adds values that pass every condition you list.

```
=SUMIFS(sum_range, range1, crit1, ...)
```

`=SUMIFS(C2:C10,B2:B10,"West",C2:C10,">100")` → **270**

SUMPRODUCT

Multiplies arrays pairwise, then adds the results.

```
=SUMPRODUCT(array1, [array2], ...)
```

`=SUMPRODUCT((B2:B10="West")*C2:C10)` → **447**

SUBTOTAL

Aggregates a range and ignores rows hidden by a filter.

```
=SUBTOTAL(function_num, ref1, ...)
```

`=SUBTOTAL(109,C2:C10) with East filtered out` → **447**

ROUND

Rounds to a set number of decimals.

```
=ROUND(number, num_digits)
```

`=ROUND(103.456,1)` → **103.5**

ROUNDUP

Always rounds away from zero.

```
=ROUNDUP(number, num_digits)
```

`=ROUNDUP(2.11,1)` → **2.2**

ROUNDDOWN

Always rounds toward zero.

```
=ROUNDDOWN(number, num_digits)
```

`=ROUNDDOWN(2.19,1)` → **2.1**

INT

Rounds down to the nearest whole number.

```
=INT(number)
```

`=INT(103.9)` → **103**

MOD

Returns the remainder after division. Handy for every-nth-row rules.

```
=MOD(number, divisor)
```

`=MOD(927,100)` → **27**

ABS

Strips the minus sign.

```
=ABS(number)
```

`=ABS(-42)` → **42**

SQRT

Square root of a positive number.

```
=SQRT(number)
```

`=SQRT(144)` → **12**

LOGICAL 8

IF

Returns one value when a test is true, another when it is false.

```
=IF(test, value_if_true, value_if_false)
```

`=IF(C6>100,"High","Low")` → **Low** (C6 is 42)

IFS

Checks conditions in order and stops at the first true one. No nesting.

```
=IFS(test1, val1, test2, val2, ...)
```

`=IFS(C7>=150,"A",C7>=100,"B",C7>=50,"C")` → **A** (C7 is 200)

AND

TRUE only when every test passes.

```
=AND(logical1, [logical2], ...)
```

`=AND(C6>40,C6<50)` → **TRUE**

OR

TRUE when at least one test passes.

```
=OR(logical1, [logical2], ...)
```

`=OR(C6>100,B6="West")` → **TRUE**

NOT

Flips TRUE to FALSE and back.

```
=NOT(logical)
```

`=NOT(C6>100)` → **TRUE**

XOR

TRUE when an odd number of tests pass. Exactly one, in the two-test case.

```
=XOR(logical1, [logical2], ...)
```

`=XOR(TRUE,TRUE)` → **FALSE**

SWITCH

Matches one value against a list of options.

```
=SWITCH(expr, val1, res1, ..., [default])
```

`=SWITCH(2,1,"Low",2,"Mid",3,"High","Other")` → **Mid**

LET

Names a value inside a formula so you calculate it once and reuse it.

```
=LET(name1, value1, ..., calculation)
```

`=LET(u,C2:C10,ROUND(AVERAGE(u),0))` → **103**

LOOKUP 11

XLOOKUP

Finds a value in one range and returns the matching item from another. Start here.

```
=XLOOKUP(lookup, lookup_array, return_array, [if_not_found], [match_mode])
```

`=XLOOKUP("Mango",A2:A10,C2:C10)` → **42**

XLOOKUP with a fallback

The fourth argument replaces #N/A with your own text.

```
=XLOOKUP(lookup, lookup_array, return_array, if_not_found)
```

`=XLOOKUP("Kiwi",A2:A10,C2:C10,"Not stocked")` → **Not stocked**

VLOOKUP

Older column-number lookup. Only searches rightward from column 1.

```
=VLOOKUP(lookup, table, col_index, [range_lookup])
```

`=VLOOKUP("Mango",A2:C10,3,FALSE)` → **42**

HLOOKUP

Same idea across a row instead of down a column.

```
=HLOOKUP(lookup, table, row_index, [range_lookup])
```

Headers Q1:Q4 in E1:H1, values 400, 650, 890, 720 in E2:H2.
`=HLOOKUP("Q3",E1:H2,2,FALSE)` → **890**

INDEX

Returns the item at a given position in a range.

```
=INDEX(array, row_num, [column_num])
```

`=INDEX(C2:C10,5)` → **42**

MATCH

Returns the position of a value, not the value itself.

```
=MATCH(lookup, lookup_array, [match_type])
```

`=MATCH("Mango",A2:A10,0)` → **5**

INDEX + MATCH

The classic pairing. MATCH finds the row, INDEX pulls the value.

```
=INDEX(return_range, MATCH(lookup, lookup_range, 0))
```

`=INDEX(C2:C10,MATCH("Mango",A2:A10,0))` → **42**

CHOOSE

Picks the nth item from a list you type in the formula.

```
=CHOOSE(index_num, value1, value2, ...)
```

`=CHOOSE(2,"Red","Green","Blue")` → **Green**

INDIRECT

Turns text into a real cell reference. Recalculates on every edit, so use sparingly.

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

`=INDIRECT("C"&6)` → **42**

OFFSET

Returns a cell or range a set distance from a starting point.

```
=OFFSET(reference, rows, cols, [height], [width])
```

`=OFFSET(A2,4,2)` → **42** (lands on C6)

TRANSPOSE

Flips a vertical range to horizontal, or the reverse.

```
=TRANSPOSE(array)
```

`=TRANSPOSE(A2:A4)` → **Apple | Banana | Cherry** across one row

TEXT 16

LEFT

Takes characters from the start of the text.

```
=LEFT(text, [num_chars])
```

`=LEFT("Spreadsheet",6)` → **Spread**

RIGHT

Takes characters from the end.

```
=RIGHT(text, [num_chars])
```

`=RIGHT("Spreadsheet",5)` → **sheet**

MID

Takes characters from the middle, by start position and length.

```
=MID(text, start_num, num_chars)
```

`=MID("Spreadsheet",7,5)` → **sheet**

LEN

Counts characters, spaces included.

```
=LEN(text)
```

`=LEN("Spreadsheet")` → **11**

TRIM

Removes leading and trailing spaces, and collapses runs of spaces inside.

```
=TRIM(text)
```

`=TRIM("  net  sales ")` → **net sales**

UPPER

Converts text to capitals.

```
=UPPER(text)
```

`=UPPER("west")` → **WEST**

LOWER

Converts text to lower case.

```
=LOWER(text)
```

`=LOWER("WEST")` → **west**

PROPER — Capitalises the first letter of each word.

```
=PROPER(text)
```

`=PROPER("steve scott")` → **Steve Scott**

TEXT — Formats a number as text using a number-format code.

```
=TEXT(value, format_text)
```

`=TEXT(0.4567,"0.0%")` → **45.7%**

TEXTJOIN — Joins items with a separator and can skip the blanks.

```
=TEXTJOIN(delimiter, ignore_empty, text1, ...)
```

`=TEXTJOIN(", ",TRUE,"Apple","","Cherry")` → **Apple, Cherry**

SUBSTITUTE — Swaps one piece of text for another, by content.

```
=SUBSTITUTE(text, old, new, [instance])
```

`=SUBSTITUTE("2025-Q1","2025","2026")` → **2026-Q1**

FIND — Position of one string inside another. Case sensitive.

```
=FIND(find_text, within_text, [start_num])
```

`=FIND("@","sam@site.com")` → **4**

SEARCH — Same as FIND but ignores case and accepts wildcards.

```
=SEARCH(find_text, within_text, [start_num])
```

`=SEARCH("SITE","sam@site.com")` → **5**

TEXTBEFORE — Everything before a delimiter, no character counting.

```
=TEXTBEFORE(text, delimiter, [instance])
```

`=TEXTBEFORE("sam@site.com","@")` → **sam**

TEXTAFTER — Everything after a delimiter.

```
=TEXTAFTER(text, delimiter, [instance])
```

`=TEXTAFTER("sam@site.com","@")` → **site.com**

TEXTSPLIT — Splits text into separate cells at each delimiter.

```
=TEXTSPLIT(text, col_delimiter, [row_delimiter])
```

`=TEXTSPLIT("Apple,West,120",",")` → **Apple | West | 120** in three cells

TEXTBEFORE, TEXTAFTER and TEXTSPLIT need Microsoft 365 or Excel 2024. On older versions, use LEFT, RIGHT and MID with FIND.

DATE & TIME 12

TODAY — Today's date. Updates every time the sheet recalculates.

```
=TODAY()
```

`=TODAY()` → **15-Aug-2026**

NOW — Current date and time.

```
=NOW()
```

`=NOW()` → **15-Aug-2026 09:41**

DATE — Builds a real date from separate year, month and day numbers.

```
=DATE(year, month, day)
```

`=DATE(2026,8,15)` → **15-Aug-2026**

DAY

Pulls the day number out of a date.

```
=DAY(serial_number)
```

`=DAY(DATE(2026,8,15))` → **15**

MONTH

Pulls the month number, 1 to 12.

```
=MONTH(serial_number)
```

`=MONTH(DATE(2026,8,15))` → **8**

YEAR

Pulls the four-digit year.

```
=YEAR(serial_number)
```

`=YEAR(DATE(2026,8,15))` → **2026**

EDATE

Same day, a set number of months later or earlier.

```
=EDATE(start_date, months)
```

`=EDATE(DATE(2026,8,15),3)` → **15-Nov-2026**

EOMONTH

Last day of the month, offset by months you choose.

```
=EOMONTH(start_date, months)
```

`=EOMONTH(DATE(2026,8,15),0)` → **31-Aug-2026**

WEEKDAY

Which day of the week a date falls on, as a number.

```
=WEEKDAY(serial_number, [return_type])
```

`=WEEKDAY(DATE(2026,8,15),2)` → **6** (Saturday, Monday = 1)

NETWORKDAYS

Counts working days between two dates, weekends excluded.

```
=NETWORKDAYS(start, end, [holidays])
```

`=NETWORKDAYS(DATE(2026,8,3),DATE(2026,8,14))` → **10**

WORKDAY

The date a set number of working days ahead.

```
=WORKDAY(start, days, [holidays])
```

`=WORKDAY(DATE(2026,8,14),1)` → **17-Aug-2026** (skips the weekend)

DATEDIF

Whole years, months or days between two dates. Good for ages and tenure.

```
=DATEDIF(start, end, unit)
```

`=DATEDIF(DATE(2000,3,1),DATE(2026,8,15),"Y")` → **26**

DATEDIF has no formula tooltip and no autocomplete. Type it in full. Units are "Y", "M", "D", "YM" and "MD".

ERROR HANDLING 8

IFERROR

Catches any error and shows your value instead.

```
=IFERROR(value, value_if_error)
```

`=IFERROR(1/0,"Check input")` → **Check input**

IFNA

Catches only #N/A, so real errors still surface.

```
=IFNA(value, value_if_na)
```

`=IFNA(XLOOKUP("Kiwi",A2:A10,C2:C10),"Not stocked")` → **Not stocked**

ISERROR

TRUE when the value is any error type.

```
=ISERROR(value)
```

`=ISERROR(1/0)` → **TRUE**

ISNA

TRUE only for #N/A.

```
=ISNA(value)
```

```
=ISNA(NA()) → TRUE
```

ISBLANK

TRUE when the cell is genuinely empty.

```
=ISBLANK(value)
```

```
=ISBLANK(A11) → TRUE (A11 is empty)
```

ISNUMBER

TRUE when the cell holds a number, not text that looks like one.

```
=ISNUMBER(value)
```

```
=ISNUMBER(C6) → TRUE
```

ISTEXT

TRUE when the value is text.

```
=ISTEXT(value)
```

```
=ISTEXT(A6) → TRUE (A6 is Mango)
```

NA

Puts a deliberate #N/A in a cell, so charts skip it instead of plotting zero.

```
=NA()
```

```
=NA() → #N/A
```

COUNTING & CONDITIONAL AGGREGATION 12

COUNT

Counts cells holding numbers.

```
=COUNT(value1, [value2], ...)
```

```
=COUNT(C2:C10) → 9
```

COUNTA

Counts every cell that is not empty, text included.

```
=COUNTA(value1, [value2], ...)
```

```
=COUNTA(A2:A10) → 9
```

COUNTBLANK

Counts the empty cells in a range.

```
=COUNTBLANK(range)
```

```
=COUNTBLANK(A2:A12) → 2 (A11 and A12 are empty)
```

COUNTIF

Counts the cells that match one condition.

```
=COUNTIF(range, criteria)
```

```
=COUNTIF(B2:B10,"West") → 5
```

COUNTIFS

Counts rows that pass every condition.

```
=COUNTIFS(range1, crit1, range2, crit2, ...)
```

```
=COUNTIFS(B2:B10,"East",C2:C10,">100") → 2
```

AVERAGE

Mean of the numbers, blanks ignored.

```
=AVERAGE(number1, [number2], ...)
```

```
=AVERAGE(C2:C10) → 103
```

AVERAGEIF

Mean of the values that match one condition.

```
=AVERAGEIF(range, criteria, [average_range])
```

```
=AVERAGEIF(C2:C10,">100") → 145 (120, 150, 200, 110)
```

AVERAGEIFS — Mean of the values that pass every condition.

```
=AVERAGEIFS(avg_range, range1, crit1, ...)
```

`=AVERAGEIFS(C2:C10,B2:B10,"East")` → **120**

MAX — Largest number in the range.

```
=MAX(number1, [number2], ...)
```

`=MAX(C2:C10)` → **200**

MIN — Smallest number in the range.

```
=MIN(number1, [number2], ...)
```

`=MIN(C2:C10)` → **42**

MAXIFS — Largest value among the rows that match.

```
=MAXIFS(max_range, range1, crit1, ...)
```

`=MAXIFS(C2:C10,B2:B10,"West")` → **150**

MEDIAN — Middle value once sorted. Less skewed by outliers than the mean.

```
=MEDIAN(number1, [number2], ...)
```

`=MEDIAN(C2:C10)` → **90**

FINANCIAL BASICS 8

PMT — Loan payment per period, principal and interest.

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

`=PMT(0.06/12,360,-300000)` → **1,798.65** a month

FV — What regular savings grow to.

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

`=FV(0.05,10,-1000)` → **12,577.89**

PV — What a stream of future payments is worth today.

```
=PV(rate, nper, pmt, [fv], [type])
```

`=PV(0.05,10,-1000)` → **7,721.73**

NPV — Present value of future cash flows. Add the day-one cost outside the function.

```
=NPV(rate, value1, [value2], ...)
```

`=NPV(0.1,3000,4200,6800)-10000` → **1,307.36**

IRR — The return rate that makes a project break even.

```
=IRR(values, [guess])
```

Cash flows -10000, 3000, 4200, 6800 in A1:A4.
`=IRR(A1:A4)` → **16.3%**

RATE — The interest rate implied by a known payment schedule.

```
=RATE(nper, pmt, pv, [fv], [type], [guess])
```

`=RATE(360,-1798.65,300000)` → **0.5%** a month, 6% a year

NPER — How many periods it takes to clear a balance.

```
=NPER(rate, pmt, pv, [fv], [type])
```

`=NPER(0.05,-1000,8000)` → **10.47** years

SLN

Straight-line depreciation for one period.

```
=SLN(cost, salvage, life)
```

`=SLN(30000,7500,10)` → **2,250** a year

Money you pay out is entered as a negative number. That is why the loan and savings examples use a minus sign.

DYNAMIC ARRAYS 13

FILTER

Returns every row that meets a condition. One formula, live results.

```
=FILTER(array, include, [if_empty])
```

`=FILTER(A2:A10,C2:C10>100)` → **Apple, Cherry, Melon, Pear**

FILTER with a fallback

The third argument covers the no-matches case.

```
=FILTER(array, include, if_empty)
```

`=FILTER(A2:A10,C2:C10>500,"None")` → **None**

SORT

Sorts a range without touching the original.

```
=SORT(array, [sort_index], [sort_order], [by_col])
```

`=SORT(C2:C10)` → **42, 60, 75, 80, 90, 110, 120, 150, 200**

SORTBY

Sorts one range using the order of another.

```
=SORTBY(array, by_array1, [order1], ...)
```

`=SORTBY(A2:A10,C2:C10,-1)` → **Melon, Cherry, Apple, Pear, ...**

UNIQUE

The distinct values in a list, duplicates dropped.

```
=UNIQUE(array, [by_col], [exactly_once])
```

`=UNIQUE(B2:B10)` → **West, East**

SEQUENCE

Generates a numbered list or grid.

```
=SEQUENCE(rows, [cols], [start], [step])
```

`=SEQUENCE(4,1,10,5)` → **10, 15, 20, 25**

XMATCH

Position of a value, with better matching than MATCH and no third argument needed.

```
=XMATCH(lookup, array, [match_mode], [search_mode])
```

`=XMATCH("Mango",A2:A10)` → **5**

RANDARRAY

A block of random numbers, whole or decimal.

```
=RANDARRAY([rows], [cols], [min], [max], [integer])
```

`=RANDARRAY(3,1,1,10,TRUE)` → **3 whole numbers from 1 to 10**

TAKE

Keeps the first or last n rows or columns.

```
=TAKE(array, rows, [columns])
```

`=TAKE(A2:C10,3)` → **the Apple, Banana and Cherry rows**

DROP

Removes rows or columns from an array. Negative numbers trim from the end.

```
=DROP(array, rows, [columns])
```

`=DROP(A2:C10,0,-1)` → **Item and Region only**

CHOOSECOLS

Pulls the columns you name, in the order you name them.

```
=CHOOSECOLS(array, col_num1, ...)
```

`=CHOOSECOLS(A2:C10,1,3)` → **Item and Units**

VSTACK Stacks ranges on top of each other into one list.

```
=VSTACK(array1, [array2], ...)
```

`=VSTACK(A2:A4,A8:A10)` → **Apple, Banana, Cherry, Peach, Pear, Plum**

HSTACK Places ranges side by side into one table.

```
=HSTACK(array1, [array2], ...)
```

`=HSTACK(A2:A10,C2:C10)` → **Item and Units as two columns**

THREE THINGS TO KNOW ABOUT SPILLED RESULTS

- Type the formula in one cell only. Excel fills the rest and outlines the range in blue.
- `#SPILL!` means something is blocking the output. Clear the cells below or to the right.
- Point at a whole spilled range with the hash operator: `=SUM(E2#)` totals whatever E2 spills, however long it grows.



Put together by **Steve Scott**
SpreadsheetPlanet.com

Get one Excel tip every weekday → spreadsheetplanet.com/newsletter

The Excel Speed Pack