Excel Guide for Accountants

The Complete Collection

33	100+	Pro
Cheatsheets	Functions	Tips

What's Inside

- Excel Fundamentals & Productivity
- Lookup & Data Functions
- Financial & Advanced Functions

Contents

Complete Collection • 33 Cheatsheets

01 10 Essential Excel Functions
02 Excel Shortcuts Cheatsheet
03 Common Excel Errors
04 12 Excel Mistakes to Avoid
05 5 Excel Features You Should Know
06 XLOOKUP Guide
07 VLOOKUP vs XLOOKUP
08 SUMIFS Cheatsheet
09 IF Function Cheatsheet
10 INDEX & MATCH Guide
11 COUNTIFS Cheatsheet
12 Date Functions Guide
13 FILTER Function Guide
14 XMATCH Cheatsheet
15 LAMBDA Function Guide
16 LET Function Guide
17 Text Functions Guide
18 Error Handling Guide
19 Lookup Functions Master
20 Conditional Functions
21 Dynamic Array Functions
22 Financial Functions
23 Statistical Functions
24 Rounding & Precision
25 Variance & Budget Analysis
26 Data Validation Essentials
27 Pivot Table Functions
28 Audit & Review Functions
29 Cash Flow & Investment
30 Depreciation Functions
31 Loan & Amortization
32 Aggregation Functions
33 Named Ranges & Tables

10 Essential Excel Functions

Master these formulas to streamline your accounting workflows

1 SUMIFS

```
=SUMIFS(sum_range, criteria_range1, criteria1, ...)
```

Adds values that meet multiple criteria. Perfect for totaling by client, region, or period.

2 XLOOKUP

```
=XLOOKUP(lookup_value, lookup_array, return_array)
```

Searches and returns matching results. Essential for reference tables and chart of accounts.

3 IFERROR

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

Returns custom value on errors. Clean up reports and avoid #DIV/0! displays.

4 EOMONTH

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

Returns the last day of a month. Critical for due dates and period-end calculations.

5 PMT

```
=PMT(rate, nper, pv)
```

Calculates loan payments. Essential for amortization and lease calculations.

6 INDEX & MATCH

```
=INDEX(array, MATCH(lookup_value, lookup_array, 0))
```

Two-dimensional lookup anywhere in a table. More flexible than VLOOKUP.

7 XNPV

```
=XNPV(rate, values, dates)
```

NPV for irregular cash flows. Use for investments with non-periodic payments.

8 COUNTIFS

```
=COUNTIFS(criteria_range1, criteria1, ...)
```

Counts cells meeting multiple criteria. Great for aging analysis and sampling.

9 TEXT

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

Formats numbers as text. Perfect for consistent date displays and labels.

10 ROUND

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

Rounds to specified decimals. Essential for avoiding rounding errors.

Excel Shortcuts Cheatsheet

Speed up your workflow with these essential keyboard shortcuts

General

Ctrl + S	Save Workbook
Ctrl + C	Copy
Ctrl + V	Paste
Ctrl + Z	Undo
Ctrl + Y	Redo
Ctrl + N	New Workbook
Ctrl + O	Open Workbook
Ctrl + P	Print

Navigation

Ctrl + →	Jump to Edge of Data
Ctrl + Home	Go to First Cell
Ctrl + End	Go to Last Cell
Ctrl + PgDn	Next Worksheet
Ctrl + PgUp	Previous Worksheet
Ctrl + Tab	Next Workbook

Formatting

Ctrl + Shift + $	Currency Format
Ctrl + Shift + %	Percentage Format
Ctrl + Shift + #	Date Format
Ctrl + Shift + !	Number Format
Ctrl + ;	Insert Current Date
Ctrl + :	Insert Current Time
Ctrl + +	Insert Rows/Columns
Ctrl + -	Delete Rows/Columns

Data & Formulas

Ctrl + T	Create Table
Ctrl + E	Flash Fill
Alt + =	AutoSum
F4	Toggle Absolute Reference
Ctrl + D	Fill Down
Ctrl + R	Fill Right

Common Excel Errors & Solutions

Quick reference guide to diagnose and fix formula errors

#DIV/0!

Division by zero or empty cell

Use IFERROR to handle: =IFERROR(A1/B1, 0)

#VALUE!

Wrong type of argument or operand

Check data types. Use VALUE() or TEXT() to convert.

#NAME?

Excel doesn't recognize text in formula

Check for typos in function names. Enclose text in quotes.

#REF!

Invalid cell reference

Update references after deleting rows/columns. Use Undo.

#N/A

Value not found in lookup

Verify lookup value exists. Check for extra spaces.

#NUM!

Invalid numeric value in formula

Check numeric arguments are within valid ranges.

#NULL!

Incorrect range operator

Use comma to separate ranges: =SUM(A1:A10, B1:B10)

#SPILL!

Dynamic array blocked by data

Clear cells where the formula needs to spill results.

#CALC!

Calculation error in array formula

Check array formula logic and source data.

######

Column too narrow to display

Widen the column or reduce font size.

12 Excel Mistakes to Avoid

Common pitfalls that slow you down and cause errors

1 Wrong Data Formats

Using text instead of numbers or dates causes calculation errors.

2 Skipping Error Handling

Not using IFERROR leads to ugly #ERROR displays in reports.

3 Ignoring Data Validation

Without input restrictions, users can enter invalid data.

4 No Conditional Formatting

Missing visual cues makes it hard to spot issues quickly.

5 Avoiding Keyboard Shortcuts

Mouse-only work is slow. Shortcuts save hours per week.

6 Skipping Named Ranges

Cell references make formulas hard to read and maintain.

7 Infrequent Saving

Losing work to crashes is preventable with AutoSave.

8 Reference Type Confusion

Mixing absolute (A1) and relative (A1) incorrectly.

9 Dirty Data

Duplicates, blanks, and inconsistencies corrupt analysis.

10 Overcomplicated Formulas

Long nested formulas are hard to debug. Use helper columns.

11 Ignoring Advanced Features

Power Query and Pivot Tables automate hours of work.

12 Too Much Manual Entry

Typing is slow and error-prone. Import and use formulas.

5 Excel Features You Should Know

Power features that transform how you work with data

1 Data Tables

What-If Analysis

See how changing one or two variables affects your results. Perfect for sensitivity analysis.

1. Set up calculation with input cells
2. Data → What-If Analysis → Data Table
3. Link inputs and see all scenarios

2 Pivot Tables

Summarise & Analyse

Transform raw data into meaningful summaries. Group by any dimension instantly.

1. Select data with headers
2. Insert → PivotTable
3. Drag fields to Rows, Columns, Values

3 Data Validation

Input Control

Restrict what users can enter. Create dropdowns and prevent errors at source.

1. Select cells → Data → Data Validation
2. Choose type (list, number, date)
3. Add error alert

4 Power Query

Data Transformation

Import, clean, and transform data. Create repeatable workflows that refresh.

1. Data → Get Data to connect
2. Transform and combine sources
3. Load to worksheet

5 Group & Outline

Structure Control

Create collapsible sections. Show summaries or detail as needed.

1. Select rows/columns
2. Data → Group
3. Use +/- to expand/collapse

XLOOKUP Guide

The modern replacement for VLOOKUP and HLOOKUP

Syntax

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

Arguments

`lookup_value`	The value you want to find
`lookup_array`	The range containing possible lookup values
`return_array`	The range containing values to return
`[if_not_found]`	Optional. Value to return if not found
`[match_mode]`	Optional. 0=exact, -1=less than, 1=greater than
`[search_mode]`	Optional. 1=first to last, -1=last to first

Examples

1. Basic XLOOKUP

```
=XLOOKUP("Apple", A2:A10, B2:B10)
```

Searches for "Apple" and returns the corresponding value.

2. Handling Not Found

```
=XLOOKUP(lookup, A2:A10, B2:B10, "Not Found")
```

Returns "Not Found" if the value doesn't exist.

3. Reverse Lookup

```
=XLOOKUP(lookup, B2:B10, A2:A10)
```

Searches in column B, returns from column A.

4. With Error Handling

```
=IFERROR(XLOOKUP(lookup, A2:A10, B2:B10), 0)
```

Wraps with IFERROR for additional control.

Pro Tips

- ✓ Use wildcards (* and ?) for partial matches
- ✓ Performs exact match by default

VLOOKUP vs XLOOKUP

Understanding when to use each lookup function

ASPECT	VLOOKUP	XLOOKUP
Syntax	VLOOKUP(value, table, col_index, [range_lookup])	XLOOKUP(value, lookup_array, return_array, [if_not_found])
Direction	Left to right only	Any direction
Default Match	Approximate (requires TRUE/FALSE)	Exact match
Column Index	Required (breaks if columns change)	Not needed
Not Found Handling	Returns #N/A error	Custom value option
Performance	Slower with large data	Faster and more efficient
Compatibility	All Excel versions	Excel 365 / 2021+

Same Task, Different Approach

Find the name of employee with ID "1003"

VLOOKUP:

```
=VLOOKUP(1003, B2:C5, 2, FALSE)
```

Searches column B, returns from 2nd column of range, exact match

XLOOKUP:

```
=XLOOKUP(1003, B2:B5, C2:C5)
```

Searches column B, returns from column C, exact match by default

When to Use Each

VLOOKUP When working with legacy spreadsheets or sharing with users on older Excel versions

XLOOKUP For new work—it's more flexible, handles errors better, and doesn't break when columns shift

SUMIFS Cheatsheet

Sum values based on multiple criteria

Syntax

```
=SUMIFS(sum_range, criteria_range1, criteria1, [criteria_range2, criteria2], ...)
```

Arguments

`sum_range`	The range of cells to sum
`criteria_range1`	The range to apply the first condition
`criteria1`	The condition to be met in criteria_range1
`[criteria_range2, ...]`	Additional ranges and conditions

Examples

Basic SUMIFS

```
=SUMIFS(C2:C10, B2:B10, "Product A")
```

Sum of Sales where Product is "Product A"

Multiple Conditions

```
=SUMIFS(D2:D10, B2:B10, "North", C2:C10, "Q1")
```

Sum Revenue where Region is "North" AND Quarter is "Q1"

Date Range

```
=SUMIFS(C2:C10, A2:A10, ">=1/1/2024", A2:A10, "<=31/1/2024")
```

Sum values for dates within January 2024

With Wildcards

```
=SUMIFS(C2:C10, B2:B10, "Product*")
```

Sum where Product starts with "Product"

Not Equal To

```
=SUMIFS(C2:C10, B2:B10, "<>Excluded")
```

Sum all except where value is "Excluded"

Greater Than Cell

```
=SUMIFS(C2:C10, C2:C10, ">"&E1)
```

Sum values greater than the value in E1

Comparison Operators

`=` Equal to (default)	`<>` Not equal to	`>` Greater than	`<` Less than
`>=` Greater than or equal	`<=` Less than or equal	`*` Wildcard (any characters)	`?` Wildcard (single character)

IF Function Cheatsheet

Logical tests and conditional returns

Syntax

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

Basic Examples

Basic IF

```
=IF(A1>10, "Yes", "No")
```

If A1 > 10, return "Yes"; otherwise "No"

Nested IF

```
=IF(A1>10, "High", IF(A1>5, "Medium", "Low"))
```

Multiple conditions with cascading results

Advanced Options

IF with AND

```
=IF(AND(A1>10, B1="Yes"), "Pass", "Fail")
```

Both conditions must be true

IF with OR

```
=IF(OR(A1>10, B1="Yes"), "Pass", "Fail")
```

Either condition can be true

Error Handling

IFERROR

```
=IFERROR(A1/B1, 0)
```

Return 0 if formula errors

IFNA

```
=IFNA(VLOOKUP(...), "Not Found")
```

Handle #N/A errors specifically

INDEX & MATCH

The powerful lookup combination for any scenario

Combined Syntax

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

INDEX Arguments

`array`	The range of cells to return a value from
`row_num`	The row number to return
`[col_num]`	Optional. The column number to return

MATCH Arguments

`lookup_value`	The value to search for
`lookup_array`	The range to search in
`[match_type]`	0 = exact, 1 = less than, -1 = greater than

Examples

Basic INDEX/MATCH

```
=INDEX(C2:C10, MATCH("Apple", A2:A10, 0))
```

Find "Apple" in column A, return value from column C

2D Lookup

```
=INDEX(B2:D10, MATCH("Q2", A2:A10, 0), MATCH("Sales", B1:D1, 0))
```

Look up both row and column headers

Left Lookup

```
=INDEX(A2:A10, MATCH(100, C2:C10, 0))
```

Return value from column to the LEFT of search column

INDEX/MATCH vs XLOOKUP

ASPECT	INDEX/MATCH	XLOOKUP
Direction	Any direction	Any direction
2D Lookup	Yes (native)	Requires nesting
Complexity	More complex	Simpler syntax
Compatibility	All versions	Excel 365/2021+

COUNTIFS Cheatsheet

Count cells based on multiple criteria

Syntax

```
=COUNTIFS(criteria_range1, criteria1, [criteria_range2, criteria2], ...)
```

Arguments

`criteria_range1`	The first range to evaluate
`criteria1`	The condition to apply to range1
`[criteria_range2, ...]`	Additional ranges and conditions

Examples

Single Condition

```
=COUNTIFS(B2:B10, "Yes")
```

Count cells where value is "Yes"

Multiple Conditions

```
=COUNTIFS(B2:B10, "North", C2:C10, ">1000")
```

Count where Region is "North" AND Sales > 1000

Date Range

```
=COUNTIFS(A2:A10, ">=1/1/2024", A2:A10, "<=31/12/2024")
```

Count entries within 2024

Not Equal

```
=COUNTIFS(B2:B10, "<>Cancelled")
```

Count all except "Cancelled"

With Wildcards

```
=COUNTIFS(B2:B10, "Product*")
```

Count cells starting with "Product"

Non-Blank Cells

```
=COUNTIFS(B2:B10, "<>")
```

Count all non-empty cells

Comparison Operators

`=` Equal (default)	`<>` Not equal	`>` Greater than	`<` Less than
`>=` Greater or equal	`<=` Less or equal	`*` Any characters	`?` Single character

Date Functions Guide

Essential functions for working with dates

Key Date Functions

TODAY()

```
=TODAY()
```

Returns the current date

Use for age calculations, due dates

NOW()

```
=NOW()
```

Returns current date and time

Timestamps, real-time tracking

DATE()

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

Creates a date from components

=DATE(2024, 12, 25) → Dec 25, 2024

EOMONTH()

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

Last day of month after adding months

=EOMONTH(A1, 3) → 3 months later, end of month

EDATE()

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

Date after adding/subtracting months

=EDATE(A1, -6) → 6 months before

NETWORKDAYS()

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

Working days between two dates

Exclude weekends and holidays

Extract Date Parts

YEAR(date) Extract year (e.g., 2024)

MONTH(date) Extract month (1–12)

DAY(date) Extract day (1–31)

WEEKDAY(date) Day of week (1–7)

WEEKNUM(date) Week number (1–52)

Pro Tips

- ✓ Dates are stored as numbers (days since 1/1/1900)
- ✓ Use DATE() to avoid regional format issues
- ✓ EOMONTH is perfect for financial period ends

FILTER Function Guide

Dynamic filtering with Excel 365

Syntax

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

Arguments

`array` The range or array to filter

`include` Boolean array (conditions that evaluate to TRUE/FALSE)

`[if_empty]` Optional. Value to return if no results

Examples

Single Condition

```
=FILTER(A2:C10, B2:B10="Yes")
```

Return rows where column B is "Yes"

Multiple Conditions (AND)

```
=FILTER(A2:C10, (B2:B10="North")*(C2:C10>1000))
```

Region is "North" AND Sales > 1000

With Empty Handler

```
=FILTER(A2:C10, B2:B10="East", "No data")
```

Returns "No data" if no matches

Key Features

Dynamic Arrays

Results automatically spill to adjacent cells

Multiple Columns

Returns entire rows, not just one column

Live Updates

Results update when source data changes

Nestable

Combine with SORT, UNIQUE, etc.

XMATCH Cheatsheet

Modern position matching in Excel 365

Syntax

```
=XMATCH(lookup_value, lookup_array, [match_mode], [search_mode])
```

Arguments

`lookup_value`	The value you want to search for
`lookup_array`	The range to search within
`[match_mode]`	0 = exact (default), -1 = next smallest, 1 = next largest, 2 = wildcard
`[search_mode]`	1 = first-to-last, -1 = last-to-first, 2 = binary ascending, -2 = binary descending

Examples

Exact Match

```
=XMATCH("Apple", A2:A10)
```

Returns position of "Apple" in the range

Wildcard Search

```
=XMATCH("*son", A2:A10, 2)
```

Finds names ending with "son"

Reverse Search

```
=XMATCH("Yes", C2:C20, 0, -1)
```

Finds last occurrence of "Yes"

XMATCH vs MATCH

Feature	XMATCH	MATCH
Exact match default	✓	✗
Reverse search	✓	✗
Wildcard support	✓	✗
Binary search	✓	✓
No sort required	✓	✗

LAMBDA Function Guide

Create your own custom functions

Syntax

```
=LAMBDA(parameter1, [parameter2, ...], calculation)
```

Arguments

`parameter1` First input parameter name

`[parameter2]` Additional parameters (up to 253)

`calculation` The formula to execute using parameters

Examples

Simple Calculation

→ 10

```
=LAMBDA(x, x*2)(5)
```

Doubles the input value

Tax Calculator

→ 120

```
=LAMBDA(price, rate, price*(1+rate))(100, 0.2)
```

Calculates price with tax

Creating Named Functions

1. Go to Formulas → Name Manager
2. Click "New" and enter a name
3. In "Refers to", enter your LAMBDA
4. Use your function anywhere!

Name: ADDTAX

```
=LAMBDA(price, rate, price*(1+rate))
```

```
Usage: =ADDTAX(100, 0.2)
```

Why Use LAMBDA?

Reusability

Define once, use everywhere

No VBA

Custom functions without macros

Advanced Usage

Recursive Factorial

```
=LAMBDA(n, IF(n<=1, 1, n*FACTORIAL(n-1)))
```

Self-referencing LAMBDA

Array Mapping

```
=MAP(A1:A5, LAMBDA(x, x^2))
```

Apply LAMBDA to each cell

LET Function Guide

Variables inside formulas

Syntax

```
=LET(name1, value1, [name2, value2, ...], calculation)
```

Arguments

- `name1` First variable name
- `value1` Value or calculation for name1
- `[name2, value2]` Additional name-value pairs
- `calculation` Final calculation using the variables

Examples

Simple Variable

```
=LET(x, A1*2, x+10)
```

Store A1*2 as "x", then use it

Multiple Variables

```
=LET(sales, B1, cost, C1, profit, sales-cost, profit/sales)
```

Calculate profit margin with named variables

Avoid Repetition

❌ Without LET (repeats VLOOKUP):

```
=IF(VLOOKUP(A1,B:C,2,0)>100, VLOOKUP(A1,B:C,2,0)*0.9, VLOOKUP(A1,B:C,2,0))
```

✓ With LET (lookup once):

```
=LET(val, VLOOKUP(A1,B:C,2,0), IF(val>100, val*0.9, val))
```

Benefits

Performance

Calculate once, reuse result

Readability

Self-documenting variable names

Pro Tips

- Use descriptive variable names (revenue vs r)
- Combine with LAMBDA for reusable functions

Text Functions Guide

Extract, transform, and manipulate text

Core Text Functions

`LEFT`	Extract characters from start	`=LEFT("Hello", 2) → "He"`
`RIGHT`	Extract characters from end	`=RIGHT("Hello", 2) → "lo"`
`MID`	Extract from middle	`=MID("Hello", 2, 3) → "ell"`
`LEN`	Count characters	`=LEN("Hello") → 5`
`FIND`	Find position (case-sensitive)	`=FIND("l", "Hello") → 3`
`SEARCH`	Find position (case-insensitive)	`=SEARCH("L", "Hello") → 3`

Transform Functions

`UPPER` `=UPPER("hello") → "HELLO"`

`LOWER` `=LOWER("HELLO") → "hello"`

`PROPER` `=PROPER("hello world") → "Hello World"`

`TRIM` `=TRIM(" hello ") → "hello"`

`CLEAN` `=CLEAN(A1) → removes non-printable chars`

Common Use Cases

Extract First Name

From "John Smith" get "John"

```
=LEFT(A1, FIND(" ", A1)-1)
```

Extract Last Name

From "John Smith" get "Smith"

```
=RIGHT(A1, LEN(A1)-FIND(" ", A1))
```

Extract Domain

From email get domain

```
=MID(A1, FIND("@", A1)+1, LEN(A1))
```

Combine Text

Join first and last names

```
=CONCAT(A1, " ", B1) or =A1&" "&B1
```

Excel 365 Text Functions

`TEXTJOIN`	`=TEXTJOIN(", ", TRUE, A1:A5)`	Join with delimiter
`TEXTSPLIT`	`=TEXTSPLIT(A1, " ")`	Split text into cells
`TEXTBEFORE`	`=TEXTBEFORE(A1, "@")`	Text before delimiter
`TEXTAFTER`	`=TEXTAFTER(A1, "@")`	Text after delimiter

Error Handling Guide

IFERROR, IFNA & debugging formulas

Common Excel Errors

Error	Cause	Fix
#VALUE!	Wrong type of argument or operand	Check data types
#REF!	Invalid cell reference	Fix deleted references
#NAME?	Unrecognized formula name	Check spelling
#DIV/0!	Division by zero	Add zero check
#N/A	Value not available	Use IFNA or IFERROR
#NULL!	Incorrect range operator	Check range syntax
#NUM!	Invalid numeric value	Verify calculations

Error Handling Functions

IFERROR Returns custom value for ANY error

```
=IFERROR(A1/B1, 0)
```

IFNA Returns custom value for #N/A only

```
=IFNA(VLOOKUP(A1,B:C,2,0), "Not Found")
```

Error Check Functions

ISERROR

Returns TRUE if value is any error

ISNA

Returns TRUE if value is #N/A

Best Practices

Use IFNA for Lookups

✓ Better:

```
=IFNA(XLOOKUP(A1,B:B,C:C), "Not Found")
```

✗ Avoid:

```
=IFERROR(XLOOKUP(A1,B:B,C:C), "Not Found")
```

Handle Division Safely

✓ Better:

```
=IF(B1=0, 0, A1/B1)
```

✗ Avoid:

```
=IFERROR(A1/B1, 0)
```

Debugging Tips

- F9 key evaluates selected formula parts
- Use Evaluate Formula (Formulas tab)
- Error Checking tool highlights issues
- FORMULATEXT() shows formula as text

Lookup Functions

Complete guide to finding data in Excel

Function Reference

`VLOOKUP`	Search first column, return from another	*Legacy, left-to-right only*
`HLOOKUP`	Search first row, return from another	*Legacy, top-to-bottom only*
`XLOOKUP`	Modern replacement for V/HLOOKUP	*Any direction, better defaults*
`INDEX`	Return value at intersection	*Works with MATCH*
`MATCH`	Find position in range	*Combine with INDEX*
`XMATCH`	Modern MATCH with more options	*Reverse search, wildcards*

Which Function to Use?

Scenario	Best Choice
Simple vertical lookup	XLOOKUP or VLOOKUP
Lookup to the left	XLOOKUP or INDEX/MATCH
Two-way lookup (row & col)	INDEX + XMATCH
Find last occurrence	XLOOKUP with -1 search
Approximate match	XLOOKUP with match_mode

Common Patterns

XLOOKUP (Recommended)

```
=XLOOKUP(A1, Products, Prices, "Not found")
```

INDEX/MATCH Classic

```
=INDEX(C:C, MATCH(A1, B:B, 0))
```

Two-Way Lookup

```
=INDEX(B2:E10, XMATCH(G1,A2:A10), XMATCH(G2,B1:E1))
```

Pro Tips

✓ Do

Use XLOOKUP for new work—it's more flexible and has better defaults

✗ Avoid

VLOOKUP with TRUE for range lookup—data must be sorted

💡 Tip

Wrap lookups in IFERROR or use XLOOKUP's if_not_found argument

💡 Tip

INDEX/MATCH still works when XLOOKUP isn't available

Conditional Functions

Logic, conditions & criteria-based calculations

Logic Functions

`IF`	Single condition check
`IFS`	Multiple conditions (no nesting)
`AND`	TRUE if ALL conditions true
`OR`	TRUE if ANY condition true
`NOT`	Reverses TRUE/FALSE
`XOR`	TRUE if ODD number true

Conditional Calculations

`SUMIF`	Sum with 1 condition
`SUMIFS`	Sum with multiple conditions
`COUNTIF`	Count with 1 condition
`COUNTIFS`	Count with multiple conditions
`AVERAGEIF`	Average with 1 condition
`AVERAGEIFS`	Average with multiple conditions

Examples

Nested IF → IFS

❌ Nested:

```
=IF(A1>90,"A",IF(A1>80,"B",IF(A1>70,"C","F")))
```

✓ IFS:

```
=IFS(A1>90,"A",A1>80,"B",A1>70,"C",TRUE,"F")
```

Multiple Conditions

```
=SUMIFS(Sales, Region, "North", Date, ">="&DATE(2024,1,1))
```

Sum sales in North from 2024

AND/OR in IF

```
=IF(AND(A1>100, B1="Yes"), "Approved", "Review")
```

Both conditions must be true

Wildcard Characters

`*`	Any characters	`="*smith" matches "John Smith"`
`?`	Single character	`="Sm?th" matches "Smith" or "Smyth"`
`~`	Escape wildcard	`="~*" matches literal asterisk`

Dynamic Array Functions

FILTER, SORT, UNIQUE & more (Excel 365)

Core Array Functions

FILTER	Return rows matching criteria	`=FILTER(A:C, B:B="Yes")`
SORT	Sort array by column	`=SORT(A:C, 2, -1)`
SORTBY	Sort by external range	`=SORTBY(Names, Scores, -1)`
UNIQUE	Remove duplicates	`=UNIQUE(A2:A100)`
SEQUENCE	Generate number series	`=SEQUENCE(10, 1, 1, 1)`
RANDARRAY	Generate random numbers	`=RANDARRAY(5, 3, 1, 100)`

Power Combinations

Filter + Sort

```
=SORT(FILTER(Data, Region="North"), 3, -1)
```

Filter by region, sort by column 3 descending

Unique + Sort

```
=SORT(UNIQUE(A2:A100))
```

Get unique values and sort them

Unique + Filter

```
=UNIQUE(FILTER(Products, Sales>1000))
```

Unique products with sales over 1000

Sequence for Dates

```
=SEQUENCE(12, 1, DATE(2024,1,1), 30)
```

Generate 12 monthly dates

Understanding Spill

Spill Range

Results automatically expand into adjacent cells

Spill Reference

Use A1# to reference entire spill range

Array Helper Functions

TAKE

First/last n rows or columns

DROP

Remove first/last n rows

VSTACK

Stack arrays vertically

HSTACK

Stack arrays horizontally

Financial Functions

Loans, valuations & time value of money

Loan & Payment Functions

`PMT`	Calculate loan payment	`=PMT(5%/12, 360, -250000)`
`IPMT`	Interest portion of payment	`=IPMT(5%/12, 1, 360, -250000)`
`PPMT`	Principal portion of payment	`=PPMT(5%/12, 1, 360, -250000)`
`RATE`	Calculate interest rate	`=RATE(360, -1342, 250000)*12`
`NPER`	Number of periods	`=NPER(5%/12, -1500, 250000)`

Valuation Functions

`NPV`	Net present value (regular)	*Assumes end-of-period cash flows*
`XNPV`	NPV with specific dates	*Use for irregular cash flows*
`IRR`	Internal rate of return	*Assumes regular periods*
`XIRR`	IRR with specific dates	*Use for irregular cash flows*
`PV`	Present value	*What future cash is worth today*
`FV`	Future value	*Value of investment over time*

Real-World Examples

Monthly Loan Payment

```
=PMT(6%/12, 60, -25000)
```

$483.32/month for 5yr car loan at 6%

Investment NPV

```
=XNPV(10%, B2:B10, A2:A10)
```

NPV of cash flows with actual dates

Savings Future Value

```
=FV(7%/12, 240, -500, -10000)
```

Value after 20yrs of $500/mo + $10k initial

Key Tips

- Use negative for cash outflows (payments you make)
- Divide annual rate by 12 for monthly calculations
- Multiply by 12 periods for annual rate
- XNPV/XIRR are more accurate for real-world scenarios

Statistical Functions

Analysis, trends & data distribution

Central Tendency

- `AVERAGE` Mean of values
- `MEDIAN` Middle value
- `MODE.SNGL` Most frequent value
- `MODE.MULT` All modes (array)

Dispersion & Variability

`STDEV.S`
Sample standard deviation
Most common

`STDEV.P`
Population std dev
Full dataset

`VAR.S`
Sample variance
Squared std dev

`VAR.P`
Population variance
Full dataset

Ranking & Position

- `MIN` Smallest value
- `MAX` Largest value
- `LARGE` Nth largest
- `SMALL` Nth smallest
- `RANK` Rank in dataset
- `PERCENTILE` 90th percentile
- `PERCENTRANK` Percentile rank
- `QUARTILE` Quartile value (0–4)

Counting Functions

- `COUNT` Count numbers
- `COUNTA` Count non-empty
- `COUNTBLANK` Count empty cells
- `COUNTIF` Count with criteria

Advanced Analysis

`CORREL`
Correlation coefficient

`COVARIANCE.S`
Sample covariance

`FORECAST.LINEAR`
Linear prediction

`TREND`
Linear trend values

`GROWTH`
Exponential trend

`LINEST`
Regression statistics

Rounding & Precision

Essential functions for accurate financial calculations

Rounding Functions

`ROUND`	Standard rounding	`=ROUND(3.456, 2) → 3.46`
`ROUNDUP`	Always rounds up	`=ROUNDUP(3.421, 2) → 3.43`
`ROUNDDOWN`	Always rounds down	`=ROUNDDOWN(3.789, 2) → 3.78`
`MROUND`	Round to multiple	`=MROUND(17, 5) → 15`
`CEILING`	Round up to multiple	`=CEILING(23, 10) → 30`
`FLOOR`	Round down to multiple	`=FLOOR(27, 10) → 20`

Truncation & Division

`TRUNC`	Truncate to decimals (no rounding)	`=TRUNC(3.789, 2) → 3.78`
`INT`	Round down to integer	`=INT(3.9) → 3`
`MOD`	Remainder after division	`=MOD(17, 5) → 2`
`QUOTIENT`	Integer portion of division	`=QUOTIENT(17, 5) → 3`

Accounting Use Cases

Round to Nearest Cent

```
=ROUND(A1*B1, 2)
```

Price × Quantity to 2 decimals

Round to Thousands

```
=ROUND(Revenue, -3)
```

Use negative decimals for left-of-decimal rounding

Currency Rounding (0.05)

```
=MROUND(Total, 0.05)
```

Round to nearest 5 cents (Swiss Franc, etc.)

Allocation Without Pennies

```
=FLOOR(Amount/3, 0.01)
```

Split amounts evenly, handle remainder separately

Avoiding Penny Errors

- ✗ Rounding errors compound
- ✗ Allocations don't sum correctly
- ✗ Tax calculations differ
- ✗ Floating point precision
- ✓ Round final results, not intermediate steps
- ✓ Use FLOOR then add remainder to last item
- ✓ Match jurisdiction's rounding rules exactly
- ✓ Use ROUND on comparisons: ROUND(A1−B1,2)=0

Variance & Budget Analysis

Essential formulas for financial reporting

Core Variance Formulas

Variance ($)	`=Actual - Budget`	Absolute difference
Variance (%)	`=(Actual-Budget)/ABS(Budget)`	Percentage change
Favorable/Unfavorable	`=IF(Actual>Budget, "Fav", "Unfav")`	For revenue items
Expense Variance	`=IF(Actual<Budget, "Fav", "Unfav")`	Lower spend = favorable

YTD & Forecasting

YTD Actual

```
=SUMIFS(Actuals, Dates, ">="&DATE(YEAR(TODAY()),1,1), Dates, "<="&TODAY())
```

Sum from Jan 1 to today

YTD Budget

```
=SUMIFS(Budget, Month, "<="&MONTH(TODAY()))
```

Budget through current month

Full Year Forecast

```
=YTD_Actual + Remaining_Budget
```

Actual + future budget periods

Run Rate Projection

```
=(YTD_Actual/MONTH(TODAY()))*12
```

Annualize current performance

Key Metrics

Budget Attainment	`=Actual/Budget`	>100% = over budget (expense) or beat target (revenue)
Spend Rate	`=YTD_Spend/Annual_Budget`	Should roughly equal elapsed time %
Forecast Accuracy	`=1-ABS(Actual-Forecast)/Actual`	Closer to 100% = better forecast
Cost per Unit	`=Total_Cost/Units`	Track efficiency over time

Pro Tips

- Use ABS() in denominators to handle negative budgets correctly
- Use IFERROR to handle divide-by-zero gracefully
- Create named ranges for Budget and Actual columns
- Consider seasonality in YTD comparisons

Data Validation Essentials

Ensure data integrity for accurate reporting

Validation Types

`Whole Number`	Quantities, counts, IDs	Between 1 and 1000
`Decimal`	Prices, percentages, rates	Between 0 and 100
`List`	Dropdown selections	Yes, No, Pending
`Date`	Transaction dates, due dates	Greater than 1/1/2020
`Text Length`	Codes, account numbers	Exactly 10 characters
`Custom`	Complex rules	=AND(A1>0, A1<B1)

Custom Validation Rules

Positive Numbers Only	`=A1>0`	Revenue, quantities
Valid Account Code	`=AND(LEN(A1)=6, ISNUMBER(VALUE(A1)))`	6-digit numeric codes
Date Not in Future	`=A1<=TODAY()`	Transaction dates
Unique Entry	`=COUNTIF($A:$A, A1)=1`	Prevent duplicates
Depends on Another Cell	`=IF($B1="Required", A1<>"", TRUE)`	Conditional requirements

Error Alert Styles

🛑 **Stop**

Prevents invalid entry

⚠️ **Warning**

Warns but allows override

ℹ️ **Information**

Just shows message

Data Cleaning Functions

`TRIM`

Remove extra spaces

`CLEAN`

Remove non-printable chars

`UPPER/LOWER/PROPER`

Standardize case

`VALUE`

Convert text to number

`DATEVALUE`

Convert text to date

`SUBSTITUTE`

Replace specific characters

Pivot Table Functions

Extract data from PivotTables for custom reports

GETPIVOTDATA Syntax

```
=GETPIVOTDATA(data_field, pivot_table, [field1, item1], ...)
```

`data_field`	Name of the value field to retrieve
`pivot_table`	Reference to any cell in the pivot table
`field, item`	Pairs specifying the filter criteria

Examples

Basic Retrieval

```
=GETPIVOTDATA("Revenue", $A$3, "Region", "North")
```

Get Revenue for North region

Multiple Criteria

```
=GETPIVOTDATA("Revenue", $A$3, "Region", "North", "Year", 2024)
```

Revenue for North in 2024

Grand Total

```
=GETPIVOTDATA("Revenue", $A$3)
```

No criteria = grand total

Reporting Patterns

Monthly Comparison

Pull specific periods dynamically

```
=GETPIVOTDATA("Sales", $A$3, "Month", A1)
```

YoY Variance

Calculate year-over-year changes

```
=GETPIVOTDATA("Rev",$A$3,"Year",2024)-GETPIVOTDATA("Rev",$A$3,"Year",2023)
```

Tips

Auto-generate GETPIVOTDATA

Click pivot cell while typing formula

Disable auto-generation

Options → Formulas → uncheck "Use GetPivotData"

Handle errors

=IFERROR(GETPIVOTDATA(...), 0)

Refresh linked formulas

Pivot refresh updates GETPIVOTDATA automatically

Audit & Review Functions

Verify, document & troubleshoot spreadsheets

Audit Functions

`FORMULATEXT`	Display formula as text	Document formulas
`ISFORMULA`	Check if cell has formula	Find hardcoded values
`INFO`	System/file information	Track file location
`CELL`	Cell/file properties	Dynamic file references
`SHEET`	Sheet number	Dynamic sheet references
`SHEETS`	Count of sheets	Verify workbook structure

Formula Investigation

Show All Formulas in Range

```
=FORMULATEXT(A1)
```

Drag down to show each formula

Find Hardcoded Cells

```
=IF(ISFORMULA(A1), "Formula", "Hardcoded")
```

Identify manual entries vs calculations

Check for External Links

```
=IF(ISERROR(FIND("[", FORMULATEXT(A1))), "Internal", "External")
```

Detect links to other workbooks

Audit Shortcuts

`Ctrl + [`	Trace Precedents	`Ctrl + ]`	Trace Dependents
`Alt, M, A, A`	Remove Arrows	`Alt, M, V`	Error Checking

Spreadsheet Review Checklist

Hardcoded values in formulas	High	Search for numbers in formulas
Circular references	High	File → Options → Formulas
External links	Medium	Edit → Links
Named range errors	Medium	Name Manager → filter errors
Hidden rows/columns	Medium	Check for discontinuous ranges
Inconsistent formulas	Medium	Error checking in ribbon

Cash Flow & Investment Analysis

NPV, IRR, and capital budgeting essentials

Net Present Value Functions

`NPV`	Net present value of periodic cash flows	`=NPV(10%, CF1:CF5)+Initial`
`XNPV`	NPV for irregular cash flow dates	`=XNPV(10%, B2:B10, A2:A10)`

Internal Rate of Return

`IRR`	Internal rate of return (periodic)	`=IRR(B2:B10)` → 12.5%
`XIRR`	IRR for irregular cash flow dates	`=XIRR(B2:B10, A2:A10)`
`MIRR`	Modified IRR with reinvestment rate	`=MIRR(CF, 8%, 10%)`

Practical Applications

Project Comparison

```
=XNPV(WACC, ProjectA_CF, Dates) - XNPV(WACC, ProjectB_CF, Dates)
```

Compare two investment options using NPV difference

Breakeven Analysis

```
=MATCH(TRUE, SCAN(0, CashFlows, LAMBDA(a,b,a+b))>0, 0)
```

Find the period when cumulative cash flow turns positive

Payback Period

```
=-InitialInvest/AVERAGE(AnnualCashFlows)
```

Simple payback (use discounted CF for more accuracy)

Present Value Factor

```
=1/(1+rate)^period
```

Manual PV factor for cash flow discounting

Investment Decision Rules

`NPV > 0`	**Accept**	Project adds value
`IRR > WACC`	**Accept**	Returns exceed cost of capital
$NPV_1 > NPV_2$	**Choose Project 1**	Higher absolute value creation
`Payback < Target`	**Accept**	Meets liquidity requirements

Watch Out For

- ✗ NPV excludes initial investment
- ✗ IRR assumes reinvestment at IRR
- ✗ Multiple IRRs for non-conventional CF
- ✗ Comparing projects of different sizes
- ✓ Add initial (negative) CF separately: =NPV(...)+B1
- ✓ Use MIRR for realistic reinvestment rate
- ✓ Rely on NPV or specify IRR guess
- ✓ Use profitability index: NPV/Initial Investment

Depreciation Functions

Fixed asset depreciation calculations for accounting

Depreciation Functions

`SLN`	Straight-line depreciation (equal each year)	`=SLN(10000, 1000, 5) → 1,800`
`DB`	Declining balance (fixed rate)	`=DB(10000, 1000, 5, 1) → 3,690`
`DDB`	Double declining balance	`=DDB(10000, 1000, 5, 1) → 4,000`
`VDB`	Variable declining balance	`=VDB(10000, 1000, 5, 0, 1)`
`SYD`	Sum-of-years digits	`=SYD(10000, 1000, 5, 1) → 3,000`

Method Comparison

Straight-Line	Equal amounts	Assets with even usage over time
Declining Balance	Decreasing amounts	Assets losing value quickly
Sum-of-Years	Accelerated decline	Technology, vehicles
MACRS (VDB)	IRS-specified rates	US tax compliance

Practical Applications

Fixed Asset Schedule

```
=SLN(PurchasePrice, SalvageValue, UsefulLife)
```

Calculate annual depreciation for each asset

Accumulated Depreciation

```
=SLN(Cost, Salvage, Life) * MIN(Period, Life)
```

Total depreciation through current period

Book Value

```
=Cost - VDB(Cost, Salvage, Life, 0, Period)
```

Net asset value at any point in time

Mid-Year Convention

```
=DB(Cost, Salvage, Life, Period, 6)
```

First year partial depreciation (6 months)

Tax vs. Book Depreciation

Method	Tax: MACRS (accelerated)	Book: Straight-line (usually)
Useful Life	Tax: IRS class lives	Book: Economic useful life
Salvage Value	Tax: Ignored (depreciate to 0)	Book: Estimated residual
Result	Tax: Deferred tax liability	Book: Matching principle

Pro Tips

- ✓ Use VDB with no_switch=TRUE to prevent automatic switch to straight-line
- ✓ Track book vs. tax depreciation for deferred tax calculations

Loan & Amortization

Payment calculations and debt schedule functions

Payment Functions

`PMT`	Calculate periodic payment amount	`=PMT(5%/12, 60, -25000) → $471.78`
`IPMT`	Interest portion of specific payment	`=IPMT(5%/12, 1, 60, -25000) → $104.17`
`PPMT`	Principal portion of specific payment	`=PPMT(5%/12, 1, 60, -25000) → $367.61`
`RATE`	Calculate interest rate	`=RATE(60, -471.78, 25000)*12 → 5%`
`NPER`	Number of periods	`=NPER(5%/12, -500, 25000) → 55.5`

Cumulative Functions

`CUMIPMT`	Cumulative interest over period range	`=CUMIPMT(5%/12, 60, 25000, 1, 12, 0) → $1,178`
`CUMPRINC`	Cumulative principal over period range	`=CUMPRINC(5%/12, 60, 25000, 1, 12, 0) → $4,483`

Building an Amortization Schedule

1	**Payment Number**	`=ROW()-1 (or sequence)`
2	**Payment Amount**	`=PMT($Rate, $Nper, $PV)`
3	**Interest Portion**	`=IPMT($Rate, Period, $Nper, $PV)`
4	**Principal Portion**	`=PPMT($Rate, Period, $Nper, $PV)`
5	**Remaining Balance**	`=PrevBalance + Principal`

Common Calculations

Mortgage Payment

```
=PMT(4.5%/12, 30*12, -350000)
```

30-year mortgage at 4.5% APR

Total Interest Cost

```
=PMT(Rate, Nper, PV)*Nper + PV
```

Lifetime interest paid on a loan

Payment Frequency Conversions

Monthly	Annual Rate / 12	Years × 12
Bi-Weekly	Annual Rate / 26	Years × 26
Weekly	Annual Rate / 52	Years × 52
Quarterly	Annual Rate / 4	Years × 4

Pro Tips

- ✓ Use negative PV for loans (cash received), positive for investments
- ✓ Type=0 for end-of-period, Type=1 for beginning payments

Aggregation Functions

SUMPRODUCT, SUBTOTAL, and AGGREGATE mastery

SUMPRODUCT Patterns

Weighted Average

```
=SUMPRODUCT(Values, Weights)/SUM(Weights)
```

Calculate weighted average without helper column

Conditional Sum

```
=SUMPRODUCT((Region="East")*(Product="A")*Sales)
```

Multiple criteria without SUMIFS

Count with Criteria

```
=SUMPRODUCT((Status="Complete")*1)
```

Count matching rows

SUBTOTAL Function Codes

1 / 101	**AVERAGE**	101 ignores hidden rows
2 / 102	**COUNT**	102 ignores hidden rows
3 / 103	**COUNTA**	103 ignores hidden rows
9 / 109	**SUM**	109 ignores hidden rows
4 / 104	**MAX**	104 ignores hidden rows

AGGREGATE Options

=AGGREGATE(function_num, options, range)

1	**AVERAGE**	Ignore errors	Ignore hidden + errors
4	**MAX**	Ignore errors	Ignore hidden + errors
5	**MIN**	Ignore errors	Ignore hidden + errors
9	**SUM**	Ignore errors	Ignore hidden + errors
14	**LARGE**	Ignore errors	Ignore hidden + errors

Accounting Applications

Filtered Trial Balance

```
=SUBTOTAL(109, DebitCol) - SUBTOTAL(109, CreditCol)
```

Works with filtered data automatically

Named Ranges & Tables

Structured references for maintainable spreadsheets

Creating Named Ranges

Name Box	Select range → Click Name Box → Type name → Enter	Fastest
Name Manager	Formulas tab → Name Manager → New	Most control
Create from Selection	Select data with headers → Formulas → Create from Selection	Bulk creation

Structured Table References

`TableName[ColumnName]`	All data in column (no headers)	`Sales[Revenue]`
`TableName[@ColumnName]`	Current row value only	`[@Quantity]*[@Price]`
`TableName[#Headers]`	Header row only	`Sales[#Headers]`
`TableName[#Data]`	All data rows (no headers)	`SUM(Sales[#Data])`

Dynamic Range Techniques

OFFSET-based Dynamic Range

```
=OFFSET(Sheet1!$A$1, 0, 0, COUNTA(Sheet1!$A:$A), 1)
```

Expands as data is added (older method)

Spill Range Reference

```
=UNIQUE(Data)#
```

Reference entire spill range with # operator

Naming Conventions

`rng_`	Named ranges: rng_SalesData, rng_Accounts	Easy to identify in formulas
`tbl_`	Tables: tbl_Transactions, tbl_Employees	Distinguishes from ranges
`const_`	Constants: const_TaxRate, const_FiscalYear	Identifies static values

Accounting Applications

Chart of Accounts Lookup

```
=XLOOKUP(AcctNum, rng_ChartOfAccounts, rng_AcctNames)
```

Named ranges make formulas self-documenting

Ready to Automate Your Accounting?

Stop spending hours on manual tasks. Let AI handle the repetitive work.

How Accounts Draft Helps

- Streamline your accounting workflows
- Reduce time spent on repetitive tasks
- AI-powered automation for accountants
- Focus on high-value work

Try Accounts Draft Free

www.accountsdraft.com

AI software that automates accounting workflows