PHOENICIA UNIVERSITY

XLOOKUP

The Modern Way to Find Anything in a Spreadsheet

A hands-on lesson on syntax, arguments, and real-world lookup patterns

2 One Function, Any Direction

> *"XLOOKUP finally does what everyone assumed VLOOKUP already did."*
>
> **— A common analyst's rule of thumb**

- Searches a range for a value and returns the matching result from another range.
- Works left-to-right or right-to-left — unlike VLOOKUP, column order no longer matters.
- Returns a clear fallback value you control, instead of a cryptic #N/A error.

3

older functions XLOOKUP replaces: VLOOKUP, HLOOKUP, and INDEX/MATCH

Introduced in Excel 2019 and Microsoft 365, it's now the standard lookup tool taught in every modern course.

3 Syntax and Arguments

One formula, six arguments — only the first three are required.

The Syntax

- =XLOOKUP(lookup_value, lookup_array, return_array, [if_not_found], [match_mode], [search_mode])
- Only the first three arguments are required; every argument in brackets has a sensible default.
- lookup_array and return_array must have the same number of rows (or columns), but can sit anywhere in the workbook.
- There's no ribbon button — type it directly, or find it under Formulas → Lookup & Reference.

Reading the Arguments

- lookup_value — what you're searching for, e.g. an employee ID typed into a cell.
- lookup_array — the column or row Excel searches within.
- return_array — the column or row Excel pulls the answer from.
- if_not_found — the text or value shown instead of #N/A when there's no match.

4 XLOOKUP vs. VLOOKUP

The same job, but XLOOKUP removes almost every VLOOKUP limitation.

VLOOKUP (The Old Way)

- Can only search left to right; the lookup column must be the first column in the range.
- Breaks silently if a column is inserted, since it counts columns by number.
- Defaults to approximate match, which can return wrong answers if you forget to fix it.
- Cannot search from the last row backward to find the most recent match.

XLOOKUP (The Modern Way)

- Searches in either direction — left-to-right or right-to-left from the lookup column.
- References whole arrays directly, so inserting a column doesn't break the formula.
- Defaults to exact match, the safer choice for almost every real lookup.
- Can search from the last row first, to find the most recent match instantly.

5 A Basic Lookup in Action

Concrete example: pulling an employee's name and department from an ID.

Building the Formula

- Type an ID into a cell, e.g. F2.
- In another cell: =XLOOKUP(F2, A2:A20, B2:B20).
- F2 holds the ID typed in; A2:A20 is the ID column; B2:B20 is the Name column.
- Press Enter — the match appears instantly, and updates if F2 changes.
- Add a second XLOOKUP with C2:C20 as the return array to pull Department too.

ID	Name	Department
E101	Dana Haddad	Finance
E102	**Karim Fares**	**Marketing**
E103	Layla Aziz	Finance
E104	Sami Nour	Operations

Search: E102 → Karim Fares, Marketing

6 Handling Missing Matches

The if_not_found argument turns a scary error into a clear message.

Using if_not_found

- The fourth argument replaces #N/A with any text, number, or blank you choose.
- No IFERROR wrapper needed — XLOOKUP handles the missing-match case on its own.
- Try =XLOOKUP(F2, A2:A20, B2:B20, "Not Found") for a friendly fallback.
- A clean fallback message is friendlier for dashboards and reports other people will read.

ID	Name
E101	Dana Haddad
E102	Karim Fares
E103	Layla Aziz

Search: E999 → *"Not Found"*

```
Formula: =XLOOKUP(F2, A2:A4, B2:B4, "Not Found")
```

7 Exact vs. Approximate Match

The fifth argument, match_mode, controls exactly how XLOOKUP decides on a match.

0 — Exact Match

The default. Returns a result only if the value matches exactly, otherwise triggers if_not_found.

-1 — Exact or Next Smaller

Falls back to the closest value below the target — useful for tax brackets or shipping-rate tables.

1 — Exact or Next Larger

Falls back to the closest value above the target when no exact match exists.

2 — Wildcard Match

Lets lookup_value use * and ? wildcards, e.g. "Sam*" to match "Samir" or "Samira."

8 Two-Way Lookup

Concrete example: finding one cell inside a full price or sales matrix.

Nesting Two XLOOKUPs

- Outer XLOOKUP finds the correct row, e.g. the product.
- Inner XLOOKUP finds the correct column, e.g. the region, and returns that column's values.
- =XLOOKUP(F2, A2:A10, XLOOKUP(G2, B1:E1, B2:E10))
- F2 is the product to find; G2 is the region; the formula returns the single cell where they intersect.

	North	South	East	West
Widget A	12	15	9	14
Widget B	22	18	**25**	20
Widget C	8	11	7	13

9 Search Modes and Performance

The sixth argument, search_mode, controls the direction and speed of the search.

1	-1	2	-2
default: searches from the first row to the last	searches from the last row to the first — most recent match	binary search, ascending (data must be pre-sorted)	binary search, descending (data must be pre-sorted)

When it matters

- Use -1 to pull the most recent transaction, price, or status update from a log sorted oldest to newest.
- Leave search_mode at its default of 1 for almost every everyday lookup — binary search only helps on very large, pre-sorted datasets.

10 Case Study: Replacing a Payroll VLOOKUP Chain

An HR analyst maintained four nested VLOOKUP formulas to pull an employee's name, department, manager, and pay grade from a roster. Every time HR reordered the roster's columns, every formula broke.

4 → 1

nested VLOOKUP formulas replaced by one XLOOKUP per field

Three changes made the difference:

- Each VLOOKUP(ID, A:F, 3, FALSE) became =XLOOKUP(ID, A:A, C:C) — no column-count to maintain.
- Every formula added a fourth argument, "Not on Roster," instead of showing #N/A.
- Reordering or inserting a roster column no longer required editing a single formula.

Outcome

A roster reorganization that used to cost an afternoon of formula repair became a non-event. New hires and department changes now flow through automatically, and the friendly fallback text made review meetings far less confusing.

11 Common Mistakes and How to Avoid Them

A few habits cause most of the confusion people run into with XLOOKUP.

The Mistake

- Selecting lookup_array and return_array with different numbers of rows, causing a #VALUE! error.
- Leaving out if_not_found and letting a raw #N/A break a downstream formula or chart.
- Copying the formula without locking ranges with $, so the arrays shift on the wrong rows.
- Assuming XLOOKUP is only available in every Excel version — older desktop installs (pre-2019) don't have it.

The Fix

- Select both arrays from the same starting row to the same ending row before writing the formula.
- Always add a fourth argument, even something simple like "" or "N/A," as a habit.
- Use $A$2:$A$20 style references for the arrays, or convert the range to a Table first.
- Check File → Account for the Excel version, or use IFS/INDEX-MATCH as a fallback on shared older files.

12 Building Toward Mastery

Rebuild an Old VLOOKUP

Find a spreadsheet with an existing VLOOKUP and rewrite it as XLOOKUP — notice how much shorter and safer it becomes.

Practice Two-Way Lookups

Build a small price or schedule matrix and practice nesting two XLOOKUPs to pull one exact cell.

Always Add a Fallback

Make if_not_found a habit on every formula, even when you don't expect a missing match.

Convert Ranges to Tables

Format your data as an Excel Table (Ctrl+T) so lookup arrays grow automatically as rows are added.

Key Takeaways

- XLOOKUP searches a range and returns a matching result — in either direction, unlike VLOOKUP.
- Only three arguments are required; if_not_found, match_mode, and search_mode all have safe defaults.
- A fourth argument for missing matches turns a cryptic #N/A into a clear, readable message.
- Nesting two XLOOKUPs unlocks a two-way lookup — one for the row, one for the column.