

How to Sort by Last Name in Excel (With Example)

Authored by
stats writer

November 19, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Sort by Last Name in Excel (With Example)*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=96584>

Sorting data efficiently is a fundamental requirement when managing large datasets, and one common challenge users face in Excel is sorting a column containing full names based specifically on the last name. Unlike programs that inherently recognize name components, Excel treats the full name as a single text string. Therefore, to execute a sort by the final component (the surname), we must first isolate that component into its own dedicated column.

This comprehensive guide details the precise, two-step methodology required to achieve accurate last-name sorting. We will utilize a robust combination of powerful Excel text functions designed to reliably extract the last name, irrespective of whether the full name includes a middle name or initial. Mastery of this technique ensures your datasets remain organized and easily searchable, greatly enhancing your data management workflow.

The core process involves: 1) Creating an adjacent column dedicated to the last name extraction, and 2) Employing the Sort feature to reorganize the data based on this newly calculated column. We will walk through a detailed, step-by-step example to illustrate this advanced sorting technique.

Preparing the Initial Data Structure

Before initiating the sorting process, it is essential to ensure your initial data is correctly formatted. For this procedure to function effectively, all names should be contained within a single column, typically listed as "First Name Last Name" or "First Name Middle Name Last Name." If names are already separated into different columns, the sorting process becomes much simpler, but for the common scenario where full names reside in column A, the following setup is necessary.

Consider the following sample dataset provided below. Our objective is to rearrange these rows so that the individuals are listed in alphabetical order based solely on their surname. Currently, the names are listed in a random or entry-order sequence, which hinders quick referencing and analysis. We must insert a blank column adjacent to the names to accommodate the extracted last names.

For demonstration purposes, let us assume the full names are entered starting in cell **A2**, with a header row in **A1** (e.g., "Full Name").

	A	B	C	D
1	Name			
2	Andy Bernard			
3	Bob Ericson			
4	Chad Anderson			
5	Doug Phelps			
6	Eric Green			
7	Frank Smith			
8	George Harrison			
9	Mr. Henry Toms			
10	Isaac King			
11	Jim Burns Randolph			
12				
13				
14				
15				
16				
17				
18				

It is recommended that you label the new column--for instance, in cell **B1**--as "Last Name." This practice not only enhances readability but is crucial later when utilizing the custom sorting dialog box, as Excel relies on column headers to identify sorting criteria.

Developing the Formula for Last Name Extraction

The crucial technical step in this procedure involves crafting a powerful formula that can reliably identify and isolate the final word in the full name string. This formula must account for variable name lengths, including the presence or absence of middle names, which complicate simple split functions.

We will construct a nested expression utilizing several core Excel text functions: RIGHT function, LEN function, FIND function, and SUBSTITUTE function. The logic hinges on identifying the final space within the text string and calculating the number of characters that follow it. By replacing the last space with a unique character, we can pinpoint its location and then use that position to instruct the RIGHT function on how many characters to pull from the end.

The specific formula to be entered into cell **B2** (assuming A2 contains the first full name) is provided below. This formula is complex but highly effective:

```
=RIGHT(A2,LEN(A2)-FIND(" ",SUBSTITUTE(A2," ", "*",LEN(A2)-LEN(SUBSTITUTE(A2," ", ""))))))
```

A Detailed Breakdown of the Name Extraction Logic

Understanding how this nested formula works is critical to troubleshooting and adaptation. The formula operates from the inside out, performing a series of calculations to locate the boundary between the first/middle names and the last name.

The innermost core determines how many spaces exist in the full name string. This is achieved by comparing the total length of the string to the length of the string after all spaces have been removed: **LEN(A2)-LEN(SUBSTITUTE(A2," ", ""))**. If a name is "John A. Smith," this subtraction yields 2 (two spaces).

The next layer, **SUBSTITUTE(A2," ", "*",...)**, is the genius behind the extraction. It uses the count calculated previously to replace only the Nth occurrence of the space (where N is the total number of spaces) with a unique placeholder, in this case, an asterisk (*). This guarantees that the space immediately preceding the last name is converted, regardless of whether there are one or multiple preceding names.

The **FIND function** then searches for the position of that asterisk. Once the asterisk's position is known, the final calculation takes place: **LEN(A2)-FIND(" ", ...)**. This arithmetic determines the length of the string starting immediately after the final space (the asterisk). This length is precisely the number of characters representing the last name.

Finally, the entire calculation is passed to the **RIGHT function**, which extracts exactly that calculated number of characters from the end of the original text string in **A2**, thus successfully isolating the last name into column B.

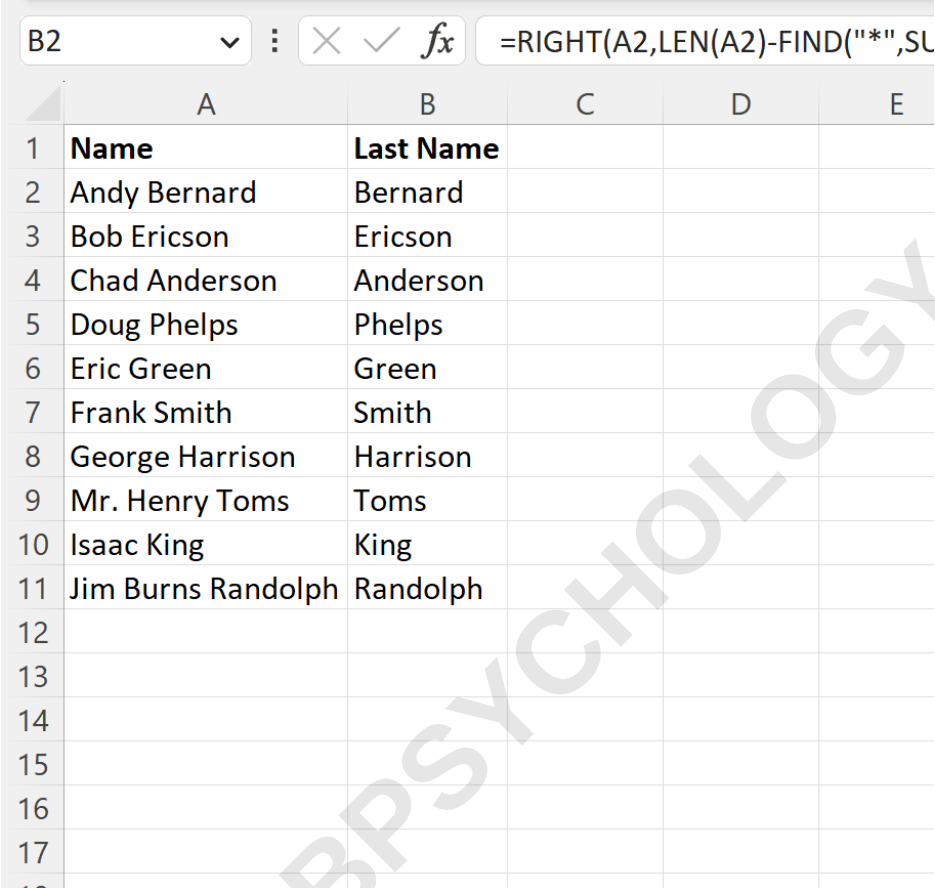
Applying the Formula and Reviewing Results

Once the extraction formula has been accurately entered into cell **B2**, the next action is to apply it consistently across the entire dataset. This is typically accomplished using the "Fill Handle" feature in Excel. Click on cell **B2**, locate the small green square (the Fill Handle) in the bottom-right corner of the cell, and double-click it. This action automatically copies the formula down to every corresponding row where data exists in column A.

The resulting column B should now contain only the last names corresponding to the full names in column A. It is vital to visually inspect the results, especially for complex names or names containing unusual spacing, to ensure the formula has functioned correctly. The robustness of this specific nested formula is demonstrated by its ability to handle names with varying numbers of

components--whether it is a simple first/last name pairing or a longer string including middle initials or second middle names.

As illustrated in the image below, column B now stands ready as the definitive key for our sorting operation. The temporary nature of this column allows us to perform complex sorting operations that are otherwise impossible with the original single-column format.



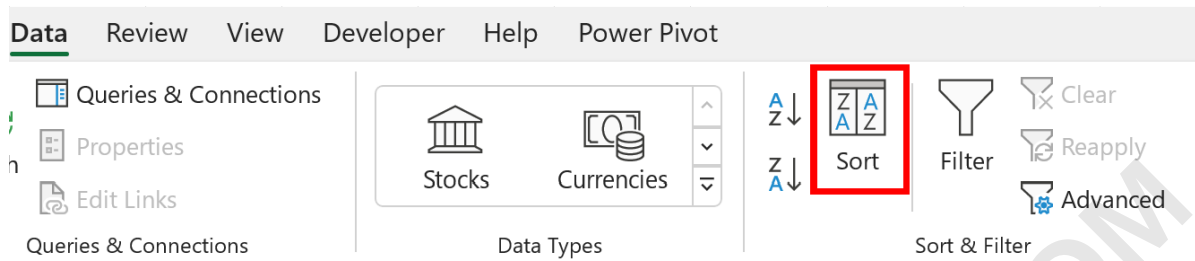
	A	B	C	D	E
1	Name	Last Name			
2	Andy Bernard	Bernard			
3	Bob Ericson	Ericson			
4	Chad Anderson	Anderson			
5	Doug Phelps	Phelps			
6	Eric Green	Green			
7	Frank Smith	Smith			
8	George Harrison	Harrison			
9	Mr. Henry Toms	Toms			
10	Isaac King	King			
11	Jim Burns Randolph	Randolph			
12					
13					
14					
15					
16					
17					

Executing the Custom Sort Based on Extracted Data

With the last names isolated in column B, we can now proceed to the actual sorting process. This requires using Excel's sophisticated Custom Sort dialog box, which allows sorting across multiple columns simultaneously, ensuring that the first name remains correctly associated with the last name during the movement of rows.

The first step is to select the entire range of data that needs to be sorted. In our example, this range spans from **A1:B11**, encompassing both the full name column and the newly created Last Name column, including their respective headers. Selecting the entire range prevents data misalignment, which is a critical error where names shift but their corresponding last names do not.

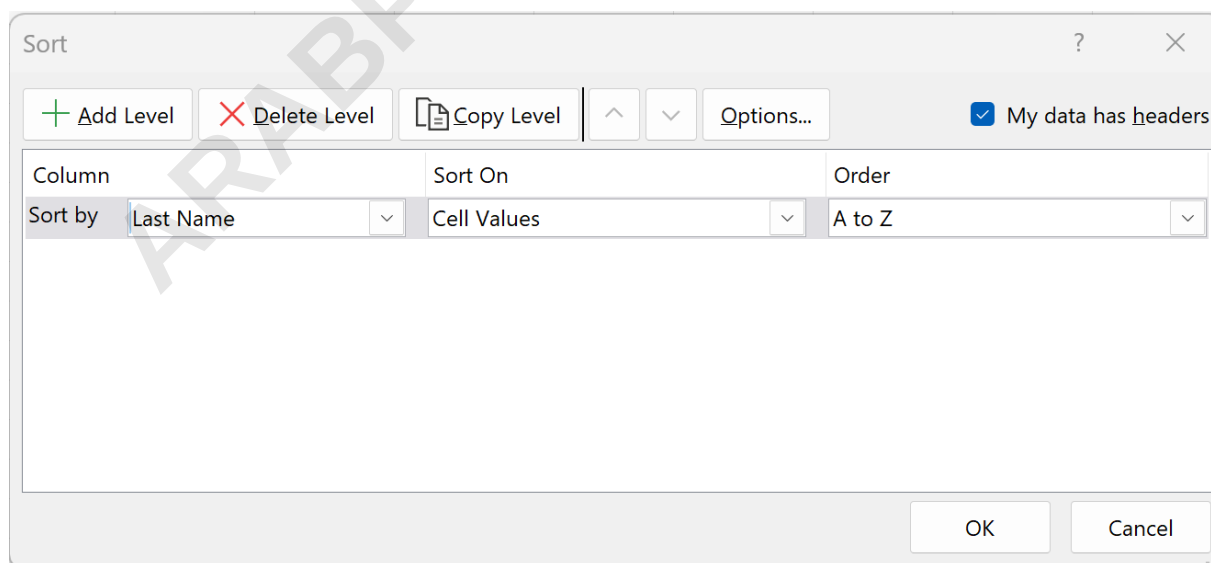
Navigate to the **Data tab** located on the Excel Ribbon. Within the **Sort & Filter** group, click the **Sort feature** icon. This action opens the custom sorting window, which provides comprehensive control over the sorting criteria.



Configuring and Finalizing the Sort Criteria

Inside the Custom Sort window, several options must be carefully configured. Since we selected the header row (A1 and B1) alongside the data, ensure that the option labeled "My data has headers" is checked. This tells Excel to use the text in the first row as criteria labels rather than attempting to sort them as data points.

The primary control in this dialog is the **Sort by** dropdown menu. Click on this menu and select the column header titled **Last Name** (or whatever label was assigned to column B). This explicitly designates the extracted surname as the key upon which the entire dataset will be reorganized. For the **Sort On** setting, select "Values," and for the **Order** setting, typically choose "A to Z" for standard alphabetical sorting.



After confirming these settings and clicking **OK**, Excel will execute the sort operation. All rows

within the selected range (A1:B11) will be instantly repositioned based on the calculated values in the Last Name column. This final arrangement confirms that the list is now logically organized by surname, fulfilling the core requirement of the task.

Final Output and Optional Cleanup

The successful execution of the Sort feature yields a rearranged table where the full names in column A are now ordered alphabetically according to the last names extracted in column B. Note that the entire row associated with each name has moved together, preserving data integrity.

	A	B	C	D	E
1	Name	Last Name			
2	Chad Anderson	Anderson			
3	Andy Bernard	Bernard			
4	Bob Ericson	Ericson			
5	Eric Green	Green			
6	George Harrison	Harrison			
7	Isaac King	King			
8	Doug Phelps	Phelps			
9	Jim Burns Randolph	Randolph			
10	Frank Smith	Smith			
11	Mr. Henry Toms	Toms			
12					
13					
14					
15					
16					
17					
18					

Once the desired sorting is complete, the column titled **Last Name** (column B) often serves no further purpose if the goal was simply rearrangement. If you do not require a separate column displaying only the surnames, it is safe to delete column B entirely. However, if the intent is to preserve the ability to re-sort quickly in the future, or if the extracted last name is needed for other analytical tasks (like filtering or pivot table creation), retaining the column is advisable.

Furthermore, if you intend to delete column B, it is highly recommended that you first convert the formulas in column B into static values. To do this, copy column B, then use the "Paste Special" command to paste the data back into the same column as "Values." This ensures that the extracted text remains intact even if the original name column (Column A) is later modified or deleted, preventing potential reference errors (like #REF!) caused by the dependence of the

formula on the original text string.

Summary of Steps and Alternative Methods

While the formulaic extraction method is robust and universally applicable across different versions of Excel, modern versions offer simpler alternatives, particularly if your data structure is consistent (e.g., always "First Last").

For users of Excel 2013 and later, the **Flash Fill** feature, found under the **Data tab**, provides a rapid, non-formulaic way to extract data based on patterns. To use Flash Fill, simply manually type the last name of the first entry (e.g., type "Smith" in B2 if A2 is "John Smith"). Then, start typing the second last name in B3. Excel detects the pattern and offers to auto-complete the rest of the column, achieving the same result as the complex formula, but without the dependency risks.

However, it is important to remember that Flash Fill is pattern-based and may struggle or fail entirely if the data quality is highly inconsistent (e.g., some names have suffixes, others have titles). For guaranteed accuracy across messy datasets, the nested **RIGHT function** **LEN function** approach remains the industry standard for programmatic data cleansing and preparation prior to using the Sort feature.