

How to Sort Data by Length in Google Sheets: A Step-by-Step Guide

Authored by
stats writer

January 30, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Sort Data by Length in Google Sheets: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128599>

Advanced Data Management: Sorting by String Length in Google Sheets

Introduction: Sorting Data by Character Count in Google Sheets

Google Sheets stands out as robust spreadsheet software, providing users with an extensive array of tools designed for organizing, manipulating, and analyzing complex data. While standard alphabetical or numerical sorting is common practice, a less frequently utilized but highly powerful feature is the ability to sort records based on the intrinsic length of their textual content. This capability is crucial for managing and analyzing specialized datasets where the length of a string, such as a product title, URL, or identifier, holds analytical significance.

To effectively sort data by length within Google Sheets, you typically leverage a combination of built-in functions, most notably the LEN function. This function calculates the exact number of characters in a given cell, transforming a textual property into a numerical metric that can then be used as the key for sorting. Although the specialized SORT function can perform this operation in a single formula (e.g., `=SORT(A:A, LEN(A:A), TRUE)`), the manual, step-by-step approach detailed below offers greater visibility, control, and accessibility for users dealing with complex ranges or those who prefer graphical interface sorting.

The core principle involves generating a temporary, adjacent column that exclusively contains the length of the strings in the source column. Once this numerical length column is established, standard sorting mechanisms can be applied to this derived column, thereby indirectly sorting the original textual data by its character count. This technique proves invaluable in scenarios requiring the organization of names, titles, or code snippets based on brevity or verbosity.

The Power of the LEN Function

The cornerstone of sorting by string length in any spreadsheet environment is the LEN function. This function is straightforward: it accepts a single text string or a reference to a cell containing a string and returns a numerical value representing the total number of characters, including letters, numbers, spaces, and punctuation. Understanding the precision of this function is vital for accurate sorting.

For instance, if a cell contains the phrase "New York," the LEN function will return 8, counting the three letters in "New," the space, and the four letters in "York." This numerical output is what allows the sorting engine of Google Sheets to process text based on magnitude rather than alphabetical order.

By implementing the LEN function across an entire column, we essentially create a quantitative

map of our qualitative data. This derived numerical map is temporary but essential, acting as the bridge between the textual data we wish to sort and the numerical logic required by the sorting algorithms available within the application interface.

Step-by-Step Guide: Preparing Your Data

Before initiating the calculation, it is necessary to properly structure your data set. For the purpose of this example, we will use a list of names, specifically those of various basketball teams. This step ensures that the source data is clearly defined and ready for the application of the length calculation formula in an adjacent column.

We begin by inputting the sample data into Column A. This column, labeled "Team," contains the textual strings we intend to sort. It is critical that the data is clean and consistent at this stage, as any trailing spaces or hidden characters will be counted by the LEN function, potentially skewing the sorting results.

The structure should resemble the layout shown in the image below, with the team names occupying the rows starting from A2 downwards. Ensure that headers are distinct and clearly label the source data.

Step 1: Enter the Data

First, let's enter the names of various basketball teams in Google Sheets:

	A	B	C	
1	Team			
2	Mavs			
3	Warriors			
4	Hawks			
5	Kings			
6	Lakers			
7	Nets			
8	Magic			
9	Timberwolves			
10	Clippers			
11	Pacers			
12				
13				
14				
15				
16				

Calculating String Lengths Using LEN

The next logical step is to utilize the [LEN function](#) to generate the corresponding numerical length for each team name. We must dedicate a new column adjacent to the source data--in this case, Column B--to store these calculated lengths. We will label this column "Length" for clarity.

The formula implementation starts in the first data row of the new column. If our team names begin in cell A2, the length calculation must begin in cell B2, referencing the string in A2. This establishes the initial link between the text and its quantitative measure.

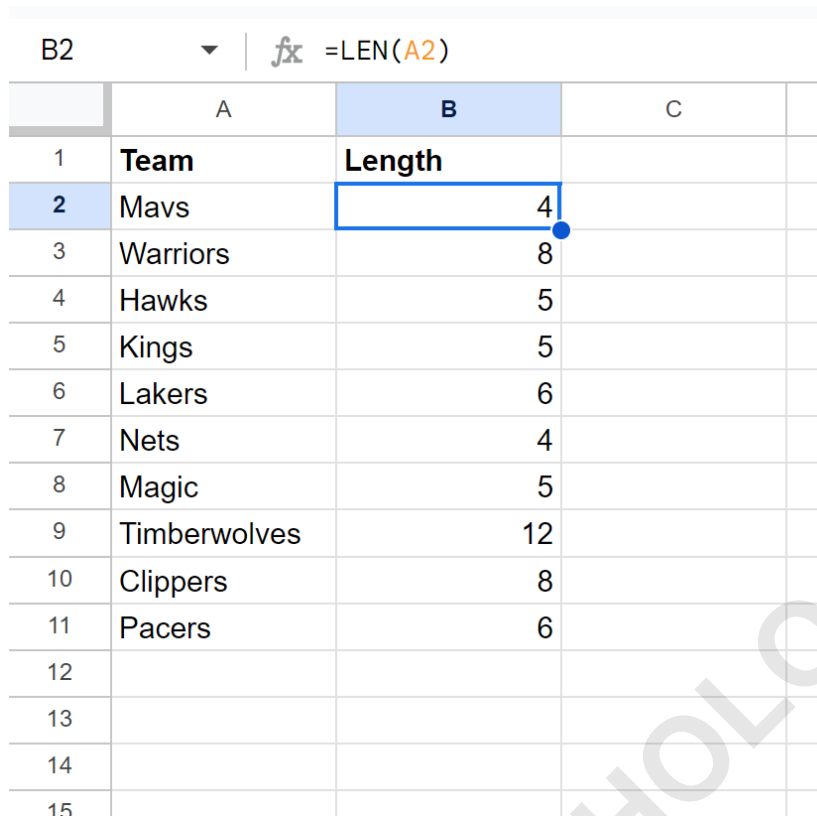
Once the formula is entered in the initial cell (B2), the efficient method for applying it to the rest of the dataset is through autofill. By clicking and dragging the fill handle (the small square at the bottom-right corner of the selected cell) down the column, the formula automatically adjusts its cell reference (A2 changes to A3, A4, and so on), quickly calculating the length for every corresponding team name.

Step 2: Calculate Length of Each String

Next, we will type the following formula into cell **B2** to calculate the length of the string in cell **A2**:

=LEN(A2)

We can then click and drag this formula down to each remaining cell in column B:



The screenshot shows a Google Sheets spreadsheet with the following data:

	A	B	C
1	Team	Length	
2	Mavs	4	
3	Warriors	8	
4	Hawks	5	
5	Kings	5	
6	Lakers	6	
7	Nets	4	
8	Magic	5	
9	Timberwolves	12	
10	Clippers	8	
11	Pacers	6	
12			
13			
14			
15			

The **Length** column displays the numerical character count in each corresponding string in the **Team** column.

Advanced Sorting Techniques: Utilizing the Data Range

With the numerical lengths now calculated and stored in Column B, the dataset is ready for the actual sort operation. It is paramount that when performing this sort, the entire data range--encompassing both the original text strings and their corresponding lengths--is selected. This ensures that when the numerical values in Column B are rearranged, the associated textual entries in Column A move with them, maintaining data integrity.

In our example, we must select the entire active cell range containing the data, which is **A2:B11**. Including the headers (A1:B1) is often optional, but if included, ensure the "Data has header row" option is checked during the sorting process to prevent the header row from being sorted along with the data. Following the selection, the process moves to accessing the specialized sorting menu within Google Sheets.

This step requires navigating to the main menu ribbon, specifically the **Data** tab. While a simple "Sort range" option exists, using the "Advanced range sorting options" provides the necessary

control to specify which column acts as the primary sort key--a crucial distinction when sorting by a calculated metric like string length.

Executing the Sort Operation

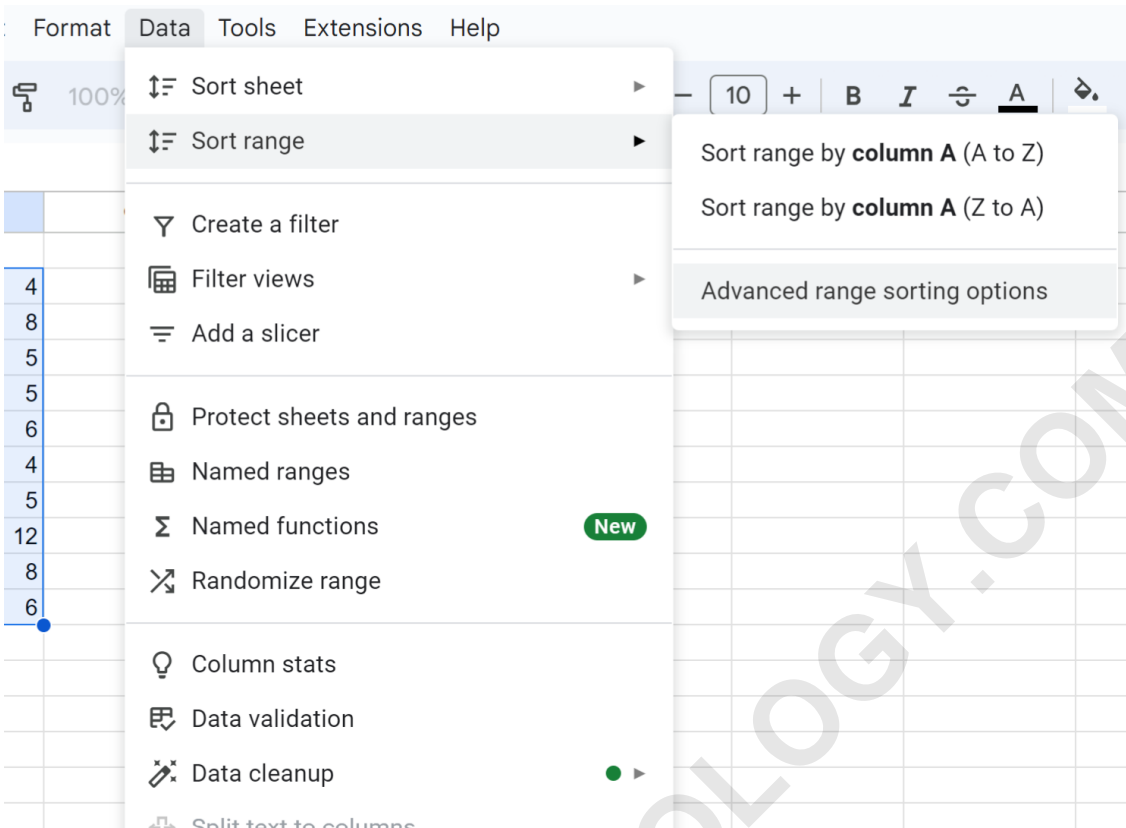
Once the advanced sorting menu is activated, the system requires input regarding the sorting criteria. Since we aim to sort the team names based on their length, we must designate the "Length" column (Column B) as the key column. This is where the preparation work pays off; the textual data is now effectively sorted numerically.

For sorting from shortest string length to longest string length (ascending order), the criterion should be set to sort Column B from A to Z (which translates to lowest number to highest number). Conversely, if the requirement is to sort from longest to shortest, the Z to A option should be selected. After confirming the criteria, clicking the **Sort** button executes the reorganization of the data range.

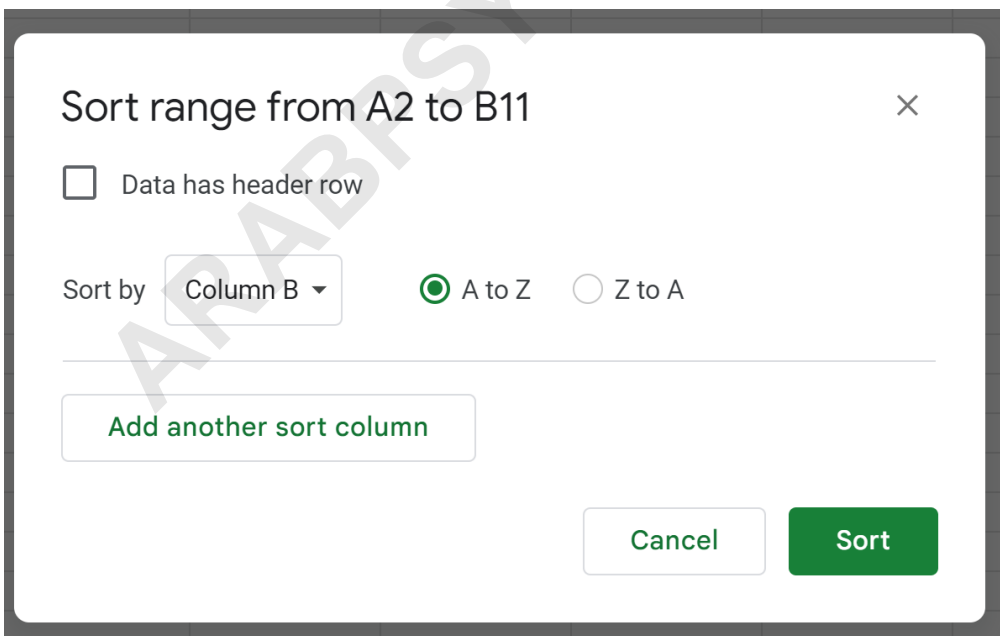
Upon execution, the rows will be instantly reordered based entirely on the numerical values in Column B. The original team names in Column A will move synchronously with their calculated lengths, resulting in a fully sorted list where the shortest names appear first (or last, depending on the chosen direction).

Step 3: Sort Strings by Length

Next, highlight the cell range **A2:B11**, then click the **Data** tab along the top ribbon, then click **Sort range**, then click **Advanced range sorting options**:



In the new window that appears, choose to **Sort by** Column B from A to Z, then click **Sort**:



The resulting table demonstrates the team names sorted accurately by their length, ranging from the shortest entry to the longest:

	A	B	C	D
1	Team	Length		
2	Mavs	4		
3	Nets	4		
4	Hawks	5		
5	Kings	5		
6	Magic	5		
7	Lakers	6		
8	Pacers	6		
9	Warriors	8		
10	Clippers	8		
11	Timberwolves	12		
12				
13				
14				
15				
16				

Important Notes and Data Cleanup

After successfully sorting the data, there are several practical considerations for maintaining a clean and efficient spreadsheet. These notes address the flexibility of the sorting direction and the necessity of the temporary length column.

Sorting Direction: The example demonstrated sorting in ascending order (A to Z), which means shortest strings first. If the requirement is to view the longest strings first, simply select **Sort by Z to A** in the advanced sorting options window. This reverses the numerical order, placing the highest character count values at the top.

Data Cleanup: The "Length" column (Column B) was purely instrumental for the sorting process. Once the sorting is complete and the textual data is organized as required, this auxiliary column is no longer necessary for the primary dataset and can be safely deleted. This practice maintains a clean worksheet focused only on the essential data columns.

Alternative Method: Using the SORT Function (Formulaic Approach)

While the step-by-step method using an auxiliary column is intuitive and excellent for visual confirmation, advanced Google Sheets users can achieve the same result using a single array formula, which avoids the need for creating and managing an extra column. This approach utilizes

the [SORT function](#) in combination with the [LEN function](#).

The syntax for this powerful combination is structured to feed the lengths directly into the sort criterion without displaying them separately. If the list of names is in Column A, the formula would be placed in a new, empty [cell range](#) (e.g., C1) and would look like this: `=SORT(A:A, LEN(A:A), TRUE)`.

In this formula: `A:A` specifies the range to be sorted; `LEN(A:A)` dynamically generates an array of lengths corresponding to every cell in Column A, which the [SORT function](#) uses as the sort key; and `TRUE` dictates an ascending sort (shortest to longest). This method is highly efficient for recurring tasks and dynamic [datasets](#), as the output automatically updates whenever the source data changes.