Learning Google Sheets: How to Remove the Last 3 Characters from a Text String

Authored by

Mohammed looti

November 11, 2025

When dealing with large datasets in Google Sheets, the need for precise data cleaning operations is paramount. A common requirement in data preparation is standardizing text entries, often called strings, by removing unwanted characters. Whether you are stripping metadata, removing identification codes, or truncating non-essential suffixes, you need a robust and scalable method to handle these manipulations across thousands of rows.

This expert tutorial provides a detailed, highly efficient solution specifically designed to remove the final three characters from any text string within your spreadsheet. Our technique hinges on the powerful synergy between two fundamental spreadsheet functions: the LEFT function and the LEN function. Understanding how to combine these tools allows you to create dynamic formulas that automatically adapt to strings of varying lengths, ensuring consistency and accuracy across complex datasets.

The core strength of this approach lies in its mathematical calculation of required length. Instead of relying on rigid, fixed-position extraction methods, the combination dynamically calculates the total length of the original string, subtracts the characters to be discarded (three, in this case), and then instructs the LEFT function to extract exactly the remaining portion. This process is crucial for anyone performing serious data preparation or analysis in a spreadsheet environment.

The Core Formula: Combining LEFT and LEN

To successfully truncate the end of a string by a fixed number of characters, we must first instruct Google Sheets to calculate the total character count and then subtract the desired truncation length (3). The result of this subtraction serves as the precise length argument for the LEFT function, dictating exactly how many characters should be retained from the beginning of the text.

The specific structure for this dynamic calculation is highly efficient. It utilizes the LEFT function as the primary extractor, relying on the nested LEN function to provide a dynamic length argument. This ensures that the formula is versatile and works correctly whether the original text is five characters long or fifty.

The formula below illustrates the required syntax. We assume the source text is located in cell A2:

=LEFT(A2,LEN(A2)-3)

Consider an example where cell **A2** contains the text "Example-123". The inner component, `LEN(A2)`, first calculates the total length, which is 11. Subtracting 3 from this result yields 8. Consequently, the outer function executes as `LEFT(A2, 8)`, extracting the first eight characters ("Example-") and successfully removing the unwanted suffix ("123"). This methodology removes the need to manually verify the length of every string, making it perfect for automating large-scale data manipulation.

Practical Application: Cleaning Trailing Suffixes

To fully appreciate the utility of this technique, let's explore a frequent data analysis task: standardizing a list of identifiers that consistently include an unwanted three-character suffix. Imagine you have a list of product codes or team names in Column A, each appended with a trailing numerical indicator like "-01" or "XYZ." Our objective is to clean this list, retaining only the fundamental identifier.

Our initial dataset, which contains these appended codes, is stored across cells A2 through A5, as shown below. Notice the varying lengths of the core names, which highlights why a dynamic solution like `LEFT(LEN())` is superior to fixed-position formulas.

	A	B	C
1	**Team**		
2	Mavericks		
3	Thunder		
4	Spurs		
5	Rockets		
6	Timberwolves		
7	Knicks		
8	Grizzlies		
9	Hawks		
10	Celtics		
11	Lakers		
12	Warriors		
13			
14			
15			

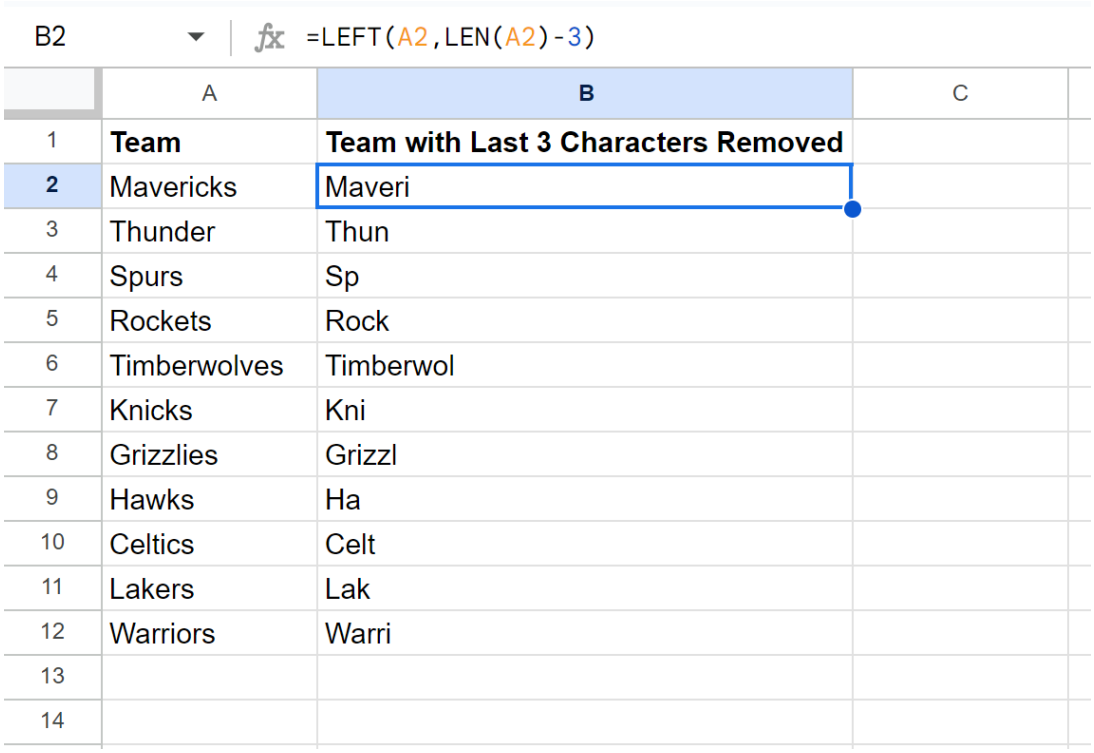
We aim to remove the consistent three-character suffix from each entry. We apply the combined LEFT and LEN formula into cell **B2** to process the first entry, referencing the data in A2.

The formula entered into **B2** is identical to our core solution:

=LEFT(A2,LEN(A2)-3)

Once the result is confirmed in **B2**, the power of relative referencing in Google Sheets comes into play. By dragging the fill handle down column B, the formula automatically adjusts (A2 becomes

A3, A4, and A5), applying the necessary truncation to every corresponding source string in Column A. This action instantly cleanses the entire list, providing standardized results, as demonstrated in the output below.



	A	B	C
1	**Team**	**Team with Last 3 Characters Removed**	
2	Mavericks	Maveri	
3	Thunder	Thun	
4	Spurs	Sp	
5	Rockets	Rock	
6	Timberwolves	Timberwol	
7	Knicks	Kni	
8	Grizzlies	Grizzl	
9	Hawks	Ha	
10	Celtics	Celt	
11	Lakers	Lak	
12	Warriors	Warri	
13			
14			

The resulting Column B now contains a clean list of identifiers, having successfully removed the fixed trailing three characters from each string. This dynamic calculation is a foundational skill for maintaining clean, actionable data tables ready for further analysis.

Deep Dive into Functionality: Understanding LEFT() and LEN()

A thorough mastery of text manipulation requires understanding the specific mechanics of the functions involved. The effectiveness of the combined formula `=LEFT(A2, LEN(A2)-3)` is entirely dependent on the precise roles that the LEFT function and the LEN function play in conjunction.

The LEFT function is an extraction tool. Its purpose is to retrieve a specified number of characters, beginning from the leftmost position of a given text string. Its syntax structure is `LEFT(string, )`. When used alone, the user must supply a fixed number for the character count. However, in our combined formula, the dynamic calculation `LEN(A2)-3` replaces that fixed number, making the extraction count variable based on the input text.

The LEN function (short for Length) is the calculation engine. It simply returns the total count of characters within a specified text input. Crucially, its output includes all elements: letters, numbers,

symbols, and perhaps most importantly, any hidden or trailing **blank spaces**. The syntax is straightforward: `LEN(text)`. This raw character count is the essential starting point from which we subtract the characters we wish to discard.

By nesting `LEN(A2)-3` within the LEFT function, we execute what is known as **dynamic string extraction**. The internal calculation determines the exact, desired length of the resulting string, and the external function performs the extraction. This powerful mechanism ensures that irrespective of the overall length of the source text, the final three characters are always excluded, yielding a consistent and predictable result.

Handling Edge Cases: Short Strings and Whitespace

While the `LEFT(LEN())` method is highly effective, real-world data is often messy, introducing potential pitfalls that can lead to errors. Expert users must anticipate and manage two primary edge cases: the presence of unintended whitespace and the manipulation of strings that are too short for the operation. Addressing these ensures reliable data manipulation within Google Sheets.

A frequent issue encountered during data import is the inclusion of unintended leading or trailing spaces. Since the LEN function counts every character, including these spaces, trailing whitespace might be removed instead of visible text. To guarantee that the calculation operates only on clean text, the best practice is to incorporate the `TRIM function`. The `TRIM()` function automatically removes all leading, trailing, and repeated interior spaces, standardizing the string before its length is calculated or extraction begins.

The robust formula incorporating whitespace removal should be structured to apply `TRIM()` to the reference cell both inside the LEN calculation and in the main LEFT argument:

=LEFT(TRIM(A2), LEN(TRIM(A2))-3)

The second critical edge case involves short strings. If the string in cell A2 contains only "AB" (length 2), the calculation `LEN(A2)-3` results in -1. Attempting to extract a negative number of characters with the LEFT function will result in an error (typically #VALUE!). To gracefully manage mixed data where some strings might be shorter than the truncation count, we use the conditional `IF()` function.

The following formula provides an error-proof structure by first checking the length of the string. If the length is greater than 3, it proceeds with the truncation; otherwise, it returns the original content unchanged, thereby maintaining data integrity:

=IF(LEN(A2)>3, LEFT(A2, LEN(A2)-3), A2)

Alternative Methods for String Truncation

While the `LEFT(LEN())` method remains the gold standard for reliable, fixed-length truncation, Google Sheets provides alternative functions that excel when dealing with more complex or pattern-based data cleaning requirements. Understanding these options allows you to choose the most appropriate tool for your specific dataset.

For spreadsheet users proficient in regular expressions, the `REGEXREPLACE function` offers a highly flexible and elegant approach. This function allows you to define a specific pattern to match and remove characters, rather than relying on a fixed count. This is especially valuable if the characters you need to remove are defined by criteria (e.g., "any numbers at the end") rather than just position.

To achieve the removal of the last three characters using `REGEXREPLACE`, we employ a regular expression pattern that identifies any three characters (`.{3}`) located precisely at the end of the text string (indicated by the dollar sign anchor, `$`). The replacement value is set to an empty string ("").

=REGEXREPLACE(A2, ".{3}$", "")

This formula is concise and powerful, offering a sophisticated alternative to the length calculation method. While slightly steeper in its learning curve due to the required knowledge of regular expression syntax, it is unmatched for versatility in complex data transformation workflows.

Summary of Techniques and Best Practices

Effective string manipulation is a cornerstone skill for any user leveraging spreadsheets for data analysis and processing. We have confirmed that the core method--combining the dynamic calculation of LEN and the precise extraction of LEFT--provides the most dynamic and reliable solution for removing a fixed number of trailing characters.

The foundational formula, `=LEFT(A2, LEN(A2)-N)`, where **N** represents the count of characters to be discarded, should be the standard approach for fixed truncation. Furthermore, integrating conditional logic using `IF()` and data normalization using `TRIM()` are essential steps to ensure resilience against common data quality issues, such as short strings or errant whitespace.

By mastering these functions, you gain the ability to significantly streamline your data cleaning workflow, whether you opt for the standard `LEFT(LEN())` method for simplicity or the advanced `REGEXREPLACE` approach for pattern-based complexity. Choosing the correct function based on the data's nature will dramatically improve the accuracy and efficiency of your analysis.

The following resources provide additional tutorials on essential text manipulation tasks in Google Sheets: