

# Google Sheets: How to remove first 3 characters from a string

Authored by  
**stats writer**

November 18, 2025

## RECOMMENDED CITATION

stats writer (2025). *Google Sheets: How to remove first 3 characters from a string*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=95480>

In the realm of spreadsheet management and data manipulation, cleaning and standardizing text entries is a fundamental requirement. One common task involves stripping unnecessary prefixes or identifiers from a text string. Specifically, this guide focuses on mastering the technique for removing the initial 3 characters from a cell's contents within Google Sheets. While this action seems simple, executing it efficiently requires combining powerful built-in functions to ensure accuracy and scalability across thousands of rows of data.

This technique is indispensable when dealing with datasets that have inherited fixed-length codes, date prefixes, or numbering schemes that are no longer needed for analysis. Instead of manually editing each cell, which is prone to error and time-consuming, we leverage the power of formulaic automation. We will thoroughly explore the most robust method using a combination of the RIGHT function and the LEN function, offering a solution that dynamically adapts to strings of varying lengths, provided only the initial characters are targeted for removal.

## The Core Solution: Combining RIGHT and LEN Functions

To effectively remove a fixed number of characters from the beginning of a text string, the most reliable method in Google Sheets involves calculating the total length of the string and then extracting a subset of characters from the right side. This calculation is achieved by nesting the LEN function inside the RIGHT function. This approach guarantees that regardless of how long the original text is, the resulting output will be accurate, avoiding the pitfalls associated with methods that rely on matching specific character patterns.

The standard syntax for implementing this solution, where we aim to remove the first three characters from the content in cell **A2**, is highly efficient and remarkably straightforward once the logic is understood. The RIGHT function requires two arguments: the text string itself, and the number of characters you wish to retain starting from the right-hand side. By subtracting the number of characters we want to remove (in this case, 3) from the total length of the string (as determined by the LEN function), we derive the exact count of characters that must be preserved.

This is the required formula structure for removing the initial three characters:

**=RIGHT(A2,LEN(A2)-3)**

This specific construction dictates that Google Sheets first calculates the total character count of the input in **A2**, then reduces that count by 3, and finally returns the corresponding number of characters from the end of the text. This guarantees the removal of exactly the first three characters, leaving the remainder of the string intact and ready for subsequent analysis or reporting tasks.

## Practical Example: Removing Prefixes from Datasets

To illustrate the practical application of the RIGHT function and LEN function combination, let us consider a common scenario encountered in data manipulation: cleaning up a list of items where an arbitrary, but fixed, three-character prefix needs to be eliminated. Suppose we are working with a list of basketball team identifiers where the first three characters represent an internal categorization code which is no longer relevant to our current project, and we only require the actual team name.

We begin with the following dataset displayed in Column A of our Google Sheets spreadsheet. Notice that the prefix is consistently three characters long across all entries:

	A	B	C	D
1	<b>Team</b>			
2	Mavericks			
3	Hawks			
4	Grizzlies			
5	Thunder			
6	Lakers			
7	Clippers			
8	Rockets			
9	Magic			
10	Knicks			
11				
12				
13				
14				
15				
16				
--				

Our objective is to populate Column B with the cleaned team names, having successfully removed the initial three characters from the entries in Column A. We start by applying the derived formula into cell **B2**, referencing the corresponding value in cell **A2**. This single operation will establish the required logic for the entire column.

The formula we input into cell **B2** is:

**=RIGHT(A2,LEN(A2)-3)**

Once the formula is entered into **B2**, we can utilize the powerful drag-and-fill functionality of Google Sheets. By clicking and dragging the fill handle (the small square at the bottom-right corner of cell B2) down the column, the relative cell reference **A2** automatically adjusts to **A3**, **A4**, and so forth, efficiently applying the character removal logic to every entry in the list. The final result clearly demonstrates the successful stripping of the unnecessary three-character prefix from all team names, validating the strength of this method for batch data manipulation.

B2    ▾    |    *fx*    =RIGHT(A2,LEN(A2)-3)

	A	B	C
1	<b>Team</b>	<b>Team with first 3 characters removed</b>	
2	Mavericks	ericks	
3	Hawks	ks	
4	Grizzlies	zzlies	
5	Thunder	nder	
6	Lakers	ers	
7	Clippers	ppers	
8	Rockets	kets	
9	Magic	ic	
10	Knicks	cks	
11			
12			
13			
14			
15			

## Deconstructing the Formula Logic

Understanding the component functions is vital for mastering advanced spreadsheet techniques and ensuring that modifications for different use cases are executed correctly. The overall formula, `=RIGHT(A2, LEN(A2)-3)`, relies heavily on the individual capabilities of the RIGHT function and the LEN function working in tandem. This nested structure is a common pattern in spreadsheet software designed to perform complex conditional or calculated extractions.

The inner core of the formula, `LEN(A2)`, serves a crucial preliminary role. The LEN function calculates the total number of characters present in the specified cell, **A2**, returning a simple numerical integer. For instance, if cell A2 contains "XYZLakers", the LEN function returns 9. This total length is the foundation upon which the extraction count is based, allowing the formula to be highly dynamic and applicable to strings of varying lengths, whether they contain 5 characters or

50.

This calculated length is then immediately modified by subtracting the fixed value, 3, which represents the number of characters we intend to remove from the beginning of the string. Following the example of "XYZLakers" (Length 9), the calculation becomes  $9 - 3 = 6$ . This result, 6, is passed directly to the second argument of the surrounding RIGHT function, instructing it precisely how many characters to return. The RIGHT function then takes the original text, "XYZLakers", and pulls the final 6 characters, yielding the desired output: "Lakers."

## Handling Fixed Prefixes Using the SUBSTITUTE Function

While the RIGHT and LEN combination is ideal for removing the first *N characters* regardless of their content, there are scenarios where the prefixed characters are constant and known, such as a fixed code like "ID-" or "ABC". In these specific cases, the SUBSTITUTE function offers an elegant and arguably cleaner alternative, particularly if the dataset might occasionally contain the prefix elsewhere within the string but we only want to ensure the starting prefix is eliminated.

The general structure for using this approach, assuming we know the prefix is exactly "ABC" and it appears at the start of cell **A1**, is presented as `=SUBSTITUTE(A1, "ABC", "", 1)`. The optional fourth argument, 1, is crucial here; it restricts the replacement operation to only the first occurrence of the target text ("ABC"). If this argument is omitted, the function will replace every instance of "ABC" found within the string, which is typically not the desired behavior when aiming solely to remove a leading prefix.

It is paramount to understand the limitations of the SUBSTITUTE function for this task. It requires the exact prefix to be known beforehand. If the first three characters vary (e.g., they represent a sequential counter or changing date), the SUBSTITUTE method fails entirely, rendering it unsuitable for dynamic character removal. Therefore, while simpler to read, this function is only recommended when dealing with perfectly standardized prefix codes that are guaranteed to be consistent across the entire column requiring data manipulation.

## Advanced Alternative: The MID Function

For users who prefer a more explicit method that defines both the starting point and the number of characters to retain, the MID function provides a powerful alternative to the RIGHT/LEN combination. The MID function extracts a segment of text starting from a specified position and continuing for a specified length. While slightly more complex in construction, it offers unparalleled flexibility if future data manipulation tasks require starting the extraction from a different position.

When applying the MID function to remove the first 3 characters, we must instruct it to begin extracting at the fourth character (position 4) and continue until the end of the string. The syntax

requires three arguments: the text, the starting position, and the number of characters to extract. To find the total remaining length, we again employ the [LEN function](#) and subtract the number of removed characters (3). The formula for removing the first three characters from cell **A2** thus becomes `=MID(A2, 4, LEN(A2)-3)`.

This construction is functionally equivalent to the RIGHT/LEN method, but it forces the user to specify the starting position (4) explicitly, making the intent slightly clearer to some users. The second argument (4) dictates that the extraction must start immediately after the three unwanted characters. The third argument, `LEN(A2)-3`, ensures that the extraction captures every single character from the starting point to the end of the original string, ensuring no data loss occurs when the prefix is stripped away.

## Addressing Common Challenges and Edge Cases

While the formulas discussed provide robust solutions for most character removal tasks, real-world data is often messy, presenting several common challenges that need to be addressed to achieve truly clean results. Two significant issues frequently encountered include the presence of unintended blank spaces and the need for dynamic character removal beyond a fixed count of three.

The most frequent issue is the accidental inclusion of leading blank spaces. [Google Sheets](#), and all text functions, treat blank spaces as valid characters. If a cell contains "Lakers" (with three leading spaces), and you apply `=RIGHT(A2, LEN(A2)-3)`, the result will be "Lakers" (including the leading spaces), which may interfere with subsequent lookups or analyses. To counteract this, it is highly recommended to wrap the entire expression within the **TRIM** function, which removes excess leading and trailing whitespace. The refined, robust formula becomes: `=TRIM(RIGHT(A2, LEN(A2)-3))`. This ensures that the final output is free of extraneous spaces that could corrupt the [data manipulation](#) pipeline.

Another important consideration is the generalization of this method. While we focused on removing 3 characters, the formula is highly adaptable. To remove a different number of starting characters--say, **N** characters--you simply replace the value 3 with the desired number **N**. For example, to remove the first five characters, the formula would be `=RIGHT(A2, LEN(A2)-5)`. This flexibility makes the [RIGHT function/LEN function](#) combination the superior choice for handling variable requirements across different projects, providing a scalable template for character stripping.

## Summary of Techniques

Successful text [data manipulation](#) in Google Sheets relies on choosing the appropriate tool for the job. We have examined the three primary methods for removing characters from the beginning of a

string, each suitable for slightly different data scenarios. Choosing between them depends entirely on whether the number of characters to be removed is fixed or dynamic, and whether the content of those initial characters is known.

The RIGHT function combined with the LEN function (`=RIGHT(A2, LEN(A2)-N)`) is the **recommended default solution**. It is impervious to changes in the content of the removed characters and works reliably for any fixed number N, making it the most robust method for removing a prefix of a fixed length from a dynamic set of data. Conversely, the MID function offers functional equivalence but requires explicit definition of the start position.

Finally, the SUBSTITUTE function (`=SUBSTITUTE(A1, "Prefix", "", 1)`) should be reserved only for highly specific use cases where a known, non-variable prefix needs to be removed. By understanding these distinctions, users can implement professional, clean, and scalable solutions for text cleaning and preparation within their spreadsheet environments.