

How to Remove the First 3 Characters from Text in Google Sheets

Authored by
stats writer

February 21, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Remove the First 3 Characters from Text in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=131968>

The Importance of String Manipulation in Modern Data Workflows

In the contemporary landscape of data management, the ability to refine and restructure information is an essential skill for any analyst. When working within **Google Sheets**, users frequently encounter datasets that require structural adjustments to ensure consistency across various reports. Removing leading characters, such as prefixes, country codes, or unwanted identifiers, is a fundamental task that can be automated using specific **functions**. By mastering these techniques, you ensure that your **data cleaning** process remains efficient, reproducible, and entirely error-free, which is critical when handling large volumes of information.

A **string** in a spreadsheet context is a sequence of characters that can include letters, numbers, symbols, and even white spaces. Often, data imported from external sources like **CSV** files or web databases includes metadata or formatting tags at the beginning of each entry. For instance, a list of product IDs might all start with a three-letter category code that needs to be stripped away for a cleaner lookup. Understanding the mathematical logic behind substring extraction allows users to solve these problems without manual intervention.

The core objective of this guide is to demonstrate how to effectively remove the first three characters from any given text entry. While there are multiple ways to achieve this, combining the **RIGHT function** with the **LEN function** provides a dynamic and robust solution. This approach adapts to varying text lengths, ensuring that the tail end of your data remains intact regardless of how long the original entry was. This flexibility is what makes formulaic manipulation superior to static editing methods.

Dissecting the RIGHT Function for Data Extraction

The **RIGHT function** is a versatile tool designed to return a specific number of characters starting from the end of a **string**. Its syntax is straightforward, requiring two primary arguments: the text you wish to manipulate and the number of characters you want to keep. In isolation, this function is perfect for extracting suffixes, such as the last four digits of a social security number or the file extension of a document name. However, its power is truly unlocked when the second argument is calculated dynamically rather than being a hard-coded integer.

In most **spreadsheet** applications, text functions are zero-indexed or one-indexed depending on the underlying engine, but **Google Sheets** treats character counts as a direct measure of length. When you tell the **RIGHT function** to extract "X" characters, it starts at the very last character and counts backward. If the string length is shorter than the number requested, the function simply returns the entire string. This behavior is important to remember when designing formulas that must handle diverse data types or empty cells.

To use the **RIGHT function** effectively for removing a prefix, we must realize that the number of

characters we want to "keep" is the total length minus the number of characters we want to "discard." This logical shift is what allows a "right-side" extraction tool to function as a "left-side" removal tool. By focusing on what remains after the first three characters are gone, we create a stable path for our data transformation.

Calculating Total Length with the LEN Function

The **LEN function** serves one primary purpose: it returns the total number of characters in a **string**, including spaces, punctuation, and special characters. This is a foundational element in **data cleaning** because it provides the context needed for other functions to operate. Without knowing how long a string is, a formula cannot accurately determine where the "middle" or "end" of the data resides. It is the measuring tape of the spreadsheet world.

When you apply **LEN** to a cell, the result is an integer. For example, if a cell contains the word "Basketball," the function will return 10. If the cell contains "Basketball " (with a trailing space), it will return 11. This sensitivity to every single character is why **Google Sheets** users must be cautious about invisible characters. In our specific use case, **LEN** provides the total "size" from which we will subtract our unwanted three characters.

Combining measurement and extraction allows for automation. Instead of manually counting how many characters are left after removing the first three for every single row, we let the **spreadsheet** perform this calculation instantly for thousands of rows. This is the essence of scalable data management. Whether your team names are five characters long or fifty characters long, the **LEN function** ensures the formula knows exactly how much text to preserve.

The Formula: Combining RIGHT and LEN

To remove the first 3 characters from a string in Google Sheets, you can use the built-in function "RIGHT". This function allows you to extract a specific number of characters from the right side of a string. By specifying a number of characters to extract that is equal to the length of the string minus 3, you can effectively remove the first 3 characters. This method is useful for manipulating and cleaning up data in your Google Sheets spreadsheet.

Google Sheets: Remove First 3 Characters from String

Often you may want to remove the first 3 characters from a **string** in **Google Sheets** to clean up prefixes or identifiers.

You can use the **RIGHT** function combined with the **LEN** function to do so:

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

This particular formula removes the first 3 characters from the string in cell **A2**. It calculates the total length of the text and then instructs the **RIGHT** function to return everything except the first three characters.

The following example shows how to use this formula in practice with a real-world dataset.

Example: Remove First 3 Characters from String in Google Sheets

Suppose we have the following list of basketball team names in our **spreadsheet**:

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

Suppose we would like to remove the first three characters from each team name to isolate the primary name without its preceding code.

We can type the following formula into cell **B2** to do so:

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

We can then click and drag this formula down to each remaining cell in column B to apply the logic across the entire column:

B2 ▾ | **fx** =RIGHT(A2,LEN(A2)-3)

	A	B	C
1	Team	Team with first 3 characters removed	
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			

Column B now displays the team names in column A with the first three characters removed from each team name. The process is instantaneous and handles names of various lengths perfectly.

How This Formula Works

The **RIGHT()** function in **Google Sheets** extracts a specific number of characters from the right side of a string. By nesting other functions within it, we can create dynamic behavior.

The **LEN()** function in **Google Sheets** is used to find the length of a string. It counts every character including spaces.

Thus, our formula displays the entire string with the first three characters removed. The subtraction (Length - 3) tells the **RIGHT** function exactly how many characters are left to keep after the first three are skipped.

Note #1: Blank spaces at the start of a string count as characters. You may need to first remove blank spaces to get your desired result, possibly using the TRIM function.

Note #2: To remove a different number of starting characters from a string, simply change the **3** in the formula to a different value, such as 5 or 10.

Additional Tutorials for Google Sheets Mastery

The following tutorials explain how to perform other common tasks in **Google Sheets** to further enhance your data manipulation skills:

Advanced Text Processing: Handling Leading Whitespace

One common pitfall in **data cleaning** is the presence of leading whitespace. If a cell contains " Team" (with two leading spaces), and you use the formula to remove three characters, you will inadvertently remove the two spaces and the first letter of the actual name. This is because the **LEN function** and the extraction logic count every character, visible or not. To prevent this, it is often wise to wrap your cell reference in a TRIM function before performing the removal.

The TRIM function is designed to remove all leading and trailing spaces, as well as extra spaces between words. By using `=RIGHT(TRIM(A2), LEN(TRIM(A2))-3)`, you create a "sanitized" version of the text before the extraction takes place. This ensures that your "minus 3" calculation is actually removing the data you intended to remove rather than accidental formatting artifacts. This level of detail is what separates basic users from data professionals who build resilient **spreadsheet** systems.

Another consideration is the use of non-breaking spaces or other non-printable characters often found in web-scraped data. These characters might look like normal spaces but have different Unicode values. In such advanced cases, the CLEAN function can be utilized alongside TRIM to ensure that the **LEN** count is strictly based on the visible alphanumeric data you wish to process. Always inspect your data for these hidden elements if your formulas produce unexpected offsets.

Scalability and Array Formulas for Large Datasets

When dealing with thousands of rows, dragging a formula down manually can be tedious and prone to errors if new data is added later. To solve this, **Google Sheets** offers the ARRAYFORMULA function. This allows you to apply the **RIGHT** and **LEN** logic to an entire column at once with a single cell entry. For example, `=ARRAYFORMULA(RIGHT(A2:A, LEN(A2:A)-3))` will automatically process every occupied cell in column A and output the results in the corresponding rows of the current column.

Using array-based processing is a hallmark of efficient **spreadsheet** design. It ensures that if you add a new team name to row 101, the formula in row 101 updates automatically without you having to "fill down" again. This is particularly useful in collaborative environments where multiple people may be inputting data. It also reduces the file's complexity, as you are managing one master formula rather than hundreds of individual ones that might be accidentally edited or deleted.

Furthermore, you can combine these array formulas with logical statements like IF or IFERROR. For instance, you might want to only remove characters if the string is longer than three characters to avoid errors. A formula like `=ARRAYFORMULA(IF(LEN(A2:A)>3, RIGHT(A2:A, LEN(A2:A)-3), A2:A))` provides a safety net. It checks if the **string** has enough characters to be trimmed; if it does, it performs the removal, and if not, it returns the original text. This prevents the formula from returning a "#VALUE!" error when encountering short strings or empty cells.

Alternative Approach: Utilizing REGEXREPLACE

While the **RIGHT** and **LEN** method is the most intuitive for most users, **Google Sheets** also supports **regular expressions** via the REGEXREPLACE function. This is a highly powerful tool for more complex text manipulation. To remove the first three characters using regex, you would use a formula like `=REGEXREPLACE(A2, "^.{3}", "")`. This tells the engine to look at the start of the string (^) and find any three characters (.{3}) and replace them with "nothing" ("").

The advantage of **regular expressions** is their precision. If you only wanted to remove the first three characters if they were specifically digits, you could change the pattern to `^{\d}{3}`. This provides a level of conditional removal that **RIGHT** and **LEN** cannot match without significantly more complex nesting. For power users, learning regex syntax opens up a world of possibilities for **data cleaning** that goes far beyond simple trimming.

However, the downside of regex is its steep learning curve. The syntax can be intimidating for beginners, and debugging a complex regex pattern is often more difficult than troubleshooting a standard arithmetic formula. For the specific task of removing exactly three characters regardless of what they are, the **RIGHT** and **LEN** combination remains the gold standard for readability and ease of maintenance. It is accessible to anyone who understands basic subtraction and spreadsheet functions.

Best Practices for Maintaining Data Integrity

Whenever you perform bulk transformations on a dataset, it is vital to maintain **data integrity**. One of the best practices is to never perform your cleaning in the original "source" column. Instead, always create a "helper column" where your formulas reside. This allows you to compare the original data with the transformed data side-by-side to ensure no mistakes were made during the process. If a formula error occurs, your raw data remains safe and untouched.

Once you are satisfied that the formula has correctly removed the first three characters across your entire dataset, you may want to "freeze" the results. You can do this by selecting the formula results, copying them, and then using "Paste Special" -> "Values only" over the same area or back into the original column. This converts the dynamic formulas into static text, which is useful if you are exporting the data to another **spreadsheet** or a database where Google-specific functions

might not be supported.

Finally, always document your cleaning steps. If you are working in a professional environment, adding a small note or a header explanation about why the first three characters were removed helps future users (or your future self) understand the logic of the **data cleaning** pipeline. Consistency in how you handle these transformations leads to more reliable analysis and more professional-grade reporting in **Google Sheets**.

ARABPSYCHOLOGY.COM