

How to Extract Text After a Character in Google Sheets

Authored by
stats writer

February 18, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Extract Text After a Character in Google Sheets*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=131459>

Efficiently managing and manipulating data within **Google Sheets** is a fundamental skill for modern professionals engaged in **data analysis**. One frequent requirement involves the need to isolate specific segments of a **string** after a particular **delimiter** or character has been identified. While basic functions like **RIGHT** allow for simple extraction based on a fixed number of characters from the right side of a text block, more complex scenarios necessitate the use of advanced **regular expression** logic. By mastering these techniques, users can automate the process of cleaning large datasets, ensuring that relevant information is separated from unnecessary **metadata** or formatting clutter.

Google Sheets: Extract Text After a Character

The Importance of Advanced String Manipulation

In the contemporary landscape of digital information, the ability to parse and organize text dynamically is invaluable. **Google Sheets** provides a robust environment for these tasks, offering a variety of functions designed to handle text strings of varying complexity. When users encounter datasets where information is concatenated--such as names joined with IDs or addresses mixed with zip codes--the standard tools of the **spreadsheet** must be utilized with precision to maintain data integrity. Extracting text after a specific character is a cornerstone of this process, allowing for the transformation of raw input into structured, actionable data.

The **REGEXEXTRACT** function stands out as one of the most powerful utilities available for this purpose. Unlike simpler functions that rely on static positions, **REGEXEXTRACT** utilizes **regular expressions** to identify patterns within the text. This flexibility is essential when the character or word serving as the "anchor" appears at different positions across various cells. By defining a pattern that looks for everything following a specific sequence, users can create dynamic formulas that adapt to the length and structure of individual entries within a column.

Furthermore, employing these advanced functions reduces the likelihood of manual entry errors, which are common when dealing with high-volume data sets. Automating the extraction process ensures consistency across the entire document, which is critical for subsequent **data analysis** or when importing the data into other software systems. Whether you are working with financial records, marketing leads, or scientific observations, the precision offered by **regular expression** patterns in **Google Sheets** provides a level of control that manual editing simply cannot match.

You can use the following formula in Google Sheets to extract all text from a cell after a specific character:

```
=REGEXEXTRACT(A2, ".*our(.*)")
```

Understanding the Formula Syntax and Logic

The **syntax** of the formula provided above is specifically designed to locate a particular sequence of characters and return everything that follows it. In this example, the formula targets cell **A2** and searches for the string "our". The use of the dot and asterisk (**.***) within the regular expression is a key component of its functionality. The first part of the expression, **.***, tells the engine to match any character that appears zero or more times until it reaches the literal string "our". This effectively ignores everything preceding the anchor point.

The second part of the expression, **(.*)**, is enclosed in parentheses, which defines a "capture group" in **regular expression** terminology. While the entire pattern matches the whole string, the function only returns the portion of the text that falls within this capture group. Therefore, by placing the parentheses after the word "our", we instruct **Google Sheets** to discard the match before and including the word "our" and only display the remaining **string**. This subtle distinction is what makes **REGEXEXTRACT** an exceptionally clean tool for data separation.

This particular formula extracts all of the text in cell **A2** that occurs after the string "our" is encountered.

This formula uses the **REGEXTRACT** function to extract all characters **(.*)** after **our**.

The following example shows how to use this formula in practice.

Practical Implementation: A Step-by-Step Guide

To visualize how this works in a real-world scenario, consider a list of phrases or data entries located in column A. Often, these entries will contain a common word or symbol that acts as a logical divider. By identifying this divider, we can apply the **REGEXEXTRACT** function across the entire range to isolate the specific details we require. This method is far more efficient than using the **FIND** and **LEN** functions in combination, as it consolidates the logic into a single, readable **formula**.

Suppose we have the following list of phrases in Google Sheets:

	A	B	C
1	Phrase		
2	Mike is our best player		
3	Chad is our best player I think		
4	Doug is our worst player		
5	Andy is our head coach		
6	Bob is our assistant coach		
7	Greg is our leader		
8	John is our best backup player		
9			
10			
11			
12			
13			
14			
15			
16			

Now suppose that we would like to extract all text from each cell after the string "our" is encountered.

To do so, we can type the following formula into cell **B2**:

```
=REGEXEXTRACT(A2,".*our(.*)")
```

Refining Data with Autofill and Drag Features

Once the initial formula has been established in the first cell of your target column, **Google Sheets** makes it incredibly simple to apply that logic to your entire dataset. By using the fill handle--the small blue square at the bottom-right corner of the selected cell--you can drag the formula down to cover all active rows. The **spreadsheet** software automatically updates the cell references (e.g., changing **A2** to **A3**, **A4**, etc.), allowing the **REGEXEXTRACT** logic to execute individually for every phrase in the list.

This scalability is a hallmark of professional **data analysis** workflows. Instead of writing unique code for every row, a single well-crafted **regular expression** can handle hundreds or thousands of rows instantaneously. This not only saves time but also provides a template that can be reused in future projects. If the underlying data changes or more rows are added, the formula remains valid and continues to provide accurate extractions based on the defined **syntax**.

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

B2 ∇ | fx =REGEXEXTRACT(A2, ".*our(.*)")

	A	B	C
1	Phrase	All Text After "our"	
2	Mike is our best player	best player	
3	Chad is our best player I think	best player I think	
4	Doug is our worst player	worst player	
5	Andy is our head coach	head coach	
6	Bob is our assistant coach	assistant coach	
7	Greg is our leader	leader	
8	John is our best backup player	best backup player	
9			
10			
11			
12			
13			
14			
15			

Column B now displays all text after "our" for each phrase in column A.

Customizing Patterns for Different Delimiters

The versatility of **REGEXEXTRACT** extends beyond simple words like "our". Users can replace the anchor text with any character or sequence that suits their needs. For instance, if you are working with email addresses and need to extract the domain name, you would use the "@" symbol as your anchor. If you are dealing with log files where data is separated by colons or semicolons, you simply update the **regular expression** to reflect these **delimiters**. This adaptability is what makes **REGEXEXTRACT** superior to more rigid functions.

In many cases, the text you wish to extract might follow a specific phrase followed by a space. It is important to include that space within the **formula** if you want the resulting **string** to be clean of leading whitespace. By being precise with the characters included in the match pattern (the part before the parentheses), you control exactly where the "cut" happens. This ensures that the extracted output is ready for immediate use in reports or further calculations without requiring additional trimming or cleaning.

To extract the text after a different specific character, we simply need to replace **our** with

something else.

For example, we could type the following formula into cell **B2** to extract all text after "is " from cell **A2**:

=REGEXEXTRACT(A2,".*is (.*)")

B2 | fx =REGEXEXTRACT(A2,".*is (.*)")

	A	B	C
1	Phrase	All Text After "is "	
2	Mike is our best player	our best player	
3	Chad is our best player I think	our best player I think	
4	Doug is our worst player	our worst player	
5	Andy is our head coach	our head coach	
6	Bob is our assistant coach	our assistant coach	
7	Greg is our leader	our leader	
8	John is our best backup player	our best backup player	
9			
10			
11			
12			
13			
14			
15			
16			

Column B now displays all text after "is " for each phrase in column A.

Handling Special Characters and Troubleshooting

When working with **regular expressions**, it is crucial to understand that certain characters have special meanings within the **REGEXEXTRACT** engine. Characters such as the period, plus sign, question mark, and parentheses are considered "metacharacters." If the character you are trying to use as a delimiter is one of these special symbols, you must "escape" it by preceding it with a backslash. For example, to extract text after a literal period, your pattern would need to include `.` to ensure **Google Sheets** interprets it as a dot rather than a wildcard.

Another common issue involves the **#N/A** error, which occurs when the function cannot find the specified pattern within the target cell. To prevent your **spreadsheet** from looking cluttered with

errors, you can wrap your **REGEXEXTRACT** formula in an **IFERROR** function. This allows you to specify a fallback value, such as an empty string or a "Not Found" message, ensuring that your data remains presentable even when some entries do not meet the extraction criteria. Proper error handling is a best practice that separates amateur work from professional-grade **data analysis**.

Finally, consider the case sensitivity of your search. By default, **regular expressions** in **Google Sheets** are case-sensitive. This means that searching for "Our" will not yield the same results as searching for "our". If your dataset has inconsistent capitalization, you may need to use additional regex flags or the **LOWER** function to standardize the text before performing the extraction. Maintaining a high level of detail in these areas will lead to more reliable results and a more robust **spreadsheet** structure.

Advanced Strategies for Complex Data Extraction

For users who require even more control, **Google Sheets** supports the use of **ARRAYFORMULA** in conjunction with **REGEXEXTRACT**. This allows you to apply the extraction logic to an entire column with a single formula entered into the top cell. This approach is highly recommended for large-scale projects, as it simplifies the management of formulas and ensures that any new data added to the bottom of the sheet is automatically processed without manual intervention.

In addition to extracting text after a single character, **regular expressions** can be configured to extract text between two specific characters or to find the last occurrence of a character in a **string**. By expanding your knowledge of regex patterns, you unlock the full potential of **Google Sheets** as a data-cleansing tool. These skills are highly transferable to other programming languages like **Python** or **JavaScript**, providing a strong foundation for any career path that involves heavy data manipulation.

In conclusion, while the **RIGHT** function offers a basic starting point for text extraction, the **REGEXEXTRACT** function provides the precision and flexibility required for professional tasks. By understanding how to define anchor points and capture groups, you can efficiently parse complex strings and streamline your **data analysis** workflows within **Google Sheets**.

The following tutorials explain how to perform other common tasks in Google Sheets:

How to use the REGEXMATCH function for data validation

Using the SPLIT function to separate text into columns

Combining multiple cells using the JOIN and CONCATENATE functions

Advanced filtering techniques using the QUERY function