

# Learning to Extract Text Before a Character in Google Sheets

Authored by  
**Mohammed loot**

November 10, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Learning to Extract Text Before a Character in Google Sheets*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=15578>

## Introduction to Advanced Text Extraction in [Google Sheets](#)

Data manipulation often requires isolating specific segments of text within a larger cell value. Whether you are dealing with file paths, email addresses, or structured product codes, the ability to extract characters appearing before a specific delimiter or [string](#) is a fundamental skill in spreadsheet analysis. While simple functions like **LEFT** combined with **FIND** can achieve this, they often struggle with complex patterns and are cumbersome to nest. Fortunately, [Google Sheets](#) provides the powerful **REGEXEXTRACT** function, which leverages [regular expressions](#) to handle these tasks efficiently and flexibly.

This guide details how to use **REGEXEXTRACT** to precisely target and extract all text that precedes a specific character sequence or [string](#) within a cell. This approach is superior for its robustness, especially when dealing with inconsistent data formatting or when the delimiter itself might contain special characters. We will explore the specific syntax necessary for this operation, ensuring you gain mastery over one of the most versatile text [functions](#) available in [Google Sheets](#).

The core formula structure is designed to define a capturing group that matches everything up to--but not including--the specified delimiter [string](#). By understanding the components of the [regular expression](#) pattern, you can adapt this method to virtually any text extraction challenge. The following formula represents the foundational structure required to extract all text preceding a chosen [string](#):

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

This specific example formula is engineered to extract all content from cell **A2** that appears before the occurrence of the literal [string](#) "our".

### Understanding the **REGEXEXTRACT Function** and Pattern Syntax

The power of the **REGEXEXTRACT** function lies in its ability to interpret [regular expressions](#). Unlike standard text [functions](#), **REGEXEXTRACT** returns only the parts of the text that are enclosed within parentheses--these are known as [capturing groups](#). To extract the text before a delimiter, we must construct a pattern that defines the desired prefix as the primary [capturing group](#).

Let us break down the expression used in the provided formula: "(.\*)our.\*". This pattern

consists of three critical components that work together to isolate the desired text:

`(.*)`: This is the primary [capturing group](#), defined by the parentheses. The dot (.) matches any single character (except newline), and the asterisk (\*) means "zero or more occurrences" of the preceding element. Combined, `(.*)` means "capture zero or more of any character." This matches the entire prefix of the [string](#) we wish to extract.

`our`: This is the literal delimiter [string](#). The [regular expression](#) engine searches for this exact sequence immediately following the captured group.

`.*`: This component matches everything that follows the delimiter [string](#), ensuring that the entire source cell matches the pattern, which is necessary for the function to execute successfully. Since this part is not within parentheses, it is matched but not returned by [REGEXEXTRACT](#).

By using this configuration, [REGEXEXTRACT](#) is instructed to first find the longest possible match for `(.*)` that still allows the remainder of the [string](#) to match the pattern `our.*`. Because the `(.*)` group is the only [capturing group](#), the function returns only the text that satisfied that initial condition--the text before the delimiter.

## Practical Example 1: Isolating Data Before the [String](#) "our"

To illustrate the application of this technique, consider a common scenario where you have a column of descriptive text, and you need to parse out the leading element based on a consistent internal marker. Suppose we are working with the following collection of phrases listed in column A of our [Google Sheets](#) document:

	A	B	C
1	<b>Phrase</b>		
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			

Our objective is to isolate and extract the text from each cell that occurs immediately preceding the [string](#) "our". This could be useful if "our" consistently marks the start of proprietary information or a specific category we wish to exclude from the extracted data.

To accomplish this, we will input the [REGEXEXTRACT](#) formula into cell **B2**, referencing the content of cell **A2**. The formula uses the previously defined [regular expression](#) pattern to identify and capture the leading text segment:

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

Upon pressing Enter, cell **B2** will display only the text from **A2** that exists before the first instance of "our". This demonstrates the immediate effectiveness of the [capturing group](#) (.\*) .

## Step-by-Step Implementation Guide and Formula Scaling

Once the formula is correctly entered into the first cell (**B2**), the process of applying this extraction logic to the rest of the dataset is straightforward. [Google Sheets](#) allows for rapid deployment of [functions](#) across contiguous cells, significantly speeding up data cleaning and preparation tasks.

To populate the remaining rows in column B, you simply need to utilize the fill handle feature. This involves clicking on the small square located at the bottom right corner of cell **B2** and dragging it down until it covers all corresponding cells in column A. [Google Sheets](#) automatically adjusts the cell reference (from **A2** to **A3**, **A4**, and so on) for each row, applying the same extraction logic to the entire dataset.

The result of dragging the formula down column B is shown below. Notice how column B now accurately displays the prefix text for every phrase in column A, cleanly truncated right before the specified delimiter [string](#) "our" is found. This confirms the successful application of the [REGEXEXTRACT function](#) for mass text extraction.

B2       =REGEXEXTRACT(A2, "(.\*)our.\*")

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

## Practical Example 2: Adapting the Formula for Different Delimiters

One of the greatest advantages of using [regular expressions](#) is the ease with which the pattern can be modified to suit different requirements. To extract text before a different specific character or [string](#), the only adjustment required is replacing the literal delimiter within the [regular expression](#) pattern. The fundamental structure--the ( . \* ) [capturing group](#) followed by the delimiter, followed by . \*--remains constant.

For instance, suppose we now wish to extract all text that appears before the specific sequence "is

" (note the inclusion of the trailing space, which makes the match more precise, ensuring we only capture text before the start of a new clause or description). We simply replace "our" with "is " inside the quoted pattern argument.

The adapted formula, which would be entered into cell **B2** to reference cell **A2**, is structured as follows. This versatility highlights why [functions](#) based on [regular expressions](#) are the industry standard for advanced text parsing:

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

After applying this formula and dragging it down the column, the output in column B reflects the new extraction criterion. As demonstrated in the resulting image below, column B now displays all text that precedes the sequence "is " for every corresponding entry in column A.

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

This powerful adaptation allows analysts to rapidly adjust their text parsing logic without resorting to complex nested [functions](#), proving the efficiency of [REGEXEXTRACT](#).

## Additional Resources for [Google Sheets](#) Text Manipulation

Mastering text extraction is essential for effective data management in [Google Sheets](#). While

**[REGEXEXTRACT](#)** is highly effective for extracting prefixes, similar techniques can be applied to isolate suffixes or middle segments of text. The following tutorials provide further instruction on advanced text manipulation techniques to round out your data processing toolkit:

[Google Sheets: How to Extract Text After a Character](#)