How to Extract Text Between Quotes in Google Sheets

Authored by
mohammed looti

January 8, 2026

ARABPSYCHOLOGY.COM

RECOMMENDED CITATION

mohammed looti (2026). *How to Extract Text Between Quotes in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=125095

Introduction: Mastering Text Extraction in Google Sheets

Extracting specific substrings from complex text data is a common requirement in data analysis and cleaning. When working within Google Sheets, developers and analysts often encounter data where crucial identifiers, names, or values are enclosed within quotation marks, serving as delimiters. While basic functions like MID, FIND, and LEN can technically be combined to solve this problem, this approach quickly becomes complex, brittle, and lengthy, particularly when dealing with varying text lengths or multiple quotes in a single cell. The superior and most robust method involves leveraging the power of REGEXEXTRACT, which utilizes Regular Expressions for pattern matching.

The traditional method, which relies on fixed positioning and character counting, requires you to first use the FIND function to locate the position of the opening quote, and then locate the closing quote. You would then subtract the two positions to determine the length of the string enclosed within. Finally, the MID function uses this starting position and calculated length to extract the target text. This process is mathematically sound but cumbersome to implement and maintain, especially for non-standardized datasets.

Fortunately, Google Sheets provides the specialized REGEXEXTRACT function, specifically designed for pulling text based on defined patterns. This powerful tool simplifies the extraction process significantly, allowing you to define the exact pattern--in this case, "any text found between two quote marks"--with a single, concise formula. This article will focus exclusively on using this highly efficient method to extract text enclosed by both double and single quotes.

Understanding the Power of REGEXEXTRACT for Data Parsing

The core function for this operation is REGEXEXTRACT. This function searches a cell for a substring that matches a defined Regular Expression (regex) pattern and returns the first match found. Unlike simple search functions, regex allows for highly flexible and dynamic pattern definition, making it ideal for isolating variable text strings based on surrounding characters, such as quotes.

To successfully extract text between quotes, we need to construct a regex pattern that achieves two goals: first, it must identify the quote marks themselves (the delimiters); and second, it must capture the text that lies internally between those delimiters. The pattern we use, which generally looks like `"(.*)"` or `"'(.*)'"`, leverages key regex components to achieve this specific capture.

The syntax for REGEXEXTRACT is straightforward: `REGEXEXTRACT(text, regular_expression)`. The `text` argument is simply the cell reference containing the source string (e.g., A2). The crucial part is the `regular_expression` argument, which must be provided as a string (enclosed in double quotes) and must contain the exact pattern to match and capture.

ARABPSYCHOLOGY.COM

The following sections detail the specific regex patterns required for both double and single quotes.

Deeper Dive into Regular Expressions Syntax

To fully appreciate the formulas used, it is essential to understand the components of the Regular Expression pattern: `(.*)`. This short sequence is the heart of the extraction process.

The Period (`.`): In regex, the period acts as a wildcard, matching any single character (except newline characters, though this is rarely an issue in a single Google Sheets cell).

The Asterisk (`*`): The asterisk is a quantifier. When placed immediately after a character or group (like the period), it means "match zero or more occurrences" of the preceding element. Therefore, `.*` means "match any sequence of characters of any length (including an empty sequence)."

Parentheses (`()`): These are the capturing group indicators. When parentheses enclose a part of the regex pattern (like the `.*`), REGEXEXTRACT understands that the user wants to return only the text matched within those parentheses, effectively ignoring the surrounding delimiters (the quotes).

When we combine these elements with the quote marks, we create a precise instruction: "Find an opening quote, then capture everything (`.*`) until you encounter the closing quote." This powerful mechanism allows for highly efficient and dynamic text processing, replacing many steps required by traditional string manipulation functions like MID.

You can use the following concise formulas in Google Sheets to extract all text in a cell between quotes:

Method 1: Extracting Text Between Double Quotes

Extracting content enclosed by double quotes (`"`) requires careful handling within the REGEXEXTRACT function, as the double quote character serves a dual purpose: it acts as the delimiter for the regex pattern itself (the surrounding quotes defining the pattern string) and as the character you are searching for within the cell.

Formula for Double Quotes:

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

Notice the use of four double quotes on either side of the `(.*)` pattern. This is necessary for proper parsing in Google Sheets. The outer pair of quotes defines the string containing the regular expression. The two inner quotes (escaped within the string) represent the actual double quotes

ARABPSYCHOLOGY.COM

you are searching for in the cell content. The pattern `"""(.*)"""` therefore translates to: "Find a double quote, capture everything, and stop at the next double quote."

This approach ensures that the formula accurately targets the exact delimiters used in the source text while satisfying the syntactic requirements of the REGEXEXTRACT function within the spreadsheet environment.

Method 2: Extracting Text Between Single Quotes

When dealing with single quotes (`'`), the syntax for the REGEXEXTRACT function becomes slightly cleaner because the single quote does not conflict with the double quotes used to define the formula string itself. This eliminates the need for the complex escaping required in Method 1.

Formula for Single Quotes:

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

In this pattern, `"'(.*)'"`, the outer double quotes define the regex string argument. The inner single quotes (`'`) are interpreted by the regex engine as the literal characters to be matched within the cell content. The core capturing group, `(.*)`, remains the same, ensuring that all characters between the single quotes are returned as the result.

Both of these powerful examples are designed to extract the specific text found between the designated quotes from cell **A2**. The following practical scenarios illustrate how to deploy and replicate these methods across a large dataset.

Practical Walkthrough: Example 1 (Double Quotes)

To solidify the understanding of Method 1, let's work through a practical scenario where we need to isolate nicknames enclosed in double quotes from a list of athletes. This is a common data cleaning task where the primary identifier is combined with secondary, quoted information.

Suppose we have the following list of athletes with their nicknames in double quotes:

ARABPSYCHOLOGY.COM

	A	B	C
1	**Name**		
2	Andy "Super" James		
3	Bob "Runaway"Smith		
4	Chad "Speedy" Anderson		
5	Doug "King" Miller		
6	Ethan "Tall Guy" Wilks		
7	Frank "The Ghost" Tomms		
8	Greg "Thriller" Dodson		
9			
10			
11			
12			
13			

Our objective is to extract only the text within the double quotes (the nicknames) and place them into Column B. This demonstrates the efficiency of using REGEXEXTRACT over attempting complex nesting of FIND and MID functions.

We can type the following formula into cell **B2** to extract the text between the double quotes in cell **A2**:

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

Once the formula is entered in B2, it will instantaneously return the extracted nickname. We can then click and drag this formula down to each remaining cell in column B using the fill handle. This action populates the entire column with the extracted nicknames:

ARABPSYCHOLOGY.COM

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

	A	B	C
1	**Name**	**Text Between Quotes**	
2	Andy "Super" James	Super	
3	Bob "Runaway"Smith	Runaway	
4	Chad "Speedy" Anderson	Speedy	
5	Doug "King" Miller	King	
6	Ethan "Tall Guy" Wilks	Tall Guy	
7	Frank "The Ghost" Tomms	The Ghost	
8	Greg "Thriller" Dodson	Thriller	
9			
10			
11			
12			
13			
14			

Column B now cleanly contains the text between the double quotes for each corresponding entry in Column A, confirming the success and efficiency of the regex approach.

Practical Walkthrough: Example 2 (Single Quotes)

In the second scenario, we address data where the delimiters are single quotes (`'`). Although the overall logic remains the same, the formula construction is simplified due to the lack of escaping required for single quotes within the string definition.

Consider a similar dataset where athlete names and aliases are provided, but this time, the aliases are enclosed in single quotes:

ARABPSYCHOLOGY.COM

	A	B	C
1	**Name**		
2	Andy 'Super' James		
3	Bob 'Runaway'Smith		
4	Chad 'Speedy' Anderson		
5	Doug 'King' Miller		
6	Ethan 'Tall Guy' Wilks		
7	Frank 'The Ghost' Tomms		
8	Greg 'Thriller' Dodson		
9			
10			
11			
12			
13			
14			
15			

Now suppose we would like to extract the text between the single quotes for each athlete.

We can type the following formula into cell **B2** to extract the text between the single quotes in cell **A2**:

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

Once the formula is active in B2, we repeat the efficient process of dragging the fill handle down the column. This action instantaneously applies the REGEXEXTRACT function to every cell in Column A, ensuring rapid and accurate data processing for the entire range.

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

ARABPSYCHOLOGY.COM

B2 =REGEXEXTRACT(A2,"'(.*)'")

	A	B	C
1	**Name**	**Text Between Quotes**	
2	Andy 'Super' James	Super	
3	Bob 'Runaway'Smith	Runaway	
4	Chad 'Speedy' Anderson	Speedy	
5	Doug 'King' Miller	King	
6	Ethan 'Tall Guy' Wilks	Tall Guy	
7	Frank 'The Ghost' Tomms	The Ghost	
8	Greg 'Thriller' Dodson	Thriller	
9			
10			
11			
12			
13			
14			

Column B now contains the text between the single quotes for each corresponding cell in Column A, demonstrating the flexibility of Regular Expressions.

Advanced Considerations and Alternative Methods

As previously noted, the formula `MID(A2, FIND(""", A2) + 1, FIND(""", A2, FIND(""", A2) + 1) - FIND(""", A2) - 1)` represents the older, multi-step approach using string manipulation functions. While this method is viable, especially for users unfamiliar with Regular Expressions, its complexity increases exponentially if you need to handle errors, multiple quotes, or varying delimiters.

One important advanced consideration for the regex method is the use of the non-greedy quantifier. If a cell contains multiple quoted strings, the default greedy pattern (`.*`) will capture everything between the very first opening quote and the very last closing quote. To strictly extract only the text between the first pair of quotes, you would use the non-greedy match: `"""(.*?)"""`. The addition of the question mark (`?`) ensures the pattern matches the shortest possible string.

For most standard data cleaning tasks in Google Sheets, however, where the quoted text only appears once per cell, the simpler greedy pattern is sufficient, highly readable, and exceptionally efficient.

ARABPSYCHOLOGY.COM

Summary of Text Extraction Techniques

The ability to efficiently extract delimited text is a fundamental requirement for working with semi-structured data in spreadsheets. By utilizing the built-in functions of Google Sheets, you can transform complex text entries into clean, actionable data columns.

We have demonstrated that the most effective and robust solution for extracting text between single or double quotes relies on the REGEXEXTRACT function combined with strategic pattern matching using the capturing group `(.*)`. This technique minimizes formula length and maximizes flexibility compared to manually calculated positional extraction using MID and FIND.

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

ARABPSYCHOLOGY.COM