

How to Easily Escape Quotes in Google Sheets

Authored by
stats writer

November 22, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Escape Quotes in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=99342>

Mastering string manipulation within Google Sheets is essential for advanced data processing and reporting. When working with formulas, incorporating double quotes (" ") directly into a text output can be challenging because double quotes typically signal the beginning or end of a text string literal within the spreadsheet environment. Attempting to include them without proper handling leads to errors, as the sheet interprets the quote as a structural element of the formula rather than a character to be displayed. This process of successfully embedding special characters like double quotes within a text string is known as escaping.

While some environments allow using a backslash (\) to escape quotes, Google Sheets utilizes specific programmatic methods--relying primarily on the CONCATENATE function and character codes--to achieve this goal effectively. Understanding these techniques is crucial for anyone needing to standardize data, create dynamic reports, or prepare data for external systems that require quoted text fields. We will explore the two authoritative methods for wrapping text strings in double quotes directly within your sheet's formulas.

Why Escaping Quotes is Necessary for Text Output

In the context of spreadsheet applications, any text value enclosed in double quotes is treated as a string literal. For example, if you type `="Hello"` into a cell, Sheets knows to output the text "Hello." However, if you want the cell output to literally be **"Hello World"**--including the surrounding quotation marks--you cannot simply type `=""Hello World""`, as the sheet will interpret the second quote as the end of the text string and the third quote as extraneous characters, resulting in a parsing error.

The fundamental problem lies in the ambiguity of the double quote character. It serves a dual purpose: defining a literal string and potentially acting as a character within that string. To resolve this conflict, we must introduce an escape sequence or an alternative representation of the quote character that the Google Sheets formula parser recognizes as text content rather than structure. The methods detailed below provide elegant and robust solutions to this common data manipulation challenge, ensuring data integrity and correct formatting.

These techniques are particularly vital when preparing comma-separated values (CSV) or JSON data where specific fields often need to be enclosed in quotes to maintain proper data structure during export or import processes. Without the ability to reliably wrap text in double quotes, complex data tasks involving integration with other systems would become significantly more difficult or even impossible to manage directly within the spreadsheet environment.

Core Methods for Wrapping Text with Quotes

There are two primary and highly effective methods available in Google Sheets for escaping quotes and wrapping a string in double quotation marks using functions. Both methods utilize the

CONCATENATE function (or the equivalent ampersand operator, **&**) to join the desired text string with the necessary quote characters.

These methods offer flexibility depending on user preference and complexity of the resulting formula. Method 1 is often considered the simplest visually, relying on consecutive quotes to signal the inclusion of a literal quote. Method 2, while slightly more technical, is universally reliable, utilizing the standardized character code for the double quote symbol.

The goal of both techniques is identical: to take the value from a cell (for example, cell **A2** containing "Product Name") and transform it into a new string that explicitly includes the outer quotes (e.g., "**Product Name**"). Understanding the underlying mechanics of these approaches will greatly enhance your ability to perform advanced text formatting operations efficiently.

Method 1: Wrap Quotes Around Quotes (Using String Literals)

```
=CONCATENATE("","",A2,"")
```

Method 2: Use CHAR(34) (Using Character Codes)

```
=CONCATENATE(CHAR(34),A2,CHAR(34))
```

Both of these powerful formulas are designed to achieve the same result: they will successfully wrap the string contained in cell **A2** in double quotes, thereby achieving the desired string escaping outcome.

Understanding the Concatenated Quote String Approach (Method 1)

Method 1, which employs four consecutive double quotes (""""), is a standard technique utilized in many SQL and spreadsheet environments. When the Google Sheets formula engine encounters this sequence within a function like CONCATENATE, it interprets the quotes in pairs. The outer two quotes define the boundaries of the string literal itself, while the inner two quotes are interpreted as the character content that should be included in the resulting output string.

Specifically, the sequence """" results in a single, literal double quote character being produced. Therefore, the structure of the formula is logically broken down as follows: start with a literal quote (""""), append the string content from cell **A2**, and finally, append another literal quote (""""). This sequence ensures that the resulting string is correctly formatted with the necessary enclosing quotation marks.

This method is generally preferred for its simplicity and ease of reading, provided the user understands the convention of using four quotes to represent one literal quote in this context. It is a

highly efficient way to handle simple quote escaping requirements without needing to refer to character codes or specialized lookup tables.

Practical Implementation: Wrapping Quotes Around Quotes (Example 1)

To illustrate this powerful technique, consider a scenario where we have a list of unquoted strings in column A that need to be prepared for an external database requiring quoted text fields. The example below shows an initial dataset in column A, listing various strings.

We will apply the concatenated quote string approach (Method 1) to column B, ensuring every entry in column A is correctly wrapped in double quotes. This transformation is necessary for maintaining data integrity and ensuring compatibility with systems that strictly require quoted parameters.

	A	B	C	D
1	String			
2	Hello there everyone			
3	Hey what is going on			
4	How is everybody doing			
5	Hi how are you			
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				

We initiate the process by inputting the following formula into cell **B2**. This specific formula is structured to concatenate the opening quote, the text reference from **A2**, and the closing quote:

=CONCATENATE("","A2","")

Once the formula is entered into **B2**, we can swiftly apply it to the remaining dataset. We drag the

formula handle down to populate the rest of the cells in column B. This action automatically adjusts the cell reference (e.g., from **A2** to **A3**, **A4**, and so on), ensuring that every string in column A is processed correctly and given the required surrounding quotes.

B2		<i>fx</i>	=CONCATENATE("","A2","")	
	A	B	C	D
1	String	String with Quotes		
2	Hello there everyone	"Hello there everyone"		
3	Hey what is going on	"Hey what is going on"		
4	How is everybody doing	"How is everybody doing"		
5	Hi how are you	"Hi how are you"		
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				

Upon reviewing the results in column B, it becomes evident that each cell now contains the corresponding string from column A, successfully encapsulated within double quotes. This visual confirmation verifies the proper implementation of the quote-wrapping technique using concatenated string literals.

Utilizing the CHAR Function for Quote Escaping (Method 2)

The second, and arguably more robust, method for escaping quotes involves leveraging the CHAR function. The CHAR function converts a numerical code into the corresponding character based on the standardized Unicode table.

Crucially, the numerical value **34** is universally assigned to the double quotation mark (") within the

Unicode system, which Google Sheets relies upon. Therefore, calling **CHAR(34)** explicitly generates a double quote character, treating it purely as data output rather than formula structure.

The formula uses the CONCATENATE function to join three distinct elements: the character generated by **CHAR(34)**, the cell reference containing the target string (e.g., **A2**), and a final character generated by **CHAR(34)**. This method avoids the visual complexity of the four-quote sequence and relies on a direct numerical mapping, offering superior clarity for users familiar with Unicode standards.

Practical Implementation: Applying CHAR(34) for Quote Escaping (Example 2)

We now demonstrate Method 2 using the same dataset from Example 1. This example showcases the reliability and clarity of using character codes for complex string composition, particularly useful when dealing with various special characters beyond just double quotes.

The implementation begins by entering the following comprehensive formula into cell **B2**. This structure clearly delineates the components being joined: the opening quote (**CHAR(34)**), the text content (**A2**), and the closing quote (**CHAR(34)**).

=CONCATENATE(CHAR(34),A2,CHAR(34))

Similar to the previous example, the next step involves using the fill handle to automatically apply this formula down to the remaining cells in column B. This ensures that every corresponding entry in column A receives the correct escaping treatment using the character code representation.

B2		<i>fx</i>	=CONCATENATE(CHAR(34),A2,CHAR(34))
	A	B	C
1	String	String with Quotes	
2	Hello there everyone	"Hello there everyone"	
3	Hey what is going on	"Hey what is going on"	
4	How is everybody doing	"How is everybody doing"	
5	Hi how are you	"Hi how are you"	
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			

The output confirms that each cell in column B successfully encapsulates the string from column A in double quotes, demonstrating the effectiveness and precision of using **CHAR(34)** for reliable quote escaping.

It is important to reiterate that the CHAR function is an indispensable tool for working with characters that are difficult or impossible to type directly into a formula string, or characters that hold special meaning, such as the double quote in this scenario. By referencing the numerical code **34**, we ensure an unambiguous interpretation by the spreadsheet software.

Choosing the Right Method for Quote Escaping

While both the quadrupled quotes (""") and the **CHAR(34)** function are valid and produce identical results, the choice between them often depends on context, audience, and complexity.

The quadrupled quote method (Method 1) is typically faster to type and is intuitive for users who are already familiar with how string literals are handled in database query languages like SQL. It results in a shorter, more concise formula, which can be advantageous in scenarios where the

formula bar is already crowded with other functions. However, new users might find the "'''''' syntax confusing, requiring explanation that four quotes equal one literal quote.

Conversely, the **CHAR(34)** method (Method 2) is highly explicit. By using a numerical reference, it removes all ambiguity about the character being generated. This method is preferred when dealing with international data or when generating various special characters (like newline characters, tabs, etc.), as the CHAR function is universally applicable across the entire Unicode set. If your audience includes spreadsheet novices, **CHAR(34)** is often easier to teach and debug because its purpose is clearly defined.

Ultimately, for the simple task of wrapping text in double quotes, both are excellent options. Experienced users often gravitate towards the shorter "'''''' method for efficiency, whereas those prioritizing clarity and compatibility with other special character generation techniques often prefer **CHAR(34)**.

Mastering String Manipulation in Google Sheets

Effectively escaping double quotes is a foundational skill in advanced Google Sheets data preparation. By employing either the CONCATENATE function with quadrupled quotes or the highly specific **CHAR(34)** function, users can reliably format strings for export, database interaction, or complex internal reporting.

Remember that the function **CHAR(34)** directly corresponds to the double quote character in the current Unicode table. This capability is not limited to quotes; you can find complete documentation for the CHAR function and other powerful text manipulation tools in the official Google Sheets support documentation, enabling you to handle virtually any character escaping requirement.

Mastering these precise string manipulation techniques ensures that your data outputs are consistently clean, well-formed, and compliant with the technical requirements of any system or application they interact with.

Note: You can find the complete documentation for the CHAR function in Google Sheets documentation.