

# TRIMQUOTES Function

Removes leading and trailing quotes or double-quotes from a string. Quote marks in the middle of the string are not removed.

- This function applies to both single quotes ( ' ) and double quotes ( " ).
  - This function is not limited to removing the outer set of quotes only. If there are multiple quotes at the beginning or the end of the string ( " " ), all sets of quotes are removed.
- The TRIMQUOTES function does not remove whitespace at the beginning and end.

**Tip:** You may need to nest this function and the TRIM function to clean up your strings. An example is provided below.

**Wrangle vs. SQL:** This function is part of Wrangle , a proprietary data transformation language. Wrangle is not SQL. For more information, see *Wrangle Language*.

## Basic Usage

### Column reference example:

```
trimquotes(MyName)
```

**Output:** Returns the values of the MyName column value with any quotes removed from the beginning and the end.

### String literal examples:

**NOTE:** For string literal values that contain quotes to remove, you can bracket them in the quote mark of a different type. Some examples are below.

```
trimquotes(&apos;&quot;Hello, World&quot;&apos;)
```

**Output:** Input string is "Hello, World". Output of the function is the string: Hello, World.

```
trimquotes(&apos;&quot;&quot;Hello, \&quot; World&quot;&quot;&apos;)
```

**Output:** Input string is "'Hello, \" World'". Output of the function is the string: Hello, " World.

Following input contains a single whitespace at the beginning and end of the string, which is the same as the previous string. To clean, you can first remove the whitespace with the TRIM function.

```
trimquotes(trim((&apos; &quot;&quot;Hello, \&quot; World&quot;&quot; &apos;)))
```

**Output:** Input string is "'Hello, \" World' ". Output of the function is the string: Hello, " World. See *TRIM Function*.

## Syntax and Arguments

```
trimquotes(column_string)
```

Argument	Required?	Data Type	Description
----------	-----------	-----------	-------------

column_string	Y	string	Name of the column or string literal to be applied to the function
---------------	---	--------	--

For more information on syntax standards, see *Language Documentation Syntax Notes*.

## column\_string

Name of the column or string constant whose beginning and end quote marks are to be trimmed.

- Missing string or column values generate missing string results.
- String constants must be quoted ('Hello, World').
- Multiple columns and wildcards are not supported.

### Usage Notes:

Required?	Data Type	Example Value
Yes	String literal or column reference	myColumn

### Examples

**Tip:** For additional examples, see *Common Tasks*.

### Example - String whitespace and quotes

The following example data contains a mixture of quotes and spaces strings. You can use the transformation listed below to clean up leading and trailing quotes and strings in a single transformation.

#### Source:

String	Description
My String	"Base string: ""My String"""
My String extra	"Base string + "" extra"""
My String	A space in front of base string
My String	A space after base string
MyString	No space between the two words of base string
My String	Two spaces between the two words of base string
"My String "	Base string + a tab character
"My String "	Base string + a return character
"My String "	Base string + a newline character

#### Transformation:

You can use the following transformation which nests the TRIM and the TRIMQUOTES functions to clean up all of the columns in your dataset.

- To apply across all columns in the dataset:

- The wildcard (\*) for columns indicates that this formula should be applied across all columns in the dataset.
- You can also select All from the Columns drop-down in the Transform Builder.
- Since all columns are String type, the results should be consistent.
- The \$col reference can be used to refer to the current column that is being evaluated. For more information, see *Source Metadata References*.

<b>Transformation Name</b>	Edit column with formula
<b>Parameter: Columns</b>	All
<b>Parameter: Formula</b>	trimquotes(trim(\$col))

**Results:**

String	Description
My String	Base string: "My String
My String extra	Base string + " extra
My String	A space in front of base string
My String	A space after base string
MyString	No space between the two words of base string
My String	Two spaces between the two words of base string
My String	Base string + a tab character
My String	Base string + a return character
My String	Base string + a newline character