

TRIM Function

Contents:

- *Basic Usage*
- *Syntax and Arguments*
 - *column_string*
- *Examples*
 - *Example - Trimming leading and trailing whitespace*
 - *Example - String cleanup functions together*

Removes leading and trailing whitespace from a string. Spacing between words is not removed.

- If a string begins or ends with spaces, tabs, or other non-visible characters, they are removed by this function.
- The TRIM function does not remove whitespace between non-whitespace values, such as spaces between words. To remove that type of whitespace, use REMOVEWHITESPACE. See REMOVEWHITESPACE Function.

Basic Usage

Column reference example:

```
derive type:single value:TRIM(MyName)
```

Output: The value of the MyName column value with whitespace removed from the beginning and the end.

String literal example:

```
derive type:single value:TRIM(' Hello, World ')
```

Output: The string Hello, World is written to the new column.

Syntax and Arguments

```
derive type:single value:TRIM(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 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 - Trimming leading and trailing whitespace

In this example, whitespace values are identified according to this table. The ASCII value column identifies that ASCII character value that represents the character.

- The ASCII character set is a standard method for representing keyboard and special characters on the computer. For more information on ASCII, see <http://www.asciitable.com/>.

Value	Definition	ASCII value
(space)	spacebar	Char(32)
(tab)	tab character	Char(9)
(cr)	carriage return	Char(13)
(nl)	newline	Char(10)

Source:

In the following example dataset, input values are represented in the `mystring`. The values in the table above are represented in the string values below.

mystring
Here's my string.
(space)(space)Here's my string.(space)(space)
(tab)Here's my string.(tab)
(cr)Here's my string.(cr)
(nl)Here's my string.(nl)
(space)(space)(tab)Here's my string.(tab)(space)(space)
(space)(space)(tab)(cr)Here's my string.(cr)(tab)(space)(space)
(space)(space)(tab)(nl)(cr)Here's my string.(cr)(nl)(tab)(space)(space)

Input:

When the above CSV data is imported into the Transformer page, it is represented as the following:

mystring
Here's my string.
(space)(space)Here's my string.(space)(space)
"(tab)Here's my string.(tab)"

"(cr)Here's my string.(cr)"
"(nl)Here's my string.(nl)"
"(space)(space)(tab)Here's my string.(tab)(space)(space)"
"(space)(space)(tab)(cr)Here's my string.(cr)(tab)(space)(space)"
"(space)(space)(tab)(nl)(cr)Here's my string.(cr)(nl)(tab)(space)(space)"

Transform:

You might notice the quote marks around most of the imported values.

NOTE: If an imported string value contains tab, carriage return, or newline values, it is bracketed by double quotes.

The first step is to remove the quote marks. You can select one of the quote marks in the data grid and then select the appropriate Replace suggestion card. The transform should look like the following:

```
replace col: mystring on: `"` with: ' ' global: true
```

Now, you can apply the TRIM function:

```
derive type:single value: TRIM(mystring) as: 'trim_mystring'
```

Results:

In the generated trim_mystring column, you can see the cleaned strings:

mystring	trim_mystring
Here's my string.	Here's my string.
(space)(space)Here's my string.(space)(space)	Here's my string.
"(tab)Here's my string.(tab)"	Here's my string.
"(cr)Here's my string.(cr)"	Here's my string.
"(nl)Here's my string.(nl)"	Here's my string.
"(space)(space)(tab)Here's my string.(tab)(space)(space)"	Here's my string.
"(space)(space)(tab)(cr)Here's my string.(cr)(tab)(space)(space)"	Here's my string.
"(space)(space)(tab)(nl)(cr)Here's my string.(cr)(nl)(tab)(space)(space)"	Here's my string.

Tip: If any bracketing double quotes are removed, then tab, carriage return, and newline values are trimmed by the TRIM function.

Example - String cleanup functions together

The following example demonstrates functions that can be used to clean up strings. These functions include the following:

- TRIM - remove leading and trailing whitespace. See *TRIM Function*.
- REMOVEWHITESPACE - remove leading and trailing whitespace and all whitespace in between. See *REMOVEWHITESPACE Function*.
- REMOVESYMBOLS - remove all characters that are not alpha-numeric or whitespace. See *REMOVESYMBOLS Function*.

Source:

In the following (space) and (tab) indicate space keys and tabs, respectively. Carriage return and newline characters are also supported by whitespace functions.

Strings	source
String01	this source(space)(space)
String02	(tab)(tab)this source
String03	(tab)(tab)this source(space)(space)
String04	this source's?
String05	Why, you @#\$\$%^&*()!
String06	this sörce
String07	(space)this sörce
String08	à mañana

Transform:

The following transforms generate new columns using each of the string cleanup functions:

```
derive type:single value: TRIM(source) as: 'trim_source'
```

```
derive type:single value: REMOVEWHITESPACE(source) as: 'removewhitespace_source'
```

```
derive type:single value: REMOVESYMBOLS(source) as: 'removesymbols_source'
```

Results:

Strings	source	removesymbols_source	removewhitespace_source	trim_source
String01	this source(space)(space)	this source(space)(space)	thissource	this source
String02	(tab)(tab)this source	(tab)(tab)this source	thissource	this source
String03	(tab)(tab)this source(space)(space)	(tab)(tab)this source(space)(space)	thissource	this source
String04	this source's?	this sources	thissource's?	this source's?
String05	"Why, you @#\$\$%^&*()!"	Why you	Why,you@#\$\$%^&*()!	Why, you @#\$\$%^&*()!
String06	this sörce	this sörce	thissörce	this sörce
String07	(space)this sörce	(space)this sörce	thissörce	this sörce
String08	à mañana	à mañana	àmañana	à mañana