

# REPEAT Function

## Contents:

- *Basic Usage*
- *Syntax and Arguments*
  - *column\_string*
  - *rpt\_count*
- *Examples*
  - *Example - REPEAT string function*
  - *Example - Padding values*

---

Repeats a string a specified number of times. The string can be specified as a String literal, a function returning a String, or a column reference.

- Since the REPEAT function matches based on fixed numeric values, changes to the length or structure of a data field can cause your recipe to fail to properly execute.
- The REPEAT function requires an integer value for the number of characters to match.

## Basic Usage

### String literal example:

```
repeat('ha', 3)
```

**Output:** Returns the string: hahaha.

### Column reference example:

```
repeat(MyString, 4)
```

**Output:** Returns the values of the MyString column value written four times in a row.

## Syntax and Arguments

```
repeat(column_string, rpt_count)
```

Argument	Required?	Data Type	Description
column_string	Y	string	Name of the column or string literal to be applied to the function
rpt_count	N	integer (positive)	Count of times to repeat the string

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

### column\_string

Name of the column or String literal to be repeated.

- 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, function, or column reference	myColumn

**rpt\_count**

Count of times to repeat the string.

- If the value is not specified, the default is 1.
- Value must be a non-negative integer.
- References to columns of integer data type are not supported.

**Usage Notes:**

Required?	Data Type	Example Value
No	Integer (non-negative)	5

**Examples**

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

**Example - REPEAT string function**

**Source:**

myStr	repeat_count
ha	0
ha	1
ha	1.5
ha	2
ha	-2

**Transformation:**

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	repeat(myStr,repeat_count)
<b>Parameter: New column name</b>	'repeat_string'

**Results:**

myStr	repeat_count	repeat_string
-------	--------------	---------------

ha	0	
ha	1	ha
ha	1.5	
ha	2	haha
ha	-2	

### Example - Padding values

In the following example, the imported `prodId` values are supposed to be eight characters in length. Somewhere during the process, however, leading 0 characters were truncated. The steps below allow you to re-insert the leading characters.

#### Source:

prodName	prodId
w01	1
w02	10000001
w03	345
w04	10402

#### Transformation:

First, we need to identify how many zeroes need to be inserted for each `prodId`:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	<code>8 - len(prodId)</code>
<b>Parameter: New column name</b>	<code>'len_prodId'</code>

Use the `REPEAT` function to generate a pad string based on the above values:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	<code>repeat('0', len_prodId)</code>
<b>Parameter: New column name</b>	<code>'padString'</code>

Merge the pad string and the original `prodId` column:

<b>Transformation Name</b>	Merge columns
<b>Parameter: Columns</b>	<code>padString,prodId</code>
<b>Parameter: Separator</b>	<code>' '</code>
<b>Parameter: New column name</b>	<code>'column2'</code>

**Results:**

When you delete the intermediate columns and rename column2 to prodId, you have the following table:

prodName	prodId
w01	00000001
w02	10000001
w03	00000345
w04	00010402