

PARSESTRING Function

Evaluates an input against the String datatype. If the input matches, the function outputs a String value. Input can be a literal, a column of values, or a function returning values. Values can be of any data type.

After you have converted your values to strings, if a sufficient percentage of inputs from a column are successfully converted to the other data type, the column may be retyped.

Tip: If the column is not automatically retyped as a result of this function, you can manually set the type to String in a subsequent recipe step.

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

```
parsestring(strInput)
```

Output: Returns the String data type value for `strInput` values.

Syntax and Arguments

```
parsestring(str_input)
```

Argument	Required?	Data Type	Description
str_input	Y	any	Literal, name of a column, or a function returning values to match

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

str_input

Literal, column name, or function returning values that are to be evaluated for conversion to String values.

- Missing values for this function in the source data result in null values in the output.
- Multiple columns and wildcards are not supported.

Usage Notes:

Required?	Data Type	Example Values
Yes	any	5 "Porsche" 3.4

Examples

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

Example - type parsing functions

Source:

The following table contains values for city, state, and zip code for locations in the United States:

City	State	Zip
San Francisco	CA	94105
Seattle	WA	98109
Portland	OR	97202
San Diego	CA	92109
Brooklyn	NY	11203
Portland	ME	4101
Boston	MA	2170

In the above table, you can see that some of the values are listed as four-digit zip codes, which are invalid. These values are likely to be interpreted as Integer values, which means that any leading zeroes are dropped. You can use the steps below to fix it.

Transformation:

Since you are working with integer values, you can use the following transformation to test the length of the values as if they were strings using the PARSESTRING function. If the values are only four characters long, then the value is merged with a leading 0:

Transformation Name	Edit column with formula
Parameter: Columns	Zip
Parameter: Formula	<code>if(len(parsestring(\$col)) == 4, merge(['0',parsestring(\$col)]), \$col)</code>

The `$col` reference points to the column that has been selected to be edited. In this case, that column is Zip. For more information, see *Source Metadata References*.

Depending on the number of rows in your dataset, the Trifacta application may not re-infer the data as Zip type. You can use the following transformation to change the data type for the column to Zip:

Transformation Name	Change column data type
Parameter: Columns	Zip
Parameter: New type	Zipcode

Results:

City	State	Zip
San Francisco	CA	94105
Seattle	WA	98109
Portland	OR	97202
San Diego	CA	92109
Brooklyn	NY	11203
Portland	ME	04101
Boston	MA	02170