DEGREES Function

Computes the degrees of an input value measuring the radians of an angle. The value can be a Decimal or Integer literal or a reference to a column containing numeric values.

- Input units are in radians.
- You can convert from degrees to radians. For more information, see *RADIANS Function*.

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

Numeric literal example:

```
round(degrees(1.0000),4)
```

Output: Returns the computation in degrees of `1.0000` radians, which is `57.2728`, rounded to 4 decimals.

Column reference example:

```
degrees(myRads)
```

Output: Returns the conversion of the values in `MyRads` column to degrees.

Syntax and Arguments

```
degrees(numeric_value)
```

Argument	Required?	Data Type	Description
numeric_value	Y	string, decimal, or integer	Name of column, Decimal or Integer literal, or function returning those types to apply to the function

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

numeric_value

Name of the column, Integer or Decimal literal, or function returning that data type to apply to the function.

- Missing input values generate missing results.
- Literal numeric values should not be quoted. Quoted values are treated as strings.
- Multiple columns and wildcards are not supported.

Usage Notes:

Required?	Data Type	Example Value
Yes	String (column reference) or Integer or Decimal literal	`3.14`

Examples

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

Example - DEGREES and RADIANS functions

This example illustrates to convert values from one unit of measure to the other.

Functions:

Item	Description
DEGREES Function	Computes the degrees of an input value measuring the radians of an angle. The value can be a Decimal or Integer literal or a reference to a column containing numeric values.
RADIANS Function	Computes the radians of an input value measuring degrees of an angle. The value can be a Decimal or Integer literal or a reference to a column containing numeric values.
ROUND Function	Rounds input value to the nearest integer. Input can be an Integer, a Decimal, a column reference, or an expression. Optional second argument can be used to specify the number of digits to which to round.

Source:

In this example, the source data contains information about a set of isosceles triangles. Each triangle is listed in a separate row, with the listed value as the size of the non-congruent angle in the triangle in degrees.

You must calculate the measurement of all three angles of each isosceles triangle in radians.

triangle	a01
t01	30
t02	60
t03	90
t04	120
t05	150

Transformation:

You can convert the value for the non-congruent angle to radians using the following:

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`ROUND(RADIANS(a01), 4)`
Parameter: New column name	`'r01'`

Now, calculate the value in degrees of the remaining two angles, which are congruent. Since the sum of all angles in a triangle is 180, the following formula can be applied to compute the size in degrees of each of these angles:

Transformation Name	`New formula`

Parameter: Formula type	`Single row formula`
Parameter: Formula	`(180 - a01) / 2`
Parameter: New column name	`'a02'`

Convert the above to radians:

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`ROUND(RADIANS(a02), 4)`
Parameter: New column name	`'r02'`

Create a second column for the other congruent angle:

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`ROUND(RADIANS(a02), 4)`
Parameter: New column name	`'r03'`

To check accuracy, you sum all three columns and convert to degrees:

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`ROUND(RADIANS(a02), 4)`
Parameter: New column name	`'checksum'`

Results:

After you delete the intermediate columns, you see the following results and determine the error in the checksum is acceptable:

triangle	a01	r03	r02	r01	checksum
t01	30	1.3095	1.3095	0.5238	179.9967
t02	60	1.0476	1.0476	1.0476	179.9967
t03	90	0.7857	0.7857	1.5714	179.9967
t04	120	0.5238	0.5238	2.0952	179.9967
t05	150	0.2619	0.2619	2.6190	179.9967