

UNICODE Function

Generates the Unicode index value for the first character of the input string.

- **Unicode** is a digital standard for the consistent encoding of the world's writing systems, so that representation of character sets is consistent around the world.
- The first 256 Unicode characters (0, 255) correspond to the ASCII character set.
- If the function cannot resolve a Unicode character from the first character, it returns a null value.

Basic Usage

Column reference example:

```
derive type:single value:UNICODE(MyChar)
```

Output: The Unicode index value for the first character in the `MyChar` column is written to the new column.

String literal example:

```
derive type:single value:UNICODE('A')
```

Output: The integer 65 is written to the new column.

Syntax and Arguments

```
derive type:single value:UNICODE(column_string)
```

| Argument | Required? | Data Type | Description |
|---------------|-----------|-----------|--|
| column_string | Y | string | Name of the column or string literal the Unicode value of which is generated |

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

column_string

Name of the column or string literal, the first character of which is converted to its corresponding Unicode value.

NOTE: If the input string contains multiple characters, the first character is mapped to its Unicode value, and the rest are ignored.

- 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 - char and unicode functions

In this example, you can see how the `CHAR` function can be used to convert numeric index values to Unicode characters, and the `UNICODE` function can be used to convert characters back to numeric values.

Source:

The following column contains some source index values:

| index |
|-------|
| 1 |
| 33 |
| 33.5 |
| 34 |
| 48 |
| 57 |
| 65 |
| 90 |
| 97 |
| 121 |
| 254 |
| 255 |
| 256 |
| 257 |
| 9998 |
| 9999 |

Transform:

When the above values are imported to the Transformer page, the column is typed as integer, with a single mismatched value (33.5). To see the corresponding Unicode characters for these characters, enter the following transform:

```
derive type:single value: CHAR(index) as: 'char_index'
```

To see how these characters map back to the index values, now add the following transform:

```
derive type:single value: UNICODE(char_index) as: 'unicode_char_index'
```

Results:

| index | char_index | unicode_char_index |
|-------|------------|--------------------|
| 1 | | 1 |
| 33 | ! | 33 |
| 33.5 | | |
| 34 | " | 34 |

| | | |
|------|---|------|
| 48 | 0 | 48 |
| 57 | 9 | 57 |
| 65 | A | 65 |
| 90 | Z | 90 |
| 97 | a | 97 |
| 122 | z | 122 |
| 254 | þ | 254 |
| 255 | ÿ | 255 |
| 256 | | 256 |
| 257 | | 257 |
| 9998 | | 9998 |
| 9999 | | 9999 |

Note that the floating point input value was not processed.