**Formatting Numbers**

Docmosis has a built-in function to format numbers:

***numFormat ( value, format )***

The input parameters are:

***value*** *= the number to format*

***format*** *= a string that represents the format to apply*

***numFormat*** will round the ***value*** to the required accuracy as indicated by the ***format***.

The ***format*** string should be enclosed in single quotes.

**Common Formats**

|  |  |  |
| --- | --- | --- |
| ***value*** | ***format*** | ***result*** |

|  |  |  |
| --- | --- | --- |
| WHOLE NUMBERS - No thousands separator. No decimal places. | | |
| 12345  12,345  123.45  12345.6  12,345.67 | #  #  #  #  # | <<{numFormat( 12345, ‘#’ )}>>  <<{numFormat( ‘12,345’, ‘#’ )}>>  <<{numFormat( 123.45, ‘#’ )}>>  <<{numFormat( 12345.6, ‘#’ )}>>  <<{numFormat( ‘12,345.67’, ‘#’ )}>> |

|  |  |  |
| --- | --- | --- |
| THOUSANDS - Comma separated thousands. No decimal places. | | |
| 123456  123456.7  1234567.89 | #,###  #,###  #,### | <<{numFormat( 123456, ‘#,###’ )}>>  <<{numFormat( 123456.7, ‘#,###’ )}>>  <<{numFormat( 1234567.89, ‘#,###’ )}>> |

|  |  |  |
| --- | --- | --- |
| CURRENCY - Leading $ sign. Comma separated thousands. Two decimal places. | | |
| 12  123.4  123.4567  1234  123456789  123456.789 | $#,###.00  $#,###.00  $#,###.00  $#,###.00  $#,###.00  $#,###.00 | <<{numFormat( 12, ‘$#,###.00’ )}>>  <<{numFormat( 123.4, ‘$#,###.00’ )}>>  <<{numFormat( 123.4567, ‘$#,###.00’ )}>>  <<{numFormat( 1234, ‘$#,###.00’ )}>>  <<{numFormat( 123456789, ‘$#,###.00’ )}>>  <<{numFormat( 123456.789, ‘$#,###.00’ )}>> |

**Decimals and Accuracy**

|  |  |  |
| --- | --- | --- |
| ***value*** | ***format*** | ***result*** |

|  |  |  |
| --- | --- | --- |
| Always show three decimal places. (Add/remove zeroes for more/less places) | | |
| 123  123.4  123.45678  1234567.8 | #.000  #.000  #.000  #.000 | <<{numFormat( 123, ‘#.000’ )}>>  <<{numFormat( 123.4, ‘#.000’ )}>>  <<{numFormat( 123.45678, ‘#.000’ )}>>  <<{numFormat( 1234567.8, ‘#.000’ )}>> |

|  |  |  |
| --- | --- | --- |
| None or one decimal place. (Add/remove #’s after the point for more/less places) | | |
| 123  123.4  123.4567 | #.#  #.#  #.# | <<{numFormat( 123, ‘#.#’ )}>>  <<{numFormat( 123.4, ‘#.#’ )}>>  <<{numFormat( 123.4567, ‘#.#’ )}>> |

|  |  |  |
| --- | --- | --- |
| Always show a minimum of four digits. (Add/remove zeroes for more/less places) | | |
| 12  123456  123.456 | 0000  0000  0000 | <<{numFormat( 12, ‘0000’ )}>>  <<{numFormat( 123456, ‘0000’ )}>>  <<{numFormat( 123.456, ‘0000’ )}>> |

**Units**

|  |  |  |
| --- | --- | --- |
| Apply units of measurement after a number. | | |
| 123.4  123  123456 | #.00 kg  # miles  #,### feet | <<{numFormat( 123.45, ‘#.00 kg’ )}>>  <<{numFormat( 123, ‘# miles’ )}>>  <<{numFormat( 123456, ‘#,### feet’ )}>> |

**Percentages**

|  |  |  |
| --- | --- | --- |
| Percentage | | |
| 0.3  1.1  0.12  0.1234  0.5  0.1234 | #%  #%  #.00%  #.00%  %#  %#.00 | <<{numFormat( 0.3, ‘#%’ )}>>  <<{numFormat( 1.1, ‘#%’ )}>>  <<{numFormat( 0.12, ‘#.00%’ )}>>  <<{numFormat( 0.1234, ‘#.00%’ )}>>  <<{numFormat( 0.5, ‘%#’ )}>>  <<{numFormat( 0.1234, ‘%#.0’ )}>> |

**Formatting input Data**

|  |  |
| --- | --- |
| Perform calculations on input data and format the result. | |
| weight = <<weight>>  units = <<units>>  distance = <<distance>>  price = <<price>>  qty = <<qty>>  testScore = <<testScore>>  testTotal = <<testTotal>> | <<{numFormat(weight,’# ’+units)}>>  <<{numFormat(distance/1000, ‘#.# km’)}>>  <<{numFormat(price\*qty, ‘$#,###.00’)}>>  <<{numFormat(testScore/testTotal, ‘# %’)}>> |

**Same Number : Different Formats**

|  |  |  |
| --- | --- | --- |
| ***value*** | ***format*** | ***result*** |

|  |  |  |
| --- | --- | --- |
| 1234567.891  1234567.891  1234567.891  1234567.891  1234567.891  1234567.891  1234567.891  1234567.891  1234567.891  1234567.891  1234567.891 | #  ,###  #,###  #.#  ,###.#  #.#####  #.0  #.00000  $#,###.00  abc #  # abc | <<{numFormat( 1234567.891, ‘#’ )}>>  <<{numFormat( 1234567.891, ‘,###’ )}>>  <<{numFormat( 1234567.891, ‘#,###’ )}>>  <<{numFormat( 1234567.891, ‘#.#’ )}>>  <<{numFormat( 1234567.891, ‘,###.#’ )}>>  <<{numFormat( 1234567.891, ‘#.#####’ )}>>  <<{numFormat( 1234567.891, ‘#.0’ )}>>  <<{numFormat( 1234567.891, ‘#.0000’ )}>>  <<{numFormat(1234567.891, ‘$#,###.00’ )}>>  <<{numFormat( 1234567.891, ‘abc #’ )}>>  <<{numFormat( 1234567.891, ‘# abc’ )}>> |