Numeric Functionality

Operators

Without any parentheses, operators will follow standard order-of-operations (ie. multiplication before addition, logical operators last).

Operator	Notes
+	Addition
-	Subtraction
*	Multiplication. 2 + 2 * 2 = 6, not 8.
/	Division. 2 + 2 / 2 = 3, not 2. 5 / 2 = 2.5.
(…)	Parentheses, to allow forced order of operations. (2 + 2) / 2 = 2.
=	Two values are equal.
<	Less than.
>	Greater than.
<=	Less than or equal.
>=	Greater than or equal.
!=	Not equal.

Functions

Function	Notes
Abs(x)	Absolute value.
Mod(x,y)	Modulo (remainder of x/y). Modulo is always positive.
Max(x,y)	Whichever value is larger.
Min(x,y)	Whichever value is smaller.
Pow(x,y)	x to the power of y.
Ceiling(x)	Rounds towards infinity, $1.1 \rightarrow 2$, $-1.1 \rightarrow -1$.
Floor(x)	Rounds towards -infinity, $1.1 \rightarrow 1$, $-1.1 \rightarrow -2$.
RoundAwayFromZero(x)	Rounds away from zero, $1.1 \rightarrow 2$, $-1.1 \rightarrow -2$.
RoundTowardsZero(x)	Rounds towards zero, $1.1 \rightarrow 1$, $-1.1 \rightarrow -1$.
RoundHalfTowardsZero(x)	Rounds to nearest, ties going towards zero. $1.5 \rightarrow 1$, $-1.5 \rightarrow -1$.
RoundHalfAwayFromZero(x)	Rounds to nearest, ties going away from zero. $1.5 \rightarrow 2$, $-1.5 \rightarrow -2$.
RoundHalfEven(x)	Rounds to nearest, ties going to nearest even integer. $1.5 \rightarrow 2$, $2.5 \rightarrow 2$.

Examples

Example	Notes
| ```
If(

    Income < 37178, Income * 0.15,

    If(Income < 74357,

        5577 + ((Income - 37178) *
0.22),
``` | Calculation of federal Canadian income tax using nested If blocks. |

```
        If(Income < 120887,

            13756 + ((Income - 74357)
* 0.26),

            25854 + ((Income -
120887) * 0.29)

        )

      )

   )
```

■
String Functionality

Strings are denoted by text contained in a double-quotation marks. A double-quotation mark can be contained in a string if it is escaped with a backslash. Due to the use of a backslash as an escape character, a literal backslash can be specified with a double-blackslash. For example:

Calculated Column Calculation	Resulting Report Value
`"Hello world!"`	`Hello world!`
`"Jack said, \"I hate my dog!\""`	`Jack said, "I hate my dog!"`
`"Costs \\ Expenses"`	`Costs \ Expenses`

Operators

Without any parentheses, operators will follow standard order-of-operations (ie. multiplication before addition, logical operators last).

Operator	Notes
+	Concatenates two strings.
=	Two values are equal. String comparisons are case-insensitive.
!=	Not equal. String comparisons are case-insensitive.

Functions

Function	Notes
Len(s)	Calculates the length of a string. s must be a string, but a numeric value is returned.
Left(s, length)	Extracts the first length characters from a string.
Mid(s, start, length)	Extracts a range of characters from a string.
Right(s, length)	Extracts the last length characters from a string.
Find(s, sub)	Finds sub in s. Returns the zero-based index, or -1 if not found.
RFind(s, sub)	Finds sub in s, starting from the right side of the string. Returns the zero-based index, or -1 if not found.
IsEmpty(s)	Equivilant to Len(s) = 0.
IsNullOrEmpty(s)	Equivilant to IsNull(s) || IsEmpty(s).

Logical Functionality

Operators

Operator	Notes
&&	Logical AND operator.
‖	Logical OR operator.

Functions

Function	Notes
If(cond, true, false)	Logical if statement, IF cond THEN true ELSE false. cond must be a numeric value or a logical comparison. true and false must be the same type, whether string or numeric.
IsNull(variable)	True / 1 if the given variable is null and cannot be used in a calculation, false / 0 if the variable has a value. IsNull can only be passed a variable, ie. the name of another report column – any other value would always be true.

String/Numeric Conversions

Function	Notes
TextToNumber(s)	Parses a string and converts it to a numeric value.
NumberToText(s)	Converts a number to a string. The resulting string will have the same number format as configured in the report's settings.