Calculator

Description

Calculator allows you to perform complex calculations using Excel-like functions, based on the popular NPM library mathjs.

Then these calculations are stored in private data.

Use cases

With the classic [Switch.Calculation:Expression=””] function, the only functions available are the basic operators '+', '-', '*', '/', '%' and 'round()'.
In addition ‘round()’ can be misleading because it rounds to the nearest integer.

With **Calculator**, you will be able to use complex math function or constant such as:

Exemple	Expression	Result
Round to the ceil	ceil(10/3)	4
Round to the floor	floor(10/3)	3
Round to a fixed precision	round(10/3, 3)	3.333
Use universal constant	pi	3.14159…
Conditional	15 > 3 ? 1 : -1	1
Conversion	1 m to mm	1000
Conversion between unit	1 point to mm	0.352778
Test	1 point < 3 mm	TRUE
Random	rand(1, 6)	Any integer between 1 and 6
Find the greatest common divisor	gcd(20, 10, 15)	5

Compatibility

Switch 2020 Fall or higher.

Connections

One single incoming connection.
At least one single traffic light outgoing connection.

Output behavior

Case	Output level
Everything went as intended.	Success
Some private data can't be evaluated (bad syntax).	Warning
None of the private data has been evaluated (bad syntax)	Error

Flow element properties

- **Calculations**
  The private data name and the calculation to be evaluated separated by a “separator” (one per line).
  Example with "=" separator :
  - addition=1+2
  - round=round(10/3, 2)
  - day=(24 h to day) + 0.5 day

- **Separator**
  The separator is the keyword character that will be used to split the private data name and the calculation.
  Default value is “=” and it must be changed for specific operator like Inferior or Equal “<=”

- **Precision**
  The number of digits after the integer, 2 by default.
  Precision must be between 0 and 15.

  eg : 10/3 with a precision of 2 digit will be evaluated to 3.33.

Documentation

For a complete documentation, refer to the js doc : https://mathjs.org/docs/index.html

Operator

Full documentation : https://mathjs.org/docs/expressions/syntax.html#operators

Operator	Name	Syntax	Associativity	Example	Result
+	Add	x + y	Left to right	4 + 5	9
+	Unary plus	+y	Right to left	4	4
-	Subtract	x - y	Left to right	7 - 3	4
-	Unary minus	-y	Right to left	-4	-4
*	Multiply	x * y	Left to right	2 * 3	6
.*	Element-wise multiply	x .* y	Left to right	[1,2,3] .* [1,2,3]	[1,4,9]
/	Divide	x / y	Left to right	6 / 2	3
./	Element-wise divide	x ./ y	Left to right	[9,6,4] ./ [3,2,2]	[3,3,2]
%	Percentage	x%	None	8%	0.08
%	Addition with Percentage	x + y%	Left to right	100 + 3%	103
%	Subtraction with Percentage	x - y%	Left to right	100 - 3%	97
% mod	Modulus	x % y	Left to right	8 % 3	2
^	Power	x ^ y	Right to left	2 ^ 3	8
.^	Element-wise power	x .^ y	Right to left	[2,3] .^ [3,3]	[8,27]
'	Transpose	y'	Left to right	[[1,2],[3,4]]'	[[1,3],[2,4]]
!	Factorial	y!	Left to right	5!	120
&	Bitwise and	x & y	Left to right	5 & 3	1
~	Bitwise not	~x	Right to left	~2	-3
|	Bitwise or	x | y	Left to right	5 | 3	7
^|	Bitwise xor	x ^| y	Left to right	5 ^| 2	7
<<	Left shift	x << y	Left to right	4 << 1	8
>>	Right arithmetic shift	x >> y	Left to right	8 >> 1	4
>>>	Right logical shift	x >>> y	Left to right	-8 >>> 1	2147483644
and	Logical and	x and y	Left to right	true and false	FALSE
not	Logical not	not y	Right to left	not true	FALSE
or	Logical or	x or y	Left to right	true or false	TRUE
xor	Logical xor	x xor y	Left to right	true xor true	FALSE

=	Assignment	x = y	Right to left	a = 5	5
? :	Conditional expression	x ? y : z	Right to left	15 > 100 ? 1 : -1	-1
:	Range	x : y	Right to left	1:4	[1,2,3,4]
to, in	Unit conversion	x to y	Left to right	2 inch to cm	5.08 cm
==	Equal	x == y	Left to right	2 == 4 - 2	TRUE
!=	Unequal	x != y	Left to right	2 != 3	TRUE
<	Smaller	x < y	Left to right	2 < 3	TRUE
>	Larger	x > y	Left to right	2 > 3	FALSE
<=	Smallereq	x <= y	Left to right	4 <= 3	FALSE
>=	Largereq	x >= y	Left to right	2 + 4 >= 6	TRUE

Constant

Full documentation : https://mathjs.org/docs/reference/constants.html

Constant	Description
e, E	Euler’s number, the base of the natural logarithm.
i	Imaginary unit, defined as i * i = -1. A complex number is described as a + b * i, where a is the real part, and b is the imaginary part.
Infinity	Infinity, a number which is larger than the maximum number that can be handled by a floating point number.
LN2	Returns the natural logarithm of 2.
LN10	Returns the natural logarithm of 10.
LOG2E	Returns the base-2 logarithm of E.
LOG10E	Returns the base-10 logarithm of E.
NaN	Not a number.
null	Value null.
phi	Phi is the golden ratio. Two quantities are in the golden ratio if their ratio is the same as the ratio of their sum to the larger of the two quantities. Phi is defined as (1 + sqrt(5)) / 2
pi, PI	The number pi is a mathematical constant that is the ratio of a circle's circumference to its diameter.
SQRT1_2	Returns the square root of 1/2.
SQRT2	Returns the square root of 2.
tau	Tau is the ratio constant of a circle's circumference to radius, equal to 2 * pi.
undefined	An undefined value. Preferably, use null to indicate undefined values.

Units

Full documentation : https://mathjs.org/docs/datatypes/units.html#reference

points, point, pts and pt are not present in the online documentation but are implemented.

Base	Unit
Length	meter (m), inch (in), foot (ft), yard (yd), mile (mi), link (li), rod (rd), chain (ch), angstrom, mil, point (pts, pt, points, point)
Surface area	m2, sqin, sqft, sqyd, sqmi, sqrd, sqch, sqmil, acre, hectare
Volume	m3, litre (l, L, lt, liter), cc, cuin, cuft, cuyd, teaspoon, tablespoon
Liquid volume	minim (min), fluiddram (fldr), fluidounce (floz), gill (gi), cup (cp), pint (pt), quart (qt), gallon (gal), beerbarrel (bbl), oilbarrel (obl), hogshead, drop (gtt)
Angles	rad (radian), deg (degree), grad (gradian), cycle, arcsec (arcsecond), arcmin (arcminute)
Time	second (s, secs, seconds), minute (mins, minutes), hour (h, hr, hrs, hours), day (days), week (weeks), month (months), year (years), decade (decades), century (centuries), millennium (millennia)
Frequency	hertz (Hz)
Mass	gram(g), tonne, ton, grain (gr), dram (dr), ounce (oz), poundmass (lbm, lb, lbs), hundredweight (cwt), stick, stone
Electric current	ampere (A)
Temperature	kelvin (K), celsius (degC), fahrenheit (degF), rankine (degR)
Amount of substance	mole (mol)
Luminous intensity	candela (cd)
Force	newton (N), dyne (dyn), poundforce (lbf), kip
Energy	joule (J), erg, Wh, BTU, electronvolt (eV)
Power	watt (W), hp
Pressure	Pa, psi, atm, torr, bar, mmHg, mmH2O, cmH2O
Electricity and magnetism	ampere (A), coulomb (C), watt (W), volt (V), ohm, farad (F), weber (Wb), tesla (T), henry (H), siemens (S), electronvolt (eV)
Binary	bits (b), bytes (B)

Arithmetic functions

Full documentation : https://mathjs.org/docs/reference/functions.html#arithmetic-functions

When the documentation mentions "math.XXX()" just ignore the "math." part.

Function	Description
abs(x)	Calculate the absolute value of a number.
cbrt(x [, allRoots])	Calculate the cubic root of a value.
ceil(x)	Round a value towards plus infinity If x is complex, both real and imaginary part are rounded towards plus infinity.
dotDivide(x, y)	Divide two matrices element wise.
dotMultiply(x, y)	Multiply two matrices element wise.
dotPow(x, y)	Calculates the power of x to y element wise.
exp(x)	Calculate the exponent of a value.
expm1(x)	Calculate the value of subtracting 1 from the exponential value.
fix(x)	Round a value towards zero.
floor(x)	Round a value towards minus infinity.
gcd(a, b)	Calculate the greatest common divisor for two or more values or arrays.
hypot(a, b, …)	Calculate the hypotenusa of a list with values.
invmod(a, b)	Calculate the (modular) multiplicative inverse of a modulo b.
lcm(a, b)	Calculate the least common multiple for two or more values or arrays.
log(x [, base])	Calculate the logarithm of a value.
log10(x)	Calculate the 10-base logarithm of a value.
log1p(x)	Calculate the logarithm of a value+1.
log2(x)	Calculate the 2-base of a value.
mod(x, y)	Calculates the modulus, the remainder of an integer division.
multiply(x, y)	Multiply two or more values, x * y.
norm(x [, p])	Calculate the norm of a number, vector or matrix.
nthRoot(a)	Calculate the nth root of a value.
nthRoots(x)	Calculate the nth roots of a value.
round(x [, n])	Round a value towards the nearest integer.
xgcd(a, b)	Calculate the extended greatest common divisor for two values.

Probability functions

Full documentation : https://mathjs.org/docs/reference/functions.html#probability-functions

When the documentation mentions "math.XXX()" just ignore the "math." part.

Function	Description
combinations(n, k)	Compute the number of ways of picking k unordered outcomes from n possibilities.
combinationsWithRep(n, k)	Compute the number of ways of picking k unordered outcomes from n possibilities, allowing individual outcomes to be repeated more than once.
factorial(n)	Compute the factorial of a value Factorial only supports an integer value as argument.
gamma(n)	Compute the gamma function of a value using Lanczos approximation for small values, and an extended Stirling approximation for large values.
kldivergence(x, y)	Calculate the Kullback-Leibler (KL) divergence between two distributions.
multinomial(a)	Multinomial Coefficients compute the number of ways of picking a1, a2, .
permutations(n [, k])	Compute the number of ways of obtaining an ordered subset of k elements from a set of n elements.
pickRandom(array)	Random pick one or more values from a one dimensional array.
random([min, max])	Return a random number larger or equal to min and smaller than max using a uniform distribution.
randomInt([min, max])	Return a random integer number larger or equal to min and smaller than max using a uniform distribution.

Statistics functions

Full documentation : https://mathjs.org/docs/reference/functions.html#statistics-functions

When the documentation mentions “math.XXX()” just ignore the “math.” part.

Function	Description
cumsum(a, b, c, …)	Compute the cumulative sum of a matrix or a list with values.
mad(a, b, c, …)	Compute the median absolute deviation of a matrix or a list with values.
max(a, b, c, …)	Compute the maximum value of a matrix or a list with values.
mean(a, b, c, …)	Compute the mean value of matrix or a list with values.
median(a, b, c, …)	Compute the median of a matrix or a list with values.
min(a, b, c, …)	Compute the minimum value of a matrix or a list of values.
mode(a, b, c, …)	Computes the mode of a set of numbers or a list with values(numbers or characters).
prod(a, b, c, …)	Compute the product of a matrix or a list with values.
quantileSeq(A, prob[, sorted])	Compute the prob order quantile of a matrix or a list with values.
std(a, b, c, …)	Compute the standard deviation of a matrix or a list with values.
sum(a, b, c, …)	Compute the sum of a matrix or a list with values.
variance(a, b, c, …)	Compute the variance of a matrix or a list with values.