Adding Mathematical Functions to Expressions

You can incorporate mathematical functions into parameters. For example, you might negate an expression in order to invert a tracking curve which you wish to use to stabilize an element (such an expression might resemble the following: **-(Transform1.translate.x)**).

You can also rely on a function to add more complex mathematical operation to your expressions. The table below list all the functions which you may incorporate into Nuke expressions

Function	Purpose	Operator Usage	Related Functions
abs (x)	Returns the absolute value of the floating-point number x.	x	See also: fabs.
acos (x)	Calculates the arc sine of x; that is the value whose sine is x.	If x is less than -1 or greater 1, asin returns nan (not a number).	See also: cos, cosh, asin, atan.
asin (x)	Calculates the arc sine of x; that is the value whose sine is x.	If x is less than -1 or greater 1, asin returns nan (not a number)>	See also: sin, sinh, acos, atan.
atan (x)	Calculates the arc tangent of x; that is the value whose tangent is x. The return value will be between -PI/2 and PI/2.	x	See also: tan, tanh, acos, asin, atan2.
atan2 (x, y)	Calculates the arc tangent of the two variables x and y. This function is useful to calculate the angle between to vectors.	x, y	See also: sin, cos, tan, asin, acos, atan, hypot.
ceil (x)	Round x up to the nearest integer.	x	See also: floor, trunc, rint.
clamp (x, min, max)	Return x clamped to [min ... max]	x, min, max	See also: min, max.
clamp (x)	Return x clamped to [0.0 ... 1.0]	x	See also: min, max.
cos (x)	Returns the cosine of x	x in radians	See also: acos, sin, tan, cosh.
cosh (x)	Returns the hyperbolic cosine of x, which is defined mathematically as (exp(x) + exp(-x)) / 2.	x	See also: cos, acos, sinh, tanh.

curve (frame)	Returns the y value of the animation curve at the given frame	optional: frame, defaults to current frame	See also: value, y.
degrees (x)	Convert the angle x from radians into degrees	x	See also: radians.
exp (x)	Returns the value of e (the base of natural logarithms) raised to the power of x.	x	See also: log, log10.
exponent (x)	Exponent of x.	x	See also: mantissa, ldexp.
fBm (x, y, z, octaves, lucanarity, gain)	Fractional Brownian Motion. This is the sum of octaves calls to noise(). For each of them the input point is multiplied by pow(lacunarity,i) and the result is multiplied by pow(gain,i). For normal use, lacunarity should be greater than 1 and gain should be less than 1.	x, y, z, octaves, lacunarity, gain	See also: noise, random, turbulence.
fabs (x)	Returns the absolute value of the floating-point number x.	x	See also: abs.
false ()	Always returns 0		See also: true.
floor (x)	Round x down to the nearest integer.	x	See also: ceil, trunc, rint.
fmod (x, y)	Computes the remainder of dividing x by y. The return value is x - n y, where n is the quotient of x / y, rounded towards zero to an integer.	x, y	See also: ceil, floor.
frame ()	Return the current frame number.		See also: x.
from_byte (color component)	Converts an sRGB pixel value to a linear value.	color_component	See also: to_sRGB, to_rec709f, from_rec709f.
from_rec709f (color component)	Converts a rec709 byte value to a linear brightness	color_component	See also: form_sRGB, to_rec709f.
from_sRGB (color component)	Converts an sRGB pixel value to a linear value.	color_component	See also: to_sRGB, to_rec709f, from_rec709f.
hypot (x, y)	Returns the sqrt(x*x + y*y). This is the length of the hypotenuse of a right-angle triangle with sides of length x and y.	x, y	See also: atan2.

int (x)	Round x to the nearest integer not larger in absolute value.	x	See also: ceil, floor, trunc, rint.
ldexp (x)	Returns the result of multiplying the floating-point number x by 2 raised to the power exp.	x, exp	See also: exponent.
lerp (a, b, x)	Returns a point on the line f(x) where f(0)==a and f(1)==b. Matches the lerp function in other shading languages.	a, b, x	See also: step, smoothstep.
log (x)	Returns the natural logarithm of x.	x	See also: log10, exp.
log10 (x)	Returns the base-10 logarithm of x.	x	See also: log, exp.
logb (x)	same as exponent()	x	See also: mantissa, exponent.
mantissa (x)	Returns the normalized fraction. If the argument x is not zero, the normalized fraction is x times a power of two, and is always in the range 1/2 (inclusive) to 1 (exclusive). If x is zero, then the normalized fraction is zero and exponent() Returns zero.	x	See also: exponent
max (x, y, ...)	return the greatest of all values	x, y, (...)	See also: min, clamp.
min (x, y, ...)	return the smallest of all values	x, y, (...)	See also: max, clamp
mix (a, b, x)	same as lerp()	a, b, x	See also: step, smoothstep, lerp
noise (x, y, z)	creates a 3D Perlin noise value. This produces a signed range centerd on zero. The absolute maximum range is from -1.0 to 1.0. This produces zero at all integers, so you should rotate the coordinates somewhat (add a fraction of y and z to x, etc.) if you want to use this for random number generation.	x, optional y, optional z	See also: random, fBm, turbulence
pi ()	Returns the value for pi (3.141592654...)		
pow (x, y)	Returns the value of x raised to the power of y.	x, y	See also: log, exp, pow

pow2 (x)	Returns the value of x raised to the power of 2.	x, y	See also: pow
radians (x)	convert the angle x from degrees into radians	x	See also: degrees
random (x, y, z)	creates a pseudo random value between 0 and 1. It will always generate the same value for the same x, y and z. Calling random with no arguments will create a different value on every invocation.	optional x, optional y, optional z	See also: noise, fBm, turbulence
rint (x)	Round x to the nearest integer.	x	See also: ceil, floor, int, trunc
sin (x)	Returns the sine of x	x in radians	See also: asin, cos, tan, sinh
sinh (x)	Returns the hyperbolic sine of x, which is defined mathematically as (exp(x) - exp(-x)) / 2.	x	See also: sin, asin, cosh, tanh
smoothstep (a, b, x)	Returns 0 if x is less than a, returns 1 if x is greater or equal to b, returns a smooth cubic interpolation otherwise. Matches the smoothstep function in other shading languages.	a, b, x	See also: step, lerp
sqrt (x)	Returns the non-negative square root of x.	x	See also: pow, pow2
step (a, x)	Returns 0 if x is less than a, returns 1 otherwise. Matches the step function other shading languages.	a, x	See also: smoothstep, lerp
tan (x)	Returns the tangent of x	x in radians	See also: atan, cos, sin, tanh, atan2
tanh (x)	Returns the hyperbolic tangent of x, which is defined mathematically as sinh(x) / cosh(x).	x	See also: tan, atan, sinh, cosh
to_byte (color component)	Converts a floating point pixel value to an 8-bit value that represents that number in sRGB space.	color_component	See also: form_sRGB, to_rec709f, from_rec709f
to_rec709f (color component)	Converts a floating point pixel value to an 8-bit value that represents that brightness in the rec709 standard when that standard is mapped to the 0-255 range.	color_component	See also: form_sRGB, from_rec709f

to_sRGB (color component)	Converts a floating point pixel value to an 8-bit value that represents that number in sRGB space.	color_component	See also: form_sRGB, to_rec709f, from_rec709f
true ()	Always Returns 1	See also: false	
trunc (x)	Round x to the nearest integer not larger in absolute value.	x	See also: ceil, floor, int, rint
turbulence (x, y, z, octaves, lucanarity, gain)	This is the same as fBm() except the absolute value of the noise() function is used.	x, y, z, octaves, lucanarity, gain	See also: fBm, noise, random
value (frame)	Evaluates the y value for an animation at the given frame.	optional: frame, defaults to current frame	See also: y, curve
x ()	Return the current frame number.		See also: frame
y (frame)	Evaluates the y value for an animation at the given frame	optional: frame, defaults to current frame	See also: value, curve

.