Mathematical tools

A set of mathematical functions that can be used in expressions.

Expressions

Expression	Description	
`AngleBetweenPositions(number, number, number, number)`	Compute the angle between two positions (in degrees).	
	number	First point X position
	number	First point Y position
	number	Second point X position
	number	Second point Y position
`AngleDifference(number, number)`	Difference between two angles	
	number	First angle, in degrees
	number	Second angle, in degrees
`DistanceBetweenPositions(number, number, number, number)`	Compute the distance between two positions.	
	number	First point X position
	number	First point Y position
	number	Second point X position
	number	Second point Y position
`Pi()`	The number Pi (3.1415...)	
`Random(number)`	Random integer	
	number	Maximum value
`RandomFloat(number)`	Random float	
	number	Maximum value
`RandomFloatInRange(number, number)`	Random float in range	
	number	Minimum value
	number	Maximum value
`RandomInRange(number, number)`	Random integer in range	
	number	Minimum value
	number	Maximum value
`RandomWithStep(number, number, number)`	Random value in steps	
	number	Minimum value
	number	Maximum value
	number	Step
`XFromAngleAndDistance(number, number)`	Compute the X position when given an angle and distance relative to the origin (0;0). This is also known as getting the cartesian coordinates of a 2D vector, using its polar coordinates.	
	number	Angle, in degrees
	number	Distance

Expression	Description	
`YFromAngleAndDistance(number, number)`	Compute the Y position when given an angle and distance relative to the origin (0;0). This is also known as getting the cartesian coordinates of a 2D vector, using its polar coordinates.	
	number	Angle, in degrees
	number	Distance
`abs(number)`	Absolute value	
	number	Expression
`acos(number)`	Arccosine, return an angle (in radian). `ToDeg` allows to convert it to degrees.	
	number	Expression
`acosh(number)`	Hyperbolic arccosine	
	number	Expression
`asin(number)`	Arcsine, return an angle (in radian). `ToDeg` allows to convert it to degrees.	
	number	Expression
`asinh(number)`	Arcsine	
	number	Expression
`atan(number)`	Arctangent, return an angle (in radian). `ToDeg` allows to convert it to degrees.	
	number	Expression
`atan2(number, number)`	2 argument arctangent (atan2)	
	number	Y
	number	X
`atanh(number)`	Hyperbolic arctangent	
	number	Expression
`cbrt(number)`	Cube root	
	number	Expression
`ceil(number)`	Round number up to an integer	
	number	Expression
`ceilTo(number, number)`	Round number up to the Nth decimal place	
	number	Expression
	number	Expression *Optional*.
`clamp(number, number, number)`	Restrict a value to a given range	
	number	Value
	number	Min
	number	Max
`cos(number)`	Cosine of an angle (in radian). If you want to use degrees, use`ToRad`: `sin(ToRad(45))`.	
	number	Expression
`cosh(number)`	Hyperbolic cosine	
	number	Expression
`cot(number)`	Cotangent of a number	
	number	Expression

Expression	Description	
`csc(number)`	Cosecant of a number	
	number	Expression
`exp(number)`	Exponential of a number	
	number	Expression
`floor(number)`	Round number down to an integer	
	number	Expression
`floorTo(number, number)`	Round number down to the Nth decimal place	
	number	Expression
	number	Expression *Optional*.
`lerp(number, number, number)`	Linearly interpolate a to b by x	
	number	a (in a+(b-a) * x)
	number	b (in a+(b-a) * x)
	number	x (in a+(b-a) * x)
`log(number)`	Logarithm	
	number	Expression
`log10(number)`	Base-10 logarithm	
	number	Expression
`log2(number)`	Base 2 Logarithm	
	number	Expression
`max(number, number)`	Maximum of two numbers	
	number	First expression
	number	Second expression
`min(number, number)`	Minimum of two numbers	
	number	First expression
	number	Second expression
`mod(number, number)`	x mod y	
	number	x (as in x mod y)
	number	y (as in x mod y)
`normalize(number, number, number)`	Remap a value between 0 and 1.	
	number	Value
	number	Min
	number	Max
`nthroot(number, number)`	Nth root of a number	
	number	Number
	number	N
`pow(number, number)`	Raise a number to power n	
	number	Number
	number	The exponent (n in “x to the power n”)
`round(number)`	Round a number	
	number	Expression
`roundTo(number, number)`	Round a number to the Nth decimal place	
	number	Expression

Expression	Description
	number Expression *Optional*.
`sec(number)`	Secant *number* Expression
`sign(number)`	Return the sign of a number (1,-1 or 0) *number* Expression
`sin(number)`	Sine of an angle (in radian). If you want to use degrees, use`ToRad`: `sin(ToRad(45))`. *number* Expression
`sinh(number)`	Hyperbolic sine *number* Expression
`sqrt(number)`	Square root of a number *number* Expression
`tan(number)`	Tangent of an angle (in radian). If you want to use degrees, use`ToRad`: `tan(ToRad(45))`. *number* Expression
`tanh(number)`	Hyperbolic tangent *number* Expression
`trunc(number)`	Truncate a number *number* Expression

---

*This page is an auto-generated reference page about the **Mathematical tools** feature of GDevelop, the open-source, cross-platform game engine designed for everyone.* Learn more about all GDevelop features here.