Scalar

The functions here all have green Scalar inputs and outputs.

Abs

In mathematics, the modulus or “absolute value” (abbreviated Abs) of a real number is its numerical value without regard to its sign. So, for example, 5 is the absolute value of both 5 and 5 and 3.14159 is the absolute value of both 3.14159 and 3.14159.

In a general frame of reference you can think of this node’s function as changing negative values into positive values or in other words inverting only the numbers that exist on the negative side of a number line. The absolute value of a number is always either positive or zero, never negative.

Add

The Add node adds the incoming or user specified value B to the incoming or user specified value A and supplies their sum as the Result output.

It is important to note the order of the values in all the math nodes as inverse operations do not always produce the same result. Addition and multiplication are exceptions but if you keep this rule throughout all of the math nodes you will run into fewer problems.

BoxStep

A box step is a ramped step function. Two limits are set by the Begin and End inputs or through the edit panel. If the input is less than the Begin input, then the output will be 0.0. Conversely, if the Begin input is greater than the End input, the output will be 1.0. If the input is between Begin and End, then the output will vary between 0.0 and 1.0 proportionally.

Effectively, the Box Step clamps between the Begin and End values and normalizes the output between 0.0 and 1.0.

Ceil

Ceil is the abbreviation for “ceiling”. This node receives scalar data from an arbitrary data item such as the luminance values of an image, the Alpha channel of a procedural or the output from another math node. It evaluates the incoming value and rounds that value up (hence the name “ceiling”) to the nearest whole number which is then output as the result.

This can be very useful for extracting stepped data from smooth gradations among many other things. See the I/O graph below to get an idea of what is going on under the hood:

Input:	0.2	0.6	1.0	1.4	1.8	2.2	2.6	3.0
Result:	1	1	1	2	2	3	3	3

Clamp

This node will clamp incoming values to a user specified cap or limiter value. There is a cap for the high limit (or top of the range) and one for the low. Values that are greater than the specified high limit will be set to the high limit value. Values that are lower than the low limit will be set to the low limit. Incoming values that are between the specified high and low limits will not be modified or scaled.

Divide

The Divide node simply divides one scalar number by another.

> A computer finds it easier to multiply than divide. Consider multiplying instead of dividing. 4 x 0.5 gives the same result as 4/2.

Floor

The Floor node finds the largest integer less than or equal to the In input. A few examples:

In:	-2.9	-2.0	-1.5	0.0	0.5	1.5	2.0	2.9
Out:	-3	-2	-2	0	0	1	2	2

The floor function can be considered as a round function much like Ceil, with the exception that it is derived from the lowest integer (the floor) rather than the highest integer (the ceiling).

Fresnel

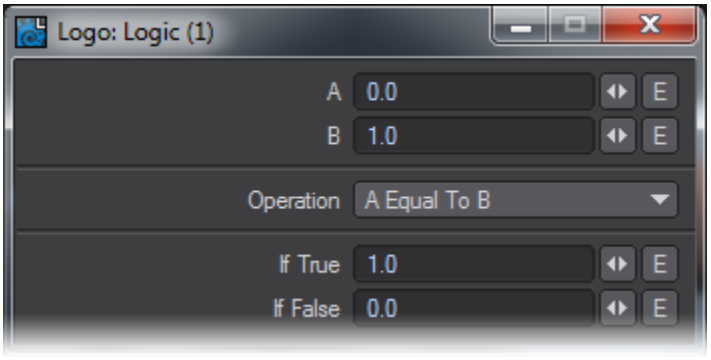
The Fresnel equations involve the amount of reflected light based on the incidence angle. You can input a Refraction index by double-clicking on the node and you should input a normal to base the math on from the Input node's Normal outputs. The Result or Inverse can be chosen as output.

Invert

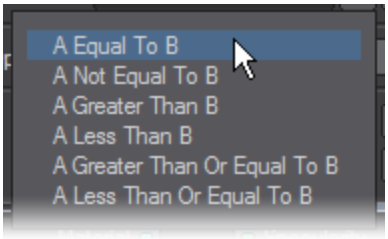
The invert node simply inverts the input. The is done by subtracting the In input from 1.0, so 0.0 will become 1.0 and 1.0 will become 0.0. The inversion will occur for intermediate values also, so 0.2 inverted is 0.8, 0.7 inverted is 0.3 and so on.

Logic

This node does have an edit panel. The Logic node can be used for typical "If, then, else" logic flow within a network. There are two inputs that can be compared in various ways. If the given comparison evaluates true then the value entered or connected to the true input is passed to the Out output. If the comparison evaluates as false then the value entered or connected to the False input is passed to the Out output.



The Out output from this node is either a pass-through of the True or False connections or the value entered into those variables via the Logic node's Edit panel. Which input True or False is passed through depends on the result of the selected Operation. The available Operations are shown here:



So for example if A Equal To B were selected the True value would be output if the vales for A and B were the same. If they were different then the value entered or connected to False would be passed to the Out output.

Max

The Max node simply takes the A and B inputs and outputs the largest. A and B may be connected to other nodes or controlled manually. This means that the node can be used to clamp the minimum value of a scalar.

Min

The Min node simply takes the A and B inputs and outputs the smallest. A and B may be connected to other nodes or controlled manually. This means that the node can be used to clamp the maximum value of a scalar.

Mod

The Mod node derives the modulus of the A input in the form Result=A mod B. The modulus is effectively the remainder leftover when A is divided by B, as in 7 divided by 2 is 3 with a remainder 1. So a few examples would be:

Input A:	0.0	1.0	2.0	3.0	4.0	5.0	6.0
Input B:	2.0	2.0	2.0	2.0	2.0	2.0	2.0
Result:	0.0	1.0	0.0	1.0	0.0	1.0	0.0

This function is useful because it can be used to repeat a regular function over a shorter period. Take, for example, a linear gradient that varies from 0.0 at the top of an object to 6.0 at the bottom. By using this gradient as the A input and making B 2.0, you can see from the above table that the gradient would ramp from 0.0 to 1.0 three times.

Multiply

The Multiply node allows one scalar to be multiplied by another.

Pow

The Pow node allows one scalar to be raised to the power of another (A Pow B or AB). For example, 23 is 2 x 2 x 2 = 8. Interestingly, you can also raise a scalar to the power of 0.5 if you wish to calculate the square root.

Sign

The Sign node simply changes the sign of the input so that positive scalars become negative and negative scalars positive.

In:	-2.0	-1.0	-0.5	0.0	0.5	1.0	2.0
Out:	2.0	1.0	0.5	0.0	-0.5	-1.0	-2.0

Smooth Step

A smooth step is a rounded, ramped step function. Two limits are set by the Begin and End inputs. If the In input is less than the Begin input, then the output will be 0.0. Conversely, if the In input is greater than the End input, then the output will be 1.0. If the input is between Begin and End, then the output will vary between 0.0 and 1.0 proportionally as a smoothed ramp. Effectively the Smooth Step clamps between the begin and end values and normalizes the output between 0.0 and 1.0 with a smoothed transition.

Subtract

The Subtract node allows one scalar to be subtracted from another.