

# Math Shaders

A collection of mathematical shaders. Math shaders can work on color or vector inputs.

## Abs

Return the **absolute value** of *input*.

## Add

Return  $input_1 + input_2$ .

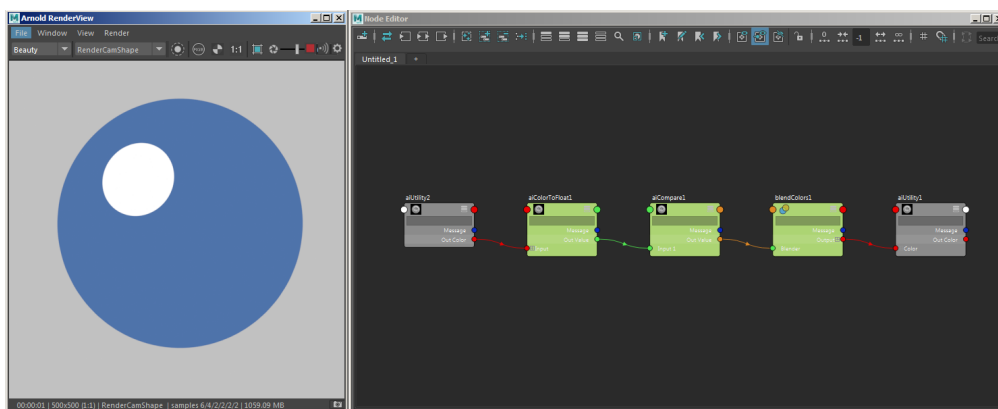
## Atan

Return the **arctangent** of  $y/x$ . The resulting value is in the range  $[-\pi/2, \pi/2]$ , using the signs of the two arguments to determine the quadrant of the result.

## Compare

Compare  $input_1$  and  $input_2$  with the following operators and return true or false:

- Equal (==)
- Not Equal (!=)
- Greater Than (>)
- Less Than (<)
- Greater Than or Equal (>=)
- Less Than or Equal (<=)



Example of a Compare shader used to create a toon shading effect

## Complement

Return **one's complement** (1 *input*). Also known as **reverse video**.

## Cross

Compute the **cross product** between two vectors, defined as the vector perpendicular to both input vectors, with its direction defined by the **right-hand rule**.

$$\mathbf{a} \times \mathbf{b} = \begin{pmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{pmatrix}$$

The length of the cross product can be interpreted geometrically as:

$$\|\mathbf{a} \times \mathbf{b}\| = \|\mathbf{a}\| \|\mathbf{b}\| \sin \theta$$

### Divide

Return  $input_1 \div input_2$ .

### Dot

Compute the **dot product** between two vectors as follows:

$$a \cdot b = a_x b_x + a_y b_y + a_z b_z$$

The result is a scalar value that can be interpreted geometrically as:

$$a \cdot b = \|a\| \|b\| \cos \theta$$

where the length of vector **a** is denoted by

**a**

and the angle between **a** and **b** is  $\theta$ .

### Exp

Return the **exponential** of input,  $e^{input}$ . This is the inverse of Ln, see also **Pow**.

### Fraction

Returns the **fractional part** of *input*. For example, an input of 123.456 would return 0.456.

### Is Finite

Return false if *input* is either **infinity** or **NaN**, and true otherwise.

### Length

Return the length of the *input* vector, with three possible distance definitions:

#### Euclidian

The "ordinary" length of the vector:  $\sqrt{x^2 + y^2 + z^2}$

## Quadrance

Euclidian distance squared, which is cheaper to compute:  $(x^2+y^2+z^2)$

## Manhattan

Measures distance following only axis-aligned directions, which is even cheaper to compute:  $(|x|+|y|+|z|)$

## Log

Return the **logarithm** of *input* to base. The argument must be greater than zero. This is the inverse of **Pow**.

## Max

Return the per-component maximum of *input*<sub>1</sub> and *input*<sub>2</sub>.

## Min

Return the per-component minimum of *input*<sub>1</sub> and *input*<sub>2</sub>.

## Modulo

Return *input* modulo *divisor*. This is the remainder of the division of *input* by *divisor*.

## Multiply

Return *input*<sub>1</sub> × *input*<sub>2</sub>.

## Negate

Return *input*.

## Normalize

Return a normalized input vector, ie. a **unit vector** pointing in the same direction.

## Pow

Return *base*<sup>*exponent*</sup>. This is the inverse of **Log**, see also **Exp**.

## Random

The Random shader outputs a random color from various types of inputs. It is useful to do variations of colors or shader properties for example.

## Reciprocal

Return the multiplicative inverse of input, ie.  $1/input$  or  $input^{-1}$ .

## Sign

- Return -1 if  $input < 0$
- Return 0 if  $input == 0$
- Return 1 if  $input > 0$

## Sqrt

Return the **square root** of  $input$ , ie.

$\sqrt{\text{input}}$

## Subtract

Return  $input_1 - input_2$ .

## Trigo

Perform various trigonometric functions on  $input$ . The *frequency* and *phase* parameters make the most sense for the sine, cosine and tangent functions, but are available on all functions for orthogonality. The *units* parameter lets you choose between **radians** and **degrees** for the argument of sine, cosine, and tangent and for the result of the inverse functions. It has no effect on the hyperbolic functions.

Function	Formula	Units Affects	Output Range
Cosine	$\cos(input \times frequency + phase)$	argument	[-1, 1]
Sine	$\sin(input \times frequency + phase)$		[-1, 1]
Tangent	$\tan(input \times frequency + phase)$		[-, ]
Arccosine	$\arccos(input \times frequency + phase)$	result	[0, ] or [0°, 180°]
Arcsine	$\arcsin(input \times frequency + phase)$		[-/2, /2] or [-90°, 90°]
Arctangent	$\arctan(input \times frequency + phase)$		[-/2, /2] or [-90°, 90°]
Hyperbolic Cosine	$\cosh(input \times frequency + phase)$	(nothing)	[1, ]
Hyperbolic Sine	$\sinh(input \times frequency + phase)$		[-, ]
Hyperbolic Tangent	$\tanh(input \times frequency + phase)$		[-1, 1]