Module
Math

The Math module contains module methods for basic trigonometric and transcendental functions. See class Float on page 528 for a list of constants that define Ruby's floating-point accuracy.

Module constants

E	An approximation of e (base of natural logarithms)
PI	An approximation of π

Module methods

acos Math.acos(*x*) → *float*

Computes the arc cosine of *x*. Returns $0..\pi$.

acosh Math.acosh(*x*) → *float*

Computes the inverse hyperbolic cosine of *x*.

asin Math.asin(*x*) → *float*

Computes the arc sine of *x*. Returns $-\frac{\pi}{2}..\frac{\pi}{2}$.

asinh Math.asinh(*x*) → *float*

Computes the inverse hyperbolic sine of *x*.

atan Math.atan(*x*) → *float*

Computes the arc tangent of *x*. Returns $-\frac{\pi}{2}..\frac{\pi}{2}$.

atanh Math.atanh(*x*) → *float*

Computes the inverse hyperbolic tangent of *x*.

atan2 Math.atan2(*y*, *x*) → *float*

Computes the arc tangent given *y* and *x*. Returns $-\pi..\pi$.

cbrt Math.cbrt(*numeric*) → *float*

1.9

Returns the cube root of *numeric*.

cos Math.cos(*x*) → *float*

Computes the cosine of *x* (expressed in radians). Returns $-1..1$.

cosh Math.cosh(*x*) → *float*

Computes the hyperbolic cosine of *x* (expressed in radians).

erf Math.erf(*x*) → *float*

Returns the error function of *x*.

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

erfc Math.erfc(*x*) → *float*

Returns the complementary error function of *x*.

$$erfc(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

exp Math.exp(*x*) → *float*

Returns e^x.

frexp Math.frexp(*numeric*) → [*fraction*, *exponent*]

Returns a two-element array containing the normalized fraction (a Float) and exponent (a Fixnum) of *numeric*.

```
fraction, exponent = Math.frexp(1234)   # =>   [0.6025390625, 11]
fraction * 2**exponent                  # =>   1234.0
```

gamma Math.gamma(*x*) → *float*

1.9 Returns the gamma function Γx. For integral *x*, the Γx approximates $factorial(x-1)$.

```
Math.gamma(2)       # =>   1.0
Math.gamma(3)       # =>   2.0
Math.gamma(4)       # =>   6.0
Math.gamma(10.34)   # =>   784993.609149316
```

hypot Math.hypot(*x*, *y*) → *float*

Returns $\sqrt{x^2+y^2}$, the hypotenuse of a right-angled triangle with sides *x* and *y*.

```
Math.hypot(3, 4)   # =>   5.0
```

ldexp Math.ldexp(*float*, *integer*) → *float*

Returns the value of *float* × $2^{integer}$.

```
fraction, exponent = Math.frexp(1234)
Math.ldexp(fraction, exponent)   # =>   1234.0
```

lgamma Math.lgamma(*x*) → [*float*, *sign*]

1.9 The first element of the returned array is the natural logarithm of the absolute value of the gamma function of *x*. The second value is −1 is the gamma function returned a negative number, +1 otherwise.

log Math.log(*numeric*) → *float*

Returns the natural logarithm of *numeric*.

log10 Math.log10(*numeric*) → *float*

Returns the base 10 logarithm of *numeric*.

log2 Math.log2(*numeric*) → *float*

1.9

Returns the base 2 logarithm of *numeric*.

sin Math.sin(*numeric*) → *float*

Computes the sine of *numeric* (expressed in radians). Returns $-1..1$.

sinh Math.sinh(*float*) → *float*

Computes the hyperbolic sine of *numeric* (expressed in radians).

sqrt Math.sqrt(*float*) → *float*

Returns the non-negative square root of *numeric*. Raises ArgError if *numeric* is less than zero.

tan Math.tan(*float*) → *float*

Returns the tangent of *numeric* (expressed in radians).

tanh Math.tanh(*float*) → *float*

Computes the hyperbolic tangent of *numeric* (expressed in radians).