

9.2. **math** — Funções matemáticas

Este módulo está sempre disponível. Ele fornece acesso a funções matemáticas definidas pelo padrão C.

Estas funções não podem ser utilizados com os números complexos; utilizar as funções com o mesmo nome do módulo `cmath` se você precisar de suporte para números complexos. A distinção entre as funções que suportam números complexos e aqueles que não é feito desde que a maioria dos usuários não querem aprender bastante, tanto a matemática como necessários para entender os números complexos. Receber uma exceção em vez de um resultado complexo permite a detecção precoce do número complexo inesperado usada como parâmetro, para que o programador pode determinar como e por que ela foi gerada em primeiro lugar.

As seguintes funções são fornecidas por este módulo. Exceto quando expressamente indicado de outra forma, todos os valores de retorno são caros alegóricos.

9.2.1. Número-teórico e representação em função

`math.ceil(x)`

Retorna o menor inteiro maior ou igual a x .

`math.copysign(x, y)`

Retorna um float com a magnitude (valor absoluto) de x , porém com o sinal de y .
Se usarmos $y = 0$, teremos: `copysign(1.0, -0.0) = -1.0` / `copysign(1.0, 0.0) = 1.0`.

`math.fabs(x)`

Retorna o valor absolute de x .

`math.factorial(x)`

Retorna x factorial. Devolve um erro se x não é inteiro, ou se x é negative.

`math.floor(x)`

Retorna o maior inteiro menor ou igual a x .

`math.fmod(x, y)`

Funciona como o operador de modulo do python, porém é melhor para lidar com números reais. O motivo é que `fmod` é definido pela plataforma da biblioteca C, na qual `fmod(x, y)` será exatamente (matematicamente; precisão infinita) igual a $x - n*y$ para um inteiro n , de tal forma que o resultado terá o mesmo sinal de x e a

magnitude menor ou igual a $\text{abs}(y)$. A operação $x\%y$ em python retorna um resultado com o sinal de y , que pode não ser computável para entradas com números reais.

`math.frexp(x)`

Retorna a mantissa e o expoente de x como o par (m, e) . m é um real e e é um inteiro, tal que $x == m * 2**e$.

`math.fsum(iterable)`

Retorna um número real preciso para a soma de valores iteráveis.

```
>>>
>>> sum([.1, .1, .1, .1, .1, .1, .1, .1, .1, .1])
0.9999999999999999
>>> fsum([.1, .1, .1, .1, .1, .1, .1, .1, .1, .1])
1.0
```

`math.isfinite(x)`

Retorna verdadeiro se x não é infinito e nem um NaN (não é um número), e falso caso contrário.

`math.isinf(x)`

Retorna verdadeiro se x é positivo ou um negativo infinito, e falso caso contrário.

`math.isnan(x)`

Retorna verdadeiro se x não é um número e falso caso contrário.

`math.ldexp(x, i)`

Retorna $x * (2**i)$.

`math.modf(x)`

Retorna a parte fracionária e a parte inteira de x . Ambos os resultados possuem o sinal de x e são reais.

`math.trunc(x)`

Retorna um valor real (da classe Real, não float) com o valor de x truncado.

9.2.2. Potência e funções logarítmicas.

Nota: e = número de Euler

`math.exp(x)`

Retorna e^{**x} .

`math.expm1(x)`

Retorna $e^{**x} - 1$. Para x pequeno é mais preciso que $\exp(x) - 1$.

```
>>>
>>> from math import exp, expm1
>>> exp(1e-5) - 1 # gives result accurate to 11 places
1.0000050000069649e-05
>>> expm1(1e-5) # result accurate to full precision
1.0000050000166668e-05
```

New in version 3.2.

`math.log(x[, base])`

Retorna, para um único argumento, o logaritmo natural de x . Com dois argumentos, retorna o logaritmo de x na base especificada.

`math.log1p(x)`

Retorna o logaritmo natural de $1 + x$. Preciso para $x \rightarrow 0$.

`math.log2(x)`

Retorna o logaritmo de x na base 2. Mais preciso que $\log(x, 2)$.

New in version 3.3.

`math.log10(x)`

Retorna o logaritmo de x na base 10. Mais preciso que $\log(x, 10)$.

`math.pow(x, y)`

Calcula x^{**y} . Para $x < 0$ e y não inteiro, devolve um erro. Converte as entradas em floats.

`math.sqrt(x)`

Retorna a raiz de x .

9.2.3. Funções Trigonométricas

`math.acos(x)`

Retorna o arcosseno de x em radianos.

`math.asin(x)`

Retorna o arcseno de x em radianos.

`math.atan(x)`

Retorna o arcotangente de x em radianos.

`math.atan2(y, x)`

Retorna o arcotangente de y/x em radianos. O sinal dos valores de entrada permitem que você determine em qual quadrante se deseja obter o resultado.

`math.cos(x)`

Retorna o cosseno de x radianos.

`math.hypot(x, y)`

Retorna a norma euclidiana, $\sqrt{x^2 + y^2}$. É o comprimento do vetor da origem até (x, y).

`math.sin(x)`

Retorna o seno de x em radianos.

`math.tan(x)`

Retorna a tangente de x em radianos.

9.2.4. Conversão angular

`math.degrees(x)`

Converte x radianos e graus.

`math.radians(x)`

Converte x graus em radianos.

9.2.5. Funções Hiperbólicas

`math.acosh(x)`

Retorna o inverso do cosseno hiperbólico de x.

`math.asinh(x)`

Retorna o inverso do seno hiperbólico de x.

`math.atanh(x)`

Retorna o inverso da tangent hiperbólica de x.

`math.cosh(x)`

Retorna o cosseno hiperbólico de x.

`math.sinh(x)`

Retorna o seno hiperbólico de x.

`math.tanh(x)`

Retorna a tangent hiperbólica de x.

9.2.6. Funções Especiais

`math.erf(x)`

Retorna o erro da função em x.

`math.erfc(x)`

Retorna o erro complementar da função em x, definida como $1 - \text{erf}(x)$, tendo mais precisão para altos valores de x.

`math.gamma(x)`

Retorna a função gamma em x.

`math.lgamma(x)`

Retorna o logaritmo natural do valor absolute da função gamma em x.

9.2.7. Constantes

`math.pi`

A constante matemática $\pi = 3.141592\dots$

`math.e`

A constante matemática $e = 2.718281\dots$