HOWTO zu den Python Mathematikfunktionen im Modul math/cmath

(C) 2021-2024 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
OSTC Open Source Training and Consulting GmbH
http://www.ostc.de

Dieses Dokument beschreibt die Mathematischen Funktionen des Moduls "math" und des Moduls "cmath" (complex math) für Komplexe Zahlen.

INHALTSVERZEICHNIS

1) Mathematische Funktionen aus Modul "math"
2) Mathematische Funktionen aus Modul "cmath" (Komplexe Zahlen)

Doku --> http://docs.python.org/3/library/math.html
http://docs.python.org/3/library/cmath.html

1) Mathematische Funktionen aus Modul "math"

exp(x)	Exponentialfunktion e^x	
expm1(x)	exp(x)-1 (vermeidet Genauigkeitsverlust für kleine x)	
exp2(x)	Zweierpotenz 2^x	3.11
log(x)	Natürlicher Logarithmus ln von x	
log(x, b)	Logarithmus von x zur Basis b	
log1p(x)	Natürlicher Logarithmus von 1+x (genau für x nahe 0)	
log10(x)	Zehnerlogarithmus log10 von x	
log2(x)	Zweierlogarithmus ld von x	
fabs(x)	Absoluter Wert der Fließkommazahl x	
factorial(x)	Fakultät x! (ValueError falls x negativ/keine Ganzzahl)	
pow(x, y)	x hoch y	
sqrt(x)	Quadratwurzel	
isqrt(n)	Ganzahliger Teil der Quadratwurzel von n	
cbrt(x)	Kubikwurzel	3.11
floor(x)	Größte ganze Zahl <= x	
ceil(x)	Kleinste ganze Zahl >= x	
sin(x) R	Sinus	
cos(x) R	Cosinus	
tan(x) R	Tangens	
acos(x) R	Arcus Cosinus (Bereich 0..pi)	
asin(x) R	Arcus Sinus (Bereich -pi/2..pi/2)	
atan(x) R	Arcus Tangens (Bereich -pi/2..pi/2)	
atan2(y, x) R	Arcus Tangens von y/x Abweichend von atan(y/x) werden Vorz. von x,y berücks.	
sinh(x)	Hyperbolischer Sinus	
cosh(x)	Hyperbolischer Cosinus	
tanh(x)	Hyperbolischer Tangens	
asinh(x)	Inverser Hyperbolischer Sinus	
acosh(x)	Inverser Hyperbolischer Cosinus	
atanh(x)	Inverser Hyperbolischer Tangens	
radians(x)	Winkel von Grad in Radiant ("Bogenmaß") umwandeln	
degrees(x)	Winkel von Radiant ("Bogenmaß") in Grad umwandeln	
gcd(*ints)	Grösster gemeinsamer Teiler (GGT)	
lcm(*ints)	Kleinstes gemeinsames Vielfaches (KGV)	
gamma(x)	Gamma Funktion	
lgamma(x)	Natürlicher Log. des Absolutwertes der Gammafkt. von x	
erf(x)	Fehlerfunktion (error function)	
erfc(x)	Komplementäre/konjugierte Fehlerfunktion	
isfinite(x)	True falls x weder Unendlich noch NaN	
isinf(x)	True falls x positiv oder negativ Unendlich	
isnan(x)	True falls x NaN (Not a Number)	

```
+--------------+-------------------------------------------------------------+
| fsum(seq)    | Genaue Fließkomma-Summe einer Sequenz von Werten            |
|              | (geht von IEEE-754 Standard Fließkommaarithmetik aus)       |
| prod(itable) | Produkt der Elemente eines Iterable (Default Start 1)       |
|              | (leer --> 1 zurück)                                         |
+--------------+-------------------------------------------------------------+
| copysign(x,  | Vorzeichen von y wird zum Vorzeichen von x                  |
         y)    | Falls "Signed zeros" möglich: copysign(1.0, -0.0) --> -1.0  |
| fmod(x, y)   | x modulo y gemäß Plattform C (kann ungleich x % y sein) ?   |
| modf(x)      | Ganzzahliger Teil + Nachkommateil von x                     |
|              | Zwei float mit dem Vorzeichen von x                         |
| frexp(x)     | Mantisse und Exponent von x als Paar (m, e)                 |
|              | (m float, e int, Zusammenhang x = m * 2.0**e)               |
| ldexp(x, i)  | Inverse Funktion von frexp() --> x * (2**i)                 |
|              | x == 0 --> m == e == 0, sonst gilt 0.5 <= abs(m) < 1.0      |
| nextafter(x, | Nächste Fließkommazahl hinter x Richtung y                  |
          y)   | --> x < y: x < erg <= y oder x > y: y <= erg < x            |
| trunc(x)     | Schneidet float x auf nächsten ganzen Wert Richtung 0 ab    |
|              | (verwendet die Magische Methode __trunc__)                  |
| ulp(x)       | Wert des kleinsten signifikanten Bits von float x           |
+--------------+-------------------------------------------------------------+
| remainder(x, | Differenz zw. x und dem nächsten ganzzahl. Vielfachen       |
          y)   | von y, d.h. (x - n*y). Falls x genau in der Mitte liegt,    |
|              | wird gerades n bevorzugt. Das Resultat ist immer exakt.     |
+--------------+-------------------------------------------------------------+
| isclose(     | Liegen die beiden float a,b nahe beieinander? --> bool      |
  a,           | rel_tol: max. Differenz relativ zur Größenordnung von a,b   |
  b,           | abs_tol: max. Differenz absolut (unabh. von Größenordnung)  |
  rel_tol=     | --> Mind. eine Differenz muss kleiner als die Tol. sein.    |
  1e-09,       | -inf, inf, NaN verhalten sich entsprechend IEEE-754 Std:    |
  abs_tol=     |   NaN ist zu nichts benachbart (auch nicht zu sich selbst)  |
  0.0)         |   inf and -inf sind nur zu sich selbst benachbart           |
+--------------+-------------------------------------------------------------+
| dist(p, q)   | Multidim. euklidischer Abstand zw. den Punkten p and q      |
|              | (Sequenz/Iterable von Koordinaten der gleichen Dimension)   |
|              | --> sqrt(sum((px - qx) ** 2 for (px, qx) in zip(p, q)))     |
+--------------+-------------------------------------------------------------+
| hypot(...)   | Multidim. euklidischer Abstand vom Ursprung zum Punkt       |
|              |    hypot(*coordinates) --> value                            |
|              | --> sqrt(sum(x**2 for x in coordinates))                    |
|              | 2-Dimensionaler Punkt (x, y) --> sqrt(x*x + y*y)            |
+--------------+-------------------------------------------------------------+
| e            | Konstante 2.718281828459045                                 |
| inf          | Konstante inf                                               |
| nan          | Konstante nan                                               |
| pi           | Konstante 3.141592653589793                                 |
| tau          | Konstante 6.283185307179586                                 |
+--------------+-------------------------------------------------------------+
| comb(n, k)   | Binomial-Koeffizient: Anz. Möglichkeiten k aus n Elementen  |
|              | ohne Wiederholung und ohne Reihenfolge auszuwählen          |
|              | --> n!/(k!*(n-k)!) falls k <= n, sonst 0                    |
|              | TypeError falls k oder n kein Integer                       |
|              | ValueError falls k oder n negativ                           |
+--------------+-------------------------------------------------------------+
| perm(n,      | Permutationen: Anzahl Möglichkeiten k aus n Elementen ohne  |
      k=None)  | Wiederholung und mit Reihenfolge auszuwählen                |
|              | --> n!/(n-k)! falls k <= n, sonst 0                         |
|              | Falls k fehlt oder None ist --> k = n + Ergebnis n!         |
|              | TypeError falls k oder n kein Integer                       |
|              | ValueError falls k oder n negativ                           |
+--------------+-------------------------------------------------------------+
    R=Ergebnis in Radiant ("Bogenmaß")
```

2) Mathematische Funktionen aus Modul "cmath" (Komplexe Zahlen)

```
+--------------+-------------------------------------------------------------+
| sin(z)      R| Sinus                                                       |
| cos(z)      R| Cosinus                                                     |
| tan(z)      R| Tangens                                                     |
| acos(z)     R| Arcus Cosinus (Bereich 0..pi)                               |
| asin(z)     R| Arcus Sinus   (Bereich -pi/2..pi/2)                         |
| atan(z)     R| Arcus Tangens (Bereich -pi/2..pi/2)                         |
+--------------+-------------------------------------------------------------+
| sinh(z)      | Hyperbolischer Sinus                                        |
```

cosh(z)	Hyperbolischer Cosinus
tanh(z)	Hyperbolischer Tangens
asinh(z)	Inverser Hyperbolischer Sinus
acosh(z)	Inverser Hyperbolischer Cosinus
atanh(z)	Inverser Hyperbolischer Tangens
exp(z)	Exponentialfunktion e^z
log(z)	Natürlicher Logarithmus ln von z
log(z, b)	Logarithmus von z zur Basis b
log10(z)	Zehnerlogarithmus log10 von z
sqrt(z)	Quadratwurzel
isfinite(z)	True falls z.real und z.img weder Unendlich noch NaN
isinf(z)	True falls z.real oder z.img positiv/negativ Unendlich
isnan(z)	True falls z.real oder z.img NaN (Not a Number)
phase(z)	Polar-Winkel phi einer komplexen Zahl
polar(z)	Konvertiert von Karthesische Koordinaten in Polar-Koord. --> (r = Abstand von 0, phi = Polarwinkel)
rect(r, phi)	Konvertiert von Polar-Koordinaten in Karthesische Koord. --> (x, y)
isclose(a, b, rel_tol= 1e-09, abs_tol= 0.0)	Liegen die beiden complex a,b nahe beieinander? --> bool rel_tol: max. Differenz relativ zur Größenordnung von a,b abs_tol: max. Differenz absolut (unabh. von Größenordnung) --> Mind. eine Differenz muss kleiner als die Tol. sein. -inf, inf, NaN verhalten sich entsprechend IEEE-754 Std: * NaN ist zu nichts benachbart (auch nicht zu sich selbst) * inf and -inf sind nur zu sich selbst benachbart
e	Konstante 2.718281828459045
inf	Konstante inf
infj	Konstante infj
nan	Konstante nan
nanj	Konstante nanj
pi	Konstante 3.141592653589793
tau	Konstante 6.283185307179586

R=Ergebnis in Radiant ("Bogenmaß")