Python math Module

Python has a built-in module that you can use for mathematical tasks.

The math module has a set of methods and constants.

Compiled By,

Md Farmanul Haque,

Technical Trainer,

GLA University,

Mathura.

Math Methods

math.acos()

Returns the arc cosine of a number

Note: The parameter passed in math.acos() must lie between -1 to 1.

```
# Import math Library
import math

# Return the arc cosine of numbers
print(math.acos(0.55))
print(math.acos(-0.55))
print(math.acos(0))
print(math.acos(1))
print(math.acos(-1))
```

Output:

```
0.9884320889261531
2.15316056466364
1.5707963267948966
0.0
3.141592653589793
```

math.acosh()

Returns the inverse hyperbolic cosine of a number

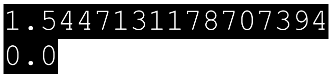
Note: The parameter passed in acosh() must be greater than or equal to 1.

```
# Import math Library
import math

# Return the inverse hyperbolic cosine of different numbers
print(math.acosh(7))
print(math.acosh(56))
print(math.acosh(2.45))
print(math.acosh(1))
```

Output:

```
2.6339157938496336
4.718419142372879
```



```
1.5447131178707394
0.0
```

math.asin()

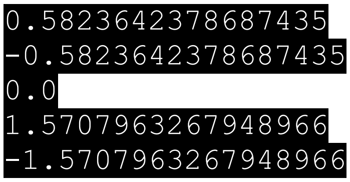
Returns the arc sine of a number

Note: The parameter passed in math.asin() must lie between -1 to 1.

```
# Import math Library
import math

# Return the arc sine of numbers
print(math.asin(0.55))
print(math.asin(-0.55))
print(math.asin(0))
print(math.asin(1))
print(math.asin(-1))
```

Output:



```
0.5823642378687435
-0.5823642378687435
0.0
1.5707963267948966
-1.5707963267948966
```

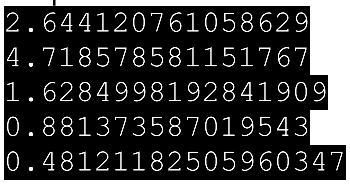
math.asinh()

Returns the inverse hyperbolic sine of a number

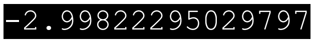
```
# Import math Library
import math

# Return the hyperbolic arc sine value of numbers
print(math.asinh(7))
print(math.asinh(56))
print(math.asinh(2.45))
print(math.asinh(1))
print(math.asinh(0.5))
print(math.asinh(-10))
```

Output:



```
2.644120761058629
4.718578581151767
1.6284998192841909
0.881373587019543
0.48121182505960347
```



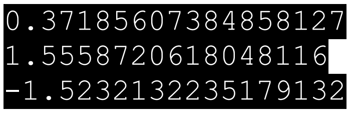
```
-2.99822295029797
```

math.atan()   Returns the arc tangent of a number in radians

```
#Import math Library
import math

#find the arctangent of some values
print (math.atan(0.39))
print (math.atan(67))
print (math.atan(-21))
```

Output:



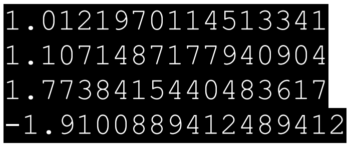
```
0.37185607384858127
1.5558720618048116
-1.5232132235179132
```

math.atan2()   Returns the arc tangent of y/x in radians

```
# Import math Library
import math

# Return the arc tangent of y/x in radians
print(math.atan2(8, 5))
print(math.atan2(20, 10))
print(math.atan2(34, -7))
print(math.atan2(-340, -120))
```

Output:



```
1.0121970114513341
1.1071487177940904
1.7738415440483617
-1.9100889412489412
```

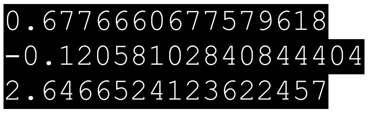
math.atanh()   Returns the inverse hyperbolic tangent of a number

Note: The parameter passed in math.atanh() must lie between -0.99 to 0.99.

```
#Import math Library
import math

#print the hyperbolic arctangent of different numbers
print (math.atanh(0.59))
print (math.atanh(-0.12))
print (math.atanh(0.99))
```

Output:



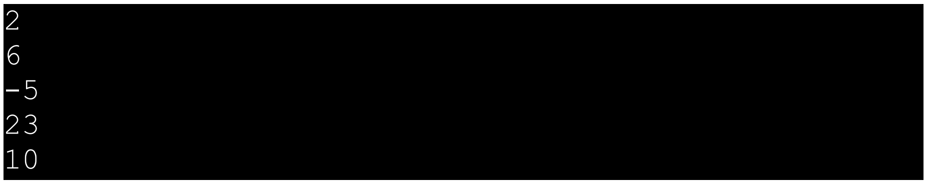
```
0.6776660677579618
-0.12058102840844404
2.6466524123622457
```

math.ceil() Rounds a number up to the nearest integer

```
#Import math library
import math

#Round a number upward to its nearest integer
print(math.ceil(1.4))
print(math.ceil(5.3))
print(math.ceil(-5.3))
print(math.ceil(22.6))
print(math.ceil(10.0))
```

Output:



```
2
6
-5
23
10
```

math.comb() Returns the number of ways to choose k items from n items without repetition and order

Note: The parameters passed in this method must be positive integers.

```
# Import math Library
import math

# Initialize the number of items to choose from
n = 7

# Initialize the number of possibilities to choose
k = 5

# Print total number of possible combinations
print (math.comb(n, k))
```

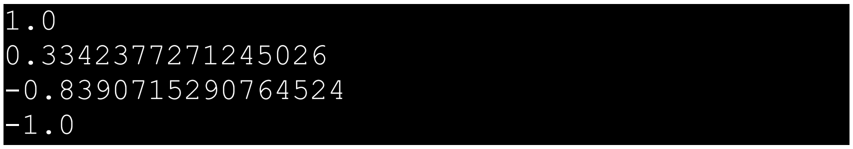
Output:
21

math.cos() Returns the cosine of a number

```
# Import math Library
import math

# Return the cosine of different numbers
print (math.cos(0.00))
print (math.cos(-1.23))
print (math.cos(10))
print (math.cos(3.14159265359))
```

Output:



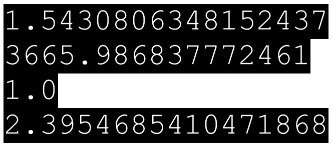
```
1.0
0.3342377271245026
-0.8390715290764524
-1.0
```

math.cosh() Returns the hyperbolic cosine of a number

```
# Import math Library
import math

# Return the hyperbolic cosine of different numbers
print (math.cosh(1))
print (math.cosh(8.90))
print (math.cosh(0))
print (math.cosh(1.52))
```

Output:



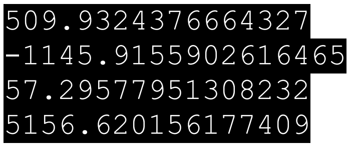
```
1.5430806348152437
3665.986837772461
1.0
2.3954685410471868
```

math.degrees() Converts an angle from radians to degrees

```
#Import math Library
import math

#Convert angles from radians to degrees:
print (math.degrees(8.90))
print (math.degrees(-20))
print (math.degrees(1))
print (math.degrees(90))
```

Output:



```
509.9324376664327
-1145.9155902616465
57.29577951308232
5156.620156177409
```

math.dist() Returns the Euclidean distance between two points (p and q), where p and q are the coordinates of that point

```
# Import math Library
import math

p = [3]
```

```
q = [1]

# Calculate Euclidean distance
print (math.dist(p, q))

p = [3, 3]
q = [6, 12]

# Calculate Euclidean distance
print (math.dist(p, q))
```

Output:

```
2.0
9.486832980505138
```

math.exp()

Returns E raised to the power of x

```
#Import math Library
import math

#find the exponential of the specified value
print(math.exp(65))
print(math.exp(-6.89))
```

Output:

```
1.6948892444103338e+28
0.0010179138409954387
```

math.expm1()

Returns E$_x$ - 1

```
#Import math Library
import math
```

Output:

```
78962960182679.69
```

```
-0.9999813562576685
```

```
#Return the exponential ex-1
print(math.expm1(32))
print(math.expm1(-10.89))
```

math.fabs() Returns the absolute value of a number

```
#Import math Library
import math

#Remove - sign of given number
print(math.fabs(-66.43))
print(math.fabs(-7))
```

Output:

```
66.43
7.0
```

math.factorial() Returns the factorial of a number

```
#Import math Library
import math

#Return factorial of a number
print(math.factorial(9))
print(math.factorial(6))
print(math.factorial(12))
```

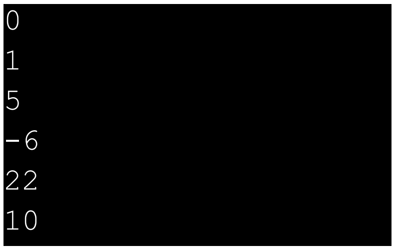
Output:

```
362880
720
479001600
```

math.floor() Rounds a number down to the nearest integer

```
#Import math library
import math
```

Output:

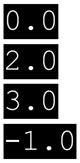


```
0
1
5
-6
22
10
```

```
# Round numbers down to the nearest integer
print(math.floor(0.6))
print(math.floor(1.4))
print(math.floor(5.3))
print(math.floor(-5.3))
print(math.floor(22.6))
print(math.floor(10.0))
```

math.fmod() Returns the remainder of x/y

Output:



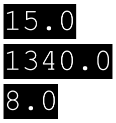
```
0.0
2.0
3.0
-1.0
```

```
# Import math Library
import math

# Return the remainder of x/y
print(math.fmod(20, 4))
print(math.fmod(20, 3))
print(math.fmod(15, 6))
print(math.fmod(-10, 3))
print(math.fmod(0, 0))
```

math.fsum() Returns the sum of all items in any iterable (tuples, arrays, lists, etc.)

Output:



```
15.0
1340.0
8.0
```

```
# Import math Library
import math

# Print the sum of all items
print(math.fsum([1, 2, 3, 4, 5]))
print(math.fsum([100, 400, 340, 500]))
print(math.fsum([1.7, 0.3, 1.5, 4.5]))
```

math.gcd() : The math.gcd() method returns the greatest common divisor of the two integers int1 and int2.

GCD is the largest common divisor that divides the numbers without a remainder.

GCD is also known as the highest common factor (HCF).

Tip: gcd(0,0) returns 0.

Example:

```
#Import math Library

import math



#find the  the greatest common divisor of the two integers

print (math.gcd(2, 6))

print (math.gcd(9, 12))

print (math.gcd(8, 36))
```

Output:

```
2

3

4
```

math.hypot() : The math.hypot() method returns the Euclidean norm. The Euclidean norm is the distance from the origin to the coordinates given.

Prior Python 3.8, this method was used only to find the hypotenuse of a right-angled triangle: sqrt(x*x + y*y).

From Python 3.8, this method is used to calculate the Euclidean norm as well. For n-dimensional cases, the coordinates passed are assumed to be like (x1, x2, x3, ..., xn). So Euclidean length from the origin is calculated by sqrt(x1*x1 + x2*x2 +x3*x3 .... xn*xn).

Example:

```
import math


#set perpendicular and base

parendicular = 13

base = 4


#print the hypotenuse of a right-angled triangle

print(math.hypot(parendicular, base))
```

Output:

```
13.6014705087354
```

math.isclose() : The math.isclose() method checks whether two values are close to each other, or not. Returns True if the values are close, otherwise False.

This method uses a relative or absolute tolerance, to see if the values are close.

Tip: It uses the following formula to compare the values: abs(a-b) <= max(rel_tol * max(abs(a), abs(b)), abs_tol)

Example:

```
#Import math Library

import math
```

```
#compare the closeness of two values

print(math.isclose(1.2900, 1.4566))

print(math.isclose(1.233, 1.233000000001))
```

Output:

```
False

True
```

math.isfinite():

Example:

```
# Import math Library

import math


# Check whether the values are finite or not

print(math.isfinite(20))

print(math.isfinite(-4.34))

print(math.isfinite(+5.34))

print(math.isfinite(math.inf))
```

Output:

```
True

True

True

False
```

math.isinf(): The math.isinf() method checks whether a number is infinite or not.

This method returns True if the specified number is a positive or negative infinity, otherwise it returns False.

Example:

```
# Import math Library

import math



# Check whether some values are infinite

print (math.isinf (116))

print (math.isinf (-5.34))

print (math.isinf (+15.34))

print (math.isinf (math.inf))
```

Output:

```
False

False

False

True
```

math.isnan(): The math.isnan() method checks whether a value is NaN (Not a Number), or not.

This method returns True if the specified value is a NaN, otherwise it returns False.

Example:

```
# Import math Library

import math


# Check whether some values are NaN

print (math.isnan (512))

print (math.isnan (math.nan))
```

Output:

```
False

True
```

math.log() : The math.log() method returns the natural logarithm of a number, or the logarithm of number to base.

Example:

```
import math


# Return the natural logarithm of different numbers

print(math.log(11.12))

print(math.log(5.3))

print(math.log(1))
```

Output:

```
2.4087452888224363

1.6677068205580761

0.0
```

math.log10(): The math.log10() method returns the base-10 logarithm of a number.

Example:

```
import math



# Return the base-10 logarithm of different numbers

print(math.log10(11.12))

print(math.log10(5.3))

print(math.log10(1))
```

Output:

```
1.0461047872460387

0.724275869600789

0.
```

math.log1p(): The math.log1p() method returns log(1+number), computed in a way that is accurate even when the value of number is close to zero.

Example:

```
# Import math Library

import math
```

```
# Return the log(1+number) for different numbers

print(math.log1p(2.7183))

print(math.log1p(2))

print(math.log1p(1))
```

Output:

```
1.3132665745863341

1.0986122886681096

0.6931471805599453
```

math.log2(): The math.log2() method returns the base-2 logarithm of a number.

Example:

```
# Import math Library

import math


# Return the base-2 logarithm of different numbers

print(math.log2(2.2))

print(math.log2(12))

print(math.log2(1))
```

Output:

```
1.1375035237499351

3.584962500721156

0.0
```

math.pow(): The math.pow() method returns the value of x raised to power y.

If x is negative and y is not an integer, it returns a ValueError.

This method converts both arguments into a float.

Tip: If we use math.pow(1.0,x) or math.pow(x,0.0), it will always returns 1.0.

Example:

```
# Import math Library

import math


# Return the value of 2 raised to the power of 5

print(math.pow(2, 5))
```

Output:

```
32.0
```

math.radians(): The math.radians() method converts a degree value into radians.

Example:

```
# Import math Library

import math
```

```
# Convert different degrees into radians

print(math.radians(90))

print(math.radians(10.03))
```

Output:

```
1.5707963267948966

0.17505652397503124
```

math.remainder(): The math.remainder() method returns the remainder of x with respect to y.

Example:

```
# Import math Library

import math


# Return the remainder of x/y

print (math.remainder(5, 2))

print (math.remainder(51, 3))

print (math.remainder(16, 4))
```

Output:

```
1.0

0.0

0.0
```

math.sin() : The math.sin() method returns the sine of a number.

Example:

```
# Import math Library
import math

# Return the sine of different numbers
print (math.sin(0.00))
print (math.sin(30))
```

Output:

```
0.0
-0.9880316240928618
```

math.sinh() : The math.sinh() method returns the hyperbolic sine of a number.

Example:

```
# Import math Library
import math

# Return the hyperbolic sine of different values
print(math.sinh(0.00))
print(math.sinh(-2.45))
```

Output:

```
0.0
```

```
-5.75102656636201
```

math.sqrt(): The math.sqrt() method returns the square root of a number.

Note: The number must be greater than or equal to 0.

Example:

```
# Import math Library
import math


# Return the square root of different numbers
print (math.sqrt(25))
print (math.sqrt(100))
print (math.sqrt(16))
```

Output:

```
5.0
10.0
4.0
```

math.tan(): The math.tan() method returns the tangent of a number.

Example:

```
# Import math Library
import math


# Return the tangent of different numbers
```

```
print (math.tan(30))

print (math.tan(-45))
```

Output:

```
-6.405331196646276

-1.6
```

math.tanh(): The math.tanh() method returns the hyperbolic tangent of a number.

Example:

```
# Import math Library

import math



# Return the hyperbolic tangent of different numbers

print(math.tanh(0))

print(math.tanh(9))

print(math.tanh(-6.28))
```

Output:

```
0.0

0.999999969540041

-0.99
```

math.trunc(): The math.trunc() method returns the truncated integer part of a number.

Note: This method will NOT round the number up/down to the nearest integer, but simply remove the decimals.

Example:

```
# Import math Library

import math



# Return the truncated integer parts of different numbers

print(math.trunc(2.34))

print(math.trunc(8.65))

print(math.trunc(-99.29))
```

Output:

```
2

8

-99
```

Constant	**Description**
math.e	Returns Euler's number (2.7182...)
math.inf	Returns a floating-point positive infinity math.inf returns inf -math.inf returns -inf
math.nan	Returns a floating-point NaN (Not a Number) value Returns nan
math.pi	Returns PI (3.1415...)

math.tau	Returns tau (6.2831...)