

Ch03-2-Functions-Library

August 7, 2020

1 3 Python Standard Libraries

1.1 Topics

- standard libraries
- include and use libraries

1.2 3.1 Standard libraries

- Python has several standard libraries (modules) you can readily import
- one can use the names (functions and data/constants) defined in those imported modules
- list of all the Python standard libraries: <https://docs.python.org/3/library/index.html>
- syntax

```
# first import library  
import libraryName
```

```
# use data and functions provided by the library  
libraryName.data  
libraryName.function()
```

1.3 3.2 math library

<https://docs.python.org/3/library/math.html>

- an important library that provides mathematical functions
- run `help(moduleName)` to get more information about the module

```
[1]: import math
```

```
[2]: help(math)
```

Help on module math:

NAME

math

MODULE REFERENCE

<https://docs.python.org/3.7/library/math>

The following documentation is automatically generated from the Python source files. It may be incomplete, incorrect or include features that are considered implementation detail and may vary between Python implementations. When in doubt, consult the module reference at the location listed above.

DESCRIPTION

This module is always available. It provides access to the mathematical functions defined by the C standard.

FUNCTIONS

`acos(x, /)`

Return the arc cosine (measured in radians) of x.

`acosh(x, /)`

Return the inverse hyperbolic cosine of x.

`asin(x, /)`

Return the arc sine (measured in radians) of x.

`asinh(x, /)`

Return the inverse hyperbolic sine of x.

`atan(x, /)`

Return the arc tangent (measured in radians) of x.

`atan2(y, x, /)`

Return the arc tangent (measured in radians) of y/x.

Unlike `atan(y/x)`, the signs of both x and y are considered.

`atanh(x, /)`

Return the inverse hyperbolic tangent of x.

`ceil(x, /)`

Return the ceiling of x as an Integral.

This is the smallest integer $\geq x$.

`copysign(x, y, /)`

Return a float with the magnitude (absolute value) of x but the sign of y.

On platforms that support signed zeros, `copysign(1.0, -0.0)` returns -1.0.

`cos(x, /)`

Return the cosine of x (measured in radians).

`cosh(x, /)`
Return the hyperbolic cosine of x .

`degrees(x, /)`
Convert angle x from radians to degrees.

`erf(x, /)`
Error function at x .

`erfc(x, /)`
Complementary error function at x .

`exp(x, /)`
Return e raised to the power of x .

`expm1(x, /)`
Return $\exp(x)-1$.

This function avoids the loss of precision involved in the direct evaluation of $\exp(x)-1$ for small x .

`fabs(x, /)`
Return the absolute value of the float x .

`factorial(x, /)`
Find $x!$.

Raise a `ValueError` if x is negative or non-integral.

`floor(x, /)`
Return the floor of x as an `Integral`.

This is the largest integer $\leq x$.

`fmod(x, y, /)`
Return `fmod(x, y)`, according to platform C.

$x \% y$ may differ.

`frexp(x, /)`
Return the mantissa and exponent of x , as pair (m, e) .

m is a float and e is an int, such that $x = m * 2.**e$.
If x is 0, m and e are both 0. Else $0.5 \leq \text{abs}(m) < 1.0$.

`fsum(seq, /)`

Return an accurate floating point sum of values in the iterable seq.

Assumes IEEE-754 floating point arithmetic.

`gamma(x, /)`

Gamma function at x.

`gcd(x, y, /)`

greatest common divisor of x and y

`hypot(x, y, /)`

Return the Euclidean distance, $\sqrt{x^2 + y^2}$.

`isclose(a, b, *, rel_tol=1e-09, abs_tol=0.0)`

Determine whether two floating point numbers are close in value.

`rel_tol`

maximum difference for being considered "close", relative to the magnitude of the input values

`abs_tol`

maximum difference for being considered "close", regardless of the magnitude of the input values

Return True if a is close in value to b, and False otherwise.

For the values to be considered close, the difference between them must be smaller than at least one of the tolerances.

`-inf`, `inf` and `NaN` behave similarly to the IEEE 754 Standard. That is, `NaN` is not close to anything, even itself. `inf` and `-inf` are only close to themselves.

`isfinite(x, /)`

Return True if x is neither an infinity nor a NaN, and False otherwise.

`isinf(x, /)`

Return True if x is a positive or negative infinity, and False otherwise.

`isnan(x, /)`

Return True if x is a NaN (not a number), and False otherwise.

`ldexp(x, i, /)`

Return $x * (2^i)$.

This is essentially the inverse of `frexp()`.

`lgamma(x, /)`

Natural logarithm of absolute value of Gamma function at x.

`log(...)`

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

Return the logarithm of x to the given base.

If the base not specified, returns the natural logarithm (base e) of x.

`log10(x, /)`

Return the base 10 logarithm of x.

`log1p(x, /)`

Return the natural logarithm of 1+x (base e).

The result is computed in a way which is accurate for x near zero.

`log2(x, /)`

Return the base 2 logarithm of x.

`modf(x, /)`

Return the fractional and integer parts of x.

Both results carry the sign of x and are floats.

`pow(x, y, /)`

Return x^y (x to the power of y).

`radians(x, /)`

Convert angle x from degrees to radians.

`remainder(x, y, /)`

Difference between x and the closest integer multiple of y.

Return $x - n*y$ where $n*y$ is the closest integer multiple of y.

In the case where x is exactly halfway between two multiples of y, the nearest even value of n is used. The result is always exact.

`sin(x, /)`

Return the sine of x (measured in radians).

`sinh(x, /)`

Return the hyperbolic sine of x.

`sqrt(x, /)`

Return the square root of x.

`tan(x, /)`

Return the tangent of x (measured in radians).

```
tanh(x, /)
    Return the hyperbolic tangent of x.
```

```
trunc(x, /)
    Truncates the Real x to the nearest Integral toward 0.
```

```
    Uses the __trunc__ magic method.
```

DATA

```
e = 2.718281828459045
inf = inf
nan = nan
pi = 3.141592653589793
tau = 6.283185307179586
```

FILE

```
/Users/rbasnet/anaconda3/lib/python3.7/lib-
dynload/math.cpython-37m-darwin.so
```

```
[3]: num = 10.5
     # math.ceil(x) - return the ceiling (or round up) of x,
     # the smallest integer greater than or equal to x
     print(math.ceil(num))
```

11

```
[4]: # math.floor(x)
     # return the floor (or round down) of x, the largest integer less than or equal
     # ↳ to x
     print(math.floor(num))
```

10

```
[5]: # math.gcd(a, b)
     # return the greatest common divisor of the integers a and b
     # if both a and b are 0, returns 0
     print(math.gcd(0, 0))
     print(math.gcd(10, 20))
```

0

10

```
[6]: # math.pow(x, y)
     # returns x raised to the power y
     print(math.pow(2, 10))
```

1024.0

```
[7]: # math.sqrt(x, y)
      # returns the square root of x
      print(math.sqrt(100))
```

10.0

```
[8]: # math.radians(x)
      # convert and return angle x in degrees to radians
      rad = math.radians(90)
```

```
[9]: # math.sin(x)
      # return the sine of x radians
      print(math.sin(rad))
```

1.0

```
[10]: # Some constants/data defined in math module
      math.pi
```

```
[10]: 3.141592653589793
```

```
[11]: math.inf
```

```
[11]: inf
```

```
[12]: math.e
```

```
[12]: 2.718281828459045
```

1.4 3.3 Other common libraries

- all Python libraries: <https://docs.python.org/3/library/index.html>
- some libraries we'll use and you may want to explore more
- **os** - operating system related
- **time** - time access and conversion
- **random** - generate pseudo-random numbers
- **sys** - system specific data and functions
- **string** - common string operations and data

```
[ ]:
```