:mod:`math` --- Mathematical functions

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 4)

Unknown directive type "module".

```
.. module:: math
   :synopsis: Mathematical functions (sin() etc.).
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 7)

Unknown directive type "testsetup".

```
.. testsetup::

   from math import fsum
```

This module provides access to the mathematical functions defined by the C standard.

These functions cannot be used with complex numbers; use the functions of the same name from the :mod:`cmath` module if you require support for complex numbers. The distinction between functions which support complex numbers and those which don't is made since most users do not want to learn quite as much mathematics as required to understand complex numbers. Receiving an exception instead of a complex result allows earlier detection of the unexpected complex number used as a parameter, so that the programmer can determine how and why it was generated in the first place.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 16); *backlink*

Unknown interpreted text role "mod".

The following functions are provided by this module. Except when explicitly noted otherwise, all return values are floats.

Number-theoretic and representation functions

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 32)

Unknown directive type "function".

```
.. function:: ceil(x)

   Return the ceiling of *x*, the smallest integer greater than or equal to *x*.
   If *x* is not a float, delegates to :meth:`x.__ceil__ <object.__ceil__>`,
   which should return an :class:`~numbers.Integral` value.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 39)

Unknown directive type "function".

```
.. function:: comb(n, k)

   Return the number of ways to choose *k* items from *n* items without repetition
   and without order.

   Evaluates to ``n! / (k! * (n - k)!)`` when ``k <= n`` and evaluates
   to zero when ``k > n``.
   Also called the binomial coefficient because it is equivalent
   to the coefficient of k-th term in polynomial expansion of the
   expression ``(1 + x) ** n``.

   Raises :exc:`TypeError` if either of the arguments are not integers.
   Raises :exc:`ValueError` if either of the arguments are negative.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 57)

Unknown directive type "function".

```
.. function:: 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*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 64)

Unknown directive type "function".

```
.. function:: fabs(x)

   Return the absolute value of *x*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 69)

Unknown directive type "function".

```
.. function:: factorial(n)

   Return *n* factorial as an integer.  Raises :exc:`ValueError` if *n* is not integral or
   is negative.

   .. deprecated:: 3.9
      Accepting floats with integral values (like ``5.0``) is deprecated.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 78)

Unknown directive type "function".

```
.. function:: floor(x)

   Return the floor of *x*, the largest integer less than or equal to *x*.  If
   *x* is not a float, delegates to :meth:`x.__floor__ <object.__floor__>`, which
   should return an :class:`~numbers.Integral` value.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 85)

Unknown directive type "function".

```
.. function:: fmod(x, y)

   Return ``fmod(x, y)``, as defined by the platform C library. Note that the
   Python expression ``x % y`` may not return the same result.  The intent of the C
   standard is that ``fmod(x, y)`` be exactly (mathematically; to infinite
   precision) equal to ``x - n*y`` for some integer *n* such that the result has
   the same sign as *x* and magnitude less than ``abs(y)``.  Python's ``x % y``
```

```
      returns a result with the sign of *y* instead, and may not be exactly computable
      for float arguments. For example, ``fmod(-1e-100, 1e100)`` is ``-1e-100``, but
      the result of Python's ``-1e-100 % 1e100`` is ``1e100-1e-100``, which cannot be
      represented exactly as a float, and rounds to the surprising ``1e100``.  For
      this reason, function :func:`fmod` is generally preferred when working with
      floats, while Python's ``x % y`` is preferred when working with integers.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 100)

Unknown directive type "function".

```
.. function:: frexp(x)

   Return the mantissa and exponent of *x* as the pair ``(m, e)``.  *m* is a float
   and *e* is an integer such that ``x == m * 2**e`` exactly. If *x* is zero,
   returns ``(0.0, 0)``, otherwise ``0.5 <= abs(m) < 1``.  This is used to "pick
   apart" the internal representation of a float in a portable way.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 108)

Unknown directive type "function".

```
.. function:: fsum(iterable)

   Return an accurate floating point sum of values in the iterable.  Avoids
   loss of precision by tracking multiple intermediate partial sums::

        >>> 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

   The algorithm's accuracy depends on IEEE-754 arithmetic guarantees and the
   typical case where the rounding mode is half-even.  On some non-Windows
   builds, the underlying C library uses extended precision addition and may
   occasionally double-round an intermediate sum causing it to be off in its
   least significant bit.

   For further discussion and two alternative approaches, see the `ASPN cookbook
   recipes for accurate floating point summation
   <https://code.activestate.com/recipes/393090/>`_\.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 129)

Unknown directive type "function".

```
.. function:: gcd(*integers)

   Return the greatest common divisor of the specified integer arguments.
   If any of the arguments is nonzero, then the returned value is the largest
   positive integer that is a divisor of all arguments.  If all arguments
   are zero, then the returned value is ``0``.  ``gcd()`` without arguments
   returns ``0``.

   .. versionadded:: 3.5

   .. versionchanged:: 3.9
      Added support for an arbitrary number of arguments. Formerly, only two
      arguments were supported.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 144)

Unknown directive type "function".

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

```
   Return ``True`` if the values *a* and *b* are close to each other and
   ``False`` otherwise.

   Whether or not two values are considered close is determined according to
   given absolute and relative tolerances.

   *rel_tol* is the relative tolerance -- it is the maximum allowed difference
   between *a* and *b*, relative to the larger absolute value of *a* or *b*.
   For example, to set a tolerance of 5%, pass ``rel_tol=0.05``.  The default
   tolerance is ``1e-09``, which assures that the two values are the same
   within about 9 decimal digits.  *rel_tol* must be greater than zero.

   *abs_tol* is the minimum absolute tolerance -- useful for comparisons near
   zero. *abs_tol* must be at least zero.

   If no errors occur, the result will be:
   ``abs(a-b) <= max(rel_tol * max(abs(a), abs(b)), abs_tol)``.

   The IEEE 754 special values of ``NaN``, ``inf``, and ``-inf`` will be
   handled according to IEEE rules.  Specifically, ``NaN`` is not considered
   close to any other value, including ``NaN``.  ``inf`` and ``-inf`` are only
   considered close to themselves.

   .. versionadded:: 3.5

   .. seealso::

      :pep:`485` -- A function for testing approximate equality
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 176)

Unknown directive type "function".

```
.. function:: isfinite(x)

   Return ``True`` if *x* is neither an infinity nor a NaN, and
   ``False`` otherwise.  (Note that ``0.0`` *is* considered finite.)

   .. versionadded:: 3.2
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 184)

Unknown directive type "function".

```
.. function:: isinf(x)

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

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 190)

Unknown directive type "function".

```
.. function:: isnan(x)

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

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 195)

Unknown directive type "function".

```
.. function:: isqrt(n)

   Return the integer square root of the nonnegative integer *n*. This is the
   floor of the exact square root of *n*, or equivalently the greatest integer
   *a* such that *a*\ Â² |nbsp| â‰¤ |nbsp| *n*.

   For some applications, it may be more convenient to have the least integer
   *a* such that *n* |nbsp| â‰¤ |nbsp| *a*\ Â², or in other words the ceiling of
   the exact square root of *n*. For positive *n*, this can be computed using
   ``a = 1 + isqrt(n - 1)``.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 209)

Unknown directive type "function".

```
.. function:: lcm(*integers)

   Return the least common multiple of the specified integer arguments.
   If all arguments are nonzero, then the returned value is the smallest
   positive integer that is a multiple of all arguments.  If any of the arguments
   is zero, then the returned value is ``0``.  ``lcm()`` without arguments
   returns ``1``.

   .. versionadded:: 3.9
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 220)

Unknown directive type "function".

```
.. function:: ldexp(x, i)

   Return ``x * (2**i)``.  This is essentially the inverse of function
   :func:`frexp`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 226)

Unknown directive type "function".

```
.. function:: modf(x)

   Return the fractional and integer parts of *x*.  Both results carry the sign
   of *x* and are floats.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 232)

Unknown directive type "function".

```
.. function:: nextafter(x, y)

   Return the next floating-point value after *x* towards *y*.

   If *x* is equal to *y*, return *y*.

   Examples:

   * ``math.nextafter(x, math.inf)`` goes up: towards positive infinity.
   * ``math.nextafter(x, -math.inf)`` goes down: towards minus infinity.
   * ``math.nextafter(x, 0.0)`` goes towards zero.
   * ``math.nextafter(x, math.copysign(math.inf, x))`` goes away from zero.

   See also :func:`math.ulp`.

   .. versionadded:: 3.9
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 249)

Unknown directive type "function".

```
.. function:: perm(n, k=None)

   Return the number of ways to choose *k* items from *n* items
   without repetition and with order.

   Evaluates to ``n! / (n - k)!`` when ``k <= n`` and evaluates
   to zero when ``k > n``.

   If *k* is not specified or is None, then *k* defaults to *n*
   and the function returns ``n!``.

   Raises :exc:`TypeError` if either of the arguments are not integers.
   Raises :exc:`ValueError` if either of the arguments are negative.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 266)

Unknown directive type "function".

```
.. function:: prod(iterable, *, start=1)

   Calculate the product of all the elements in the input *iterable*.
   The default *start* value for the product is ``1``.

   When the iterable is empty, return the start value.  This function is
   intended specifically for use with numeric values and may reject
   non-numeric types.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 278)

Unknown directive type "function".

```
.. function:: remainder(x, y)

   Return the IEEE 754-style remainder of *x* with respect to *y*.  For
   finite *x* and finite nonzero *y*, this is the difference ``x - n*y``,
   where ``n`` is the closest integer to the exact value of the quotient ``x /
   y``.  If ``x / y`` is exactly halfway between two consecutive integers, the
   nearest *even* integer is used for ``n``.  The remainder ``r = remainder(x,
   y)`` thus always satisfies ``abs(r) <= 0.5 * abs(y)``.

   Special cases follow IEEE 754: in particular, ``remainder(x, math.inf)`` is
   *x* for any finite *x*, and ``remainder(x, 0)`` and
   ``remainder(math.inf, x)`` raise :exc:`ValueError` for any non-NaN *x*.
   If the result of the remainder operation is zero, that zero will have
   the same sign as *x*.

   On platforms using IEEE 754 binary floating-point, the result of this
   operation is always exactly representable: no rounding error is introduced.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 299)

Unknown directive type "function".

```
.. function:: trunc(x)

   Return *x* with the fractional part
   removed, leaving the integer part.  This rounds toward 0: ``trunc()`` is
   equivalent to :func:`floor` for positive *x*, and equivalent to :func:`ceil`
   for negative *x*. If *x* is not a float, delegates to :meth:`x.__trunc__
   <object.__trunc__>`, which should return an :class:`~numbers.Integral` value.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 307)

Unknown directive type "function".

```
.. function:: ulp(x)

   Return the value of the least significant bit of the float *x*:

   * If *x* is a NaN (not a number), return *x*.
   * If *x* is negative, return ``ulp(-x)``.
   * If *x* is a positive infinity, return *x*.
   * If *x* is equal to zero, return the smallest positive
     *denormalized* representable float (smaller than the minimum positive
     *normalized* float, :data:`sys.float_info.min <sys.float_info>`).
   * If *x* is equal to the largest positive representable float,
     return the value of the least significant bit of *x*, such that the first
     float smaller than *x* is ``x - ulp(x)``.
   * Otherwise (*x* is a positive finite number), return the value of the least
     significant bit of *x*, such that the first float bigger than *x*
     is ``x + ulp(x)``.

   ULP stands for "Unit in the Last Place".

   See also :func:`math.nextafter` and :data:`sys.float_info.epsilon
   <sys.float_info>`.

   .. versionadded:: 3.9
```

Note that :func:`frexp` and :func:`modf` have a different call/return pattern than their C equivalents: they take a single argument and return a pair of values, rather than returning their second return value through an 'output parameter' (there is no such thing in Python).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 332); ***backlink***

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 332); ***backlink***

Unknown interpreted text role "func".

For the :func:`ceil`, :func:`floor`, and :func:`modf` functions, note that *all* floating-point numbers of sufficiently large magnitude are exact integers. Python floats typically carry no more than 53 bits of precision (the same as the platform C double type), in which case any float *x* with `abs(x) >= 2**52` necessarily has no fractional bits.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 337); ***backlink***

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 337); ***backlink***

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 337); ***backlink***

Unknown interpreted text role "func".

Power and logarithmic functions

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-`

main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 347)

Unknown directive type "function".

```
.. function:: cbrt(x)

   Return the cube root of *x*.

   .. versionadded:: 3.11
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 354)

Unknown directive type "function".

```
.. function:: exp(x)

   Return *e* raised to the power *x*, where *e* = 2.718281... is the base
   of natural logarithms.  This is usually more accurate than ``math.e ** x``
   or ``pow(math.e, x)``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 361)

Unknown directive type "function".

```
.. function:: exp2(x)

   Return *2* raised to the power *x*.

   .. versionadded:: 3.11
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 368)

Unknown directive type "function".

```
.. function:: expm1(x)

   Return *e* raised to the power *x*, minus 1.  Here *e* is the base of natural
   logarithms.  For small floats *x*, the subtraction in ``exp(x) - 1``
   can result in a `significant loss of precision
   <https://en.wikipedia.org/wiki/Loss_of_significance>`_\; the :func:`expm1`
   function provides a way to compute this quantity to full precision::

      >>> 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

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 385)

Unknown directive type "function".

```
.. function:: log(x[, base])

   With one argument, return the natural logarithm of *x* (to base *e*).

   With two arguments, return the logarithm of *x* to the given *base*,
   calculated as ``log(x)/log(base)``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 393)

Unknown directive type "function".

```
.. function:: log1p(x)

   Return the natural logarithm of *1+x* (base *e*). The
   result is calculated in a way which is accurate for *x* near zero.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 399)

Unknown directive type "function".

```
.. function:: log2(x)

   Return the base-2 logarithm of *x*. This is usually more accurate than
   ``log(x, 2)``.

   .. versionadded:: 3.3

   .. seealso::

      :meth:`int.bit_length` returns the number of bits necessary to represent
      an integer in binary, excluding the sign and leading zeros.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 412)

Unknown directive type "function".

```
.. function:: log10(x)

   Return the base-10 logarithm of *x*.  This is usually more accurate
   than ``log(x, 10)``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 418)

Unknown directive type "function".

```
.. function:: pow(x, y)

   Return ``x`` raised to the power ``y``.  Exceptional cases follow
   the IEEE 754 standard as far as possible.  In particular,
   ``pow(1.0, x)`` and ``pow(x, 0.0)`` always return ``1.0``, even
   when ``x`` is a zero or a NaN.  If both ``x`` and ``y`` are finite,
   ``x`` is negative, and ``y`` is not an integer then ``pow(x, y)``
   is undefined, and raises :exc:`ValueError`.

   Unlike the built-in ``**`` operator, :func:`math.pow` converts both
   its arguments to type :class:`float`.  Use ``**`` or the built-in
   :func:`pow` function for computing exact integer powers.

   .. versionchanged:: 3.11
      The special cases ``pow(0.0, -inf)`` and ``pow(-0.0, -inf)`` were
      changed to return ``inf`` instead of raising :exc:`ValueError`,
      for consistency with IEEE 754.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 437)

Unknown directive type "function".

```
.. function:: sqrt(x)

   Return the square root of *x*.
```

Trigonometric functions

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 445)

Unknown directive type "function".

```
.. function:: acos(x)

   Return the arc cosine of *x*, in radians. The result is between ``0`` and
   ``pi``.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 451)

Unknown directive type "function".

```
.. function:: asin(x)

   Return the arc sine of *x*, in radians. The result is between ``-pi/2`` and
   ``pi/2``.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 457)

Unknown directive type "function".

```
.. function:: atan(x)

   Return the arc tangent of *x*, in radians. The result is between ``-pi/2`` and
   ``pi/2``.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 463)

Unknown directive type "function".

```
.. function:: atan2(y, x)

   Return ``atan(y / x)``, in radians. The result is between ``-pi`` and ``pi``.
   The vector in the plane from the origin to point ``(x, y)`` makes this angle
   with the positive X axis. The point of :func:`atan2` is that the signs of both
   inputs are known to it, so it can compute the correct quadrant for the angle.
   For example, ``atan(1)`` and ``atan2(1, 1)`` are both ``pi/4``, but ``atan2(-1,
   -1)`` is ``-3*pi/4``.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 473)

Unknown directive type "function".

```
.. function:: cos(x)

   Return the cosine of *x* radians.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 478)

Unknown directive type "function".

```
.. function:: dist(p, q)

   Return the Euclidean distance between two points *p* and *q*, each
   given as a sequence (or iterable) of coordinates.  The two points
   must have the same dimension.
   Roughly equivalent to::

       sqrt(sum((px - qx) ** 2.0 for px, qx in zip(p, q)))

   .. versionadded:: 3.8
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 491)

Unknown directive type "function".

```
.. function:: hypot(*coordinates)

   Return the Euclidean norm, ``sqrt(sum(x**2 for x in coordinates))``.
   This is the length of the vector from the origin to the point
   given by the coordinates.

   For a two dimensional point ``(x, y)``, this is equivalent to computing
   the hypotenuse of a right triangle using the Pythagorean theorem,
   ``sqrt(x*x + y*y)``.

   .. versionchanged:: 3.8
      Added support for n-dimensional points. Formerly, only the two
      dimensional case was supported.

   .. versionchanged:: 3.10
      Improved the algorithm's accuracy so that the maximum error is
      under 1 ulp (unit in the last place).  More typically, the result
      is almost always correctly rounded to within 1/2 ulp.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 511)

Unknown directive type "function".

```
.. function:: sin(x)

   Return the sine of *x* radians.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 516)

Unknown directive type "function".

```
.. function:: tan(x)

   Return the tangent of *x* radians.
```

Angular conversion

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 524)

Unknown directive type "function".

```
.. function:: degrees(x)

   Convert angle *x* from radians to degrees.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 529)

Unknown directive type "function".

```
.. function:: radians(x)

   Convert angle *x* from degrees to radians.
```

Hyperbolic functions

Hyperbolic functions are analogs of trigonometric functions that are based on hyperbolas instead of circles.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 541)

Unknown directive type "function".

```
.. function:: acosh(x)

   Return the inverse hyperbolic cosine of *x*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 546)

Unknown directive type "function".

```
.. function:: asinh(x)

   Return the inverse hyperbolic sine of *x*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 551)

Unknown directive type "function".

```
.. function:: atanh(x)

   Return the inverse hyperbolic tangent of *x*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 556)

Unknown directive type "function".

```
.. function:: cosh(x)

   Return the hyperbolic cosine of *x*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 561)

Unknown directive type "function".

```
.. function:: sinh(x)

   Return the hyperbolic sine of *x*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 566)

Unknown directive type "function".

```
.. function:: tanh(x)

   Return the hyperbolic tangent of *x*.
```

Special functions

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 574)

Unknown directive type "function".

```
.. function:: erf(x)

   Return the `error function <https://en.wikipedia.org/wiki/Error_function>`_ at
   *x*.

   The :func:`erf` function can be used to compute traditional statistical
   functions such as the `cumulative standard normal distribution
   <https://en.wikipedia.org/wiki/Normal_distribution#Cumulative_distribution_function>`_::

     def phi(x):
         'Cumulative distribution function for the standard normal distribution'
         return (1.0 + erf(x / sqrt(2.0))) / 2.0

   .. versionadded:: 3.2
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 590)

Unknown directive type "function".

```
.. function:: erfc(x)

   Return the complementary error function at *x*.  The `complementary error
   function <https://en.wikipedia.org/wiki/Error_function>`_ is defined as
   ``1.0 - erf(x)``.  It is used for large values of *x* where a subtraction
   from one would cause a `loss of significance
   <https://en.wikipedia.org/wiki/Loss_of_significance>`_\.

   .. versionadded:: 3.2
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 601)

Unknown directive type "function".

```
.. function:: gamma(x)

   Return the `Gamma function <https://en.wikipedia.org/wiki/Gamma_function>`_ at
   *x*.

   .. versionadded:: 3.2
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 609)

Unknown directive type "function".

```
.. function:: lgamma(x)

   Return the natural logarithm of the absolute value of the Gamma
   function at *x*.

   .. versionadded:: 3.2
```

Constants

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-`

main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 620)

Unknown directive type "data".

```
.. data:: pi

   The mathematical constant *Ï€* = 3.141592..., to available precision.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 625)

Unknown directive type "data".

```
.. data:: e

   The mathematical constant *e* = 2.718281..., to available precision.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 630)

Unknown directive type "data".

```
.. data:: tau

   The mathematical constant *Ï„* = 6.283185..., to available precision.
   Tau is a circle constant equal to 2\ *Ï€*, the ratio of a circle's circumference to
   its radius. To learn more about Tau, check out Vi Hart's video `Pi is (still)
   Wrong <https://www.youtube.com/watch?v=jG7vhMMXagQ>`_, and start celebrating
   `Tau day <https://tauday.com/>`_ by eating twice as much pie!

   .. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 641)

Unknown directive type "data".

```
.. data:: inf

   A floating-point positive infinity.  (For negative infinity, use
   ``-math.inf``.)  Equivalent to the output of ``float('inf')``.

   .. versionadded:: 3.5
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst, line 649)

Unknown directive type "data".

```
.. data:: nan

   A floating-point "not a number" (NaN) value. Equivalent to the output of
   ``float('nan')``. Due to the requirements of the `IEEE-754 standard
   <https://en.wikipedia.org/wiki/IEEE_754>`_, ``math.nan`` and ``float('nan')`` are
   not considered to equal to any other numeric value, including themselves. To check
   whether a number is a NaN, use the :func:`isnan` function to test
   for NaNs instead of ``is`` or ``==``.
   Example::

      >>> import math
      >>> math.nan == math.nan
      False
      >>> float('nan') == float('nan')
      False
      >>> math.isnan(math.nan)
      True
      >>> math.isnan(float('nan'))
      True
```

```
.. versionchanged:: 3.11
   It is now always available.

.. versionadded:: 3.5
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 675)

Unknown directive type "impl-detail".

```
.. impl-detail::

   The :mod:`math` module consists mostly of thin wrappers around the platform C
   math library functions.  Behavior in exceptional cases follows Annex F of
   the C99 standard where appropriate.  The current implementation will raise
   :exc:`ValueError` for invalid operations like ``sqrt(-1.0)`` or ``log(0.0)``
   (where C99 Annex F recommends signaling invalid operation or divide-by-zero),
   and :exc:`OverflowError` for results that overflow (for example,
   ``exp(1000.0)``).  A NaN will not be returned from any of the functions
   above unless one or more of the input arguments was a NaN; in that case,
   most functions will return a NaN, but (again following C99 Annex F) there
   are some exceptions to this rule, for example ``pow(float('nan'), 0.0)`` or
   ``hypot(float('nan'), float('inf'))``.

   Note that Python makes no effort to distinguish signaling NaNs from
   quiet NaNs, and behavior for signaling NaNs remains unspecified.
   Typical behavior is to treat all NaNs as though they were quiet.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)math.rst`, line 694)

Unknown directive type "seealso".

```
.. seealso::

   Module :mod:`cmath`
      Complex number versions of many of these functions.
```