Math & Cmath

Math provides many mathematical functions and constants like e and pi. Cmath provides the complex number versions of those functions. Import and print math.\_\_all\_\_ or cmath.\_\_all\_\_ to see every provided function if you don’t find the one you’re looking for in this shortened list.

To write complex numbers for use with cmath, use ‘j’ for the imaginary part, such as: -1.3 + 1j. Note that you can’t just use ‘j’ by itself. You must specify 1j if that is what you want.

**General:**

**degrees(x)/radians(x):** Converts from radians to degrees, and vice-versa.

**dist(p, q):** Returns the distance between p and q. Inputs are iterables representing coordinates.

**sin(x)/cos(x)/tan(x):** Basic trig functions.

**asin(x)/acos(x)/atan(x):** Inverses of basic trig functions.

**exp(x):** Returns e^x.

**pow(x, y):** Returns x^y. Nearly the same as doing x \*\* y, but pow will always return a float.

**log(x, [base]):** Returns log base e of x. You can optionally specify your own base to use.

**factorial(x):** Returns the factorial of x (x!), x must be an int.

**comb(n, k)/perm(n, k):** Returns the number of ways to choose k items from n items (no repetition). Order is ignored for comb, but not ignored for perm.

**Other:**

**ceil(x)/floor(x)/trunc(x):** Round towards +inf, -inf, and 0, respectively.

**isclose(a, b, rel\_tol=1e-09):** Returns whether the input floats are within rel\_tol of each other (this is a good substitute for == when using floats, since floats are often close but not equal).

**lcm(\*integers)/gcd(\*integers):** Returns the least common multiple/greatest common denominator, respectively, of the arguments. Any number of arguments are allowed.

**fmod(x, y):** Modulo like x % y, but the remainder will be a float (no rounding). Uses C standard for modulo, so negatives will act differently than the Python modulo operation.

**modf(x):** Not to be confused with fmod. Returns the fractional and integer parts of x as a tuple.

**Complex Numbers:**

**phase(x):** Returns the angle the complex number makes with the positive x axis.

**polar(x)/rect(r, phi):** Conversions between polar coordinates and rectangular coordinates.