Common Python Commands

Useful Python functions, adapted from Python documentation at http://docs.python.org/2/.

a+b
Return the sum of a and b

a-b
Return the difference between a and b

a*b
Return the product of a and b

a/b
Return the result of dividing a and b
If a and b are integers, this result will be an integer, and represent the number of times b fits into a

1.0*a/b
Return the value of a divided by b as a double (decimal)

a**b
Return the value of a raised to the b

((a)**(b))**(1.0/c)
Return the value of a raised to the b divided by c ($a^{b/c} = \sqrt[c]{a^b}$)

a += b
Changes a to the value of a + b. This is equivalent to a = a + b

list1[i]
Return the i th item in a list named list1

len(s)
Return the length (the number of items) of an object.
The argument may be a sequence (string, tuple or list) or a mapping (dictionary).

Common mathematical functions in Python from http://docs.python.org/2/library/math.html.

math.exp(x)
Returns e**x.

math.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).

math.log10(x)

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

math.pow(x, y)

Return x raised to the power y. Exceptional cases follow Annex 'F' of the C99 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 ValueError.

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

math.sqrt(x)

Return the square root of x.

math.acos(x)

Return the arc cosine of x, in radians.

math.asin(x)

Return the arc sine of x, in radians.

math.atan(x)

Return the arc tangent of x, in radians.

math.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 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.

math.cos(x)

Return the cosine of x radians.

math.sin(x)

Return the sine of x radians.

math.tan(x)

Return the tangent of x radians.

math.pi

The mathematical constant π = 3.141592..., to available precision.

math.e

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