Operator precedence

In a Python expression the operations do not take place from left to right; the operators have precedence which determines the order they occur.

for example : $1 + 2 * 3 = 7$

not 9

That happens because * has higher precedence than +, so the first evaluation leads to

$1 + 6 \Rightarrow 7$.

Here is a simplified table of operator precedence in order from most important to least important.

1. ( ) — expressions in parentheses are evaluated first.
2. ** — exponentiation
3. +x, -x — positive or negative numbers (be careful this isn't confused with addition and subtraction though)
4. *, @, /, //, % — multiplications, divisions, modulus
5. +, - — addition & subtraction
6. in, not in, is, is not, <, <=, >, >=, !=, == — membership, comparison, identity operators
7. not x
8. and
9. or
10. = — assignment expression

Bit operators were not included here.

In other languages you have to check what the precedence rules are!

Use parentheses to ensure the order of operation you want. For example $2^{1/2} = \sqrt{2}$ if you write `2**1/2` though you get 1.0

`2**1/2` ⇒ 1.0 not $\sqrt{2}$

Use parentheses to group 1/2 so it will evaluate to 0.5, and then do the exponentiation.

`2**(1/2)` ⇒ $\sqrt{2}$

When in doubt about the order of operations, use parentheses to make sure it does what you mean.