**Mathematics**

Time to address one of the cornerstones of programming: mathematics. In order to carry out mathematical operations we should first take a look at what operators are available:

* + add
* - subtract
* / divide
* \* multiply
* % modulus

Let's write a short script to test each of these operators and see what they do.

**Maths.py**

print("24 + 8 =", 24 + 8)

print("88 - 17 =", 88 - 17)

print("6 \* 12 =", 6 \* 12)

print("50 / 2 =", 50 / 2)

print("25 % 4 =", 25 % 4)

Most of those should come naturally to you. However, you might not be familiar with modulus. What modulus does is return the remainder of a division operation. So with 25 % 4, 4 goes into 25 six times with 1 left over.

Save the above code and open a new file. Now we'll try something a bit harder.

**More\_maths.py**

my\_int = 18

result = (my\_int + 6) \* 3

print("Result:", result)

apples = 7

oranges = 12

bananas = 5

average = int((apples + oranges + bananas) / 3)

print("Average fruit quantity:", average)

operation = 3 + 2 + 1 - 5 + 4 % 2 - 1 / 4 + 6

print("Operation:", operation)

minute = 1

hour = minute \* 60

minutes\_in\_12\_weeks = ((hour \* 24) \* 7) \* 12

print("Minutes in a 12 week period:", minutes\_in\_12\_weeks)

One thing to point out before we continue is the use of 'int()' when calculating an average of 3 values. 'int()' forces the result of the operation to be an integer rather than a float, basically removing the decimal point and any numbers following it. It depends on the situation, but there are times when you may want to force your variable data types for consistency. Maintaining control over data types can also help prevent unwanted results.

As you can see we can create some rather long operations. Knowing the order of operations will help you understand and predict the results of these operations:

* Parenthesis
* Exponentials
* Multiplication
* Division
* Addition
* Subtraction

This is known as PEMDAS. Operations within parenthesis are done first, then exponentials. Multiplication and division share equal priority as do Addition and subtraction which are done last.

Try solving the following problem on paper first before typing it into the editor and seeing if you got the same result:

(5 + 3 - 1) \* (21 / 3) + 6

One thing we haven't covered yet is the exponential operator '\*\*'. This allows you to multiply a value to a certain power. We can see how this works with the following script.

**Exponentials.py**

one = 5 \*\* 2

two = (one / 2) \*\* 3

three = two - (one + 4) \*\* 1.5

print("One:", one)

print("Two:", two)

print("Three:", three)

Remember to save this as a separate script.

**A note on operators**

Remember to type operators with a space around them. It looks neater and is easier to read which is important when you move on to writing longer, more complex scripts.