

# Computer Applications - Python Exercise Sheet 1

All exercises adapted from MIT OpenCourseware under the [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 United States license](https://creativecommons.org/licenses/by-nc-sa/3.0/).

## Instructions

You can answer 1.2, 1.3 and 1.4 on paper: a printed version of this, or in your notebook, or in your logbook. If you prefer, you can do this digitally.

You **must** answer 1.1 in a text file named **homework1.py**.

*In both cases you must ensure you save it somewhere you can retrieve it, and bring it to every lab.*

## Exercise 1.1 - Equations

In this section you will write some code in a Python file. This is a text file; when you save it, name it **homework1.py**. You can edit these in IDLE.

**Part I:** Input the following sets of equations, and note the difference between *int* arithmetic and *float* arithmetic. You can do this just in your interpreter (you don't need to turn anything in for this part), but pay attention to the output!

1.  $\frac{5}{2}$ ,  $\frac{5}{2.0}$ , and  $\frac{5.0}{2}$  Note that as long as one argument is a float, all of your math will be floating point!
2.  $7 * (\frac{1}{2})$  and  $7 * (\frac{1}{2.0})$
3.  $5 ** 2$ ,  $5.0 ** 2$ , and  $5 ** 2.0$
4.  $\frac{1}{3.0}$  Note the final digit is rounded. Python does this for non-terminating decimal numbers, as computers 3.0 cannot store infinite numbers!

**Part II:** In homework1.py, transcribe the following equations into Python (without simplifying!), preserving order of operation with parenthesis as needed. *Save each as the value of a variable, and then print the variable.*

1.  $\frac{3 \times 5}{2 + 3}$
2.  $\sqrt{7 + 9} \times 2$  (Not been taught how to square root in Python? Search for the answer on Google!)
3.  $(4 - 7)^3$
4.  $\sqrt[4]{-19 + 100}$
5.  $6 \bmod 4$  - If you aren't familiar with modular arithmetic, it is pretty straightforward - the modulus operator, in the expression  $x \bmod y$ , gives the remainder when  $x$  is divided by  $y$ . Try a couple modular expressions until you get the hang of it.



4. var1
5. 1var
6. my-name
7. your\_name
8. COLOR

## Exercise 1.4 - Types

It is important that we know the type of the values stored in a variable so that we can use the correct operators (as we have already seen!). Python automatically infers the type from the value you assign to the variable. Write down the type of the values stored in each of the variables below. Pay special attention to punctuation: values are not always the type they seem!

1. a = False
2. b = 3.7
3. c = 'Alex'
4. d = 7
5. e = 'True'
6. f = 17
7. g = '17'
8. h = True
9. i = '3.14159'

To verify your answers, you can use the interactive Python shell, but first try to do the exercise without help.

```
>>> x = 100
>>> type(x)
<type 'int'>
>>>
```

## Sources

[http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-189-a-gentle-introduction-to-programming-using-python-january-iap-2011/assignments/MIT6\\_189IAP11\\_hw1.pdf](http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-189-a-gentle-introduction-to-programming-using-python-january-iap-2011/assignments/MIT6_189IAP11_hw1.pdf)

[http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-189-a-gentle-introduction-to-programming-using-python-january-iap-2011/assignments/MIT6\\_189IAP11\\_hw1\\_written.pdf](http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-189-a-gentle-introduction-to-programming-using-python-january-iap-2011/assignments/MIT6_189IAP11_hw1_written.pdf)

<http://mechanicalmooc.wordpress.com/>