SCIE1000, Computing Tutorial Week 2, Semester 1, 2011.

Aims

The aims of this computing tutorial are:

- To introduce the use of *variables* to store information in the computer as the program runs.
- To explain the use of *functions* and function *arguments* or *parameters* in Python.
- To introduce the use of the Python `input` command to allow your programs to accept information from the user while the programs are running.

Overview

[Ref: Appendix A of the Lecture notes, Section A.3, Pages 398–406.]

One of the reasons that computers have become a vital tool in all fields of science is their ability to store information and data. Variables are the key to understanding how data is stored when a computer program runs. It is also important to be able to get data into the computer program in the first place (for example, from a human typing things on a keyboard).

In SCIE1000 computing tutorials you will gradually build up your Python programming knowledge. Important new concepts will be introduced over the next five weeks, but will rely on you having a good understanding of the material covered in previous weeks. Ask a tutor for help if something from a previous week is unclear.

1. *[Ref: Appendix A of the Lecture notes, Section A.3.1]*

   (a) Read Section A.3.1 of the Python Notes carefully, then type the following program into Python:

```
# A weight/height program.
from __future__ import division
from pylab import *

weight = 38.5
height = 155
print weight," kg"
print height," cm"
```

   Save the program into a file from the IDLE Editor window, and run it.

   (b) Modify your program from Part (a) so that, in addition to the current output, it also prints the height in metres. Do this by adding a new variable `heightm` into the program.

2. *[Ref: Appendix A of the Lecture notes, Section A.3.2]*

   In programming it is vital to understand the use of functions. We will start with simple mathematical functions that you will have seen (e.g. on a calculator). These functions *take a single input argument* and *produce a single output (or return value).*

   Write a Python program to calculate each of the following and print out the results:

   (a) $\sqrt{38.5}$

   (b) $\sqrt{38.5 \times 155}$

   (c) $0.0167\sqrt{38.5 \times 155}$

(d) $38.5^{0.425}$ (recall that Python uses `a**b` to find a^b).

(e) $38.5^{0.425} \times 155^{0.725}$

(f) $0.007184 \times 38.5^{0.425} \times 155^{0.725}$

3. *[Ref: Appendix A of the Lecture notes, Sections A.3.1, A.3.2]*

   Variables can be used to break down and simplify complicated mathematical expressions.

   (a) Copy the answers to Parts (c) and (f) of Question 2 into a new Python program (open a new Editor window and use "copy-and-paste" with the mouse).

   (b) Did you use variables to store the output (result) of each of the calculations? If not, modify your program so that the results are stored in variables called `BSA1` and `BSA2` respectively.

   (c) The two values you have found are the estimated body surface areas (BSAs) for a person with height 155 cm and weight 38.5 kg, calculated using the Mosteller formula and the Dubois & Dubois formula. (We covered these formulae in your tutorial today. In fact, we have just estimated Nicole Richie's BSA.) Use your variables to calculate and print the *difference* between the estimates. Compare your results with the answers to Question 5(a) on your **tutorial** sheet this week. (If the answers differ significantly, then you have made an error somewhere.)

4. *[Refs: Appendix A of the Lecture notes, Section A.3.3]*

   Modify your code from Question 3 (make a copy so you still have the solution to Question 3) so that the program asks the user to enter a weight (in kg) and height (in cm), and then calculates the difference in BSA estimates for that weight and height. Test your program using a weight of 38.5 kg and height of 155 cm.

5. **Assessment Question (1.5%): BSA program.**

   Modify your program so that it:

   - prints a suitable introductory message
   - prompts the user to enter a weight in kg and a height in cm
   - calculates the estimated BSA using each formula
   - prints a message for each formula, identifying which formula was used and the corresponding BSA
   - prints a message showing the difference in estimated BSAs, with some text saying what is being printed.

   Your program must be commented, with sensible variable names. Test your program using a weight of 38.5 kg and a height of 155 cm.

   This program will contribute 1.5% towards your final assessment for SCIE1000. **To receive these marks, you must demonstrate your program and its output to a tutor during your tutorial.** Program output, including text messages and correct values, are together worth up to 1.0%. Appropriate comments and variable names are together worth 0.5%.

6. *[Refs: Appendix A of the Lecture notes, Sections A.3.4, A.3.5]*

   Consider the following Python program. This program is intended to perform the kind of calculation that you have done in Question 2(a) of this week's **tutorial** sheet. However the program contains 4 errors (Hint: 3 of them are small python "bugs" and the other is more an error of logic).

```
from __future__ import division
from pylab import *

#This program calculates the pressure of an (ideal) gas, given the
#quantity, the volume occupied, and the temperature in degrees Celsius
#---

#Print a welcome message
print "---Welcome to GasCalc v1.0---"

# First, get the radius value from the user
n = input("Please enter the amount of gas (in moles, mol): ")
V = input("Please enter the volume occupied (in cubic cm): ")
t = input("Please enter the temperature (in degrees C): ")

#We also have the constant, R (in L atm/mol K)
R = 0.0821

#First, we need to convert T to Kelvin
T = T + 273.15

#Now, do the calculation
P = (n*R*T/V

#Finally, print the results
print "The pressure of the gas is ", P, " in atmospheres, atm"
```

(a) Study the code and try to identify (and correct) the errors.

(b) The program is available from the course Blackboard site. Download it and open the file in the Editor. Correct the program and verify that it works correctly.