PH2150 Scientific Computing Skills

Andrew Casey

May 28, 2019

The aim of this first problem sheet is to introduce you to the environment in which we will be running *Python* during the course. The computers in the teaching lab have several version of *Python* installed and as we will see it can be used in differnt ways. However during the course the questions and notes will assume that you are going to be using the Anaconda installation, with a Python 3.x version. An environment file, PH2150 should exist which will ensure that all (or at least most) of the packages that you need will be installed and compatible.

1 Problem Sheet 1, Ex1:

Launch Anaconda, select the PH2150 or python 3.x (if PH2150 is not installed on your machine) environment. Within the environment tab identify which version of the packages matplotlib, numpy, scipy and pyqt aer installed on your machine and make a note.

Across campus Anaconda should be available but not all the packages that you need will be installed, this environment package manager is one of the ways in which you can install new packages and update existing ones. Nb. Over the summer there is normally a new computer build installed and Anaconda may be updated to a newer version with the latest python 3.7.

2 Problem Sheet 1, Ex2: Using Python as a calculator

Within Anaconda launch, Spyder, an IDE (Integrated Development Environment) *Scientific PYthon Development EnviRonment*,

First within the IPython console window:

1. calulate the result of 4/7 where 4 and 7 are integer numbers (If you have selected a python 2.x environment by mistake this will give a result of zero.)

2. calculate the square root of nineteen

3. calculate the factorial of 19

4. calulate the sine of 37 degrees

Now in the *Editor* Window write a program that prints (to screen) the results of the previous 4 calulations and save your code as a *.py file:

Is there a difference between running the code from the developer environment or typing directly into the interpreter? add a comment to your code. Nb. some Ipython consoles preload some import statements.

(Nb. Some of the differences between 2.x and 3.x are shown here, in 2.x to print the screen you use the statement: print 'the thing you want to print' However in 3.x, you use the function: print('the thing you want to print'), here the variable is in brackets (it is the argument of the function print).

*Note these functions can be found in the **math** library, to access import **math**, loads the names into your workspace or code, dir(math) will list all of the names of the functions in math, math.funcname() will let you use function help(math.funcname) will give you some information about how to use the function.*

3 Problem Sheet 1, Ex3:

A question to practice requesting inputs from users and returning the output after evaluation: The problem is as follows. A ball is dropped from a tower of height h, built on level ground. The ball has initial velocity zero and accelerates downwards under gravity. Write a program that asks the user to enter the height in meters of the tower and a time interval t in seconds (using the input function), then prints on the screen the height of the ball from the ground at time t after it is dropped, ignoring air resistance.

(In 3.x the function input() behaves like raw_input in 2.x)

4 Problem Sheet 1, Ex4:

A satellite is to be launched into a circular orbit around the Earth so that it orbits the planet once every T seconds.

a) Show (on paper) that the altitude h above the Earths surface that the satellite must have is $h = (GMT^2/4\pi^2)^{1/3} - R$, where $G = 6.67\text{x}10^{-11}$ $\text{m}^3 kg^{-1} s^{-2}$ is Newtons gravitational constant, $M = 5.97\text{x}10^{24}$ kg is the mass of the Earth, and $R = 6371$ km is its radius. Before writing a program it is important that you understand what your variables are and what you want the algorithm to do.

b) Write a program that asks the user to enter the desired value of T and then calculates and prints out the correct altitude in meters.

c) Use your program to calculate the altitudes of satellites that orbit the Earth once a day (so-called geosynchronous orbit), once every 90 minutes, and once every 45 minutes. What do you conclude from the last of these calculations?

5 Problem Sheet 1, Ex5:

Using the write() method described in the notes, create a file containing the first law of thermodynamic:

"In all cases in which work is produced by the agency of heat, a quantity of heat is consumed which is proportional to the work done; and conversely, by the expenditure of an equal quantity of work an equal quantity of heat is produced."

Write a program to read out the sixth word of the string and then print the whole statement.

Print (to screen) a list of the functions that can operate on the string, *hint type the string name in the interpreter window followed by a '.' and then press tab (auto complete) to see available functions*

6 Submit via Turnitin

When complete, the code plus comments should be submitted as a single file (PS1.py) via Turnitin Problem Sheet 1. The turnitin page will be available in the Autumn term after the Moodle changeover. For now save your work on your y: drive (+ usb memory) This problem sheet will be assessed in the first week of the Autumn term.