Python programming for scientists - group 5

This course is based on previous material by

Thomas Robitaille, Markus Demleitner, Robert Schmidt,
Yiannis Tsapras and Dylan Nelson

The web page for this course is
https://www.ita.uni-heidelberg.de/~gn014/teaching.html

My contact: gdespali@uni-heidelberg.de

Giulia Despali, 19-23.09.2022

To pass:

Submit the solutions to all three problem sets by **October 9th** (two weeks after the end of the course), by email:

- you must **send all of them**
- at least 60% must be correct
- format: python notebook or normal python code. Please include your surname in the name of the file:
  problem_sheet_1_firstname_surname.ipynb

- you can de-register from the course until Sept. 23rd, if you decide that you do not want to submit the solutions
- you can work in groups but each person then must submit separate solution files

To pass:

The following criteria are taken into account for grading problem sets - the notebook/program should:

- run with no errors *under python3*
- produce the correct results
- be clearly readable, and include text to explain what you are doing and why
- not include any unused code! Delete additional tests and commented code from the final version
- be well documented so that it is easy to understand what each part does.

There is never a unique solution to a problem, so it does not matter if your programs do not look the same as somebody else's! What matters most is that you get a chance to make mistakes and learn from them.

Schedule:

The block course lasts five days, 19-23 September 2021.
Each day will follow this schedule:

* 9:00-10:30 - lecture/exercises
* 10:30-11:00 - break
* 11:00-12:30 - lecture/exercises
* 12:30-14:00 - break for lunch
* 14:00-17:00 - exercises/finish homework

In the afternoon you can solve the practice problems that are provided for each day (these are not mandatory and you don't need to submit solutions, but they are useful to learn)
You can ask questions and we can solve issues together
You can also start working on the three final homework (although I cannot help you with that!)

The course will be in English.

Topics:

Day 1:

- Using the IPython Notebook
- Introduction to Python
- Defining Variables
- Control flow, introduction to functions

Day 2:

- More on Functions
- Modules
- Reading/writing data from files

Day3

- using standard Python libraries:
- Numpy: arrays, operations
- Matplotlib: plotting, data visualisation

Day4

- Introduction to Scipy
- interpolation, integration, fitting scientific data

Day5

- Object-oriented programming, speeding up codes
- using online resources (depending on how much time we have left)

How to set up:

(1) CIP Pool and Jupyter servers

If you use the CIP Pool terminals for the course, you will first need to log in to your account. Now, to run the Python codes and read the Notebooks, you should connect to the Jupyter server of the Kirchhoff Institute for Physics (KIP). You can do so by clicking the following URL:

https://jupyter.kip.uni-heidelberg.de/

You log in with your student ID and password. If all went well, you see a kind of file browser. It shows your home directory _on the Jupyter server_ (which is not the same as your home directory on the physics servers).

Download the notebooks for the day on the computer and upload them to the Jupyter server

(2) your own laptop:

unless you are already familiar with Python
and with the terminal, the easiest solution is to download the Anaconda package for your operative system:

https://www.anaconda.com/products/distribution

Then open a terminal and type “Jupyter notebook” or open Jupyter from the interactive Anaconda prompt

What is good programming?

Being good at programming is not simply knowing a lot about a programming language, or how to write very fast programs. It is about:

* understanding a problem conceptually and being able to translate it into code
* thinking of new ways to tacklea problem
* knowing what tools to use
* knowing how to fix your program when it does not work
* writing a program that is fast enough, **not** the fastest possible
* writing a program that can be understood by other people

(or by yourself in a year!)

This is important in science in order to have reproducible results

Why learn Python?

Python is becoming the standard and most used programming language both in science and industry:

- it is easy to learn
- a huge variety of libraries and codes already written in Python exist online and are ready to be used
- it can be used for data analysis, data visualisation, machine learning, artificial intelligence, web developement and so on

Python is an **interpreted language**, which means that the code is not compiled in advance, which makes
it slower than languages like C/C++ or Fortran. However it as a simple syntax, is interactive and can
be interfaced with C/Fortran Codes