Math 246 Unit 2, part B: Creating Python programs

Professor Brenton LeMesurier, revised October 3, 2015

Now we start creating Python programs, which at first simply means putting commands into files instead of typing them in interactively.

To start, create a new file in Spyder:

- with main menu item **File**, item **New file ...**, or
- use the "blank page" icon at the left above the file editing pane.

We will work through several versions of this, starting with one that implements the most basic mathematical formulas, and workng towards one that is flexible, robust, and well-documented.

Exercise 2.0 Solving quadratic equations, Version 0

Create a Python file named `quadratic_solver_0.py` that computes the roots of the quadratic equation

$$2x^2 - 10x + 8 = 0$$

and displays them to the screen.

Aside: Naming rules for variables, files, and folders in Python

Note well the name I specified above: the names allowable in Python are more restricted than the names that can sometimes be used for computer files and folders in other contexts, so here are the rules!

1. Python variable names should start with a letter and use only letters, digits and the underscore "_": **no spaces or funky punctuation**.
2. *When tempted to use a space for readability, use an underscore instead.*
3. Python file names should usually follow the same rule as above, except that they will end with a suitable suffix: ".py" for Python code files and ".ipynb" for notebooks like this one.
4. It is not an official rule, but for your sanity, I suggest that you *avoid UPPER CASE letters*!

Exercise 2.1 Refinements

There is a lot of room for improvement on the above "version zero"; for the following, leave the above file as it is, and create one or more new versions under new names like `quadratic_solver_1.py` and so on. Then you can submit multiple versions, to show your progress and get feedback on each stage if needed. (I am also a document pack-rat, and like to be able to revert to previous versions if I have gone astray.)

1. **Avoid duplication**, by making a new version capable of solving any quadratic, not just one. Do this interactively, with input of the coefficients a, b, and c using function input().
2. Make it more **user friendly**, by making a version that displays messages explaining what is going on, what the input should be, and what the output means. (From now on, never just print out a bunch of numbers!)
3. Make the file itself more understandable, for the sake of another person reading it, or even you re-reading it some weeks later:

   (a) add introductory comments at the top of the file giving an overall description;

   (b) add comments at particular points that are not self-explanatory, but ...

   (c) make the code inherently self-explanatory whenever possible, by for example choosing variables names that are descriptive.
4. Make it more **robust**: not all choices of the coefficients will give two distinct real roots, so work out all the possibilities, and try to handle them all. *This is where some pencil-and-paper preparation is important: as we go on, I will emphasize proper mathematical preparation before typing a single line of code!*
5. Finally, from now on all documents that you submit, such as Python files, should have a *title*, your *name* and the *date* at the top, as you would with any paper document that your submit.

Submit your work to the MATH 246 Dropbox for Unit 2, part B in OAKS

Submit the various Python files created. In general, such files will be all you need to submit, since I can run them to see that they work correctly.