TMTplus Introduction to Scientific Programming

Mahdi Farnaghi, Mahdi Khodadadzadeh & Robert Ohuru

March 2021

Note from the teaching staff

The materials that we use for this course are somewhat new and have been heavily reworked. We will likely have made mistakes in this work or have glossed over issues that deserved better or different treatment. Where you find issues worth noting please do report these to us as it will allow to repair and improve.

By the way, welcome to the Exercise book for Introduction to Scientific Programming! We believe this book is self-explanatory, so go ahead and practice.

Chapter 6

Dictionaries & their expressions

Dictionaries provide a direct look-up mechanism that allows to associate a bunch of information with a single characteristic key value, which is often a string. In the old days, a dictionary was called "associative memory," because of this direct access mechanism. Dictionaries are commonplace in languages such as Python, and come in many disguises. Not now but later we will see that JSON objects are also dictionaries, for instance.

The exercises

The main goal of each exercise is to use the Python programming language to review concepts you have learned during the lectures.

In this exercise, keep in mind what is the power of dictionaries: what can you encode with them, and how they can be used for various bookkeeping tasks while a script is running, and so forth. In short: acknowledge the power of these constructs for coding later.

6.1 The use of dictionaries

Ex 6.8 We want to create a dictionary that you will use as your agenda. This dictionary will have as a key the name of your friends or relatives and as a value a string that holds their phone number. Add four friends to your agenda. Print the contents of the entire dictionary. We have indicated that a phone number is a string. Could or should we have used integer as the data type, or are strings better?

Ex 6.9

Explain whether or not the key of a dictionary can be a mutable element (e.g., for example a list). What are the types that a dictionary can handle as key value type?

Ex 6.10 You should have flagged three possible types for keys above. However, the value associated with a key can be anything. Store together with the phone number of your friends also their e-mail address. For this purpose, you have to re-assign to each entry in your agenda a small list or tuple of two elements with this combined content. Which of the two is the better choice: list or tuple?

Ex 6.11 Iterate over your friends in the agenda to print the information stored. Print the contacts as follows: first the name, then an arrow symbol, then the telephone number and the e-mail address between parenthesis. Notice that you need to access the second element of a tuple/list.

Ex 6.12 Modify one contact in your agenda by changing the phone number. Now modify another contact in your agenda by adding a second phone number after the e-mail address. This means that for this friend, you need to re-assign a new tuple or list, this time containing three elements.

Ex 6.13 Imagine that, after your script has read in a raster image you get the following dense matrix:

```
dense_matrix = [ [0,0,0,1,0,0,0,4,0,4], [0,0,0,0,0,0,0,4,4,0],
                 [0,2,0,0,0,0,4,0,3,0], [0,0,0,0,0,0,0,0,0,0],
                 [0,0,0,3,0,5,8,2,0,5] ]
```

Because we don't need the zero values, presumably because they represent NODATA, use the techniques discussed in the slides to create a new sparse matrix that only contains the non-zero values. Ensure you save as a key in the matrix the position that the non-zero values have in the `dense_matrix`. You need to use a nested for or while loop. Make up your mind before you code ... that is, give thought before coding.

Ex 6.14

Dictionaries are often used in tasks that require some form of inventory, i.e., some *indexed listing of cases.* Especially situations where upfront you do not know for which or for how many cases your inventory will hold information, a dictionary is a good data structure. This is because at any stage you can add an extra key with its associated value and also you can update the value associated with an existing key.

This last exercise is more a *conceptual thought experiment*, and is certainly not yet a coding exercise. (But those of you who feel adventurous or challenged are invited to try out the below and create a script that actually does it.) The problem at hand could play out in a scientific setting in natural language studies.

Suppose we have a novel by a well-known author as a digital text file. And the question we pose ourselves aims to understand a few things about the author's writing style. Does she write long or short sentences, for instance? To answer that question, we can script some analytics. For instance, read in the text file, split the text into separate sentences, determine the length of every sentence, and do our administration, keeping tally. This administration can be done using a simple dictionary. After we have completed going through the text, we might want to produce a histogram that shows how many sentences (y-axis) the novel has at which sentence length (x-axis). We will learn about charts soon, so we will not actually produce a chart yet. But let's address the other bits. Here are our questions. We want you to provide an answer in words only.

1. What sort of dictionary would be useful to do the script's bookkeeping that we alluded to above?
2. What would be a good variable name for that dictionary?
3. How should this dictionary be initialized at start of the script?
4. Can you think up how a full text (the complete novel) could be split up to separate sentences? Which would be your Python tools? What are the concerns for the splitting process?
5. What should happen if you encounter a sentence with a length you have not seen before? What should happen to the dictionary, specifically, to keep book properly?
6. What should happen if you encounter a sentence with a length that has occurred before? What should happen to the dictionary, specifically?
7. Can you sketch roughly the structure of the required script? I..e, provide a script in high level pseudo-code.