Isepp Rebane
4/30/2023
Foundations of Programming: Python (IT FDN 110 A Sp 23)
Assignment 05

Lists and Directories

Introduction

Programming is all about data. Strings, integers, floats, they are all data types. What if you need to work with multiples of these types of data? You would interact with a collection of data, which was covered last week. Dictionaries are similar to lists, but have differences. This week covers dictionaries, source control, separation of concerns, and script templates.

Dictionaries

Dictionaries are very similar to lists, and look very similar to lists as well. However, instead of using a numeric (index) value to specify a section in a list, dictionaries utilize a key (character) value instead. For more advanced collections of data, say an excel spreadsheet or a database, dictionaries are ideal because you can sort data into rows, columns, and tables. Example of a dictionary bellow:

```
dicvar = {}
dicvar = {“name”:”isepp”, “dob”:”8/31”, “state”:”wa”}
```

Working with dictionaries

In the following example, you will see how you can view a “row” in a table, from a saved file. The `split()` function splits the values apart, while the `strip()` function removes the carriage return.

```
userFile = 'users.txt'
dataFile = None
dicRow = {}
lstTable = []

dataFile = open(userFile, "r")
for row in dataFile:
      lstRow = row.split(",")
      dicRow = {"name":lsftRow[0],"dob":lstRow[1],"state":lstRow[2].strip()}
      lstTable.append(dicRow)
datafile.close()
print (lstTable)
```

“Separation of Concerns”

This is a term that really resonated with me when I watched this week’s lecture, and did more reading up on what this actually means. Separation of concerns is a design principle for separating a computer program into distinct sections. Most of the programs created this far have been short. Depending on the program, many lines of code can be utilized, so it’s important to keep data organized. Not only for the programmer to read the code, but for others as well. Whether something is being changed, deleted, scaled up, or down, it’s vital in becoming effective at coding, whether as a hobby, or professionally. The term *separation of concerns* was most likely coined by Dutch computer scientist Edsger W. Dijkstra in his 1974 paper "On the role of scientific thought".

Let me try to explain to you, what to my taste is characteristic for all intelligent thinking. It is, that one is willing to study in depth an aspect of one's subject matter in isolation for the sake of its own consistency, all the time knowing that one is occupying oneself only with one of the aspects. We know that a program must be correct and we can study it from that viewpoint only; we also know that it should be efficient and we can study its efficiency on another day, so to speak. In another mood we may ask ourselves whether, and if so: why, the program is desirable. But nothing is gained—on the contrary!—by tackling these various aspects simultaneously. It is what I sometimes have called "the separation of concerns", which, even if not perfectly possible, is yet the only available technique for effective ordering of one's thoughts, that I know of. This is what I mean by "focusing one's attention upon some aspect": it does not mean ignoring the other aspects, it is just doing justice to the fact that from this aspect's point of view, the other is irrelevant. It is being one- and multiple-track minded simultaneously.

Templates

When a script needs to be created multiple times, but certain parts need to be changed, you want to have a sense of consistency. Templates are a great way to create programs that are different, but similar in style. Very basic example bellow:

```
#Name
#Filename
#Change log, date, time

var1 = “”
var2 = ()
var3 = []

#ADD NEW CODE HERE
#BODY#
end()
```

GitHub

You have a program that changes the name of every WiFi hotspot at the company you work for, great! What if you need to reverse the changes you made to the program? That is where source control comes in. Git is a system developed by Linux creator, Linus Torvald used for source control. GitHub is an online service where code is stored It’s used for version control, backups, and peer review. It is the largest source code host in the world.

This Week's Assignment

As per the assignment text.

Modify a new script that manages a "ToDo list." This project is like the last one, but different enough to be a challenge. The "ToDo" file will contain two columns of data, "Task" and "Priority." Load the columns into a Python Dictionary object. Each dictionary object represents one row of data, and these rows must be added to a Python List object to create a table of data. I have provided a starting template to I want you to modify and use for your program. You will note that it is both easier and harder to work with someone else's template and code, and that is part of the assignment.

Code is also to be posted into GitHub to share with peers. My program running in PyCharm(Figure 1), and in terminal (Figure2).

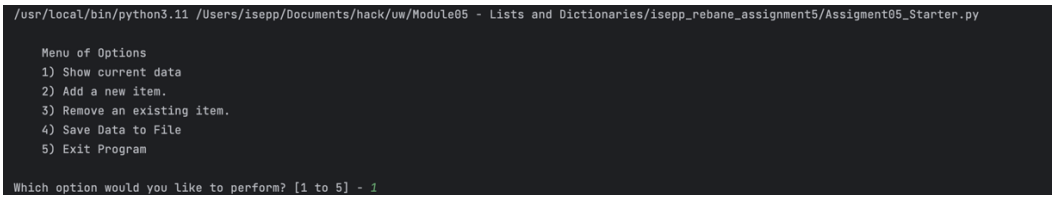

Figure 1

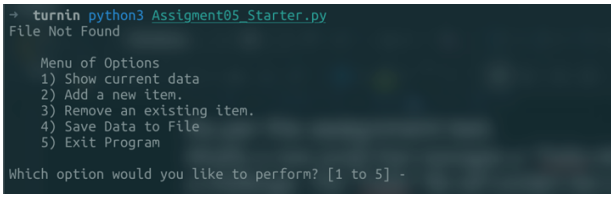

Figure 2

As my code is getting longer, I'll leave screenshots out. The name of the file is `Assigment05_Starter.py`

I made this script in one while loop, with multiple elif, and for loops depending on the choice that was made during run. Separation of concerns is a notable in this file.

Line 1-10:
Script header information.

Line 12-20:
Variables used listed.

Line 26-34:
I just a try and except for error handling. When the program starts, it looks to see if the ToDoList.txt file exists and prints out the table. If not found, the except statement prints out that the file does not exist.

Line 36-48:
User menu and data input.

Line 50-54:
Print current data.

Line 57-65:
During this assignment, I kept my program simple, maybe too simple. Adding more conditions or features typically broke something when I coded, so I left them out. Program asks the user for a task and priority, saves the variables, places them in a dictionary, appends it to the table, and prints the entry for a simple confirmation of what was entered.

Line 67-74:
I used a == operator to match the key in the dictionary if a row was to be deleted.

Line 76-83:
In a loop, file is opened, and the data is appended. A “Data Saved!” message is printed to confirm the write was made.

Line 85-85:
While loop breaks if option 5 is made.

This write up, as well as my script is saved on my GitHub page, located at the following URL. https://github.com/Tricepp

Summery

- Covered what a dictionary is and how they differ from a list.
- A quick example working with dictionaries.
- Separation of Concerns, it’s origin, and why it’s important.
- Templates.
- Git and GitHub.
- An overview of this week’s assignment, and a breakdown of my code.