Alexandra Ralevski

Nov. 12, 2020

Foundations of Python

Assignment 05

https://github.com/ARalevski/IntroToProg-Python

Creating Scripts Using Lists and Dictionaries in Python

# Introduction

In Module05 I will be learning about how to use lists and dictionaries in Python and write a script using both. I will also learn about the ‘Separation of Concerns’ and the value of using functions when writing code.

# Lists

Lists are a collection of values that you manage under one variable name, for example user\_list. You can access the individual values of the list by indexing. Data from list files are always stored in memory. However, if you close the program or script before saving, the data disappears. To save the data, it can be saved to a text file. If you do not specify the location of the file, it will save right next to where the script file is saved. To use the data, you load it from the hard drive back up into memory, you never use the data directly from the file. You read data from a file to a list by using the ‘read’ function in Python (using the letter ‘r’) when opening the file. For example: objFile = open(‘FileName’, ‘r’). You the call the data from the rows you want and print it with the separators you want.

# Dictionaries

A dictionary is similar to a list and tuple in that it contains a sequence of elements. It also has additional methods specific to it. An additional feature of a dictionary is that the subscript is character data (referred to as a key) instead of numeric data (an index in a list). You use the braces {} operator to indicate you are using a dictionary. To read data from a file into a dictionary, you can use the ‘key ‘to refer to the specific rows you want.

# Separations of Concerns

Many scripts involve several hundred if not thousands of lines of code. It is therefore important to organize your code in a way that it can easily be disassembled and re-built in other locations or for other projects/systems. Each section of the program should address a separate ‘concern’, i.e. a portion of information that affects the code. This is called the ‘Separation of Concerns (SoC)’. Most programs can be divided into three sections known as ‘Data’, ‘Processing’, and ‘Presentation’ (or Input-Output).

# Functions

A function is a named set of one or more statements. Because functions must have a name, you can use that name to call a particular function throughout your code. Because they group statements together, functions help you to organize your code, especially in the ‘Processing’ and ‘Presentation’ sections.

# Script Template

A script template is a template of pre-written sections (usually a header and SoC) that help keep your code more consistent and make it look more professional. Most IDEs, including PyCharm, allow you to save custom templates and use them with each new file.

# Error Handling (Try-Except)

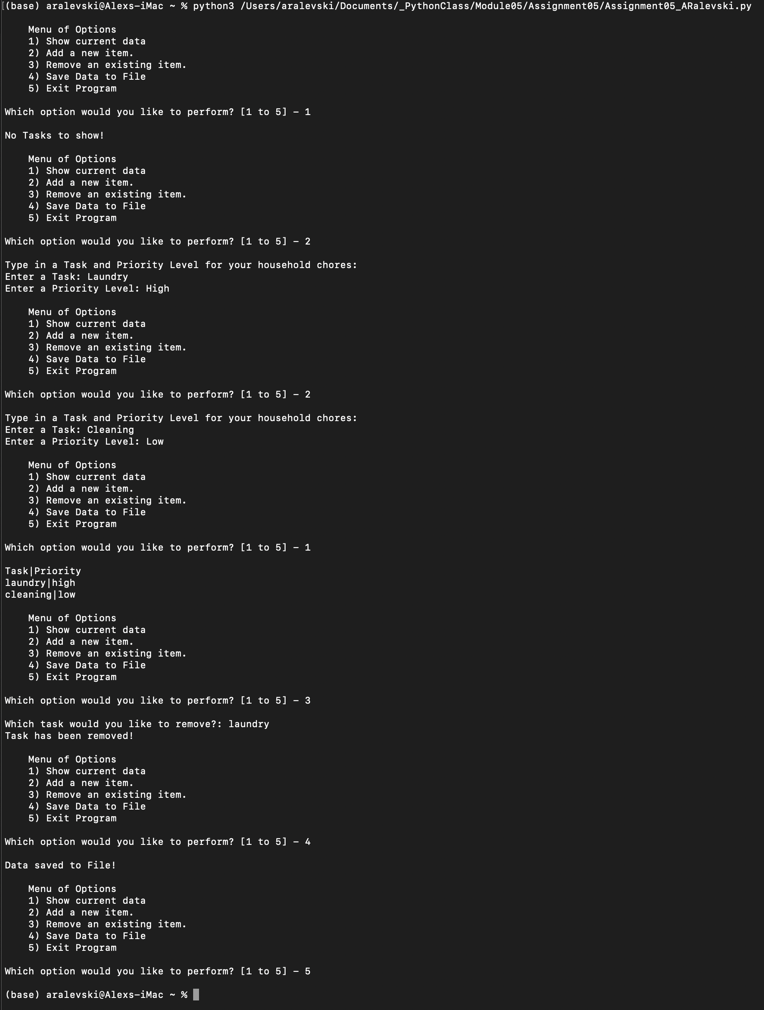
When a user is using your code, they can oftentimes make mistakes. Rather than displaying the standard error messages you can display more user-friendly messages using a Try-Except block. This also allows for a simple and organized way to group statements together to be processed. Because custom error messages can vary between developers, the idea of ‘Structured Error Handling’ was developed to create similar structures of code so the output to the user would look very similar regardless of who is writing the code. This is done by using a ‘Try-Except’ or ‘Try-Catch’ Block, which takes some data (the ‘Try’ section), and if it works, great, but if the user makes a mistake or there is an error the code will jump to the ‘Except’ section with a custom error message.

# GitHub

GitHub is an online repository for storing, sharing, and working with code with other people. Traditionally this was done as a network share on an organization’s server, but sometimes it more convenient to have this data on the internet, as long as it can be secured. GitHub uses the source control software Git.

# Writing Code using Dictionaries and Lists

I wrote a script to create a ‘ToDo list’ of household tasks and their priority level using rows of dictionaries loaded into a table. The output of the code running in the Terminal is shown in Fig. 1.

****

***Figure 1: Module05 Script running the Terminal***

# Summary

In summary, I have written a script that creates a ‘ToDo list’ of household tasks and their priority level using lists and dictionaries. I learned how to use GitHub and upload files, as well as the best ways to create a script template and write better error handling messages.