Kristin Temperly

02/22/2022

Intro to Programming (Python)

Assignment06

GitHub URL: <https://github.com/xkkt0700/IntroToProg-Python-Mod06>

Functions, Parmeters, and Arguments

**Introduction**

**In this module, we learned the concept of a Python function. Parameters were defined in addition to the passing of agruments into a parameter. Randal noted that many in the industry use the terminology interchangably, which is not entirely accurate.**

**Chapter 6 in our textbook also taught us the about global and local variables and the difference between the two. Michael Dawson used the concept of a house with tinted windows as an analogy to understand the behavior of global vs. local variables. I will not forget this analogy.**

**Functions**

**Functions are a way to simplify our code by placing it in a re-usable block. Using functions allows the programmer to execute the exact same block when called, which in turn help with the overall organization, or Separation of Concepts. To execute a function, it must be explicitly declared and explicitly called in our code. More specifically, functions are defined using the “def” command before the function name. Functions can range from very simple to very complex. The assignment in our To-Do list this week receives input to an argument to a function. Functions are often documented using what’s called a docstring, as exemplified by Randal’s code, which begins and ends with three double quotes.**

**Parameters and Arguments**

**Parameters are variable names inside of the parenthesis of a function header. Parameters catch values sent to the function when the function is called using agruments. Arguments get passed into the parameter. The parameter uses the argument. Parameters are typically lower case and without a prefix.**

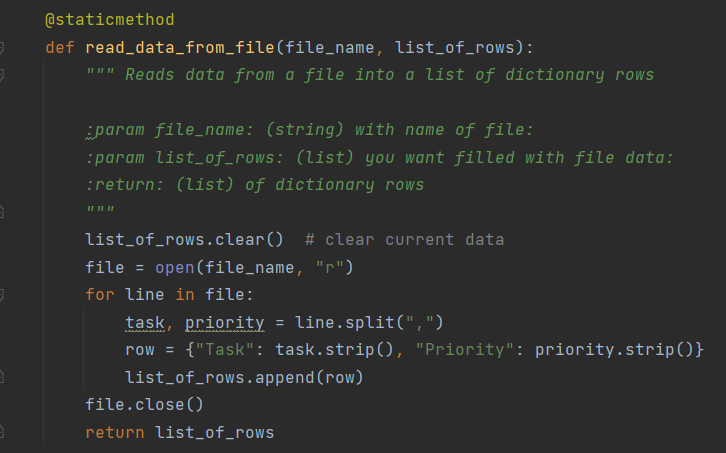
**Classes**

**Using classes is a way of organizing or grouping functions. In the starter file that Randal provided, he uses classes as a way to organize his code. One section manages the processing, and the other manages input/output or IO, so intuitively Randal used those names to identify the classes.**

**Putting It All Togheter**

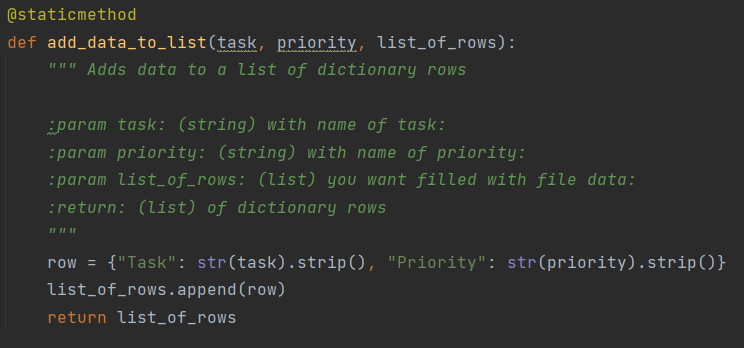
**The assignment this week was similar to the assigment last week. In fact, the two assignments provide the exact same user experience. However; from the perspective of the programmer, we organized the code better, made it more efficient, and easier for other programmers to read, understand and troubleshoot. Using classes and functions also reduces repetiveness. Instead of using the for while loop exclusively like we did last week, we instead used classes and funtions to better section the code according to best practice methods.**

**The script begins with the “Processing” class, which defines four functions. The first function called Proessor.read\_data\_from\_file passes two arguments, the file name and the list table. See listing 1. Note that the program will error out if the file does not already exist.**

****

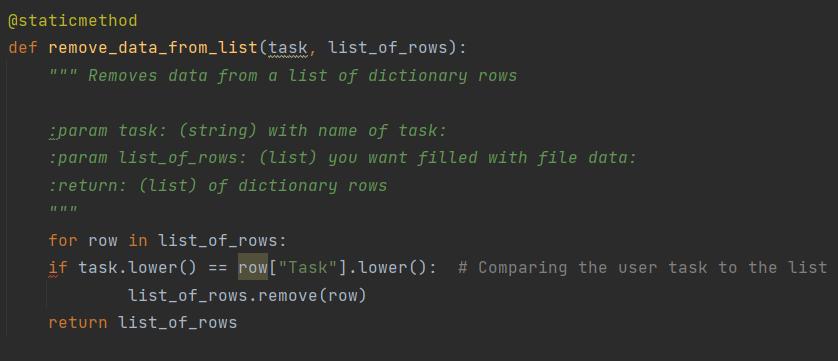
**Listing 1. read\_data\_from\_file**

**The next function is called add\_data\_to\_list and it defined by the code in listing 2. Its purpose is explained by its title and it simply appends new rows of data by passing three parameters, illustrated in green below. Adding a task is later called by passing the task and priority to the function as arguments, which then creates a new dictionary row, and then appends to the list table.**



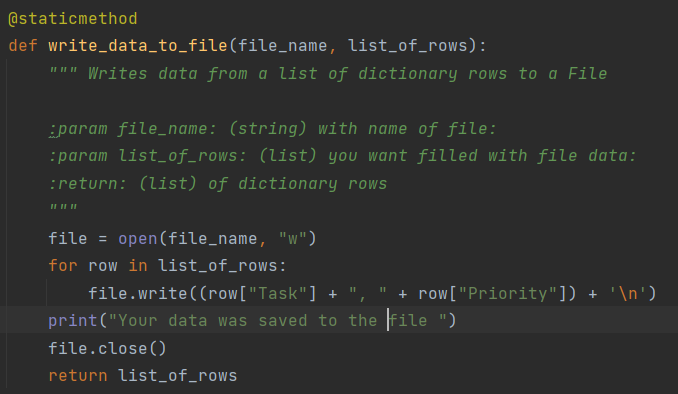
**Listing 2. add\_data\_to\_list**

**The next function is called remove\_data\_from list and it too receives a definition. This function uses two parameters, in green below. See listing 3. Its purpose is explained by its title, and it resembles exactly what we completed in the assignment last week, except this week we’re doing it with a function. When the function is later called, the task and the task list are passed as arguments. A loop is used to review the list looking for a match. If a match is found, the row is deleted.**



**Listing 3. remove\_data\_from\_list**

**Next, the write\_data\_to\_file function is defined. It too is described by its title and also is identical in purpose to our asisgnment last week. Except this week we did it using a function. There seems to be a pattern here!**



**Listing 4. write\_data\_to\_file**

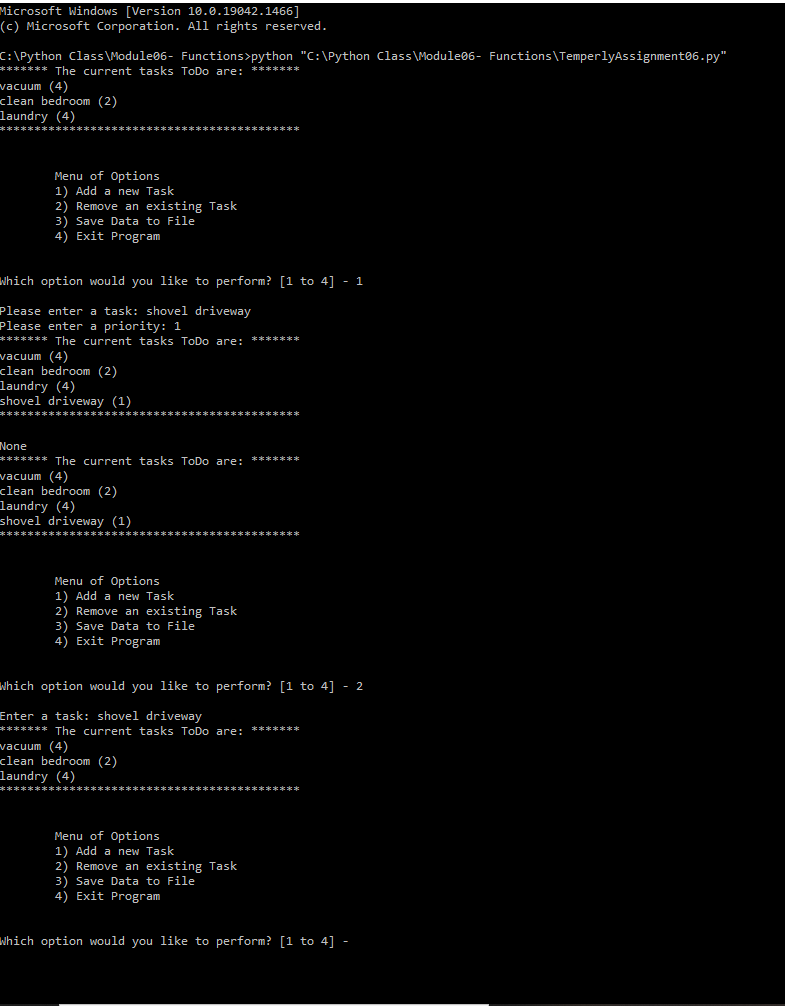
**These four functions are later explicity called by passing arguments through a parameter, as described earlier in this write up. When the user has completed adding and deleting tasks to the list, they can save the list to the file. At that time, a connection to the file is initiated and each row is written to the file. The connection is then closed, something Randal recommends but not required by Python. Finally, the final option of the menu breaks the loop and ends the program.**

**Summary**

**In module 06, we learned about functions, parameters and arguments. Functions help to keep the programmers code organized and readable for other programmers. Functions with like purposes can and should be grouped into classes, as illustrated by the two different classes in our starter code this week. Parameters are variables inside the parenthesis of the function, and arguments are passed when the function is called.**

**The more I read Python code, the better I understand it! I feel like I’m really starting to learn Python. One thing that still confuses me is the concept of encapsulation.**

**Listing 5 is an image of the assignment this week running in command prompt.**



**Listing 5. Execution of assignment06 using the command prompt.**