Kip Keeney

Randal Root

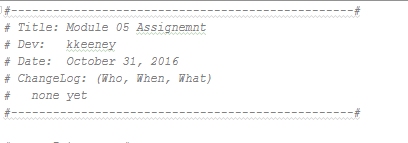
IT FDN 100 A (Python)

31 October 2016

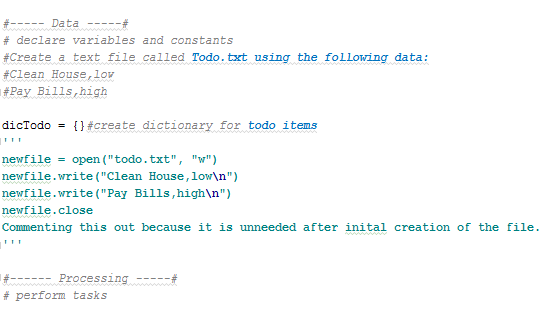
Module 05

In module 05, we covered List, Dictionaries, Try-Excepts, Functions, Script Templates and Github. As concepts everything made a lot of sense to me, but trying to put these items into practice was a bit more difficult. This is one of the first assignments where I actually don’t think that the labs helped with the last assignment, so in this write up I will skip going over those tasks.

First thing I did was add a script template as well as information about the author of the file:

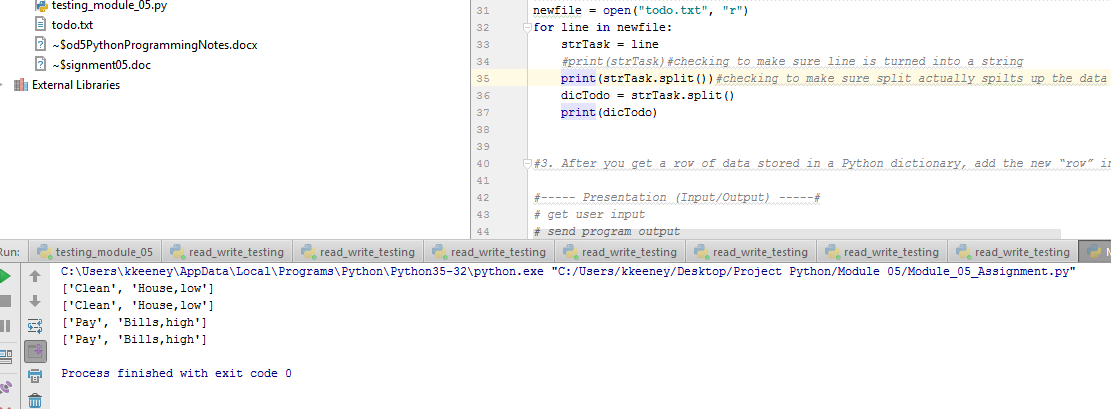


I thought that we actually had to create the text file using the code, so I added some code to my assignment that created the file with the initial data. I later commented that code out once the file was created. Here is why my code to create the file looked like:



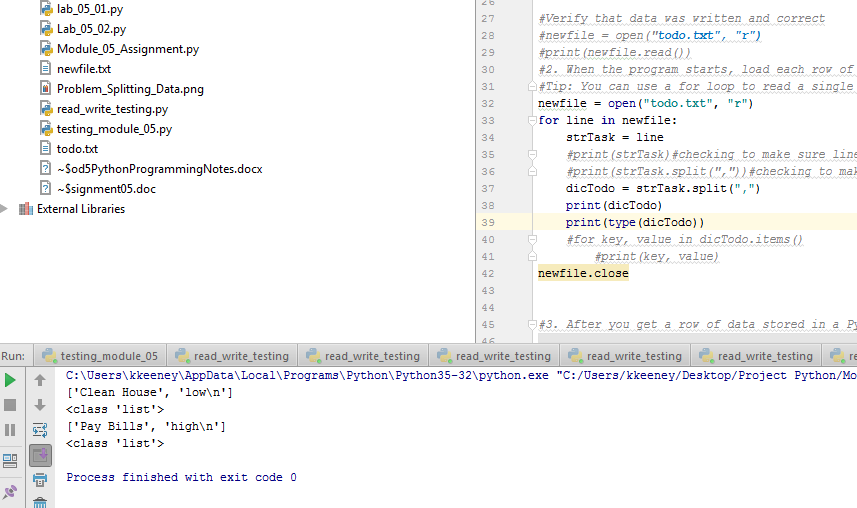
I created this based upon the information gleaned from the youtube video that was not part of the regular lesson.

Once the file was created, I needed to read the data out of it and store that data in a dictionary. This was by far the toughest part for me and where I spent the majority of my time. I ended up needing to do a lot of testing because I found that I often had the incorrect type or the data didn’t get added in the manner I thought. For example, when I tried to add the data initially, it split on the space and not on the comma. Here’s how the problem presented itself:

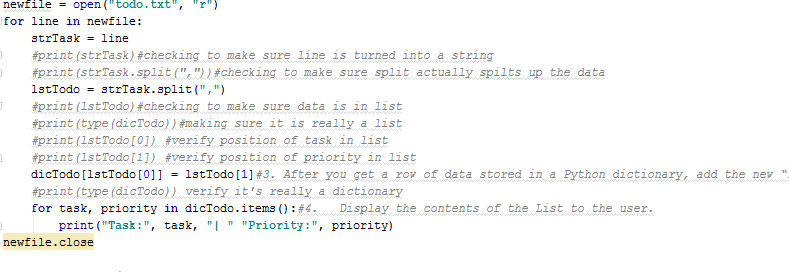


“Pay” and “Bills,high” were where the file wanted to split the data naturally. This was easily solved after I discovered the issue, by adding an argument strTask.split(“,”). Then the data was split by the comma and not by the space.

Initially I misunderstood how the Dictionaries functioned and I thought I was creating one, but it wasn’t functioning at all like a dictionary. I ended up adding more comments to identify the types of all my lists, dictionaries, etc. This image displays what I was running into initially:

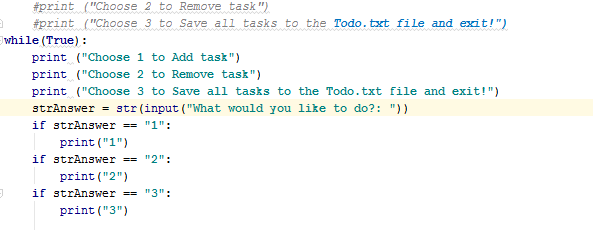


Even though I called out dicTodo as a dictionary early in the code, it somehow was turned into a list when I tried to add data to it. In the end, this section of the code ended up looking like this:

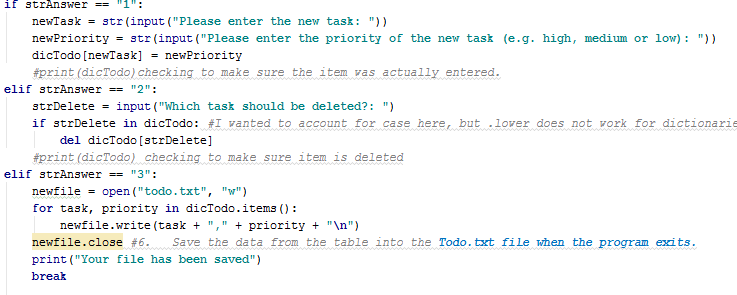


This proved very useful to me, mostly to verify that what I was doing was correct. It looks super messy and it should probably be cleaned up, but I think I’ll keep it for my own personal reference.

After I got the data into a dictionary successfully, I needed to ask the user what to do next. I found that this was a bit easier. I knew I would need a while loop and I wrote up what I thought the code might look like here:



With this basic outline, I could treat each option as an individual problem to solve and it made this much easier. I ended up changing the last two ifs, to elifs, but what was curious to me is that it still seemed to function just fine with the ifs in place. Here’s the final result of the last section:



I found option 1 and option 3 to be the least difficult. We have essentially done option 3 in previous assignments and option 1 was done because of work done earlier in the code. The 2 option took me a little bit and I’m not super happy with it, but it works. I really wanted to account for lower/upper case issues, but I ran out of time.

All in all I found this to be the most difficult assignment to date. I will definitely need to spend some more time with dictionaries as I find converting data and putting it into a dictionary a bit challenging.