Michael Clark

March 9, 2021

IT FDN 110 B

Assignment 08

[https://github.com/MClark89/IntroToProg-Python-Mod08](https://github.com/MClark89/IntroToProg-Python-Mod07)

Objects and Classes

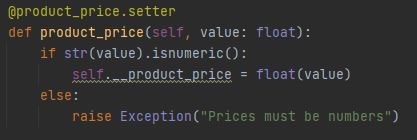
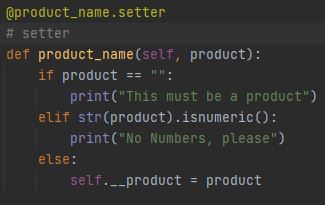
Introduction

This week we developed an understanding for classes and objects. Classes are in a simple way a blueprint for the code, while objects are statements used to define the class. In this weeks assignment we were tasked with creating something similar to previous homework’s but this time we incorporated creating classes in order to practice.

The Product

## Initializing the Class

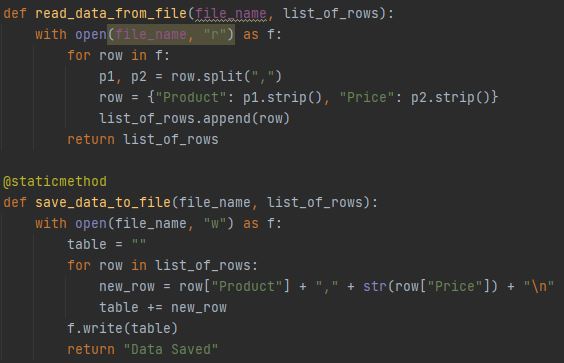
The first step of this long journey was starting the program by describing the products. In Figure 1, it shows that we defined the function by naming it ‘product’ and then we made the initial variables ‘product’ and ‘price’. The next steps were to create code that first defined how to store the data and then create code that incorporates error handling. Specifically, for the product, we wanted to make sure that the end user did not enter a number in the product section of the input (Figure 1). We also needed to make sure that the end user entered an integer into the price section (Figure 1). Both these error handling scripts were chosen to be used because these errors were thought to be the most common that might occur with the end user.



***Figure 1. Error Handling for both Product and Price***

## Processing

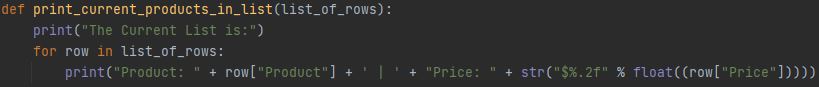
The processing section was very straight forward. In previous homework’s we have created this script before, so I just copied the same basic structure and adapted it to this assignment by changing the row descriptions to “Product” and “Price” respectively (Figure 2). One major difference that was included in this assignment that was not included in previous ones was the use of an f string. In the previous homework I incorporated formatting a string, but upon further research I found that using f strings made the code more readable and were just in general more widely used than format when dealing with strings.



***Figure 2. Reading and Writing of Data to a Table***

## Presentation

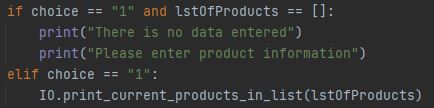
Similar to the Processing section of this project the Presentation was relatively straight forward due to this already being somewhat created in previous assignments. One notable change was when defining the print\_current\_products\_in\_list, in order to make the price showed up as a specific integer, I use ‘%.2f’ (Figure 3). This function creates a placeholder that turns the given value into an integer with two decimal places, similar to how the U.S. currency is written, i.e. 9.00 instead of 9.0. This makes the data appear in the .txt folder more readable to the human.



***Figure 3. Code that shows the current user input***

## Main Body of Script

For the final step, we were tasked with completing different steps in order to make the program run smoothly. This section of code was also somewhat like previous homework’s, but with a twist. This section consisted largely of code that reacted to user input. In Figure 4 it shows that depending on which number from the menu was entered, it would trigger certain IO modules that were created earlier in the script. These would then run and if they ran into errors due to end user input problems, the if statement would be activated.



***Figure 4. Script Showing what happens when end user selects option number one***

## Testing Final Product

When testing the final product, Figure 5 Shows that the main menu appears when the application is launched. Then based off the selection of the menu, when the second option is called upon, the data is collected from the user and loaded onto a .txt file. When selecting option one, the program relays the data on the .txt file back to the end user. While the third choice in the main menu saves the data and exits the program.

## 

***Figure 5. Example of end user code experience***

Summary

As we learned more about python this week, we added more tools to our tool belt. Classes are defined as groups of data that behave in similar ways. These classes can be used to easily group information in a concise manner that is easy for the computer to organize when using multiple variables. We also combined multiple tools that were created in previous homework’s that helped build a strong program that was relatively resistant to user input errors.