Nikita Daharia

June 8, 2022

Foundations of Programming: Python

Assignment 08

https://github.com/nikitadaharia/IntroToProg-Python-Mod08/upload

Introduction:

This paper aims to demonstrate the python programming skills I have learned in this course. Using classes in python, I will create a program that allows a user to store products and their price into a text file using the pseudocode provided. The latest version of PyCharm Community version is used for this assignment on a Mac OS.

Writing a Python script allows a user to store products and their price into a text file:

I started by creating a new sub-folder called Assignment08 inside of the _PythonClass folder (created in Module 01) in the Documents folder on a Mac OS. Then I created a new project in PyCharm that uses the _PythonClass\Assignment08 folder as its location.

Then global variables used throughout the code were defined as seen in Figure 1.

```
# Data ---------------------------------------------------------------------- #
strFileName = 'products.txt'
lstOfProducts = []
```

Figure 1. Snippet of the python script defining the global variables.

I started with the class Product to define and save the data using the function initialize data() from the starter code provided as shown in Figure 2.

```python
class Product:
    """Stores data about a product:
    changelog: (When,Who,What)
    RRoot,1.1.2030,Created Class
    Nikita Daharia,06.08.2022,Modified code to complete assignment 8
    """

    @staticmethod
    def initialize_data(file_name: lstOfProducts = strFileName) -> lstOfProducts:
        lstOfProductss = [{"Product": "Hair dryer", "Price": "150.50"}, {"Product": "Hair straigtner", "Price": "100.00"}
        file = open(file_name, "w")
        for row in lstOfProducts:
            file.write(row["Product"] + "," + row["Price"] + "\n")
        file.close()
```

Figure 2. Snippet of python script defining and storing data.

Processing the data:

I created various functions for processing the data to and from a file and a list of products. These functions were used to save and read, add, remove, and write the data that was prepared from the previous class “Product”.

```python
@staticmethod
def add_data_to_list(product, price, list_of_rows):
    """ Adds data to a list of dictionary rows
    :param task: (string) with name of task:
    :param priority: (string) with name of priority:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """

    row = {"Product": str(product).strip(), "Price": str(price).strip()}
    lstOfProducts.append(row)
    return list_of_rows

@staticmethod
def remove_data_from_list(product, list_of_rows):
    """ Removes data from a list of dictionary rows
    :param task: (string) with name of task:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """

    for row in lstOfProducts:
        if (row["Product"] == product):
            lstOfProducts.remove(row)
            print("\nThe",product.lower(),"has been removed from the list!")
    return list_of_rows
```

Figure 3. Snippet of python script using functions to add and remove data from a file.

Presenting the data:

For presenting the data, a class IO was provided in the starter code. I added functions to display a menu of choices to the user, get menu choice from the user, show current tasks in the list and get the product name to be removed from the list. A few examples of these functions is shown in figure 4 and 5 below.

```python
@staticmethod
def input_product_to_remove():
    """  Gets the product name to be removed from the list
    :return: (string) with product
    """
    pass

    product = input("Which product would you like to remove?: ")
    product = product.capitalize()
    return product
```

Figure 4. Snippet of the python script to remove data from the list.

```python
@staticmethod
def output_menu_products():
    """  Display a menu of choices to the user
    :return: nothing
    """
    print('''
    Menu of Options
    1) Add a new Product
    2) Remove an existing Product
    3) Save Data to a File
    4) Exit Program
    ''')
    print()  # Add an extra line for looks
```

Figure 5. Snippet of the python script to display a menu of choices to the user as output

Handling exceptions in python:

Throughout the script, I added exception handlings to catch errors in data and to prevent the script of terminating the program.

```
# The exception handling will handle an error for a non-existing file
try:
    FileProcessor.read_data_from_file( file_name=strFileName, list_of_rows=lstOfProducts)  # read file data
except:
    print("\nThe file does not exist!")
```

Figure 6. Snippet of the python script displaying exception when the user inputs a non-existing file.

Run the Script by right clicking on the file and choosing Run.

```
        Menu of Options
        1) Add a new Product
        2) Remove an existing Product
        3) Save Data to a File
        4) Exit Program


Which option would you like to perform? [1 to 4] - 1

Enter a Product: Hair dryer
Enter a Price: 150.50

The hair dryer has been added to the list!
```

Figure 7. Snippet of output displayed in PyCharm Shell after running the python script.

Run the script on the Terminal window

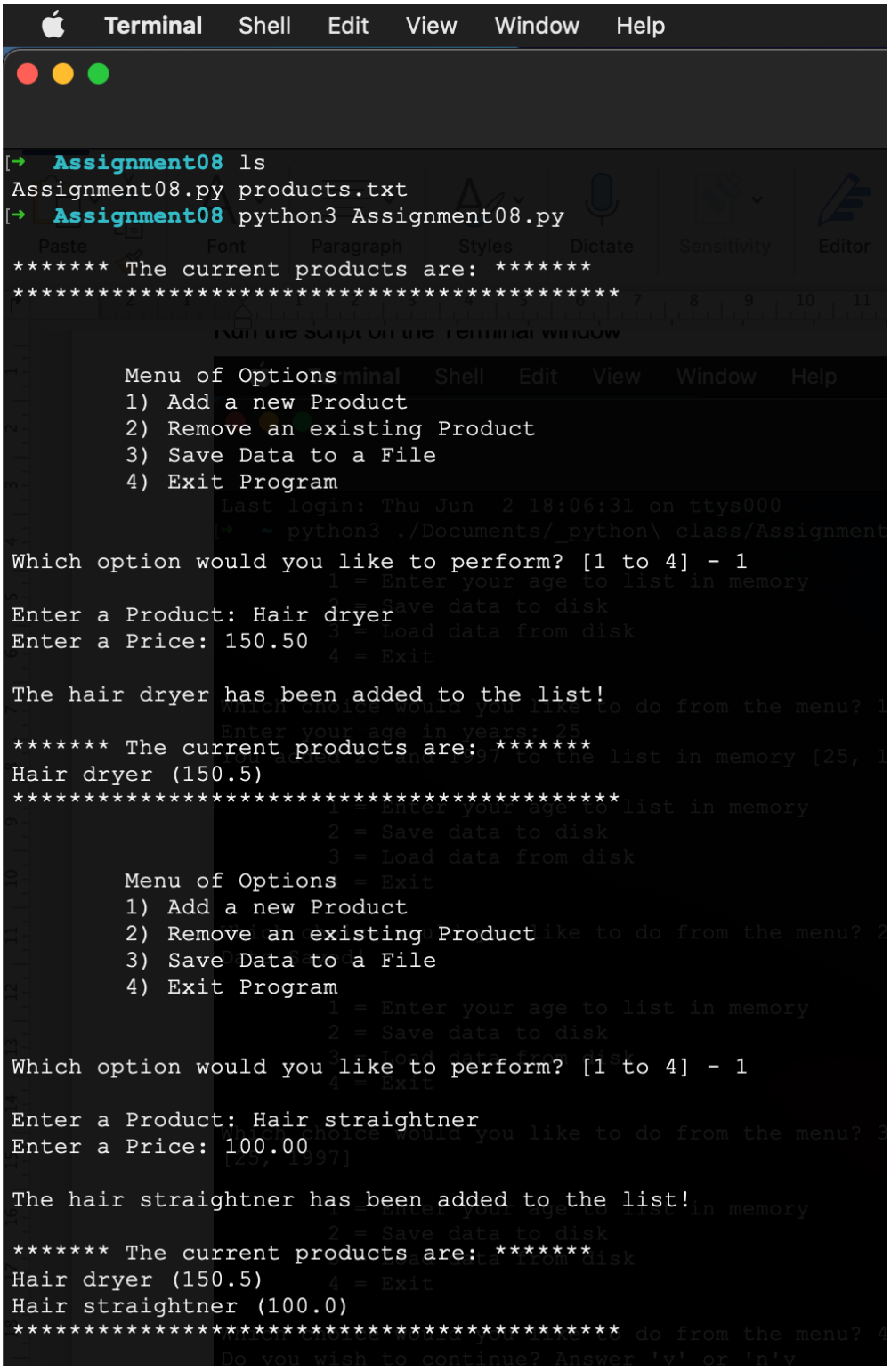

Figure 8. Output displayed in Terminal window after running the python script.

Lastly, the text file products.txt was located and opened that it worked and the updated data was saved.

Summary:

Python is a simple yet powerful language programming language that runs on Windows, Linus/Unix, and Mac OS. I used PyCharm to create a python script that allows a user to store products and their price into a text file using the pseudocode provided. The script was run both in PyCharm and Terminal . Finally, the code was verified by locating the text file products.txt.