Kyle Biondich

7/31/2023

IT FDN 110 A

Assignment 04

# For Loop Inventory

## Introduction

Week 4 of the course introduced data collections and for loops in python. The following paragraphs outline the methods that were used to capture user input from a menu, and either add a new item to a list, read the total list, or write the list to an external file.

## Intended Outcome

The intended outcome for this week is a script file that generates a menu for the user to choose from, and from this menu,

1. capture a list of household items and their estimated values
2. read the items contained is the list
3. write the list to an external file.

After each input from the user, the scripts checks to see if the user entered a specific value to exit the script.

```
PS D:\DEV\uw\assignments\_PythonClass\Assignment04> python .\HomeInventory.py

  Enter Q to quit at any time
  Select an Option:

  1) Add Data to List
  2) Display Current Data
  3) Exit and Save to File
```

***Figure 1: Intended Outcome: HomeIventory.py Menu***

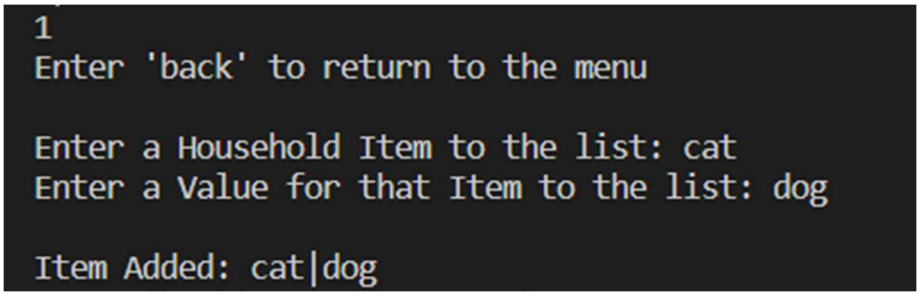


*Figure 2: Intended Outcome: HomeInventory.py Menu 1*

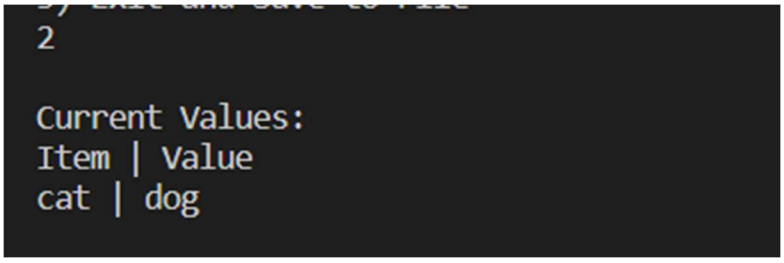


*Figure 3: Intended Outcome: HomeInventory.py Menu 2*

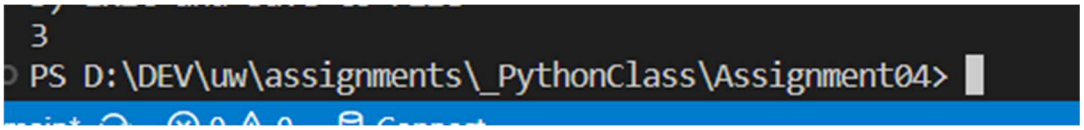


*Figure 4: Intended Outcome: HomeInventory.py Menu 3*

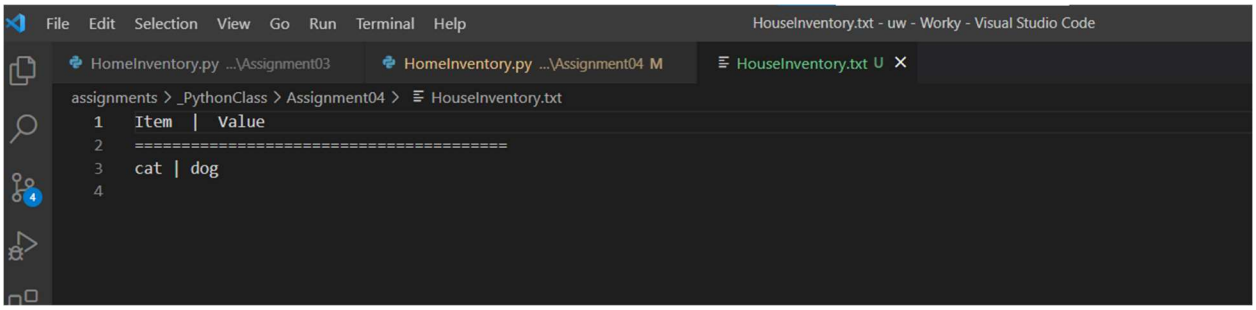


*Figure 5: Intended Outcome: HomeInventory.txt Saved File Contents*

## Step 1 – Display a Menu

After the header section, I define a list that will contain the menu choices that will be presented to the user. I also create a variable that will open a text file that the user's input will be saved to, and I create the list that will be written to that file.

```
10  # pseudo funs:
11      # Step 1
12      # Display a menu of choices to the user
13      # ("Add Data to List", "Display Current Data", "Exit and Save to File")
14  # ----------------------------------------------#
15  # store the list of menu items in a list
16  menuItems = ["Add Data to List", "Display Current Data", "Exit and Save to File"]
17  objFile = open("HouseInventory.txt","a")
18  householdItems = []
```

***Figure 6: Step 1***

## While Looping

Next, I wrap the rest of the script in a 'while' loop so the user will always be presented with the menu upon completion of the step they chose. I then print how to end the script and then use a 'for' loop to loop through the menuItems list and generate the list of choices for the user to select. Then, userChoice provides an input prompt to capture the user input (Python Loops, n.d.).

```
19  # Wrap everything in a while loop so the user returns to the menu after each selected action
20  while(True):
21      # present the user a menu to choose what to do
22
23      # Begin priting the menu: Tell the user what to do
24      print("\nEnter Q to quit at any time")
25      print("Select an Option: \n")
26
27      # use a for loop to loop through the menuItems list
28      for i in menuItems:
29          # for formatting purposes, grab the position of i in the list and add 1 to it
30          index = menuItems.index(i) + 1
31          # convert index to a string for the print function
32          stringIndex = str(index)
33          # print the position of i along with the stored value in the menuItems list at that position
34          print(stringIndex + ") " + i)
35
36      # capture the user choice
37      userChoice = input()
38      # store items in a list
39
```

***Figure 7: While Looping***

## Add New Items, See the Items, Save data

Next, the script checks the userChoice value and determines what to do. Entering

- ‘Q’ - quits the script
- 1 – adds a new item to the houseHoldItems list
- 2 – Prints the contents of houseHoldItems to the console
- 3 – saves the contents of householdItems to a text file

```
41      if(userChoice.lower() == "q"): break
42      # pseudo funs:
43          # Step 2
44          # Add a new item to the List(Table) each time the user makes that choice
45
46
47      elif userChoice == "1":
48          # open a saved file of the inventory list and append input to the bottom of the file
49          while(True):
50              print("Enter 'back' to return to the menu \n")
51              houseItem = input("Enter a Household Item to the list: ")
52              if(houseItem.lower() == "q"): break
53              elif(houseItem.lower() == "back"): break
54              else:
55                  houseItemValue = input("Enter a Value for that Item to the list: ")
56                  if(houseItemValue.lower() == "q"): break
57                  elif(houseItemValue.lower() == "back"): break
58                  else:
59                      item = houseItem + " | " + houseItemValue
60                      householdItems.append(item)
61                      print("\nItem Added: " + houseItem + "|" + houseItemValue)
62
63      # pseudo funs:
64          # Step 3
65          # Display the data in the List(Table) each time the user makes that choice
66      elif userChoice == "2":
67          print("\nCurrent Values:")
68          print("Item | Value")
69          for items in householdItems:
70              print(items)
71
72      # pseudo funs:
73          # Step 4
74          # Exit the program and save the data to a text file when the user makes that choice
75      elif userChoice == "3":
76          objFile.write("Item  |  Value \n")
77          objFile.write("------------------------------------ \n")
78          for items in householdItems:
79              objFile.write(items + "\n")
80          objFile.close()
81          break
```

***Figure 8: Script Options***

## Observations

One of the problems I encountered early on was making sure that the userChoice evaluation at lines 47, 66, and 75 was comparing to a string. It took me a while to remember that input only produces stings. I also realized that I don't need lines 50 or 57 as I initially had them coded as a 'continue', but realized that that wouldn't jump out to the first while loop.

## Summary

In summary, utilizing all the resources provided to the class and the online lecture, this paper outlines all the steps that were taken to create a python script that results in a successful execution of the intended outcome (Figure 1). Following the steps outlined above will allow for the audience to recreate the presented result.

## References

*Python Loops*. (n.d.). Retrieved from Tutortials Point: https://www.tutorialspoint.com/python/python_loops.htm