Name: Coco Jones
Date: February 10, 2023
Class: IT FDN 110 A
Module: 05
GitHub: https://github.com/cocopuffnagoya/IntroToProg-Python

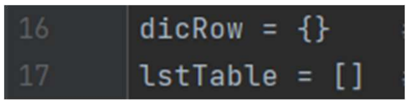
Lists and Dictionaries

Introduction

In module 05, I learned Python Lists and Dictionaries. This week's assignment was to create a ToDoList using Python dictionary and list. I am documenting the knowledge by answering the assignment questions and using the scripts that I create for the assignment where applicable.

What is the difference between a List and a Dictionary?

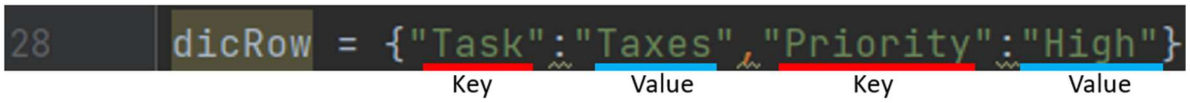
In the assignment this week I used both a List and a dictionary. (Fig 1) In Line 16 I declared dicRow as a dictionary and in Line 17 I declared lstTable as a List. One noticeable difference is that a Dictionary uses {} curly brackets while a List uses [] square brackets.



```
16    dicRow = {}
17    lstTable = []
```

Fig 1 – A Dictionary uses curly brackets and a List uses square brackets

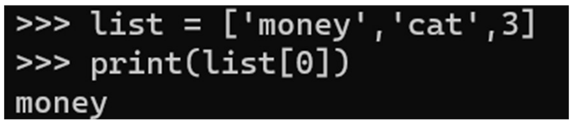
While a List can hold different data values separated with "**,**" (comma), a dictionary can only hold keys and values. In this week's assignment, I created dictionaries of "Task" and "Priority". "Task" and "Priority" are called "Keys" and are equivalent to columns in Excel and sort of a header of the data. Data values assigned to the keys – in this example, "Taxes" and "High" are called "Values". A dictionary holds sets of Keys and Values. A key and a value are separated with ":". Each dataset is separated with ",". (Fig 2)



```
dicRow = {"Task":"Taxes","Priority":"High"}
```

Fig 2 – A dictionary holds sets of Keys and Values

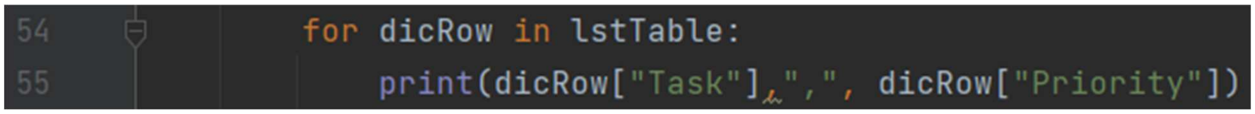
When you call a value in a List, you can use an subscript or an index number. For example, let's say that you have a List that contains 3 values 'money', 'cat' and 3. if you need to get the 0th Value of the list. You can get the value by indicating the index number [0]. (Fig 3)



```
>>> list = ['money','cat',3]
>>> print(list[0])
money
```

Fig 3 – you can use an index number to get the Nth number of the list

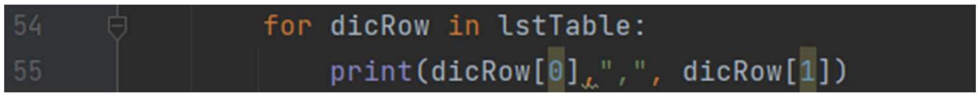
On the other hand, you cannot use an index number in a dictionary. You have to specify the key of the value you are getting from the dictionary. In the example below (Fig 4), the names of the keys “Task” and “Priority” are used to print each value.



```
for dicRow in lstTable:
    print(dicRow["Task"],",", dicRow["Priority"])
```

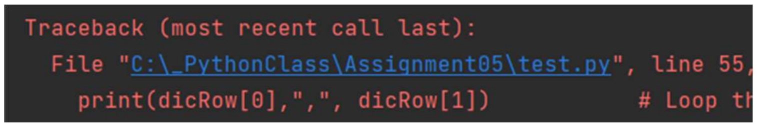
Fig 4 – Keys are used instead of Index Numbers for a Dictionary

If index numbers are used instead of keys (Fig 5), python gives you an error. (Fig 6)



```
for dicRow in lstTable:
    print(dicRow[0],",", dicRow[1])
```

Fig 5 – if index numbers are used for a dictionary…



```
Traceback (most recent call last):
  File "C:\_PythonClass\Assignment05\test.py", line 55,
    print(dicRow[0],",", dicRow[1])          # Loop th
```

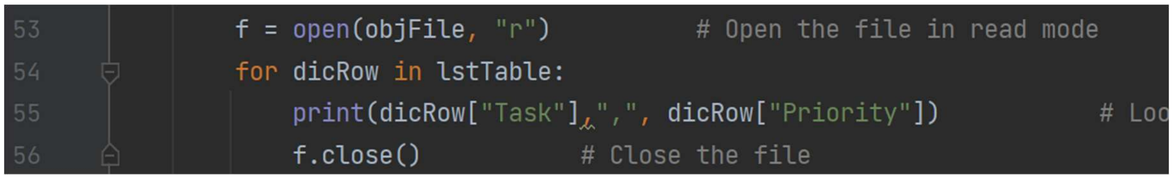
Fig 6 – Python gives an error message

What is the difference between an index and a key?

As mentioned above, one of the differences between a List and a Dictionary is that a List uses an Index (or subscript) to call a value while a dictionary uses a Key to call a Value. As seen in Fig 3 above, to get the 0th value from the list of [“money”, “cat”,3], you use the index number [0]. But if you wants to get a value from a Dictionary, you need to type out the names of the keys “Task” and “Priority” as seen in Figure 4,5 and 6.

How do you read data from a file into a List?

To read data from a text file into a List, you first need to open the file by using the open() function in Line 53 and inside the parenthesis specify the file name (or file path) and you need to put “r” indicate that you are opening the file in “r”ead mode. Then you use the for loop to go through the List in Line 54 and get the information inside the List. Also after you finish the loop, you need to close the file as seen in Line 56. (Fig 7)



```
53    f = open(objFile, "r")          # Open the file in read mode
54    for dicRow in lstTable:
55        print(dicRow["Task"],",", dicRow["Priority"])          # Loo
56        f.close()          # Close the file
```

Fig 7 – Use the open () function with “r” to open and load the data in a file to a List

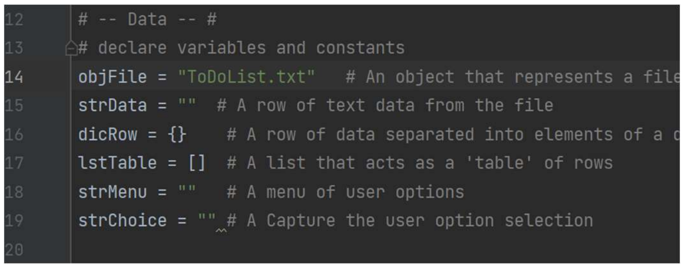
How do you read data from a file into a dictionary?

The scripts in Fig 7 above load data from the objFile to a List and each dictionary. The for loop script in Line 54 – 55 loops through the lstTable in the text file and prints the values assigned to the Task and Priority keys of each dictionary. (Fig 7)

What is the programming pattern called "Separations of Concerns?"?

The longer and more complicated your coding gets, the more you need to organize your code. In programming. there is this concept of Separation of Concerns that your coding should be organized and broken into 3 different blocks – Data, Processing and Presentation.

In this week's coding assignment, my code was broken into three different blocks. First, variables or constants were defined in the first block of Data. (Fig 8)

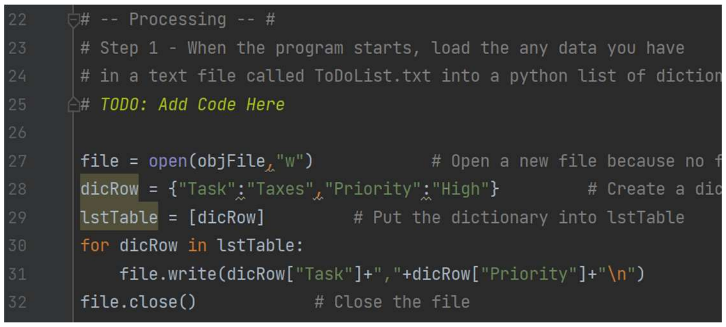


```
# -- Data -- #
# declare variables and constants
objFile = "ToDoList.txt"   # An object that represents a file
strData = ""  # A row of text data from the file
dicRow = {}    # A row of data separated into elements of a c
lstTable = []  # A list that acts as a 'table' of rows
strMenu = ""   # A menu of user options
strChoice = ""# A Capture the user option selection

```

Fig 8 – Variables are defined and data types are defined in the first Data section

Secondly, the data was processed before being presented to the user in the "Processing" section. In the assignment example, in this section, a dictionary was defined with the two keys "Task" and "Priority". And the dictionary was stored into the list of LstTable and the first row of data was written into the text file. (Fig 9)

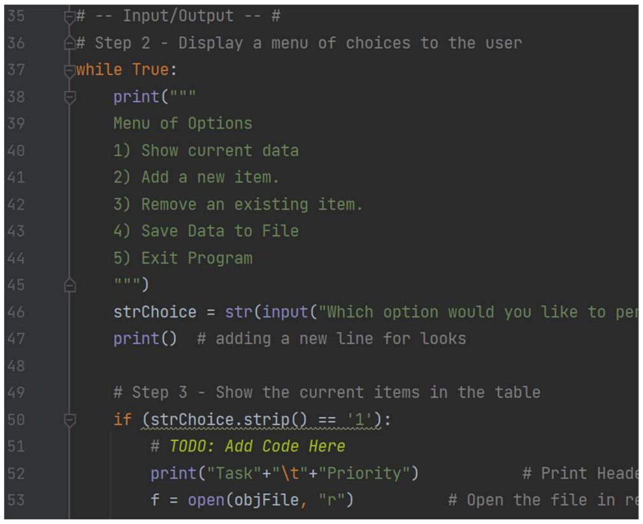


```
# -- Processing -- #
# Step 1 - When the program starts, load the any data you have
# in a text file called ToDoList.txt into a python list of diction
# TODO: Add Code Here

file = open(objFile,"w")            # Open a new file because no f
dicRow = {"Task":"Taxes","Priority":"High"}         # Create a dic
lstTable = [dicRow]         # Put the dictionary into lstTable
for dicRow in lstTable:
    file.write(dicRow["Task"]+","+dicRow["Priority"]+"\n")
file.close()            # Close the file
```

Fig 9 – The first dictionary is created and stored into a List and the data is written into a file

And finally, the section of Presentation (Input/Output) was written and this section allows the user to input or output the data. (Fig 10)



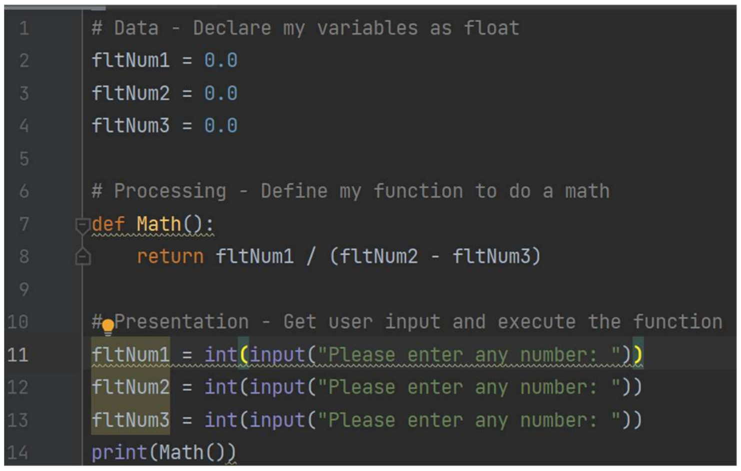
```
# -- Input/Output -- #
# Step 2 - Display a menu of choices to the user
while True:
    print("""
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program
    """)
    strChoice = str(input("Which option would you like to per
    print()  # adding a new line for looks

    # Step 3 - Show the current items in the table
    if (strChoice.strip() == '1'):
        # TODO: Add Code Here
        print("Task"+"\t"+"Priority")           # Print Heade
        f = open(objFile, "r")          # Open the file in re
```

Fig 10 – The block of Presentation (Input/Output)

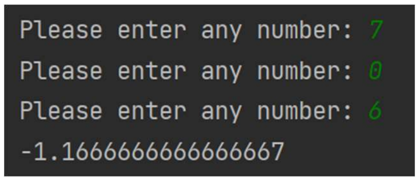
How would you use a function to organize your code?

You can create your own function and use it throughout your program. Your function may be made of multiple functions or multiple calculations, but once your function is defined, you can use it as many times as you like throughout your code so you can organize your code. A complicated math function is created in Line 7 – 8 and is executed in Line 14 (Fig 11). The result of the program is shown in Fig 12. The function Math () can be used many times throughout the code.



```
1   # Data - Declare my variables as float
2   fltNum1 = 0.0
3   fltNum2 = 0.0
4   fltNum3 = 0.0
5
6   # Processing - Define my function to do a math
7   def Math():
8       return fltNum1 / (fltNum2 - fltNum3)
9
10  #Presentation - Get user input and execute the function
11  fltNum1 = int(input("Please enter any number: "))
12  fltNum2 = int(input("Please enter any number: "))
13  fltNum3 = int(input("Please enter any number: "))
14  print(Math())
```

Fig 11 – A function is defined and used in the code

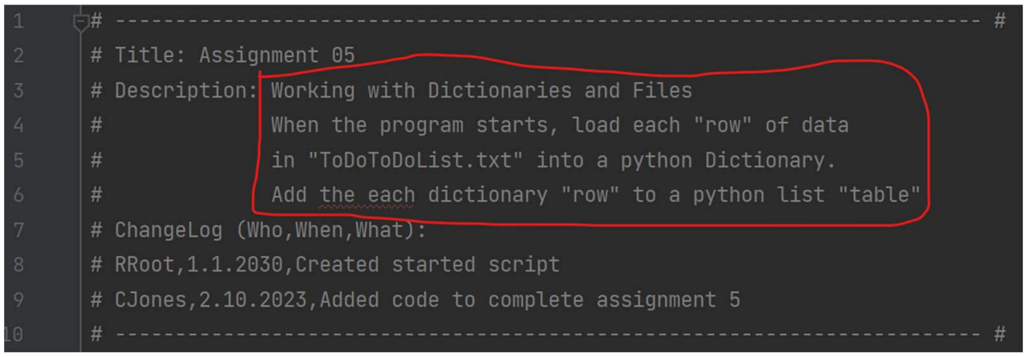


```
Please enter any number: 7
Please enter any number: 0
Please enter any number: 6
-1.1666666666666667
```

Fig 12 – Result of the math program

Why is a script template useful?

A script template is useful. It gives consistency to your script and especially if you are working with your team, a template helps you code consistently with your team and understand the flow and help you start coding easily. In the assignment, I used the assignment starter template that Professor Root created. This was useful because the header information (e.g. Title, Description, Change log) was already there. I only needed to add my information. Also, the description in the header lays out the steps that need to be taken while the assignment question was not clear enough. (Fig 13)



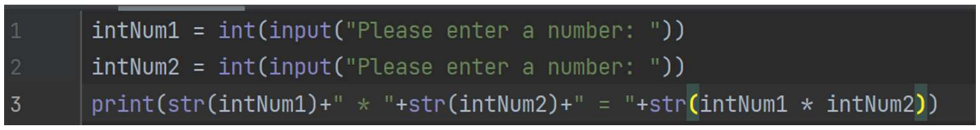
```
# ---------------------------------------------------------------------- #
# Title: Assignment 05
# Description: Working with Dictionaries and Files
#              When the program starts, load each "row" of data
#              in "ToDoToDoList.txt" into a python Dictionary.
#              Add the each dictionary "row" to a python list "table"
# ChangeLog (Who,When,What):
# RRoot,1.1.2030,Created started script
# CJones,2.10.2023,Added code to complete assignment 5
# ---------------------------------------------------------------------- #
```

Fig 13 – The steps to be taken are laid out in the header

Also, not all but part of the code was already provided by Professor Root and the structure of Data > Processing > Presentation was there, so it was very easy to start this assignment.

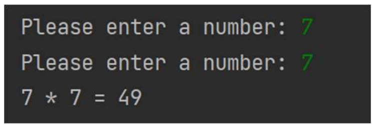
Why is error handling using Try-Except recommended?

Error handling using Try-Except is recommended because python returns its error message that does not always make sense. So it is wise to show your own error message for a situation where your code could error. As a test, I created a simple math script like below (Fig 14). It asks the user to enter 2 numbers and python multiplies the numbers and shows the answer. (Fig 15)



```
intNum1 = int(input("Please enter a number: "))
intNum2 = int(input("Please enter a number: "))
print(str(intNum1)+" * "+str(intNum2)+" = "+str(intNum1 * intNum2))
```

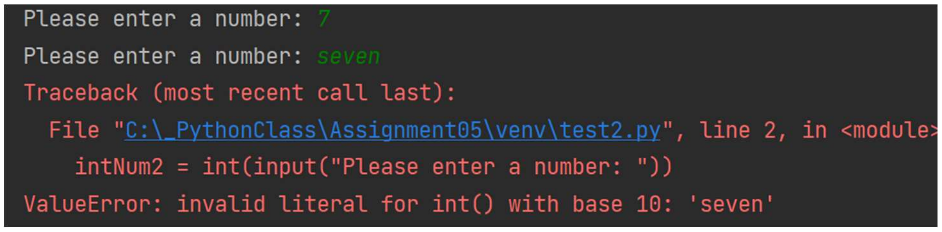
Fig 14 – Test script



```
Please enter a number: 7
Please enter a number: 7
7 * 7 = 49
```

Fig 15 – The answer is shown

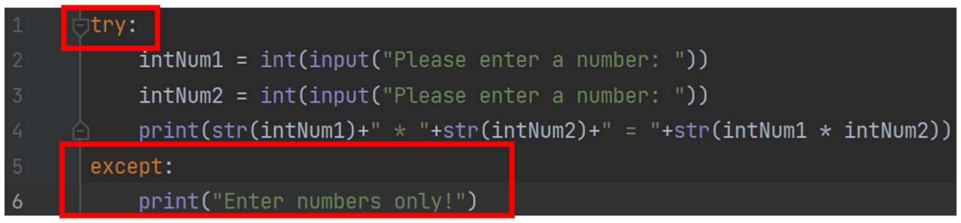
This math program works as long as the user enters 2 numbers that can be converted to integers. If the user enters a string, that cannot be converted to an integer by the int() function, python returns an error in red below. And the error message is hard to understand. (Fig 16)



```
Please enter a number: 7
Please enter a number: seven
Traceback (most recent call last):
  File "C:\_PythonClass\Assignment05\venv\test2.py", line 2, in <module>
    intNum2 = int(input("Please enter a number: "))
ValueError: invalid literal for int() with base 10: 'seven'
```

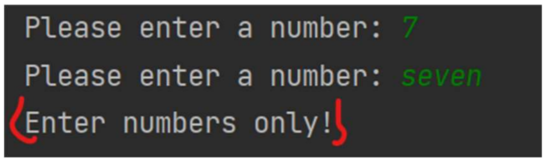
Fig 16 – Python error message is not easy to understand

To show a more understandable message, you can create your own error message using Try-Except. (Fig 17). The result will be shown in Fig 18.



```
try:
    intNum1 = int(input("Please enter a number: "))
    intNum2 = int(input("Please enter a number: "))
    print(str(intNum1)+" * "+str(intNum2)+" = "+str(intNum1 * intNum2))
except:
    print("Enter numbers only!")
```

Fig 17 – Try-Except is used in the program so that your custom error message is displayed



```
Please enter a number: 7
Please enter a number: seven
Enter numbers only!
```

Fig 18 – Result of the program when there is an error.

What is GitHub, and why is it used?

GitHub is a portal that you can publish your code and collaborate with your teammate or other people. You can publish your code and your work. In the UW class, it is required to publish your homework and code in GitHub from this week. The purpose is peer review and providing feedback to each other.

Summary

In module 05, I learned Python Lists and Dictionaries. I documented the knowledge that I acquired this week by answering the assignment questions.

References

https://www.youtube.com/watch?v=P5wOsnPjn6Y Intro to Python Mod05 (Last accessed: 2/13/2023)

Python Programming, Third Edition, Course Technology, Michael Dawson 2019 ISBN-13 978-1-4354-5500-9