

Name: Axel Adams
Date: Feb, 28, 2023
Course: Intro to Programming (PYTHON)
<https://github.com/H-imperialis/Intro-to-Programming---Module-07>

Assignment 07: The Pickle Module and Handling Exceptions

Introduction:

In this assignment, we are tasked with researching and executing the pickle module and the practice of exception handling within a program of our own design. Pickling is a manner of manipulating data into a binary format. Exception handling is a preemptive manner of dealing with errors in execution of a program. In this assignment, I apply both of these tools to the code generated for Assignment 5 - the To Do List. I chose to use this code to showcase these techniques because we are all familiar with it and by applying these techniques to a script we are already familiar with, allows us to see how it works differently.

Pickling

Topic: What is Pickling?

The clearest explanation I have found for pickling was in “Think Python: How to Think Like a Computer Scientist” by Allen B Downey. Pickling is a way to overcome data with different types without having to manually convert that data to a coherent type - i.e. a byte stream. The pickle module converts almost any object into a serialized string. Two operations `dump()` and `load()` allow the coder to convert a collection of data to a “pickled” file and then convert that back to a human-readable file, respectively.

Topic: Pickling the To Do List

The most helpful resource I identified for application of the pickle module was at <https://www.datacamp.com/tutorial/pickle-python-tutorial>. This demonstrated in a straight forward way how to convert a file to pickle format. The reason I felt that this was such a good explanation was that it explains the benefit of pickling (i.e. preserving the complex data arrangements such as nested lists or complex data streams such as dictionaries - as opposed to just saving everything in a string). An important first step is to import the pickle module. This will allow you to perform the pickling functions. I have adjusted the ToDoList code to include pickling options in the menu:

Figure 1: New Menu

```
while (True):  
    print("""  
        How would you like to alter your To Do List?  
        Menu of Options  
        1) Show current data  
        2) Add a new item.  
        3) Remove an existing item.  
        4) Save Data to File  
        5) Pickle My List  
        6) DePickle and Display My List  
        7) Exit Program  
        """)  
    strChoice = str(input("Which option would you like to perform? [1 to 8] - "))
```

Figure 2: Pickle Codes

```
#Step 7 - Pickle My List
elif(strChoice.strip() == '5'):
    with open('ToDoList.pkl', 'wb') as f: #We create a separate pickled file, using the "write binary"
        pickle.dump(lstTable, f) #we write the pickled data
        f.close()

#Step 8 Depickle and Display list
elif (strChoice.strip() == '6'):
    with open('ToDoList.pkl', 'rb') as f: # we open the list and read in read binary context
        ToDoList_pickle = pickle.load(f)
        print(ToDoList_pickle)
```

We can then compare the output of the text file vs the pickle file.

Figure 3: Text Data vs Pickle Data

<pre>wash dishes,high walk dog,high feed dog,high water plants,medium</pre>	<pre>ÄiN]i{}i(âTaskiâ wash dishesiâPriorityiâhighiu} i(hâwalk dogihâhighiu}i(hâfeed dogihâhighiu}i(hâ water plantsihâmediumiue.</pre>
ToDoList.txt	ToDoList.pkl.txt

Topic: Pickle Caveats

It is important to know that you should only unpickle data that you trust. Within an unknown serialized file, there may be malicious code that may become problematic when converted from binary format.

Exception Handling:

Topic: What is Exception Handling?

Exception handling is when the coder tries to preemptively manage errors that could arise during the execution of the code. The most straight forward explanation of exception handling that I encountered was at <https://www.freecodecamp.org/news/exception-handling-python/>. I felt that this was a great explanation of exception handling because it clearly explained how to apply the try/except clauses. This website demonstrates an example in which these clauses are used in a similar situation in which the user is tasked with removing a student from a student dictionary. In the To Do List code, one of the major user errors that could arise is if the user input an item that did not exist on the to do list.

Figure 4: Exception Handling - Dealing with Incorrect Input for Item to Remove

```
# Step 5 - Remove a new item from the list/Table
elif (strChoice.strip() == '3'):
    if len(lstTable) == 0:
        print("The To Do List is empty.")
    else:
        strTask = str(input("Which task would you like to remove?:")).lower()
        #for dicRow in lstTable:
        while True:
            try:
                for row in lstTable:
                    if row["Task"].lower() == strTask.lower():
                        lstTable.remove(row) # we remove the item from the list table
                        print("This item has been removed from your To Do List!")
            except IndexError:
                print("This task is not on your To Do List!")
            else:
                print("This task is not on your To Do List!")
            break
```

In this instance, I use a while loop to after the input and try to remove the list containing the users input. If the user input a incorrect item that does not exist on the ToDoList, it generates an error message stating that this item is not on the to do list. There are a number of inbuilt exceptions that exist in Python. A summary of them is well-described in <https://docs.python.org/3/library/exceptions.html>. This is useful because it describes specific exceptions that can be applied in designing a try/except structure. See Figure 5 for results of trying to remove an item that does exist on the ToDoList.

Exception handling can also be use to manage if the user does not pick an item from the menu. A while loop is used to cycle through the To Do List menu and the options linked to the different options is placed within the “try:” clause and the except clause is used at the end of the script to prompt the user to select from the menu.

Figure 5: Result of Inputting an Incorrect Item

- 4) Save Data to File
- 5) Pickle My List
- 6) DePickle and Display My List
- 7) Exit Program

Which option would you like to perform? [1 to 8] - 12

Please choose from the choices above.

How would you like to alter your To Do List?

Menu of Options

- 1) Show current data
- 2) Add a new item.
- 3) Remove an existing item.
- 4) Save Data to File
- 5) Pickle My List
- 6) DePickle and Display My List
- 7) Exit Program

Which option would you like to perform? [1 to 8] - cat

Please choose from the choices above.

Conclusion:

Pickling and exception handling are two useful tools for the developer. The pickle module allows the user to convert data to a binary format while preserving the complexity inherent to nested lists and dictionaries. Exception handling is a way of dealing with an incorrect user input that might result in an error that could result in execution of the program. In this case, I have used these two tools to improve the To Do List program from earlier in the course.