Ethan Morgan

8/24/2021

IT FDN 110 A, Foundations of Programming (Python)

Assignment 07

GitHub Link: <https://github.com/ethan-morgan/IntroToProg-Python-Mod7> (External Link)

Intro to Programming (Python)

# Introduction

Assignment 07 provided an opportunity to working error handling and ‘pickling’ in Python. I chose to split up the error handling and ‘pickling’ examples into two different scripts. This was done to explain clearly how each aspect is used in Python.

# Error Handling Script

Error handling is important in scripting because it provides the user a better way understand errors that may arise in the code. The example I chose to use was a script that would not error out if the user’s input was not what was expected. Figure 1 shows the ‘Assignment07\_ErrorHandling.py’ script. The script includes multiple comments that provides guidance through the code and how it it runs.

# ------------------------------------------------- #  
# Title: Assignment 07, Error Handling  
# Description: Create an example code with error handling  
# ChangeLog: (Who, When, What)  
# Ethan Morgan,08/24/2021,Created Script  
# ------------------------------------------------- #  
  
# This script provides an example of error handling in Python.  
# It takes user input, but the script requires a numeric input.  
# The script uses a try/except error handling to deal with wrong inputs.  
# If a string is input, the code will request a new input until a number input.  
  
while True: # 'while' statement used to continue to ask for user input  
 print() # print blank line for output clarity  
 print("Input a Float Value Below!")  
 user\_input = input("Input Value: ") # user input requested  
 try:  
 user\_input\_float = float(user\_input) # converts user input into float type data  
 # note, if a string is input for 'user\_input' the code skips to the 'except' below  
 print() # print blank line for output clarity  
 print("The Float Value You Entered Was: ",user\_input\_float) # print out user float input  
 print("Congrats you Entered a Float Value and Not a String!") # print feedback user did it correct  
 break # 'break' used to stop code once user input was correct  
 except:  
 print() # print blank line for output clarity  
 # the user's input was not a float or integer  
 print("Your Input of",user\_input,"is a String!") # print feedback user didn't input correctly  
 print("Please Input a Number, Not a String!") # print feedback user didn't input correctly

Figure ‘Assignment07\_ErrorHandling.py’ Script

A ‘while’ loop is used continually ask the user to input a numeric value. Then a ‘try/except’ error handling capability is used to prevent the script from erroring out if the user inputs something other than a numeric value. If the user inputs a string, the first line of code in the ‘try’ clause will push the user into the ‘except’ clause and inform the user input a number, not a string.

Figure 2 and Figure 3 show the ‘Assignment07\_ErrorHandling.py’ script running in PyCharm and Command OS/Shell respectively. The user initially input ‘Potato’ and the script passed back that the user needs to input a numeric value and not a string. As discussed previously, the try/except error handling was used to prevent the script from erroring out and giving the user the opportunity to input a numeric value to run through the script completely.

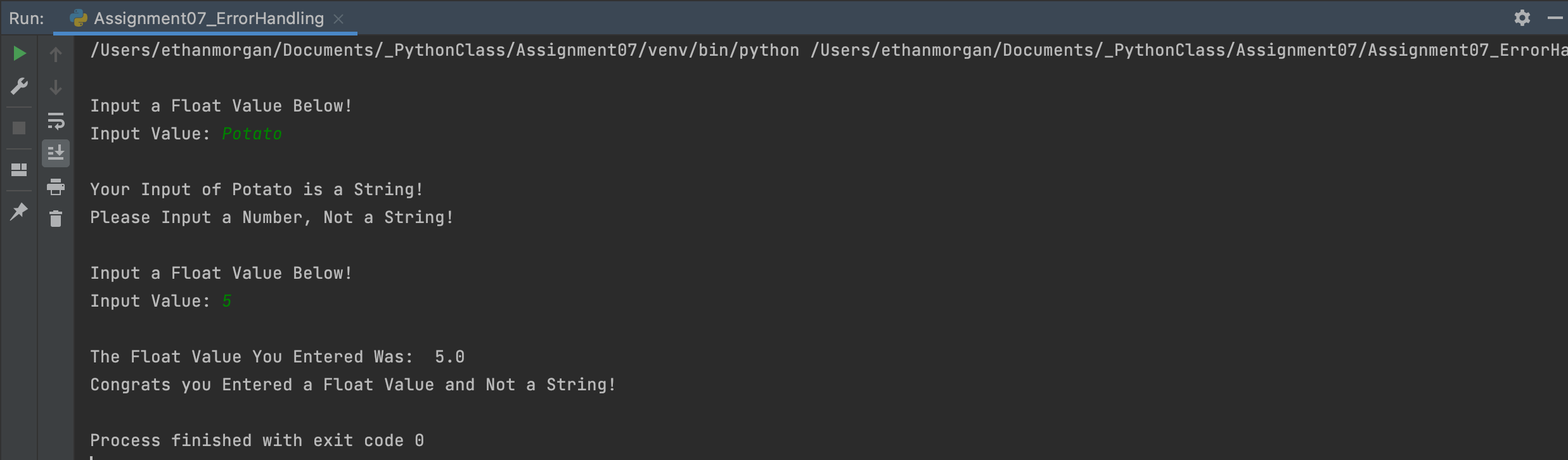


Figure ‘Assignment07\_ErrorHandling.py’ Script Running in PyCharm

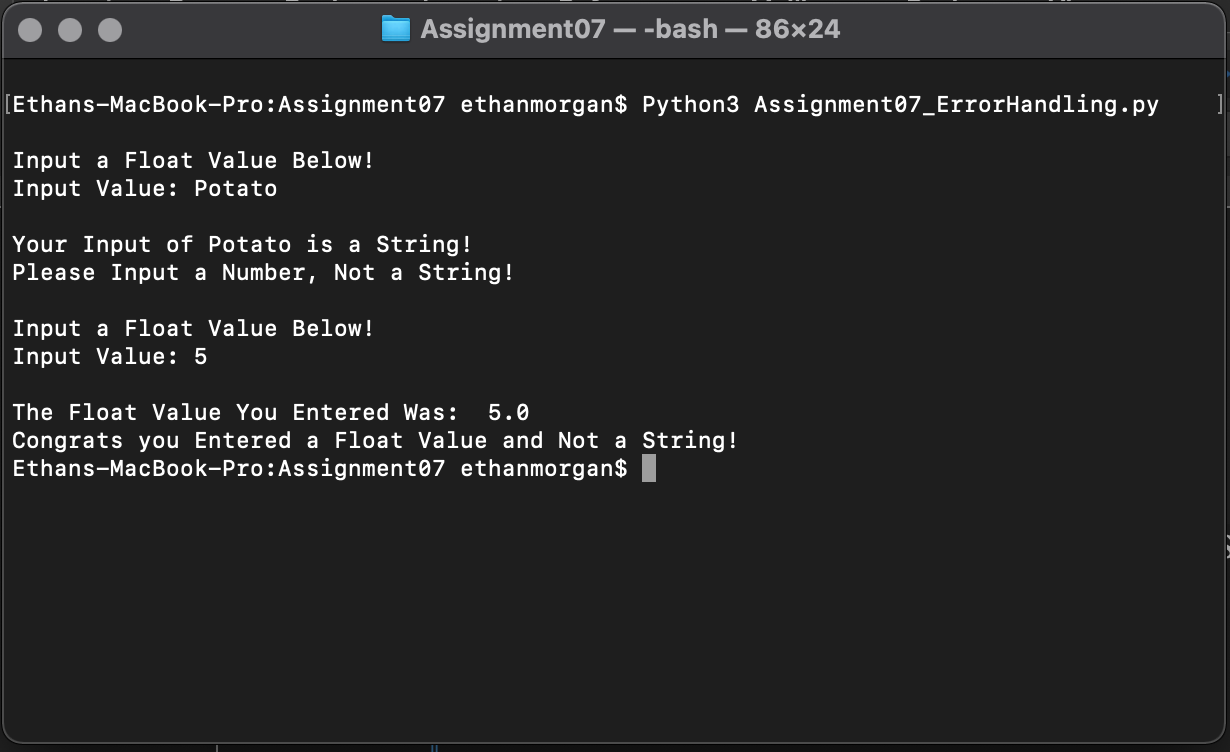


Figure ‘Assignment07\_ErrorHandling.py’ Script Running in Command Shell/OS

The external site link below was very good at describing how a ‘try/except’ statement is to be used in Python.

Web Article Link: <https://docs.python.org/3/tutorial/errors.html> (External Site)

This article was helpful because it gave a detailed description of how a ‘try/except’ statements runs through the code. I especially like the discussion about embedded ‘try/except’ statements.

# Pickling Script

Pickling is a function that can be called to help automate some tasks and deal with multiple variations of data. The scrip shown in Figure 4 imports ‘Pickle’ to help pull various data from a List, writes it to a binary file that is difficult/impossible to comprehend visually, and then open the binary file to print the data to show that nothing has changed to said data.

# ------------------------------------------------- #  
# Title: Assignment 07, Pickling  
# Description: Create an example code that uses Pickling  
# ChangeLog: (Who, When, What)  
# Ethan Morgan,08/24/2021,Created Script  
# ------------------------------------------------- #  
  
import pickle # import the 'pickle' code to be used in this script  
  
# Dictionary of Floats/Integers, Stringers, and Booleans  
dictionary\_data = {  
 'Floats\_or\_Integers': [1, 2.0],  
 'Strings': ("Activities", "Actions"),  
 'Booleans': (True, False)}  
  
# Print the dictionary define previously  
print()  
print("This is a Dictionary Prior to Being Pickled...")  
print(dictionary\_data)  
  
# Creates a file or rewrites a file named 'dictionary\_data.dat'  
# The 'wb' writes the data to a file in binary  
with open('dictionary\_data.dat', 'wb') as row:  
 pickle.dump(dictionary\_data, row)  
  
# Opens the binary file and inputs the data in a different dictionary  
# The 'load' command pulls the data from the binary file  
with open('dictionary\_data.dat', 'rb') as row:  
 dictionary\_data\_unbinary = pickle.load(row)  
  
# Print the dictionary to show that the data hasn't changed  
print()  
print("This is the same Dictionary Post Being Pickled...")  
print(dictionary\_data\_unbinary)

Figure ‘Assignment07\_Pickling.py’ Script

Figure 5 and Figure 6 show the ‘Assignment07\_Pickling.py’ script running in PyCharm and Command OS/Shell respectively. Note the dictionary that was defined in the script prior to being written to a binary file is the same as the dictionary after reading from a binary file.

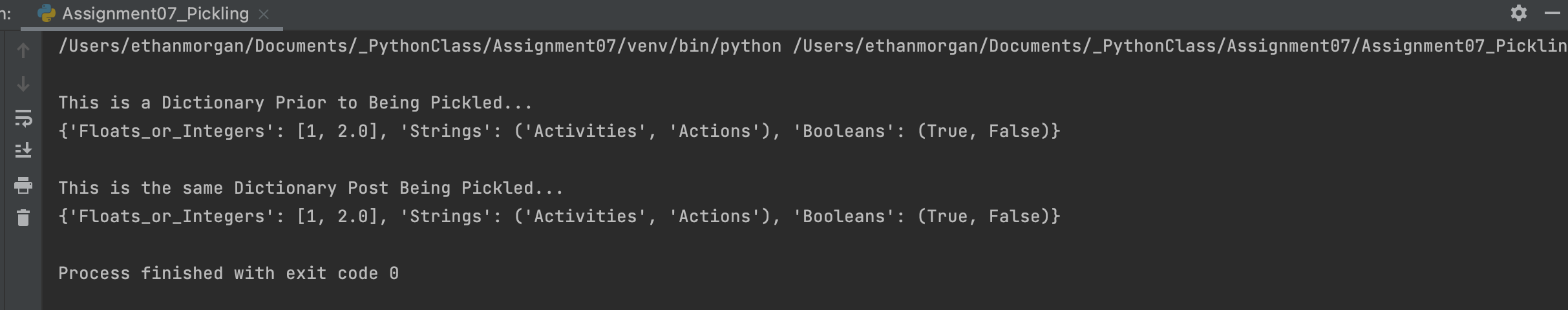


Figure ‘Assignment07\_Pickling.py’ Script Running in PyCharm

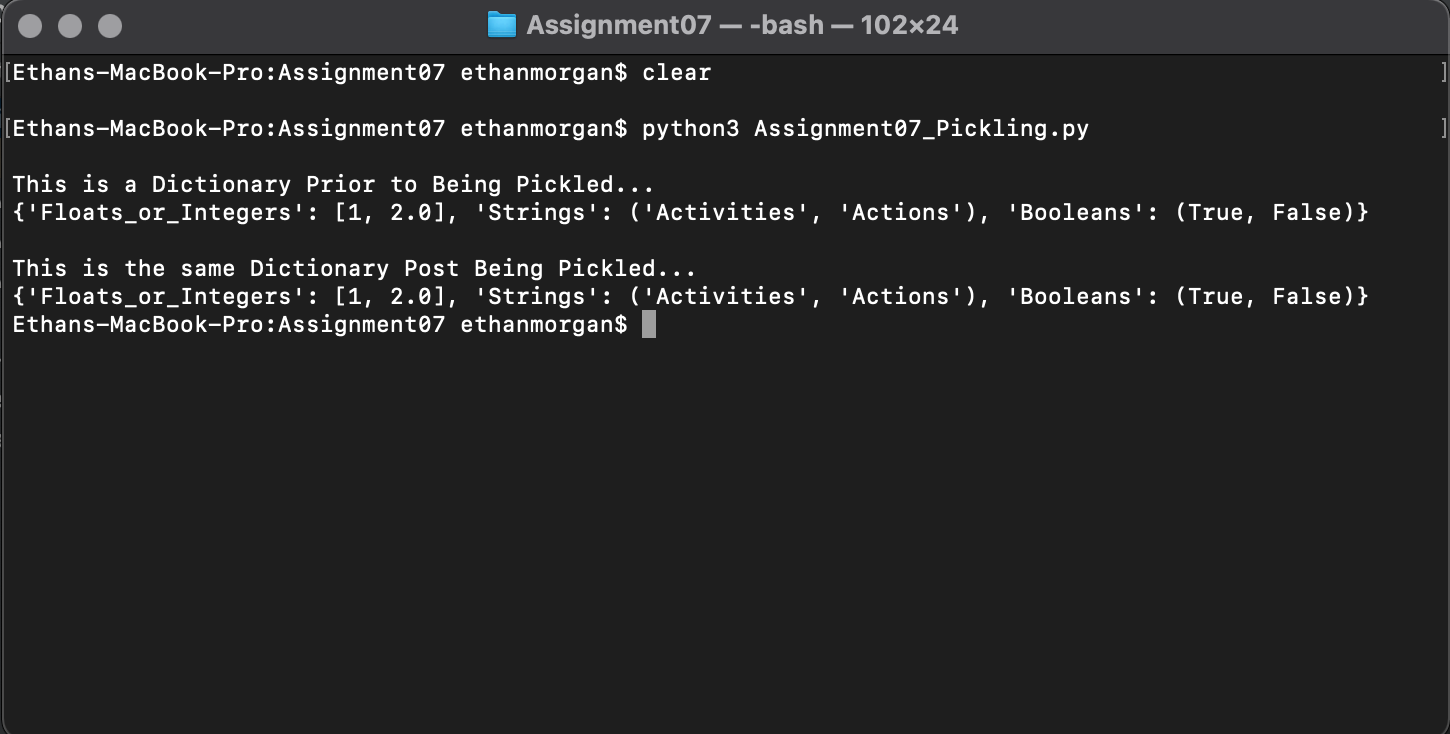


Figure ‘Assignment07\_Pickling.py’ Script Running in Command Shell/OS

Figure 7 shows the actual binary file and note that it is hard to decipher exactly what is in the file even though some words are visable.

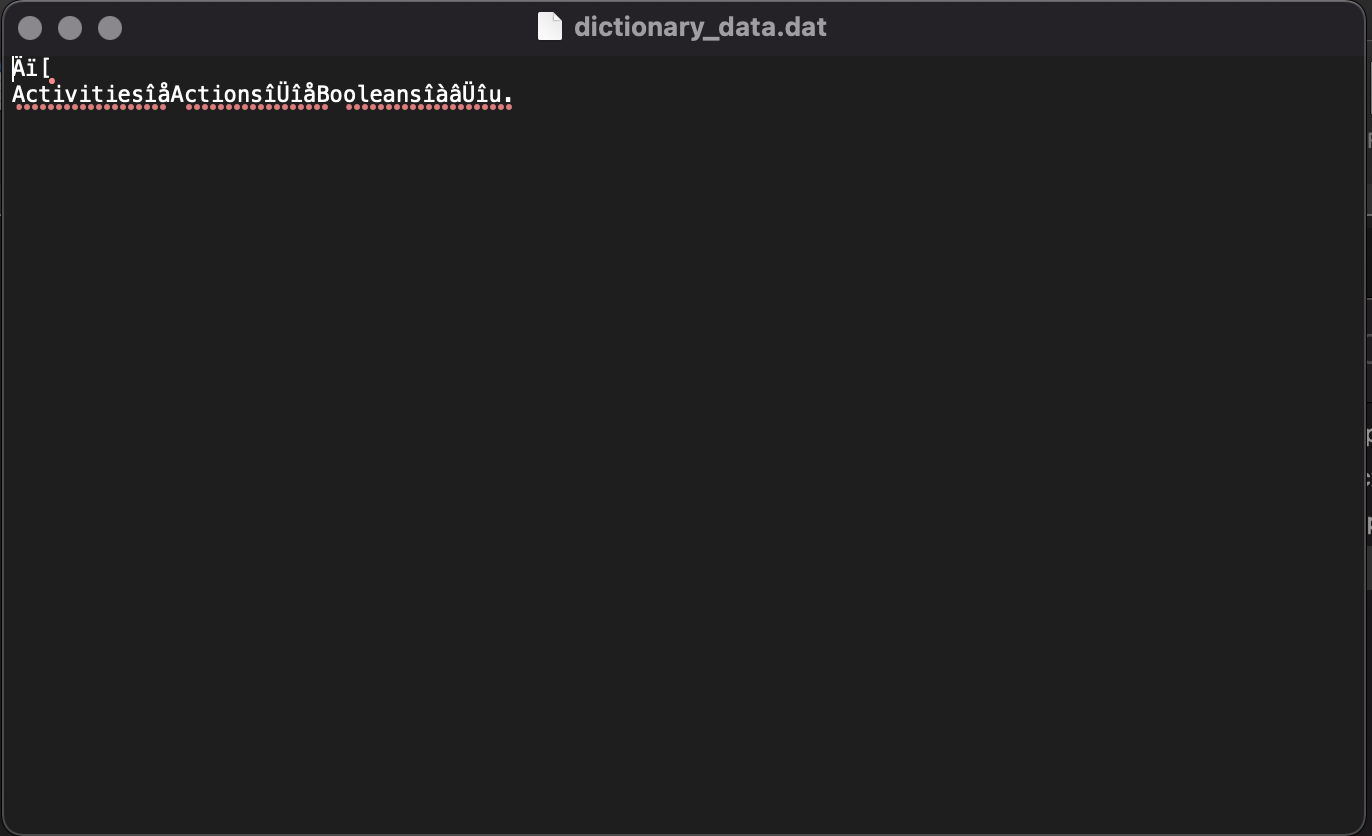


Figure ‘Assignment07\_Pickling.py’ Script Created Binary File

The web article shown below was helpful in describing what ‘Pickle’ does in Python. While the script shown in Figure 4 only does a fraction of what ‘pickling’ can be done, it does show some of the capabilities. I found this article helpful because it shows the full breadth of what ‘Pickle’ can do.

Web Article Link: <https://docs.python.org/3/library/pickle.html> (External Site)

# Summary

Assignment 07 provided us to work with ‘try/except’ statements and using Pickle as a way of working with binary files. This assignment also forced us to do some of our own research on topics within Python.