Vivian Ta

IT FDN 100 A

Assignment 7 Notes

Nov. 17, 2018

I separated my python exception handling and pickling code into two separate files.

Python Exception Handling:

I did most of my reading from Python’s official docs on the exception handling piece since it had a lot of different use cases: <https://docs.python.org/3/tutorial/errors.html>.

This is what I came up with after reading up on it for the first time.

# Python Exception Handling Code  
import sys  
  
def add\_numbers():  
 try:  
 userArg = int(sys.argv[1]) # define arg1 from user  
 userArg2 = int(sys.argv[2]) # define arg2 from user  
 combined = userArg + userArg2 # add both args  
 print(combined) # print the result  
 # catch not enough args provided error  
 except IndexError as e:  
 print("Please provide at least 2 args, eg: python3 <script> <arg> <arg2>")  
 # catch args that are not numbers error  
 except ValueError as e:  
 print("This function can only add numbers")  
 # catch all for everything else  
 except Exception as e:  
 print("Unknown Error")  
  
add\_numbers()

After reading further down, I realized that I wanted to catch one more error: the user entering in more than 2 args. Reading through the docs, I realized that I can do a raise to stop the code for a manual exception. I added that conditional into the def.

# Python Exception Handling Code  
import sys  
  
def add\_numbers():  
 try:  
 userArg = int(sys.argv[1]) # define arg1 from user  
 userArg2 = int(sys.argv[2]) # define arg2 from user  
 if len(sys.argv) > 3: raise Exception("Too many args")  
 combined = userArg + userArg2 # add both args  
 print(combined) # print the result  
 # catch not enough args provided error  
 except IndexError as e:  
 print("Please provide at least 2 args, eg: python3 <script> <arg> <arg2>")  
 # catch args that are not numbers error  
 except ValueError as e:  
 print("This function can only add numbers")  
 # catch all for everything else  
 except Exception as e:  
 print(e)  
  
add\_numbers()

Python Pickling:

It was a bit confusing at first to understand the need for pickling through the instructor’s video, but after reading up on [serialization](https://en.wikipedia.org/wiki/Serialization) it started making more sense on why it’s necessary. From my understanding, it basically translates non-readable data into a human-readable format.

My sources for this are:

<https://docs.python.org/2/library/pickle.html>

<https://pythontips.com/2013/08/02/what-is-pickle-in-python/>

Here’s my code example for pickling:

# Python Pickling Code  
import pickle  
  
# creating a dict to be dumped by pickle  
dict = {"dairy": "milk, eggs, butter",  
 "grains": "wheat, barley",  
 "vegetables": "carrots, tomato, celery"}  
  
# opening a file in binary write mode  
file = open("food\_pyramid.dat", "wb")  
  
# dump the dict to the file  
pickle.dump(dict, file)  
  
# close the file  
file.close()  
  
# open the file in binary read mode  
dat = open("food\_pyramid.dat", "rb")  
  
# assign the load to a var  
food\_pyramid = pickle.load(dat)  
  
# print the var for the data  
print(food\_pyramid)  
  
dat.close()

When I wanted to add another value to the file, I realized that I could do that pretty easily with dump as it’ll just append to the end of the file, but when reading the file back, it only does it line by line so I added another load for it. The stackoverflow link shows a few other examples such as packing the data into a single object and then unpacking it but I don’t quite understand them yet so didn’t venture down that route yet.

<https://stackoverflow.com/questions/15463387/pickle-putting-more-than-1-object-in-a-file>

The final code is located in `assignment7-pickling.py`.