Felix Chen

2/28/23

IT FDN 110A

Assignment07

https://f3liz.github.io/IntroToProg-Python-Mod07/

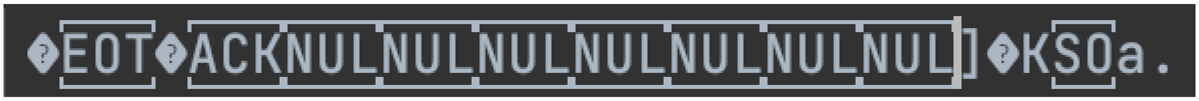
Pickling and Error Handling

Introduction

This week's module we had to research pickle and structured error handling. We also had to demonstrate it in our script. My script this week was simple, using pickle and error handling in the same script it will pickle a user's favorite number and unpickle back out to the user. It will also catch errors, for example if a user didn't use a numeric value.

Pickling

From what I have researched on my own, pickling in python is serializing and de-serializing python object structures. Even after reading articles and watching a few videos, I still didn't completely understand it so I just messed around with the pickle in python. I have come to understand that I can use it to store and retrieve data with fewer lines of code. Link:https://docs.python.org/3/library/pickle.html#:~:text=%E2%80%9CPickling%E2%80%9D%20is%20the%20process%20whereby,back%20into%20an%20object%20hierarchy (This website helped me the most).



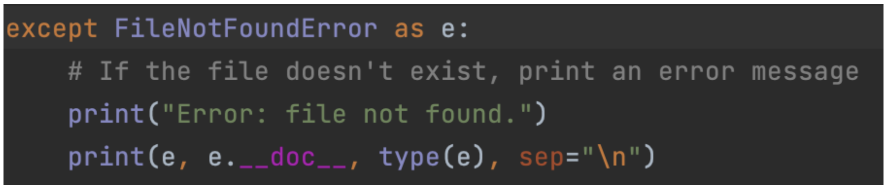
```
�EOT�ACKNULNULNULNULNULNULNUL]�KSOa.
```

Figure 1: Pickled data from messing around with pickle in PyCharm

Error Handling

In the video for this week's module, Randal had gone over structured exception handling (try - except), exception class, catching specific exceptions, and creating custom exception classes. I had used try - except blocks before in previous assignments and labs from this course so I felt comfortable with that. The catching specific exceptions part of the lecture was interesting to me because it was almost like trying to predict any errors that could pop up and getting ahead of those before it could happen. That is something I already try to do in my assignments because I try to think of things that could go wrong for the user and prevent it from happening while trying to make it a little more user friendly.

Link: https://www.youtube.com/watch?v=6SPDvPK38tw (This video helped me most with error handling).



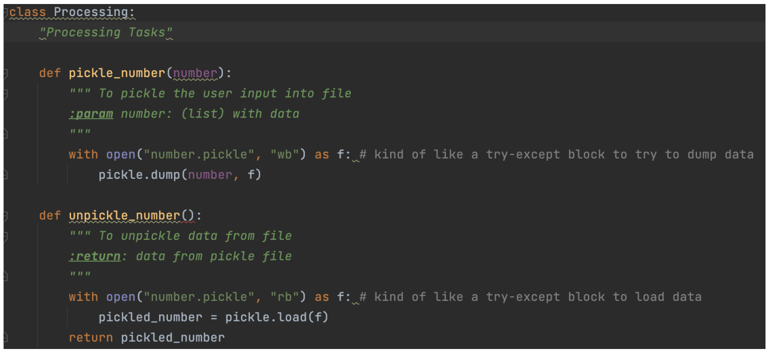
```
except FileNotFoundError as e:
    # If the file doesn't exist, print an error message
    print("Error: file not found.")
    print(e, e.__doc__, type(e), sep="\n")
```

Figure 2: Code I used to catch a specific exception

Steps for my script

My first step for my script this week was to figure out what I wanted my script to do. I kind of used up alot of my time this week reading and watching videos on pickling and exception handling so I decided to do something simple so I wouldn't end up turning in my assignment late. I came up with a script that would ask a user for their favorite number then it would pickle that number to a file and unpickle it back to them. This way the user could see that it was pickled when they opened the file. I also had a few exception handling stuff in there like catching specific exceptions and try - except blocks.

I first started with making two functions, 1 to pickle the data and the other to unpickle it to the file. I tried this “with open” kind of statement(?) that I saw a lot of when I was doing my research on pickling.



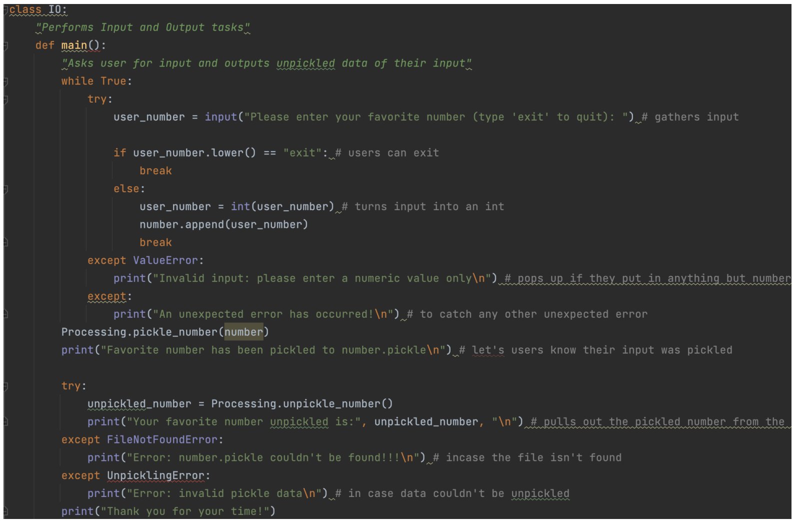
```
class Processing:
    "Processing Tasks"

    def pickle_number(number):
        """ To pickle the user input into file
        :param number: (list) with data
        """
        with open("number.pickle", "wb") as f: # kind of like a try-except block to try to dump data
            pickle.dump(number, f)

    def unpickle_number():
        """ To unpickle data from file
        :return: data from pickle file
        """
        with open("number.pickle", "rb") as f: # kind of like a try-except block to load data
            pickled_number = pickle.load(f)
        return pickled_number
```

Figure 3: My pickling and unpickling functions of my script

Then my last function is just a giant function that will ask the user for input and then checks to see if the input meets the requirements of what I want it to be and will pickle it to the file and unpickle back thus creating the file and letting users know it was pickled. It also contains some exception handling to catch some specific errors.

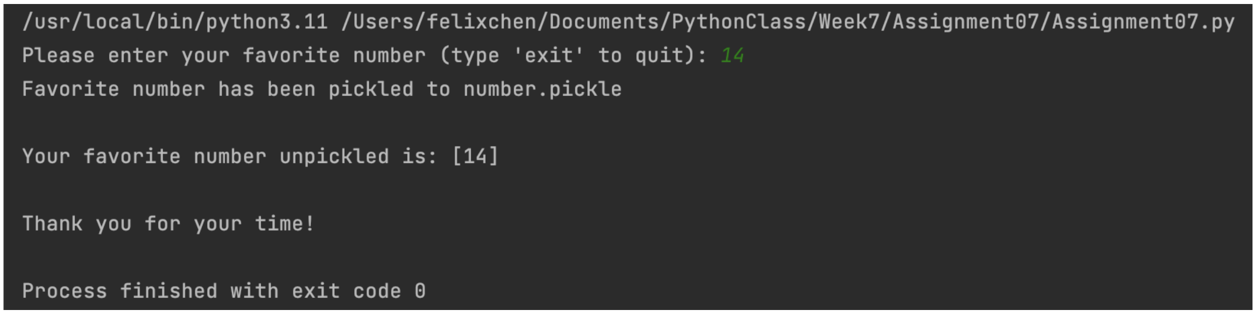


```
class IO:
    "Performs Input and Output tasks"
    def main():
        "Asks user for input and outputs unpickled data of their input"
        while True:
            try:
                user_number = input("Please enter your favorite number (type 'exit' to quit): ") # gathers input

                if user_number.lower() == "exit": # users can exit
                    break
                else:
                    user_number = int(user_number) # turns input into an int
                    number.append(user_number)
                    break
            except ValueError:
                print("Invalid input: please enter a numeric value only\n") # pops up if they put in anything but number
            except:
                print("An unexpected error has occurred!\n") # to catch any other unexpected error
        Processing.pickle_number(number)
        print("Favorite number has been pickled to number.pickle\n") # let's users know their input was pickled

        try:
            unpickled_number = Processing.unpickle_number()
            print("Your favorite number unpickled is:", unpickled_number, "\n") # pulls out the pickled number from the
        except FileNotFoundError:
            print("Error: number.pickle couldn't be found!!!\n") # incase the file isn't found
        except UnpicklingError:
            print("Error: invalid pickle data\n") # in case data couldn't be unpickled
        print("Thank you for your time!")
```

Figure 4: My main function of the script



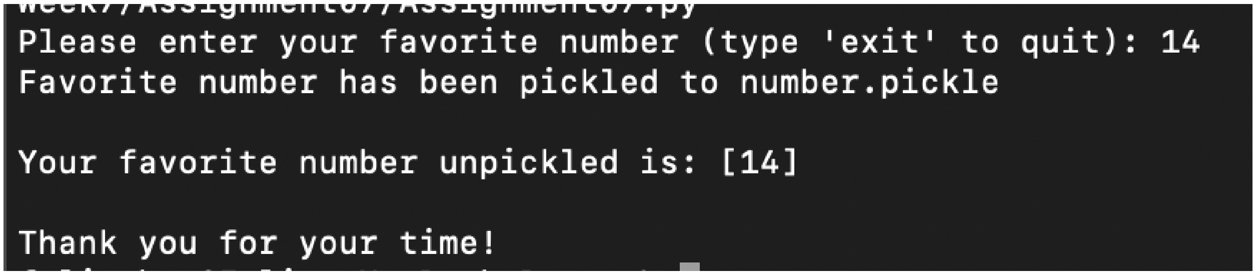
```
/usr/local/bin/python3.11 /Users/felixchen/Documents/PythonClass/Week7/Assignment07/Assignment07.py
Please enter your favorite number (type 'exit' to quit): 14
Favorite number has been pickled to number.pickle

Your favorite number unpickled is: [14]

Thank you for your time!

Process finished with exit code 0
```

Figure 5: My script working in PyCharm

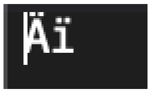


```
Week7/Assignment07/Assignment07.py
Please enter your favorite number (type 'exit' to quit): 14
Favorite number has been pickled to number.pickle

Your favorite number unpickled is: [14]

Thank you for your time!
```

Figure 6: My script working in Terminal



```
Äï
```

Figure 7: Result of my script

Summary

I enjoyed this week's lecture and how we had to also do our own research on a few topics. I felt like this week's lecture combined with researching on our own has helped me understand pickling and error handling better. I do regret using a lot of my time this week on researching the topics so I didn’t save much time for the assignment. That is one thing I would do better next time so I could come up with something a little more complex than the script I had written for this week’s assignment.