Name: Rohini Dattatray Nawale

Date:01/06/2023

Course Foundations of Programming: Python

Assignment 07

Creating A Python Script

Introduction

In Python, pickling is the process of converting an object into a byte stream. This byte stream can then be stored on disk or transmitted over a network. When the byte stream is later read, it can be used to recreate the original object.

Pickle is a powerful tool that can be used to save the state of a program, or to transmit data between different programs. However, it is important to note that pickling is not secure. Any object that can be pickled can also be unpickled, which means that an attacker could potentially inject malicious code into your program.

Benefits of Pickling

- Here are some of the benefits of using pickling:
- It can be used to save the state of a Python program, so that it can be resumed later.
- It can be used to transmit data between different Python programs.
- It is a relatively efficient way to store data.

Writing My Script and Observation

How Do You Pickle?

It’s easy to pickle, especially if you are familiar with reading and writing strings to plain text files. First, you need to open an object file with the open() function, but you have to open the object file with access mode **‘wb’**. The b stands for binary

Next, you write a Python object to the binary file using the pickle.dump() function. This step is the serialization process. In my example, I assigned a list of dictionary elements to the variable debutAlbums. The list object is then written to the binary object file pickling

How do you Unpickle?

Unpickling is just as easy as pickling. First, the pickle module is imported. Second, a binary object file is opened with the open() function, but this time with access mode ‘rb’. Finally, the pickle.load() function is used to unpickle the file into a variable i.e. de-serialize the data in the file into an object.

Exception Handling

Exception handling is a way to handle errors that arise during the execution of code. Errors can result because of bad user inputs, or missing files, or bad programming. There are many reasons that an error might occur. If an error occurs in a script that does not have exception handling, the script will exit with an error and print the error to the screen that is not usually understood by the user. Exception handling is a way to capture these errors, provide useful messages to the user, and allow the program to continue running. In Python, we use the try statement to handle exceptions.

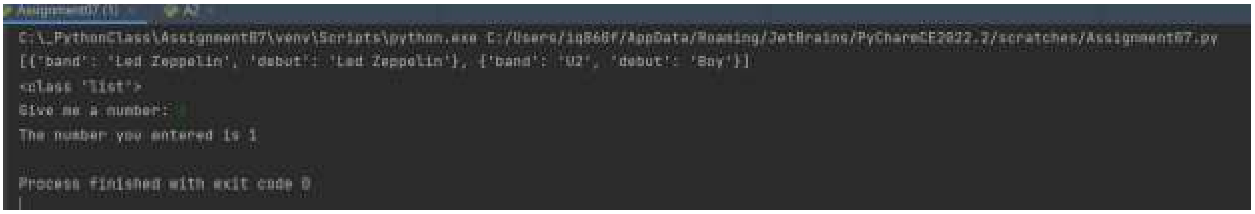
Try Statement

A try statement consists of a try clause and an except clause. And there can be multiple except clauses. The try clause begins with the keyword "try" followed by a colon, followed by an indented block of code. It is in this indented block that you place code to be "checked" for errors. The code in the block will try to run. If there are no exceptions the code will execute as intended. If an error is encountered the script will jump to the except clause.

Except Clause

The except clause begins with the "except" keyword, followed by named exceptions, followed by a colon, followed by an indented block of code. The script will check if the named exception or exceptions of the except clause matches the exception from the try clause. If so, the block statement of the except clause will execute. This behavior is similar to if-else statements in that only the first else statement that evaluates to True executes; the others will be skipped even if they would evaluate to True. Similarly, In the try statement, only the first except clause to have a match executes. If there are no matching except clauses the script will exit with an error just as it would if there had been no try statement.

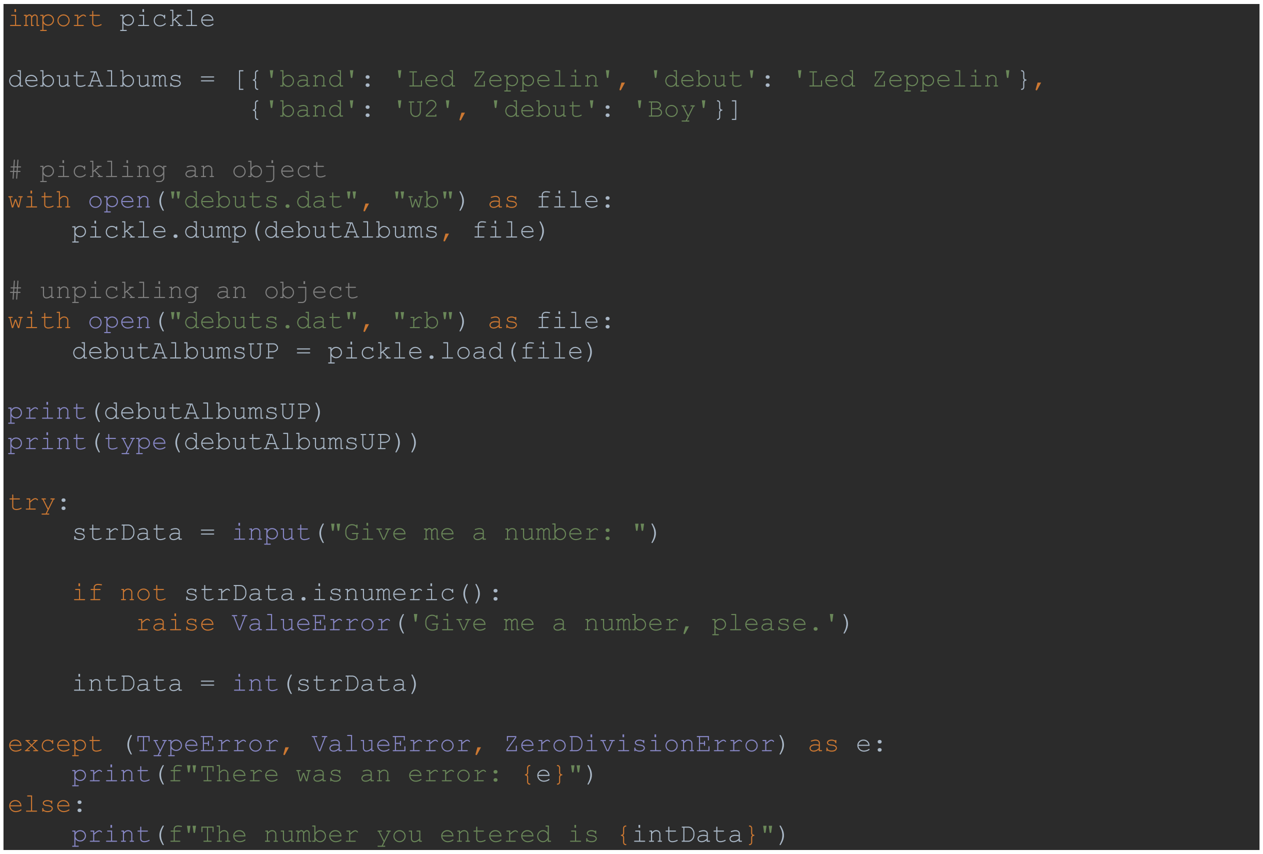
Fig01-output of code



```
C:\_PythonClass\Assignment07\venv\Scripts\python.exe C:/Users/1q868f/AppData/Roaming/JetBrains/PyCharmCE2022.2/scratches/Assignment07.py
[{'band': 'Led Zeppelin', 'debut': 'Led Zeppelin'}, {'band': 'U2', 'debut': 'Boy'}]
<class 'list'>
Give me a number: 
The number you entered is 1

Process finished with exit code 0
```

Fig02- Details of code



```
import pickle

debutAlbums = [{'band': 'Led Zeppelin', 'debut': 'Led Zeppelin'},
               {'band': 'U2', 'debut': 'Boy'}]

# pickling an object
with open("debuts.dat", "wb") as file:
    pickle.dump(debutAlbums, file)

# unpickling an object
with open("debuts.dat", "rb") as file:
    debutAlbumsUP = pickle.load(file)

print(debutAlbumsUP)
print(type(debutAlbumsUP))

try:
    strData = input("Give me a number: ")

    if not strData.isnumeric():
        raise ValueError('Give me a number, please.')

    intData = int(strData)

except (TypeError, ValueError, ZeroDivisionError) as e:
    print(f"There was an error: {e}")
else:
    print(f"The number you entered is {intData}")
```

- The code first imports the pickle module. The pickle module is a Python library that allows you to serialize and deserialize Python objects.
- The code then creates a list of dictionaries, where each dictionary represents a debut album. The dictionaries contain the following keys:
- band: The name of the band
- debut: The name of the debut album
- The code then uses the pickle module to serialize the list of dictionaries to a file called debuts.dat. This means that the list of dictionaries is converted into a sequence of bytes that can be stored on disk.
- The code then opens the file debuts.dat in read-binary mode.
- The code then uses the pickle module to deserialize the list of dictionaries from the file debuts.dat. This means that the sequence of bytes is converted back into the original list of dictionaries.
- The code then prints the list of dictionaries.
- The code then prints the type of the list of dictionaries.
- The code then prompts the user to enter a number.
- The code then checks if the user input is a number. If it is not a number, the code raises a ValueError exception.
- If the user input is a number, the code converts it to an integer.

- The code then checks if there were any errors during the execution of the code. If there were any errors, the code prints the errors.
- If there were no errors, the code prints the number that the user entered.

Summary:

In summary, the code creates a list of debut albums, serializes the list to a file, deserializes the list from the file, prints the list, and prompts the user for a number. The code then checks if the user input is a number, converts it to an integer, and prints the number.

Note** C:\Users\iq860f\ -**iq860** is my system ID. Thanks.