Janelle Bernales

May 31, 2023

Intro to Programming (Python)

Assignment 07

Intro to Programming: Using Pickles and Try/Except Error Handling Exceptions

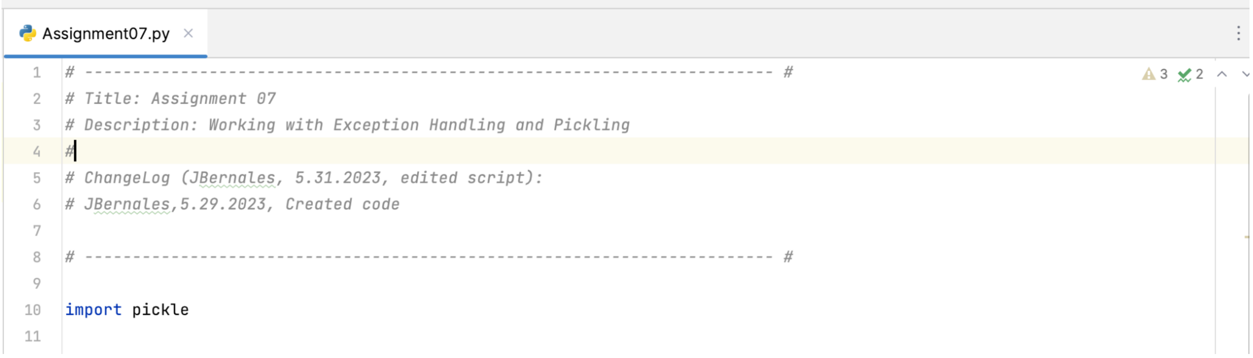
Introduction

In Module 07 I learned how to make scripts more user-friendly by displaying custom error handling messages and to use “pickles” to retrieve data from a binary file and use it within your script. Using “Try/Else” exceptions are helpful for catching any errors caused by unexpected/incompatible user input and displaying an error message to help the user change their input to the correct format- in this case, entering only letters for the user’s name, not numbers or special characters.

Creating the Program

I created a new script for Module 07 titled “Assignment07.py” within the “Assignment07” subfolder in my “_PythonClass” folder.

I added my header notes as usual. While I would normally include a section here defining the variables used in the script, I didn’t add one this time because each variable is used only once in this short, simple program:

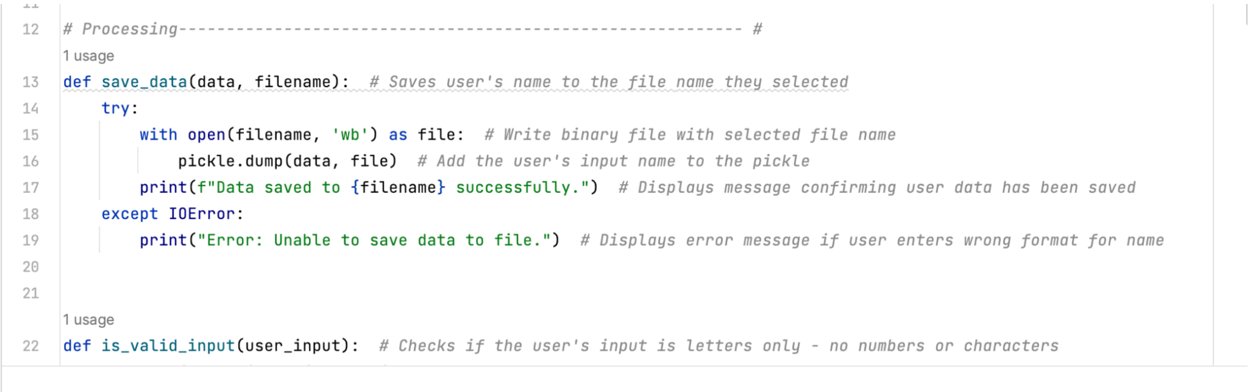


```
# ------------------------------------------------------------------------- #
# Title: Assignment 07
# Description: Working with Exception Handling and Pickling
#
# ChangeLog (JBernales, 5.31.2023, edited script):
# JBernales,5.29.2023, Created code

# ------------------------------------------------------------------------- #

import pickle
```

Fig. 1: The script header notes followed by importing the pickle used to save the user's name input

Define functions used to save data

Next, I added a defined function ("save_data") to write and save the user's input to a "pickle." In this module I learned a pickle is a Python function that allows you to convert objects into a binary format that can be saved to a file. This data can later be loaded and reconstructed in the script by importing the pickle:



```
# Processing----------------------------------------------------------- #
def save_data(data, filename):  # Saves user's name to the file name they selected
    try:
        with open(filename, 'wb') as file:  # Write binary file with selected file name
            pickle.dump(data, file)  # Add the user's input name to the pickle
        print(f"Data saved to {filename} successfully.")  # Displays message confirming user data has been saved
    except IOError:
        print("Error: Unable to save data to file.")  # Displays error message if user enters wrong format for name


def is_valid_input(user_input):  # Checks if the user's input is letters only - no numbers or characters
```

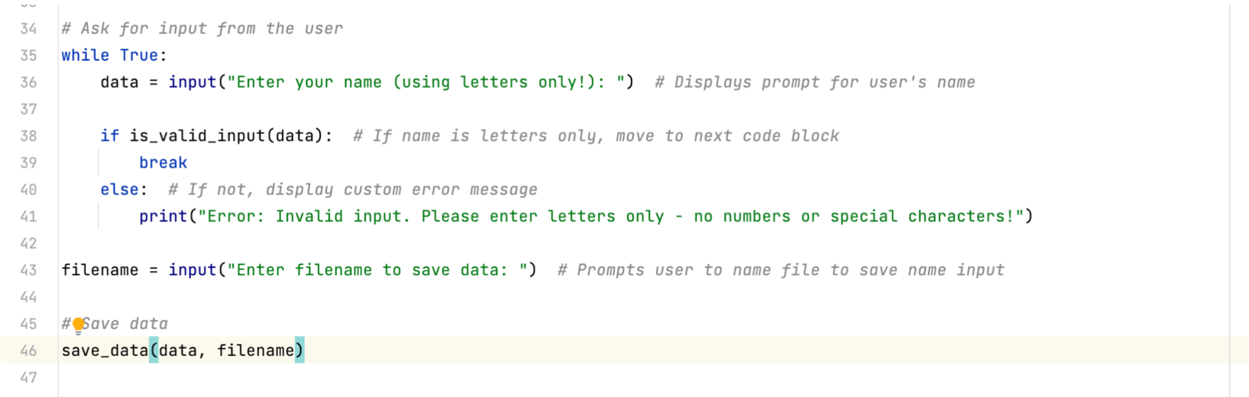
Fig. 2: Pickling is an efficient way to store and share specific databases, but it's also not a secure encrypted method of sharing data, so use with caution.[1]

I also defined a function, "is_valid_input(user_input)", that is used to check and make sure that the user's input name consists of a string only, with no numbers or special characters. If the user tries to enter a response in incompatible format, the program will produce a custom text error telling the user that the program can only accept letters as input.

Using "try / except" error exception handling

[1] Dawson, Michael. *Python Programming for the Absolute Beginner: Michael Dawson*. Course Technology Cengage Learning, 2010.

In the main body of the script, I included a very simple prompt asking the user to enter their name and a file name under which to save their name:



```
# Ask for input from the user
while True:
    data = input("Enter your name (using letters only!): ")  # Displays prompt for user's name

    if is_valid_input(data):  # If name is letters only, move to next code block
        break
    else:  # If not, display custom error message
        print("Error: Invalid input. Please enter letters only - no numbers or special characters!")

filename = input("Enter filename to save data: ")  # Prompts user to name file to save name input

# Save data
save_data(data, filename)
```

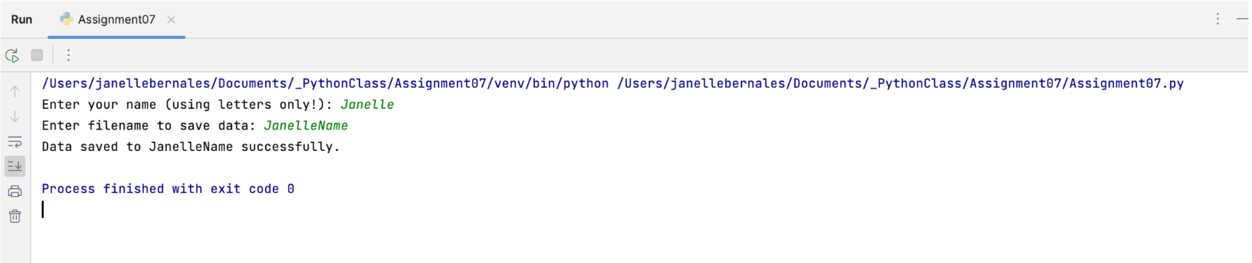
Fig. 3: The script prompts a user to enter their name. It will save the data to a file that the user names, unless the user enters special characters or numbers which will trigger an error handling message.

This code block also checks to make sure that the user only enters letters for their name. The "if / else" condition in this condition accepts only letters as input and will display an error message otherwise.

As long as the user enters input in a string format, the program will allow the user to save their data to a binary file.

Running the script

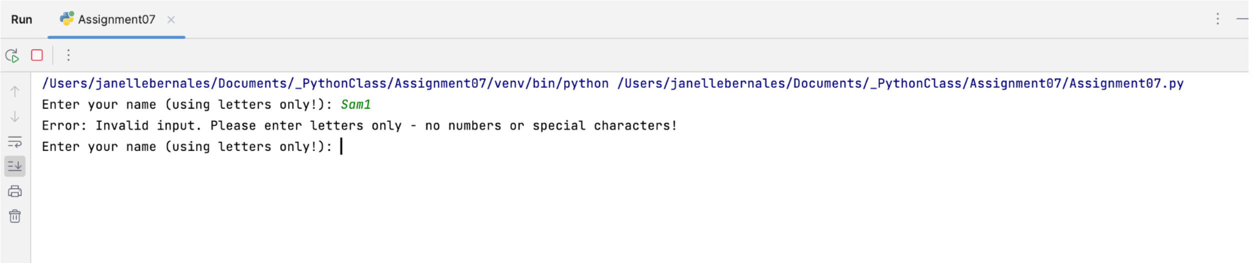
With the pickle storing the user's data imported, the functions defined, and the conditions for the "try / except" handling error defined, it's time to test out the script. This is how the program runs in Pycharm if the user enters their input using letters only:



```
/Users/janellebernales/Documents/_PythonClass/Assignment07/venv/bin/python /Users/janellebernales/Documents/_PythonClass/Assignment07/Assignment07.py
Enter your name (using letters only!): Janelle
Enter filename to save data: JanelleName
Data saved to JanelleName successfully.

Process finished with exit code 0
```

Fig. 4: The program successfully runs and saves the user's input to the binary file written by the pickle as long as the user's input is a string.

However, if the user attempts to enter any numbers or special characters as input, it will trigger an error message and will require the user to try again before saving to a file:



```
/Users/janellebernales/Documents/_PythonClass/Assignment07/venv/bin/python /Users/janellebernales/Documents/_PythonClass/Assignment07/Assignment07.py
Enter your name (using letters only!): Sam1
Error: Invalid input. Please enter letters only - no numbers or special characters!
Enter your name (using letters only!): 
```

Fig. 6: The error message triggered when the user attempts to enter input that includes numbers or special characters.

Summary

In Module 07 I learned how to use error handling exceptions to make your code more user friendly and help ensure that the program works correctly when requiring users to enter input. Using “try / except” error handling allows the programmer to create a custom error message to be displayed, rather than the technical code errors that Python displays by default.

Pickles, or the ability to write and store data to a binary file, also provide an easy way to store data in a numerical file that can be consistently retrieved from computer to computer and server to server. While I’m still working on getting a better understanding of how to use pickles in code, having this tool in my toolkit seems like a helpful function to store info that other programmers can easily retrieve by importing the pickle.