Luyao Xu

Instructor Randal Root

IT Foundation 100

August 27 2018

Assignment 07

In this assignment, I am going to create a simple example to show how I use Python exception handling and Python Pickling, and write the project to a blog. I create a furniture sale list in my example as following:

1. First, I introduced my task by using pseudo code.(Fig.1):

# -------------------------------------------------#  
# Title: Working with errors and pickling  
# Dev: Luyao Xu  
# Date: 08/28/2018  
# Desc: Furniture sale list  
# ChangeLog:Luyao, 08/28/2018, Create a simple example of Python Exception handling and pickling.  
# -------------------------------------------------#  
*'''  
1) Create a simple example of how you would use Python Exception Handling. Make sure to comment your code.  
  
2) Create a simple example of how you would use Python Pickling. Make sure to comment your code.  
  
'''*

Fig.1 outline task

2. I declare my variables in the data section. (Fig.2):

# --Data--#  
  
# declare variables and constants  
# f = An object that represents a file  
# name\_input= Input of the name of furniture  
# price\_input= Input of a price the furniture

Fig.2 Variable identification

3. I am entering my processing section. First, I import pickle in order to save data by using picking in my future steps, and then I create a global list to save my furniture. (Fig3)

# --Processing--#  
  
import pickle  
  
FURNITURE = [] # create a global list to save furniture

Fig3. Import pickle and create variable

4. Define my first function “add\_furniture” to capture user input and add the input to the global list. When user enter exit, the program quite. (Fig.4)

# define the method  
def add\_furniture():  
 print("Please input a furniture you want to sale, and list its price.") # capture user input  
 print("Enter 'Exit' to quite.") # Program exit when user input 'exit'.  
 while True:  
 name\_input = input("Please enter a furniture you want to sell: ")  
 if name\_input.lower() == 'exit': # if user input'exit', program exit  
 break  
 else:  
 price\_input = float(input("Please enter the price of the furniture: ")) # possible point of exception  
 FURNITURE.append([name\_input, price\_input]) # append to global list

Fig4. Add data into global list and exit program

5. Define my second function “dump\_data” to save the data into a binary file by using pickling (Fig5.):

# Save data to the file using pickle  
def dump\_data(furniture, filename):  
 f = open(filename, 'wb') # Create and write into a binary file  
 pickle.dump(furniture, f) # dump the text contents to f  
 f.close() # File close

Fig5. Save data to binary file

6. Define my third function “load\_data” to read data from the binary file by using pickling (Fig6):

# Read data with pickle from a binary file  
def load\_data(filename):  
 f = open(filename, 'rb') # open and read text from the binary file  
 data = pickle.load(f) # Load data from f  
 f.close() # File close  
 return data

Fig6. Read data from binary file

7. I present my code and use Python exception code for error handling. (Fig7.)

# Error handling  
#--Presentation--#  
try:  
 add\_furniture()  
 dump\_data(FURNITURE, 'furniture.dat')  
 file\_content = load\_data('furniture.dat')  
  
 # format and print the items in the file  
 print('Here is a summary of what you want to sell:')  
 for i in file\_content:  
 print(i[0] + ' for $' + str(i[1]))  
  
  
except Exception as e: # capture exception and print the error  
 print('An error has occurred:')  
 print(e)

Fig7. Presentation and error handling

Then I test my code in Python(Fig8):

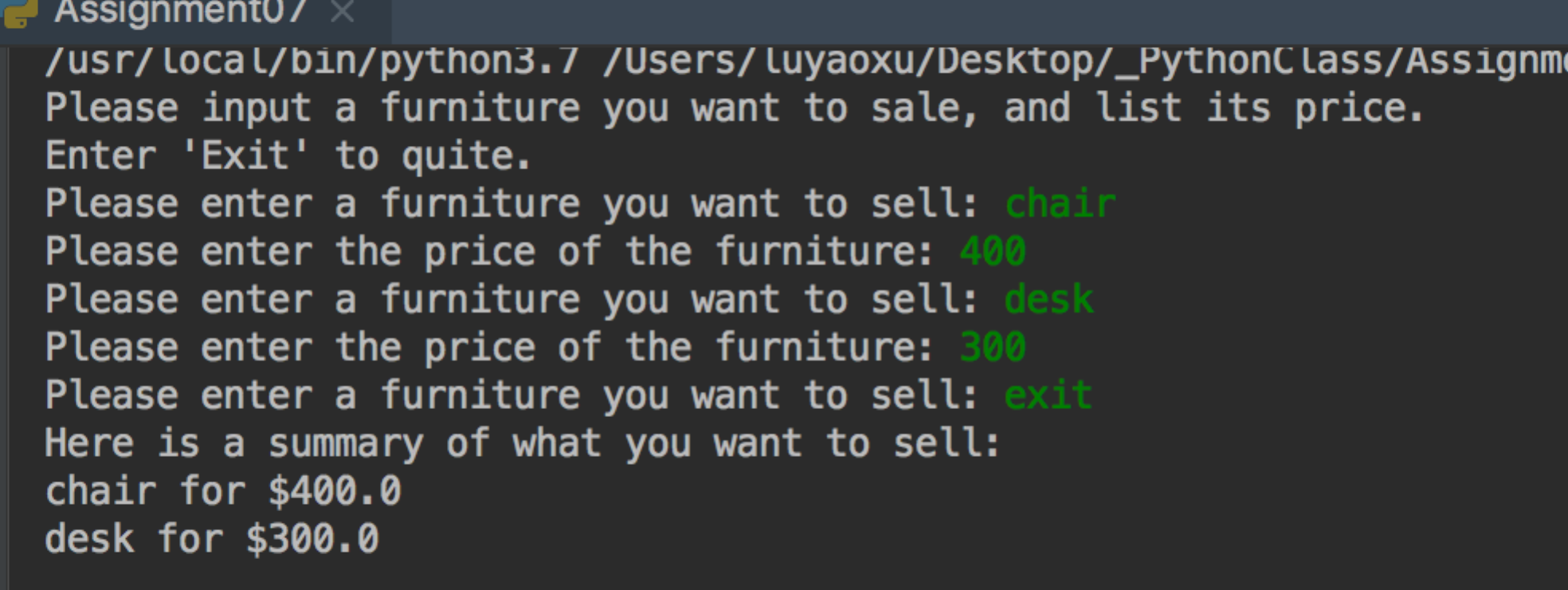


Fig8. Code test

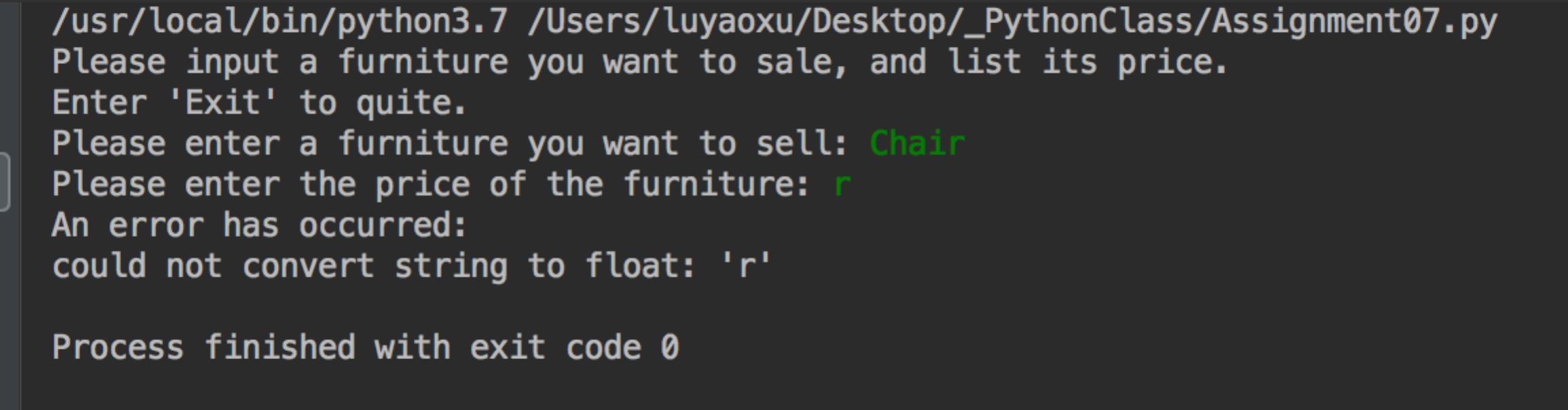
Test error handling code (Fig9): 

Fig9. Error handling code test

My entire code showing as below(Fig 10):

# -------------------------------------------------#  
# Title: Working with errors and pickling  
# Dev: Luyao Xu  
# Date: 08/28/2018  
# Desc: Furniture sale list  
# ChangeLog:Luyao, 08/28/2018, Create a simple example of Python Exception handling and pickling.  
# -------------------------------------------------#  
*'''  
1) Create a simple example of how you would use Python Exception Handling. Make sure to comment your code.  
  
2) Create a simple example of how you would use Python Pickling. Make sure to comment your code.  
  
'''*# --Data--#  
  
# declare variables and constants  
# f = An object that represents a file  
# name\_input= Input of the name of furniture  
# price\_input= Input of a price the furniture  
  
# --Processing--#  
  
import pickle  
  
FURNITURE = [] # create a global list to save furniture  
  
# define the method  
def add\_furniture():  
 print("Please input a furniture you want to sale, and list its price.") # capture user input  
 print("Enter 'Exit' to quite.") # Program exit when user input 'exit'.  
 while True:  
 name\_input = input("Please enter a furniture you want to sell: ")  
 if name\_input.lower() == 'exit': # if user input'exit', program exit  
 break  
 else:  
 price\_input = float(input("Please enter the price of the furniture: ")) # possible point of exception  
 FURNITURE.append([name\_input, price\_input]) # append to global list  
  
  
# Save data to the file using pickle  
def dump\_data(furniture, filename):  
 f = open(filename, 'wb') # Create and write into a binary file  
 pickle.dump(furniture, f) # dump the text contents to f  
 f.close() # File close  
  
  
# Read data with pickle from a binary file  
def load\_data(filename):  
 f = open(filename, 'rb') # open and read text from the binary file  
 data = pickle.load(f) # Load data from f  
 f.close() # File close  
 return data  
  
  
# Error handling  
#--Presentation--#  
try:  
 add\_furniture()  
 dump\_data(FURNITURE, 'furniture.dat')  
 file\_content = load\_data('furniture.dat')  
  
 # format and print the items in the file  
 print('Here is a summary of what you want to sell:')  
 for i in file\_content:  
 print(i[0] + ' for $' + str(i[1]))  
  
  
except Exception as e: # capture exception and print the error  
 print('An error has occurred:')  
 print(e)

Fig 10. Entire code