Kate Beverly

August 25, 2022

Foundations of Programming: Python

Assignment 07

https://github.com/kbev12/IntroToProg-Python-Mod07

To Do Script

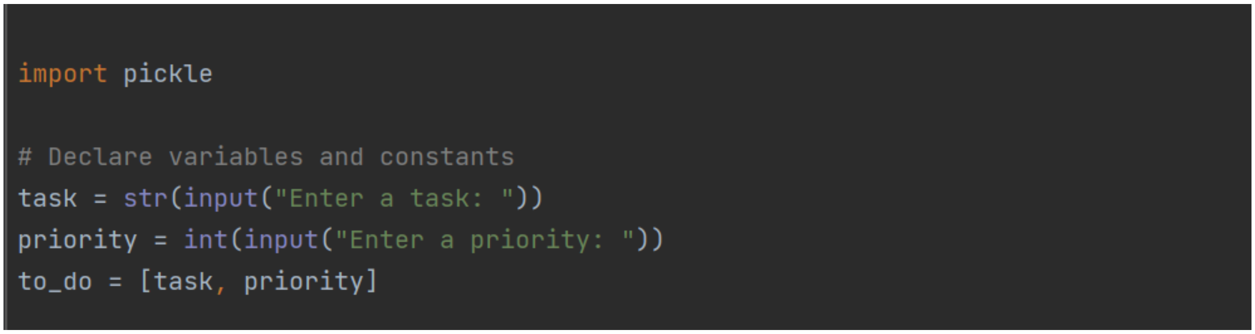
Introduction

For assignment 07 for the class Foundations of Programming: Python I worked on a python script that pickles a file, loads and prints the unpickled file, and ends the program.

Creating the Script

Pickling

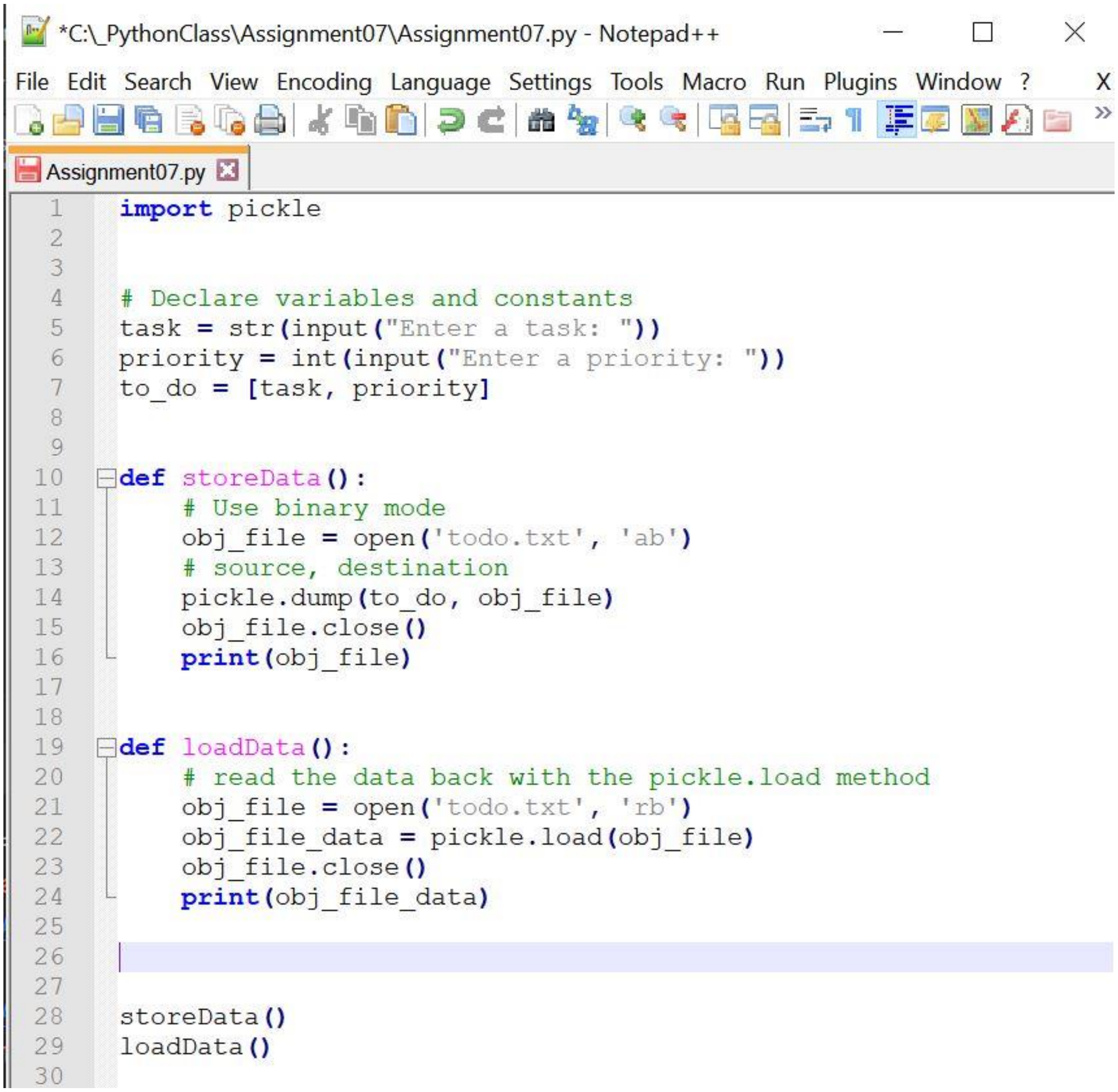
The script begins with importing the pickle module and declaring the variables (**Figure 1**).



```python
import pickle

# Declare variables and constants
task = str(input("Enter a task: "))
priority = int(input("Enter a priority: "))
to_do = [task, priority]
```

Figure 1. Importing pickle module and declaring the program variables

I started creating the script (Figure 2) by taking a string input for a task, int input for the priority, and assigning them to a dictionary. I created a function to append to ‘to_do.txt in binary. Using pickle.dump the to_do object is serialized to obj_file. The function loadData unpickles todo.txt by opening the file in read binary and loading the file to the variable obj_file_data.

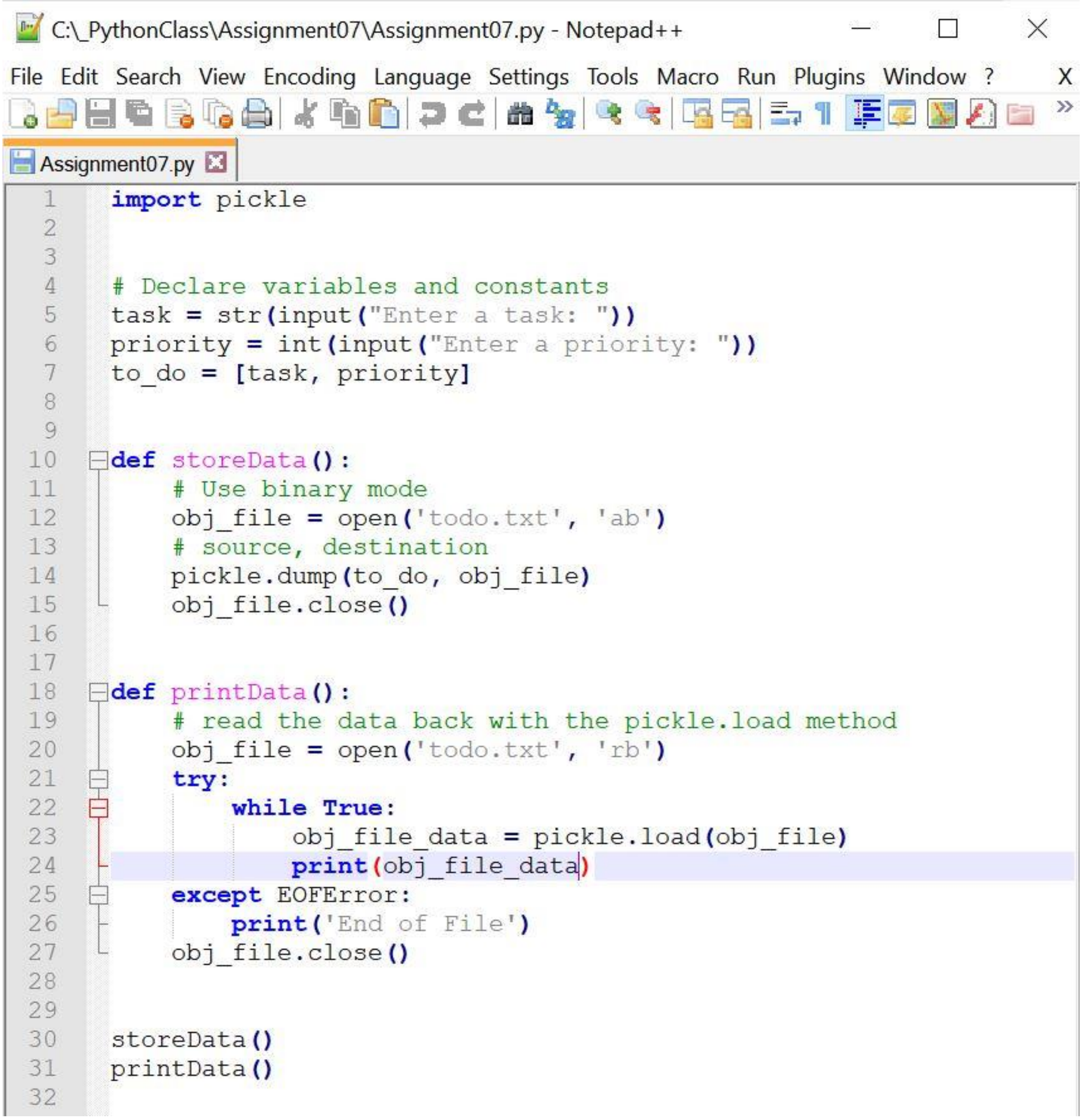

*C:_PythonClass\Assignment07\Assignment07.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
X
Assignment07.py

```
import pickle


# Declare variables and constants
task = str(input("Enter a task: "))
priority = int(input("Enter a priority: "))
to_do = [task, priority]


def storeData():
    # Use binary mode
    obj_file = open('todo.txt', 'ab')
    # source, destination
    pickle.dump(to_do, obj_file)
    obj_file.close()
    print(obj_file)


def loadData():
    # read the data back with the pickle.load method
    obj_file = open('todo.txt', 'rb')
    obj_file_data = pickle.load(obj_file)
    obj_file.close()
    print(obj_file_data)



storeData()
loadData()
```

Figure 2. Pickling and unpickling todo.txt

Exceptions

To enhance the script I updated the loadData function to printData function to add in a try/except (Figure 3). The function still opens todo.txt in read binary but now it loads each line and prints it to the terminal. Except once it receives an End of File error it prints ‘End of File’ and then closes the file.

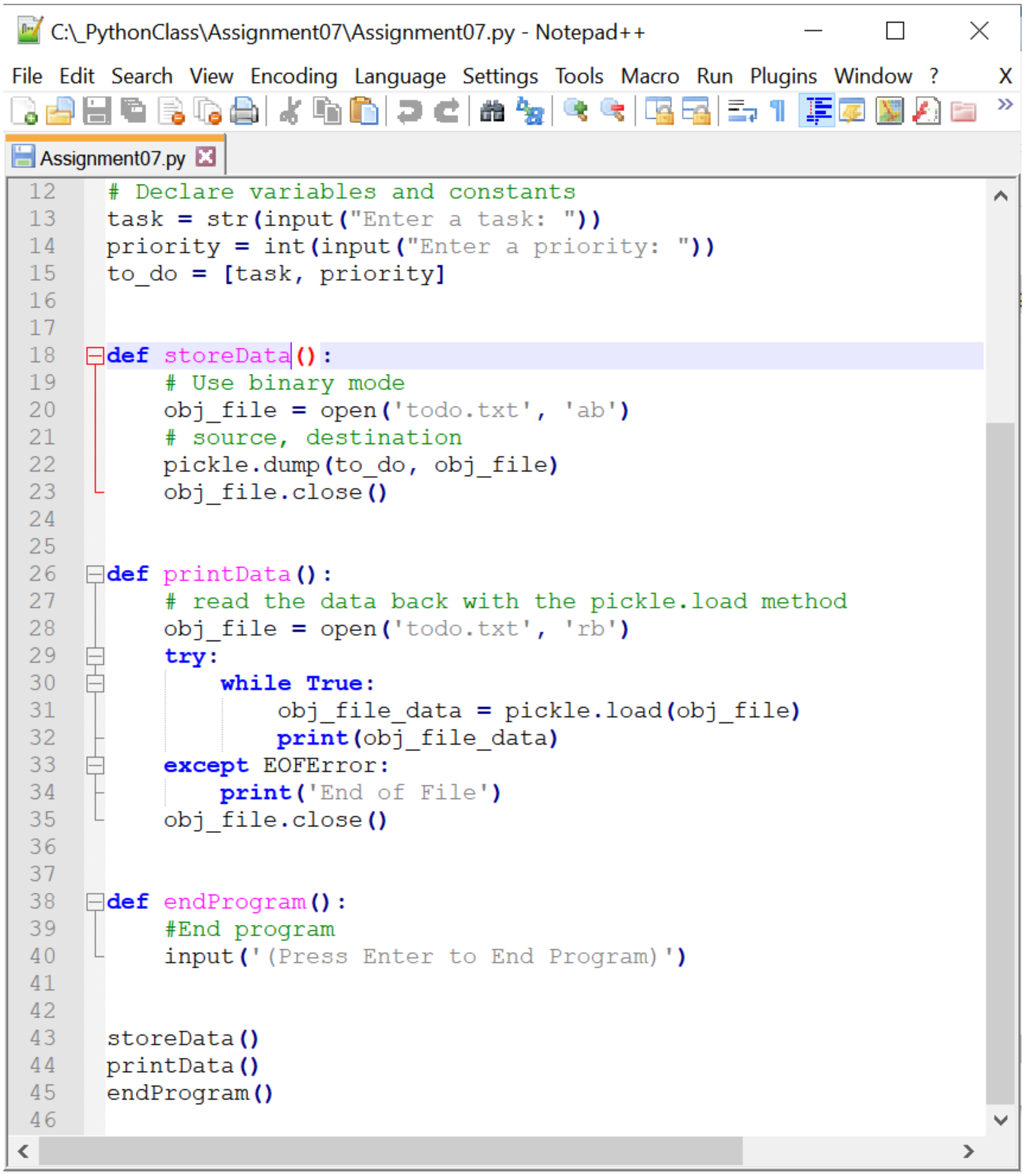


C:_PythonClass\Assignment07\Assignment07.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Assignment07.py

```
import pickle


# Declare variables and constants
task = str(input("Enter a task: "))
priority = int(input("Enter a priority: "))
to_do = [task, priority]


def storeData():
    # Use binary mode
    obj_file = open('todo.txt', 'ab')
    # source, destination
    pickle.dump(to_do, obj_file)
    obj_file.close()


def printData():
    # read the data back with the pickle.load method
    obj_file = open('todo.txt', 'rb')
    try:
        while True:
            obj_file_data = pickle.load(obj_file)
            print(obj_file_data)
    except EOFError:
        print('End of File')
    obj_file.close()


storeData()
printData()

```

End of Script

The script ends by asking the user to press enter to end the program. (Figure 4) Each of the functions are called in order.

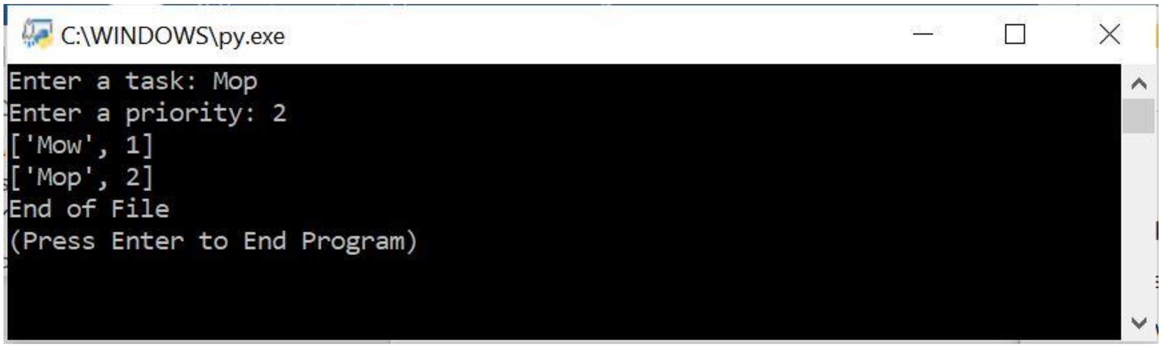

C:_PythonClass\Assignment07\Assignment07.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
X
Assignment07.py

```
# Declare variables and constants
task = str(input("Enter a task: "))
priority = int(input("Enter a priority: "))
to_do = [task, priority]


def storeData():
    # Use binary mode
    obj_file = open('todo.txt', 'ab')
    # source, destination
    pickle.dump(to_do, obj_file)
    obj_file.close()


def printData():
    # read the data back with the pickle.load method
    obj_file = open('todo.txt', 'rb')
    try:
        while True:
            obj_file_data = pickle.load(obj_file)
            print(obj_file_data)
    except EOFError:
        print('End of File')
    obj_file.close()


def endProgram():
    #End program
    input('(Press Enter to End Program)')


storeData()
printData()
endProgram()

```

Figure 4 Program ends with user's input

Testing the Script

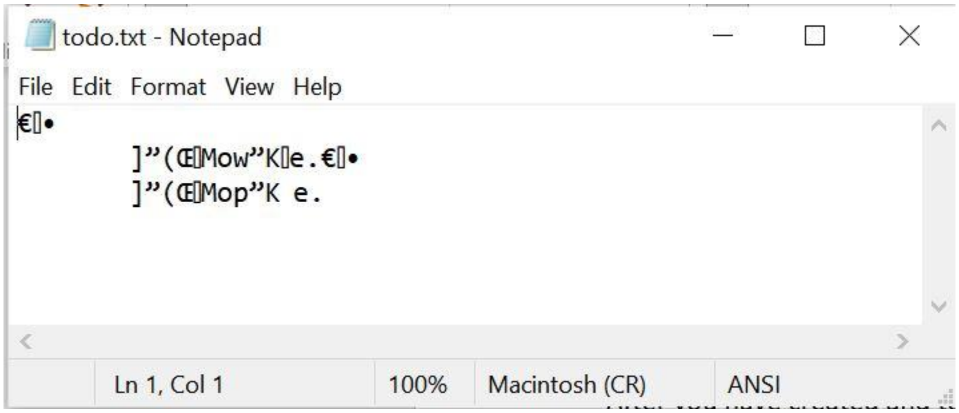
I tested the program in both PyCharm and in the command prompt and it successfully completed using both (**Figure 5**).



```
Enter a task: Mop
Enter a priority: 2
['Mow', 1]
['Mop', 2]
End of File
(Press Enter to End Program)
```

Figure 5. Testing the program

The script successfully ran and updated the text file todo.txt (**Figure 6**).



```
€•
        ]”(ŒMow”Ke.€•
        ]”(ŒMop”K e.
```

Figure 6. Updated todo.txt

Summary

For Foundations of Programming: Python seventh assignment I worked on a python script that pickles a file, loads and prints the unpickled file, and ends the program.