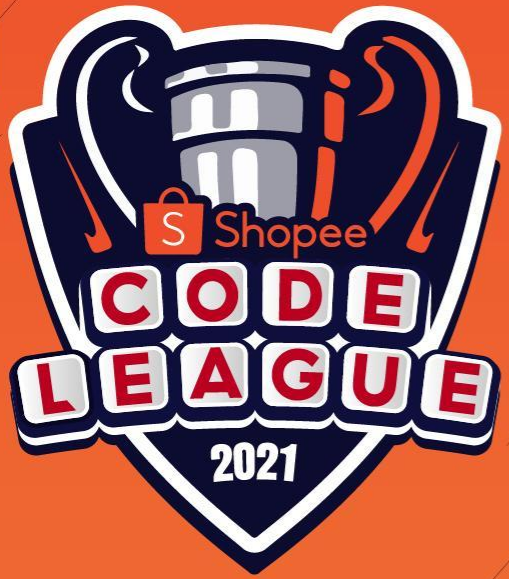

OOP With Python

Unit 01 - Files

{<oding:lab}

{<oding:lab}

Saving and Loading Data with Files

Persistent Data Storage

Why do we use files?

- Allow persistent storage of data
- Data continue to exist after program is stopped or computer turned off
- Data easily transferred between computers

Opening a File

- Open a file with the open() function

```
1    dataFile = open("myDataFile.txt","w")
```

Permission argument that specifies what the program is allowed to do with the file:

File Directory / Name

Argument	Associated Actions
r	Read
w	Write only (If file does not exist, program will create a one)
r+	Read and Write (Error if file does not exist)
a	Append (Can be used to create a new file)

Writing to and Saving a File

- .write() function writes a string to the file (2)
  - To store numbers, you need to convert them to strings (3)
- .close() function save and closes the file (5)

```
1    dataFile = open("myDataFile.txt", "w")
2    number = 134
3    dataFile.write(str(number))
4    dataFile.write("This is my data file which I am able to write to")
5    dataFile.close()
```

Reading a File

- .read() function read the file and returns content as a string (2)

```
1    dataFile = open("myDataFile.txt", "r")
2    content = dataFile.read()
3    print("Content of file is:\n" + content)
```

Reading line by line

- .readline() function read the file one line at a time

```
1    dataFile = open("myDataFile.txt", "r'")
2    fileInList=[]
3    line = dataFile.readline()
4    while line:
5        fileInList.append(line)
6        line = dataFile.readline()
7    dataFile.close()
8    print(fileInList)
```

Let's Experiment with writing to and reading from a File (Demo/Practice - 11)

- Write a program to:
  - write some text to a file
  - read from a text file
  - append to a text file

Check Point - 11

- Every student must be able to
  - write some text to a file
  - read from a text file
  - append to a text file
- For students who are waiting, try the following:
  - Update your shopping list program and have it read from and write to a file
  - Create a to-do list program

The Pickle Module

A Module to help us write and read complex data to and from files

What is Pickle Module?

- Standard file open with read and write requires everything to be in strings
- Pickle modules makes it easy to store other datatypes such as numbers (integers, floats), lists, dictionaries
- File written with pickle is not human readable
- https://docs.python.org/3/library/pickle.html

Using Pickle

- Import module => open file with “wb” => “Dump” data into file

```
1   import pickle
2
3   myShoppingList = ["Milk", "Pens", "Erasers", "Water"]
4
5   #Opening the file with "wb"
6   myFile = open("shoppingList.txt", "wb")
7
8   #Put the data into file
9   pickle.dump(myShopingList, myFile)
10  myFile.close()
```

Loading Multiple Variables/Objects

- To store multiple variables/objects, put them in a list, or tuple

```
1   #Data to store
2   myShoppingList = ["Milk", "Pens", "Erasers", "Water"]
3   myStudyList = ["Math", "Physics"]
4   myPhoneNumber = 1234567
5
6   #Open the file with "wb" attribute
7   #Give the file name a .pickle extension to identify it as pickle file
8   myFile = open("myData.pickle", "wb")
9
10  #Put the data into the file
11  pickle.dump([myShoppingList, myStudyList, myPhoneNumber], myFile)
```

Reading from a Pickle File

- Open file with “rb” attribute
- Use the .load() function

```
1    #Open the file with "rb" attribute
2    myFile = open("myData.pickle", "rb")
3
4    #Load the data to an object (variable)
5    myData = pickle.load(myFile)
```

Let's Experiment with using Pickle (Demo/Practice - 12)

- Write a program to:
  - write some data to a file with pickle
  - read from a pickle file

Check Point - 12

- Every student must be able to
  - write data to a file using pickle
  - read data from a pickle file
- For students who are waiting, try the following:
  - Update your shopping list program and have it read from and write using Pickle
  - Experiment with Pickle.
    - Another way to put in multiple data is to call the .dump() function multiple times, before closing the file. To retrieve the data, you will need to call .load() the same number of times

Additional Notes

- Alternate way to put in multiple objects/variables
  - Call .dump() multiple times (before closing the file)
  - Call .load() multiple times to retrieve
  - Experiment with it!
- Other ways to open a pickle file
  - Append
- Other modules for storage
  - shelve - https://docs.python.org/3/library/shelve.html

Handling Exceptions

Making your programs more robust with Try ... Except

Human Proofing your Programs

- Humans are not perfect. They may enter a string when they are suppose to enter a number.
- What happens if the user enters a text in the following program?

```
1    #Without try . . . except
2    birthYear = int(input("Please enter your birth year: "))
3    print("Your age is " + str(2018 - birthYear))
```

Try ... Except

- Use try... except to handle exceptions
- Concept is similar to if ... else, just that this is only to check for exceptions

```
1    #Simple use of try . . . except
2    userInput = input("Please enter your birth year: ")
3
4    #Executes 'try' portion of the code only if userInput is integer
5    try:
6        birthYear = int(userInput)
7        print("Your age is " + str(2018 - birthYear))
8    #Executes 'except' portion of the code if userInput is not integer
9    except:
10       print("You need to enter a number!")
```

Let's Experiment with Try ... Except (Demo/Practice - 13)

- Write a simple program that prints out the content in a file. If the file is not found, print out an error message.
  - Hint: Google for 'Built-in Exceptions' in Python

Check Point - 13

- Every student must be able to
  - Use try ... except
- For students who are waiting, try the following:
  - Make your earlier programs more robust

Additional Notes on Try ... Except

- This is not just for human errors
- It is used to capture and handle all kinds of exceptions that may occur
  - Failure to read a file
  - Unexpected data type
  - And many more
- https://docs.python.org/3/tutorial/errors.html

To-Do List

Create a To-Do list program

To Do List Program Specifications

- A To Do List program which allows user to create, display and delete to do items. The list will be stored on persistent storage. The next time the user run the program, the to do list will still be there.
- Make sure your program is robust (i.e. able to handle the common input selection errors)
- Teacher to demonstrate the program

Additional Challenges

- Read a text file which contains an essay and perform an analysis on the distribution of the letters
- Open a url and read a webpage
  - Check out the package urllib
  - https://docs.python.org/3/library/urllib.html
  - Use this library to open and read a webpage

Reflections

Persistent Storage

- We can use files to achieve persistent storage of data
- Two key methods to write to file:
  - Writing as text
  - Using pickle module
- There are other methods to write to file
- There are also other methods to achieve persistent storage
  - Database
- Questions to Ask yourself: Why do we need persistent storage?