Python Basics Assignment 9

1.To what does a relative path refer ?

Ans: The relative path is the path to some file with respect to your current working directory (PWD).
For example: if Absolute path to a file called stuff.txt is: D:/mihir/FSDS-Ineuron.ai/Live Class/Assignments/Python Basics/iNeuron.txt If my PWD is D:/mihir/FSDS-Ineuron.ai/Live Class/Assignments/Python Basics , then the relative path to iNeuron.txt would be: Python Basics/iNeuron.txt
Note: PWD + relative path = absolute path

2.Where does an absolute path start with your Operating System ?

Ans: In Linux based Operating systems the absolute path starts with **/** where as in Windows based Operating systems absolute path starts with **C:**

3.What does the functions os.getcwd() and os.chdir() do ?

Ans: os.getcwd() method tells us the location of current working directory (CWD) whereas os.chdir() method in Python used to change the current working directory to specified path. These functions are similar to linux commands pwd and cd

```
import os
print(os.getcwd()) # Prints the current Working Directory
path = r'D:\mihir'
os.chdir(path)
print(os.getcwd())
```

```
D:\mihir\FSDS-Ineuron.ai\Live Class\Assignments\Python Basics
D:\mihir
```

4.What are . and .. folders ?

Ans: . Represents the Current Directory Whereas .. Represents the Parent Directory of the Current Directory
For Example: if the below path is my absolute path:
D:\\mihir\\FSDS-Ineuron.ai\\Live Class\\Assignments\\Python Basics
Then . represents the path D:\\mihir\\FSDS-Ineuron.ai\\Live Class\\Assignments\\Python Basics
Where as .. represents the path D:\\mihir\\FSDS-Ineuron.ai\\Live Class\\Assignments\\Python Basics

5.In C:\bacon\eggs\spam.txt which part is the dir name and which part is the base name ?

Ans: For C:\bacon\eggs\spam.txt
The Directory name is C:\\bacon\\eggs
The Base name is spam.txt

```
import os
path = r'C:\bacon\eggs\spam.txt'
```

```
print(os.path.dirname(path))
print(os.path.basename(path))
```

```
C:\bacon\eggs
spam.txt
```

6.What are the three mode arguments that can be passed to the open() function ?

Ans: A file can be Accessed in python using open() function. Open function takes two arguments filename and mode of operation (optional). If mode is not provided, the default mode of opening is Read mode.
So, the syntax is: **open(filename, mode)**

- **'r'** – Read Mode: This is the default mode for open(). The file is opened and a pointer is positioned at the beginning of the file's content.
- **'w'** – Write Mode: Using this mode will overwrite any existing content in a file. If the given file does not exist, a new one will be created.
- **'r+'** – Read/Write Mode: Use this mode if you need to simultaneously read and write to a file.
- **'a'** – Append Mode: With this mode the user can append the data without overwriting any already existing data in the file.
- **'a+'** – Append and Read Mode: In this mode you can read and append the data without overwriting the original file.
- **'x'** – Exclusive Creating Mode: This mode is for the sole purpose of creating new files. Use this mode if you know the file to be written doesn't exist beforehand.

7.What happens if an existing file is opened in write mode ?

Ans: Using this mode, we can overwrite any existing content in a file. If the given file does not exist then a new one will be created.

8.How do you tell the difference between read() and readlines() ?

Ans: The main difference is that **read()** will read the whole file at once and then print out the first characters that take up as many bytes as we specify in the parenthesis

- whereas the **readline()** that will read and print out only the first characters that take up as many bytes as we specify in the parenthesis. We may want to use readline() when we are reading files that are too big for our RAM.
- The **read()** would treat each character in the file separately, meaning that the iteration would happen for every character.
- The **readline()** function, on the other hand, only reads a single line of the file. This means that if the first line of the file were three lines long, the readline() function would only parse (or iterate/operate) on the first line of the file.

9.What data structure does a shelf value resemble ?

Ans: It contains key and values it represents dictionary. The Shelve Module of Python is a very popular module of Python which works like an effective tool for persistent data storage inside files using a Python program. As the name of this module suggests, i.e., Shelve, we can easily interpret that it will work as a shelf object to keep all our data inside a file and save all the necessary information. In the Shelve Module, a shelf object is defined, which acts like a dictionary-type object, and it is persistently stored in the disk file of our computer. That's how we can save all the data and information through Python Shelve Module in our system and keep it till whenever we want.

```
# Importing the Shelve Module
import shelve
# Creating a shelve file with open() function
shelveVariable = shelve.open("SampleFile")
print("A Shelve file with SampleFile name is successfully created")
# Making entries inside the shelve file
shelveVariable['Employee Name'] = 'Jackie Chan'
shelveVariable['Employee Age'] = 27
shelveVariable['Employee department'] = 'Management'
shelveVariable['Employee Performance'] = 'Excellent'
shelveVariable['Employee Score'] = 9.2
shelveVariable.close()
print("All entries are successfully made inside the sample shelve file")
```

```
A Shelve file with SampleFile name is successfully created
All entries are successfully made inside the sample shelve file
```