PYTHON

LESSON 5

CODE FIRST GIRLS

NANODEGREE → FOUNDATION MODULE

AGENDA

01 File handling

02 Reading and writing files

03 Converting dictionaries into csv files

04 Problem solving with Python

PYTHON FILE HANDLING

IMPORTANT POINTS

- Imagine, we are creating a program that works with a large amount of data
- We cannot expect that data to be stored in a variable as the variables are volatile in nature.
- The data is stored externally, for example a file
- As files are non-volatile in nature, the data will be stored permanently and using python we will handle these files in our applications.

TYPES OF FILES IN PYTHON

- Binary file
- Text file

BINARY FILES IN PYTHON

IMPORTANT POINTS

- We can open some binary files in the normal text editor but we CANNOT read the content present inside the file.

- All the binary files will be ENCODED in the binary format, which can be understood only by a computer or machine.

EXAMPLE BINARY FILES

- **Document files:** .pdf, .doc, .xls etc.
- **Image files:** .png, .jpg, .gif, .bmp etc.
- **Video files:** .mp4, .3gp, .mkv, .avi etc.
- **Audio files:** .mp3, .wav, .mka, .aac etc.
- **Database files:** .sql, .accde, .frm, .sqlite etc.
- **Archive files:** .zip, .rar, .iso, .7z etc.
- **Executable files:** .exe, .dll, .class etc.

TEXT FILES IN PYTHON

IMPORTANT POINTS

- Text files do not have any specific encoding and it can be opened in normal text editor itself.

TEXT FILES IN PYTHON

- **Web standards:** html, XML, CSS, JSON etc.
- **Source code:** c, app, js, py, java etc.
- **Documents:** txt, tex, RTF etc.
- **Tabular data:** csv, tsv etc.
- **Configuration:** ini, cfg, reg etc.

FILE HANDLING OPERATIONS

DEFINITION

Most common

- open / read / write / close

Other

- rename / delete

Mode

- ‘r’ – Read Mode
- ‘w’ – Write Mode
- ‘a’ – Append Mode
- ‘r+’ – Read or Write Mode (same file)
- ‘a+’ – Append or Read Mode (same file)

for BINARY files use the same modes with the letter **‘b’** at the end.

'rb' or 'wb' and so on

EXAMPLE CREATE & OPEN FILE

```
file_object =
open(file_name, mode)
```

READING FROM A FILE

DEFINITION

- In order to read a file in python, we must open the file in read mode.

- There are three ways in which we can read the files in python.

EXAMPLE

```
read()

readline()

readlines()
```

What is the difference?

WRITE TO A FILE

DEFINITION

- we need to open or create a new file
- write a line or lines into the new file
- once finished, then we must close the file to save the changes

EXAMPLE

See DEMO(s)

CONTEXT MANAGER

PYTHON DATA TYPE

- **Context managers** facilitate the proper handling of resources.
- In the case of performing file operations a context manager would automatically open and close the file for us.
- The syntax is slightly different, it uses the with word. We can call it a with block

EXAMPLE

```
with open('people.txt', 'w+') as text_file:

    people = 'Joanne \nSusan \nAmina'

    text_file.write(people)

-------------------------------------------------------------------
--------


with open('people.txt', 'r') as text_file:
    contents = text_file.read()

print(contents)
```

WORKING WITH CSV FILES

IN PYTHON

- **csv** is a Python library that simplifies reading and writing tabular text from/to a csv file.

- The library needs to be imported first to be able to access its functions and objects.

EXAMPLE

See DEMO(s) &
Complete EXERCISE(s)

PROBLEM SOLVING

IN PYTHON

Write a Python program to count the occurrences of a word in a text file

- Your program will take a word from the user and count the number of occurrences of that word in a file.

EXAMPLE STEPS

1. Take the file name and the word to be counted from the user.
2. Read each line from the file and split the line to form a list of words.
3. Check if the word provided by the user and any of the words in the list are equal and if they are, then increment the word count.
4. Exit

CODE
FIRST
GIRLS

THANK YOU!