Ch 9: File I/O in Python

We use files to store data. Files get stored in SSD/HDD so we can use them

Two types of files: • Binary files (.jpg, .gif, .mp4)

• Text files (.txt, .py, .css)

If you open any movie in a video player, it will read the movie and display it. Video players are made specially to read such files.

On opening the same movie in Notepad, computer gets hanged/frozen because the software can't read binary files.

To open a text file, we have open() function. Generally we pass two parameters (filename, mode).

After opening the file we do operations with the file and then close it using close() function

File encoding in Windows, Linux is by default UTF-8.

Modes of opening a file:-

*** r - Opens a file only to read → default mode

*** w - Opens a file for changing by overwriting original file. New file gets created if there is no file.

*** a- Opens a file to append in it. New file gets created if file is not present

*** r+ - Opens a file for both reading and writing

There are also 'x', 't', 'b' but are not used much.

exclusive creation text mode binary mode

SAMPLE PROGRAM:

```
                        → mode
f = open ('file.txt', 'r') → opening the file
content = f.read() → Reading the file
print(content) → printing the content
f.close() → closing the file
```

f.read(5) → read first 5 characters

f.readline() → read first line of the file

f.readlines() → Returns a list of lines present in the file

□ Writing in a file

f = open('testing.txt', 'w') → mode
f.write("This is changed file") → Writing on a file
f.close()

□ Appending in a file

f = open("testing.txt", 'a') → mode
f.write("Content added at the end")
f.close()

→ This doesn't replace the content, instead it adds content to the end.

□ Some more functions

f = open("testing.txt")

→ print (F.mode) → Tells the mode in which file is opened
→ r

→ print (F.name) → Tells the file name
→ testing.txt

→ print (f.closed) → Returns true if a file is closed

→ print (f.tell()) → Tells the location of the file pointer

f.seek(3) → Change the location of file pointer to 3rd character.

□ with block

```
with open('testing.txt', 'r') as f:
    print(f.readline())
```

→ no need to close; file gets closed automatically in a with block.

• TRY THIS:-

S1. Create a customModule.py file

S2. Inside another main.py file you can import this module just as we do normally

[Try to keep both files in same directory]

Observance:- An __pycache__/ folder gets autogenerated.

* We can now use all the functions, variables, classes in our code.

Quiz →

Write a program to display the multiplication table of 8 using both while loop and for loop.