

File I/O

→ Python can be used to perform operation on a file. (read & write data)

Type of all files

1. Text files: .txt, .dock, .log etc.

2. Binary Files: .mp4, .mov, .png, .jpeg etc.

* open read & close File

→ We have to open a file before reading or writing

f = open("file_name", "mode")

Sample.txt

w

r: read mode

demo.dock

w: Write mode

data = f.read()

f.close()

Meaning

'r' = open for reading (default)

'w' = open for writing, truncating the file first

'x' = Create a new file and open it for writing

'a' = open for writing, appending to the end of the file if it exists.

'b' = binary Mode

't' = text mode (default)

'+' = open disk file for updating (reading and writing)

Reading a file

`data = f.read()` # reads entire file

`data = f.readline()` # reads one line at a time.

f.read:-

Writing to a file

`f = open("demo.txt", "w")`

`f.write ("this is a new line")` # overwrites the entire file

`f = open("demo.txt", "a")`

`f.write ("this is a new line")` # adds to the files.

with Syntax

`with open("demo.txt", "a") as f:`

`data = f.read()`

With `open(...)` as... is the modern, recommended way to open and work with file in Python

- with :- statement (which creates a context manager)

2. That file object implements the context manager Protocol:

- `--enter--()` → called when entering the with block → usually return self (the file object)
- `--exit--()` → called when exiting the block
③ closes the files.

Deleting a File

- Using the os module

module (like a code library) is a file written by another Programmer that generally has a functions we can use.

```
import os
```

```
os.remove(filename)
```