WHY NEED FILE HANDLING?

TYPES OF FILES IN PYTHON?

* What you may know as a file is slightly different in python.
  + Windows
    - Image
    - Text
    - Executable
    - Audio
  + Python
    - Binary
    - Text

WHAT IS FILE HANDLING?

* File handling as an important part of any web application.
* CRUD Operation:
  + Create
  + Read
  + Update
  + Delete

PYTHON FILE HANDLING SYSTEM

* The key function for working with files in Python is open() function.
* Create File -> Open File -> Work File -> Close File
* Open()
  + Filename
  + Mode
* Syntax: open(filename,mode)
* Filename: any name that you want.
* Mode: Different modes of opening a file.
  + “r” – Read – Default value. Opens a file for reading, error if file does not exist.
  + “a” – Append – Opens a file for appending, creates the file if it does not exist.
  + “w” – Write – Opens a file for writing, creates the file if it does not exist.
  + “x” – Create – Creates the specified file, returns an error if the file exists..
* In addition, you can specify if file should be handled as binary or text mode.
  + “t” – Text – Default value. Text Mode.
  + “b” – Binary – Binary mode (e.g. Images).
* Example Code:
  + Example:
    - f = open(“demofile.txt”)
  + Example:
    - f = open(“demofile.txt”, “r”)
  + Note: Make sure file exist or else error.

FILE OPERATIONS FOR READING

* file.read()
* Lots of ways to read a text file.
* All Characters | Some characters
* Example:
  + file = open(“testfile.txt”, “r”)
  + print file.read()
  + to read every single character include in file.
* Example:
  + file = open(“testfile.txt”, “r”)
  + print file.read(5) 🡪 This 5 indicates what?
  + 5 indicates first 5 characters of file to read and nothing else.
* Example:
  + file = open(“testfile.txt”, “r”)
  + print file.readline()
  + line by line input
* Example:
  + file = open(“testfile.txt”, “r”)
  + print file.readline(3)
  + Read third line only.
* Example:
  + file = open(“testfile.txt”, “r”)
  + print file.readlines()
  + Read lines separately

LOOPING OVER A FILE OBJECT

* Fast and efficient
* Example:
  + file = open(“testfile.txt”, “r”)
  + for line in file: -----------------------🡪 Looping over the object
  + print file.readlines()

PYTHON FILE WRITE METHOD

* Writing to an existing file
* To write to an existing file, you must add a parameter to open() function
  + “a” – Append – will append to end of file.
  + “w” – Write – will overwrite any existing content
* Example:
  + file = open(“demofile.txt”, “a”)
  + file.write(“Python”)
* Example:
  + file = open(“demofile.txt”, “w”)
  + file.write(“Python”)
* Example:
  + file = open(“demofile.txt”, “w”)
  + file.write(“This is a test”)
  + file.write(“Python”)
  + file.close()
* Note: the “w” method will overwrite the entire file.

CREATING A NEW FILE

* open() method again
* To create a file in Python, use the open() method, with one of the following parameters:
  + “x”- Create – will create a file, return as error if file exist.
  + “a”- Append – will create a file if specified file does not exist.
  + “w” – Write – will create a file if the specified file does not exist.
    - File = open(“testfile.txt”,”x”)
    - File = open(“testfile.txt”,”w”)

DELETION OPERATIONS

* os.remove() function
* to delete a file you must import os module and run it os.remove() function:
* Example:
  + Import os
  + os.remove(“demofile.txt”)
* Check if file exist:
  + Import os

If os .path.exist(“demofile.txt”):

os.remove(“demofile.txt”)

else:

print(“file does not exist”)

* Deleting a folder:
  + Import os
  + os.rmdir(“myfolder”)

CONCLUSION