OS Module

Used to interact with the operating system.

OS module documentation would be helpful.

OS is a built-in python library in python.

import os

# create a folder using mkdir

os.mkdir("Data")

import os

# now if i want to create folders from day 0 - 100 inside the Data folder

for i in range(100):

    os.mkdir(f"Data/Data {i+1}")

another program helpful example:

import os

# now if i again try to create Data folder, it will throw an error that Data already exists

# os.mkdir("Data")

# so its better to add a condition to check whether we have Data or not

try:

    os.mkdir("Data")

except FileExistsError as fe:

    print("File Already Exists.")

for i in range(100,120):

    os.mkdir(f"Data/Error {i+1}")

import os

# this will check if there is a folder called Data or not

print(os.path.exists("Data"))

this os.path.exists can also check for files also not only folders.

import os

# this will check if there is a file called requirements.txt or not

print(os.path.exists("requirements.txt"))

import os

# now in every day folder inside Data folder, i need to create main.py file

# didn't used os module, till now i dont know how to create file using os module.

for i in range(100):

    with open(f"Data/Data {i+1}/main.py","w") as f:

        pass

import os

# now lets say i wanted to rename the names of folders inside Data

for i in range(100):

    os.rename(

        f"Data/Data {i+1}",

        f"Data/Research {i+1}",

    )

import os

# list out the directories or data inside Data folder

folders = os.listdir("Data")

# looping out the folders and inside their content

for folder in folders:

    print(folder)

    print(os.listdir(f"Data/{folder}"))

    print("------")

import os

# run cls command on command prompt

os.system("cls")

# run dir command on cmd

os.system("dir")

os.remove() can remove files and directories that are not in used.

Local vs. Global Variable

x = 4

def hello():

    # i can not update the global x by doing so

    # this is the local x that im declaring and it will be destroyed after the function is being completed.

    # x = 2

    # you can modify global x by doing below

    global x

    x = 2

    print("X calling from function: ",x)

hello()

print("Outside the function: ",x)

FILE IO in Python

File handling in python is used to handle files which are text or binary.

Modes are ‘r’ for reading, ‘w’ for writing, ‘a’ for appending.

f = open("data.txt",'r')

reading the content of a file,

f = open("data.txt",'r')

# print(f) # on the output you will see some object

# how you are going to see the content of a file?

text = f.read()

print(text)

f.close()

you can’t do something like opening file in write mode and using read() function, no it will not gonna work out.

This will throw an error:

f = open("data.txt",'w')

# print(f) # on the output you will see some object

# how you are going to see the content of a file?

text = f.read()

print(text)

f.close()

modes explanation:

read( r ) mode opens the file for reading only and gives an error if file doesn’t exist. This is the default mode if no mode passed as a parameter.

Write(w) mode opens the file for writing and creates a new file if file doesn’t exist. If you write something to the file , it will remove the content of file and write new content to it.

Append(a) mode opens the file for appending only and creates a new file if the file does not exist. It will update the content inside the file , not gonna remove the contents of a file.

Create( x ) mode creates a file and gives an error if the file already exists.

Text ( t ) or Binary (b) sometimes we also do need to mention that we are opening a text file or binary file, by default the mode will be “rt” which indicates read text , but lets say you wanna open binary file, then you can read the file using “b” like “rb” like that you can use it. We can use this when we want to open jpg file, pdf file or exe file

Reading from a file:

f = open("data2.txt","r")

print(f.read())

writing to a file:

f = open("data.txt","w")

f.write(

    '''Hello world,

nice to hear about you buddy!'''

)

f.close() # closing is important

appending to a file:

f = open("data.txt","w")

f.write(

    '''Hello world,

nice to hear about you buddy!'''

)

f.close() # closing is important

now using f.close() will be boring like you everytime you have to write it while file handling.

So what you can do like following:

with open("data.txt","w") as f:

    f.write("Code with harry")

read,readlines and other file handling methods

# readline method: it will read the first line

f = open("data.txt","r")

while True:

    line = f.readline()

    if not line:

        break

    print(line)

print("f after reading all the lines: ",f.readline())

# will f.readline still print the first line of text or not?

#readline() method program to extract numbers from text and converting them into a list of numbers

f = open("data.txt","r")

l = []

while True:

    line = f.readline()

    if not line:

        break

    m = line.split(",")

    m = line.split("\n")

    for i in range(len(m)):

        if not m[i]:

            m.pop(i)

    m = m[0].split(",")

    for item in m:

        l.append(int(item))

print(l)

readlines() method will read whatever is in the file and returns a list of it

# readlines() method

f = open("data.txt","r")

print(f.readlines())

hello

my name is zain

and im not a terrorists

writelines() method

# writelines method

# the data you are providing in writelines function must be a string not an int

f = open("data.txt","w")

f.writelines(["12,23,1,23,12,2","\n","231,213,21,1,3,4,2,2"])

f.close()