Get Password From User

#!/usr/bin/python3

import getpass

#ba in vaghti pass vared misheh ,display nemisheh va mirizimesh tuye password file --paramiko

print("enter password for windows: " )

win\_pass =getpass.getpass()

print(f"you have enterd the pass {win\_pass}")

#!/usr/bin/python3.6

import paramiko

ip = "10.100.8.153"

myusername ='pouria'

mypassword ='Aa@123456'

ssh = paramiko.SSHClient()

ssh.load\_system\_host\_keys()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect(ip,

username=myusername,

password=mypassword,

look\_for\_keys = False )

ssh\_stdin, ssh\_stdout, ssh\_stderr =ssh.exec\_command("ls")

output = ssh\_stdout.readlines()

ssh.close()

----------------------Another Example------------

#!/usr/bin/python3

import paramiko

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect(hostname = '10.100.8.153',username = "p.jalilian",password ="Aa@123456",port = "22" )

sftp\_client = ssh.open\_sftp()

source\_file\_path = r"/appserver/nginx.tar.gz"

destination\_file\_path=r"/home/nginx.tar.gz"

sftp\_client.put(source\_file\_path,destination\_file\_path)

ssh.close()

Working wirth OS

#!/usr/bin/python3

import os

import time

import datetime

#Python method walk() generates the file names in a directory tree by walking the tree either top-down or bottom-up.

#followlinks − This visits directories pointed to by symlinks, if set to true.Leads to unstoppable recursions; it is set to true.The walk () function does not take the record of the directories that it has already traversed.

#onerror − This can show error to continue with the walk, or raise the exception to abort the walk.

#topdown − If optional argument topdown is True or not specified, directories are scanned from top-down. If topdown is set to False, directories are scanned from bottom-up.

print("Here topdown is False")

time.sleep(5)

for root ,dirs ,files in os.walk("/home/pouria", topdown = False) :

for name in files :

print(os.path.join(root,name))

for name in dirs:

print(os.path.join(root, name))

print("here topdown is True")

time.sleep(5)

import os

for (root ,dirs ,files) in os.walk("/home/pouria", topdown = True) :

for name in files :

print(os.path.join(root,name))

for name in dirs:

print(os.path.join(root, name))

time.sleep(5)

path = '/home/'

i = 0

for (path,dirs,files) in os.walk(path):

print(path)

print(dirs)

print(files)

print("------------------")

i +=1

if i >=5 :

break

#Print files in a directory in a row.

path ="/home/pouria/"

files=os.listdir(path)

for f in files :

print(f)

if f =='date.txt' :

print('hello!!!')

#with open(dirs.joinpath(f),'r'):

#print(f.read())

#in kar nakard--- 'list' object has no attribute 'joinpath' -hadafam in bud ke agar resid be un file made nazar,printesh koneh.

#---- os.walk() uses some features that allow excluding the hidden directories.

path ="/home/"

for root,dirs,files in os.walk(path) :

print(root)

#the bellow is like a comprehension concept.

dirs[:] = [d for d in dirs if not d.startswith('.')]

for dir in dirs :

print(os.path.join(root,dir))

for file in files :

print(os.path.join(root, file))

#searching a file ends with mp3 simple search by name

dir\_path = '/home/pouria/'

for root,dirs,files in os.walk(dir\_path) :

for file in files :

if file.endswith('mp3'):

path\_new=root+str(file)

#print("file modification time of a file :" +os.path.getmtime(path\_new)

#print("file creation time of a file "+ os.path.getctime(path\_new)

print(root+str(file))

print(path\_new)

path=r'/home/pouria/date.txt'

m\_time\_date\_txt=os.path.getmtime(path)

print("modification time of a given file is ", m\_time\_date\_txt)

new\_mtime=datetime.datetime.fromtimestamp(m\_time\_date\_txt)

print("the m time was in fact :", new\_mtime)

#pathlib.Path('file\_path').stat().st\_mtime: Best cross-platform way to get file modification time in Python.

## file creation timestamp in float

#c\_time = os.path.getctime(path)

#renaming a file

old\_name = "/home/pouria/date.txt"

new\_name = "/home/pouria/date-time.txt"

os.rename(old\_name,new\_name)

#finding a file using glob

import os.glob

os.chdir('/home/pouria/')

for file in glob.glob("\*.mp3") :

print(file)

#!/usr/bin/python3

import os

import tarfile

from tqdm import tqdm

path='/appserver/tarsample/'

#path=['1','2']

all1=os.listdir(path)

allnew=' '.join(all1)

print("-------------------")

print("all1 is : ",all1)

print("allnew is : ",allnew)

print("type all1 is : ",type(all1))

print("type allnew is : ",type(allnew))

#string ------ > print(type(allnew))

#----list ----->> print(type(all1))

for i in range(len(all1)):

print(all1[i])

print("type all1[0] is ",type(all1[1]))

strname='alireza'

if strname.endswith('a')

print("hello")

#!/usr/bin/python3

#we can do it using os / or / pathlib

def rename\_os():

#using os module

os.rename ('esme file avali'.'new esm')

def rename\_pathlib() :

#using pathlib

file = Path('/home/p.jalilian/name.png')

file.rename(/home/p.jalilian/namename.png)

#ino khub nemidunam

if \_\_name\_\_ == "\_\_main\_\_" :

rename\_os()

rename\_pathlib()

#using shutil to move file to another fodler

def move\_file():

os.mkdir('/home/p.jalilian/svg')

shutil.move('/home/file1','/home/p.jalilian/svg/file1')

#inja chon esme file nadadim ,kole motavaha ro montaghel mikoneh,ama bala faghat tak file

def move\_files():

shutil.move('/home/','/home/p.jalilian/')

if \_\_name\_\_ == "\_\_main\_\_" :

move\_files():

move\_file()

#!/usr/bin/python3

import glob, os

os.chdir("/home/")

how to search for a specific file extension with python

for file in glob.glob("\*.mp3") :

print(file)

python find files recursive :

import glob

files = glob.glob(path + "/\*\*/\*.txt", recursive = True)

import os

#!/usr/bin/python3

import os

os.mkdir("ali")

print(os.path.isdir("/home/el"))

print(os.path.exists("/home/el/myfile.txt"))

python get files in directory :

from pathlib import Path

for txt\_path in Path("/path/folder/directory").glob("\*.txt"):

print(txt\_path)

python find specific file in directory :

text\_files = glob.glob(path + "/\*\*/\*.txt")

#!/usr/bin/python3.6

import os

import time

print ("the os is ", os.name)

time.sleep(10)

#print current working directory ----like pwd

print("your current working directory is :" + os.getcwd())

#change to another directory directly ----- like cd

os.chdir("/home/pouria")

print("your new WD is :" ,os.getcwd())

#if we want to change working directory relative to our current -----:

parent\_dir = os.getcwd()

added\_dir ='keke'

#mode needs to be searched and documented

mode = 0o777

path = os.path.join(parent\_dir, added\_dir)

if os.path.exists :

print("the path is ",path,"and it's contents are : ",os.listdir(path))

print(os.listdir(parent\_dir))

print(os.listdir(path))

new\_added='jalilian'

new\_path =os.path.join(parent\_dir, new\_added)

#agar ye dir nabud,besazad anra -----then ... harkari khast dakhelesh bokoneh :

if not os.path.exists(new\_path) :

os.mkdir(new\_path, mode)

os.makedirs('/home/pouria/keke/hasti/fakhravar', mode)

if not os.path.exists('/home/appserver/jalilian/fakhravar/hasti') :

os.makedirs('/home/appserver/jalilian/fakhravar/hasti', mode)

if not os.path.exists('/home/jalilian/hasti/fakhravar') :

os.makedirs('/home/jalilian/hasti/fakhravar', mode)

print("list of directory is /home/appserver/jalilian ", os.listdir('/home/appserver/jalilian/'))

#how to remove path using python

print ("before deleting --deleted the contents of /home/appserver/ is ", os.listdir('/home/appserver/'))

time.sleep(10)

os.rmdir ('/home/appserver/deleted')

print("after deleting --deleted the contents of /home/appserver/ is like ",os.listdir('/home/appserver/'))

#how to remove files

print ("before deleting --deletyed.file the contents of /home/file/ is ", os.listdir('/home/file/'))

time.sleep(10)

os.remove('/home/file/deletyed.file')

print ("after deleting -- deletyed.file the contents of /home/file/ is ", os.listdir('/home/file/'))

#!/usr/bin/python3

with open(r"/home/pouria/life.txt", mode = "w") as f :

f.write("Life ------")

#Note: when opening a file in Write (‘w’) mode, the current file contents will be truncated. Use this mode carefully.

#The r before the file path ensure that the path name string is read as a raw string. The backslashes in the path will then be treated as literals and not as escape characters

from pathlib import Path

dir\_path =Path('/home/pouria/')

file\_name = 'dotin.txt'

#check if directory exists

if dir\_path.is\_dir() :

with open(dir\_path.joinpath(file\_name), mode = "w") as f :

f.write("Dotin .....")

print("File created")

else :

print("Directory does not exists")

#here directory does not exist

dir\_path =Path('/home/jasem/')

file\_name = 'dotin.txt'

if dir\_path.is\_dir() :

with open(dir\_path.joinpath(file\_name), mode = "w") as f :

f.write("Dotin .....")

print("File created")

else :

print("Directory does not exists")

#---------------Append text to an existing file

with open(r"/home/pouria/life.txt",mode ="a") as f :

f.write("\nThis is a new Line appended to life.txt\n ")

#write multiple items each in a sepereted line :

#the file will be created just here the code is executed

languages = ['python','java','c#','R','go']

file\_name='languages.txt'

with open (file\_name,'w') as f :

for item in languages:

f.write(item +'\n')

#read a file -----------------:

with open(r"/home/pouria/life.txt",mode="r") as f:

print("New Text File Content is : ",f.read())

Interactive Input

#!/usr/bin/python3

color = input("what is your fav color ? ")

print("you said :",color)

Working With Date

#!/usr/bin/python3

import datetime

import time

#printing current date on a system

print("current date and time is ",datetime.datetime.now())

#just print date ---then we just import date

print("current day is ",datetime.date.today())

#important -----how to see functions a module provides us-----using a built in function :

print("what datedite module gives us",dir(datetime))

#how to set\_date\_using\_python ----- :

d1 = datetime.date(2022, 9, 2)

print(d1)

#how to set\_time\_using\_python ----- :

t1= datetime.time(22,21,31)

print(t1)

#epoch time ------first second from 1/jan/1970 ----till now :

d2\_from\_timestamp = datetime.date.fromtimestamp(1567262626)

#convert a epoch timestamp to a readable format :

print("input fromtime stamp converted to datetime is ", d2\_from\_timestamp )

#convert the current time to epoch time stamp ---:

print("current datetime in epoch format : ",datetime.datetime.now().timestamp())

#fetch year,monthe,day from a datetime.date.now() to be used later

d1\_now = datetime.date.today()

print("todays year is :{0} , month is {1} , and day is {2} ".format(d1\_now.year,d1\_now.month , d1\_now.day))

#fetch hour,minute,seconds,microseconds from a datetime.time.now() to be used later

t1\_now =datetime.datetime.now().time()

print ("current time is :")

print (t1\_now)

time.sleep (1)

print("refered to current time ,t1\_now.hour {0} , t1\_now.minute : {1} , t1\_now.seconds: {2} , t1\_now.microseconds : {3} :".format(t1\_now.hour, t1\_now.minute,t1\_now.second,t1\_now.microsecond))

#calculating delta between to given time :

time\_1 =datetime.datetime(2022, 10, 10, 10, 10, 10, 10 )

time\_2 =datetime.datetime(2023, 10, 11, 11, 10, 10,10 )

delta\_time = time\_2 - time\_1

print("print delta time given betweeen to different times : ", delta\_time)

#printing type of given time/dtae

print("type of time is : ",type(delta\_time))

#tafavote estefadeh az \_\_name\_\_ dar type e yek variable

print("type of time is : ",type(delta\_time).\_\_name\_\_)

#jense unix timestamp be sanieh ast ---- :

print("total delta seconds ? :",delta\_time.total\_seconds())

#working with time delta ----- :

tm1 = datetime.datetime.now() - datetime.datetime(year=2022 , month=9 , day=1, hour=12,minute=50 )

print("delta time for this example is :",tm1)

#inja yani mikham baram time delta ro hesab bokoni va dar vahede saat natijeh ro culculate bokoni ---jense khoruji delta hour khahad bud

tm2 = datetime.timedelta(hours=1)

#inja dareh mige dela ro ke hesab kardi ,hala dar vahede saat be man neshunesh bedeh,total ro taghsim bar 1 saat be onvane vahede payeh behesh pass midim

print("timedelta in hours would be :",tm1 /tm2)

#inja ---yek string dadeh shodeh ke mikhaym un ro convert konim be zaman :

#str to time

#tuy in scenario ---python agar bekhaym behesh begim sale 2022 bayad behesh 20 ro pas bedim,20 ro nemifahmeh .

#nokte --filed hay marbut be tarikh yani sal mah ruz ba harfe kuchik hastan ama unhaei ke be time eshareh dareh yani saat/daghighe va sanieh bozorg hastan

strdate = "2022/10/10-14.40.52"

#ba strptime ---- ba hamun format miaym vorudi midim va khoruji midim --va harja az . ya / ya ... estefadeh shodeh man hamun ro midam,datetime miad tebghe gofteye man h/m/s/y/m/d ro moikesheh birun azash

print("strdate in str format is : " ,strdate , "and when it is converted to datetimeformat would be : ",datetime.datetime.strptime(strdate,"20%y/%m/%d-%H.%M.%S") )

datetostr = datetime.datetime.now().strftime("20%y//%m//%d// ---- %H--%M--%S")

print ("current time is :",datetime.datetime.now(),"and when it is converted to string it is now like :" ,datetostr)

print("type of converted time/date is :" , type(datetostr).\_\_name\_\_)

Working With NTP

#!/usr/bin/python3

#import telnetlib

#import time

#telnetlib.Telnet('10.100.8.153', '22',2)

import ntplib

import time

NIST = '10.100.8.254'

ntp = ntplib.NTPClient()

ntpResponse = ntp.request(NIST)

if (ntpResponse):

now = time.time()

diff = now-ntpResponse.tx\_time

print(diff)

#!/usr/bin/python3

#import telnetlib

#import time

#telnetlib.Telnet('10.100.8.153', '22',2)

from ntptimenow import NTPTimeNow

NTPTimeNow(poolservers:str='10.100.8.254',version:int=3)

now = NTPTimeNow().ntp\_update() # It's datetime.

Operators

#!/usr/bin/python3

# < ------ <= ----- > ------ >= ------ ==

# False -----= > 0 -------- True ------ => 1

#print(False > True ) ----->> Will return False. casue it will seem like (0>1)

#when we compare strtings ----- eg. if "aaaA" > "aaaB" ---> will return False because A is less than B inj ascii number.

# in fact,it will check the first mismatch ,not the length of string.

#print ('a' <= 'b') ----> will print True

SSH Using Paramiko

#!/usr/bin/python3

import paramiko

import os

ssh = paramiko.SSHClient()

ssh.load\_system\_host\_keys()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect("10.100.8.152", username="Administrator", password="567tyuGHJbnm")

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command("echo import os >pyos.py")

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command('echo os.mkdir("jafar")>>pyos.py')

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command("py pyos.py")

out = ssh\_stdout.readlines()

print(out)

ssh.close()

#!/usr/bin/python3

import paramiko

import os

ssh = paramiko.SSHClient()

ssh.load\_system\_host\_keys()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect("10.100.8.152", username="Administrator", password="567tyuGHJbnm")

#ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command("echo import os >C:\pyos.py && echo print(os.mkdir(r"C:\hasan"))>>C:\pyos.py && python C:\pyos.py")

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command("echo import os >pyos.py")

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command('echo os.mkdir("jafar")>>pyos.py')

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command("py pyos.py")

out = ssh\_stdout.readlines()

print(out)

ssh.close()

#echo import os >pyos.py && echo print(os.mkdir(r"C:\ali"))>>pyos.py &&python pyos.py

Working With Permissions

#!/usr/bin/python3

import os

print(os.stat("/home/pouria/1.mp3"))

print(os.access('/home/pouria/1.mp3', os.X\_OK))

os.chmod("/home/pouria/1.mp3", stat.S\_IRWXU)

SSH To Windows From Linux Using Paramiko

#!/usr/bin/python3

import paramiko

ssh = paramiko.SSHClient()

ssh.load\_system\_host\_keys()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect("10.100.8.152", username="Administrator", password="567tyuGHJbnm")

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command("echo import datetime >date.py")

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command("echo import time >>date.py")

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command('echo print(datetime.datetime.now()) >>date.py')

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command("py date.py")

out = ssh\_stdout.readlines()

print(out)

ssh.close()

out = ssh\_stdout.readlines()

print(out)

ssh.close()

#echo import os >pyos.py && echo print(os.mkdir(r"C:\ali"))>>pyos.py &&python pyos.py

Working With Regex

#!/usr/bin/python3

import re

#re.search is just a boolian comprator ,it returns true or false

txt = "This is Ali"

x=re.search("^T.\*i$",txt )

if x:

print("pattern is here!!-----x")

else:

print("pattern not found")

txt = "This is hasan abbas"

y=re.search(".\*hasan$",txt )

if y:

print("pattern is here!!----y")

else:

print("pattern not found----y")

#findall reaturns date if it matches and we can print its contents

txt = "pouria ooo oooo pouria jaliloain ooo"

x = re.findall("o", txt)

print(x)

txt = "pouria ooo oooo pouria jaliloain ooo"

x = re.findall("oo", txt)

print(x)

txt = "pouria ooo oooo pouria jaliloain ooo"

x = re.findall("jasem", txt)

print(x)

#re.search just returns first match not all occurances ----- as we know strings are indexed from 0 ---so if a match is returned(the first one) ,it gives start and end index number of the match ,eq <re.Match object; span=(4, 8), match='rain'> ----- if no match is retuned , None will be returned

txt = "The rain in Spain"

x = re.search("rain", txt)

print(x)

#re.split(\_...., ..... ) ------------------ it returnes the given string sepereted by parameter(even a space or character or a word as a seperator ---eg : if "the rain will rain now " and we split ("rain") --result wiil be an array ["the ",will " now"] as result

#Split the string at every white-space character:

txt = "The rain in Spain"

x = re.split("i", txt)

print(x)

#The sub() function replaces the matches with the text of your choice: --it acts like sed --by default it replaces all ocurances in a line ,but multiline must be checked

# re.sub("source pattern","alternative pattern",file.txt,number of occurances of sed)------ x = re.sub("\s", "#", txt, 2) ---here \s is space and just two first spaces will be converted to #

txt = "The rain in Spain"

print(txt)

x = re.sub("i", "#", txt)

print(x)

#re.search worked simple boolian --if we want to get start and end position on index of matched object ----->> (10,13) ----means match started at index 10 and its end character at 12 ,and in index number 13 the exztra info is started and is not part of the match ---- just the first match ois gived ,not all ocurances

txt = "The rain inn in Spain inn hassan i mmmmi"

x = re.search(r"inn", txt)

print(x.span())

#in re.search ----in we use x.string() in resulted printed ---- all the containing line will be printed

txt = "The rain in in in Spain"

x = re.search(r"in", txt)

print(x.string)

#------using re.search as refrence and x=re.search(" "," ")

#print(x.group()) ---- >> Search for an upper case "S" character in the beginning of a word, and print the word---it just prints first one not all

txt = "The rain in Spain Spania"

x = re.search(r"\bS\w+", txt)

print(x.group())

Working With Request

#!/usr/bin/python3

import requests

import datetime

import time

from getpass import getpass

from requests.adapters import HTTPAdapter

#from requests.auth import HTTPBasicAuth

#just get a url data using get method

response = requests.get("http://google.com")

print(response.status\_code)

#agar chizi chizi joz 200/201/302/... yani okeye,yani error nadeh --ama agar status.code khasi moheme baramun,if ro base response.status\_code set mikonim.

if response :

print("successfully get the data from google")

else :

print("faild")

#exception handling in requests ----- :

try :

response = requests.get("https://api.github.com")

response.raise\_for\_status()

except requests.exception.HTTPAdapter as http\_err:

print("https error eccoured: ".format(http\_err))

except Exception as err :

print("other error occured :".format(err))

else :

print("successfully gotten the url ")

#----------------------------------------------------------------------------------

#neshan dadane content url get shodeh

time.sleep(5)

print ("url gotten contents are in raw byte :",response.content)

time.sleep(5)

print ("url gotten contents are in text :",response.text)

time.sleep(5)

print ("url gotten contents are in jason :",response.json())

time.sleep(5)

print ("url gotten contents headers are :",response.headers)

time.sleep(5)

time.sleep(5)

#print content type :

print ("url gotten contents type is :",response.headers['Content-Type']) # not case sensitive

#using query strings in python get ------for search

new\_response = requests.get('https://api.github.com/search/repositories',

params={'q':'netmiko+language:python'})

new\_response\_in\_json = new\_response.json()

repositories = new\_response\_in\_json['items'][0]

print("repository name :",repositories["name"])

print("repository description: ", repositories["description"])

#----------------------------------------------------------------------------------

#other http methods ////post///delete//patch

requests.post('https://httpbin.org/post',data={'key':'value'})

request.delete('https://httpbin.org/delete')

requests.patch('https://httpbin.org/patch', data ={'key':'value'})

#working wityh API

response1 =requests.post('https://bin.org/post',data={'key':'value'})

response2 =requests.post('https://httpbin.org/post',data=[('key','value')])

response3 =requests.post('https://httpbin.org/post',json={'key':'value'})

#content type by default hamisheh app/json ast.

print("header type :",response1.headers["content-type"])

#Authentication in Python ---Basic Auth using username/password

requests.get('https://api.github.com/user',auth=HTTPBasicAuth('username',getpass()))

#ssl Check Certification

requests.get('https://api.github.com/',verify=False)

#timeout set

requests.get('https://api.github.com/',timeout=(2,5)) #a=time to establish a connection b=time to wait after establish a connection and for getting data back in seconds

#requests session -----------------baray behbud performance dar request va kar ba API ha bayad az object type session dar module request estefadeh konim,kar hay pichidehtari ba in method misheh kard:

#masalan baray inke vaghti kar ba session tamum shod,session i baz namuneh miaym unro dakhele with gharar midim ke vaghti karesh tamum shod connection ro bebandeh

#performance tuning of an API communication

#kole module requests.session ra ba komake with ba session seda mikonim.

with requests.Session() as session :

session.auth = ('username',getpass())

response\_new\_session =session.get('https://api.github.com/user')

print(response\_new\_session.headers)

print(response\_new\_session.json())

#----------------------------------------------------------------------------------

#masalan mikhaym ruy raftar default http dast bebaroim,masalan max\_retries ro dastkari konim.

retries\_adapter = HTTPAdapter(max\_retries=3)

session =requests.Session()

session.mount('https://api.github.com/user', retries\_adapter)

session.get('https://api.github.com/user'))

Working With Rsync

#!/usr/bin/python3

import os

import sysrsync

sysrsync.run(source='/home/user/files',

destination=r'C:\Users\Administrator\Desktop\files',

destination\_ssh='root@10.100.8.153',

options=['-a'],

exclusions=['excluded.1', 'excluded.2'])

# runs 'rsync -a /home/user/files/ myserver:/home/server/files --exclude file\_to\_exclude --exclude unwanted\_file'

Working With SFTP

#!/usr/bin/python3

import paramiko

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect(hostname='10.100.8.153',username = 'pouria',password = 'Aa@123456',port=22)

sftp\_client.put("/appserver/pyprojects/synwindows.py","/home/pouria/")

sftp\_client.close()

ssh.close()

Working With Shutil

#!/usr/bin/python3

import os

import shutil

source = "/home/p.jalilian/usersinfo"

destination0 = "/home/p.jalilian/passwdbackupcopyfile"

#shutil.copyfile ---- >> permission ba tavajoh be mask va default ha set misheh-timestamp fargh dareh ba asli

#shutil.copyfile(source, destination)

shutil.copyfile(source, destination0)

#shutil.copy ----- >> permisison are the same and source ,but some metadata like timestamps are not ok

destination1 = "/home/p.jalilian/passwdbackupcopy"

shutil.copy(source, destination1)

#shutil.copy2 ------ >>some metadata are the dsame as source --permissions are the same-mtime as we see in ls are the same

destination2 = "/home/p.jalilian/passwdbackupcopy2"

shutil.copy2(source, destination2)

#shutil.copymode ----- > permisison ro az source ruy des emal mikoneh--hardo az ghabl hastan

destination3 = "/home/p.jalilian/newfile"

shutil.copymode(source, destination3)

#shutil.copystat --->>>>all meta data of source to dest,not just mtime and perm ---src and dst must exist already or be handled before shutil.just metadata like stat result---but owner is not changed,

destination4 = "/home/p.jalilian/newfile"

shutil.copystat(source , destination4)

#shutil.copyfileobj------copy contents from srcfile to dest file--src will be open in rread mode and dest in write mode and ...

#f1=open('f1.txt','r')

#f2=('f2.txt','w')

#shutil.copyfileobj(f1,f2))

#shutil.copytree ---> copy whole directory structure.source must exist but dest will be created---metadata will be kept at destination

new\_source = '/home/p.jalilian'

new\_destination = '/home/keke'

shutil.copytree(new\_source,new\_destination)

#shutil.rmtree(our\_destination\_dir\_path)

Working With ZIP

#!/usr/bin/python3

import zipfile

import shutil

with zipfile.ZipFile('/appserver/filesssss.zip','r') as myarchive :

# myarchive.extractall()

myarchive.extract('list/trest/testpasswd')

# myarchive.extractall()

#by default haminja extract mikoneh ke file py hast

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

#!/usr/bin/python3

import zipfile

import shutil

def create\_zip(files):

with zipfile.ZipFile('/home/p.jalilian/archivaname.zip','w') as archive :

for file in files :

archive.write(file)

if \_\_name\_\_ == '\_\_main\_\_':

files = ["/home/file1","/home/file2","/appserver/file4"]

create\_zip(files)

#nokte=src files hazf nemishan----in overwrite mikoneh,age az ghabl basheh delete va tebghe list jadid ijad misheh

#ba in kar agar esme ham name dar maghsad basheh ,replace misheh ----ama gahi ma mikhaym ke append besheh

def append\_zip(files):

with zipfile.ZipFile('/home/p.jalilian/archivaname.zip','a') as archive :

for file in files :

archive.write(file)

if \_\_name\_\_ == '\_\_main\_\_':

files = ["/home/file3","/home/file2","/appserver/file4"]

append\_zip(files)

#in baray absolute file oke ast,ama agar nested dir basheh va bekhaym hamasho zip koneh digeh dastan darim

directory\_path = '/home/p.jalilian/'

def create\_zip\_nested\_dir(directory\_path):

shutil.make\_archive('/appserver/archiveName.zip', 'zip', directory\_path)

#!/usr/bin/python3

import zipfile

import shutil

with zipfile.ZipFile('/appserver/filesssss.zip','r') as archive :

print(archive.namelist())

print(archive.getinfo('list/trest/testpasswd'))

#nokteh : inja relative esme file ro migim --yani /appserver/files/list/;... nemizanim,bejash az list/... minevisim

#namayesh etelaate file

with zipfile.ZipFile('/appserver/filesssss.zip','r') as archive :

with archive.open('list/trest/testpasswd') as myfile :

print(myfile.read())

# print(archive.namelist())

# print(archive.getinfo('list/trest/testpasswd'))

######### be komake in mohata ra chap kardam

with zipfile.ZipFile('/appserver/filesssss.zip','r') as myarchive :

# myarchive.extractall()

myarchive.extract('list/trest/testpasswd')

myarchive.extractall()

#!/usr/bin/python3

import shutil

import zipfile

import os

#in baray absolute file oke ast,ama agar nested dir basheh va bekhaym hamasho zip koneh digeh dastan darim

#misheh dir ro behesh pass bedim ,agar nadim meghdar pishfarz bala ro mikhuneh --yani ---- create\_zip\_nested\_dir("/hom/ali") --- ya create\_zip\_nested\_dir()

directory\_path = '/home/p.jalilian/'

#def create\_zip\_nested\_dir(directory\_path):

shutil.make\_archive('/appserver/filesssss', 'tar', '/home/p.jalilian/')

#khodesh .zip mizaneh tahesh

#if \_\_name\_\_ == '\_\_main\_\_':

# create\_zip\_nested\_dir("/home/p.jalilian/")

String Processing

#!/usr/bin/python3

sample=" Python Is Ok "

print(sample[0])

print(sample[0:3])

print(sample[3:])

print(sample[:-3])

print(sample[-2:-7777777:-1])

print(sample.rstrip())

print(sample.lstrip())

print(sample.strip())

print(sample.lstrip().rstrip())

sample2="10.102.0.101,10.102.0.105,10.102.0.110"

print(sample2.split(','))

print(sample2.split(','))

print(' '.join(sample2.split(',')))

sample3='hasan reza ali P'

print("Is" in sample)

print("Iso" in sample)

print(sample3.find('P'))

print(sample3.find('reza'))

print(sample3.find('K'))

sample4='102 103 104 105 108'

a=sample4.replace('108','1111')

print(a)

print(sample4.replace('108','1111'))

print(sample4)

sample5='jahan \n doust'

sample6='jahan \\n doust'

sample7=r"jahan \n doust"

print(sample5)

print(sample6)

print(sample7)

print("jalil \n hasan")

print(r"jalil \n hasan")

print(r"""salam hesam,khubi \n che khabar

halet khube ??""")

print(R"""salam hesam,khubi \n che khabar

halet khube ??""")

Sync Linux-Windows

#!/usr/bin/python3

import sysrsync

#agar slash dar src bashad yani tush ro sync bokon ama agar slash nadasht yani folder ham besaz ----agar maghsad ham bashad baz ham hamin modeli ast.---tanazor srcdr/dstdir va dstcontents ba srccontents bargharar bashad

sysrsync.run(source='/home/user/files/',

destination=r'C:\Users\Administrator\Downloads',

destination\_ssh='administrator@10.100.8.152',

options=['-a','-v'])

# runs 'rsync -a /home/users/files/ myserver:/home/server/files'

Working With Tar Files

#!/usr/bin/python3

import os

import tarfile

from tqdm import tqdm

path='/appserver/tarsample.tar'

#path=['1','2']

#allnew=' '.join(all1)

print("-------------------")

tar = tarfile.open(path,"r:tar")

tar.extractall()

tar.close()

#print(allnew)

#print(all)

#string ------ > print(type(allnew))

#----list ----->> print(type(all1))

Telnet

#!/usr/bin/python3

#import telnetlib

#import time

#telnetlib.Telnet('10.100.8.153', '22',2)

#Telnet.close()

import pyfiglet

import sys

import socket

from datetime import datetime

ascii\_banner = pyfiglet.figlet\_format("PORT SCANNER")

print(ascii\_banner)

# Defining a target

#if len(sys.argv) == 2:

# translate hostname to IPv4

# target = socket.gethostbyname(sys.argv[1])

#else:

# print("Invalid amount of Argument")

target='10.100.8.153'

# Add Banner

print("-" \* 50)

print("Scanning Target: " + target)

print("Scanning started at:" + str(datetime.now()))

print("-" \* 50)

try:

# will scan ports between 1 to 65,535

for port in range(1,65535):

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

socket.setdefaulttimeout(1)

# returns an error indicator

result = s.connect\_ex((target,port))

if result ==0:

print("Port {} is open".format(port))

s.close()

except KeyboardInterrupt:

print("\n Exiting Program !!!!")

sys.exit()

except socket.gaierror:

print("\n Hostname Could Not Be Resolved !!!!")

sys.exit()

except socket.error:

print("\ Server not responding !!!!")

sys.exit()

Working With Thread

#!/usr/bin/python3

import threading

import time

import datetime

start = time.perf\_counter()

def do\_something():

print("Pouria")

time.sleep(5)

t1=threading.Thread(target=do\_something)

for i in range(1000):

t = threading.Thread(target=do\_something)

print(i)

t.start()

finish= time.perf\_counter()

print(f'Finished in {round(finish-start, 2)} seconds(s)')

#!/usr/bin/python3

import threading

import time

import datetime

start = time.perf\_counter()

def do\_something():

print("Pouria")

time.sleep(1)

threads = []

t1=threading.Thread(target=do\_something)

for i in range(10000):

#10000 ta thread misazeh ke harkodum yek print hastand\_yek sanieh wait-

t = threading.Thread(target=do\_something)

print(i)

t.start()

threads.append(t)

for thread in threads:

thread.join()

#vaghti join konim,migeh vaysa thread hara bezaram kenare ham,bad run konam

#ma join mizanim ke ghabl az start kardane har thread anhara ba ham join ---yeki konad va yekbareh ,dfar one vahed pass bedeh be cpu ke ejra besheh

finish= time.perf\_counter()

print(f'Finished in {round(finish-start, 2)} seconds(s)')

#ta inja shod min 13:00 <https://www.youtube.com/watch?v=IEEhzQoKtQU>

#!/usr/bin/python3

import threading

import time

from datetime import datetime

print("first time is :")

print(datetime.now())

class cpuchecker:

def printwall(self):

time.sleep(2)

print("hello world")

def \_\_init\_\_(self):

self.printwall()

# t = threading.Thread(target=self.printwall)

# t.start()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

print("at the end time is -----:")

print(datetime.now())

#!/usr/bin/python3

import threading

import time

from datetime import datetime

print("first time is :")

print(datetime.now())

class cpuchecker:

def printwall(self):

time.sleep(2)

print("hello world")

def \_\_init\_\_(self):

t = threading.Thread(target=self.printwall)

t.start()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

cpuchecker()

print("at the end time is -----:")

print(datetime.now())

#!/usr/bin/python3

import threading

import time

import datetime

start = time.perf\_counter()

def do\_something():

print("Hi Pouria")

time.sleep(2)

t1=threading.Thread(target=do\_something)

t2=threading.Thread(target=do\_something)

t3=threading.Thread(target=do\_something)

t4=threading.Thread(target=do\_something)

t2=threading.Thread(target=do\_something)

t1.start()

t2.start()

t3.start()

t4.start()

######inja be mahze inke mibine result un ghesmat montazere cpu hastesh,mire soraghe thread badi---yani dar lahze hamero chap mikoneh,dar surati ke harkodum hodum 1 sanieh tuy cpu zaman gereftan.ama cpu be komake thread montazereshun namundeh.

finish= time.perf\_counter()

print(f'Finished in {round(finish-start, 2)} seconds(s)')

Working With Time

#!/usr/bin/python3

from datetime import datetime

now = datetime.now()

print(now.year)

print(now.month)

print(now.hour///minute/second)

print(now.date())

print(now.date())

print(now.date())

print(now.time())

Working With Input

#!/usr/bin/python3

#in order to have command line arguments passed to script like ---: ./myscript.sh FCBCMS ---we need sys module

import sys

#print ("number of p-assed args by user is : ",len(sys.argv))

#print("arguments are :",sys.argv)

sys.argv.remove(sys.argv[0])

inja agar n ta pass dadeh bashim,tuy in maryhale avali ro ke az sefr shoru misheh ro hazf mikoneh.

nokteh : masalan mikhaym arg haro sum ko0nim----

arguments = sys.argv

sum=0

for arg in arguments :

try :

number= int(arg)

sum = sum + number

except Exeption:

print("bad input")

nokteh :hamisheh ino havasemun basheh.input ro control konim ,age lazeme converty koni,agar kolan eshtebah bud,error handle konim va ye message bedim.

Working With Date

#!/usr/bin/python3

import paramiko

ssh = paramiko.SSHClient()

ssh.load\_system\_host\_keys()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect("10.100.8.152", username="Administrator", password="567tyuGHJbnm")

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command("echo import datetime >date.py")

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command("echo import time >>date.py")

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command('echo print(datetime.datetime.now()) >>date.py')

ssh\_stdin,ssh\_stdout,ssh\_stderr = ssh.exec\_command("py date.py")

out = ssh\_stdout.readlines()

print(out)

ssh.close()

out = ssh\_stdout.readlines()

print(out)

ssh.close()

#echo import os >pyos.py && echo print(os.mkdir(r"C:\ali"))>>pyos.py &&python pyos.py

#!/usr/bin/python3

import datetime

import time

print("current date and time is ",datetime.datetime.now())

Working With DirSync

#!/usr/bin/python3

#import paramiko

#ssh = paramiko.SSHClient()

#ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

#ssh.connect(hostname = '10.100.8.153',username = "p.jalilian",password ="Aa@123456",port = "22" )

#sftp\_client = ssh.open\_sftp()

#source\_file\_path = "/home/file1.txt"

#destination\_file\_path=r"/home/file1.txt"

#sftp\_client.put(source\_file\_path,destination\_file\_path)

#get IP ---then based on that show list and operations---provide manual menue------------read inpout source and input dest dir -----or for paramiko read input path file to be transfered

#ssh.close()

import dirsync

#specify Source And Destination

source\_to\_sync = r"/appserver/source\_sync/"

destination\_to\_sync = r"/appserver/destination\_sync/"

dirsync.sync(source\_to\_sync, destination\_to\_sync,'sync', verbose = True )

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-----------------------------------------------------