**-----------------------------------------SYSTEM Module----------------------------------------------**

import os

**Cpu information**

def cpuinfo():

info='cat /proc/cpuinfo'

os.system(info)

cpuinfo()

**Cpu Architecture information**

def cpuArchiInfo():

info='lscpu'

os.system(info)

cpuArchiInfo()

**CPU Family**

def CPUFamily():

info='lscpu | grep "CPU family"'

os.system(info)

CPUFamily()

**Sockets**

def socket():

info='ss -s'

os.system(info)

socket()

**uptime**

def uptime():

info='uptime'

os.system(info)

uptime()

**View user activity in the system**

def last():

info='last'

os.system(info)

last()

**-----------------------------------------processing Module----------------------------------------------**

**no.of processing unit**

def nproc():

info='nproc'

os.system(info)

nproc()

**Processor**

def processor():

info='cat /proc/cpuinfo | grep "model name"'

os.system(info)

processor()

**Current process id**

def currentPId():

print("The current process id: ",os.getpid())

currentPId()

**Parent process id**

def PPId():

print("The parent process id: ",os.getppid())

PPId()

**Vendor id**

def vendorID():

info='lscpu | grep "Vendor ID"'

os.system(info)

vendorID()

**List out the currently process and their processor ids**

def ps():

info='ps'

os.system(info)

ps()

**To get information about processor**

def dmidecode():

info='sudo dmidecode -t processor'

os.system(info)

dmidecode()

**To get information about processor frequency**

def dmidecode():

info='sudo dmidecode -s processor-frequency'

os.system(info)

dmidecode()

**-----------------------------------------memory Module----------------------------------------------**

**Cache Information**

def cacheinfo():

info='lscpu | grep "cache"'

os.system(info)

cacheinfo()

**Cache Size**

def cacheSize():

info='cat /proc/cpuinfo | grep "cache size"'

os.system(info)

cacheSize()

**Ram info in kb**

def RamInfoInKB():

info='free '

os.system(info)

RamInfoInKB()

**Ram info in mb**

def RamInfoInMB():

info='free -m'

os.system(info)

RamInfoInMB()

**Ram info in gb**

def RamInfoInGB():

info='free -g'

os.system(info)

RamInfoInGB()

**Memory information**

def hwinfo():

info='hwinfo --memory'

os.system(info)

hwinfo()

**To show virtual memory statistics of the system**

def vmstat():

info='vmstat'

os.system(info)

vmstat()

**-----------------------------------------Date Module----------------------------------------------**

**To show time gmt or utc**

def date():

info='date -u'

os.system(info)

date()

**To show date and time 2 years ago**

def date():

info='date --date="2 year ago"'

os.system(info)

date()

**To show date and time 5 seconds ago**

def date():

info='date --date="5 sec ago"'

os.system(info)

date()

**To show date and time of previous day**

def date():

info='date --date="yesterday"'

os.system(info)

date()

**To show date and time 6 months ago**

def date():

info='date --date="6 month ago"'

os.system(info)

date()

**To show date and time of upcoming particular week day**

def date():

info='date --date="next tue"'

os.system(info)

date()

**To show date and time after 2 days**

def date():

info='date --date="2 day"'

os.system(info)

date()

**To show date and time of next day**

def date():

info='date --date="tomorrow"'

os.system(info)

date()

**Date**

def date():

info='date'

os.system(info)

date()

**Calender**

def calender():

info='cal'

os.system(info)

calender()

**-------------------------------------------------File Module------------------------------------------------**

**Create File:**

def createFile():

f=open("putting.txt","w+")

print("Created File Successfully!")

createFile()

**Write in File**

def writeInFile():

f=open("putting.txt","w+")

for I in range(10):

f.write("This is line %d\r\n"%(i+1))

print("Write in File Successfully!")

f.close()

writeInFile()

**read file**

def readFile():

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

if f.mode=="r":

contents=f.read()

print(contents)

readFile()

**Create zip file**

from os import path

from zipfile import ZipFile

def zipFile():

if path.exists("output2.txt"):

src=path.realpath("output2.txt")

with ZipFile("output2.zip","w") as newzip:

print("Create zip file successfully")

zipFile()

**unzipfile**

from os import path

from zipfile import ZipFile

def unzipFile():

with ZipFile("output2.zip","r") as newzip:

newzip.extractall()

print("unzip file successfully")

unzipFile()

**Delete file**

import os

try:

os.remove('test.txt')

print("File delete successfully")

except IOError:

print("File not delete")

**File Renamed**

import os

fd="file1.txt"

os.rename(fd,'new.txt')

print("File Successfully Renamed")

**-----------------------------------------------------------Network Module--------------------------------------------**

**Find the ip address and Name**

def nslookup():

info='nslookup google.com'

os.system(info)

nslookup()

**Route trace of packets to take the host**

def traceroute():

info='traceroute facebook.com'

os.system(info)

traceroute()

**To find the ip and subnet mask**

def ifconfig():

info='ifconfig'

os.system(info)

ifconfig()

**Tell about the various network connection,routing table and interface statistics**

def netstat():

info='netstat'

os.system(info)

netstat()

**List all tcp ports**

def netstat():

info='netstat -at'

os.system(info)

netstat()

**List all udp ports**

def netstat():

info='netstat -au'

os.system(info)

netstat()

**information about user login**

def w():

info='w'

os.system(info)

w()

**---------------------User Module-----------------------**

**User name**

def users():

info='users'

os.system(info)

users()

**Version Information**

def usersVersion():

info='users --version'

os.system(info)

usersVersion()

**Create user**

def createUser():

info='sudo useradd smith'

os.system(info)

createUser()

**Delete user**

def deleteUser():

info='sudo userdel smith'

os.system(info)

deleteUser()

**how to check all user we have newly created**

def newUserCreatedInfo():

info='cat /etc/passwd'

os.system(info)

newUserCreatedInfo()

**Create user with specific id**

def createUserID():

info='sudo useradd -u 1700 john'

os.system(info)

createUserID()

**How to check user id**

def checkUserID():

info='id -u john'

os.system(info)

checkUserID()

**Create user with some description**

def createUserDescription():

info='sudo useradd -c "CS Department" Hales'

os.system(info)

createUserDescription()

**Create user with expiry date**

def createUserExpire():

info='sudo useradd -e 2019-01-09 mack'

os.system(info)

createUserExpire()

**How to check user expiry date**

def checkUserExpire():

info='sudo chage -l mack'

os.system(info)

checkUserExpire()

**------------------Storage Module----------------------**

**Check file system Disk space usage**

def df():

info='df -h'

os.system(info)

df()

**Same as file system disk space usage but also show dummy file system disk space usage**

def df():

info='df -a'

os.system(info)

df()

**View all disk partition**

def fdisk():

info='sudo fdisk -l'

os.system(info)

fdisk()

**View partition on a specific disk**

def fdisk():

info='sudo fdisk -l /dev/sda1'

os.system(info)

fdisk()

**View the size of your partition**

def fdisk():

info='sudo fdisk -s /dev/sda1'

os.system(info)

fdisk()

**Create a hard disk partition**

def fdisk():

info='sudo fdisk /dev/sda1'

os.system(info)

fdisk()

**Delete a hard disk partition**

def fdisk():

info='sudo fdisk /dev/sda1'

os.system(info)

fdisk()

**Tell about the disk and partition list**

def hwinfo():

info='sudo hwinfo --block --short'

os.system(info)

hwinfo()