Sys Module:-

> Information about constant ,functions and the methods of interpreter

> Responsible for controlling and interacting with interpretor

> any information on operating system

> example: Piecharm on idle

> some files as sysmodule

> import sys

print(sys.version)

> ex: command line argument

print(sys.argv[])

> argv is list which contains the command line argument passed to a script

------

Various functions:-

stdress: Store error messages

stdin: Accept input from the user

stdout: Print to the screen

end: Quit or end the script

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

2)

os module:

> interacting with os

> primary use of this is

-- create folders

-- remove folders

-- move folders

> Sometimes it is used for change in the working directory

> also access the name of your file path <list dir>

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

Operation with OS module

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

osmodule.py

import os

print(os.getcwd())

=> for current working directory

print(os.getcwd())

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

Changing directory :

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

DIRECTORY SHOULD BE / OR \\ ONLY WORKS

> os.chdir(<directory name>)

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

import os

os.chdir("<directory name>")

print(os.getcwd())

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

CREATING FOLDER

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

os.mkdir(<directory name>/newfolder)

-----------

import os

os.mkdir(<directory name>/newfolder)

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

REMOVING DIRECTORY

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

os.rmdir("<directory name>")

---------

import os

os.rmdir("<directory name>")

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

REMOVING DIRECTORY WHICH IS NOT EMPTY

-----------

shuitl.rmtree("<directory name>")

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

import os

os.rmtree("<directory name>")

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

REMOVING FILE

os.remove("<file path">)

import shuilt

shutil.rmtree("<file path>")

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

2(a)

OS.PATH

import os

print(os.path.join("Dir","file")

a) Join: It takes one or more parts and joints them by using the current OS path separator

print(os.path.join("Dir","file")

b) Split: for scripting purpouse

print(os.path.split("Dir/file.py")

Import os

print(os.path.split("Dir/file.py")

c) exists: Check whether the path exist or not

mostly in TRUE OR FALSE

print(os.path.exists("<file name>")

import os

print(os.path.exists("<file name>")

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

3)

Sub process:

Let us interact with os to create new processes, past the info into and out of them and then return codes

Used by COIL function

So we can simply write the process call it and it will basically run by a command & and wait for the,nds and wait for the commands to finish and then gives back the return code