OS MODULE

The OS module in python provides functions for interacting with the operating system. OS, comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system.

1. os.name: This function gives the name of the operating system dependent module imported. The following names have currently been registered: 'posix', 'nt', 'os2', 'ce', 'java' and 'riscos'

```
import os
print(os.name)
```

2. os.getcwd(): Function os.getcwd(), returns the Current Working Directory(CWD) of the file used to execute the code, can vary from system to system.

```
import os
print(os.getcwd())
```

3. os.error: All functions in this module raise OSError in the case of invalid or inaccessible file names and paths, or other arguments that have the correct type, but are not accepted by the operating system. os.error is an alias for built-in OSError exception.

```
import os
try:
    # If the file does not exist,
    # then it would throw an IOError
    filename = 'GFG.txt'
    f = open(filename, 'rU')
    text = f.read()
    f.close()

# Control jumps directly to here if
#any of the above lines throws IOError.
except IOError:

    # print(os.error) will <class 'OSError'>
    print('Problem reading: ' + filename)
```

6. os.rename(): A file old.txt can be renamed to new.txt, using the function os.rename(). The name of the file changes only if, the file exists and user has sufficient privilege permission to change the file.

```
import os
fd = "GFG.txt"
os.rename(fd,'New.txt')
os.rename(fd,'New.txt')
```

5. os.close(): Close file descriptor fd. A file opened using open(), can be closed by close()only. But file opened through os.popen(), can be closed with close() or os.close(). If we try closing a file opened with open(), using os.close(), Python would throw TypeError.

```
import os
fd = "GFG.txt"
file = open(fd, 'r')
text = file.read()
print(text)
os.close(file)
```

-we can check the existence of file using os. path. exist(file_name) command.

ocessing > os module commands-1.py > ...

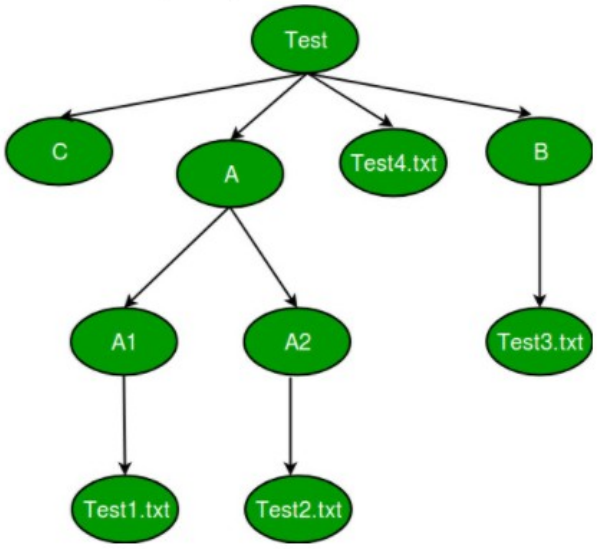
```
import os
a=os.getcwd() #to get the current working directory
print(a)
b=os.mkdir('name of folder') #to create the folder in presnt directory
print(b)
''' if we try to create the folder with same name again,it will give error
    so better approch is to check the file exist or not'''
c=os.path.exists('osmodule') #to check folder exist or not
print(c)
'''file creation'''
#we can create a file like this also ,opening file name with append mode and close it then
open('test_filename.txt','a').close()
#for changing current working directory-
os.chdir('path you want to change to')
#as we use 'f' for formatted string, similarly we can use 'r' for escape sequecmce
#if we use 'r' then no need for double \\
os.listdir() #it will give all folder as list in cwd & if we want other dir then put path
```

How to get all the paths of file present in the directory?

```
'''how to print all paths of files present in the dir'''
import os
for item in os.listdir():
    path = os.path.join(os.getcwd(),item)
    #using join func , we can join two paths
    print(path)
```

os.walk () in Python

How to traverse file system in Python ? Suppose we have given below file structure in our system and we want to traverse all it's branches completely from top to bottom ?



How does os.walk() work in python ?

OS.walk() generate the file names in a directory tree by walking the tree either top-down or bottom-up. For each directory in the tree rooted at directory top (including top itself), it yields a 3-tuple (dirpath, dirnames, filenames).

- **root :** Prints out directories only from what you specified.
- **dirs :** Prints out sub-directories from root.
- **files :** Prints out all files from root and directories.

```
'''what to do if you want to get every file and inside file of dir ?'''
import os
for (root,dirs,files) in os.walk(r'F:\Study Material'):
        print (root)
        print (dirs)
        print (files)
        print ('--------------------------------')

```

```
Mohd.Uzair@UzairPC MINGW64 /g/PythonPractice
$  env C:\\Users\\Mohd.Uzair\\AppData\\Local\\Programs\\Python\\Python38-32\\python.exe c:\\Users\\Mohd.Uzair\\.vscode\\extensions\\ms-python.p
\pythonFiles\\lib\\python\\debugpy\\launcher 52103 -- "g:\\PythonPractice\\os module commands-3.py"
F:\Study Material
['AMCAT Material', 'B.Tech [EE]', 'Current Affair Best 300 MCQ PDF', 'Jio Papers', 'Maths Formulas']
[]
--------------------------------
F:\Study Material\AMCAT Material
['AMCAT QB']
['5_6298834874073612478.zip', 'AMCAT AUTOMATA.pdf', 'amcat automata3.pdf', 'amcat automata4.pdf', 'amcat automata5.pdf', 'AMCAT Question Bank-1
on Bank-7.pdf', 'AMCAT_Quants.pdf', 'Cracking the Coding Interview.pdf', 'GM AMCAT QUANTS.pdf']
--------------------------------
F:\Study Material\AMCAT Material\AMCAT QB
['amcat', 'amcat automata & prog', 'AMCAT GD', 'amcat pr', 'Amcat Pro', 'cocubes', 'COCUBES GD', 'ELITMUS GD', 'IBM', 'infosys', 'infy', 'JAVA
mp']
['AMCAT Computer Programming.pdf', 'AMCAT Computer Programming1.pdf', 'amcat01.pdf', 'amcat4.pdf', 'AMCAT5.pdf', 'AMCAT6.pdf', 'AMCAT7.pdf', 'A
nce.pdf', 'AMCATENGLISH.pdf', 'AMCATLogical.pdf', 'AMCAT_AUTOMATA.pdf', 'GM AMCAT ENGLISH.pdf', 'GM AMCAT Logical.pdf', 'GM AMCAT QUANTS.pdf',
aid Paper.pdf', 'TCS-Aptitude-Paid.pdf', 'TCS-Latest-Placement-Paper-Questions-with-solutions.pdf', 'Tcs.txt', 'Topics-list.pdf']
--------------------------------
F:\Study Material\AMCAT Material\AMCAT QB\amcat
[]
['AMCAT Computer Programming1.pdf', 'GM New AMCAT English.pdf', 'GM New AMCAT Logical.pdf', 'GM New AMCAT Quants - Aptitude.pdf']
--------------------------------
F:\Study Material\AMCAT Material\AMCAT QB\amcat automata & prog
[]
['5_6172256917935620177.pdf', 'amcat  prog.pdf', 'AMCAT_paid.pdf']
--------------------------------
```