import os  
  
# GET CURRENT WORKING DIRECTORYprint(os.getcwd())  
# C:\Users\Basir Payenda\PycharmProjects\python\_work

# CHANGE DIRECTORYos.chdir("C:/Users/Basir Payenda")  
print(os.getcwd())  
# C:\Users\Basir Payenda

# TO LIST **FILES** AND **FOLDERS** IN CURRENT DIRECTORYprint(os.listdir())

# ['.anaconda', '.atom', '.bash\_history', '.conda', '.quokka', '.vscode', 'AppData', 'Application Data', 'Contacts',  
# 'Cookies', 'Desktop', 'Documents', 'Downloads', 'environment', 'Envs', 'Favorites', 'hell', 'j\_notebook', 'Links']

*# mkdir(), makedirs() - TO CREATE A NEW DIRECTORY IN CURRENT DIRECTORY*os.mkdir("New Folder")  
os.makedirs("Folder-1/Folder-2")

# You can create nested directories which you can't by 'mkdir'

*# rmdir(), removedirs() - TO REMOVE MASTER AND BRANCH DIRECTORIES*os.rmdir("New Folder") # this folder must be empty of any files and folders  
os.removedirs("Folder-1/Folder-2")

*# TO RENAME A FILE OR DIRECTORY*os.rename("requirements.txt", "renamed.txt")

# TO GET THE INFORMATION OF A FILE/ DIRECTORYprint(os.stat("renamed.txt"))  
# Output below  
# os.stat\_result(st\_mode=33206, st\_ino=9007199254886456, st\_dev=3167539864, st\_nlink=1, st\_uid=0, st\_gid=0, st\_size=336, st\_atime=1551595121, st\_mtime=1551595122, st\_ctime=1551595121)

*# to print out the last modification time*mod\_time = os.stat("renamed.txt").st\_mtime *# 1551595122.1092997*mod\_time = os.stat("renamed.txt").st\_size *# 336*

# to change that timestamp to human readable format  
from datetime import datetime  
mod\_time = datetime.fromtimestamp(mod\_time)  
print(mod\_time) *# 2019-03-03 11:38:42.109300*

*# TO VIEW ENTIRE DIRECTORY TREE AND FILES WITHIN IT*

*# dirpath, dirnames, filenames*

# the os.walk() function will return three values on each iteration through the loop:

# 1. A string of the current folder’s name

# 2. A list of strings of the folders in the current folder

# 3. A list of strings of the files in the current folder

# renaming these three variables is optional:

for dirpath, dirnames, filenames in os.walk("C:/Users/Basir Payenda/Desktop"):  
 print("Current Path: ", dirpath)  
 print("Directories: ", dirnames)  
 print("Files: ", filenames)  
 print()

'''Current Path: C:\py

Directories: ['.git', '.vscode', 'Automate the Boring Stuff with Python', 'BeautifulSoup', 'Class', 'CSV', 'decorators', 'Dict', 'Eric Matthews', 'files', 'for loop', 'Function', 'HTML CSS JS', 'Input, while', 'JSON', 'learning\_log', 'List Comprehensions', 'List, Tuple, Set, Dictionary', 'matplotlib', 'modules', 'my\_project', 'practice', 'testing your code', 'Useful\_examples', 'virtualenv', 'virtualenviron', 'Working with API']

Files: ['eg.py']

Current Path: C:\py\.git

Directories: ['hooks', 'info', 'logs', 'objects', 'refs']

Files: ['COMMIT\_EDITMSG', 'config', 'description', 'HEAD', 'index', 'packed-refs']

Current Path: C:\py\.git\hooks

Directories: []

Files: ['applypatch-msg.sample', 'commit-msg.sample', 'fsmonitor-watchman.sample', 'post-update.sample', 'pre-applypatch.sample', 'pre-commit.sample', 'pre-push.sample', 'pre-rebase.sample', 'pre-receive.sample', 'prepare-commit-msg.sample', 'update.sample']'''

*# TO JOIN SEVERAL PATHS*file\_path = os.path.join("C:/Users/Basir Payenda","Desktop")  
print(file\_path)  
*# C:/Users/Basir Payenda\Desktop*

*# basename(), dirname(), split(), exists()*print(os.path.basename("/Basir Payenda/python.docx")) *# python.docx*print(os.path.dirname("/Basir Payenda/python.docx")) *# Basir Payenda*print(os.path.split("/Basir Payenda/python.docx"))

*# ('/Basir Payenda', 'python.docx')*print(os.path.exists("/Basir Payenda/python.docx")) *# False*

*# To access environment variables*

os.environ.get('Home')

*# IF NO EXTENSION, TO CHECK IF IT IS FILE OR DIRECTORY, IT CHECKS*

*# EXISTENCE OF THEM EITHER*print(os.path.isfile("C:/Users/Basir Payenda/Desktop/new.txt"))

*# True*

print(os.path.isdir("C:/Users/Basir Payenda/Desktop"))

*# True*

print(os.path.splitext("/temp/py.txt"))  
*# ('/temp/py', '.txt')*

*# TO CHECK EVERYTHING THAT IS AVAILABLE IN os.path MODULE*print(dir(os.path))

*# dirpath, dirnames, filenames*

*# basename(), dirname(), split(), exists()*

*# st\_mtime, splitext()*

*# To delete a file from a directory*

os.unlink('source\_fol/eggs.txt') *# deletes eggs.txt*

**Practice:**

1. Get the current directory
2. Change the current directory
3. Preview the content of current directory
4. Make directory and nested directories
5. Remove directory and nested directories
6. Rename a file or a directory
7. Get all related information of a directory
   * Get the modified date of a file and change it to a human readable format
8. See the master and branch areas of a directory along with its files
9. Access environment variables
10. Concatenate two path
11. Get the base name of a path, name of folder
12. Split a path to its formation parts
13. Check if it is a file or a folder and split the name and extension of a file

# Automate The Boring Stuff with Python

import os

path = 'C:\\new\py\os\os\_tuts.py'

# os.path.join() builds a path that works in all OS

path = os.path.join('', 'new', 'py', 'os', 'os\_tuts.py')

print(path) # C:new\py\os\os\_tuts.py

# mkdirs

os.mkdir('new\_dir')

# getcwd(), chdir()

print(os.getcwd()) # c:\py\files

os.chdir('C:\\Users')

print(os.getcwd()) # C:\Users

# basename and dirname

path = 'C:\\new\py\os\os\_tuts.py'

basename = os.path.basename(path) # C:\new\py\os

dirname = os.path.dirname(path) # os\_tuts.py

# abspath, isabs, relpath

path = 'C:\\new\py\os\os\_tuts.py'

res = os.path.abspath('os\_mod.py')

print(res) # c:\py\others\os\_mod.py

relpath = os.path.relpath(path, 'c:\\new\py\json')

print(relpath)

# os.path.sep

sep\_path = path.split(os.path.sep)

print(sep\_path) # ['C:new', 'py', 'os', 'os\_tuts.py']

# os.path.getsize()

fs = os.path.getsize(os.path.join(os.getcwd(), 'files.py'))

print(fs) # 539

# Examples

filesize = 0

for filename in os.listdir('C:\\Windows\\System32'):

filesize = filesize + os.path.getsize(os.path.join('C:\\Windows\\System32', filename))

print(filesize) # 2159434078

# Example: Delete all txt files within a current working directory

for filename in os.listdir():

if filename.endswith('.txt'):

os.unlink(filename)

print(filename)