

Python OS Module

- **import os**

- **print(os.getcwd())**

Returns the current working directory, then copy and paste that path to the the os.chdir

- **os.chdir('/Users/coreshafer/Desktop/')**

- **print(os.getcwd())**

The change directory is verified through the code above.

- **print(os.listdir())**

Prints all the files in the current working directory, in this case the desktop.

- **os.mkdir('OS-Demo-2')**

- **os.makedirs('OS-Demo-2/Sub-Dir-1')**

The difference between 'mkdir' & 'makedirs' is simple. mkdir can create the initial directory, where makedirs can make multiple level directories.

- **os.rmdir('OS-Demo-2/Sub-Dir-1')**

- **os.removedirs('OS-Demo-2/Sub-Dir-1')**

The way of deleting directories is similar to creating. 'Os.rmdir' removes that specific directory, where os.removedirs removes multiple level directories. Use it with caution.

- **os.rename('test.txt', 'demo.txt')**

Renames test.txt to demo.txt in the same directory.

- **`print(os.stat('demo.txt'))`**

```
os.stat_result(st_mode=33188, st_ino=40275850, st_dev=16777224, st_nlink=1, st_uid=501, st_gid=20, st_size=20,
st_atime=1459819317, st_mtime=1459819315, st_ctime=1459819315)
[Finished in 0.0s]
```

`st_size` correlates with the size of the file. (20 bytes). The `st_mtime` shows when the file was last modified, but it returns a timestamp. All of these specific pieces can be printed out individually as follows.

- **`print(os.stat('demo.txt').st_mtime)`**

Returns as 1459819315.0 and we need to get it into a human readable format. Therefore we import the `datetime` module, and save the timestamp as a variable which is then called on using `datetime`.

```
1 import os
2 from datetime import datetime
3
4 os.chdir('/Users/coreyschafer/Desktop/')
5
6 mod_time = os.stat('demo.txt').st_mtime
7 print(datetime.fromtimestamp(mod_time))
```

- Check if a path exists with the following code

`print(os.path.exists('/tmp/test.txt'))`

Will return True or False.

- **`print(os.path.isdir('/tmp/test.txt'))`**

Returns true if it is a directory

- **`print(os.path.isfile('/tmp/test.txt'))`**

Returns true if it is a file

- **`print(dir(os.path))`**

Prints out all available methods in the OS module.

- Traverse the directory tree and print all directories and files within the directory. `os.walk` method is a generator that yields a tuple of 3 values as it walks the directory tree. This is useful if you forgot where you saved a file.

```
1
2 import os
3 from datetime import datetime
4
5 os.chdir('/Users/coreyschafer/Desktop/')
6
7 for dirpath, dirnames, filenames in os.walk('/Users/coreyschafer/Desktop/'):
8     print('Current Path:', dirpath)
9     print('Directories:', dirnames)
10    print('Files:', filenames)
11    print()
12
```

I

Current Path: /Users/coreyschafer/Desktop/
Directories: ['Demo-Folder', 'Module-OS', 'OS-Demo', 'Screenshots', 'Videos']
Files: ['demo.txt']

Current Path: /Users/coreyschafer/Desktop/Demo-Folder
Directories: ['Sub-Dir1', 'Sub-Dir2']
Files: ['demo-folder-file-1.txt']

Current Path: /Users/coreyschafer/Desktop/Demo-Folder/Sub-Dir1
Directories: []
Files: ['sub-dir1-file.txt']

Current Path: /Users/coreyschafer/Desktop/Demo-Folder/Sub-Dir2
Directories: []

- `print(os.environ.get('HOME'))`
- `File_path = os.path.join(os.environ.get('HOME'), 'test.txt')`
- output - `/Users/coreyschafer/test.txt`

This code brings together two paths without having to worry about slashes.