AUTOMATE FILE SEGREGATION 1

What is our GOAL for this MODULE?

In this class, we learned about the **os** and **shutil** modules of **Python.**

What did we ACHIEVE in the class TODAY?

- Understood the importance of **os** and **shutil** modules

Which CONCEPTS/ CODING BLOCKS did we cover today?

- The **pip install**
- The **os, shutil** modules
- The **while** loop

How did we DO the activities?

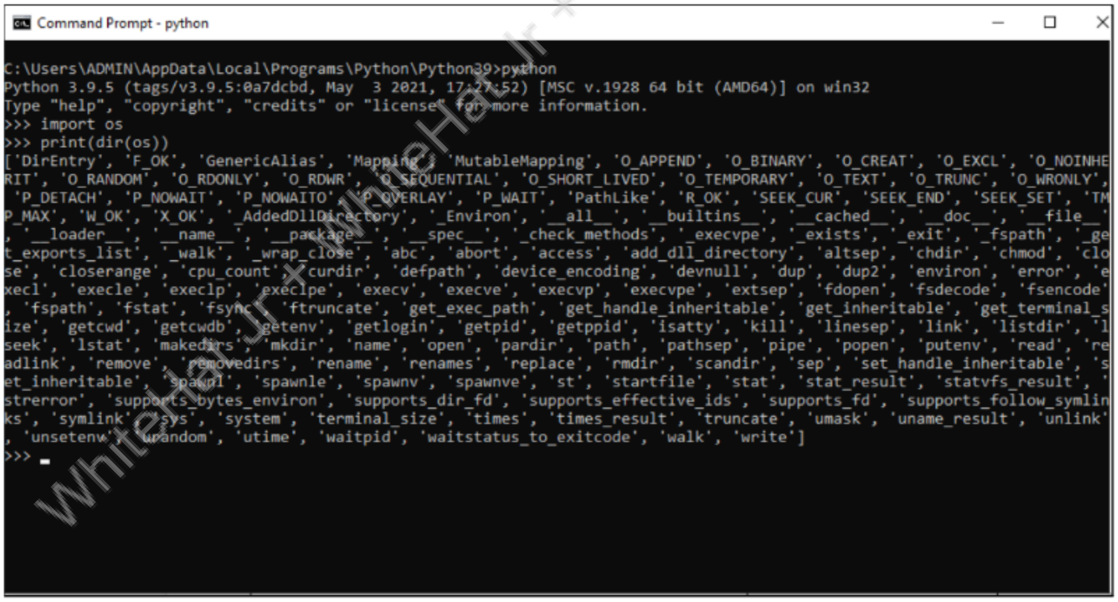
1. **OS module**: Our operating system provides us with different functionalities like moving files, copying files, and so on. The **os** module in Python provides us a way of using these operating system-dependent functionalities.

    - Import **os** module.

```
>>> import os
```

    - We can use **dir()** on the os module to see the methods of the os module.

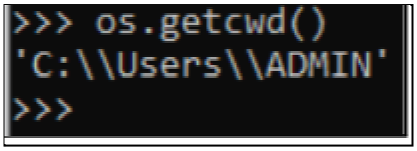
```
>>> print(dir(os))
```



```
C:\Users\ADMIN\AppData\Local\Programs\Python\Python39>python
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> print(dir(os))
['DirEntry', 'F_OK', 'GenericAlias', 'Mapping', 'MutableMapping', 'O_APPEND', 'O_BINARY', 'O_CREAT', 'O_EXCL', 'O_NOINHE
RIT', 'O_RANDOM', 'O_RDONLY', 'O_RDWR', 'O_SEQUENTIAL', 'O_SHORT_LIVED', 'O_TEMPORARY', 'O_TEXT', 'O_TRUNC', 'O_WRONLY',
 'P_DETACH', 'P_NOWAIT', 'P_NOWAITO', 'P_OVERLAY', 'P_WAIT', 'PathLike', 'R_OK', 'SEEK_CUR', 'SEEK_END', 'SEEK_SET', 'TM
P_MAX', 'W_OK', 'X_OK', '_AddedDllDirectory', '_Environ', '__all__', '__builtins__', '__cached__', '__doc__', '__file__'
, '__loader__', '__name__', '__package__', '__spec__', '_check_methods', '_execvpe', '_exists', '_exit', '_fspath', '_ge
t_exports_list', '_walk', '_wrap_close', 'abc', 'abort', 'access', 'add_dll_directory', 'altsep', 'chdir', 'chmod', 'clo
se', 'closerange', 'cpu_count', 'curdir', 'defpath', 'device_encoding', 'devnull', 'dup', 'dup2', 'environ', 'error', 'e
xecl', 'execle', 'execlp', 'execlpe', 'execv', 'execve', 'execvp', 'execvpe', 'extsep', 'fdopen', 'fsdecode', 'fsencode'
, 'fspath', 'fstat', 'fsync', 'ftruncate', 'get_exec_path', 'get_handle_inheritable', 'get_inheritable', 'get_terminal_s
ize', 'getcwd', 'getcwdb', 'getenv', 'getlogin', 'getpid', 'getppid', 'isatty', 'kill', 'linesep', 'link', 'listdir', 'l
seek', 'lstat', 'makedirs', 'mkdir', 'name', 'open', 'pardir', 'path', 'pathsep', 'pipe', 'popen', 'putenv', 'read', 're
adlink', 'remove', 'removedirs', 'rename', 'renames', 'replace', 'rmdir', 'scandir', 'sep', 'set_handle_inheritable', 's
et_inheritable', 'spawnl', 'spawnle', 'spawnv', 'spawnve', 'st', 'startfile', 'stat', 'stat_result', 'statvfs_result', '
strerror', 'supports_bytes_environ', 'supports_dir_fd', 'supports_effective_ids', 'supports_fd', 'supports_follow_symlin
ks', 'symlink', 'sys', 'system', 'terminal_size', 'times', 'times_result', 'truncate', 'umask', 'uname_result', 'unlink'
, 'unsetenv', 'urandom', 'utime', 'waitpid', 'waitstatus_to_exitcode', 'walk', 'write']
>>>
```

    - Check our current working directory by using **getcwd()** in the **os** module.

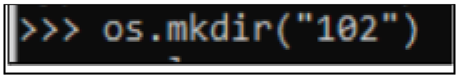
```
>>> os.getcwd()
```



```
>>> os.getcwd()
'C:\\Users\\ADMIN'
>>>
```

- The **mkdir()** method in the os module is used to create new files or folders.

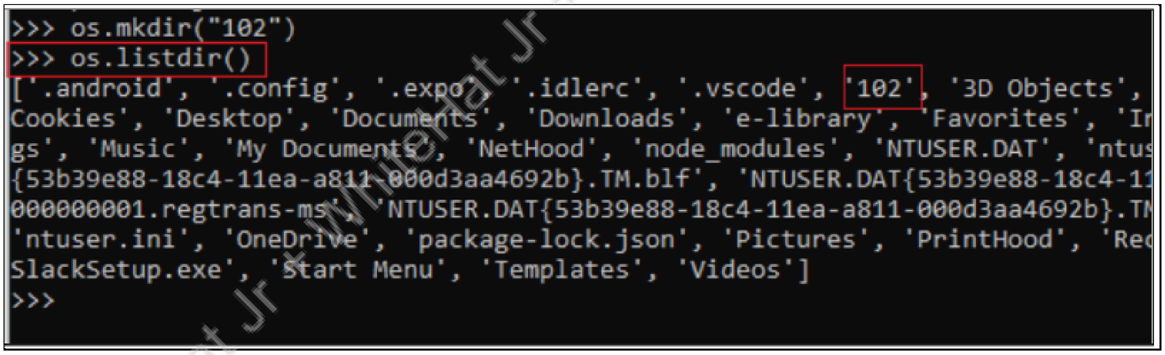
```
os.mkdir("name of the file/folder to create")
```



```
>>> os.mkdir("102")
```

- To get all the files and folders of the current directory, we can use the **listdir()** method in the **os** module.

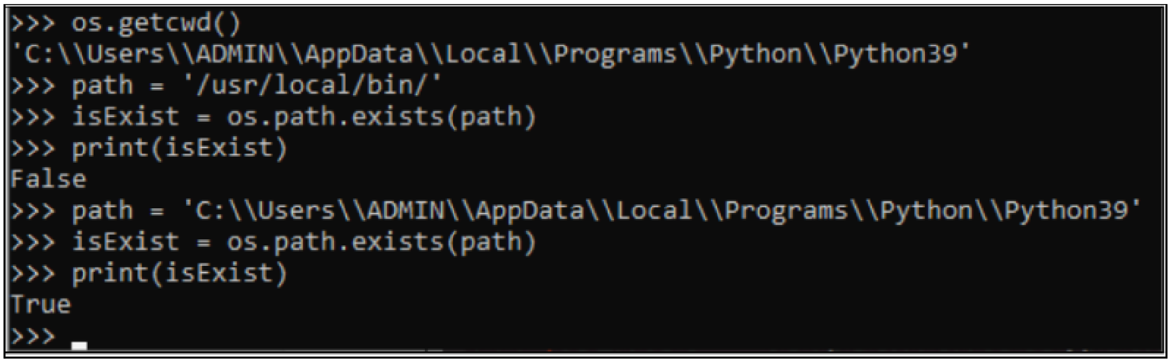
```
>>> os.listdir()
```



```
>>> os.mkdir("102")
>>> os.listdir()
['.android', '.config', '.expo', '.idlerc', '.vscode', '102', '3D Objects',
Cookies', 'Desktop', 'Documents', 'Downloads', 'e-library', 'Favorites', 'In
gs', 'Music', 'My Documents', 'NetHood', 'node_modules', 'NTUSER.DAT', 'ntus
{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM.blf', 'NTUSER.DAT{53b39e88-18c4-11
000000001.regtrans-ms', 'NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM
'ntuser.ini', 'OneDrive', 'package-lock.json', 'Pictures', 'PrintHood', 'Rec
SlackSetup.exe', 'Start Menu', 'Templates', 'Videos']
>>>
```

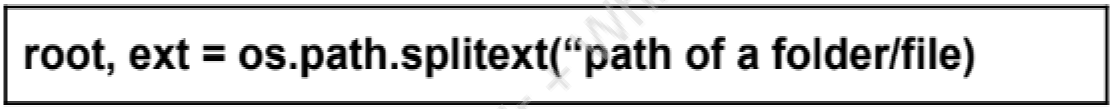
- To check if a particular file or folder is present in a given folder path, we can use the **exists()** method in the **os.path** module.

```
os.path.exists("path of a folder/file)
```

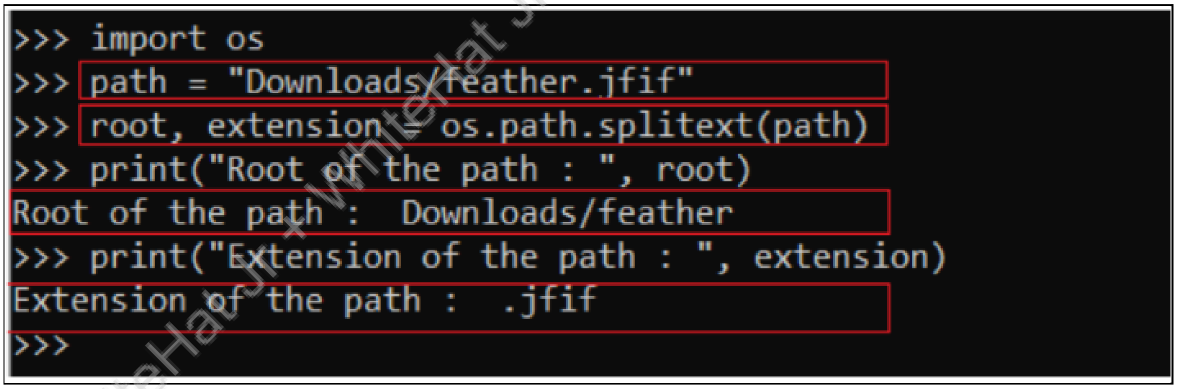


```
>>> os.getcwd()
'C:\\Users\\ADMIN\\AppData\\Local\\Programs\\Python\\Python39'
>>> path = '/usr/local/bin/'
>>> isExist = os.path.exists(path)
>>> print(isExist)
False
>>> path = 'C:\\Users\\ADMIN\\AppData\\Local\\Programs\\Python\\Python39'
>>> isExist = os.path.exists(path)
>>> print(isExist)
True
>>> _
```

- The **os.path.splitext()** method in **Python** is used to split the path name into the **(root, ext)** pair where **root** is the whole path except the extension and **ext** is the file extension with a dot.



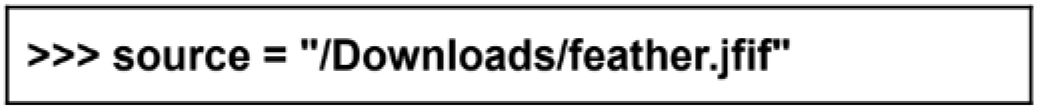
```
root, ext = os.path.splitext(“path of a folder/file)
```



```
>>> import os
>>> path = "Downloads/feather.jfif"
>>> root, extension = os.path.splitext(path)
>>> print("Root of the path : ", root)
Root of the path :  Downloads/feather
>>> print("Extension of the path : ", extension)
Extension of the path :  .jfif
>>>
```

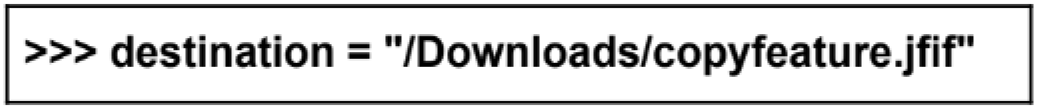
2. The **shutil** module: This helps us in automating the process of copying and removal of files and directories.
   Steps to **copy** a file:

   - Take a **source** variable and assign a file path.



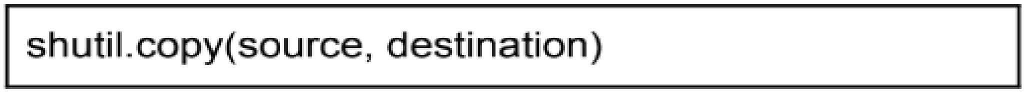
```
>>> source = "/Downloads/feather.jfif"
```

   - Take a **destination** variable, assign a path where we want to create a copy.

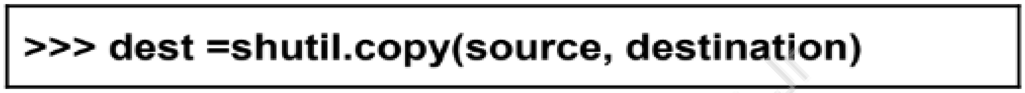


```
>>> destination = "/Downloads/copyfeature.jfif"
```

- Use the **copy()** method in the **shutil** module to copy the file from source to destination.

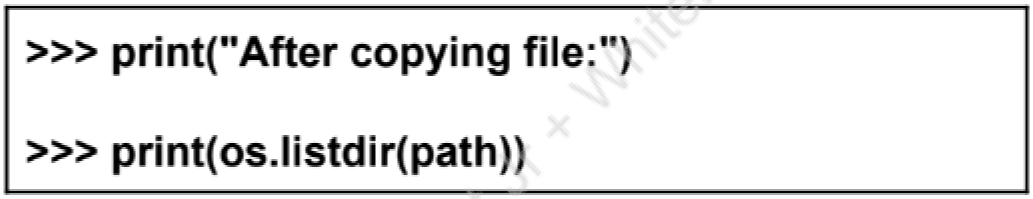


```
shutil.copy(source, destination)
```



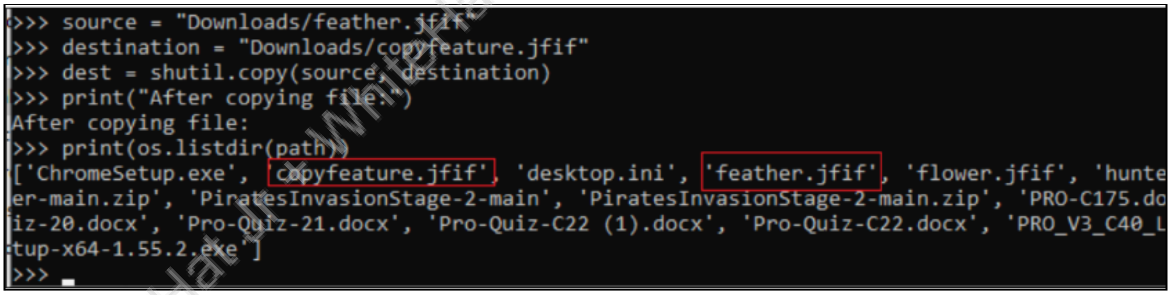
```
>>> dest =shutil.copy(source, destination)
```

- Print the content of the folder to see if the copy is created.



```
>>> print("After copying file:")

>>> print(os.listdir(path))
```



```
>>> source = "Downloads/feather.jfif"
>>> destination = "Downloads/copyfeature.jfif"
>>> dest = shutil.copy(source, destination)
>>> print("After copying file:")
After copying file:
>>> print(os.listdir(path))
['ChromeSetup.exe', 'copyfeature.jfif', 'desktop.ini', 'feather.jfif', 'flower.jfif', 'hunte
er-main.zip', 'PiratesInvasionStage-2-main', 'PiratesInvasionStage-2-main.zip', 'PRO-C175.do
iz-20.docx', 'Pro-Quiz-21.docx', 'Pro-Quiz-C22 (1).docx', 'Pro-Quiz-C22.docx', 'PRO_V3_C40_L
tup-x64-1.55.2.exe']
>>> _
```

- There are other methods of **shutil()** to move the files or remove files altogether.
- The **shutil.move()** to move a file from source to destination.

What's next?

In the next class, we are going to continue by writing a code using the **os** and **shutil** module to move the image files from the download folder to a different folder.

EXTEND YOUR KNOWLEDGE:

Learn more about the **os** module here