

In []:

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

posix

In []:

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

/content

In []:

```
#3.os.listdir('.')-To print files and directories in the current directory on your system
import os
print(os.listdir('.'))
```

['.config', 'sample_data']

In []:

```
#os.chdir('.')-This function is used to change the CWD
import os
print(os.getcwd())
os.chdir('.')
print(os.getcwd())
```

/content

/

In []:

```
#makedirectory-mkdir
os.mkdir("/content/python")
print(os.listdir('.'))
```

['media', 'srv', 'var', 'mnt', 'etc', 'root', 'tmp', 'lib', 'proc', 'sbin', 'lib64', 'boot', 'run', 'sys', 'bin', 'dev', 'usr', 'opt', 'home', 'content', 'python', '.dockerenv', 'tools', 'datalab', 'lib32', 'python-apt', 'NGC-DL-CONTAINER-LICENSE']

In []:

```
#remove directory
os.rmdir("/content/python")
```

In []:

```
#remove a file
os.remove(path)
```

In []:

```
#remove a file
os.remove('s.py')
```

```
-----
-
FileNotFoundError                                Traceback (most recent call las
t)
```

```
<ipython-input-22-b33dfde451db> in <module>
```

```
    1 #remove a file
----> 2 os.remove('s.py')
```

```
FileNotFoundError: [Errno 2] No such file or directory: 's.py'
```

In []:

```
os.rename(old.new)
```

In []:

```
os.mkdir('/content/remya')
os.chdir('\content\remya')
for (root,dirs,files) in os.walk('.', topdown=True):
    print(root )
    print(dirs )
    print(files)
    print('-----')
```

```
-----
-
FileExistsError                                Traceback (most recent call las
t)
```

```
<ipython-input-31-6021539ce6ee> in <module>
```

```
----> 1 os.mkdir('/content/remya')
      2 os.chdir('\content\remya')
      3 for (root,dirs,files) in os.walk('.', topdown=True):
      4     print(root )
      5     print(dirs )
```

```
FileExistsError: [Errno 17] File exists: '/content/remya'
```

In []:

```
import os
os.mkdir('/content/hello2')
```

In []:

```
#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.
```

```
import os  
os.chdir('/content/hello2')  
for (root,dirs,files) in os.walk('.', topdown=True):  
    print(root )  
    print(dirs )  
    print(files)  
    print('-----')
```

```
.  
['.ipynb_checkpoints', 'folder1']  
['pgm1.py']  
-----  
./ipynb_checkpoints  
[]  
[]  
-----  
./folder1  
[]  
[]  
-----
```

In []:

```
os.path.isfile("/content/hello2/pgm1.py")
```

Out[]:

True

In []:

```
os.path.isdir("/content/hello2")
```

Out[]:

True

In []:

```
os.path.getsize("/content/hello2/pgm1.py")
```

Out[]:

15

In []:

```
os.path.exists("/content/hello2/pgm1.py")
```

Out[]:

True

In []:

```
#The following program will count the number of  
#files in the current directory.
```

```
import os
def countfiles(path):
    count=0
    lst=os.listdir(path)
    for f in lst:
        if os.path.isfile(f):
            count=count+1
        else:
            os.chdir(f)
            count=count+countfiles(os.getcwd())
            os.chdir('.')
    return count

c=countfiles('.')
print("number of files...=",c)
```

number of files...= 1

In []:

```
#Print the names of the files in the current directory having ".py" extension.
```

```
import os
path=os.getcwd()
lst=os.listdir(path)
for f in lst:
    if '.py' in f:
        print(f)
```

pgm1.py

In []:

```
#The current version number of Python
```

```
import sys
print(sys.version)
```

3.7.13 (default, Apr 24 2022, 01:04:09)
[GCC 7.5.0]

In []:

```
import sys
print(sys.version)
print(sys.version_info)
#search path for all Python modules
print(sys.path)
```

3.7.13 (default, Apr 24 2022, 01:04:09)
[GCC 7.5.0]
sys.version_info(major=3, minor=7, micro=13, releaselevel='final', serial=0)
['/content', '/env/python', '/usr/lib/python37.zip', '/usr/lib/python3.7',
'/usr/lib/python3.7/lib-dynload', '', '/usr/local/lib/python3.7/dist-packa
ges', '/usr/lib/python3/dist-packages', '/usr/local/lib/python3.7/dist-pac
kages/IPython/extensions', '/root/.ipython']

In []:

```
#This is generally used to safely exit from  
#the program in case of generation of an exception.  
import sys  
sys.exit()
```

An exception has occurred, use %tb to see the full traceback.

SystemExit

In []:

```
#Name of the platform on which Python is running, e.g. "linux2" for Linux  
#and "win32" for Windows  
print(sys.platform)  
#A string containing the name of the executable binary (path and executable file name)  
for the Python interpreter.  
print(sys.executable)
```

```
linux  
/usr/bin/python3
```

In []: