working with os module:
----
        the 'os' module is a builtin module in python.

        if we want to display all the properties of 'os' module,by using following command,

        import os
        print(dir(os))

ex1:
--
wap to get the current working directory?

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

output:

C:\Python310

ex2:
---
wap to make a directory?

```
import os
os.mkdir('entriapp')
print("directory created successfully")
```

output:
-----
directory created successfully

ex3:
---
wap to make a directory on different location?

```
import os
os.mkdir('E:\\krishna\\entri')
print("directory created successfully")
```

output:
------
directory created successfully

ex4:
----
wap to change the directory?

```
import os
```

```
print(os.getcwd())
os.chdir('entriapp')
print(os.getcwd())

output:
------
C:\Python310
C:\Python310\entriapp

ex5:
----
wap to create a directories by using iteration?

import os
os.chdir('entriapp')
for d in ['Vagdevi','python','django']:
    os.mkdir(d)
print("Directories are created successfully")

output:
------
Directories are created successfully

ex6:
----
wap to create a files?

import os
os.chdir('entriapp')
f1=open('sample.txt','w')
f1.close()
f2=open('demo.py','w')
f2.close()
f3=open('welcome.html','w')
f3.close()
f4=open('base.css','w')
f4.close()
f5=open('test.java','w')
f5.close()
f6=open('addition.py','w')
f6.close()
print("Files are created successfully")

output:
-----
Files are created successfully

ex7:
---
wap to display list of all the files and directories in the given specified
```

location?

```
import os
os.chdir('entriapp')
print(os.listdir())
print('*'*20)
print(os.listdir('E:\\krishna'))
```

output:

```
['addition.py', 'base.css', 'demo.py', 'django', 'python', 'sample.txt',
'test.java', 'Vagdevi', 'welcome.html']
********************
['demo4.txt', 'Divyakola Siva Krishna_329597.pdf', 'entri', 'Guido', 'iglobal',
'mybirths.csv', 'mydemo.txt', 'mysample.txt', 'mysimple.txt', 'rama', 'sachin',
'sample3.txt', 'sample4.txt', 'simple3.txt', 'siva', 'vagpython']
```

ex8:
---
wap to return directories from the given specified location?

```
import os
os.chdir('entriapp')
for i in os.listdir():
    if os.path.isdir(i):
        print(i)
```

output:

```
django
python
Vagdevi
```

ex9:
---
wap to return file's from the given specified location?

```
import os
os.chdir('entriapp')
for i in os.listdir():
    if os.path.isfile(i):
        print(i)
```

output:

```
addition.py
base.css
demo.py
sample.txt
test.java
```

welcome.html

ex10:
----
wap to return the .py extension files from the given specific location?

```
import os
os.chdir('entriapp')
for i in os.listdir():
    if os.path.isfile(i):
        if i.endswith('.py'):
            print(i)
```

output:
------
addition.py
demo.py

ex11:
----
wap to rename the files and directories?

```
import os
os.chdir('entriapp')
print(os.listdir())
print('*'*20)
os.rename('addition.py','add.py')
os.rename('test.java','test.py')
os.rename('django','dj')
print(os.listdir())
```

output:

['addition.py', 'base.css', 'demo.py', 'django', 'python', 'sample.txt',
'test.java', 'Vagdevi', 'welcome.html']
****
['add.py', 'base.css', 'demo.py', 'dj', 'python', 'sample.txt', 'test.py',
'Vagdevi', 'welcome.html']

ex12:
----
wap to return the no.of item's in the cureent working directory/specified location?

```
import os
os.chdir('entriapp')
print(len(os.listdir()))
```

output:

9

ex13:
----
wap to return the no.of files and directories in the current working location or given specified location?

```
import os
os.chdir('entriapp')
f_c,d_c=0,0
for i in os.listdir():
    if os.path.isdir(i):
        d_c+=1
    else:
        f_c+=1
print("No.Of File's: ",f_c)
print("No.Of Dir's: ",d_c)
```

output:

```
No.Of File's:  6
No.Of Dir's:  3
```

ex14:
----
wap to remove the dir's and file's from the given specified location/current working directory?

```
import os
os.chdir('entriapp')
print(os.listdir())
os.remove('add.py')
os.rmdir('dj')
print(os.listdir())
```

output:

```
['add.py', 'base.css', 'demo.py', 'dj', 'python', 'sample.txt', 'test.py', 'Vagdevi', 'welcome.html']

['base.css', 'demo.py', 'python', 'sample.txt', 'test.py', 'Vagdevi', 'welcome.html']
```

ex15:
---
wap to get login details?

```
import os
print(os.getlogin())
```

output:

```
------
DELL

ex16:
----
wap to get the os module path?

import os
print(os.path)

output:
------
<module 'ntpath' from 'C:\\Python310\\lib\\ntpath.py'>

ex17:
----
wap to get the os name?

import os
print(os.name)

output:
------
nt

ex18:
---
wap to get the processId and parent processID?

import os
print(os.getpid())
print(os.getppid())

output:
-------
21032
16644

ex19:
---
wap to return the list directories,sub-dir's and files in the given
specified/current location?

        C:\\Python310
              |--->entriapp
              |       |--->python
              |       |       |--->django
              |       |       |       |--->demo.py
              |       |       |--->flask
              |       |       |--->test.py
```

```
              |       |--->vagdevi
              |       |--->base.css
              |       |--->sample.txt
              |       |--->welcome.html
```

```
import os
for rootdir,subdir,file in os.walk('entriapp'):
    print(rootdir,subdir,file)
```

```
output:
------
entriapp ['python', 'Vagdevi'] ['base.css', 'sample.txt', 'welcome.html']
entriapp\python ['django', 'flask'] ['test.py']
entriapp\python\django [] ['demo.py']
entriapp\python\flask [] []
entriapp\Vagdevi [] []
```

ex20:
----
wap to kill the process?

we can kill the process by using kill() of os module.

```
import os
os.kill(processID,signal)
```

```
import os
os.kill(os.getpid(),1)
```

```
import os
os.kill(os.getppid(),1)
```

working with sys module
----
the 'sys' module is builtin module in python.

if we want to display all the properties of sys module by using following commands,

```
import sys
print(dir(sys))
```

ex1:

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

```
3.10.1 (tags/v3.10.1:2cd268a, Dec  6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)]

ex2:
-----
import sys
print(sys.version_info)

sys.version_info(major=3, minor=10, micro=1, releaselevel='final', serial=0)

ex3:
----
import sys
print(sys.path)

['', 'C:\\Python310\\Lib\\idlelib', 'C:\\Python310\\python310.zip',
'C:\\Python310\\DLLs', 'C:\\Python310\\lib', 'C:\\Python310',
'C:\\Python310\\lib\\site-packages']

ex4:
----
import sys
print(sys.modules)

{
'sys': <module 'sys' (built-in)>,
'builtins': <module 'builtins' (built-in)>,
'_frozen_importlib': <module '_frozen_importlib' (frozen)>,
'_imp': <module '_imp' (built-in)>,
'_thread': <module '_thread' (built-in)>,
.....
.....
}

ex5:
---
import sys
print(sys.copyright)

Copyright (c) 2001-2021 Python Software Foundation.
All Rights Reserved.

Copyright (c) 2000 BeOpen.com.
All Rights Reserved.

Copyright (c) 1995-2001 Corporation for National Research Initiatives.
All Rights Reserved.

Copyright (c) 1991-1995 Stichting Mathematisch Centrum, Amsterdam.
All Rights Reserved.
```

ex6:

```
import sys
print(sys.platform)
```

```
win32
```

ex7:

```
import sys
print(sys.getwindowsversion())
```

```
sys.getwindowsversion(major=10, minor=0, build=19042, platform=2, service_pack='')
```

ex8:

```
import sys
print(sys.getsizeof(5))
28

print(sys.getsizeof(3.2))
24

print(sys.getsizeof([5,3,7,2]))
120
```

ex9:

```
import sys
class test:
    def m1(self):
        pass
t1=test()
print(sys.getrefcount(t1))
t2=t1
print(sys.getrefcount(t1))
t3=t2
print(sys.getrefcount(t1))
t4=test()
print(sys.getrefcount(t4))
```

ex10:
----
wap to exit from program execution at time middle of the program execution?

by using exit() of sys module

if we want to exit from the loop execution at the time middle of the loop execution by using break statement.

```
import sys
i=0
while True:
    print("hai")
    if i==3:
        sys.exit()
    i+=1
print("bye")
```

output:
-----
hai
hai
hai
hai

command line arguments:
----

the programmer or developer to assign the values/passing the arguments to the python file through the command prompt at runtime,that type of arguments are called command-line arguments.

in python we can implement the command line arguments by using argv variable of sys module.

in python,by default our file name also act as a one commandline argument.

the argv variable to return the output as list object.

by default every command line argument treated as a string object.

ex1:
----
wap to print list of command line arguments?

```
import sys
print(sys.argv)
```

output:

```
C:\Users\DELL\Desktop>python demo.py hai 2 4.5 3j
['demo.py', 'hai', '2', '4.5', '3j']
```

ex2:
----
wap to perform addition of two numbers by using command line arguments?

```
import sys
a=int(sys.argv[1])
b=int(sys.argv[2])
```

```
print("the sum of %d and %d is:%d"%(a,b,a+b))

output:
------
C:\Users\DELL\Desktop>python demo.py 4 5
the sum of 4 and 5 is:9
```