os.listdrives()
Return a list containing the names of drives on a Windows system.

```
>>> import os
>>> list1=os.listdrives()
>>> print(list1)
['C:\\', 'E:\\']
```

os.remove(*path*)
Remove (delete) the file *path*. If *path* is a directory, an OSError is raised.

```
# Program to remove file

import os
import os.path
name=input("Enter FileName ")
if os.path.exists(name):
    os.remove(name)
    print("file deleted...")
else:
    print("File not found")
```

Output

```
Enter FileName e:\\app.txt
File not found

Enter FileName e:\\app.log
file deleted...
```

os.rename(src, dst)

Rename the file or directory src to dst. If dst exists, the operation will fail with an OSError.

```
# Program to rename file
import os
import os.path

old_fname=input("Enter Old FileName ")
new_fname=input("Enter new FileName ")
if os.path.exists(old_fname):
    os.rename(old_fname,new_fname)
    print("renamed...")
else:
    print("old filename not exists")
```

Output

```
Enter Old FileName e:\\file1.txt
Enter new FileName e:\\new_file1.txt
renamed...

Enter Old FileName e:\\workshop
Enter new FileName e:\\python_workshop
renamed...
```

shutil.copyfile(src, dst)

Copy the contents (no metadata) of the file named src to a file named dst and return dst in the most efficient way possible. src and dst are path-like objects or path names given as strings.

Example:

```
# Creating copy of the file
```

```
import shutil
import os.path

src_file=input("Soruce FileName ")
dst_file=input("Dest FileName ")
shutil.copy(src_file,dst_file)
print("copied...")
```

Output

```
Soruce FileName e:\\error.log
Dest FileName e:\\error1.log
copied...

Soruce FileName e:\\error.log
Dest FileName e:\\python_workshop
copied...
```

os.walk(top, topdown=True)

Generate the file names in a directory tree by walking the tree either top-down or bottom-up. For each directory in the tree rooted at directory top (including top itself), it yields a 3-tuple (dirpath, dirnames, filenames).

```
import os

a=os.walk(".")
for name in a:
    print(name)

b=os.walk("e:\\",topdown=False)
for name in b:
    print(name)
```

Output
Display complete tree structure of the directory and sub directories including files.

os.system(*command*)
Execute the command (a string) in a subshell.

```
# Program to run external commands or programs

import os

print("1. Notepad ")
print("2. Paint")
print("3. Calculator ")
opt=int(input("Enter your option "))
if opt==1:
    os.system("notepad")
elif opt==2:
    os.system("mspaint")
elif opt==3:
    os.system("calc")
```

Output

```
1. Notepad
2. Paint
3. Calculator
Enter your option 3
```

How to shutdown, restart and logout operating system?

```
os.system("shutdown /s /t 0")
os.system("shutdown /r /t 0")
os.system("shutdown /l /t 10")
```

/s shutdown
/r restart
/l logout
/t time

datetime and calendar modules

“datetime” module provides the classes or data types to work with date and time data.
“datetime” is a default module which comes with python software.
“datetime” module provides the following classes or data types to work with date and time

1. date
2. time
3. datetime
4. timedelta

date and time data types are immutable.

date class or data type

date class represents date object.
Date object is created with 3 attributes
1. day
2. month
3. year

***class* datetime.date(*year*, *month*, *day*)**
All arguments are required. Arguments must be integers, in the following ranges:
MINYEAR <= year <= MAXYEAR
1 <= month <= 12
1 <= day <= number of days in the given month and year
If an argument outside those ranges is given, ValueError is raised.

Example:
```
import datetime

d1=datetime.date(2024,3,16)
print(d1)
print(d1.day)
print(d1.month)
print(d1.year)
```

Output
```
2024-03-16
16
3
2024
```

classmethod date.today()
Return the current local date.

```
>>> import datetime
>>> d2=datetime.date.today()
>>> print(d2)
2024-03-16
```

classmethod date.fromisoformat(date_string)

Return a date corresponding to a *date_string*

```
from datetime import date
d1=date.fromisoformat('2019-12-04')
d2=date.fromisoformat('20191204')
print(d1,d2,sep="\n")
```

Output

```
2019-12-04
2019-12-04
```