Files

Every application required to store or save data permanently. The data can be saved permanently using 2 systems

1. File System
2. Database System

What is file?

File is collection of data or information.
File is named memory location on secondary storage device or hard disk.
Files are used for storing or saving data permanently.

Types of files

1. Text File
2. Binary File

In text file data is stored in text format (OR) text file allows only string data. There are different types of text files

1. Txt
2. CSV (Comma Separated Values)
3. JSON (Java Script Object Notation)
4. XML (Extensible Markup Language)
5. HTML (Hyper Text Markup Language)

In Binary file data is stored in bytes format (OR) binary file allows only bytes data. There are different types of binary files

1. Images
2. Audio files
3. Video files
4. PDF
5. Document
6. PPT

Basic steps for working with files

1. Open File
2. Write/Read
3. Close File

How to open the file?

Python provides a predefined function called open()
This function opens the file and return corresponding file object.

Syntax: variable-name=open(filename,mode)

Filename → it is a string which represents file
Mode → it is a string which represents file opening mode

Modes	**Description**
w	Write – it opens the file in write mode. This mode create new file. If any file exists with given filename, it truncates and create new file.
r	Read – it opens the file in read mode. It allows reading data from existing file. If given filename not exists, it raises FileNotFoundError
a	Append – it opens the in append mode. If file exists it allows adding data to existing file. If file not exists it creates new file and allows to add/append
x	Exclusive Creation, it opens the file in write mode. If given filename exists, it raises FileExistsError. If given filename is not exists it creates new file for writing data
w+r	Write and Read
r+w	Read and Write (Update)
t	Text (default) -- wt,rt,at,xt,w,r,a,x
b	Binary -- wb, rb,ab,xb,wb+rb

Text files
Text file is a collection characters and text file allows only string data. The following functions or methods are used for writing strings inside text file.

1. write
2. writeline
3. print

write(s)
This method allows writing string inside file and after writing it returns length of string (count of characters).

Syntax: file-object-name.write(string)

Example:

```
import sys
try:
  fobj=open("file1.txt","w")
  fobj.write("Python")
  fobj.write("Java")
  fobj.write("Oracle")
  fobj.write(str(12))
except:
    t=sys.exc_info()
    print(t[1])
finally:
    fobj.close()
```

Output

Output is saved inside file1.txt, which is created in current working directory

writelines(list)

This method allows writing multiple strings.
These strings are represented within list. In order to write in new file each string must suffix with \n

Syntax: file-object-name.writelines(list)

Example:

```
import sys

try:
    fobj=open("file2.txt","a")
    data=["101\n","naresh\n","python\n","5000\n"]
    fobj.writelines(data)
    print("data is saved inside file2.txt")
except:
    t=sys.exc_info()
    print(t[1])
finally:
    fobj.close()
```

Output

data is saved inside file2.txt

Example:

```
# Write a program to store student marks details
# into marks.txt
import sys
try:
    fobj=open("marks.txt","a")
    while True:
        rno=int(input("Rollno :"))
        name=input("Name :")
        sub1=int(input("Subject1Marks :"))
        sub2=int(input("Subject2Marks :"))
        print(rno,name,sub1,sub2,file=fobj)
        ans=input("Add another student?")
        if ans=="no":
            break
    print("student marks details are saved inside marks.txt")
except:
    t=sys.exc_info()
    print(t[1])
finally:
    fobj.close()
```

Output

```
Rollno :1
Name :naresh
Subject1Marks :60
Subject2Marks :70
Add another student?yes
Rollno :2
Name :suresh
Subject1Marks :90
Subject2Marks :60
Add another student?yes
Rollno :3
Name :kishore
Subject1Marks :60
Subject2Marks :30
Add another student?yes
```

```
Rollno :4
Name :kiran
Subject1Marks :80
Subject2Marks :90
Add another student?yes
Rollno :5
Name :ramesh
Subject1Marks :20
Subject2Marks :50
Add another student?no
student marks details are saved inside marks.txt
```

Reading data from text file
The data can read from text file using the following methods

1. read()
2. readline()

read()
This method read size of characters from file and returns in one string.
If there is no characters read, it returns empty string.

Syntax: file-object.read(size=-1)

If size is –ve or not given, it reads complete file/all characters and return as one string.

Example:

```
# Write a program to read charactes from file1.txt

try:
    fobj=open("file1.txt","r")
    ch1=fobj.read(1)
    print(ch1)
    ch2=fobj.read(1)
    print(ch2)
    s1=fobj.read(4)
    print(s1)
    s2=fobj.read()
    print(s2)
```

```
except:
    print("Error")
finally:
    fobj.close()
```

Output

```
P
y
thon
JavaOracle12
```

Example:

```
# Write a program to find/count vowels in file1.txt

try:
    fobj=open("file1.txt","r")
    c=0
    while True:
        ch=fobj.read(1)
        if ch=='':
            break
        if ch in "aeiouAEIOU":
            c+=1
    print(f'Count of Vowels {c}')
except:
    print("Error")
finally:
    fobj.close()
```

Output

```
Count of Vowels 6
```

readline()

This method read one line from file (OR) it read all the characters until it found newline character \n

Syntax: file-object.readline(size=-1)

Size represents size characters but not lines

Example:

```
# Write a program to content of file2.txt

try:
    fobj=open("file2.txt","r")
    while True:
        s=fobj.readline()
        if s=='':
            break
        print(s,end='')
except:
    print("Error")
finally:
    fobj.close()
```

Output

```
101
naresh
python
5000
```

Example:

```
# Write a program to read marks details from marks.txt
# calculate total,avg and result
import sys
try:
    fobj=open("marks.txt","r")
    while True:
        stud=fobj.readline()
        if stud=='':
            break
        rno,name,sub1,sub2=stud.split()
        total=int(sub1)+int(sub2)
        avg=total/2
        result="PASS" if int(sub1)>=40 and int(sub2)>=40 else "FAIL"
        print(f'{rno}\t{name}\t{sub1}\t{sub2}\t{total}\t{avg:.2f}\t{result}')
except:
    t=sys.exc_info()
    print(t[1])
finally:
    fobj.close()
```

Output

```
1       naresh     60     70     130    65.00PASS
2       suresh     90     60     150    75.00PASS
3       kishore    60     30     90     45.00FAIL
4       kiran      80     90     170    85.00PASS
5       ramesh     20     50     70     35.00FAIL
```