

- A file is some information or data **which stays???** in the computer
- storage devices. We already know about different kinds of file, like music files, video files, text files, etc.

Introduction to file handling

- Files - Huge volume or Collection of data
- Types - Binary, Text, etc.
- Open any file before read/write.

Opening a File

file object = open(file name, [access mode],[buffering])

There are six methods

r – Read only (This mode opens the text files for reading only. It raises the I/O error if the file does not exist. This is the default mode for opening files).

w – Write only (This mode opens the file for writing only. The data in existing files are modified and overwritten. If the file does not already exist in the folder, a new one gets created.)

r+ – Read and Write (This method opens the file for both reading and writing. If the file does not exist, an I/O error gets raised.)

w+ – Write and Read (This mode opens the file for both reading and writing. The text is overwritten and deleted from an existing file.)

Opening a File

a- Append Only (This mode allows the file to be opened for writing. If the file doesn't yet exist, a new one gets created. The newly written data will be added at the end, following the previously written data.

a+ - Append and Read (Using this method, you can read and write in the file. If the file doesn't already exist, one gets created. The newly written text will be added at the end, following the previously written data.

Buffering: If the buffering value is set to 0, no buffering takes place. If the buffering value is 1, line buffering is performed while accessing a file. If you specify the buffering value as an integer greater than 1, then buffering action is performed with the indicated buffer size.

File Object Attributes

Attribute	Description
file.closed	Returns True if file is closed, False otherwise
file.mode	Returns Access mode with which file was opened.
file.name	Returns name of the file.

EXAMPLE

#procedure to Open a file

```
f1 = open("note.txt","r")  
print("Name of the file: ",f1.name  
) print("Closed or not: ", f1.close  
d) print("Opening mode: ", f1.mode)
```

Functions for file Handling

- The read functions contains different methods:
 - `read()` # return one big string
 - `readline()` #return one line at a time
 - `readlines()` #returns a **list** of lines
- This method writes a sequence of strings to the file.
 - `write ()` # Used to write a fixed sequence of characters to a file
 - `writelines()` # writelines can write a list of strings.
- The append function is used to append to the file instead of overwriting it.
- To append to an existing file, simply open the file in append mode ("a"):
- When you're done with a file, use `close()` to close it and free up any system resources taken up by the open file.

Functions Operation Examples

- To open a text file, use:
 - `f = open("demo.txt", "r")`
- To read a text file, use:
 - `f = open("demo.txt","r")`
 - `print (f.read())`
- To read one line at a time, use:
 - `f = open("hello.txt", "r")`
 - `print (f.readline())`
- To read a list of lines use:
 - `f = open("hello.txt.", "r")`
 - `print (f.readlines())`

Functions Operation- Read

```
f=open("hello.txt")  
for line in f:  
    print(line, end=' ')  
f.close()
```

With context manager

```
with open("hello.txt",'r') as f:  
    f1=f.read()           // readlines()  
    print(f1)
```

Also `f1=f.read(20)`

```
f=open("filesample.txt","r")
fcontent=f.read()
print(len(fcontent))
f.seek(0)
flines=f.readlines()
print(len(flines))
f.close()
```

Functions Operation - Write

```
f2=open("hello2.txt","w")
```

```
f2.write("hello2") // A text file by name hello2 will be created
```

```
f2.close()
```

Functions Operation Examples

- To write to a file, use:
 - `fh = open("hello.txt","w")`
 - `fh.write("Hello World")`
 - `fh.close()`
- To write to a file, use:
 - `fh = open("hello.txt", "w")`
 - `lines-of-text = ["a line of text", "another line of text", "a third line"]`
 - `fh.writelines(lines-of-text)`
 - `fh.close()`
- To append to file, use:
 - `fh = open("Hello.txt", "a")`
 - `fh.write("Hello World again")`
 - `fh.close ()`

**Functions Operation – Copying the contents
from one file and write to another file**

Functions Operation – Copying the contents from one file and write to another file

```
f1=open("note.txt","r+")  
s1=f1.read()  
f1.close()
```

```
f2=open("s2.txt","w+")  
f2.write(s1)  
f2.close()
```

```
print("File copied")
```

Playing Randomly in files

- `fileObject.tell()` -> current position within a file
- `fileObject.seek(offset [,from])` -> Move to new file position.
 - Argument offset is a byte count.
Optional argument whence defaults to 0 (offset from start of file, offset should be ≥ 0); other values are 1 (move relative to current position, positive or negative), and 2 (move relative to end of file, usually negative, although many platforms allow seeking beyond the end of a file)

Example for random seeking

```
fobj = open('/tmp/tempfile', 'w')
fobj.write('0123456789abcdef')
fobj.close()

fobj = open('/tmp/tempfile')
fobj.tell()      # tell us the offset position
0L

fobj.seek(5)      # Goto 5th byte
fobj.tell()
5L

fobj.read(1)      #Read 1 byte
'5'

fobj.seek(-3, 2)  # goto 3rd byte from the end
fobj.read()      # Read till the end of the file
'def'
```


Tips and Tricks makes it Easier ...

- Number of characters in a file is same as the length of its contents. `def charcount(filename):`
`return len(open(filename).read())`
- Number of words in a file can be found by splitting the contents of the file.
`def wordcount(filename):`
`return len(open(filename).read().split())`
- Number of lines in a file can be found from readlines method. `def linecount(filename):`
`return len(open(filename).readlines())`

Write a function to read a text file and print the longest line in it.

```
def input_stats(filename):  
    fid = open(filename)  
    l=fid.readlines()  
  
    longest = ""  
    for line in l:  
        if len(line) > len(longest):  
            longest = line  
  
    print("Longest line =", len(longest))  
    print(longest)  
  
input_stats("temp.txt")
```

Marks.csv file has the following data: Name, Mark1, Mark2 and Mark3 of 'n' no of students. Write a program to read the data from the file and calculate the total marks scored by each student. Then copy the content in another file called result.csv along with the total marks scored by each student.

 Markscsv - Notepad

File	Edit	Format	View	Help
Name, Maths, CSE, Physics				
ABBC, 100, 98, 89				
NNN, 87, 99, 66				
NNB, 67, 88, 99				

 Result - Notepad

File	Edit	Format	View	Help
Name, Maths, CSE, Physics, Total				
ABBC, 100, 98, 89, 287				
NNN, 87, 99, 66, 252				
NNB, 67, 88, 99, 254				

```
fp=open("Marks.csv","r+")
fw=open("Result.txt","w+")
fp.seek(0)
l=[]
line=fp.readline()
heading=line
heading=heading.replace("\n","")
fw.write(heading+",Total\n")
while True:
    line=fp.readline()
    if(not line):
        break
    line=line.replace("\n","")
    st=line.split(',')
    total=int(st[1])+int(st[2])+int(st[3])
    fw.write(line+", "+str(total)+"\n")
fp.close()
fw.close()
```