Files

Python provides the facility of working on Files.File is a named location on disk to store related information. It is used to permanently store data in a non-volatile memory (e.g. hard disk). File open,writing,reading,and closing are the 4 types of file operations.

Opening a File:

The open() function is used to open files in our system, the filename is the name of the file to be opened. The open function takes two arguments, the name of the file and and the mode for which we would like to open the file.

The mode indicates, how the file is going to be opened "r" for reading, "w" for writing and "a" for a appending. By default, when only the filename is passed, the open function opens the file in read mode.

Syntax

fileObject=open(file_name , access_mode)

- **file_name:** It is the name of the file which you want to access.
- **access_mode:** It specifies the mode in which File is to be opened. There are many types of mode like read, write, append. Mode depends the operation to be performed on File. Default access mode is read.

Example

```
f = open("python.txt",'w') # open file in current directory
f = open("/home/rgukt/Desktop/python/hello.txt")# specifying full path
```

Modes of File

There are different modes of file in which it can be opened.The default is reading in text mode. In this mode, we get strings when reading from the file. On the other hand, binary mode returns bytes and this is the mode to be used when dealing with non-text files like image or exe files.

Modes	Description
<**r**>	It opens a file in read-only mode while the file offset stays at the root.
<**rb**>	It opens a file in (binary + read-only) modes. And the offset remains at the root level.
<**r+**>	It opens the file in both (read + write) modes while the file offset is again at the root level.
<**rb+**>	It opens the file in (read + write + binary) modes. The file offset is again at the root level.
<**w**>	It allows write-level access to a file. If the file already exists, then it’ll get overwritten. It’ll create a new file if the same doesn’t exist.
<**wb**>	Use it to open a file for writing in binary format. Same behavior as for write-only mode.
<**w+**>	It opens a file in both (read + write) modes. Same behavior as for write-only mode.
<**wb+**>	It opens a file in (read + write + binary) modes. Same behavior as for write-only mode.
<**a**>	It opens the file in append mode. The offset goes to the end of the file. If the file doesn’t exist, then it gets created.
<**ab**>	It opens a file in (append + binary) modes. Same behavior as for append mode.
<**a+**>	It opens a file in (append + read) modes. Same behavior as for append mode.
<**ab+**>	It opens a file in (append + read + binary) modes. Same behavior as for append mode.

Write Method: This method writes a sequence of strings to the file.If the file already exists, then it’ll get overwritten. It’ll create a new file if the same doesn’t exist.

Write () Method : Used to write a fixed sequence of characters to a file

Syntax: fileobject.write(string)

writelines() Method: writelines can write a list of strings.

Syntax: fileobject.writelines(lines_of_text)

Example

```
fh = open("hello.txt","w")
fh.write("Hello World")
fh.close()
f = open("new.txt", "w")
lines_of_text = ["a line of text\n", "another line of text\n", "a third line\n"]
f.writelines(lines_of_text)
f.close()
```

Reading from a File: The read mode contains different methods, read(),readline() and readlines()

read() Method: Reads the given no. of bytes. It may read less if EOF is hit.

Syntax: file.read(size)

readline() Method: It’ll read an entire line from the file.

Syntax:file.readline(size)

readlines() Method: returns a list of lines

Syntax: file.readlines()

Example

```
f = open("new.txt", "w")

lines_of_text = ["a line of text", "another line of text", "a third line"]

f.writelines(lines_of_text)

f.close()

r=open("new.txt","r")

print r.read(3)

print r.readline()
print r.readlines()

r.close()
```

append mode()

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"). It will creates a new file if it does not exist.

Example

```
fh = open("new.txt", "a")
fh.write("Hello World again")
fh.close
```

Closing a File: Once you are finished with the operations on File at the end you need to close the file.It is done by the close( ) method. Always make sure you explicitly close each open file, once its job is done and you have no reason to keep it open. Because There is an upper limit to the number of files a program can open. If you exceed that limit, there is no reliable way of recovery, so the program could crash.Open files always stand a chance of corruption and data loss.

Syntax: fileobject.close( )

Example

```
obj=open("abcd.txt","w")
obj.write("Welcome to the world of Python\nYeah its great\n")
obj.close()
obj1=open("abcd.txt","r")
s=obj1.read( )
print (s )
obj1.close()
```

Note: This method is not entirely safe. If an exception occurs when we are performing some operation with the file, the code exits without closing the file. A safer way is to use a try...finally block.

Python with statement

`with` statement will automatically close the file after the nested block of code. The advantage of using a `with` statement is that it is guaranteed to close the file no matter *how* the nested block exits. If an exception occurs before the end of the block, it will close the file before the exception is caught by an outer exception handler.

Example

```
with open('output.txt', 'w') as f:
    f.write('Hi there!')
```

seek and tell methods

seek() Method: We can change our current file cursor (position) using the seek() method.

Syntax: fh.seek(6)

Tell Method: The tell() method returns our current position (in number of bytes).

Syntax: fh.tell()

Example:

```
#filename=input("Enter your file name")
f = open("try.txt","r+w")
f.seek(6)
print (f.tell())
f.write("that")
print (f.read())
f.close()
```

Exercise

1. Write a python program to compare two files?

```
import filecmp
print filecmp.cmp('file1.txt', 'file2.txt')
```

2. Write a python program to compare two files without using any functions?

```
f1=open("one.txt","r")
f2=open("two.txt","r")
a=f1.read()
b=f2.read()
if(a==b):
        print "Both are identical"
else:
        print "Both are not identical"
```

3. Write a python program to copy the content of one file to other file?

```
from shutil import copyfile
copyfile('c.txt', 'n.txt')
```

4. Write a python program to copy the content of one file to other file without using any function?

```
f=open("c.txt","r")
f1=open("n.txt","w")
for line in f:
    f1.write(line)
f.close()
```

```
f1.close()
```

5. Write a prgram to print the frequency of each word in a file and maximum word in the file?

```
f=open("file.txt","r")
d=f.readlines()
mylist=[]
m=[]
mydict={}
for i in range(0,len(d)):
	mylist.extend(d[i].split())
for i in mylist:
	m.append((len(i),i))
	if i not in mydict.keys():
		mydict[i]=mylist.count(i)
print max(m),"\n"
print mydict
```

6. Write a progrm to count the no of lines in a file?

```
f=open("t.txt","r")
c=f.readlines()
print len(c)
f.close()
```

7. Write a program to replace all the the “ IIIT” words into “RGUKT” in the file?

```
with open("t.txt","r+w") as f:
 c=f.read()
 c=c.replace("IIIT","RGUKT")
 f.seek(0)
 f.write(c)
```

8.Write a python program to insert one line into an existing file as a 3rd line?

```
import os
with open("t.txt","r") as f:
	with open("n.txt","w") as f1:
		c=0
		for line in f:
			c=c+1
			f1.write(line)
			if(c==2):
				f1.write("This is third line\n")
os.remove("t.txt")
os.rename("n.txt","t.txt")
```

9.**Write a python program to insert one word after *good* in 4th line of a file?**

```
import os
f=open("free.txt","r")
n=open("new.txt","w")
lines=f.readlines()
c=0
for line in lines:
        c=c+1
        if(c==2 and "good" in line):
                m=[]
                m.extend(line.split())
                c=m.index("good")+1
                m.insert(c,"morning")
                line=" ".join(m)
                line=line+"\n"
        n.write(line)
os.remove("free.txt")
os.rename("new.txt","free.txt")
```

10.Write a python program to add one word to the ending of each line in a file?

```
Import os
f=open("free.txt","r")
n=open("new.txt","w")
lines=f.readlines()
c=0
for line in lines:
                m=[]
                m.extend(line.split())
                m.insert(len(m),"students")
                line=" ".join(m)
                line=line+"\n"
                n.write(line)
os.remove("free.txt")
os.rename("new.txt","free.txt")
```

11. Write a Python program to mail merger?

```
# Names are in the file names.txt
# Body of the mail is in body.txt
# open names.txt for reading
names_file=open("names.txt",'r')
# open body.txt for reading
body_file=open("body.txt",'r')
# read entire content of the body
body = body_file.read()
# iterate over names
for name in names_file:
        mail = "Hello "+name+body
        # write the mails to individual files
        mail_file=open(name.strip()+".txt",'w')
        mail_file.write(mail)
            mail_file.close()
names_file.close()
body_file.close()
```