Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

O-LEVEL (M3-R5.1)
Python Programming By
Mrs. Geeta
Python Instructor
Zed-King Group of Institutions
Kaithal

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

O-LEVEL (M3-R5.1)

Python Programming

Video Tutorial

O-LEVEL (M3-R5.1) Full Video Playlist Available on YouTube Channel **Zed-king Computer Institute Kaithal**

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

Free Online Computer Classes On YouTube Channel

https://www.youtube.com/@zedkinginstitute

For Free PDF Notes

Our Telegram Channel

t.me/boost/Zedking_Institute

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

O-LEVEL (M3-R5.1)
CHAPTER-3

File Processing

File:-

File is a collection of related records which is saved on the hard disk. When any data is written on a computer, it is first stored in its RAM (Random Access Memory) and when the computer is turned off, all that data gets lost and when that data is saved then it Stored permanently on the secondary storage device (hard drive) of the computer.

There are two types of files in Python programming language.

1. High level file
2. Low level file

1. **High Level File:-** Library function is used in high level file, it can be understood by the user.
2. **Low Level File:-** It is also called system oriented file, this file is managed through system calls.

Data Storing mode:-

In Python programming language, files are stored in two modes-

1. Text file format
2. Binary file format
3. CSV File

1. **Text file format:-**

This is a human readable file, it can be edited through a text editor. In the text file, data is stored in the form of characters and lines. EOL (end of line) character is used to indicate the termination of the line in a text file which is terminated by a new line. The ASCII value of EOF is 26. All input functions do not read data after the EOF character.

1. **Binary file format:-**

This file format contains data like image, audio, video etc. is stored

Operation on file:-

To work on a file, the file is opened, then the data in the file is read or written and finally the file is closed.

Some important operations to handle files in Python are given below-

1. Opening File
2. Closing File
3. Reading File
4. Writing File
5. Append to a file
6. Renaming File
7. Deleting File

1. Opening File

When the data of a file has to be manipulated, the file is first opened. To open the file in Python, Python's in-built 'open()' method is used.

Syntax:- obj = open('file','mode') #opens a file in given mode

Parameters : There are 8 parameters for open() method but only 2 important parameters are given here.

fileName : Here, along with the file name, its extension is also given. If the file is at the same location, then there is no need to give the path.

mode : Here mode is given to open the file, if mode is not given then default is 'r'(reading) mode.

f=open("file_name.extension_name", "mode")

Eg. f=open("ak.txt", "w")

2. Closing File

When the file is opened with open() method, then it is also closed by performing some operations on the file. 'Close()' method is used to close the file.

Syntax fileObj.close() or f.close()

```
f=open("abc.txt","w")
f.write("ak it solutionn")
f.write("YouTube Channeln")
f.close()
```

To open the file, opening mode has to be specified.

There are many modes to open a text file which are as follows-

i. w **ii. r** **iii. a** **iv. w+** **v. r+** **vi. a+**

i. w mode:-

To input data in the file, the file is opened in write mode. If the file is already created then the previous data is erased and new data is written. If the file is not available then a new file is created. And data is written in it.

ii. r mode:-

It opens the file only in read mode, this is the default.

iii. a mode:-

This mode is used to work on already available files. This mode does not override (erase) the already available data, that is, it reaches the end without removing the data. If the file is not already available then a new file is created.

iv. w+ mode:-

The file is opened for reading and writing. If the file exists then it is overwritten and if the file does not exist then a new file is created.

v. r+ mode:-

The file is opened for reading and writing, the previous data is not deleted from the file, the file pointer goes to the beginning of the file.

vi. a+ mode:-

The file is opened for appending and reading. If the file already exists then it reaches the end without removing the data and if the file does not exist then a new file is created.

Binary File:-

There are following modes to open binary file-

i. wb(write binary):-

The file is opened for writing in binary format. If the file exists then it is overwritten and if the file does not exist then a new file is created.

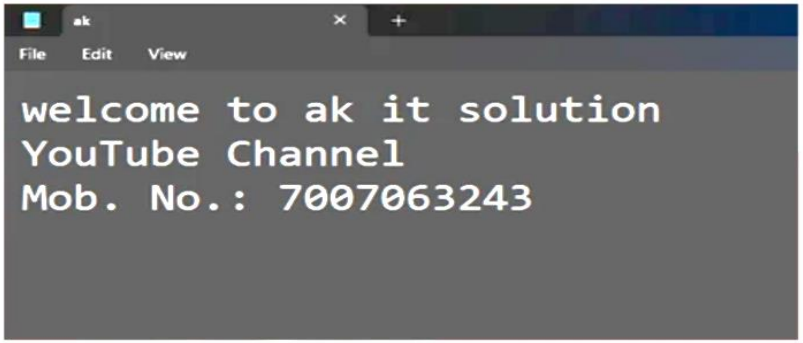
ii. rb(read binary):-

The file is opened for reading in binary format. This happens at the beginning of the file.

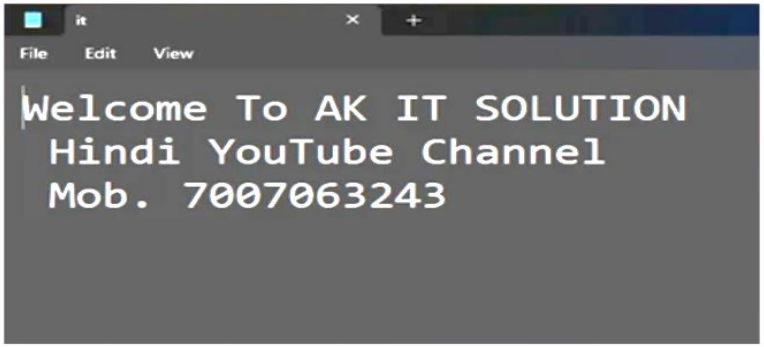
iii. ab(append binary):-

The file is opened for appending in binary format. If the file already exists then it reaches the end without removing the data and if the file does not exist then a new file is created.

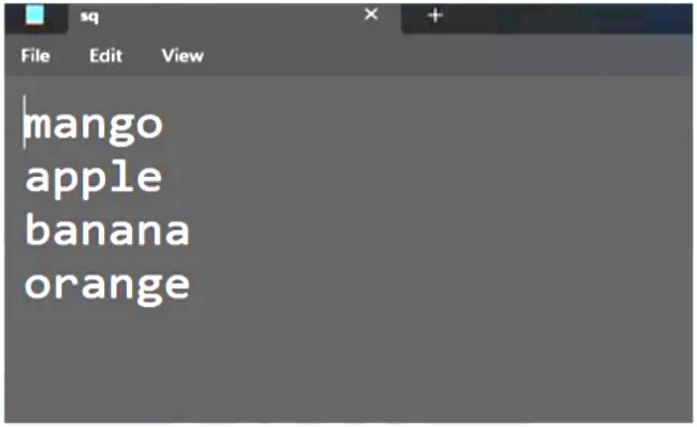
```
f=open("ak.txt",'w')
f.write('welcome to ak it solution\n')
f.write('YouTube Channel\n')
f.write('Mob. No.: 7007063243\n')
f.close()
```




```
f=open('it.txt','w')
f.writelines("Welcome To AK IT
    SOLUTION \n Hindi YouTube
    Channel\n Mob. 7007063243")
f.close()
```




```
fruit=['mango\n','apple\n','banana\n','orange\n']
f=open('sq.txt','w')
f.writelines(fruit)
f.close()
```



Write data to text file:-

Two methods are used to write characters in a text file.

1. **write method:-** Only string can be given in this
2. **write lines method:-** Only list of lines can be given in this

Reading character data from the text file:-

Files से data को read करने के लिए निम्न functions का प्रयोग किया जाता है-

The following functions are used to read data from files-

i. read():-

इससे फाइल की सभी data को read किया जाता है

With this, all the data of the file is read.

```
f=open('ak.txt','r')
print(f.read())
f.close()
```

```
Shell
>>> %Run 1211.py
 welcome to ak it solution
 YouTube Channel
 Mob. No.: 7007063243

>>>
```

iii. readline():-

इससे फाइल की केवल एक लाइन को read किया जाता है

With this, only one line of the file is read.

```
f=open('ak.txt','r')
print(f.readline())
f.close()
```

```
Shell
>>> %Run 1211.py
 welcome to ak it solution

>>>
```

```
f=open('ak.txt','r')
print(f.readline())
print(f.readline())
f.close()
```

```
Shell
>>> %Run 1211.py
 welcome to ak it solution

 YouTube Channel

>>>
```

```
f=open('it.txt','r')
l1=f.readline()
print(l1)
l2=f.readline()
print(l2)
l3=f.readline()
print(l3)
f.close()
```

```
Shell
>>> %Run 1211.py
 Welcome To AK IT SOLUTION

  Hindi YouTube Channel

  Mob. 7007063243

>>>
```

iv. readlines():-

इससे फाइल की सभी लाइन को read किया जाता है

With this, all the lines of the file are read.

```
f=open('it.txt','r')
print(f.readlines())
f.close()
```

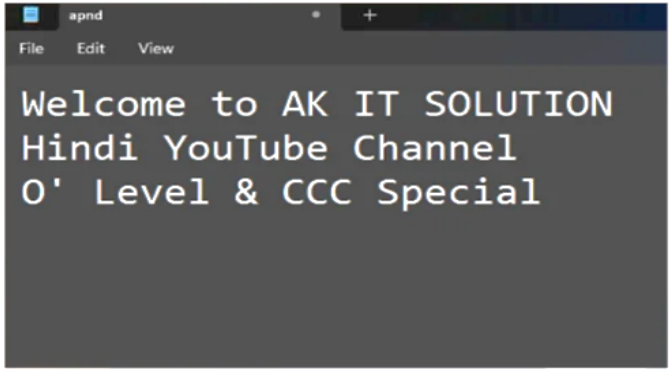
```
>>> %Run 1211.py
 ['Welcome To AK IT SOLUTION \n', ' Hindi YouTube Channel\n', ' Mob. 7007063243']
>>>
```

Append to a file:-

इसके द्वारा file के end में data को write किया जाता है इस mode में file pointer को उपलब्ध file के end में रखा जाता है।

Through this, data is written at the end of the file. In this mode, the file pointer is kept at the end of the available file.

```
f=open('apnd.txt','a')
f.write("Welcome to AK IT SOLUTION")
f.close()
f=open('apnd.txt','a')
f.write("Hindi YouTube Channel")
f.write("O' Level & CCC Special")
f.close()
```



Note:- जब हम किसी फाइल को open करते हैं तो insertion point starting में रहता है इसलिए write mode से किसी फाइल में कुछ लिखने से पुराना data delete हो जाता है। append mode से insertion point last में चला जाता है और पुराने data के साथ-साथ नया data भी file के last में insert हो जाता है।

Note:- When we open a file, the insertion point remains at the starting point, hence writing something in a file through write mode deletes the old data. With append mode, the insertion point goes to the last and along with the old data, the new data is also inserted at the last of the file.

tell():-

This function returns the current position of the cursor. That is, it is used to know the current position of the cursor in the file.

"the tell() method is used to print the byte number at which the file pointer exists"

```
f=open('tt.txt','w')
f.write("Welcome to ak it solution youtube channel ")
f.close()
f=open('tt.txt','r')
print(f.tell())
print(f.read(2))
print(f.tell())
print(f.read(12))
print(f.tell())
f.close()
```

```
Shell
>>> %Run 1211.py
 0
 We
 2
 lcome to ak
 14
>>>
```

Zed-King
GROUP OF INSTITUTIONS
ISO 9001 : 2015 Certified

seek():-

इसका use pointer को move कराने के लिए किया जाता है।

It is used to move the pointer

```
f=open('sik.txt','w')
f.write("Welcome to ak it solution Hindi Computer Youtube channel")
f.close()
f=open('sik.txt','r')
print("Current Cursor Position: ",f.tell())
f.seek(5)
print("Current curson position: ",f.tell())
f.seek(15)
print("Current cursor position: ",f.tell())
f.close()
```

Shell

```
>>> %Run 1211.py
 Current Cursor Position:  0
 Current curson position:  5
 Current cursor position:  15

>>>
```

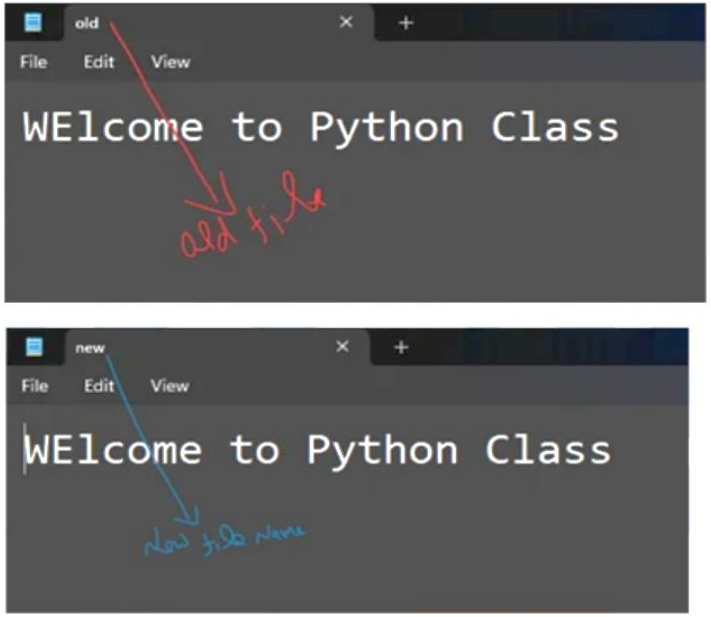
Renaming File:–

Python provides the os module for renaming files, which contains some in-build (predefined) methods. To use this module in python program, firstly the os module has to be imported.

Rename function is used to change the name of the file.

Syntax:- **os.rename(“old_file_name”, “New_file_Name”)**

```
import os
f=open("old.txt",'w')
st="WElcome to Python Class"
f.write(st)
f.close()
os.rename("old.txt","new.txt")
f.close()
```



With Statement:-

While opening a file in Python, with statement is used. When a file is opened with with statement then there is no need to close it, the file gets closed automatically.

Syntax:-

with open("file_name.extension_name", "mode") as fp:

How to check file is closed or not

print(fp.closed)

```
with open("ii.txt",'w') as fp:
   a="akhilesh"
   fp.write(a)
   print(fp.closed)
print(fp.closed)
```

Deleting File (Remove File):-

Python में किसी फाइल को डिलीट करने के लिए remove( ) method का प्रयोग किया जाता है

remove( ) method is used to delete a file in Python

Syntax:- **os.remove(file_name)**

```
import os

os.remove("old.txt")
```

Binary Files:-

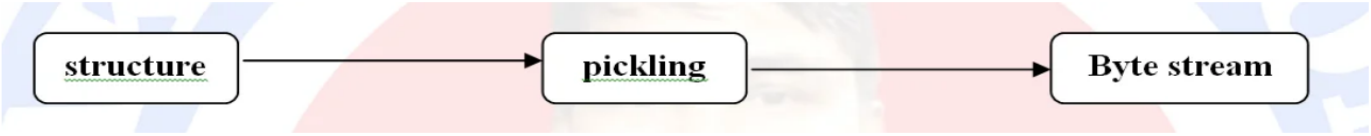
जो फ़ाइलें 0 और 1 (आंतरिक रूप से) के रूप में मौजूद होती हैं, उन्हें बाइनरी फ़ाइलें कहा जाता है। बाइनरी में केवल दो अंक 0 और 1 होते हैं।

The files which are present in the form of 0 & 1 (internally) are referred to as binary files. Binary consist of only two digits 0 & 1.

Pickling & unpickling:-

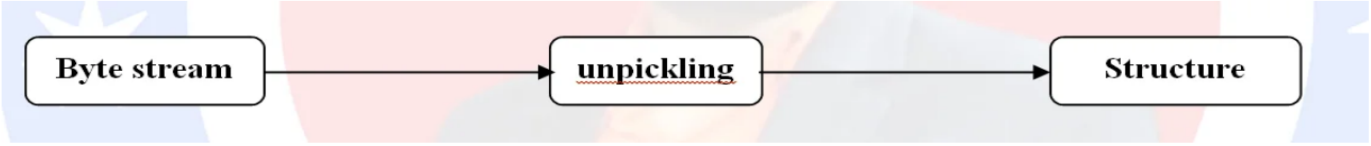
पिकलिंग फ़ाइल में लिखने से पहले संरचना (जैसे सूची, शब्दकोश इत्यादि) को बाइट स्ट्रीम में परिवर्तित करने की प्रक्रिया को संदर्भित करता है।

Pickling refers to the process of converting structure (such as list, dictionary etc.) to a byte stream before writing to the file.



अनपिकलिंग फ़ाइल में लिखने से पहले बाइट स्ट्रीम को संरचना (जैसे सूची, शब्दकोश आदि) में परिवर्तित करने की प्रक्रिया को संदर्भित करता है।

Unpickling refers to the process of converting byte stream to structure (such as list, dictionary etc) before writing to the file.



```
import pickle
def write():
    f=open('abc.dat','wb')
    l=['amit','anil','sunil','dinesh']
    pickle.dump(l,f)
    f.close()
write()
print("Data Inserted")
```

```
import pickle
def read():
    f=open('abc.dat','rb')
    lst=pickle.load(f)
    print(lst)
    f.close()
read()
print("file Reading done")
```

Pickle module:-

Python provides a module called pickle module which consist two functions which is used to read or write data to the file.

i. dump():- This function is used to dump (write) the content to the file.

Syntax:- structure-pickle.dump(<structure>,fileobject)

ii. load():- this function is used to read data from a file.

Syntax:- structure-pickle.load(fileobject)

CSV:- Comma Separated Values