

CHAPTER 5 : FILE HANDLING

NOTES.....

A file in itself is a bunch of bytes stored on some storage device like hard disk, thumb drive etc.

TYPES OF FILE

TEXT FILE

- 1) A text file stores information in ASCII or unicode characters
- 2) each line of text is terminated, (delimited) with a special character known as EOL

BINARY FILES

- 1) A binary file is just a file that contains information in the same format in which the information is held in memory i.e the file content that is returned to you is raw.
- 2) There is no delimiter for a line
- 3) No translation occurs in binary file
- 4) Binary files are faster and easier for a program to read and write than are text files.
- 5) Binary files are the best way to store program information.

Steps to process a FILE

- 1) Determine the type of file usage :
in this section we determine whether file need to be open or not
- 2) open the file and assign its references to a file object or file handle
- 3) Process as required
- 4) close the file

OPENING AND CLOSING FILES

- 1) **open()** function is used to open a file

Syntax:

file variable/file handle=open(file_name,access mode)

Example

a) F= open('abc.txt,'w')

this statement opens abc.txt in write mode

Note : if file mode is not mentioned in open function then default file mode i.e 'r' is used

2) **close()** : the close() method of a file object flushes any unwritten

| Text File Mode | Binary File Mode | Description | Notes |
|-----------------------|-------------------------|--------------------|--|
| 'r' | 'rb' | Read only | File must exist already ,otherwise python raises I/O error |
| 'w' | 'wb' | Write only | *If the file does not exist ,file is created. *If the file exists, python will truncate existing data and overwrite in tne file. So this mode must be used with caution. |
| 'a' | 'ab' | append | *File is in write only mode. *if the file exists, the data in the file is retained and new data being written will be appended to the end. *if the file does not exist ,python will create a new file. |
| 'r+' | 'r+b' or 'rb+' | Read and write | *File must exist otherwise error is raised. *Both reading and writing operations can take place. |
| 'w+' | 'w+b' or 'wb+' | Write and read | *File is created if doesn't exist. *If the file exists, file is truncated(past data is lost). *Both reading and writing operations can take place. |
| 'a+' | 'a+b' or 'ab+' | Write and read | *File is created if does not exist. *If file exists, files existing data is retained ; new data is appended. *Both reading and writing operations can take place. |

information and close the file object after which no more writing can be done

SYNTAX: fileobject.close()

FILES MODE

it defines how the file will be accessed

TEXT FILE HANDLING

Methods to read data from files

| S.N O. | Method | Syntax | Description |
|-----------|--------------------|-----------------------------|---|
| 1 | Read() | <filehandle>.read([n]) | Reads at most n bytes ;If no n is specified, reads the entire file. Returns the read bytes in the form of a string . In [11]:file 1=open(“E:\\mydata\\info.txt”) In [12]:readInfo=file1.read(15) In [13]:print(readInfo)#prints first 15 #characters of file In [14]:type(readInfo) Out[14]:str |
| 2 | Readline() e() | <filehandle>.readline([n]) | Reads a line of input ;if n is specified reads at most n bytes. Returns the read bytes in the form string ending with in(line)character or returns a blank string if no more bytes are left for reading in the file. In [20]:file1 = open(“E:\\mydata\\info.txt”) In [20]: readInfo =file1.readline() In [22]:print (readInfo) |
| 3 | readlines() s() | <filehandle>.readlines() | Read all lines and returns them in a list In [23]:file1 =open(“E:\\mydata\\info text”) In [24]:readInfo =file1.readlines() In [25]:print (readInfo) In [26]:type (readInfo) Out[26]:list |

Writing data into files

| S. NO | Name | Syntax | Description |
|--------------|--------------|-----------------------------|--|
| 1 | Write() | <filehandle>.write(str1) | Write string str1 to file referenced by <filehandle> |
| 2 | Writelines() | <filehandle>.writelines (L) | Writes all strings in list L as lines to file referenced by <filehandle> |