

UNIT V
FILES, MODULES, PACKAGES

1. FILE AND ITS OPERATION

- File is a collection of record.
- A file stores related data, information, settings or commands in secondary storage device like magnetic disk, magnetic tape, optical disk, flash memory.

File Type

1. Text file
2. Binary file

Text file	Binary file
Text file is a sequence of characters that can be sequentially processed by a computer in forward direction Each line is terminated with a special character called the EOL or end of line character	A binary files store the data in the binary format(i.e .0's and 1's) It contains any type of data (pdf,images,word doc,spreadsheet,zip files,etc)

Mode in File

Module	Description
r	Read only
w	mode Write
a	only Appending
r+	only Read and write only

Differentiate write and append mode:

Write mode	Append mode
<ul style="list-style-type: none"> • It is used to write a string in a file • If file is not exist it creates a new file • If file is exit in the specified name, the existing content will overwrite in a file by the given string 	<ul style="list-style-type: none"> • It is used to append (add) a string into a file • If file is not exist it creates a new file • It will add the string at the end of the old file

File Operation:

- ✓ Open a file
- ✓ Reading a file
- ✓ Writing a file
- ✓ Closing a file

1. Open () function:

- Python's built-in open function to get a file object.
- The open function opens a file.
- It returns a something called a file object.
- File objects can turn methods and attributes that can be used to collect

Syntax:

```
file_object=open("file_name" , "mode")
```

Example:

```
fp=open("a.txt","r")
```

Create a text file

```
fp=open ("text.txt","w")
```

2. Read () function

Read functions contains different methods

- read() – return one big string
- readline() – return one line at a time
- readlines() – return a list of lines

Syntax:

```
file_name.read ()
```

Example:

```
fp=open("a.txt","w")
```

```
print(fp.read())
```

```
print(fp.read(6))
```

```
print (fp.readline())
```

```
print (fp.readline(3))
```

```
print (fp.readlines())
```

a.txt

A file stores related data, information, settings or commands in secondary storage device like magnetic disk, magnetic tape, optical disk, flash memory.

Output

Reading file using looping:

- Reading a line one by one in given file

```
fp=open("a.txt","r")
for line in fp:
    print(line)
```

3. Write () function

This method is used to add information or content to existing file.

Syntax:

```
file_name.write( )
```

Example:

```
fp=open("a.txt","w")
fp.write("this file is a.txt")
fp.write("to add more lines")
fp.close()
```

Output: a.txt

```
A file stores related data,
information, settings or commands
in secondary storage device like
magnetic disk, magnetic tape,
optical disk, flash memory.
this file is a.txt to
add more lines
```

4. Close () function

It is used to close the file.

Syntax:

```
File name.close()
```

Example:

```
fp=open("a.txt","w")
fp.write("this file is a.txt")
fp.write("to add more lines")
fp.close()
```

Splitting line in a text line:

```
fp=open("a.txt","w")
for line in fp:
    words=line.split()
print(words)
```

2. Write a program for one file content copy into another file:

```
source=open("a.txt","r")
destination=open("b.txt","w")
for line in source:
    destination.write(line)
source.close()
destination.close()
```

Output:

Input a.txt	Output b.txt
A file stores related data, information, settings or commands in secondary storage device like magnetic disk, magnetic tape, optical disk, flash memory	A file stores related data, information, settings or commands in secondary storage device like magnetic disk, magnetic tape, optical disk, flash memory

3. Write a program to count number of lines, words and characters in a text file:

```
fp = open("a.txt","r")
line =0
word = 0
character = 0
for line in fp:
    words = line . split ( )
    line = line + 1
    word = word + len(words)
    character = character +len(line)
print("Number of line", line)
print("Number of words", word)
print("Number of character", character)
```

Output:

```
Number of line=5
Number of words=15
Number of character=47
```