

FILE MANIPULATIONS

File manipulation means change or access the content in the file.

- File Positions
- Renaming and Delete a File
- Directories in Python

1. File Positions

There are two methods to access the positions of the file.

1. tell()
2. seek()

i) tell() method

This method is used to tell the current position within a file. It starts from the beginning of the file and this method followed by read() and write() method.

Syntax:

```
file_object.tell()
```

Example

```
f=open("F:/Python/test.txt","r")
print(f.tell())      ;#0th position
print(f.read())
print(f.tell())      ;#Last Position
f.close()
```

Output

```
0
This is the First Line in the file
This is the Second Line in the file
73
```

ii) seek() method

This method is used to change the current file position.

Syntax:

```
file_object.seek(offset, from)
```

- seek() method set the file's current position at the offset.

- The default argument of offset is 0. The offset represents the number of the bytes to be moved.

The from argument represents three values.

- 0 → represents the beginning of the file
- 1 → represents the current position as reference
- 2 → represents the end of the file

Program to use read(), tell(),seek() methods

sample.txt - Problem solving and python programming

```
f1=open("F:/Python/sample.txt","r")
str=f1.read(10)
print("read string is : ",str)
position=f1.tell()
print("current file position:",position)
position=f1.seek(0,0)
str=f1.read(10)
print("again read string is :",str)
f1.close()
```

Output:

```
read string is: Problem So
current file position : 10
again read string is : Problem So
```

RENAMING AND DELETING A FILE

Two file processing operations are there, they are

- rename() method
- remove() method

i) rename() method

The rename() method takes two argument, the current filename and new filename.

```
os.rename(current_filename, new_filename)
```

Syntax:

Example

```
import os
os.rename("test.txt","new.txt ")
```

ii) remove() method

The remove() method is used to delete the file. The argument contains file name.

Syntax:

```
os.remove(filename)
```

Example:

```
import os
os.remove("new.txt ")
```

DIRECTORIES IN PYTHON

All files are contained within various directories. The os module has several methods to create, remove and change directories.

S.No	Name	Syntax	Description	Example
1	mkdir()	os.mkdir("new_dir")	This method is used to create a directory	os.mkdir("test")
2	chdir()	os.chdir("new_dir")	This method is used to change a directory	os.chdir("new_dir")
3	getcwd()	os.getcwd()	This method is used to display current directory	os.getcwd()
4	rmdir()	rmdir('dir_name')	This method is used to remove a directory	os.rmdir('new_dir')

FORMAT OPERATOR

The argument of write has to be a string, so if we want to put other values in a file, we have to convert them to strings. The easiest way to do that is with str:

```
f=open('stringsample.txt','w')
f.write(5)          #TypeError
f.write(str(5))
```

An alternative is to use the format operator, %. The first operand is the format string, which contains one or more format sequences, which specify how the second operand is formatted. The result is a string.

Example:

```
var=8
print("The Value is : %d"%var)
```

Output:

The Value is : 8

Some of the format strings are.

<u>Sl No</u>	Conversion	Meaning
1	d	Signed integer decimal.
2	i	Signed integer decimal
3	o	Unsigned octal.
4	u	Unsigned decimal.
5	x	Unsigned hexadecimal (lowercase).
6	X	Unsigned hexadecimal (uppercase).
7	e	Floating point exponential format (lowercase).
8	E	Floating point exponential format (uppercase).
9	f	Floating point decimal format.
10	F	Floating point decimal format.