File Handling in Hindi - Python in Hindi

Python - File Handling

जब Computer पर कोई data लिखा जाता है तब वो पहले अपने RAM(Random Access Memory) पर store होता है और जब Computer को turn off किया जाता है तब वो सारा data lose हो जाता है |

और जब उस data को save किया जाता है तब वो Computer के Secondary Storage Device(Hard Drive) पर permanently store किया जाता है |

Python में File को handle करने के लिए कुछ महत्वपूर्ण operatios निचे दिए गए है |

1. Opening File
2. Closing File
3. Reading File
4. Writing File
5. Renaming File
6. Deleting File

1. Opening File

File के data को जब manipulate करना हो तो पहले file को open किया जाता है |

Python में file को open करने के लिए python के in-built 'open()' method का इस्तेमाल किया जाता है |

Syntax for open() Method

```
fileObj = open(fileName, mode)
```

Parameters : open() method के लिए 8 parameters होते है लेकिन यहाँ पर सिर्फ 2 महत्वपूर्ण parameters दिए गए है |

fileName : यहाँ पर file name के साथ उसका extension भी दिया जाता है | अगर same location पर file हो तो उसे path देने की जरुरत नहीं होती है |

mode : यहाँ पर file को open करने के लिए mode दिया जाता है | अगर mode दिया नहीं जाता है तो default 'r'(reading) mode होता है |

Example for Opening File

```
file = open('textfile.txt') #default mode is 'r'
#or
file = open('textfile.txt', 'r')
```

Modes for Opening a File

Modes	Description
a(append)	file को appending के लिए open किया जाता है | अगर file पहले से ही exist होती है तो बिना data remove किये वो end पर पहुँच जाता है और अगर file exist नहीं होती है तो new file create की जाती है |
ab(append in binary)	file को binary format में appending के लिए open किया जाता है | अगर file पहले से ही exist होती है तो बिना data remove किये वो end पर पहुँच जाता है और अगर file exist नहीं होती है तो new file create की जाती है |
a+(append and read)	file को appending और reading के लिए open किया जाता है | अगर file पहले से ही exist होती है तो बिना data remove किये वो end पर पहुँच जाता है और अगर file exist नहीं होती है तो new file create की जाती है |
ab+(append and read in binary)	file को binary format में appending और reading के लिए open किया जाता है | अगर file पहले से ही exist होती है तो बिना data remove किये वो end पर पहुँच जाता है और अगर file exist नहीं होती है तो new file create की जाती है |
r(read)	ये default होता है | file को reading के लिए open किया जाता है | ये file के beginning पर होता है |
rb(read in binary)	file को binary format में reading के लिए open किया जाता है | ये file के beginning पर होता है |
r+(read and write)	file को reading और writing के लिए open किया जाता है | ये file के beginning पर होता है |
rb+(read and write in binary)	file को binary format में reading और writing के लिए open किया जाता है | ये file के beginning पर होता है |
w(write)	file को writing के लिए open किया जाता है | अगर file exist होती है तो उसे overwrite किया जाता है और अगर file exist नहीं होती है तो new file create की जाती है |
wb(write in binary)	file को binary format में writing के लिए open किया जाता है | अगर file exist होती है तो उसे overwrite किया जाता है और अगर file exist नहीं होती है तो new file create की जाती है |
w+(write in binary)	file को reading और writing के लिए open किया जाता है | अगर file exist होती है तो उसे overwrite किया जाता है और अगर file exist नहीं होती है तो new file create की जाती है |
wb+(write and read in binary)	file को binary format में reading और writing के लिए open किया जाता है | अगर file exist होती है तो उसे overwrite किया जाता है और अगर file exist नहीं होती है तो new file create की जाती है |

2. Closing File

जब file को open() method से open किया जाता है तब file पर कुछ operations करके close भी किया जाता है |

File को close करने के लिए 'close()' method का इस्तेमाल किया जाता है |

close() method को जब file object की मदद से call किया जाता है तब file पर write हुए hidden data को flush किया जाता है |

जहा पर file को close किया जाता है उसके बाद बिना file open किये उस file पर कोई भी operation नहीं किया जा सकता है |

Syntax for close() Method

```
fileObj.close()
```

Parameters : close() method के लिए कोई parameter नहीं होता है |

Example for Closing File in Python

Source Code :

```
file = open("textfile.txt",'a') #Opening File

file.close() #Closing File
```

3. Reading File

File का data read करने के लिए read() method का इस्तेमाल किया जाता है |

read() Method in Python

```
fileObj.read(size)
```

Parameter :

size : Optional. जितना data लेना है उसकी size यहाँ पर दी जाती है | अगर दिया नहीं जाता है तो end of file तक data को read किया जाता है |

अगर size पर invalid value दी जाती है तो end of file तक data को read किया जाता है |

textfile.txt

```
Hello World
```

Example पर file को read mode पर open किया गया है और open किये हुए file का data read करने के लिए read() method का इस्तेमाल किया गया है और बाद में file को close किया गया है |

Source Code :

```
file = open("textfile.txt",'r')
print(file.read())
file.close()
```

Output :

Hello World

Example for read() Method with Parameter

Example पर read() method पर parameter में size सिर्फ 2 दी हुई है इसीलिए file से सिर्फ पहले दो character ही return हुए है |

Source Code :

```
file = open("textfile.txt",'r')
print(file.read(2))
file.close()
```

Output :

He

4. Writing File

File पर data को write करने के लिए write() method का इस्तेमाल किया जाता है |

write() Method in Python

fileObj.write(str)

Parameter :

str : जो string file पर write करना है वो string यहाँ पर दिया जाता है |

write() method file को overwrite करके शुरुआत से data को write किया जाता है |

Returning Value :

write() method ये file के character की size return करता है |

textfile.txt

Hello World

Example पर write mode पर file को open किया गया है उसके बाद file पर data को write() method की मदद से write करके file को close किया गया है |

Source Code :

```
file = open("textfile.txt",'w+')
file.write("Hello")
file.close()
```

Output :

textfile.txt
Hello

5. Renaming File

File को rename करने के लिए 'os' module को import करके उसके rename() method का इस्तेमाल किया जाता है |

rename() Method in Python

os.rename(fileName, newFileName)

Parameter :

fileName : जिस file का name rename करना है उस file का name यहाँ पर दिया जाता है |

newFileName : जो नाम file को देना है वो file का नाम यहाँ पर दिया जाता है |

अगर पहले parameter में file exist नहीं होती है तो 'FileNotFoundError' ये exeception raise होता है |

अगर दुसरे parameter में file exist होती है तो 'FileExistsError' ये exeception raise होता है |

Source Code :

```
import os
os.rename("textfile.txt", "myfile.txt")
#rename textfile.txt to myfile.txt
```

6. Removing File

File को remove करने के लिए 'os' module को import करके उसके remove() method का इस्तेमाल किया जाता है |

rename() Method in Python

os.rename(fileName)

Parameter :

fileName : जिस file को remove करना है उस file का नाम यहाँ पर दिया जाता है |

अगर parameter पर दी हुई file exist नहीं होती है 'FileNotFoundError' ये exception raise होता है |

Source Code :

```
import os
os.remove("myfile.txt")
#removed file 'myfile.txt'
```

File Methods in Python

File को handle करने के लिए और भी method है | वो सभी methods निचे दिए गए है |

close()	open किये गए file को close किया जाता है	
flush()	file stream से writing buffer को flush करने के लिए इस्तेमाल किया जाता है	
fileno()	file descriptor integer में return करता है	
isatty()	file; terminal device से connect हुआ है या नहीं ये boolean value में return करता है	
read()	File का data read करने के लिए इस्तेमाल किया जाता है	
readline()	file के line से दिए हुए size तक character/byte को read करके return किया जाता है	
readlines()	file के line से दिए हुए दिखाऊ size तक character/byte को read करके lines को list में return किया जाता है	
seek()	इसका इस्तेमाल offset पर file की current position set करने के लिए किया जाता है	
tell()	file पर data read या write करने के file की current position return की जाती है	
writable()	file पर data write किया गया है या नहीं ये boolean value में return किया जाता है	
write()	File पर string को write करके उस string की length को return करता है	
writelines()	इसका इस्तेमाल दिए हुए sequence को file पर write किया जाता है	