MARUDHAR KESARI JAIN COLLEGE FOR WOMEN, VANIYAMBADI

PG DEPARTMENT OF COMPUTER APPLICATIONS

Subject Name : PYTHON PROGRAMMING

CLASS : I-BCA

SUBJECTCODE: 23UCA11

UNIT-V

Python File Handling: Types of files in Python -Opening and Closing files-Reading and Writing files: write() and write lines() methods-append() method read() and readlines() methods–with keyword–Splitting words–File methods-File Positions-Renaming and deleting Files.

Introduction

Filehandlingisanintegralpartofprogramming.FilehandlinginPythonissimplifiedwithbuilt-in methods, which include creating, opening, and closing files.

Whilefilesareopen,Pythonadditionallyallowsperformingvariousfileoperations,suchasreading, writing, and appending information.

FileHandlingin Python

Filehandlingisanimportantactivityineverywebapp.Thetypesof activities that you can perform on the opened file are controlled by Access Modes. These describe how the file will be used after it has been opened.

These modes also specify where the file handle should be located withinthefile.Similartoapointer,afilehandleindicateswheredata should be read or put into the file.
InPython,therearesixmethodsoraccessmodes,whichare:

1. **ReadOnly('r'):**Thismodeopensthetextfilesforreadingonly.The start of the file is where the handle is located. It raises the I/O error if the file does not exist. This is the default mode for opening files as well.
2. **Read and Write('r+'):** This method opens the file for bothreading andwriting.Thestartofthefileiswherethehandleislocated.Ifthe file does not exist, an I/O error gets raised.
3. **WriteOnly ('w'):**Thismodeopensthefileforwritingonly.Thedata in existing files are modified and overwritten. The start of the file is where the handle is located. If the file does not already exist in the folder, a new one gets created.
4. **Write and Read ('w+')**: This mode opens the file for both reading andwriting.Thetextisoverwrittenanddeletedfromanexistingfile. The start of the file is where the handle is located.

5. **Append Only ('a')**: This mode allows the file to be opened for writing.Ifthefiledoesn'tyetexist,anewonegetscreated.Thehandle is setat theend of the file. The newlywritten data willbe added at the end, following the previously written data.
6. **Append and Read ('a+'):** Using this method, you can read and write inthefile.Ifthefiledoesn'talreadyexist,onegetscreated.Thehandle is set at the end of the file. The newly written text will be added at the end, following the previously written data.
   Belowisthecoderequiredtocreate,writeto,andreadtextfilesusing the Python file handling methods or access modes.

HowtoCreateFiles in Python

InPython,youusethe`open()`functionwithoneofthefollowing options – "x" or "w" – to create a new file:

- **"x" –Create**:thiscommandwillcreateanewfileifandonlyifthere is no file already in existence with that name or else it will return an error.
  ExampleofcreatingafileinPythonusingthe"x"command:

```
#creatingatextfilewiththecommandfunction"x"

f=open("myfile.txt","x")
```

We've now created a new empty text file! But if you retry the code above – for example, if you try to create a new file with the same nameasyouusedabove(ifyouwanttoreusethefilenameabove)you willgetanerrornotifyingyouthatthefilealreadyexists.It'lllooklike the image below**w" – Write**: this command will create a new text file whether or not there is a file in the memory with the new specified name. It does not return an error if it finds an existing file with the same name – instead it will overwrite the existing file.

Exampleofhowtocreateafilewiththe"w"command:

```
#creatingatextfilewiththecommandfunction"w"
```

```
f=open("myfile.txt","w")
```

```
#This"w"commandcanalsobeusedcreateanewfilebutunlikethethe"x"commandthe"w"commandwill overwrite any existing file found with the same file name.
```

Withthecodeabove,whetherthefileexistsorthefiledoesn'texistin the memory, you can still go ahead and use that code. Just keep in mind that it will overwrite the file if it finds an existing file with the same name.

HowtoWritetoaFileinPython

TherearetwomethodsofwritingtoafileinPython,whichare:

The`write()`method:

Thisfunctioninsertsthestringintothetextfileonasingleline.

Basedonthefilewehavecreatedabove,thebelowlineofcodewill insert the string into the created text file, which is "myfile.txt."

```
file.write("HelloThere\n")
```

The`writelines()`method:

Thisfunctioninsertsmultiplestringsatthesametime. Alistofstring elements is created, and each string is then added to the text file.

Usingthepreviouslycreatedfileabove,thebelowlineofcodewill insert the string into the created text file, which is "myfile.txt."

```
f.writelines(["HelloWorld","YouarewelcometoFcc\n"])
```

Example:

```
#Thisprogramshowshowtowritedatainatextfile.

file=open("myfile.txt","w")
```

```
L=["ThisisLagos\n","ThisisPython\n","ThisisFcc \n"]

#iassigned["ThisisLagos\n","ThisisPython\n","ThisisFcc\n"]to#variable L, youcanuseanyletteror word of your choice.
#Variablearecontainersinwhichvaluescanbestored. #
The \n is placed to indicate the end of the line.

file.write("HelloThere\n")
file.writelines(L)
file.close()

#Usetheclose()tochangefileaccess modes
```

HowtoReadFromaTextFileinPython

Therearethreemethodsofreadingdatafromatextfilein Python. They are:

The`read()`method:

Thisfunctionreturnsthebytesreadasastring.Ifnonisspecified, it then reads the entire file.

Example:

```
f=open("myfiles.txt","r")
#('r')opensthetextfilesforreadingonly print(f.read())
#The"f.read"printsoutthedatainthetextfileintheshellwhenrun.
```

Thereadline()method:

Thisfunctionreadsalinefromafileandreturnsitasastring.Itreads at most n bytes for the specified n. But even if n is greater than the length of the line, it does not read more than one line.

```
f=open("myfiles.txt","r")
print(f.readline())
```

The`readlines()`method:

Thisfunctionreads allofthelinesandreturnsthemasstringelements in a list, one for each line.

Youcanreadthefirsttwolinesbycalling `readline()`twice,reading the first two lines of the file:

```
f=open("myfiles.txt","r")
print(f.readline())
print(f.readline())
```

HowtoCloseaTextFilein Python

Itisgoodpracticetoalwaysclosethefilewhenyouaredonewithit.

Exampleofclosingatextfile:

Thisfunctionclosesthetextfilewhenyouaredonemodifyingit:

```
f=open("myfiles.txt","r")
print(f.readline())
f.close()
```

Theclose()functionattheendofthecodetellsPythonthatwell,Iam done with this section of either creating or reading – it is just like saying End.

Example:

Theprogrambelowshowsmoreexamplesofwaystoreadandwrite data in a text file. Each line of code has comments to help you understand what's going on:

```
#Programtoshowvariouswaystoreadand #
write data in a text file.

file=open("myfile.txt","w")
L=["ThisisLagos\n","ThisisPython\n","ThisisFcc \n"]

#iassigned["ThisisLagos\n","ThisisPython\n","ThisisFcc\n"] #to
variable L
```

```
#The\nisplacedtoindicateEndofLine

file.write("HelloThere\n")
file.writelines(L)
file.close()
#usetheclose()tochangefileaccess modes



file=open("myfile.txt","r+")
print("OutputoftheReadfunctionis")
print(file.read())
print()

#Theseek(n)takesthefilehandletothenth #
byte from the start.
file.seek(0)

print("TheoutputoftheReadlinefunctionis")
print(file.readline())
print()

file.seek(0)

#Toshowdifferencebetweenreadand readline

print("OutputofRead(12)functionis")
print(file.read(12))
print()

file.seek(0)

print("OutputofReadline(8)functionis")
print(file.readline(8))
```

```
file.seek(0)
#readlinesfunction
print("OutputofReadlinesfunctionis")
print(file.readlines())
print()
file.close()
```

This istheoutputoftheabovecodewhenrunintheshell. Iassigned "This isLagos","This isPython", and "This isFcc" to "L" and then asked it to print using the "file.read"function.

Thecodeaboveshowsthatthe"readline()"functionisreturningthe letter based on the number specified to it, while the "readlines()" function is returning every string assigned to "L" including the \n.
Thatis, the"readlines()"functionwillprintoutalldatainthefile.