Persistent

File Handling in Python

Objectives

At the end of this session, you will be able to :

- Get an overview of Python File handling
- Opening a File
- Reading from a file, writing to a file
- Closing a File

File Handling

File objects in Python

File objects in Python can be used not only to access normal disk files, but also any other type of file that uses that abstraction.

The open() built-in function in Python returns a file object which is then used for all succeeding operations on a file.

The basic syntax of the open() built-in function is as shown below:

```
file_object = open(file_name, access_mode='r', buffering=-1)
```

Example 1: Open a file for read

```
fp = open('/etc/python/test.py')
```

Example 2: Open a file for write

```
fp = open('/etc/python/test.py', 'w’)
```

Example 3: Open a file for read/write

```
fp = open('/etc/python/test.py‘, ‘r+’)
```

Access modes for file objects

File Mode	Description
r	open a file for read
w	open a file for write
a	open a file for append
r+	open a file for read and write
rb	open a file for binary read
wb	open a file for binary write
ab	open a file for binary append
rb+	open a file for binary read and write

File built-in methods

- Once open() has completed successfully and returned a file object, all subsequent access to the file can be done using the returned “handle”

- **File methods come in four different categories:**

  input: read(), readline(), readlines()

  output: write(), writelines()

  movement within a file: seek(), tell()

  miscellaneous

File object methods

File Object Method	Description
file.close()	Close file
file.flush()	flush internal buffer for file
file.read (size=-1)	read all or size bytes of file as a string and return it
file.readline()	read and return one line from file (includes trailing "\n")
file.readlines()	read and returns all lines from file as a list (includes all trailing "\n" characters)
file.tell()	return current location within file
file.write(str)	write string str to file
file.writelines(list)	write list of strings to file
file.seek(off, whence)	move to a location within file, off bytes offset from whence (0 ==beginning of file, 1 == current location, or 2 == end of file)

File reading writing example

Example

```
f = open("myfile.txt", "w")         #file  object created in write mode
print (dir(f))                      #list of all attributes and functions of passed object
print ("------------------------------------------------------------“)
f.write("Welcome to PSL\n")
f.write("ZZZZZZZ\n")
f.write("AAAAAAAAAA\n")
f.close()


f1 = open ("myfile.txt")
print ("1st line = ", f1.readline())
f1.close()
```

File reading example

Example

```
fin=open('data.txt','r')

data =fin.readlines()   #list of all lines

print (data)

for line in data:

    print (line)

fin.close()
```

File reading – seek, tell example

Example

```
f = open('target.txt', "w")
f.write("hello World\n")
f.close()
f = open("target.txt")   #by default read mode
while True:
        str = f.readline()
        if str: print (str)
        else: break
f.seek(5)                            #absolute positioning
print ("Current position = ", f.tell())
print (f.read())                     #" World\n"
f.close()
```

Assignments

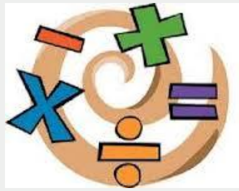
1. Accept a string from user, check whether it is Palindrome
2. Read file content of given file “empdata” and print total sal

File data:

1a:ABC:25:25000

2a:XYZ:30:30000

3a:LMN:45:60000



Summary :

With this we have come to an end of session, where we discussed about ….

- File handling

In the next session we will discuss about

- Exceptions handling in Python

Reference material

- http://www.tutorialspoint.com/python
- http://www.learnpython.org/
- http://docs.python.org/2/tutorial/

Questions

Key contacts

Sakshi Jamgaonkar

sakshi_jamgaonkar@persistent.com

Asif Immanad

asif_immanad@persistent.co.in

Persistent

Thank you!

Persistent University