



**AC Patil**  
College of Engineering

*Jawahar Education Society's  
A. C. Patil College of Engineering, Kharghar  
Navi Mumbai 410210*

Student Name:

PRN No.: 2211110

Course Name: C.S.E. (IoT CS BC)

Course code: CSL405

Year: S.E.

Semester: IV

Roll No.:

### Experiment Evaluation Sheet

Experiment No.: 3

Experiment Name:  
To access files and directory in Python

Sr No.	Evaluation Criteria	Marks (Out of 9)	Performance Date	Correction Date and Signature of Instructor
1	Experiment Performance			
2	Journal Performance			
3	Punctuality			
Total				

**Aim :** To access files and directory in Python

**Software required :** Python

## **Theory :**

### **Reading and Writing Files:**

`open()` returns a file object, and is most commonly used with two positional arguments and one keyword argument: `open(filename, mode, encoding=None)`. If you're not using the `with` keyword, then you should call `f.close()` to close the file and immediately free up any system resources used by it. After a file object is closed, either by a `with` statement or by calling `f.close()`, attempts to use the file object will automatically fail.

### **read() :**

To read a file's contents, call `f.read(size)`, which reads some quantity of data and returns it as a string (in text mode) or bytes object (in binary mode). `size` is an optional numeric argument. When `size` is omitted or negative, the entire contents of the file will be read and returned; it's your problem if the file is twice as large as your machine's memory. Otherwise, at most `size` characters (in text mode) or `size` bytes (in binary mode) are read and returned. If the end of the file has been reached, `f.read()` will return an empty string (`''`).

### **readline() :**

`f.readline()` reads a single line from the file; a newline character (`\n`) is left at the end of the string, and is only omitted on the last line of the file if the file doesn't end in a newline. This makes the return value unambiguous; if `f.readline()` returns an empty string, the end of the file has been reached, while a blank line is represented by `\n`, a string containing only a single newline.

### **os.listdir(path='.') :**

Return a list containing the names of the entries in the directory given by `path`. The list is in arbitrary order, and does not include the special entries `'.'` and `'..'` even if they are present in the directory. If a file is removed from or added to the directory during the call of this function, whether a name for that file be included is unspecified. `path` may be a path-like object. If `path` is of type `bytes` (directly or indirectly through the `PathLike` interface), the filenames returned will also be of type `bytes`; in all other circumstances, they will be of type `str`. This function can also support specifying a file descriptor; the file descriptor must refer to a directory. Raises an auditing event `os.listdir` with argument `path`.

## **Code 3.a : Read the contents of the file**

```
f = open("text.txt", "r")
content = f.read()
print(content)
f.close()
```

## **Output 3.a :**

```
● student@csiot-ThinkCentre-M70s:~/CHETAN_I_007/PythonLab/03Practical$ python3 readFile.py
Humpty Dumpty sat on a wall,
Humpty Dumpty had a great fall;
All the king's horses and all the king's men
Couldn't put Humpty together again.
```

## **Code 3.b : Count the number of characters of the file**

```
f = open("text.txt", "r")
content = f.read()
num_chars = len(content)
print("The number of characters in the file is:", num_chars)
f.close()
```

**Output 3.b :**

```
● student@csiot-ThinkCentre-M70s:~/CHETAN_I_007/PythonLab/03Practical$ python3 numOfChar.py
The number of characters in the file is: 141
```

**Code 3.C : Count the number of lines in file.**

```
f = open("text.txt", "r")
lines = sum(1 for line in f)
print("The number of lines in the file is:", lines)
f.close()
```

**Output 3.C :**

```
● student@csiot-ThinkCentre-M70s:~/CHETAN_I_007/PythonLab/03Practical$ python3 numOfLines.py
The number of lines in the file is: 4
```

**Code 3.D : Count the number of lines in file containing a specific word.**

```
f = open("text.txt", "r")
count = 0
for line in f:
    if line.startswith("Humpty"):
        count += 1
print("The number of lines that start with Humpty is:", count)
f.close()
```

**Output 3.D :**

```
● student@csiot-ThinkCentre-M70s:~/CHETAN_I_007/PythonLab/03Practical$ python3 numOfSLines.py
The number of lines that start with Humpty is: 2
```

**Code 3.E : List the files in current directory.**

```
import os
dir_list = os.listdir(".")
for item in dir_list:
    if os.path.isfile(item):
        print(item)
```

**Output 3.E :**

```
● student@csiot-ThinkCentre-M70s:~/CHETAN_I_007/PythonLab/03Practical$ python3 filesInDir.py
numOfSLines.py
numOfLines.py
readFile.py
text.txt
filesInDir.py
numOfChar.py
```

**Conclusion :**

From this practical we learn how to access files and directories. in Python.