---: File Handling :---

File handling is used to store data permanently in a file.

Basic operations:-

1. **open('file_name with extention','mode')** (default extention and mode are .txt and r)
2. Perform **read()/write()** operations.
3. **close()** the file.

Type of files:

1. **text file(.txt) -** It stores data/characters in ASCII form.
2. **binary(.dat) -** It is used to store audio, video, or image.
3. **csv(.csv) –** It is used to store data in key value format.

Mode:-

1. **'r' :- read mode**(default mode)-It open file in read mode. If file not exist then it give error of FileNotFoundError: [Errno 2] No such file or directory: filename.
2. **'w' :- write mode** – It open file in write mode and if in file previous data exist then it override with new data. If file not exist, it create new file.
3. **'a' :- append mode** - It open file in append mode and if in file, previous data exist then cursor position in last of the previous data. If file not exist, it create new file.
4. **'x' :- exclusive mode(Create mode)** – This mode is used to create a new file only.

File object attributes –

1. **closed:** It returns true if the file is closed and false when the file is open.
2. **encoding:** Encoding used for byte string conversion.
3. **mode:** Returns file opening mode
4. **name:** Returns the name of the file which file object holds.
5. **newlines:** Returns "\r", "\n", "\r\n", None or a tuple containing all the newline types seen.

open	read mode	write mode	close
open()	read(n)	write()	closed
	read()	writelines()	close()
	readline()	writable()	
	readlines()		
	readable()		

open("file_name","mode")		
If file exist	**Mode**	**If file not exis**
Give error: file already exist.	**"x"**	Created a new file with given name.
Write mode, cursor present in zero index position. That means previous data will be destroyed.	**"w"**	Created a new file with given name.
Normal as read mode.	**"r"**	Give error: file not exist.
Normal as append mode with cursor position ahead of previous data.	**"a"**	Created a new file with given name.

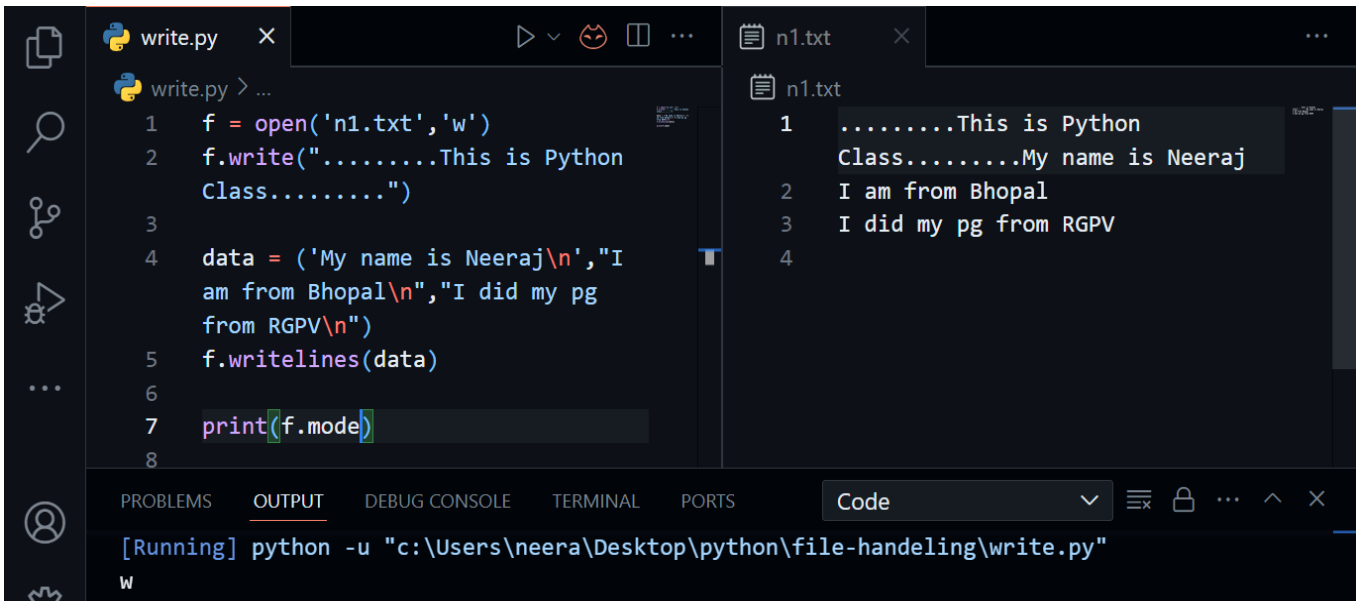
Create Mode examples:--

1. **if file was not exist, then it create a new file with mention name**

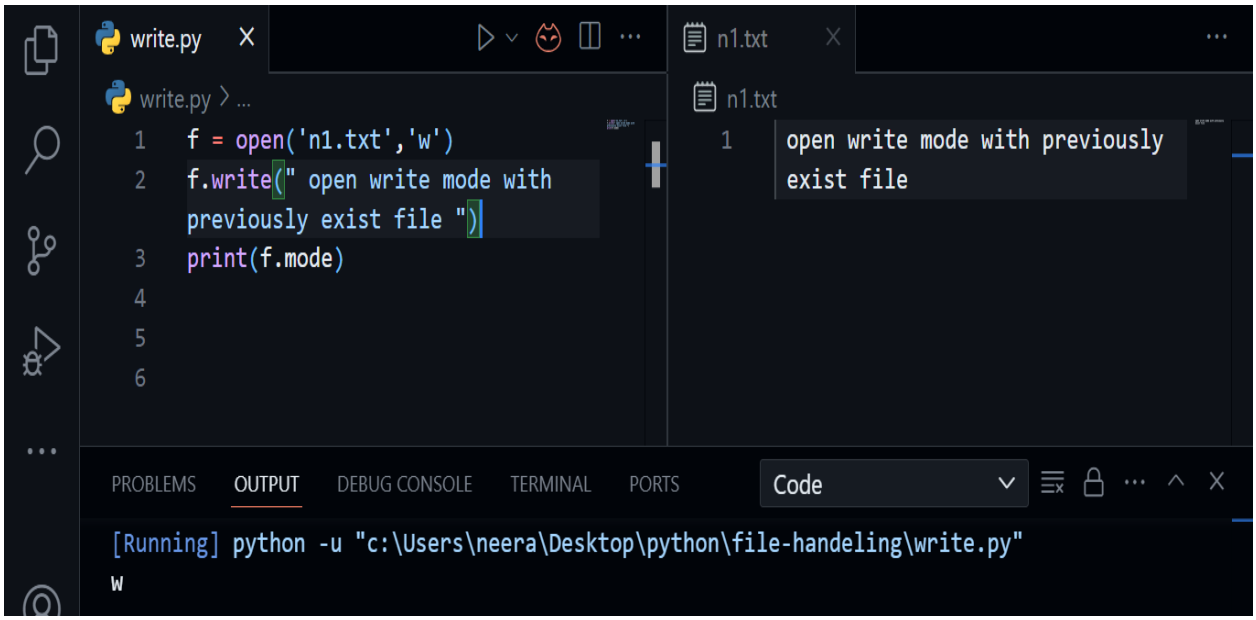
2. **If file already exist, then it give error that already exist.**

Write mode example:---

1. **if file was not exist, then it create a new file with mention name**

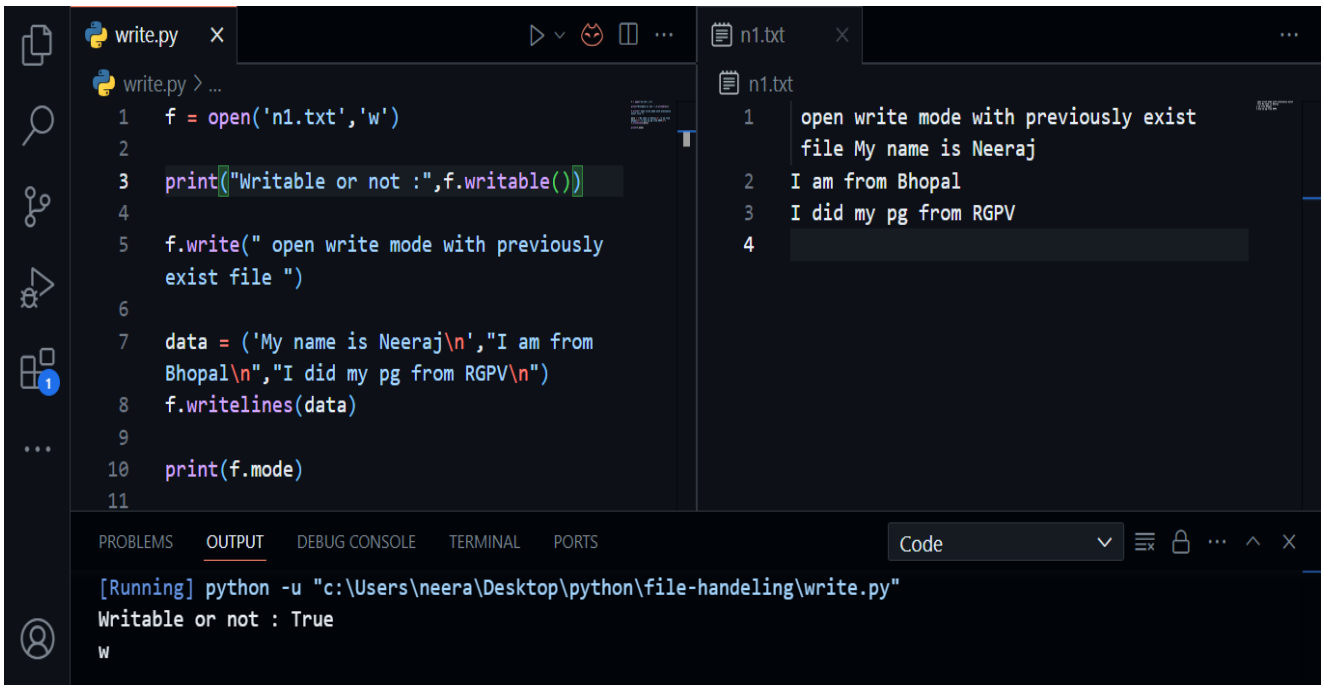


2. Write mode, cursor present in zero index position. That means previous data will be destroyed.



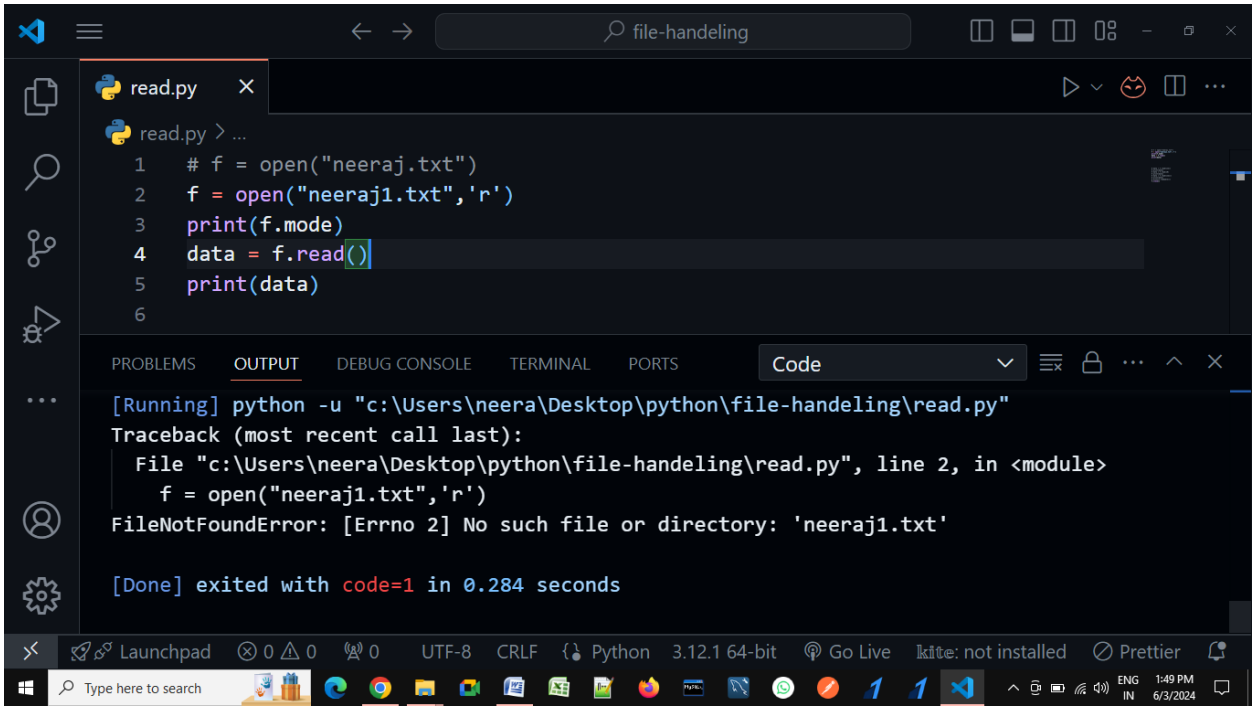
Write mode methods:

1. **write() – it is used to write single line of data.**
2. **writelines() – It is used for multiple lines of data**
3. **writable() – To check file is writable or not.**



Read mode:--

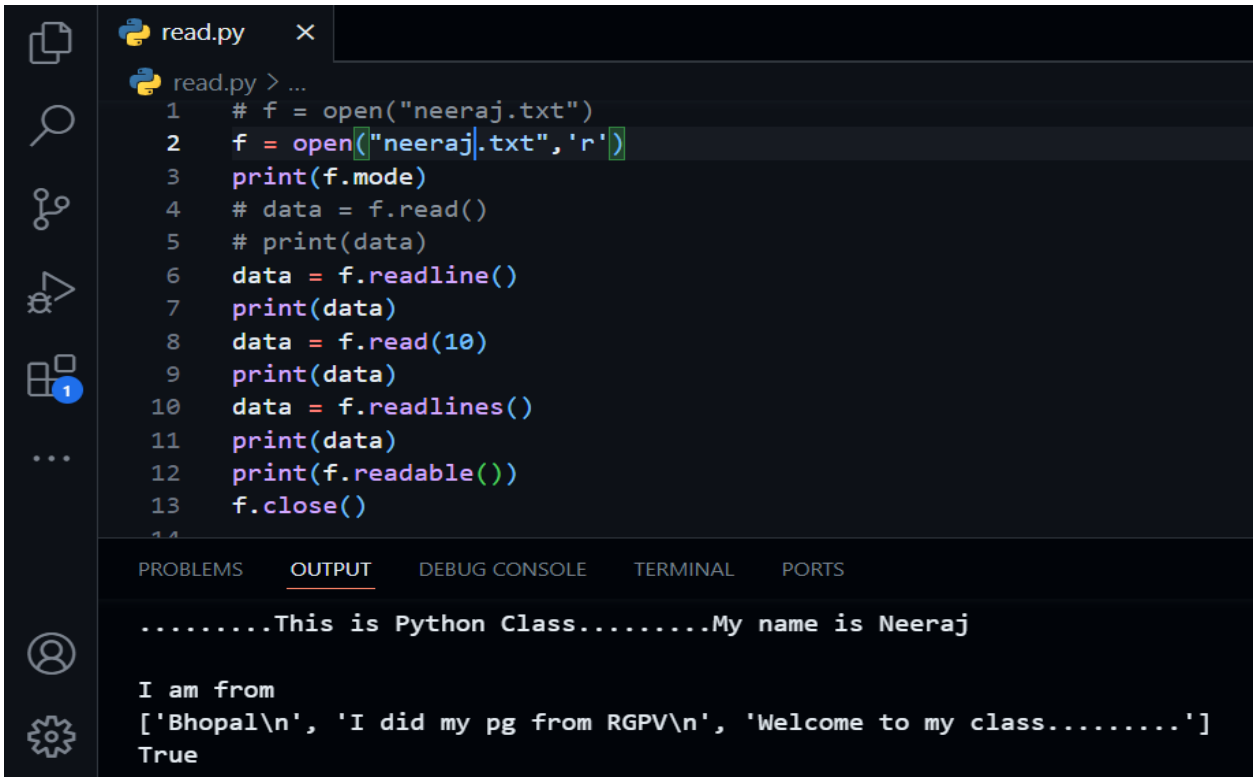
1. Give error: file not exists.



2. **If File exists, then Normal as read mode**

Read mode methods :------

1. **read(n)**
2. **read()**
3. **readline()**
4. **readlines()**
5. **readable()**



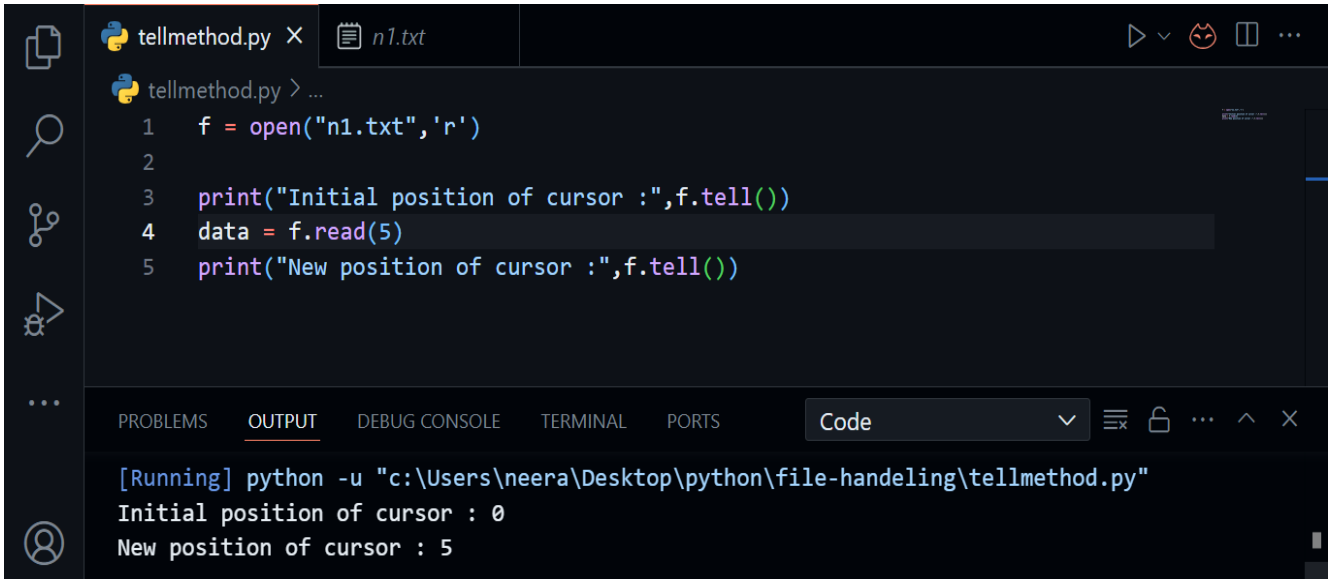
Delete data, file, or folder with python:---

```
import os, shutil

os.remove('new1/n4.txt')
print(".....n4 file deleted .....")

os.remove('n4.txt')
print(".....n3 file deleted.....")

os.mkdir("new2")
print(".....new1 folder created.....")
```

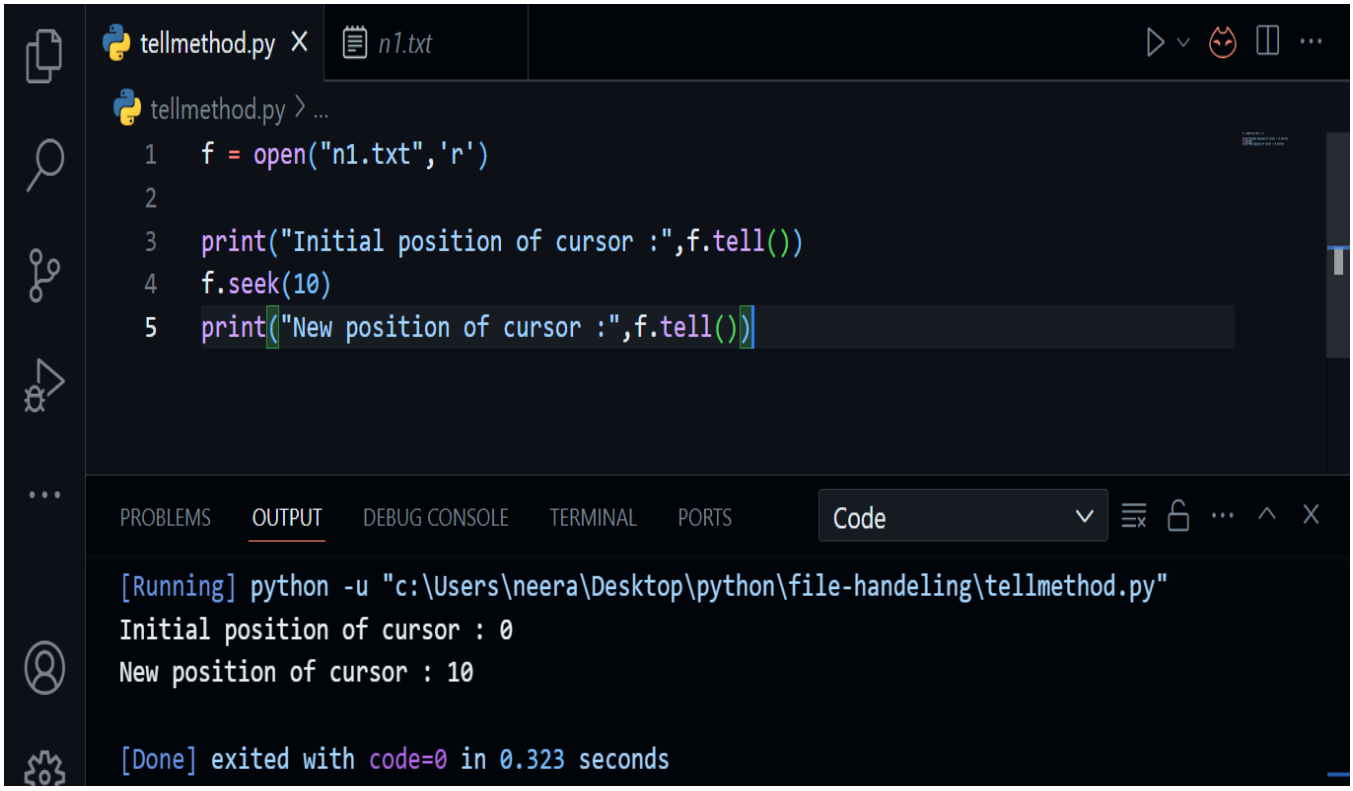
```
os.chdir("new2")
print(".....change fron one directory to another directory.....")

x = os.getcwd()
print(x)

os.chdir("neeraj")
print(".....change fron one directory to another directory.....")

# get current working directory.....
x = os.getcwd()
print(x)

f = open('new1/n4.txt','a')
print(".....create new files within the new1 folder.....")

os.rmdir('new1')
print(".....Delete empty folder.....")

shutil.rmtree('new1')
print(".....Delete empty folder.....")

os.rename('new1',"neeraj")
print(".....Rename Folder name.....")

os.rename('n1.txt',"neeraj.txt")
print(".....Rename File name.....")
```

tell() & seek()

tell() : With the help of tell() we find out the current position of cursor.



seek() : With the help of seek() method, we can move cursor from our required positions.



Syntax:--

seek(attribute1, attribute2)

attribute1 : Where we want our cursor

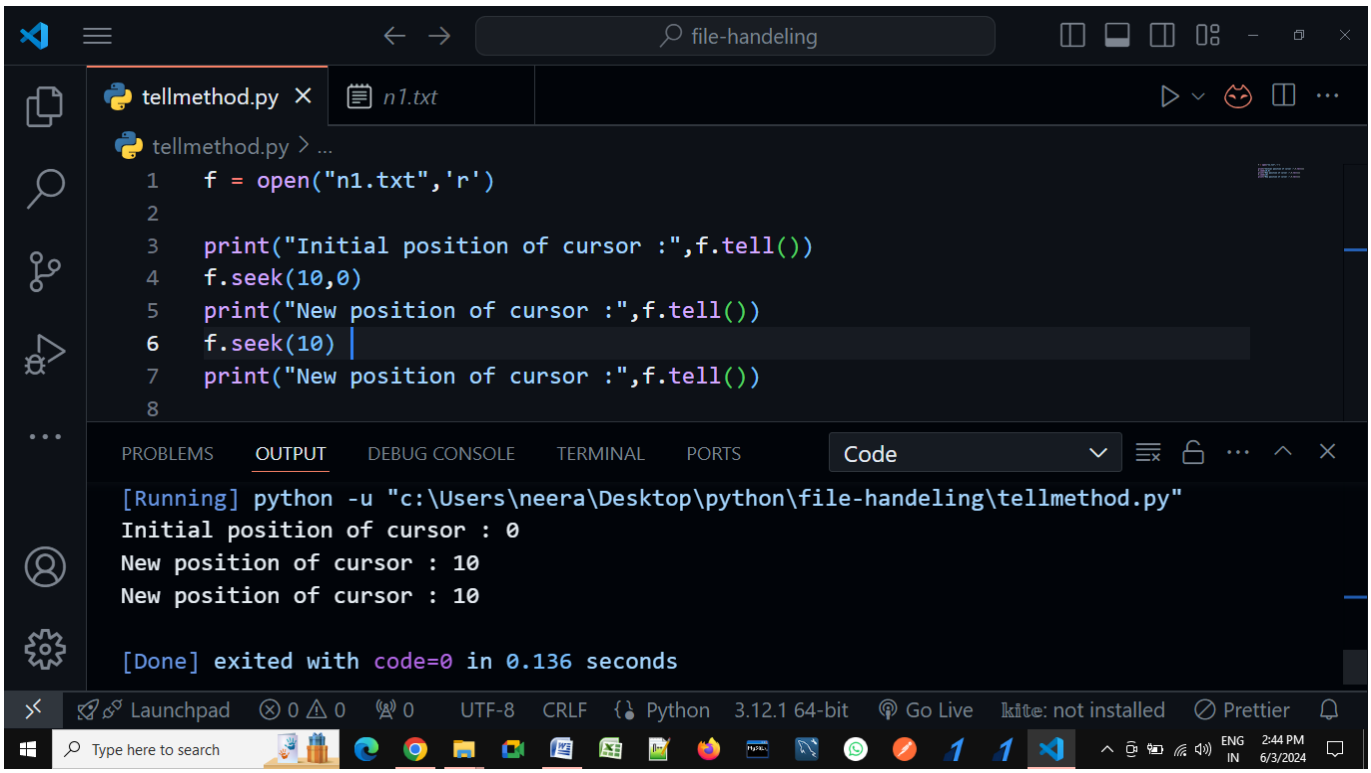
attribute2: Start from which position

1. 0 (start from beginning)- by-default that means not required to write.
2. 1 (start from current position)
3. 2 (start from last position(for negative indexing))

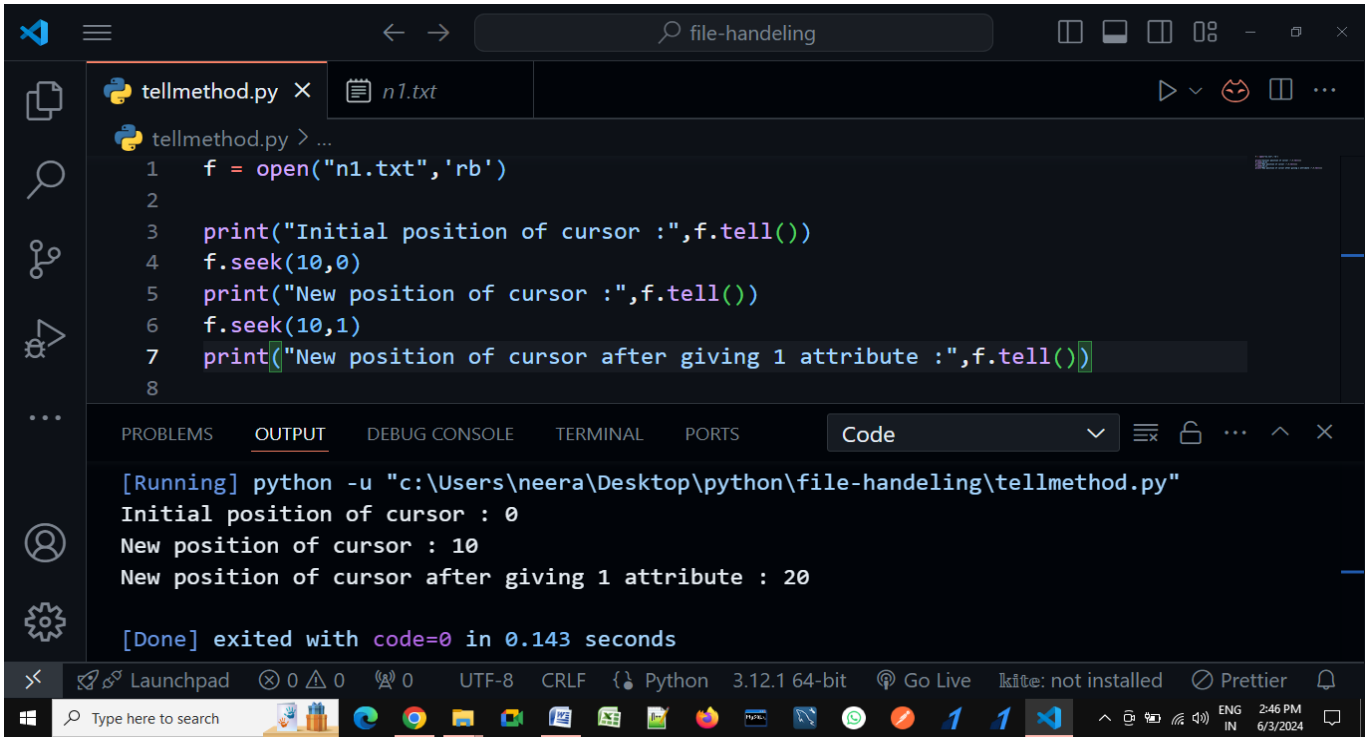
Note: In python 3.2, 1 and 2 both are used only in binary mode

Examples:----

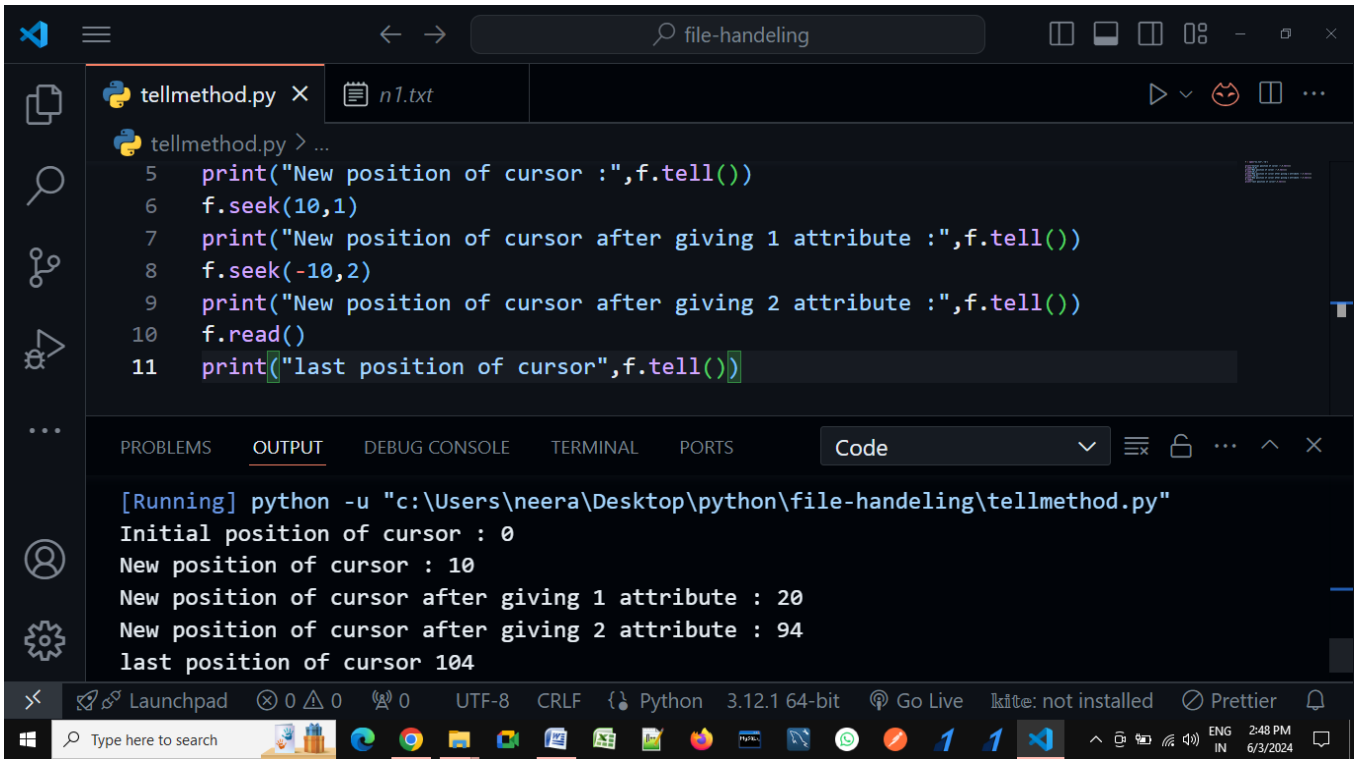
Attribute2 : 0 (start from beginning)----------



Attribute2 : 1 (start from current position)



Attribute2 : 2 (start from last position(for negative indexing))



Binary file:

Store non-simple objects like dictionaries, tuple, list, sets, nested-list objects. First all objects are serializes and after that it stores in binary_files (like .dat ,.pkl/.pickle,.exe, .bin etc).

|----- **pickling** --> serialize --> objects converted into byte-stream.

|----- **write** --->pickle.dump(data,file)

import pickle --

|----- **unpickling** ->deserialize->byte-stream converted into objects.

|----- **read** ---> pickle.load(file)

Pickling prosess:-----

```
import pickle


f = open('firstb.exe','ab+')
data = ['neeraj','jai','rahul','vishnu']
pickle.dump(data,f)
print("Data inserted successfully.....")
f.close()

O/P:--
Data inserted successfully.....
```

Unpickling prosess:-----

```
import pickle

f = open('firstb.exe','rb+')
data = pickle.load(f)
print(data)
f.close()
```

```
O/P:-
['neeraj', 'jai', 'rahul', 'vishnu']
```