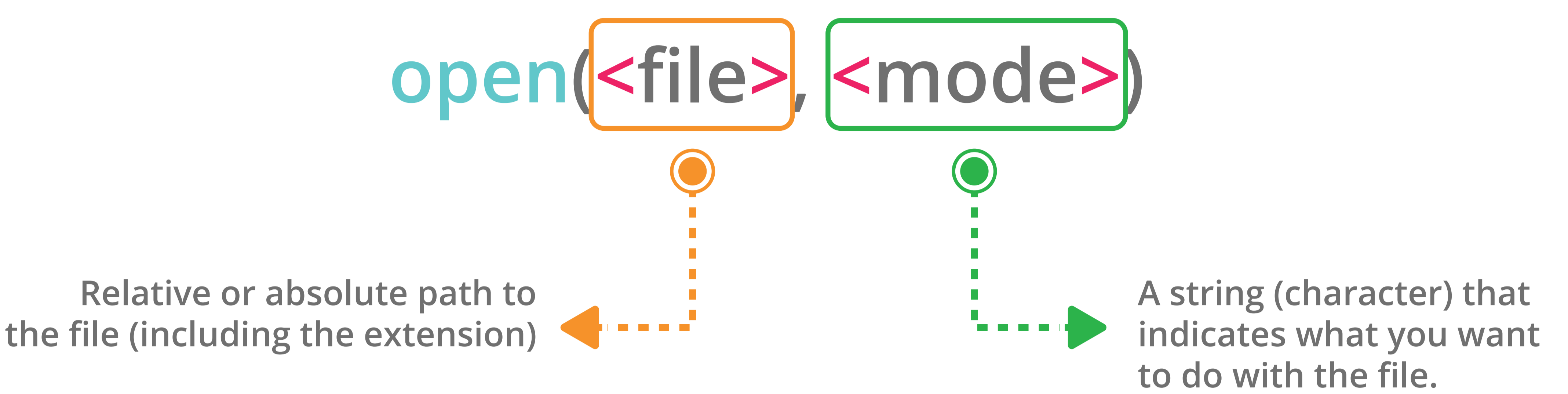


To read a text file in Python, you follow these steps:

- First, open a text file for reading by using the `open()` function
- Second, read text from the text file using the file `read()`, `readline()`, or `readlines()` method of the file object.
- Third, close the file using the file `close()` method. This frees up resources and ensures consistency across different python versions



Method	Description
<code>writable()</code>	Returns whether the file can be written to or not
<code>readable()</code>	Returns whether the file stream can be read or not
<code>read()</code>	Returns the file content
<code>readline()</code>	Returnsone line from the file
<code>readlines()</code>	Returnsa list of lines from the file
<code>write()</code>	Writes the specified string to the file
<code>writelines()</code>	Write a list of strings to the file
<code>close()</code>	Closes the file
<code>flush()</code>	Flushes the internal buffer
<code>seek()</code>	Change the file position
<code>tell()</code>	Returnsthe current file
<code>truncate()</code>	Resizes the file to a specified

Reading file

```
# Reading the txt file
file_name = "authors.txt"
file = open(file_name, "r")
content = file.read()
print(content)
```

English,Charles Severance
English,Sue Blumenberg
English,Elloitt Hauser
Spanish,Fernando TardÃfo MuÃfÂ±iz

```
# Printing the path of file
print(file.name)
# Printing the mode of file
print(file.mode)
# Printing the file with '\n' as a new file
print(content)
# Printing the type of file
print(type(content))
```

authors.txt
r
English,Charles Severance
English,Sue Blumenberg
English,Elloitt Hauser
Spanish,Fernando TardÃfo MuÃfÂ±iz

<class 'str'>

```
# Close the file
file.close()

# Verificati on of the closed file
file.closed
```

True

Another way to read a file

Keyword File Object Variable that will be used to refer to the file object

```
with open("<file>", "<mode>") as <var>:  
    #Do what you need with the file
```

```
...  
# Verification of the closed file  
file.closed
```

True

```
...  
# See the content of the file  
print(content)
```

English,Charles Severance
English,Sue Blumenberg
English,Elloitt Hauser
Spanish,Fernando TardÃfo MuÃfÃ±iz

```
...  
# Reading the first 20 characters in the text file  
fname = 'authors.txt'  
with open(fname, 'r') as f:  
    print(f.read(20))
```

English,Charles Seve

```
...  
# Reading certain amount of characters in the file  
fname = 'authors.txt'  
with open(fname, 'r') as f:  
    print(f.read(10))  
    print()  
    print(f.read(20))  
    print()  
    print(f.read(50))  
    print()  
    print(f.read(100))
```

English,Ch

arles Severance
Engl

ish,Sue Blumenberg
English,Elloitt Hauser
Spanish,

Fernando TardÃfo MuÃfÃ±iz

```
...  
# Reading first line in the text file  
with open(fname, 'r') as f:  
    print('The first line is: ', f.readline())
```

The first line is: English,Charles Severance

```
...  
# Difference between read() and readline()  
with open(fname, 'r') as f:  
    print(f.readline(10))  
    print()  
    print(f.read(20)) # This code returns the next 20 characters in  
    the line.
```

English,Ch

arles Severance
Engl

Loop usage in the text file

```
...  
with open(fname, 'r') as f:  
    line_number = 1  
    for line in f:  
        print('Line number', str(line_number), ':', line)  
        line_number+=1
```

Line number 1 : English,Charles Severance

Line number 2 : English,Sue Blumenberg

Line number 3 : English,Elloitt Hauser

Line number 4 : Spanish,Fernando TardÃfo MuÃfÃ±iz

Methods

read(n) function

- Reads atmost n bytes from the file if n is specified, else reads the entire file.
- Returns the retrieved bytes in the form of a string.

```
with open(fname, 'r') as f:
    print(f.read())
```

English,Charles Severance
English,Sue Blumenberg
English,Elloitt Hauser
Spanish,Fernando TardÃfo MuÃfÂ±iz

```
with open(fname, 'r') as f:
    print(f.read(30))
```

English,Charles Severance
Engl

readline() function

Reads one line at a time from the file in the form of string

readlines() function

Reads all the lines from the file and returns a list of lines.

```
with open(fname, 'r') as f:
    content=f.readlines()
    print(content)
```

['English,Charles Severance\n', 'English,Sue Blumenberg\n', 'English,Elloitt Hauser\n', 'Spanish,Fernando TardÃfo MuÃfÂ±iz\n']

strip() function

Removes the leading and trailing spaces from the given string.

```
with open(fname, 'r') as f:
    len_file = 0
    total_len_file = 0
    for line in f:
        # Total length of line in the text file
        total_len_file = total_len_file+len(line)
        # Lenght of the line after removing leading and trailing spaces
        len_file = len_file+len(line.strip())
    print(f'Total lenght of the line is {total_len_file}.')
    print(f'The length of the line aft er removing leading and trail-
ing spaces is {len_file}.')
```

Total lenght of the line is 106.
The length of the line aft er removing leading and trailing spaces is 102.

Size of the text file

```
with open(fname, 'r') as f:
    str = ""
    for line in f:
        str+=line
    print(f'The size of the text file is {len(str)}..')
```

The size of the text file is 26.
The size of the text file is 49.
The size of the text file is 72.
The size of the text file is 106.

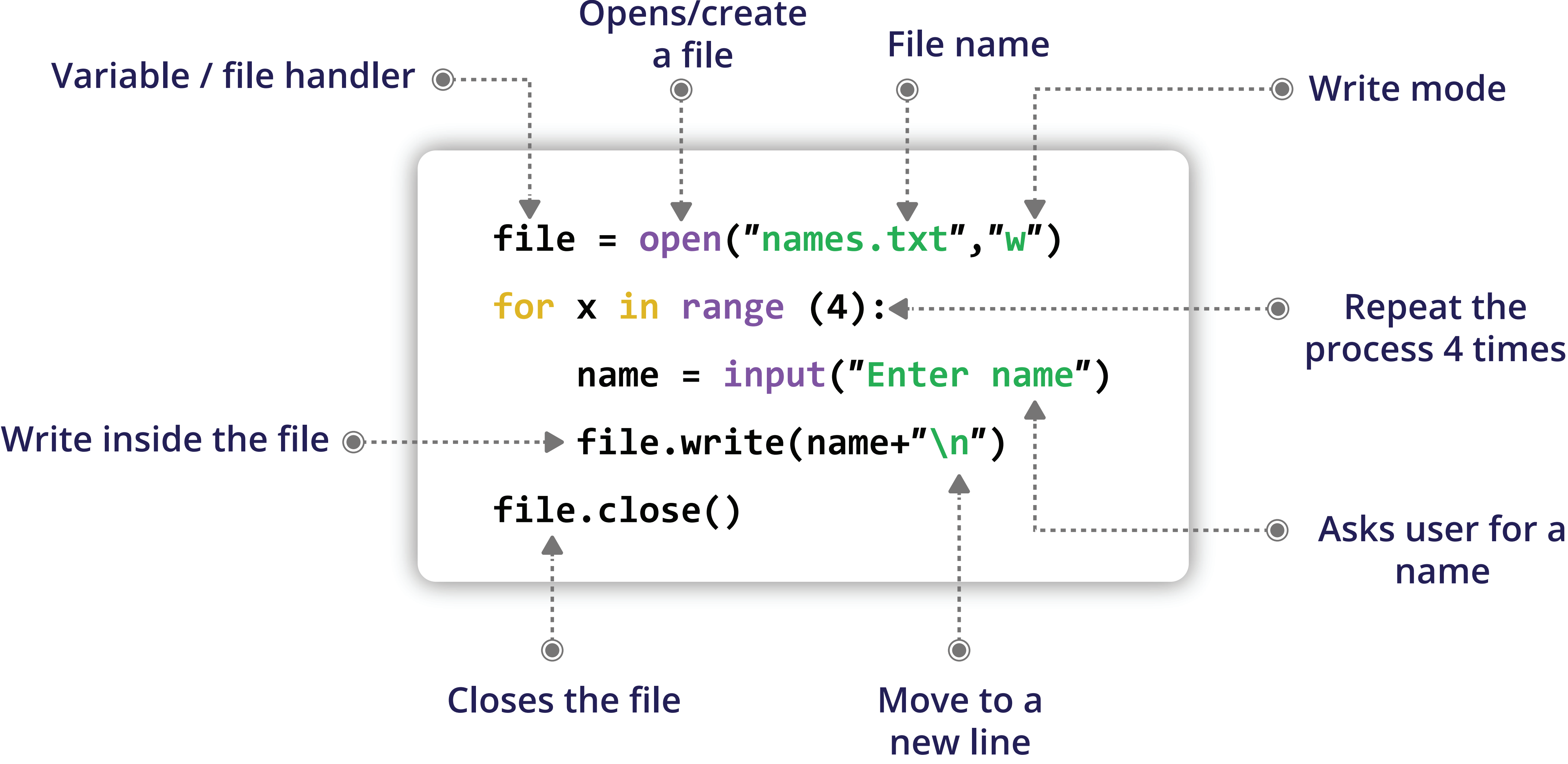
Number of lines in the text

```
with open(fname, 'r') as f:
    count = 0
    for line in f:
        count = count + 1
    print(f'The number of lines in the text file is {count}.')
```

The number of lines in the text file is 1.
The number of lines in the text file is 2.
The number of lines in the text file is 3.
The number of lines in the text file is 4.

Writing Files in Python To write to a text file in Python, you follow these steps:

- First, open the text file for writing (or appending) using the open() function.
- Second, write to the text file using the write() or writelines() method.
- Third, close the file using the close() method.



Character	Function
r	Open file for reading only. Starts reading from beginning of file. This default mode.
rd	Open a file for reading only in binary format. Starts reading from beginning of file.
r+	Open file for reading and writing. File pointer placed at\beginning of the file.
w	Open file for writing only. File pointer placed at beginning of the file. Overwrites existing file and creates a new one if it does not exists.
wb	Same as w but opens in binary mode.
w+	Same as w but also allows to read from file.
wb+	Same as wb but also allows to read from file.
a	Open a file for appending. Starts writing at the end of file. Creates a new file if file doesnot exist.
ab	Same as a but in binary format. Creates a new file if file does not exist.
a+	Same as a but also open for reading.
ab+	Same as ab but also open for reading.

```
# Writing lines to a file.
fname = 'pcr_file.txt'
with open(fname, 'w') as f:
    f.write("I dedicate this book to Nancy Lier Cosgrove Mullis.\n")
    f.write("Jean-Paul Sartre somewhere observed that we each of us
make our own hell out of the people around us.")

# Checking the file whether it was written or not
with open(fname, 'r') as f:
    content = f.read()
    print(content)
```

I dedicate this book to Nancy Lier Cosgrove Mullis.
Jean-Paul Sartre somewhere observed that we each of us make our own hell
out of the people around us.

Appending files

```
# Wrting and then reading the file
new_file = 'pcr_file.txt'
with open(new_file, 'w') as f:
    f.write('Overright\n')
with open(new_file, 'r') as f:
    print(f.read())
```

Overright

Other modes

a+

Appending and Reading. Creates a new file, if none exists.

```
fname = 'pcr_file.txt'
with open(fname, 'a+') as f:
    f.write("From F. Lee Bailey\n")
```

```
# To verify the text file whether it is added or not
with open(fname, 'r') as f:
    print(f.read())
```

Overright
From F. Lee Bailey

tell() and seek() functions with a+

```
with open(fname, 'a+') as f:
    print("First location: {}".format(f.tell())) # it returns the
current position in bytes
    content = f.read()
    if not content:
        print('Read nothing.')
    else:
        print(f.read())

    f.seek(0, 0)

    """
    seek() function is used to change the position of the File Handle
to a given specific positi on.
    File handle is like a cursor, which defines from where the data has to be read or written in
the file.
    Syntax: f.seek(offset, from_what), where f is file pointer
    Parameters:
    Offset: Number of positions to move forward
    from_what: It defines point of reference.
    Returns: Return the new absolute position.
    The reference point is selected by the from_what argument. It accepts three
values:
    0: sets the reference point at the beginning of the file
    1: sets the reference point at the current file position
    2: sets the reference point at the end of the file
    """

    print('\nSecond locati on: {}'.format(f.tell()))
    content = f.read()
    if not content:
        print('Read nothing.')
    else:
        print(content)
    print('Locati on aft er reading: {}'.format(f.tell()))
```

First location: 31
Read nothing.

Second locati on: 0
Overright
From F. Lee Bailey

Locati on aft er reading: 31

•r+ Reading and writing. Cannot truncate the file.

```
with open(fname, 'r+') as f:
    content=f.readlines()
    f.seek(0,0) # writing at the beginning of the file
    f.write('From The San Diego Union-Tribune' + '\n')
    f.write("Refreshing ... brashly confident ... indisputably entertaining." + "\n")
    f.write("To my family..." + '\n')
    f.seek(0,0)
    print(f.read())
```

From The San Diego Union-Tribune
Refreshing ... brashly confident ... indisputably entertaining.
To my family...
From The San Diego Union-Tribune
Refreshing ... brashly confident ... indisputably entertaining.
To my family...

Copy the file

```
# Let's copy the text file 'pcr_file.txt' to another one 'pcr_file_1.txt'
fname = 'pcr_file.txt'
with open(fname, 'r') as f_reading:
    with open('pcr_file_1.txt', 'w') as f_writing:
        for line in f_reading:
            f_writing.write(line)
```

```
# For the verificati on, execute the following codes
fname = 'pcr_file_1.txt'
with open(fname, 'r') as f:
    print(f.read())
# Now, there are 2 files from the same file content.
```

From The San Diego Union-Tribune
Refreshing ... brashly confident ... indisputably entertaining.
To my family...
From The San Diego Union-Tribune
Refreshing ... brashly confident ... indisputably entertaining.
To my family...

```
# Writi ng the student names into a file
fname = open(r'student_name.txt', 'w')
for i in range(3):
    name = input('Enter a student name: ')
    fname.write(name)
    fname.write('\n') # To write names as a new line
fname = open(r'student_name.txt', 'r')
for line in fname:
    print(line)
fname.close()
```

Enter a student name: mahesh
Enter a student name: mahesh
Enter a student name: mahesh
mahesh

mahesh

Mahesh

```
lines = ['Hello, World!', 'Hi, Python!']
with open('new_file.txt', 'w') as f:
    for line in lines:
        f.write(line)
        f.write('\n')
with open('new_file.txt', 'r') as f:
    print(f.read())
```

Hello, World!
Hi, Python!

```
# Add more lines into the file
more_lines = ['Hi, Sun!', 'Hello, Summer!', 'Hi, See!']
with open('new_file.txt', 'a') as f:
    f.writelines('\n' .join(more_lines))
with open('new_file.txt', 'r') as f:
    print(f.read())
f.close()
```

Hello, World!
Hi, Python!
Hi, Sun!
Hello, Summer!
Hi, See!