

Python



```
info = open("newfile.txt", "w")
```

```
text = "We are adding a text string into the file"
```

```
info.write(text)
```

Python File Tell() Method

```
obj = open("demo.txt", "w")
```

```
content = obj.write("Python")
```

```
print("Read line in the specified file: ", content)
```

```
obj.close()
```

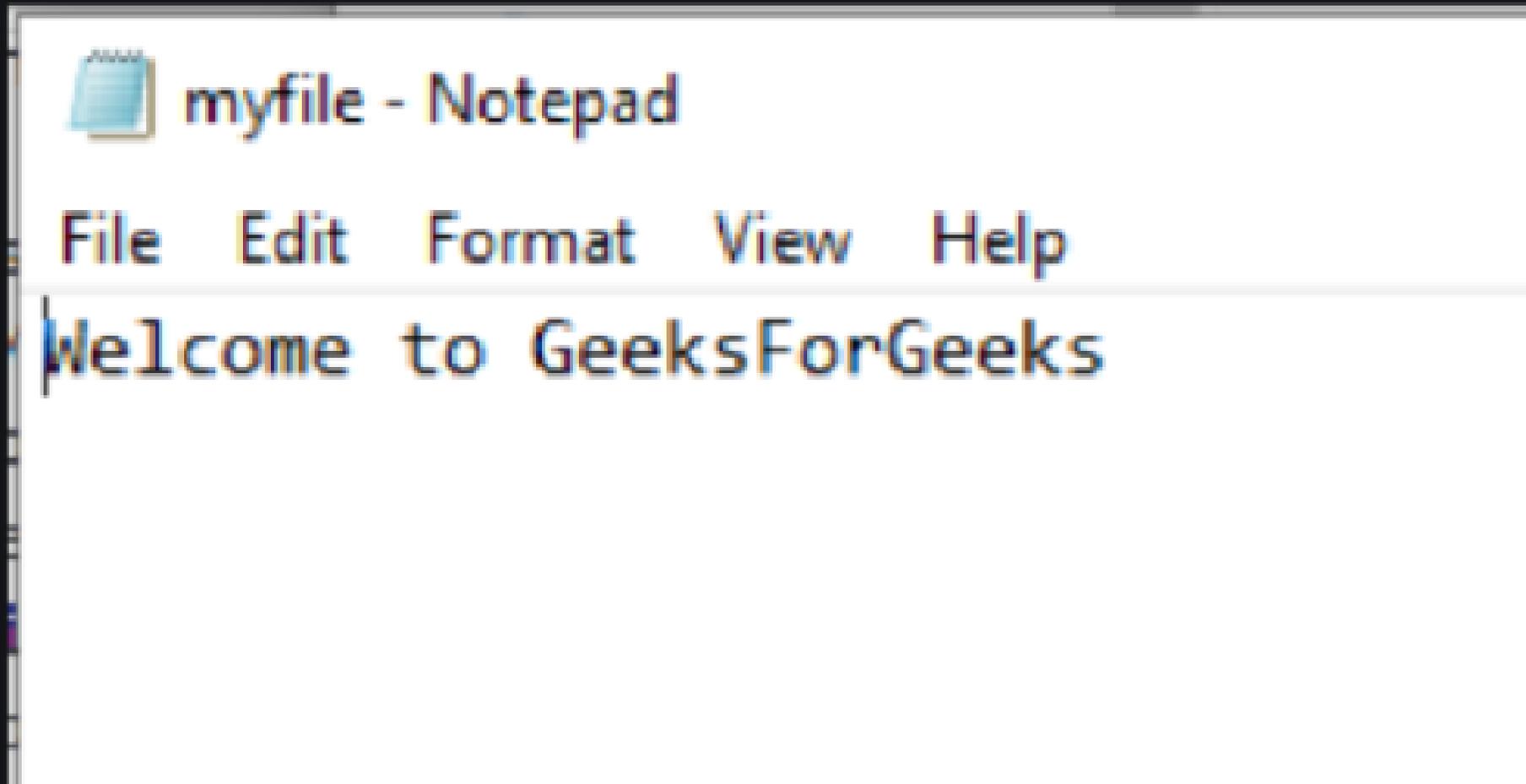
tell() method:

Access modes govern the type of operations possible in the opened file. It refers to how the file will be used once it's opened. These modes also define the location of the **File Handle** in the file. **File handle** is like a cursor, which defines from where the data has to be read or written in the file. Sometimes it becomes important for us to know the position of the File Handle. `tell()` method can be used to get the position of File Handle. `tell()` method returns current position of file object. This method takes no parameters and returns an integer value. Initially file pointer points to the beginning of the file(if not opened in append mode). So, the initial value of `tell()` is zero.

syntax :

```
file_object.tell()
```

Let's suppose the text file named "myfile" looks like this:



Example 1: Position of File Handle before reading or writing to file.

Python3

```
# Python program to demonstrate
# tell() method

# Open the file in read mode
fp = open('myfile.txt', 'r')

# Print the position of handle
print(fp.tell())

#Closing file
fp.close()
```

output :

0

Example 2: Position of File Handle after reading data from file.

Python3

```
# Python program to demonstrate  
# tell() method  
  
# Opening file  
fp = open('sample.txt', 'r')  
fp.read(8)  
  
# Print the position of handle  
print(fp.tell())  
  
# Closing file  
fp.close()
```

Output :

8

Example 3: For binary files. Let's create a binary file and we will notice the position of handle before writing and after writing to binary file.

Python3

```
 # Python program to demonstrate  
# tell() method  
  
# for reading binary file we  
# have to use 'wb' in file mode.  
 fp = open('sample2.txt', 'wb')  
 print(fp.tell())  
  
# Writing to file  
fp.write(b'1010101')  
  
print(fp.tell())  
  
# Closing file  
fp.close()
```

Output :

0

7