Python Session
Summary 19-06-2023

File handling

- File handling allows you to perform various operations on files, including writing, reading, and appending content.

- To read the content of a file, you can open the file in read mode ("r"), use the read() method to retrieve the content, and to write content to a file, you can open the file in write mode ("w") or append mode ("a") using the open() function.

- **Reading a file**

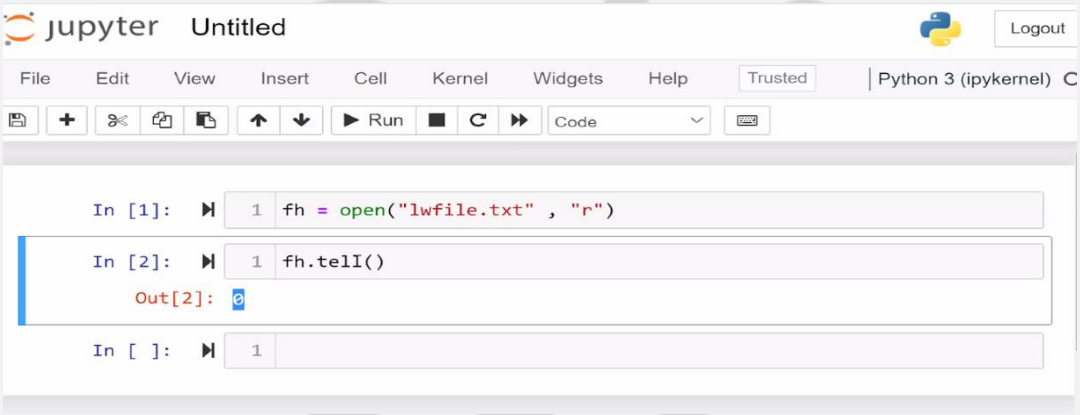
  - To open a file we use the open() function

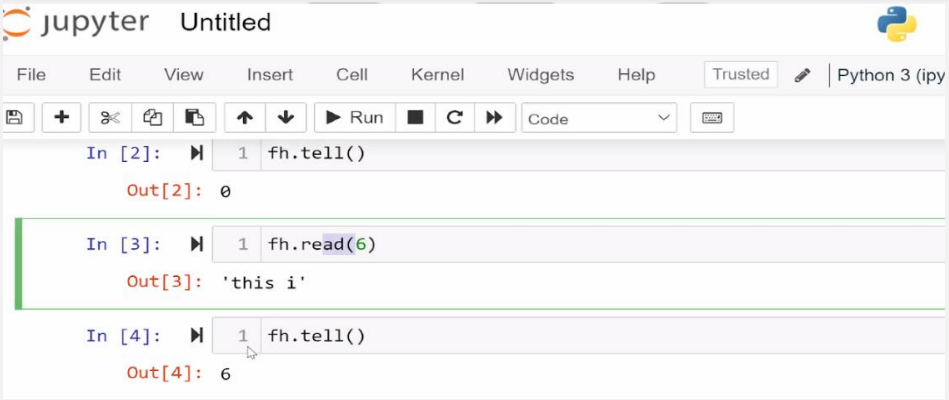
  - The "r" mode is used to open the file in read-only mode, allowing you to read the contents of the file but not modify or delete anything inside.

- When you open a file, Python creates a file object, which acts as a file handler. This file object allows you to perform various operations on the file, such as reading, writing, or closing the file.

- To determine the current position within the file, you can use the tell() method. The tell() method returns the current file position. This can be helpful if you need to keep track of your position while reading or writing the file.

- Here's an example of how you can open a file in read-only mode, read its contents, and use the tell() method to get the current file position:



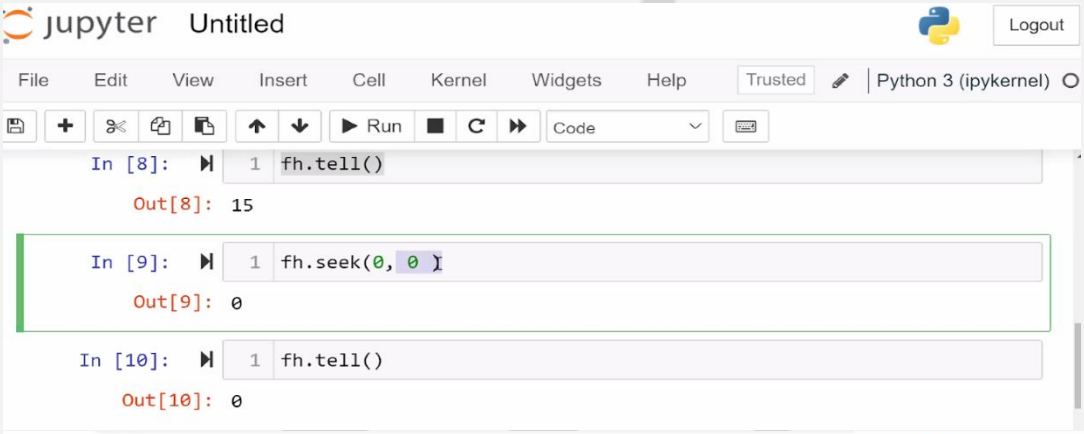


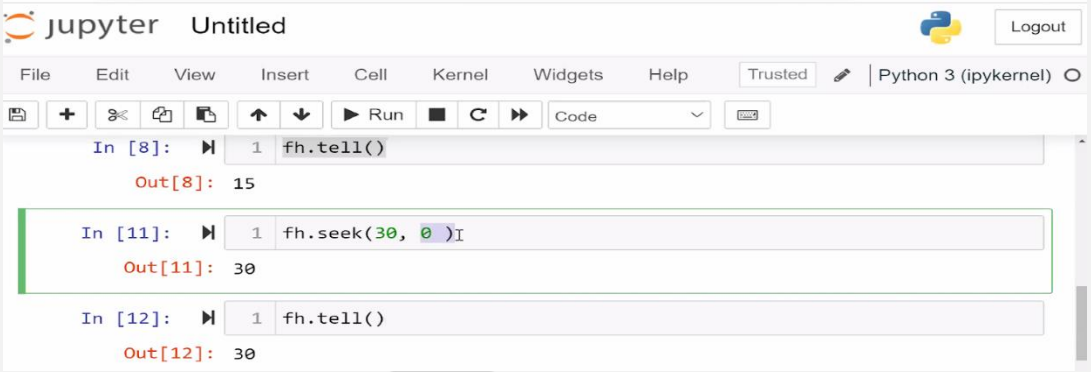
- **Seek() function**

  - The seek() function allows you to move the cursor to a specific position in the file, which can be useful for random access or when you want to read data from a specific location.

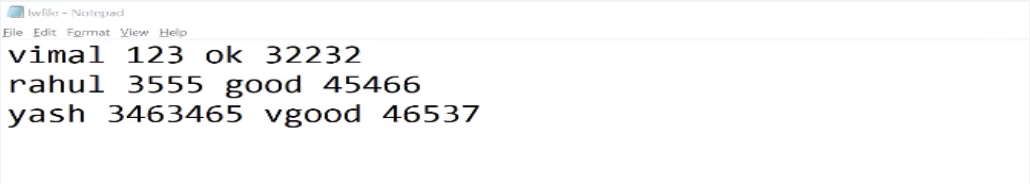
- The seek() function takes two arguments: offset and whence. The offset specifies the number of bytes to move the file position, and the whence parameter determines the reference point from where the offset is applied. Here are the three possible values for whence:

  0 (default): The offset is relative to the beginning of the file.
  1: The offset is relative to the current file position.
  2: The offset is relative to the end of the file.





- In Python, when working with text, characters are indeed treated individually. You can iterate over a string or a file line by line, character by character, or word by word, depending on your specific requirements.

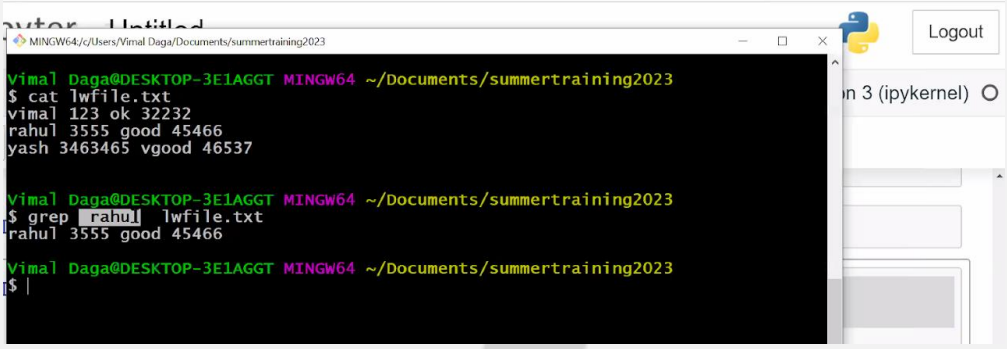


```
In [29]:  myfile = open("lwfile.txt")

In [30]:  myfile.read()

Out[30]: 'vimal 123 ok 32232\nrahul 3555 good 45466\nyash 3463465 vgood 46537
\n\n'

In [28]:  myfile.close()
```

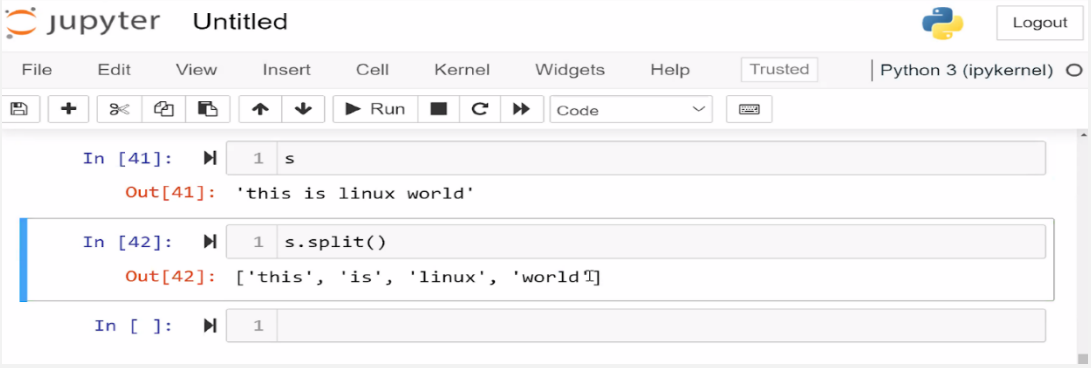
- If you want to traverse a file by line, you can iterate over the file object using a for loop.

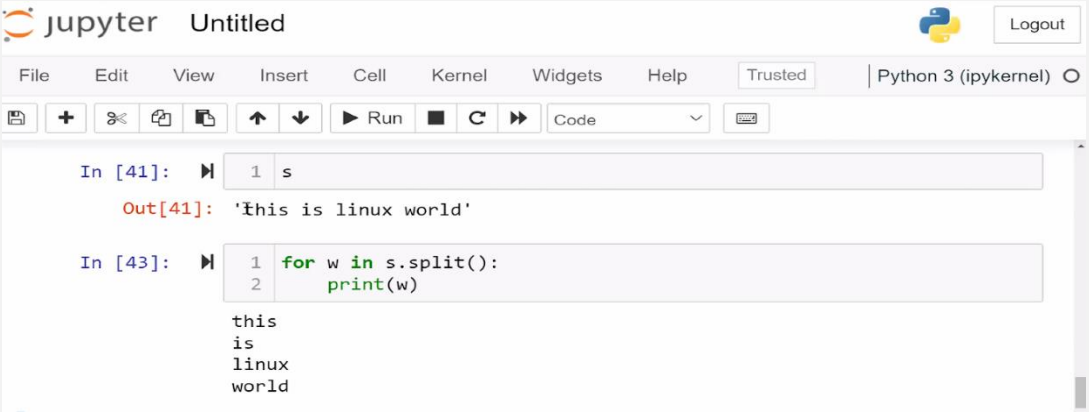
```
In [34]:  myfile = open("lwfile.txt")

In [ ]:  for  x in myfile:
             print(x)
```

- In Linux, the grep command is commonly used to search for specific words or patterns in files.

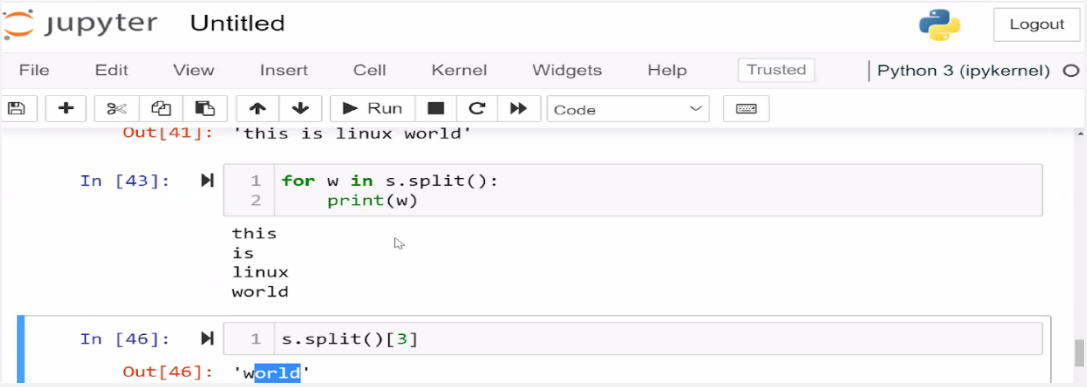


- If you want to replicate the behavior of the grep command in Python and search for a specific word in a string or a file, you can use the split() function to split the string into individual words and then check if the desired word is present.





- Words can also be extracted by position.

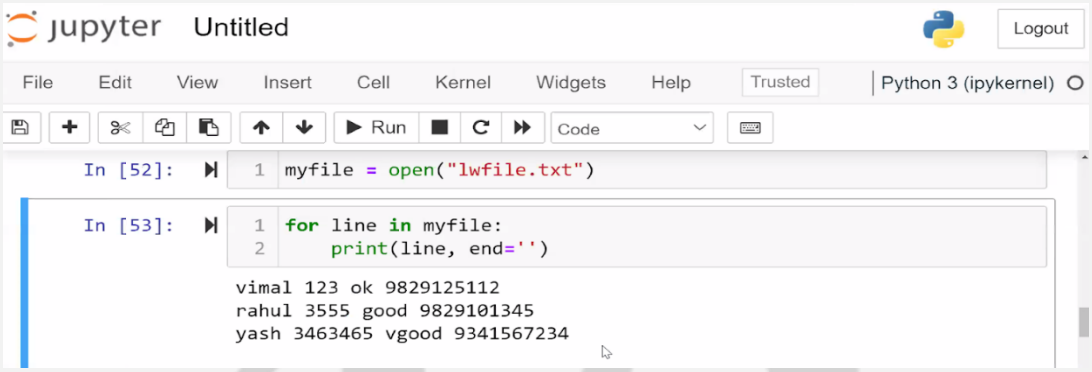


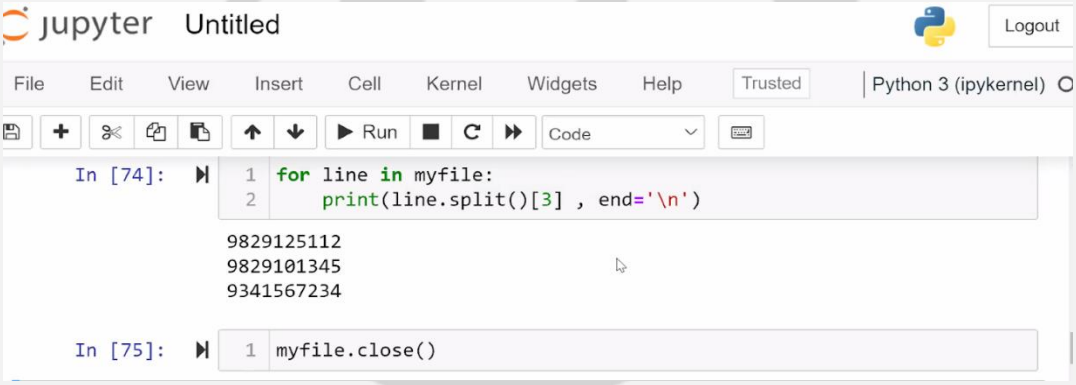
- By default, the print() function in Python adds an end-of-line character ('\n') at the end of the printed output.

- If you want to change the default end-of-line behavior, you can specify the end parameter of the print() function. By default, end='\n', but you can

change it to any other character or string. For example, to print without a newline character, you can use:

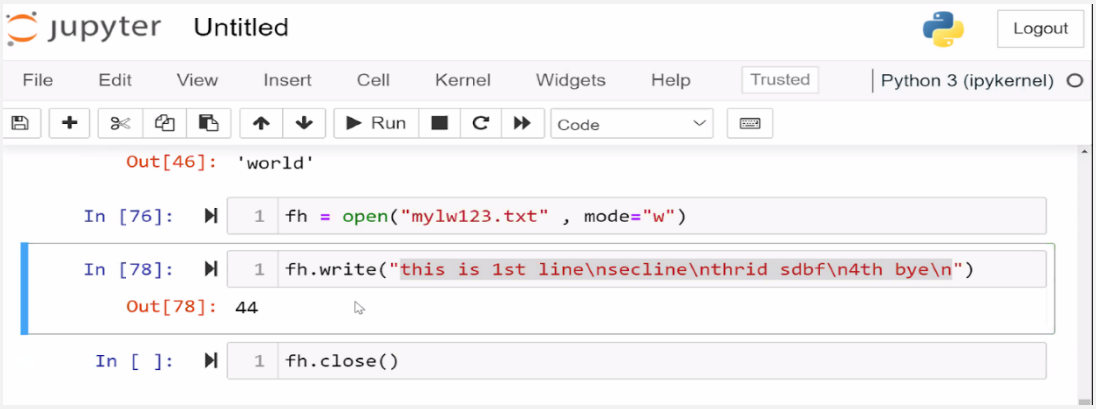
print("Hello", end="")





- The open() function is used to open the file in write mode ("w"). The file object is assigned to the variable fh

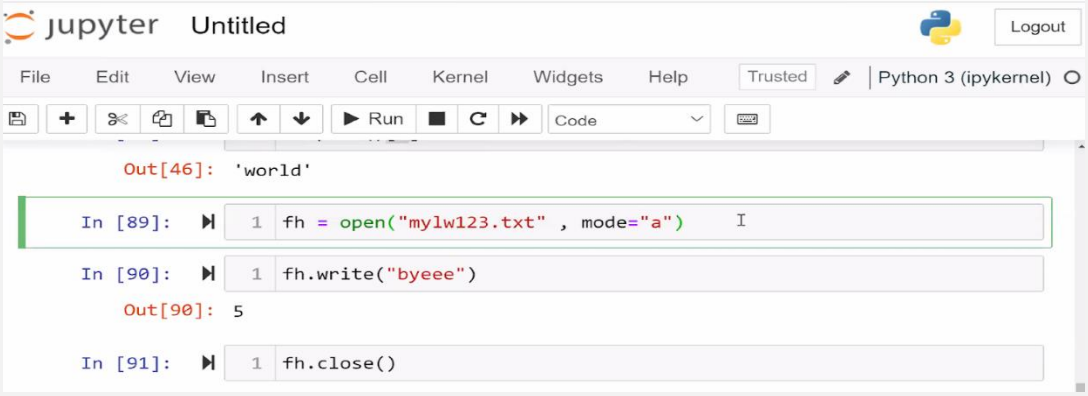
- The write() function is then used to write the string to the file. This will overwrite any existing content in the file.



- We close the file using the close() function. It's essential to close the file after writing to ensure that all changes are saved and resources are freed.

Steps to append data to a File in Python

- **Step 1 –** Check if the file exists in the specified Path & it has written permissions.
- **Step 2** – Open the file in append mode
  - ➢ You first need to open the file in append mode. You can do it with the open() function.
  - ➢ When opening the file, you should specify the file name and the mode in which you want to open the file.
  - ➢ To open a file in append mode, use the 'a' mode.
- **Step 3** – Append data to file



- **Step 4** – Close the file

- If you want to perform both write and read operations on a file using file handling in Python, you can use the r+ mode when opening the file. The r+ mode allows you to read from and write to the file.