**FILE HANDLING IN COMPUTER SCIENCE**

The file handling plays an important role when the data needs to be stored permanently into the file. A file is a named location on disk to store related information. We can access the stored information (non-volatile) after the program termination.

The file-handling implementation is slightly lengthy or complicated in the other programming language, but it is easier and shorter in Python.

48.6M

832

History of Java

In Python, files are treated in two modes as text or binary. The file may be in the text or binary format, and each line of a file is ended with the special character.

Hence, a file operation can be done in the following order.

* Open a file
* Read or write - Performing operation
* Close the file

## Opening a file

Python provides an ****open()**** function that accepts two arguments, file name and access mode in which the file is accessed. The function returns a file object which can be used to perform various operations like reading, writing, etc.

****Syntax:****

1. file object = open(<file-name>, <access-mode>, <buffering>)

The files can be accessed using various modes like read, write, or append. The following are the details about the access mode to open a file.

1. fileptr = open("file.txt","r")
3. **if** fileptr:
4. **print**("file is opened successfully")

## The close() method

Once all the operations are done on the file, we must close it through our Python script using the ****close()**** method. Any unwritten information gets destroyed once the ****close()**** method is called on a file object.

We can perform any operation on the file externally using the file system which is the currently opened in Python; hence it is good practice to close the file once all the operations are done.

The syntax to use the ****close()**** method is given below.

1. fileptr = open("file2.txt", "w")
3. # appending the content to the file
4. fileptr.write('''''Python is the modern day language. It makes things so
5. It is the fastest-growing programing language''')
7. # closing the opened the file
8. fileptr.close()