File

Files are very essential for storing and sharing information. Various access modes are used to define who a file has to be used, once it is **open**. Filehandle which works similarly to the cursor helps in this purpose. It helps to define the point to read and write data within a particular file.

Open()

In order to read, write, or append to the file, first, it needs to be opened using the open() function. The open() function has almost 8 parameters and among them, 2 are most important: filename and access mode. The file needs to be in the same directory as the running python program (.py file). The file argument represents the file path in your system.

File_object = open(file= "File_Name", mode="Access_Mode")

There are a total of 6 access modes in python and for this course, we will be only discussing the most used ones.

Read Only ('r'): Opens a text file for reading and the handle is pointed at the start of the file. If the file does not exist or the given path is wrong, it raises **`FileNotFoundError`** error.
Write Only ('w'): Opens a text file for writing and the handle is pointed at the start of the file.
If the file does not exist, then create a new file. If the file exists, then **overwrites** its contents.
Append ('a'): Opens a text file for writing and the handle is pointed at the end of the file. If the file does not exist, then create a new file. If the file exists, then adds the new contents at the end of the old contents.
Exclusive creation ('x'): Opens a text file for exclusive creation. The handle is pointed at the start of the file. If the file does not exist, then create a new file. If the file exists, then raises a **`File exists`** **error.**

Text file used in the "reading from a file" example
Filename: `"Sample_text.txt"` **File contents:** We love Python programming. After we have mastered Python programming, should we call ourselves "Python programmers" or "Snake charmers?" Coding in Python is very easy. Coding in Python takes less time. This is the BRACU CSE110 course.

Reading from a file

There are three ways to read from a file.

fh.read([n])	
Here, n is the number of characters. If n is not mentioned by default print the whole text.	
Code	**Output**

`fh = open("Sample_text.txt", "r")`
`print(fh.read())`
	`We love Python programming.`
	`After we have mastered Python programming,`
	`should we call ourselves "Python programmers"`
	`or "Snake charmers?"`
	`Coding in Python is very easy.`
	`Coding in Python takes less time.`
	`This is the BRACU CSE110 course.`

`fh = open("Sample_text.txt", "r")` `print(fh.read(5))`	`We lo`

fh.readline([n])

Here, n is the number of characters. If n is not mentioned, then by default prints the line pointed by the Filehandle. Prints only 1 line at a time.

Code	Output
`fh = open("Sample_text.txt", "r")` `print(fh.readline())`	`We love Python programming.`

`fh = open("Sample_text.txt", "r")`
`print(fh.readline())`
`print(fh.readline())`
	`We love Python programming.`
	
	`After we have mastered Python programming,`

`fh = open("Sample_text.txt", "r")` `print(fh.readline(10))`	`We love Py`

fh.readlines()

Returns a list with each line of the file as its elements.

Code	Output

`fh = open("Sample_text.txt", "r")`
`lines_in_list = fh.readlines()`
`print(lines_in_list)`
 	
`for line in lines_in_list:`
`  print(line)`
	`['We love Python programming.\n', 'After we have mastered Python programming,\n', 'should we call ourselves "Python programmers"\n', 'or "Snake charmers?"\n', 'Coding in Python is very easy.\n', 'Coding in Python takes less time.\n', 'This is the BRACU CSE110 course.\n']`
	
	`We love Python programming.`
	`After we have mastered Python programming,`
	`should we call ourselves "Python programmers"`
	`or "Snake charmers?"`
	`Coding in Python is very easy.`
	`Coding in Python takes less time.`
	`This is the BRACU CSE110 course.`

Closing Python Files with close()

When the opened file is not required in the program anymore or it has to be operated in a different mode than the current one, then the file is closed. **The close() function is used to close the file and free the memory obtained by the file.**

Code	Output
`fh = open("Sample_text.txt", "r")` `fh.close()`	`No output is shown`

Writing to a file

There are two ways to write in a file: write(s) and writelines(S).

fh.write(s)	
Write the single line s at the end of the file	
Code	Output

`fh = open("Sample_text.txt", "w")`
`str1 = "Writing this new line.\nThis is 2nd line \nThis is 3rd line"`
`fh.write(str1)`
`fh.close()`
 	
`#changing the mode of operation, write to read`
`fh = open("Sample_text.txt", "r")`
`print(fh.read())`
`fh.close()`
	`Writing this new line.`
	`This is 2nd line`
	`This is 3rd line`

fh.writelines(S)	
Here, S is a sequence of strings. Writes each element of S as a separate line to file.	
Code	Output
`fh = open("Sample_text.txt", "w")` `str1 = "Writing this new line."`	`Writing this new line.This is the 2nd line.This is the 3rd`

`str2 = "This is the 2nd line."`
`str3 = "This is the 3rd line."`
 	
`List_of_lines = [str1, str2, str3]`
`fh.writelines(List_of_lines)`
`fh.close()`
 	
`#changing the mode of operation, write to read`
`fh = open("Sample_text.txt", "r")`
`print(fh.read())`
`fh.close()`
	`line.`

Appending to a file

For appending, we open the file in append mode (‘a). Then, the writing to the file is similar to the ‘write only mode’. We use the write() function to append additional lines to the file. If the file does not exist, then a new file is created with the given contents.

Appending using the fh.write(s) [when the file exists]	
appending the single line s at the end of the file	
Code	Output

`#viewing the file contents`
`fh = open("Sample_text.txt", "r")`
`print(fh.read())`
`fh.close()`
 	
`print("==================")`
 	
`#changing the mode of operation from read to append`
`fh = open("Sample_text.txt", "a")`
`str1 = "\nAdding an additional line"`
`fh.write(str1)`
`fh.close()`
 	
`print("==================")`
 	
`#changing the mode of operation from write to read`
`fh = open("Sample_text.txt", "r")`
	`We love Python programming.`
	`After we have mastered`
	`Python programming,`
	`should we call ourselves`
	`"Python programmers"`
	`or "Snake charmers?"`
	`Coding in Python is very easy.`
	`Coding in Python takes less time.`
	`This is the BRACU CSE110 course.`
	
	`==================`
	`==================`
	`We love Python programming.`
	`After we have mastered`
	`Python programming,`
	`should we call ourselves`

`print(fh.read())`
`fh.close()`
	`"Python programmers"`
	`or "Snake charmers?"`
	`Coding in Python is very easy.`
	`Coding in Python takes less time.`
	`This is the BRACU CSE110 course.`
	
	`Adding an additional line`

Appending using the fh.write(s) [when the file does not exist]	
It will create a new file, and work similar to write only mode.	
Code	Output

`fh = open("Sample2.txt", "a")`
`str1 = "This file does not exist\nso it will create a new file"`
`fh.write(str1)`
`fh.close()`
 	
`print("==========")`
 	
`#changing the mode of operation from write to read`
`fh = open("Sample2.txt", "r")`
`print(fh.read())`
`fh.close()`
	`==========`
	`This file does not exist`
	`so it will create a new file`

Appending using the fh.writelines(s)	
here, s is a sequence of multiple lines.	
Code	Output

`#viewing the file contents`
`fh = open("Sample2.txt", "r")`
`print(fh.read())`
`fh.close()`
	`This file does not exist`
	`so it will create a new file`
	
	`==========`

`print("==================")`
 	
`fh = open("Sample2.txt", "a")`
`lines = ["it can also write\n",`
`"multiple lines\n", "from a list\n"]`
`fh.writelines(lines)`
`fh.close()`
 	
`print("==================")`
 	
`#changing the mode of operation from write to read`
`fh = open("Sample2.txt", "r")`
`print(fh.read())`
`fh.close()`
	`==================`
	`This file does not exist`
	`so it will create a new file`
	`it can also write`
	`multiple lines`
	`from a list`

Exclusive creation of files

Writing using the fh.write(s) in Exclusive creation ('x') mode [when the file exists]	
Here, in the example, the file `Sample2.txt` already exists, so it raises an error.	
Code	Output

`fh = open("Sample2.txt", "x")`
`line = ["the file exists"]`
`fh.write(line)`
`fh.close()`
	`File exists: 'Sample2.txt'`
	
	`[SHOWS AN ERROR]`

Writing using the fh.write(s) in Exclusive creation ('x') mode [when the file does not exist]	
Here, in the example, the file `Sample5.txt` does not exist. So the code is working without error.	
Code	Output

`fh = open("Sample5.txt", "x")`
`line = ["the file does not exist\n",`
`"so it is creating a new file\n"]`
`fh.writelines(line)`
`fh.close()`
	`the file does not exist`
	`so it is creating a new file`

`#changing the mode of operation from write to read`
`fh = open("Sample5.txt", "r")`
`print(fh.read())`
`fh.close()`	

Style Guide for Python Code

For every programming language, there are few coding conventions followed by the coding community of that language. All those conventions or rules are stored in a collected document manner for the convenience of the coders, and it is called the “Style Guide” of that particular programming language. The provided link gives the style guidance for Python code comprising the standard library in the main Python distribution.

Python style guide link: https://www.python.org/dev/peps/pep-0008/