**Level 1: File Handling Definitions**

Use the following resources to answer the questions about file handling in Python.

* <https://www.pythonforbeginners.com/files/reading-and-writing-files-in-python>
* <https://www.pythonforbeginners.com/cheatsheet/python-file-handling>

1. Explain the function of each of the following file handling commands
   1. The open() function-

The open () function is the function in which it is used to open a file for writing or use in Python. This will return a file object, so it will most be commonly used with two arguments. The argument is nothing more than a value that has been provided to a function, which is relayed when you call it. So, for instance, if we declare the name of a file as “Test File” that name would considered to be an argument.

* 1. The read() method

The file.read() reads the string from the open file. This command will display all the text inside the file, and add.

* 1. The readline() method

This function only reads one line from the text.

* 1. The write() method

The write() method writes any string to an open file. This is the string that you want to be written.

* 1. The close() method

Flushes any unwritten information and closes the file object. This closes the entire file completely, terminating resources in use, in turn freeing them up for the system to deploy elsewhere.

1. Research and explain the “Mode” used to open files in a Python program.
   1. ‘r’ mode

The Read mode is used when the file is only being read.

* 1. ‘w’ mode

The Write mode is used to edit and write new information to the opened file.

* 1. ‘a’ mode

Appending mode used to add new types of data towards the file

* 1. ‘r+’ mode

Special read and write mode, which handles both read and write actions when creating the file.

* 1. Explain when and where the mode is used in a Python program

The mode is used in a Python programming whenever you want to use a command like write information or read information from the file.

The first line sets “fh” in which it opens the file, the second line reads and prints the file and the third line closes the file.

1. Provide example code which opens a text file for reading and prints the contents of the file to the console display.
   1. Explain what each line of the program does.

fh=open("Welcome.txt","r")

print(fh.read())

fh.close

The first line sets “fh” in which it opens the file, the second line reads and prints the file and the third line closes the file.

1. Provide example code which opens a text file for writing and writes some data to the file.
   1. Explain what each line of the program does.
2. fh=open("Welcome.txt","w")
3. fh.write("Good Morning")
4. fh.write("Good Night")
5. fh.close()

The fh=(“Welcome.txt”,”w”). This turns the file into the “write mode”. I then used fh.write and made two messages and then used fh.close() to close the file.

1. Research and explain the difference between a “File Name” (type Python string) and   
   a File Object (type Python object).

File object is an object returned by a call to open, a filename is the name of a file.

**Level 2: Reading & Writing Files**

1. Add a text file to your project as follows:
   * Click on “Add File” icon in the files pane/window.
   * Type “myfile.txt” and return.
   * “myfile.txt” is now open in the editor pane/window.
   * Type some text into “myfile.txt”
   * Make sure to add several lines of text. A sample file contents could look like:

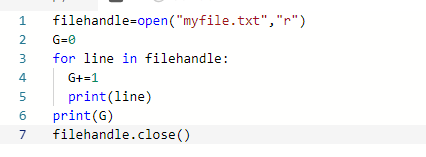
*Hello kind student*

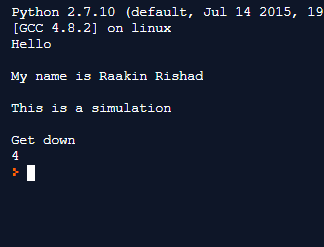
*This is a message from your computer*

*I hope you are having fun learning to program*

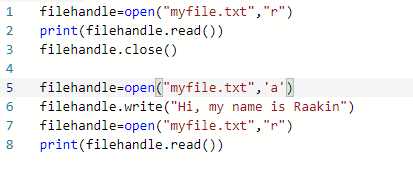
*Remember to ask Mr. Nestor questions when you don’t understand*

1. Write a program that opens “myfile.txt” for reading and prints the contents to the file to the console display.
   1. The program should also print out the number of lines in the file
   2. Provide a listing of your program below



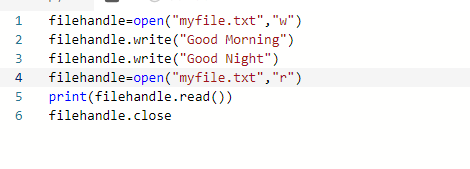


1. Write a program that opens “myfile.txt” for appending new contents to the file.
   1. You can “hard code” some commands to write new text to the file
   2. Make sure to use the close() method when your are finished.   
      (What happens if you don’t?)
   3. How can you tell that your program worked? (That the contents changed?)
   4. Provide a listing of your program below



If you do not close the file once you’re finished the entire file, it can be changed or modified by others. The program worked because the first 3 lines of the program prints the original text. The last line prints the text after it has been modified which means that it will display a change in the entire text.

1. Write a program that opens “myfile.txt” for writing new contents to the file.
   1. You can “hard code” some commands to write new text to the file
   2. Explain the difference between appending and writing to a file.
   3. Provide a listing of your program below



**Level 3: Folders & Binary Files**

1. Add a folder called “resources” to your project as follows:
   * Click on “Add Folder” icon in the files pane/window.
   * Type “resources” and return.
2. Drag and drop your “myfile.txt” file into the “resources” folder.
3. Run you program from Level 2 to see what happens.
   1. Why does it give an error?
   2. How can you modify the file name string used by the open() function so that it also includes the “resources” folder?
   3. Fix the open() function so that the program runs correctly and provide your program listing below.

fh=open("resources/myfile.txt","r")

print(fh.read())

fh.close()

This creates a path to the file by using a slash towards the file, and also puts it on read.

1. Research and explain the “Binary Mode” used to open files in a Python program.
   1. What is the ‘rb’ mode and how is it different from the ‘r’ mode

“Rb” mode is when the entire file is opened in binary mode and it opens it to be for reading in the binary mode.

* 1. What is the ‘wb’ mode and how is it different from the ‘w’ mode

The Wb mode is when the file opens in binary mode. It opens to writing the file.

1. Add the “Penguin.bmp” binary image file to your repl project as follows:
   1. Download the “Penguin.bmp” file from the GitHub repository to your desktop
   2. Drag and drop the “Penguin.bmp” from your desktop to the “resources” folder in your repl project
   3. Click on the “Penguin.bmp” to make sure everything is ok.
2. Modify your Level 2 program to open the “Penguin.bmp” and print its contents to the screen.
   1. Provide a listing of your modified code below
   2. Explain what you see as output compared to the penguin image itself

filehandle=open("resources/Penguin.bmp","rb")

print(fh.read())

filehandle.close()

When the program has run, there will be different types of numbers and symbols instead of a picture of a penguin.