**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 open the files you want to open.

* 1. The read() method

Reads a string from an open file

* 1. The readline() method

Read one line from the file

* 1. The write() method

Writes any string to an open file

* 1. The close() method

Flushes any unwritten information and closes the file object

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

Read mode Used when the file is only being read

* 1. ‘w’ mode

Write mode used to edit and write new information to the file

* 1. ‘a’ mode

Appending mode used to add new data to the end of the file

* 1. ‘r+’ mode

Special read and write mode, which is used to handle both actions when working with a file

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

Whenever you want to use a command like write to write something into the file or read to read the file and etc.

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.
2. fh=open("Hi.txt","r")
3. print(fh.read())
4. fh.close()

So the first line sets “fh” to open the file, the second line reads the file to the console and the last 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("Hi.txt","w")
3. fh.write("THATS GOODS")
4. fh.write("OK BUDDY")
5. fh.close()

First line sets “fh” to open the file, the second and third line write text into the file and the last line closes it

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

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

G=0

for line in fh:

G+=1

print(line)

print(G)

fh.close()

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
2. fh=open("myfile.txt","r")
3. print(fh.read())
4. fh.close()
5. fh=open("myfile.txt",'a')
6. fh.write("Hi Hows it going")
7. fh=open("myfile.txt","r")
8. print(fh.read())

If you don’t close the file once you’re finished the file can be tampered with by others. You can tell the program worked because the first 3 lines of the program print the original text, then the last line print the text after it has been modified and you can see the changes to the 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

fh=open("myfile.txt","w")

fh.write("HEllo")

fh.write("HEllaso")

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

print(fh.read())

fh.close()

Appending is when you add text to the end of the file, write overwrites the text in the file with the provided text

**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.
4. fh=open("resources/myfile.txt","r")
5. print(fh.read())
6. fh.close()

Basically you add a forward slash to create a path to the file.

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 file is opened in binary mode. It opens it in read binary mode.

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

WB mode is when the file is opened in binary mode. I opens it in write binary mode

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
3. fh=open("resources/Penguin.bmp","rb")
4. print(fh.read())
5. fh.close()

When run, all you see is a bunch of numbers instead of a penguin.