**Level 1: Reading a Text File**

1. Open a new Python Repl and run the following program.

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

fileContents = fileHandle.read()

print(fileContents)

fileHandle.close()

1. Why does this program produce a run-time error?

This program produce a run-time error because there is no such file or directory; ‘myfile.txt’

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

1. Switch back to main.py pile and run the program.
   1. What gets printed out?

When I get back to the main.py pile it gets an error it says that there is no such file call myfile,txt

* 1. Explain the result.

1. Load and run the following program.

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

line = fileHandle.readline()

count = 1

while line :

print("Line ", count, " : ",line.strip())

line = fileHandle.readline()

count += 1

fileHandle.close()

1. Compare and contrast the output of the first and second program
   1. How is the read() function similar to the readline() function?

They are similar because they both process lines of text that are on “myfile.txt” and print them out.

* 1. How is the read() function different from the readline() function?  
     The read() function is different because it reads the entire file while the readline()

functions reads only one line in the file.

1. Research the Python open() function for file I/O (input / output).
   1. How do you specify which file to open?

You can specify to open which file by typing in myfile so python know that is myfile but if you type in your file the python wouldn’t give you the original file it would give you something else

* 1. Modify the program to open a different file.

**with** open('mydata.txt') **as** fp:

**for** line **in** iter(fp.readline, ''):

process\_line(line)

1. Research how to open a file in a sub-directory.
   1. Modify the second program to open a file in a sub-directory.
   2. Demo your program to Mr. Nestor
   3. List your program modifications below

I created new file which was computer then I created code which was

fileHandle = open("computer/myfile.txt","r")

fileContents = fileHandle.read()

print(fileContents)

fileHandle.close()

answer

Python 3.6.1 (default, Dec 2015, 13:05:11)

[GCC 4.8.2] on linux

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

line = fileHandle.readline()

count = 1

while line :

print("Line ", count, " : ",line.strip())

line = fileHandle.readline()

count += 1

fileHandle.close()

**Level 2: Writing a Text File**

1. Research the Python open() function for file I/O (input / output).
   1. What does the file mode “r” mean?

The r means open for reading (deaflut)

* 1. What mode is used to open a file for writing?

Mode that is used to open a file for writing is ‘W’

* 1. What other file modes can be used? List and explain their meanings.

‘x’ open for exclusive creation, failing if the file already exists

‘a’ open for writing, appending to the end of the file if it exists

‘b’ binary mode

‘t’ text mode (default)

‘+’ open a disk file for updating (reading and writing)

‘U’ [universal newlines](https://docs.python.org/3/glossary.html#term-universal-newlines) mode (deprecated)

1. Load and run the following program.

print("Enter test to write to a file")

print("Type STOP to end the program")

print(" ")

lineNumber = 0

while True :

lineNumber += 1

userPrompt = "Enter Line " + str(lineNumber) + " : "

userText = input(userPrompt)

if userText == "STOP" :

break

print(userText)

1. Modify the program to open a text file for writing.
   1. Demo your program to Mr. Nestor

List your program modifications below  
file=open("userText","w")

print("test")

print("STOP")

print(" ")

lineNumber = 0

while True :

lineNumber += 1

userPrompt = "Enter Line " + str(lineNumber) + " : "

userText = input(userPrompt)

if userText == "STOP" :

break

open("userText")



1. Replace the line “print(userText)” with a command to write the value of “userText” to an open file.
   1. Verify that text was written to your file
   2. Demo your program to Mr. Nestor
   3. List your program modifications below
2. fileHandle = open("myfile.txt","w")
3. print("Type STOP to end the program")
4. print(" ")
5. lineNumber = 0
6. while True :
7. lineNumber += 1
8. userPrompt = "Whats our favorite food? " + str(lineNumber) + " : "
9. userText = input(userPrompt)
10. if userText == "STOP" :
11. break
12. fileHandle.write(userText+'\n')
13. fileHandle.close()

**Level 3: Binary Files**

1. Add a folder to your repl workspace and call it “images”.
2. Locate and download a “BMP” format image file and add it to your images folder.
   1. The file must be a BMP file. JPG, GIF, PNG, etc. will not work
   2. Add the image by using “drag-and-drop” onto your images folder.
   3. You can use the “Penguin.bmp” file from the GitHub Topic B folder if you want
3. Load the following program
   1. Add it to your repl
   2. Modify the “open” command to read your image file
   3. Run the program and examine the data output.

"""

Function to convert 4 bytes (1 word) into a decimal integer

"""

def convertWordToInteger(dataWord) :

result = int(dataWord[3])

result += 256 \* int(dataWord[2])

result += 512 \* int(dataWord[1])

result += 1024 \*int(dataWord[0])

return result

"""

Function to display raw file data

Each data byte is displayed in row order

"""

def dumpRawData(rawData) :

idx = 0

for row in range(8) :

rowText = " ";

for col in range(8) :

rowText += str(rawData[idx]).zfill(3) + " "

idx += 1

print(rowText)

"""

Main program code begins here

- Start with opening and reading the data file

"""

handle = open("Penguin.bmp", "rb")

rawData = handle.read(64)

handle.close()

"""

Print out the RAW data contained at the start of the file

- This is the Header Information

- A BPM (Bitmap) Image has a well defined Header

- Each grouping of bytes has a specific meaning

"""

print(" ")

print("RAW Image Header Data (64 bytes)")

dumpRawData(rawData)

print(" ")

"""

According to the BMP specification the first two bytes

have the value "BM".

"""

print("First Two Bytes")

print(str(rawData[0]).zfill(3), str(rawData[1]).zfill(3))

print(" ")

"""

According to the BMP specification the image Width

is contained in the 4 bytes (1 word) biginning at

position 18

"""

print("Image Width Data")

dataText = str(rawData[18]).zfill(3) + " "

dataText += str(rawData[19]).zfill(3) + " "

dataText += str(rawData[20]).zfill(3) + " "

dataText += str(rawData[21]).zfill(3)

print("Image Width: (raw)", dataText)

dataText = str(rawData[21]).zfill(3) + " "

dataText += str(rawData[20]).zfill(3) + " "

dataText += str(rawData[19]).zfill(3) + " "

dataText += str(rawData[18]).zfill(3)

print("Image Width: (re-ordered)", dataText)

dataWord = [rawData[21],rawData[20],rawData[19],rawData[18]]

print("Image Width: (pixels)", convertWordToInteger(dataWord))

print(" ")

"""

According to the BMP specification the image Height

is contained in the 4 bytes (1 word) biginning at

position 22

"""

print("Image Height Data")

dataText = str(rawData[22]).zfill(3) + " "

dataText += str(rawData[23]).zfill(3) + " "

dataText += str(rawData[24]).zfill(3) + " "

dataText += str(rawData[25]).zfill(3)

print("Image Width: (raw)", dataText)

dataText = str(rawData[25]).zfill(3) + " "

dataText += str(rawData[24]).zfill(3) + " "

dataText += str(rawData[23]).zfill(3) + " "

dataText += str(rawData[22]).zfill(3)

print("Image Width: (re-ordered)", dataText)

dataWord = [rawData[25],rawData[24],rawData[23],rawData[22]]

print("Image Width: (pixels)", convertWordToInteger(dataWord))

print(" ")

"""

END OF PROGRAM

"""

1. Decode the meaning of the first two bytes of data in the header data of a BMP file.
   1. What are the values of the first two bytes?

The value is 066

* 1. Look up the values in an ASCII character table. Google “ASCII Character Table” or Download the ASCII Conversion Chart from the GitHub Topic B folder.
  2. What ASCII characters do these two bytes represent?

Two bytes represent 6 and 36 hex

1. Open and examine the BMP file format specification for the “Signature” data field
   1. Open the URL listed below to access the document
   2. According to the document, the first two bytes of data are the “Signature”
   3. What is the description of the “Signature” in a BMP file?

The description of the signature in the BMP file shows the first 2 bytes of the file

* 1. How does this compare to your answer to question #4 above?

This compare to the question #4 because it tells what are the first two bytes of data in the file and its usually called the signature of the file

<http://www.ece.ualberta.ca/~elliott/ee552/studentAppNotes/2003_w/misc/bmp_file_format/bmp_file_format.htm>

1. Examine the BMP file format specification for the Width data field
   1. Locate the “Width” data field in the BMP specification document.
   2. What is the size, in bytes, of this field?

4 bytes

* 1. What is the value, in bytes, of this field for your image file? (Look at the program output)

034 002 000 000

* 1. What is the value, in decimal, of this field for your image file? (Look at the program output)

 000 000 002 034

1. Examine the BMP file format specification for the Height data field
   1. Locate the “Height” data field in the BMP specification document.

  Locate the “Height” data field in the BMP specification document

* 1. What is the size, in bytes, of this field?

The size of this field is 4 bytes.

* 1. What is the value, in bytes, of this field for your image file? (Look at the program output)

The size is 045 001 000 000.

* 1. What is the value, in decimal, of this field for your image file? (Look at the program output)

The size is 000 000 001 045.

1. Open your BMP image file in an application program like Paint or Photoshop.
   1. What is the size of your image file?

Width is 287 and height is 301

* 1. How does this compare to the output of the program?

If you compare both output it is the same