File – Collection of data stored on secondary storage (disk)

Can’t store in memory?

* Memory is not permanent. When electricity is off, it is gone.

Two types of files:

1. Text Files – Human Readable

* ASCII, UNICODE

2. Binary Files – Not Readable

* Specific Encodings

Buffering

Reading from Disk is slow compared to Memory

*myFile = open(“myFile.txt”, “r”)*

*aFile = open(“temp.txt”, “r”)*

*for line\_str in aFile:*

*print(line\_str, end = ‘’) 🡪 end = ‘’ because in most files, there is \n*

fileObject.readline() 🡪 return the next line as a string.

.readlines() 🡪 return a list of all lines from the file

.read(N) 🡪 read N characters and return a single string

* If N is omitted, defaults to all remaining characters

*myFile = open(“myFile.txt”, “r”)*

*first\_str = aFile.readline()*

*>> ‘First Line\n’*

*second\_str = aFile.readline()*

*>> ‘Second Line\n’*

*myFile = open(“myFile.txt”, “r”)*

file\_contents = aFile.readlines()

file\_contents

>>[‘First Line\n’, ‘Second Line\n’]

*aFile = open(“temp.txt”, “w”)*

*print(“first line”, File = aFile)*

*print(“second line\_”, file=aFile, end = ‘’)*

*print(“third line”, file = aFile)*

*aFile.close()* 🡪 if don’t close, file will be empty. Everything you do is a buffer (memory)

first line

second line\_third line

-or-

aFile.write(“First Line\n”) 🡪 print does \n for you, write doesn’t

aFile.close()

line\_list = [“First\n”, “Second\n”]

aFile.writelines(line\_list)

fileObject.tell() 🡪 index of where you are in the file position

fileObject.seek(a) 🡪 specify index you want

fileObject.seek(a,b) 🡪 b = 0,1,2. Offset from beginning, current, end(negative)

* Note: text files only allow (a,0) or **(0,2) from the end**

seek cannot be used in between multi-unicode characters

Serialization (pickle)

Take python object and transfer into a file

pickle.dump(x,f) 🡪 Pickle object x into a file f

x = pickle.load(f)