**Reading and Writing Files**

Reading a file

The open function in Python opens a file. As an argument, it takes a filename as a string, and it returns a file object. A file object is a type of variable that represents a file on your hard drive.

We usually work with the file object through methods rather than directly using its name. The read() method reads the contents of the file. The command returns a string, which we can store.

To read a file, of course you need to know its location. open of course will open any file specified in the path. A file within your current working directory is easiest to open. The code below returns your working directory.

import os  
print(os.getcwd())

Using a text editor, please make a file. I made one called “temp3.”

file\_obj=open("temp3") # temp3 is the file name; file\_obj is a file object  
data=file\_obj.read() #data is string; the read method applies to the file objectprint(type(data))  
print(data.rstrip("\n")) #rstrip  
file\_obj.close()

rstrip returns a copy of the string with trailing characters removed.

data=file\_obj.read().rstrip() #both reads file and removes new line

Writing to a file

We use a two-argument version of the open() function to write to a file. The second argument says what we want to do with the file- the mode of the file. It can be ‘r’. ‘w’, or ‘a’. The default is reading, ‘r’.

file\_obj=open("temp3",'a')

file\_obj.write ("You bet!") #unlike print, there is no newline automatically added

file\_obj.write ("You bet! " + "For sure!\n")

file\_obj.close() #Use the close method when you are finished with a file.

As we have done before, we can also read a file line by line by treating the file object as a list.

file\_obj=open("temp3")

for line in file\_obj: #captures one line at a time.

print ("it is " + line)

file\_obj.close()