# File Objects - Reading and Writing to Files

txt\_file = open("plan.txt", "r")  
  
print(txt\_file.name) *# plan.txt*print(txt\_file.mode) *# r*txt\_file.close()

*# Using Context Manager*with open("test.txt", "r") as f:  
 f\_contents = f.read()  
 print(f\_contents)

with open('docs/para.txt', 'r') as file:

print(file)

# <\_io.TextIOWrapper name='docs/para.txt' mode='r' encoding='cp1252'>

*# Read all lines of a file*with open("test.txt", "r") as f:  
 for i in f:  
 print(i, end="")

*# The code above is identical to the one below:*with open("test.txt") as f:  
 con = f.read()  
 print(con)

*# The code above is identical to the one below:*with open("test.txt") as f:  
 con = f.read(5) *# prints only 5 chars (5 bytes) of the file*  
 print(con)

***# readline()***with open("test.txt", "r") as f:  
 print(f.readline())  
*# 1) Line 1*

***# readlines() -*** *reads until EOF using readline() and* ***returns a list*** *containing the lines*with open("test.txt", "r") as f:  
 all = f.readlines()  
 print(all)  
*# ['1) Line 1\n', '2) Line 2\n', '3) Line 3\n', '4) Line 4\n', '5)*

*# Line 5\n', '6) Line 6\n', '7) Line 7\n']*

*# Use the hint parameter to limit the number of lines returned. If the total number of bytes returned exceeds the specified number, no more lines are returned.*

with open('docs/pi.txt', 'r') as file:

content = file.readlines(4)

print(content)

*# e.g. read till end of the file*with open("test.txt", "r") as f:  
 size\_to\_r = 5  
 i = f.read(size\_to\_r)   
 while len(i) > 0:  
 print(i, end="")  
 i = f.read(size\_to\_r)

***# The tell() method returns the current file position in a file stream.***

*# You can change the current file position with the*[*seek()*](https://www.w3schools.com/python/ref_file_tell.asp)*method.*  
with open("test.txt", "r") as f:  
 size\_to\_r = 5  
 i = f.read(size\_to\_r)  
 print(f.tell()) *# we are in 5th position of our file!*

***# seek() - set the current file position***with open("test.txt", "r") as f:  
 size\_to\_r = 7  
 i = f.read(size\_to\_r)

print(i, end="")

f.seek(0)

*# now, second read() will start from beginning of the file!*  
 i = f.read(size\_to\_r)  
 print(i, end="")  
 *# we are in 10th position of our file!*with open("test2.txt", "w") as f: *# WRITE TO A FILE*  
 f.write("Hello world!")

with open("test3.txt", "w") as f:  
 f.write("Hello world!")  
 f.seek(0)  
 f.write("Hello Python!")

***# append()***

with open("test4.txt", "a") as f:  
 f.append("Hey there, what is goin on?")

*# Copy the content of one file to another*with open("test.txt", "r") as rf:  
 with open("test\_copy.txt", "w") as wf:  
 for line in rf: *# write all lines of it to test\_copy.txt* wf.write(line)

with open("docs/para.txt", "r") as f:

content\_para = f.read()

with open('docs/para\_copy.txt', 'w') as f\_w:

f\_w.write(content\_para)

***# working with binary files***

*# Make a copy of an image*

*# to mess around with files other than text files you have read and write that file in binary mode. (****rb: read binary, wb: write binary****)*with open("testpic.jpg", "rb") as rfb:  
 with open("testpic\_copy.jpg", "wb") as wfb:  
 for i in rfb:  
 wfb.write(i)

***# working with binary files***

*# Make a copy of an image, result is identical to above e.gs*with open("testpic.jpg", "rb") as rbf: *# e.g.*  
 with open("testpic2\_copy.jpg", "wb") as wbf:  
 chunk\_size = 2000  
 res = rbf.read(chunk\_size)  
 while len(res) > 0:  
 wbf.write(res)  
 res = rbf.read(chunk\_size)

# Automate Parsing and renaming multiple files

import os

for f in os.listdir('.'):

if f.endswith('.mp4'):

f\_name, f\_ext = os.path.splitext(f)

lesson, course, lesson\_number = f\_name.split('-')

lesson = lesson.strip()

course = course.strip()

lesson\_number = lesson\_number.strip()[1:].zfill(2)

ideal\_name = f'{lesson\_number}-{lesson}{f\_ext}'

os.rename(f, ideal\_name)