**# File handling allows a program to store data permanently so it can be used later — for example, saving user information, logs, configuration settings, etc.**

a = open("file\_handling/demo\_file.txt", "r")

print(a.read())  #to read the content

**# Using the with statement**

**# You can also use the with statement when opening a file:**

with open("file\_handling/demo\_file.txt") as f:

  print(f.read())

**# Close Files**

**# It is a good practice to always close the file when you are done with it.**

f = open("file\_handling/demo\_file.txt")

f.close()

**# Read Only Parts of the File**

**# By default the read() method returns the whole text, but you can also specify how many characters you want to return:**

with open("file\_handling/demo\_file.txt") as f:

  print(f.read(5))

**# Read Lines**

**# You can return one line by using the readline() method:**

with open("file\_handling/demo\_file.txt") as f:

  print(f.readline())

**# By calling readline() two times, you can read the two first lines:**

with open("file\_handling/demo\_file.txt") as f:

  print(f.readline())

  print(f.readline())

**# By looping through the lines of the file, you can read the whole file, line by line:**

with open("file\_handling/demo\_file.txt") as f:

  for x in f:

    print(x)

**# Write to an Existing File**

with open("file\_handling/demo\_file.txt", "a") as f:

    f.write("Now the file has more content!")

**# open and read the file after the appending:**

with open("file\_handling/demo\_file.txt") as f:

  print(f.read())

**# Note: the "w" method will overwrite the entire file.**

**# Overwrite Existing Content**

**# To overwrite the existing content to the file, use the w parameter:**

with open("file\_handling/demo\_file.txt", "w") as f:

  f.write("Woops! I have deleted the content!")

**# open and read the file after the overwriting:**

with open("file\_handling/demo\_file.txt") as f:

  print(f.read())

**Create a New File**

f = open("myfile.txt", "x")

**# Delete a File**

# To delete a file, you must import the OS module, and run its os.remove() function:

**import os**

os.remove("myfile.txt")

**# Check if File exist:**

# To avoid getting an error, you might want to check if the file exists before you try to delete it:

**import os**

if os.path.exists("file\_handling/demo\_file.txt"):

  os.remove("file\_handling/demo\_file.txt")

else:

  print("The file does not exist")