Table of Contents

Chapter 11

File Input/Output

RAM: Volatile
HDD: NonVolatile

File handling is an important part of any web application.
Python has several functions for creating, reading, updating, and deleting files.

File Handling

The key function for working with files in Python is the open() function.
The open() function takes two parameters; filename, and mode.
There are four different methods (modes) for opening a file:
"r" - Read - Default value. Opens a file for reading, error if the file does not exist
"a" - Append - Opens a file for appending, creates the file if it does not exist
"w" - Write - Opens a file for writing, creates the file if it does not exist
"x" - Create - Creates the specified file, returns an error if the file exists

In addition you can specify if the file should be handled as binary or text mode
"t" - Text - Default value. Text mode
"b" - Binary - Binary mode (e.g. images)

File Read

To open the file, use the built-in open() function.
The open() function returns a file object, which has a read() method for reading the content of the file.
Ex.

```
f = open("myText.txt",'r')
data = f.read() # reads whole file
print(data)
```

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.
Ex.

```
f = open("myText.txt",'r')
print(f.read(5))
```

Return the 5 first characters of the file.

Read Lines

You can return one line by using the readline() method.

Ex.

```
f = open("myText.txt",'r')
print(f.readline())
```

By calling readline() two times, you can read the two first lines.
By looping through the lines of the file, you can read the whole file, line by line:

Loop through the file line by line:

```
f = open("demofile.txt", "r")
for x in f:
  print(x)
```

Close Files

It is a good practice to always close the file when you are done with it.
Close the file when you are finish with it:

```
f = open("myFile.txt", "r")
print(f.readline())
f.close()
```

Write to an Existing File

To write to an existing file, you must add a parameter to the open() function:
"a" - Append - will append to the end of the file
"w" - Write - will overwrite any existing content
"x" - Create - will create a file, returns an error if the file exist

Ex.

```
f = open("myText.txt","w")
f.write("I will delete the previous word or in simple word I will overwrite")
f = open("myText.txt","a")
f.write("I will append your words to the current file")
f = open("newText.txt","x")
f.write("I will create a file if not present and I will raise an error if the file is already present")
f.close()
```

Delete a File

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

Ex.

```
import os
os.remove("newText.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.

Ex.

Check if file exists, then delete it:

```
import os
if os.path.exists("newFile.txt"):
  os.remove("newFile.txt")
else:
  print("The file does not exist")
```

Delete Folder

To delete an entire folder, use the os.rmdir() method.

Remove the folder "myfolder":

Ex.

```
import os
os.rmdir("myfolder")
```

with Statement

The best way to open and close file automatically is done by “with” statement. You don’t need to write close() as it is done automatically.

Ex.

```
with open("myFile.txt","r") as f:
    f.read()
```

Ex2.

```
# 1) without using with statement
 f = open('myFile.txt', 'w')
 f.write('Hello Madi!')
 f.close()
 # 2) without using with statement
 f = open('myFile.txt', 'w')
 try:
    f.write('Hello Madi')
 finally:
    f.close()
#3) using with statement
with open('myFile.txt', 'w') as f:
    f.write('Hello Madi!')
```

Ex. all:

```
# # Read a file:
# f = open("myText.txt","r")
# print(f.read()) # I will read whole file
# print(f.read(5)) # I will read only 5 characters
# print(f.readline()) # I will read only the first line and if you use again I will go for the second line
# f.close() # I will close the file
# # Write a file:
# f = open("myText.txt","w")
# f.write("I will delete the previous word or in simple word I will overwrite")
# f = open("myText.txt","a")
# f.write("I will append your words to the current file")
# f = open("newText.txt","x")
# f.write("I will create a file if not present and I will raise an error if the file is already present")
# # Delete a file:
# import os
# os.remove("newText.txt")
# # Check and delete:
# import os
# if os.path.exists("newText.txt"):
#   os.remove("newText.txt")
# else:
#   print("The file does not exist")
# # Delete Folder:
# import os
# os.rmdir("forRemove")
# # Note: You can only remove empty folders.
```