



# Chapter 8:Files

Prepared by: Hanan Hardan

# File handling

- 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, Whatever you write to file will get appended and original content will not overwritten, creates the file if it does not exist
  - "w" - Write - Opens a file for writing, creates the file if it does not exist
  - "r+" - Opens a file for both reading and writing
  - "w+" - Opens a file for reading and writing. If the file does not exist then it will create new one. If it exist then it will overwrite it

# File handling

- **Syntax:**
- To open a file for reading it is enough to specify the name of the file:  
`f = open("demofile.txt")`
- The code above is the same as:  
`f = open("demofile.txt", "r")`
- **Note:** Make sure the file exists, or else you will get an error.

# Open a File:

Assume we have the following file, located in the same folder as Python:

```
demofile.txt
```

```
Hello! Welcome to demofile.txt  
This file is for testing purposes.  
Good Luck!
```

- 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:

# Open a File:

## Example

```
f = open("demofile.txt", "r")  
print(f.read())  
f.close()
```

- Assume we have the file, located in the different place you must determine the path:

```
f = open("c:\\data\\demofile.txt", "r")  
print(f.read())  
f.close()
```

# Read Only Parts of the File:

- By default the `read()` method returns the whole text, but you can also specify how many character you want to return:

Example: Return the 5 first characters of the file:

```
f = open("demofile.txt", "r")  
print(f.read(5))  
f.close()
```

# Read Lines:

- You can return one line by using the `readline()` method:

Example: Read one line of the file:

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

- By calling `readline()` two times, you can read the two first lines:

Example: Read two lines of the file:

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



# Read Lines:

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

Example: Loop through the file line by line:

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

Example: Open the file "demofile.txt" and append content to the file:

```
f = open("demofile.txt", "a")  
f.write("Now the file has one more line!")  
f.close()
```

# Write to an Existing File:

- **Example:** Open the file "demofile.txt" and overwrite the content:

```
f = open("demofile.txt", "w")  
f.write("Woops! I have deleted the content!")  
f.close()
```

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

# Create a New File:

- To create a new file in Python, use the `open()` method, with one of the following parameters:
- "a" - Append - will create a file if the specified file does not exist
- "w" - Write - will create a file if the specified file does not exist

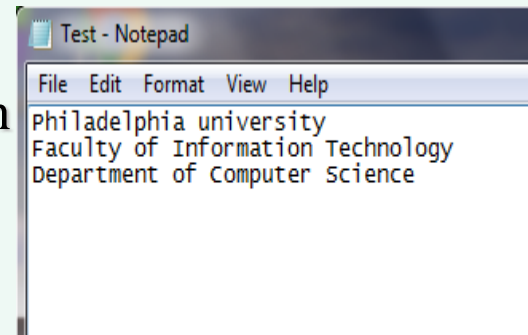
Example: Create a file called "myfile.txt":

```
f = open("myfile.txt", "w")
```

- Result: a new empty file is created!

# Example 1

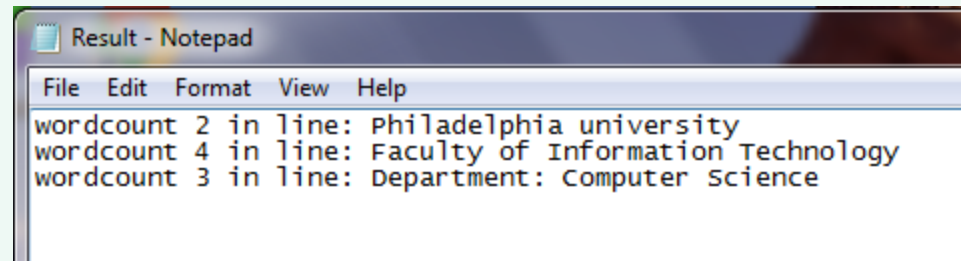
Example : If you have the following file  
We want to create a new file that has all the line in  
the above file plus word count for each line.



```
f=open("Test.txt","r")
f_out=open("Result.txt","a")
for line in f:
    tokens=line.split(" ")
    f_out.write("wordcount "+str(len(tokens))+ " in line: "+line)
f.close()
f_out.close()
```

# Example 1

- Output:



```
Result - Notepad
File Edit Format View Help
wordcount 2 in line: Philadelphia university
wordcount 4 in line: Faculty of Information Technology
wordcount 3 in line: Department: Computer Science
```

# Delete a File:

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

Example: Remove the file "demofile.txt":

```
import os
```

```
os.remove("Result.txt")
```

# Check if File exist:

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

Example: Check if file exist, *then* delete it:

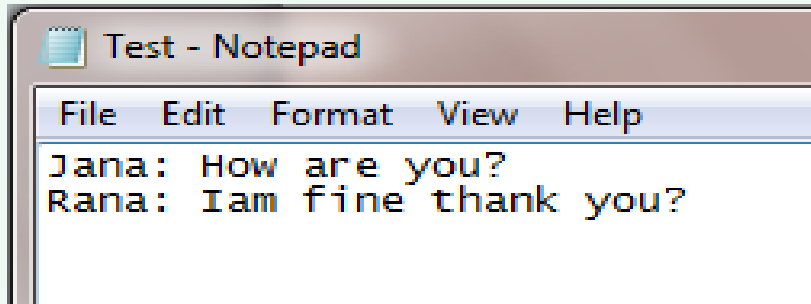
```
import os
if os.path.exists("Result.txt "):
    os.remove("Result.txt ")
else:
    print("The file does not exist")
```



# Example 2

Write python code that:

- Read the file name from user
- Find the number of characters, words and lines in the file
- Print the result with appropriate message



```
Test - Notepad
File Edit Format View Help
Jana: How are you?
Rana: Iam fine thank you?
```

# Example 2

- Solution (1):

```
filename =input('What is the filename? ')
```

```
source = open(filename)
```

```
text = source.read()
```

```
numchars = len(text)
```

```
numwords = len(text.split())
```

```
numlines = len(text.split('\n'))
```

```
print(numlines, numwords,numchars)
```

```
source.close()
```

# Example 2

- Solution (2):by using for statement

```
filename =input('What is the filename? ')
source = open(filename)
numlines = numwords = numchars = 0
for line in source:
    numchars += len(line)
    numwords += len(line.split())
    numlines += 1

print (numlines, numwords, numchars)
source.close()
```