



HANDLING FILES



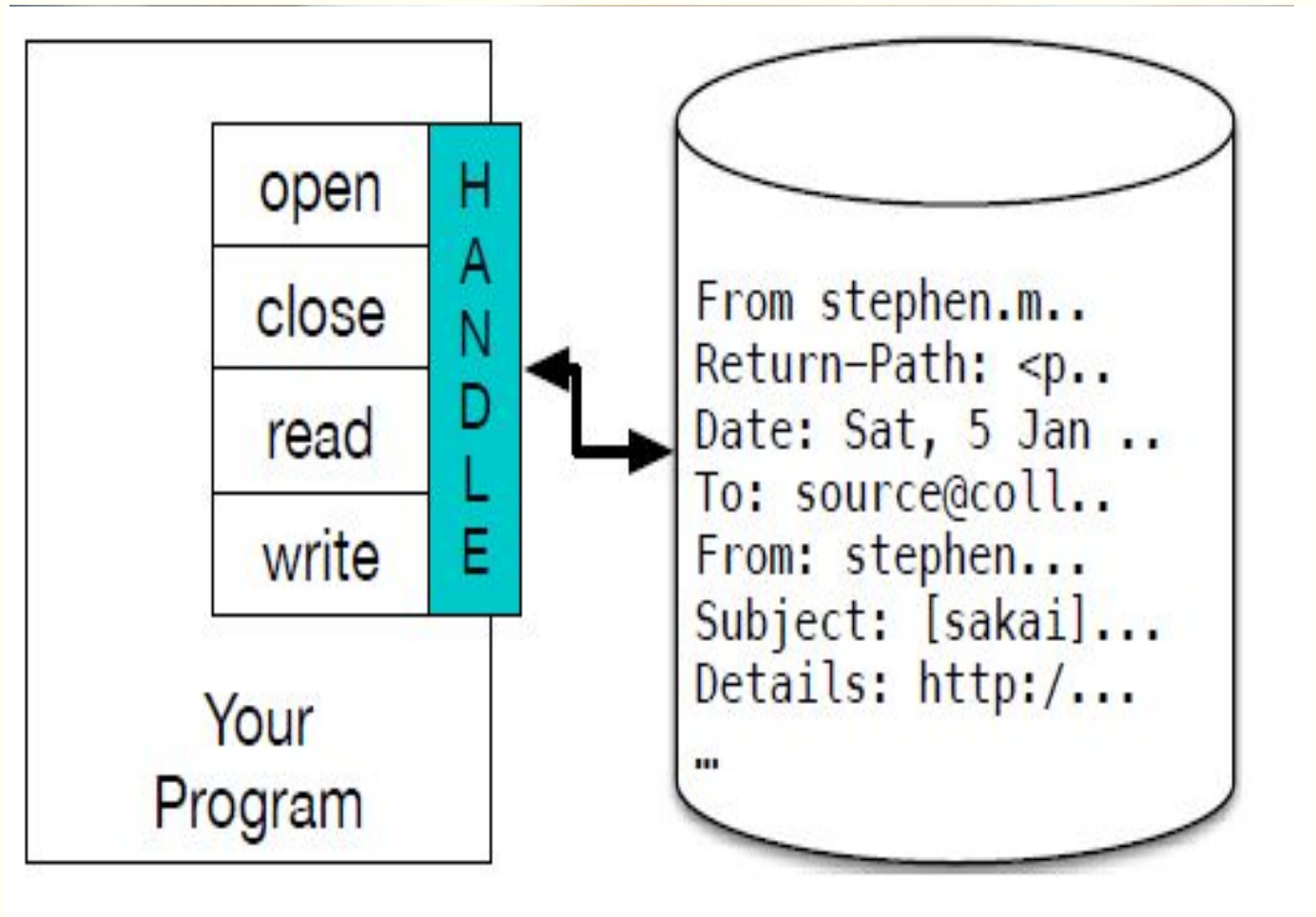
Introduction to Files Concept

- So far, we have learned how to write programs and communicate our intentions to the *Central Processing Unit using conditional execution, functions, and iterations*.
- We have learned how to create and use data structures in the *Main Memory*. *The CPU and memory are where our software works and runs. It is where all of the “thinking” happens.*
- In this chapter, we start to work with *Secondary Memory (or files)*. *Secondary memory is not erased when the power is turned off.*
- In this session, you'll learn about Python file operations. More specifically, opening a file, reading from it, writing into it, closing it and various file methods you should be aware of.

File Operations

The main file operations are:

- Opening a File
- Reading a File
- Writing a File/Appending a File
- Closing a File



Opening A File

- When we want to read or write a file (say on your hard drive), we first must *open the file*.
- Opening the file communicates with your operating system, which knows where the data for each file is stored.
- When you open a file, you are asking the operating system to find the file by name and make sure the file exists.
- Syntax to open a File:
 - `>>> fhand = open('mbox.txt') → open("filename.ext", "mode(r,w,a)")`
 - `>>> print(fhand)`
 - `<_io.TextIOWrapper name='mbox.txt' mode='r' encoding='cp1252'>`

Writing a File

- To write a file, you have to open it with mode “w” as a second parameter:
- Syntax for Writing a File:
- `>>> fout = open('mbox.txt', 'w')`
- `>>> print(fout)`
- `<_io.TextIOWrapper name='output.txt' mode='w' encoding='cp1252'>`
- `>>> line1 = "This here's the wattle,\n"`
- `>>> fout.write(line1)`
- 24
- `>>> fout.close()`

Reading a File

- To write a file, you have to open it with mode “r” as a second parameter:
- Syntax for Reading a File:
- `>>> fread=open("mbox.txt","r")→ open("filename.ext","r")`
- `>>> print(fread)`
- `<_io.TextIOWrapper name='mbox.txt' mode='r' encoding='cp1252'>`
- `>>> inp=fread.read()`
- `>>> print(inp)`

Appending Data into a File

- To append data into a file which already exists, “a” (append) mode of operation is used.
- Syntax To Append Data into a file:
- `>>>append=open(“mbox.txt”, ”a”)`
- `>>>append.write(“Appending Data into File “)`
- `>>>append.close()`
- After appending data into the file, repeat the file reading process to the availability of the appended data.

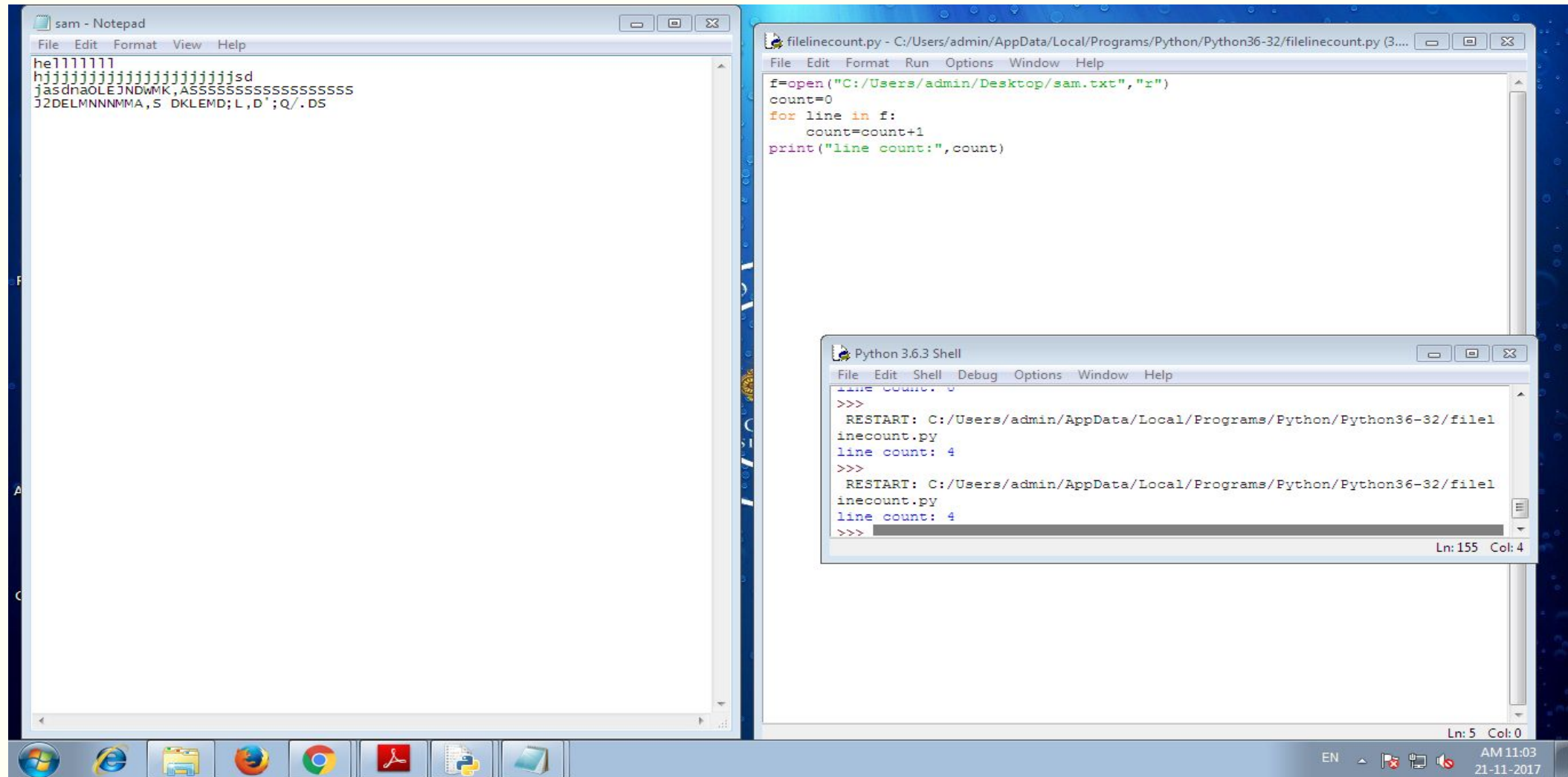
Letting the User Choose the File Name

- `fname = input('Enter the file name: ')`
- `try:`
- `fhand = open(fname)`
- `except:`
- `print('File cannot be opened:', fname)`
- `exit()`
- `count = 0`
- `for line in fhand:`
- `if line.startswith('Subject:')`
- `count = count + 1`
- `print('There were', count, 'subject lines in', fname)`

Using Try, Except, and Open

- `fname = input('Enter the file name: ')`
- **try:**
- `fhand = open(fname)`
- **except:**
- `print('File cannot be opened:', fname)`
- `exit()`
- `count = 0`
- **for line in fhand:**
- **if line.startswith('Subject:')**
- `count = count + 1`
- `print('There were', count, 'subject lines in', fname)`

Finding number of lines in a file



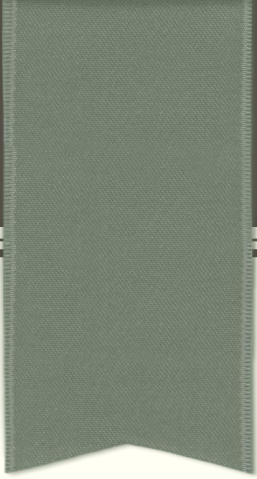
Summary

Python File Modes

Mode	Description
'r'	Open a file for reading. (default)
'w'	Open a file for writing. Creates a new file if it does not exist or truncates the file if it exists.
'x'	Open a file for exclusive creation. If the file already exists, the operation fails.
'a'	Open for appending at the end of the file without truncating it. Creates a new file if it does not exist.
't'	Open in text mode. (default)
'b'	Open in binary mode.
'+'	Open a file for updating (reading and writing)

Exercise

- What is a file?
- What are the file operations ?



THANK YOU ALL!!
**QUESTIONS
PLEASE?**

