PYTHON: FILE PROCESSING

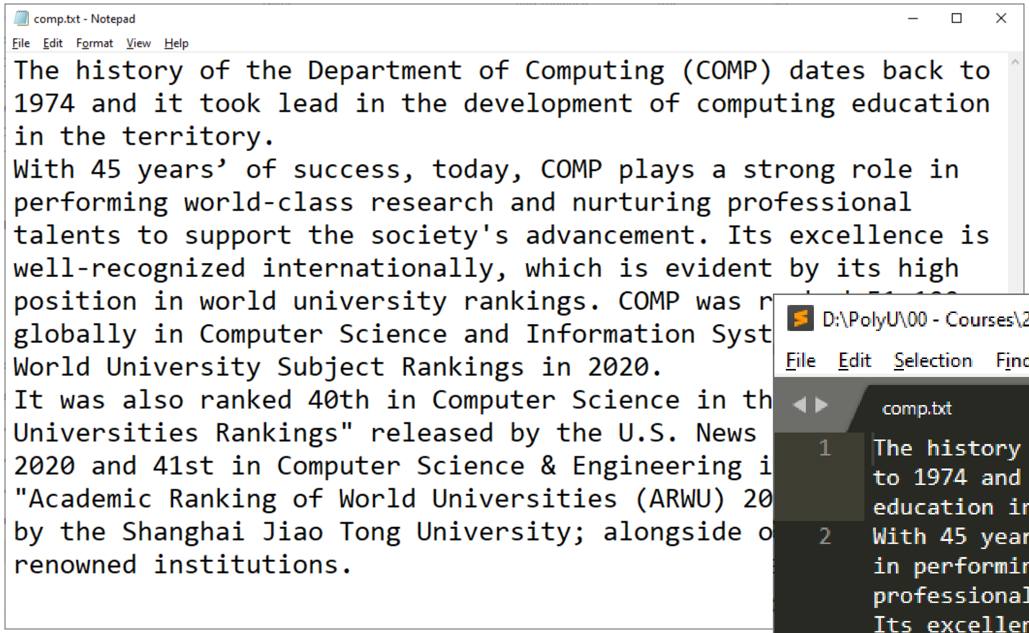
Objectives

- Files
  - To understand basic file processing concepts and techniques for reading and writing text files in Python.
  - To be able to understand and write programs that process textual information.

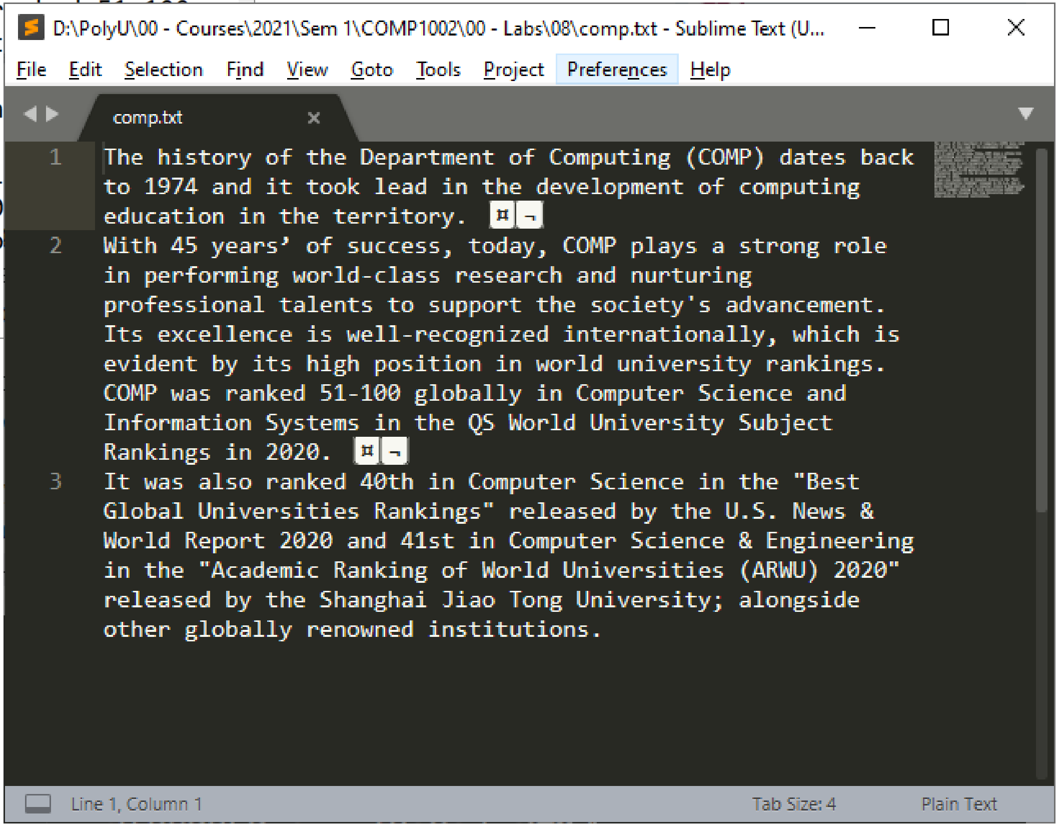
File (object)

- A *file* is a sequence of data that is stored in secondary memory (disk drive).
- Two types of files: *text file* and *binary file*
  - A text file contains characters, structured as lines of text.
  - A binary file is a file formatted in a way that only a computer program can read.
- A text file usually contains more than one line of text. Lines of text are separated with a special character, the *newline* character.

newline character in a text file



```
The history of the Department of Computing (COMP) dates back to
1974 and it took lead in the development of computing education
in the territory.
With 45 years’ of success, today, COMP plays a strong role in
performing world-class research and nurturing professional
talents to support the society's advancement. Its excellence is
well-recognized internationally, which is evident by its high
position in world university rankings. COMP was r
globally in Computer Science and Information Syst
World University Subject Rankings in 2020.
It was also ranked 40th in Computer Science in th
Universities Rankings" released by the U.S. News
2020 and 41st in Computer Science & Engineering i
"Academic Ranking of World Universities (ARWU) 20
by the Shanghai Jiao Tong University; alongside o
renowned institutions.
```


D:\PolyU\00 - Courses\2021\Sem 1\COMP1002\00 - Labs\08\comp.txt - Sublime Text (U...
File Edit Selection Find View Goto Tools Project Preferences Help
comp.txt
Line 1, Column 1
Tab Size: 4
Plain Text

```
1  The history of the Department of Computing (COMP) dates back
   to 1974 and it took lead in the development of computing
   education in the territory. ¤¬
2  With 45 years’ of success, today, COMP plays a strong role
   in performing world-class research and nurturing
   professional talents to support the society's advancement.
   Its excellence is well-recognized internationally, which is
   evident by its high position in world university rankings.
   COMP was ranked 51-100 globally in Computer Science and
   Information Systems in the QS World University Subject
   Rankings in 2020. ¤¬
3  It was also ranked 40th in Computer Science in the "Best
   Global Universities Rankings" released by the U.S. News &
   World Report 2020 and 41st in Computer Science & Engineering
   in the "Academic Ranking of World Universities (ARWU) 2020"
   released by the Shanghai Jiao Tong University; alongside
   other globally renowned institutions.
```

How to end a line? (http://en.wikipedia.org/wiki/Newline)

- LF (chr(10)): Unix and Unix-like systems (Linux, OS X – Mac OS).
- CR (chr(13) or <^M>): Apple II family, Mac OS up to version 9
- CR+LF (chr(13)+chr(10)): Microsoft Windows
- These normally would not cause you too much problems.
- Python as installed on your machines understand the file convention.
- However, when editing programs in Windows and compiling in Unix/Linux in future (or vice versa), watch out for the difference in end-of-line representation.
  - It is safer if you use ASCII mode to transfer the program files via sftp / winscp. Avoid the binary mode for these files.
  - Unix script files containing extra <^M> will not run correctly (Unix uses just LF but Windows uses both LF and CR, with an excessive CR).

A typewriter

Watch:
https://www.youtube.com/watch?v=FkUXn5bOwzk

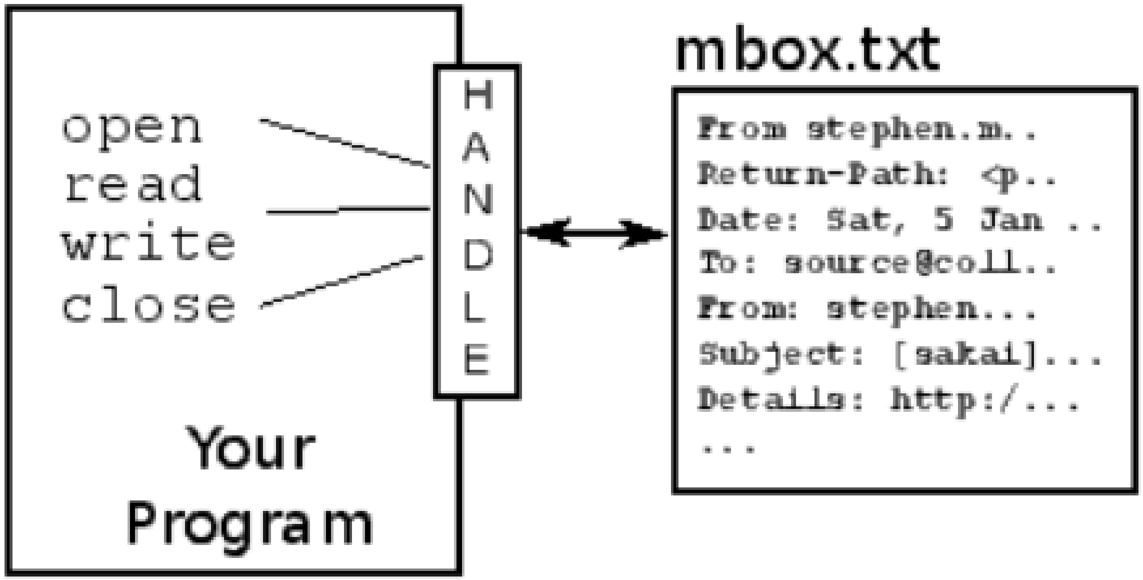
Image Sources: https://photodune.net/item/woman-typing-vintage-typewriter-top-view/22937908

File Processing

- Open a file in a secondary storage.
- Read / write the file.
- Save the file if write.
- Close the file.

Opening a file in a secondary storage

- If successful, a “file handler” will be returned.
  - `infile = open("myfile.txt", "r")`
  - `infile = open("myfile.txt", "w")`



```
From stephen.m..
Return-Path: <p..
Date: Sat, 5 Jan ..
To: source@coll..
From: stephen...
Subject: [sakai]...
Details: http:/...
...
```

Source: http://www.pythonlearn.com/html-008/cfbook008.html

Reading/writing and saving a file

- Load (part of) the file into the main memory.
- Read the file from the memory.
- Write the file to the memory.
- Saving will write the file in the memory to a secondary storage.

EXERCISE 9.1

Try to create a small test file (`test.txt`) in the directory where your program starts execution and type your program as follows:

```
infile = open("test.txt", "r")
data = infile.read()
print(len(data))
print(data)
```

Look at the ASCII code of the first 30 characters. Identify <LF> (10) or <CR> (13) if possible.

```
for i in range(30):
    print(data[i], ord(data[i]), end=" ")
```

Do you observe something special?

EXERCISE 9.2

Try

```
infile = open("test.txt", "r")
for i in range(5):
      line = infile.readline()
      print(len(line))
      print(line[:-1])
```

- What happens if the file contains fewer than 5 lines?
- Could you print the individual character and their ASCII code?

EXERCISE 9.3

Try

```
infile = open("test.txt", "r")
for line in infile.readlines():
      print(len(line))
      print(line)
```

- Does the output look the same as the original file?
- What do you observe when compared with Ex 9.2?

Three file read methods

- Note that if you are to run file methods inside a Python program, the file should reside in the same directory as the program. See the first line when you run the program:
- ================= RESTART: C:\Users\Desktop\Ex9.py =================

- `<filevar>.read()` – returns the entire remaining contents of the file as a single (possibly large, multi-line) string.
  - You will handle individual characters/strings by yourself.
- `<filevar>.readline()` – returns the next line of the file. This is all text up to *and including* the next newline character.
  - You will process just one line and repeat this for next line.
- `<filevar>.readlines()` – returns a list of the remaining lines in the file. Each list item is a single line *including* the newline characters.
  - You get all the lines, and could choose to process each line in turn.
  - This is most flexible, but consumes more memory than the second method.

EXERCISE 9.4

Try

```
input_file = open("some.txt", "r")
output_file = open("clone.txt", "w")
content = input_file.read()
output_file.write(content)
output_file.close()
```

Find the new `clone.txt` file and look inside the content.

This is how you copy a file.

Watch out that a previous `clone.txt` file, if exist, will be overwritten.

EXERCISE 9.5

Try

```
input_file = open("some.txt", "r")
output_file = open("clone2.txt", "a")
content = input_file.read()
output_file.write(content)
output_file.close()
```

Open the new `clone2.txt` file.

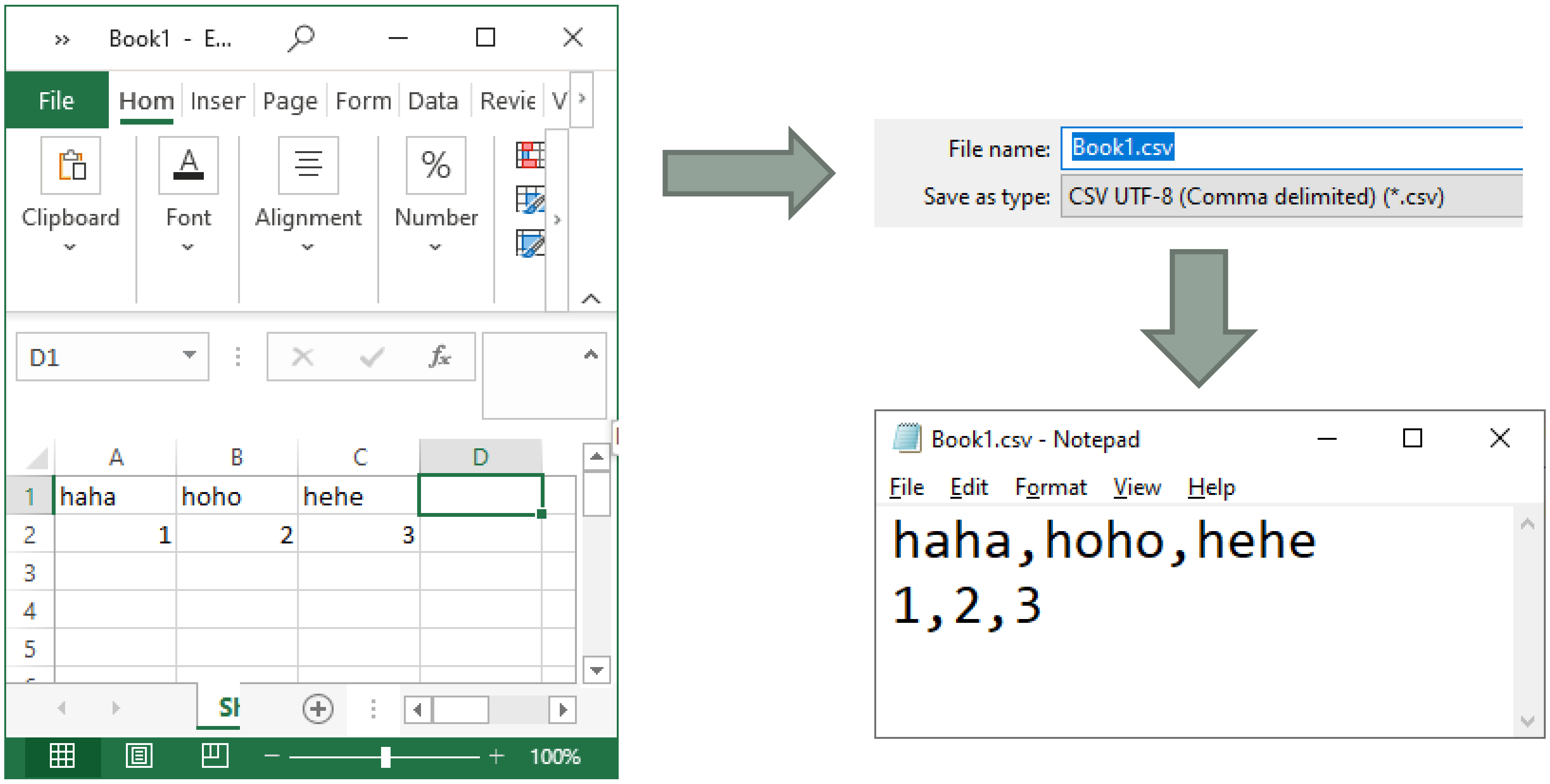
See if there is any difference whether `clone2.txt` exists before program execution or not.

Write and append methods

- Opening a file for writing prepares the file to receive data.
- If you open an existing file for writing, you wipe out the file's original contents. If the named file does not exist, a new one is created.
- It is important to close a file that is written to, otherwise the tail end of the file may not be written to the file.
    - This phenomenon is due to a special arrangement called buffering (in order to improve efficiency).

Comma-separated values (csv) files

- A comma-separated values (CSV) file is a delimited *text file* that uses a comma to separate values.



	A	B	C	D
1	haha	hoho	hehe	
2	1	2	3	
3				
4				
5				

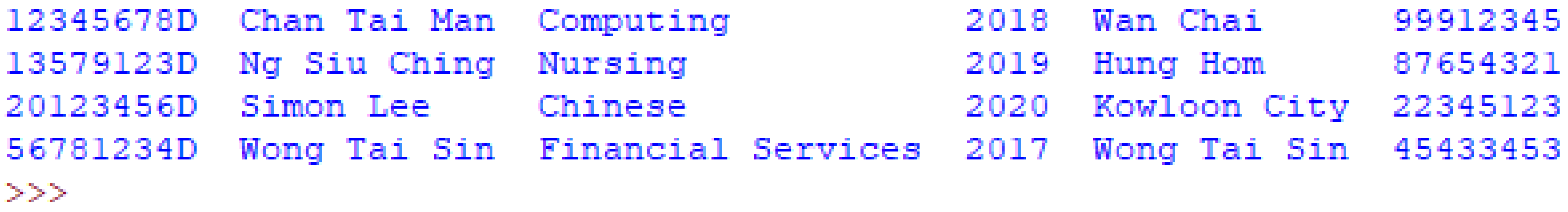
File name: Book1.csv
Save as type: CSV UTF-8 (Comma delimited) (*.csv)

Book1.csv - Notepad

```
haha,hoho,hehe
1,2,3
```

EXERCISE 9.6

Download *student.csv* from Blackboard which contains the data in Example 2 of Lab 8. Write a Python program that reads the file and displays the data in terms of a table as follows:



```
12345678D  Chan Tai Man   Computing           2018  Wan Chai      99912345
13579123D  Ng Siu Ching   Nursing             2019  Hung Hom      87654321
20123456D  Simon Lee      Chinese             2020  Kowloon City  22345123
56781234D  Wong Tai Sin   Financial Services  2017  Wong Tai Sin  45433453
>>>
```

Useful Tools

- Sublime Text 3 with RawLineEdit
  - To view the ending characters of a line
  - https://www.sublimetext.com/
  - https://facelessuser.github.io/RawLineEdit/
- HxD
  - To view and edit file in byte (hex) level
  - https://mh-nexus.de/en/hxd/

END