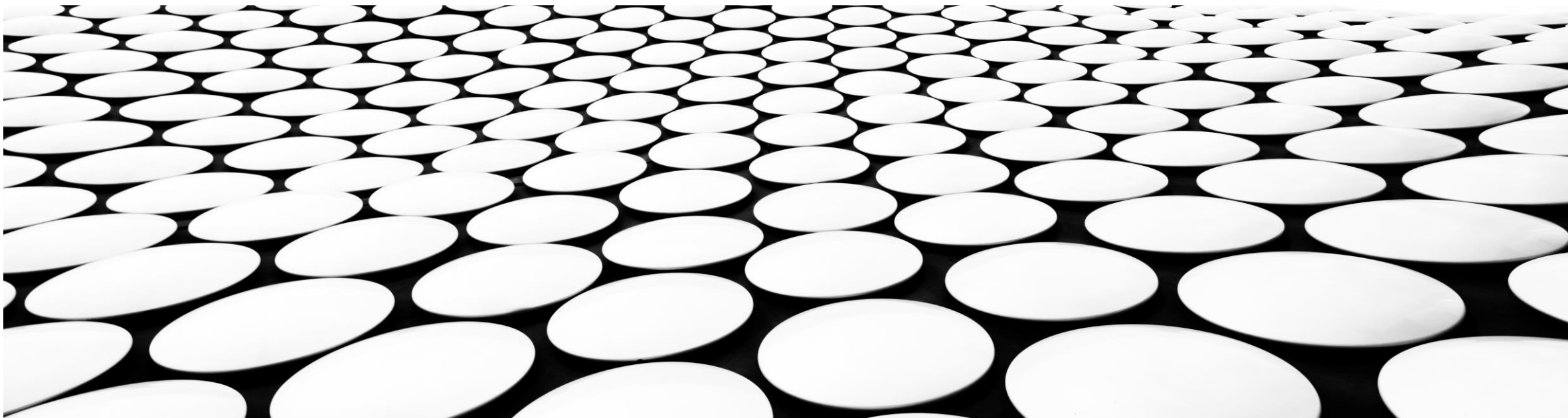

KĨ THUẬT LẬP TRÌNH PYTHON

NGUYỄN MẠNH HÙNG



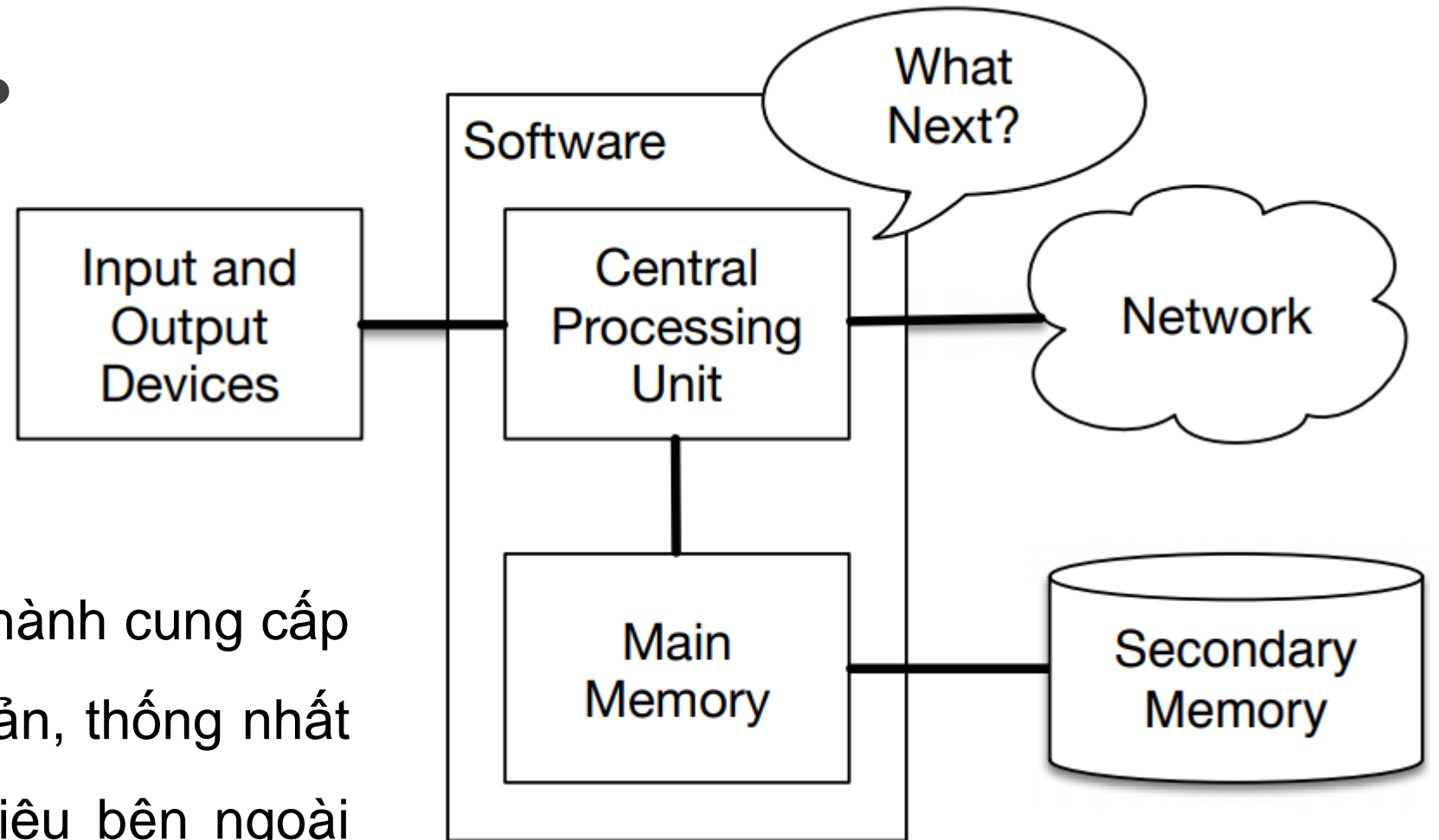


CHƯƠNG 3.3: NHẬP VÀ XUẤT FILE



BỘ NHỚ THỨ CẤP

- Hầu hết các hệ điều hành cung cấp một giao diện đơn giản, thống nhất để thao tác với dữ liệu bên ngoài thông qua các đối tượng file (tệp).



KIỂU DỮ LIỆU FILE (TỆP)

- Python định nghĩa **file** như là một *chuỗi các ký tự văn bản* (ASCII hoặc Unicode) hay các *byte*.
- Chuỗi ký tự có thể được xử lý như một chuỗi các dòng có độ dài thay đổi; mỗi dòng được kết thúc bằng một ký tự xuống dòng.
- Các tệp văn bản thông thường có thể được quản lý trực tiếp bằng các đối tượng file và các phương thức tích hợp sẵn để đọc và ghi các dòng dữ liệu.
- Các tệp là một chuỗi các byte có thể bao gồm ký tự hoặc các đối tượng dữ liệu khác được mã hóa dưới dạng byte, và không có ranh giới dòng rõ ràng. Một số tệp dạng byte có thể được xử lý bởi các thư viện như *pickle*.

THAO TÁC VỚI FILE

| Operation | Interpretation |
|--|---|
| <code>output = open(r'C:\spam', 'w')</code> | Create output file ('w' means write) |
| <code>input = open('data', 'r')</code> | Create input file ('r' means read) |
| <code>input = open('data')</code> | Same as prior line ('r' is the default) |
| <code>aString = input.read()</code> | Read entire file into a single string |
| <code>aString = input.read(N)</code> | Read up to next N characters (or bytes) into a string |
| <code>aString = input.readline()</code> | Read next line (including \n newline) into a string |
| <code>aList = input.readlines()</code> | Read entire file into list of line strings (with \n) |
| <code>output.write(aString)</code> | Write a string of characters (or bytes) into file |
| <code>output.writelines(aList)</code> | Write all line strings in a list into file |
| <code>output.close()</code> | Manual close (done for you when file is collected) |
| <code>output.flush()</code> | Flush output buffer to disk without closing |
| <code>anyFile.seek(N)</code> | Change file position to offset N for next operation |
| <code>for line in open('data'): use line</code> | File iterators read line by line |
| <code>open('f.txt', encoding='latin-1')</code> | Python 3.X Unicode text files (str strings) |
| <code>open('f.bin', 'rb')</code> | Python 3.X bytes files (bytes strings) |
| <code>codecs.open('f.txt', encoding='utf8')</code> | Python 2.X Unicode text files (unicode strings) |
| <code>open('f.bin', 'rb')</code> | Python 2.X bytes files (str strings) |

1. MỞ FILE

`afile = open(filename, mode)`

- *filename*: tên file có đường dẫn đầy đủ; nếu không, file được cho là nằm trong thư mục làm việc.
- *mode*: chế độ xử lý ('r' – mặc định, 'w', 'a')

```
afile=open('myfile.txt')  
afile
```

```
<_io.TextIOWrapper name='myfile.txt' mode='r' encoding='cp1258'>
```

- Trình biên dịch báo lỗi khi không tìm thấy file:

```
afile=open('myfile')
```

Đường dẫn,
tên file phải
đầy đủ

FileNotFoundError

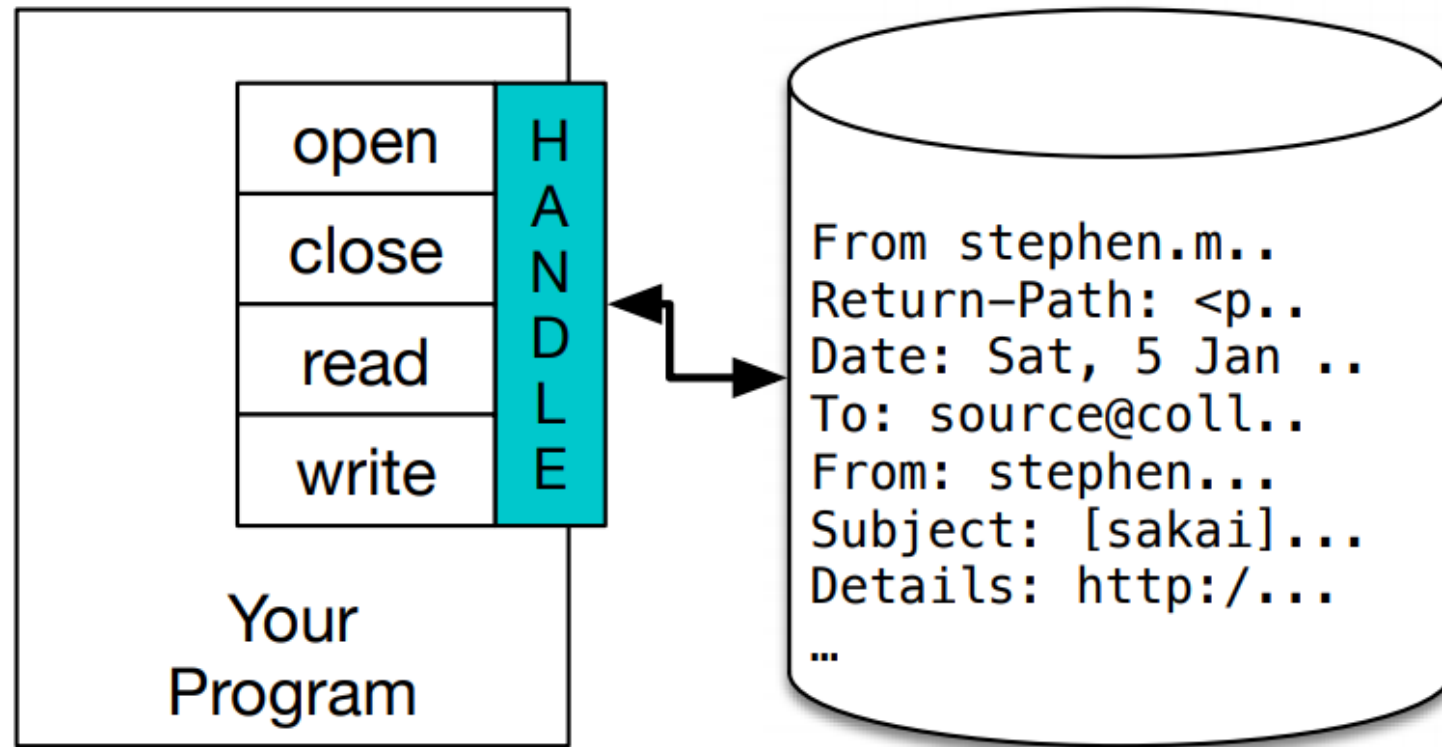
all last)

<ipython-input-3-89dc015586bf> in <module>

----> 1 afile=open('myfile')

FileNotFoundError: [Errno 2] No such file or directory: 'myfile'

Traceback (most recent c



MỘT SỐ LƯU Ý KHI SỬ DỤNG FILE

- Trong tất cả các trường hợp, file text có dạng chuỗi. Do đó nội dung khi đọc từ file hay ghi vào file đều là chuỗi.
- Một file text bao gồm các dòng. Mỗi dòng được kết thúc bởi kí tự ‘\n’.
- Python không tự động thêm vào bất cứ định dạng nào và cũng không tự động chuyển đổi các đối tượng sang dạng chuỗi khi ta ghi vào file.
- Các file đầu ra luôn được lưu vào bộ nhớ tạm, có nghĩa là văn bản được ghi có thể không được chuyển từ bộ nhớ sang đĩa ngay lập tức cho đến khi file được đóng.

2. ĐỌC FILE

- Phương thức **read()**: đọc và hiển thị toàn bộ nội dung file như một chuỗi, bao gồm cả kí tự kết thúc dòng '\n'.

```
f=open('myfile.txt')  
f.read()
```

```
'From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008\nReturn-Path: <postmaster@collab.sakaiproject.org>\nDate: Sat, 5 Jan 2008 09:12:18 -0500\nTo: source@collab.sakaiproject.org\nFrom: stephen.marquard@uct.ac.za\nSubject: [sakai] svn commit: r39772 - content/ranches/\nDetails: http://source.sakaiproject.org/viewsvn/?view=rev&rev=39772'
```

- Hiển thị file theo dòng:

```
f=open('myfile.txt')  
print(f.read())
```

```
From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008  
Return-Path: <postmaster@collab.sakaiproject.org>  
Date: Sat, 5 Jan 2008 09:12:18 -0500  
To: source@collab.sakaiproject.org  
From: stephen.marquard@uct.ac.za  
Subject: [sakai] svn commit: r39772 - content/branches/  
Details: http://source.sakaiproject.org/viewsvn/?view=rev&rev=39772
```

- Đọc từng dòng của file sử dụng vòng lặp **for**:

```
f=open('myfile.txt')  
for line in f:  
    print(line)
```

From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008

Return-Path: <postmaster@collab.sakaiproject.org>


Date: Sat, 5 Jan 2008 09:12:18 -0500

To: source@collab.sakaiproject.org

From: stephen.marquard@uct.ac.za

Subject: [sakai] svn commit: r39772 - content/branches/

Details: <http://source.sakaiproject.org/viewsvn/?view=rev&rev=39772>



Tại sao dẫn
cách dòng lớn?

- Phương thức **readline()**: trả về chuỗi là nội dung của dòng tiếp theo trong file, bao gồm cả kí tự kết thúc dòng.

```
f=open('myfile.txt')  
f.readline()  
f.readline()
```

```
'Return-Path: <postmaster@collab.sakaiproject.org>\n'
```

- Phương thức **startswith(<st>)**: chỉ đọc những dòng bắt đầu với <st>.

```
f=open('myfile.txt')
for line in f:
    line=line.rstrip()
    if line.startswith('From'):
        print(line)
```

```
From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008
From: stephen.marquard@uct.ac.za
```

CÂU LỆNH **WITH**

- Câu lệnh **with** được dùng để chắc chắn rằng ta đã đóng file khi kết thúc công việc với nó.
- Cú pháp:

with *expression* **as** *variable* :
suite

```
with open('myfile.txt') as source:
    for line in source:
        line=line.rstrip()
        print(line)
```

From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008

Return-Path: <postmaster@collab.sakaiproject.org>

Date: Sat, 5 Jan 2008 09:12:18 -0500

To: source@collab.sakaiproject.org

From: stephen.marquard@uct.ac.za

Subject: [sakai] svn commit: r39772 - content/branches/

Details: <http://source.sakaiproject.org/viewsvn/?view=rev&rev=39772>

3. GHI FILE

- Để ghi file, ta phải mở file ở chế độ ghi:

`afile = open(filename, 'w')`

+ Nếu *filename* đã tồn tại, việc mở file để ghi sẽ xóa tất cả dữ liệu cũ và ghi mới.

+ Nếu *filename* chưa có, một file mới sẽ được tạo.

- Phương thức **write()**: ghi dữ liệu (dưới dạng chuỗi) vào file, và trả về số kí tự được ghi.
- Đối tượng file sẽ theo dõi vị trí của chuỗi được ghi, nếu muốn ghi lần nữa thì dữ liệu mới sẽ được nối vào điểm cuối.

```
fw=open('output.txt','w') # Mở file để ghi
line1="Binh yen\n"        # Kết thúc dòng bằng '\n'
line2="nam trong tam tri"
fw.write(line1)
fw.write(line2)
fw.close()                 # Đóng file để chuyển dữ liệu
                             # từ bộ nhớ tạm vào file
fr=open('output.txt')      # Mở file để đọc
print(fr.read())
```

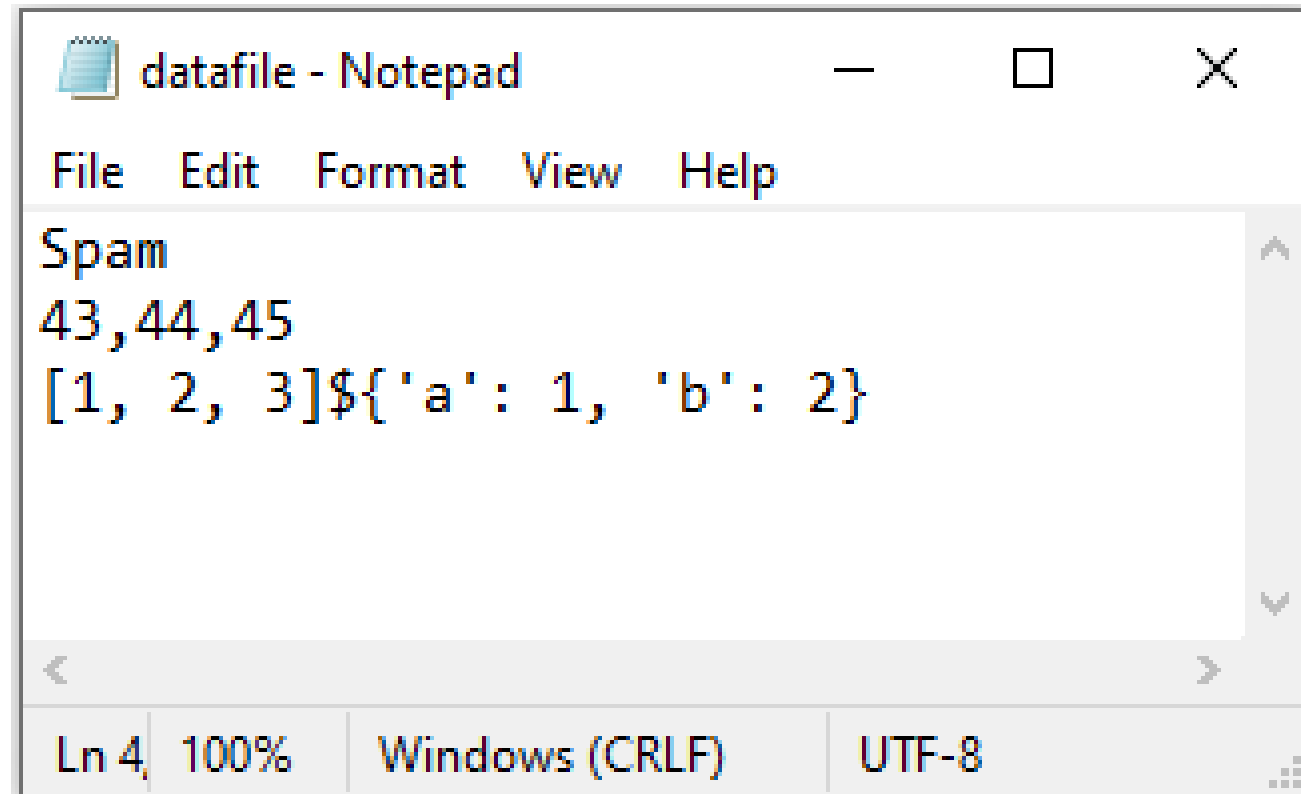
```
Binh yen
nam trong tam tri
```

4. LƯU ĐỐI TƯỢNG PYTHON VÀO FILE

- Để lưu các đối tượng của Python vào file, ta phải chuyển đổi thành chuỗi.

```
x,y,z = 43, 44, 45
st = 'Spam'
Dict = {'a': 1, 'b': 2}
lst = [1, 2, 3]

f=open('datafile.txt','w')           # tạo file
f.write(st+'\n')                     # kết thúc dòng với '\n'
f.write('%s,%s,%s\n'%(x,y,z))        # chuyển đổi số thành chuỗi
f.write(str(lst)+'$'+str(Dict)+'\n')  # chuyển đổi chuỗi và
                                     # ngăn cách bằng '$'
f.close()
```



```
datafile - Notepad
File Edit Format View Help
Spam
43,44,45
[1, 2, 3]${'a': 1, 'b': 2}
```

Ln 4, 100% Windows (CRLF) UTF-8

- Để đọc file và nhận lại các đối tượng Python, ta phải thực hiện chuyển đổi:

```
f=open('datafile.txt')  
line=f.readline().rstrip()  
print(line)
```

Spam

```
line=f.readline().rstrip()  
parts=line.split(',')  
numbers=[int(p) for p in parts]  
numbers
```

[43, 44, 45]

```
line=f.readline().rstrip()  
parts=line.split('$')  
objects=[eval(p) for p in parts]  
objects
```

[[1, 2, 3], {'a': 1, 'b': 2}]

LƯU ĐỐI TƯỢNG PYTHON VỚI THƯ VIỆN **PICKLE**

```
import pickle
Dict = {'a': 1, 'b': 2}
f=open('datafile.pkl','wb')
pickle.dump(Dict,f)           # Lưu đối tượng bất kì vào file
f.close()
```

```
f=open('datafile.pkl','rb')
e=pickle.load(f)             # Tải đối tượng bất kì từ file
e

{'a': 1, 'b': 2}
```

LƯU ĐỐI TƯỢNG PYTHON THEO ĐỊNH DẠNG **JSON**

- JSON là một định dạng trao đổi dữ liệu mới, được hỗ trợ bởi nhiều hệ thống.
- JSON rất gần với từ điển Python và danh sách theo cú pháp, nên việc dịch sang và từ các đối tượng Python là dễ dàng và được thực hiện một cách tự động bởi thư viện chuẩn **json**.

- Lưu đối tượng từ điển trong Python:

```
name = dict(first='Bob', last='Smith')
rec = dict(name=name, job=['dev', 'mgr'], age=40.5)
rec
```

```
{'name': {'first': 'Bob', 'last': 'Smith'}, 'job': ['dev', 'mgr'],
'age': 40.5}
```

- Chuyển định dạng JSON:

```
import json
S=json.dumps(rec)
S
```

```
'{"name": {"first": "Bob", "last": "Smith"}, "job": ["dev", "mgr"], "age": 40.5}'
```


- Lưu định dạng JSON vào file:

```
import json
fp=open('testjson.txt', 'w')
json.dump(rec, fp , indent=4)
fp.close()
```

- Tải đối tượng Python từ file:

```
P=json.load(open('testjson.txt'))
print(P)
type(P)
```

```
{'name': {'first': 'Bob', 'last': 'Smith'}, 'job': ['dev', 'mgr'],
'age': 40.5}
```

```
dict
```

```
testjson - Notepad
File Edit Format View Help
{
    "name": {
        "first": "Bob",
        "last": "Smith"
    },
    "job": [
        "dev",
        "mgr"
    ],
    "age": 40.5
}
110% Windows (CRLF) UTF-8
```

Thực hành:

- **Bài tập C3-22:** Viết một chương trình thực hiện các thao tác sau đây

- + Đọc file ‘**mailbox.txt**’ và in ra nội dung

Nếu file không tồn tại trong thư mục làm việc, thì đưa ra thông báo:

“Không thể mở file: mailbox.txt ”

- + Lọc và in ra nội dung các dòng bắt đầu từ cụm từ “**From:**”

- **Bài tập C3-23:** Các file '**2012boysnames.txt**' và '**2012girlsnames.txt**' lưu trữ tên của các bé trai và bé gái sinh năm 2012. Hãy viết một chương trình đọc các file này và đưa ra thông tin về các tên bé trai, bé gái có độ dài ngắn nhất, dài nhất.
- **Bài tập C3-24:** File '**elements.txt**' lưu trữ kí hiệu viết tắt và tên của các nguyên tố hóa học. Hãy viết một chương trình thực hiện:
 - + đọc file và tạo ra một từ điển gồm các mục từ có
 - key** = kí hiệu nguyên tố hóa học,
 - value** = tên nguyên tố hóa học.
 - + cho người dùng nhập một kí hiệu hóa học vào từ bàn phím và hiển thị tên của nguyên tố hóa học tương ứng.

■ **Bài tập C3-25:** Viết một chương trình thực hiện các thao tác sau đây:

- + Đọc file '**goldprice.csv**' (giá vàng theo tháng) và hiển thị nội dung.
- + Nhập vào năm muốn xem giá vàng.
- + Lọc và in ra giá vàng theo các tháng trong năm đó.
- + Tính giá vàng trung bình của năm đó.

Minh họa:

Nhập vào năm tra cứu: 1952



| |
|-----------------|
| 1952-01, 34.490 |
| 1952-02, 34.490 |
| 1952-03, 34.680 |
| 1952-04, 34.820 |
| 1952-05, 34.730 |
| 1952-06, 34.530 |
| 1952-07, 34.570 |
| 1952-08, 34.580 |
| 1952-09, 34.510 |
| 1952-10, 34.630 |
| 1952-11, 34.720 |
| 1952-12, 34.790 |



giá vàng trung bình = 34,628

- **Bài tập C3-26:** '**heart_rate.txt**' là file text lưu trữ dữ liệu về nhịp tim của một người vận động gắng sức trong khoảng thời gian 20 phút:

```
Time (sec),Heart Rate (BPM)
0.0, 103
5.0, 103
10.0, 103
15.0, 105
20.0, 108
```

Hãy viết một chương trình thực hiện các công việc sau đây:

- (a) Đọc file và in ra 5 dòng đầu tiên không kể tiêu đề.
- (b) In ra tóm tắt thống kê về dữ liệu Heart Rate: trung bình, độ lệch tiêu chuẩn, giá trị nhỏ nhất và lớn nhất, trung vị.
- (c) Tìm đường hồi quy tuyến tính phù hợp nhất mô tả Heart Rate theo Time.
- (d) Lưu kết quả ở (b) và (c) vào một file có tên '**results.txt**'