



HCMUTE

TRƯỜNG ĐẠI HỌC

**SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

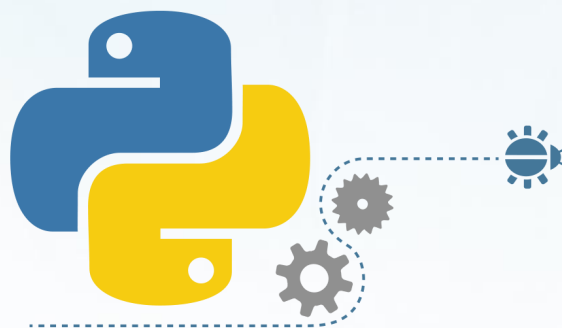
HCMC University of Technology and Education



KHOA CÔNG NGHỆ THÔNG TIN  
**BỘ MÔN HỆ THỐNG THÔNG TIN**

# **NHẬP MÔN LẬP TRÌNH PYTHON (IPPA233277)**

## **XỬ LÝ TẬP TIN**



**GV. Trần Quang Khải**

1. Hiểu được lý do vì sao phải lưu và đọc tập tin
2. Phân biệt được các loại tập tin thông dụng
3. Các thao tác trên các loại tập tin khác nhau (XML, JSON, CSV, Excel,...)



1. Giới thiệu
2. Thao tác trên tập tin
3. Xử lý tập tin XML
4. Xử lý tập tin JSON
5. Xử lý tập tin CSV
6. Xử lý tập tin Excel



- Tập tin (file) và thư mục (folder) là thành phần cơ bản của hệ thống lưu trữ dữ liệu bền vững
- Một tập tin chứa thông tin hoặc dữ liệu được lưu trên thiết bị lưu trữ của máy tính
- Python cung cấp khả năng xử lý tập tin như thành phần cơ bản của ngôn ngữ, được chia thành 2 tác vụ chính:
  - ✓ Tác vụ quản lý: không ảnh hưởng đến nội dung (đổi tên, di chuyển, xóa, sao chép, phân quyền,...). Python cung cấp nhiều hàm thuộc thư viện `os` (`import os`)
  - ✓ Tác vụ nội dung: có tương tác với nội dung tập tin (đọc, ghi,...). Python thực hiện theo quy trình 03 bước: mở tập tin – xử lý – đóng tập tin.



- Python chia tập tin thành 02 loại:
  - ✓ Tập tin văn bản:
    - Chứa nội dung chủ yếu là text và các định dạng trình bày (tab, xuống dòng, căn lề,...).
    - Được cấu trúc như một dãy các dòng, mỗi dòng là dãy các ký tự và một dòng tối thiểu là một ký tự dù cho đó là dòng trống
    - Các dòng được ngăn cách bởi một ký tự “new line” và mặc định là “\n”
  - ✓ Tập tin nhị phân (binary file): Python xem như các dãy byte dữ liệu và thường thao tác theo các khối dữ liệu để tăng tốc xử lý
- Có nhiều loại tập tin khác nhau: text-file, XML, JSON, CSV, Excel,...



- Làm việc với tập tin gồm 03 bước:
  - ✓ Mở tập tin: yêu cầu hệ thống chuẩn bị các điều kiện cần thiết để đọc/ghi nội dung tập tin bao gồm định vị dữ liệu trên vùng lưu trữ và khởi tạo các bộ nhớ đệm
  - ✓ Làm việc với tập tin: là bước chính của quá trình thực hiện các thao tác liên quan đến nội dung tập tin
  - ✓ Đóng tập tin: đảm bảo các thay đổi được cập nhật lên vùng lưu trữ và giải phóng các tài nguyên được cấp phát trong quá trình làm việc với tập tin
- Các bước này có thể phát sinh IOError





- Thao tác mở tập tin tốn nhiều thời gian cho việc:
  - ✓ Kiểm tra người dùng có mở quá nhiều tập tin
  - ✓ Kiểm tra tập tin có tồn tại trong hệ thống
  - ✓ Kiểm tra chương trình có quyền truy cập nội dung
  - ✓ Kiểm tra có thể thao tác với tập tin trong thời điểm hiện tại vì:
    - Tập tin có thể bị khóa bởi chương trình khác
    - Tập tin có thể chỉ đọc vì được ghi trên thiết bị không cho phép ghi dữ liệu
    - Tập tin có thể chỉ ghi vì nó là loại thiết bị không cho phép đọc
- Do đó, cần mở tập tin khi cần thiết và chọn loại tập tin phù hợp với mục đích sử dụng



- Cú pháp:
  - `open(file, [, access_mode][, buffering])`
- buffering: là 0 nghĩa là sẽ không có trình đệm nào diễn ra; 1 thì trình đệm dòng được thực hiện khi truy cập; với số nguyên > 1, thì hoạt động đệm thực hiện với kích cỡ bộ đệm đã cho; với số âm, kích cỡ bộ đệm sẽ là mặc định
- access\_mode được cho trong bảng bên
- Thuộc tính của tập tin:
  - ✓ File.closed: Trả về True nếu file đã đóng, ngược lại là False
  - ✓ File.mode: Trả về chế độ truy cập của file đang được mở
  - ✓ File.name: Trả về tên của file





MODE	Ý NGHĨA	MODE	Ý NGHĨA
r	Mở file chỉ để đọc	w	Mở tập tin văn bản để ghi, nếu tập tin không tồn tại sẽ tạo mới
r+	Mở file để đọc và ghi	w+	Mở tập tin văn bản để đọc và ghi, nếu tập tin không tồn tại thì sẽ tạo mới
rb	Mở file trong chế độ đọc cho định dạng nhị phân, đây là chế độ mặc định. Con trỏ tại phần bắt đầu của file	wb	Mở tập tin nhị phân để ghi, nếu tập tin không tồn tại thì sẽ tạo mới
rb+	Mở file để đọc và ghi trong định dạng nhị phân. Con trỏ tại phần bắt đầu của file	wb+, w+b	Mở tập tin nhị phân để đọc và ghi, nếu tập tin không tồn tại thì sẽ tạo mới
a	Mở tập tin văn bản để ghi tiếp vào cuối nếu tập tin đã tồn tại, nếu tập tin không tồn tại thì sẽ tạo mới	x+	Tạo tập tin văn bản mới để đọc và ghi, sinh lỗi nếu tập tin đã tồn tại
ab	Mở tập tin nhị phân để ghi tiếp vào cuối nếu tập tin đã tồn tại, nếu tập tin không tồn tại thì sẽ tạo mới	xb+, x+b	Tạo tập tin nhị phân mới để đọc và ghi, sinh lỗi nếu tập tin đã tồn tại
ab+, a+b	Mở tập tin nhị phân để đọc và ghi tiếp vào cuối nếu tập tin đã tồn tại, nếu tập tin không tồn tại thì sẽ tạo mới	b	Mở tập tin nhị phân để đọc
x	Tạo tập tin văn bản mới để ghi, sinh lỗi nếu tập tin đã tồn tại	t	Mở tập tin văn bản để đọc (đây là giá trị mặc định của mode khi gọi hàm open)



- Thao tác đóng tập thực hiện những công việc sau:
  - ✓ Đẩy mọi dữ liệu trên vùng đệm xuống thiết bị lưu trữ
  - ✓ Cập nhật thông tin trên hệ thống (file size, last update, ...)
  - ✓ Giải phóng vùng dữ liệu dùng cho quá trình làm việc với tập tin
- Sử dụng từ khóa with giúp tự động đóng tập tin

```
with open("test.txt", encoding = 'utf-8') as f:  
    # thực hiện các thao tác với tệp  
    ...  
    # biến f bị hủy, tập tin được tự động đóng lại
```

- Khi quên đóng tập tin:
  - ✓ Gây hao tổn tài nguyên lưu trữ trên bộ nhớ chính, đặc biệt các tập tin có dung lượng lớn
  - ✓ Hệ điều hành sẽ khóa tập tin và ngăn cho các chương trình khác truy xuất.
  - ✓ Tuy nhiên, chương trình sẽ tự động đóng lại tất cả các tập tin đang mở khi kết thúc chương trình



- Cấu trúc thông dụng khi đóng tập tin

```
try:  
    # mở tập tin  
    f = open("test.txt", encoding = 'utf-8')  
    # thực hiện các thao tác với thư mục  
    ...  
finally:  
    # đóng tập tin  
    f.close()
```



- Sử dụng hàm `read(n)` để đọc `n` byte tiếp theo
- Sử dụng hàm `readline(n)` để đọc một dòng từ tập tin, tối đa `n` byte. Nếu không cung cấp giá trị `n`, thì trả về chuỗi là dữ liệu được đọc tới khi gặp ký tự hết dòng trừ dòng cuối (`\n`) hoặc hết tập tin (EOF)
- Sử dụng hàm `readlines(n)` để đọc các dòng cho đến hết tập tin và trả về một danh sách các chuỗi, nếu có giá trị `n` thì đọc tối đa `n` byte
- Đọc tất cả các dòng của tập tin

```
for line in open("test.txt", encoding = 'utf-8'):
```

```
    # thực hiện các thao tác với từng dòng
```

```
    ...
```

```
with open('workfile') as f:
```

```
    for line in f:
```

```
        print(line, end = '')
```



- Sử dụng hàm `write(data)` để ghi dữ liệu (`data`) vào trong tập tin và trả về số byte ghi được
  - Làm việc được với tập tin văn bản và tập tin nhị phân
  - Nếu tập tin văn bản, `data` phải là kiểu chuỗi
  - Nếu tập tin nhị phân, `data` phải là khối byte (kiểu `bytearray` hoặc kiểu `bytes`)
- Sử dụng hàm `writelines(data)` để ghi toàn bộ nội dung dữ liệu vào tập tin theo từng dòng:
  - Chỉ làm việc với kiểu file văn bản dữ liệu `data` phải là danh sách các `str`
  - Nếu cố dùng kiểu dữ liệu khác sẽ phát sinh lỗi `TypeError`



- Để xóa tập tin, cần sử dụng thư viện os bằng cách khai báo **import os**
- Xóa tập tin: **os.remove("demofile.txt")**
- Kiểm tra tập tin có tồn tại trong hệ thống **os.path.exists("demo.txt")**
- Xóa thư mục (folder): **os.rmdir("myfolder")**





- XML có thể dùng DOM hoặc SAX để đọc

Ví dụ:

```
<?xml version="1.0" encoding="UTF-8" ?>
<employees>
  <employee>
    <id>1</id>
    <name>Trần Duy Thanh</name>
  </employee>
  <employee>
    <id>2</id>
    <name>Lê Hoàng Sử</name>
  </employee>
  <employee>
    <id>3</id>
    <name>Hồ Trung Thành</name>
  </employee>
</employees>
```

```
from xml.dom.minidom import parse
import xml.dom.minidom
```

*# Mở file xml bằng minidom parser*

```
DOMTree = xml.dom.minidom.parse("employees.xml")
```

```
collection = DOMTree.documentElement
```

*# Lấy tất cả tag là employee*

```
employees = collection.getElementsByTagName("employee")
```

*# Duyệt vòng lặp để lấy toàn bộ dữ liệu ra*

```
for employee in employees:
```

```
    tag_id = employee.getElementsByTagName('id')[0]
```

```
    id=tag_id.childNodes[0].data
```

```
    tag_name = employee.getElementsByTagName('name')[0]
```

```
    name=tag_name.childNodes[0].data
```

```
    print(id, '\t', name)
```



- Chuyển đổi đối tượng Python sang chuỗi JSON

Ví dụ:

```
import json

pythonObject = {
    "ten": "Trần Duy Thanh",
    "tuoi": 50,
    "ma": "nv1"
}

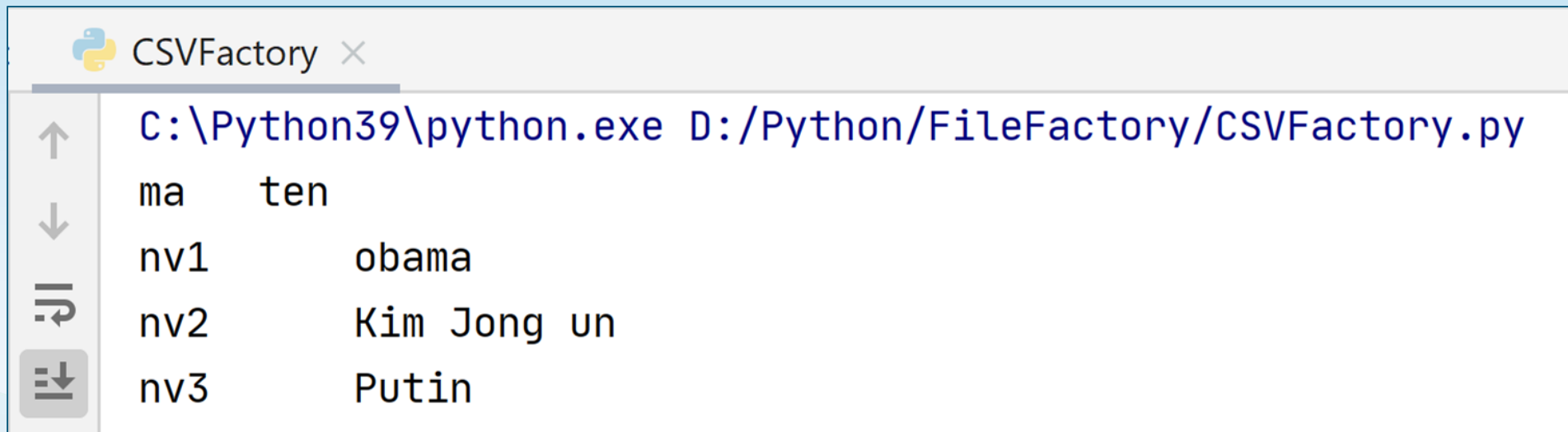
jsonString = json.dumps(pythonObject)

print(jsonString)
```



```
import csv

with open('datacsv.csv', newline='') as f:
    reader = csv.reader(f, delimiter=';', quoting=csv.QUOTE_NONE)
    for row in reader:
        print(row[0], "\t", row[1])
```



```
CSVFactory x
C:\Python39\python.exe D:/Python/FileFactory/CSVFactory.py
ma      ten
nv1      obama
nv2      Kim Jong un
nv3      Putin
```



- Bên cạnh đó, cũng có nhiều thư viện hỗ trợ việc tương tác với tập tin excel như `xlsxwriter`, `openpyxl`,...

```
import xlsxwriter
```

```
import xlsxwriter
```

```
# thêm một dòng dữ liệu
```

```
worksheet.write('A2',1)  
worksheet.write('B2','SP1')  
worksheet.write('C2', 'Coca')  
worksheet.write('D2', '15')  
worksheet.write('E2', '15000')
```

```
# thêm một dòng dữ liệu
```

```
worksheet.write('A3',2)  
worksheet.write('B3','SP2')  
worksheet.write('C3', 'Pepsi')  
worksheet.write('D3', '20')  
worksheet.write('E3', '18000')
```

```
# chèn Logo vào
```

```
worksheet.insert_image('B5', 'logo_UEL.png')
```

```
# tạo một file excel cùng 1 sheet
```

```
workbook = xlsxwriter.Workbook('demo.xlsx')  
worksheet = workbook.add_worksheet()
```

```
# thiết lập các cột cho file
```

```
worksheet.set_column('A:A', 5)  
worksheet.set_column('B:B', 15)  
worksheet.set_column('C:C', 20)  
worksheet.set_column('D:D', 15)  
worksheet.set_column('E:E', 15)
```

```
# định dạng tiêu đề cột in đậm
```

```
bold = workbook.add_format({'bold': True})
```

```
# thêm dòng tiêu đề và định dạng in đậm
```

```
worksheet.write('A1', 'STT',bold)  
worksheet.write('B1', 'MÃ SẢN PHẨM',bold)  
worksheet.write('C1', 'TÊN SẢN PHẨM',bold)  
worksheet.write('D1', 'SỐ LƯỢNG',bold)  
worksheet.write('E1', 'ĐƠN GIÁ',bold)
```

```
workbook.close()
```



- Đọc dữ liệu từ excel

```
from openpyxl import load_workbook
wb = load_workbook('demo.xlsx')
print (wb.sheetnames)
ws = wb[wb.sheetnames[0]]
for row in ws.values:
    for value in row:
        print(value, "\t", end= ' ')
    print("")
```



1. Với tác vụ nội dung, Python thực hiện theo quy trình?

A. Mở tập tin – Xử lý – Đóng tập tin

B. Mở tập tin – Xử lý – Lưu trữ – Đóng

C. Tìm kiếm – Mở – Xử lý – Lưu trữ - Đóng

D. Tất cả đều sai

2. Thuộc tính trong tập tin cho biết tập tin mở hay đóng

A. File.closed

B. File.mode

C. File.name

D. File.state

3. Chế độ mode = w cho biết điều gì?

A. Mở tập tin văn bản để ghi, nếu không tồn tại sẽ tạo mới

B. Mở tập tin văn bản để ghi, nếu không tồn tại sẽ báo lỗi

C. Mở tập tin để đọc & ghi, nếu không tồn tại sẽ tạo mới

D. Mở tập tin để đọc & ghi, nếu không tồn tại sẽ báo lỗi

4. Thao tác nào sau đây không nằm ở bước mở tập tin?

A. Kiểm tra tập tin có tồn tại trong hệ thống

B. Kiểm tra quyền của tập tin cần truy cập

C. Kiểm tra người dùng có mở quá nhiều tập tin

D. Kiểm tra thông tin về dung lượng và định vị tập tin

5. Thao tác nào sau đây không nằm ở bước đóng tập tin

A. Đưa dữ liệu vào bộ nhớ đệm để xuống thiết bị lưu trữ

B. Cập nhật thông tin của tập tin

C. Xóa các vùng nhớ phát sinh trong quá trình xử lý

D. Phát thông báo đến các chương trình liên quan





- ✓ Họ tên : **Trần Quang Khải**
- ✓ Email : **khaitq@hcmute.edu.vn**
- ✓ Zalo (mã Qr)

