



**TRƯỜNG ĐẠI HỌC THƯƠNG MẠI**  
**THUONGMAI UNIVERSITY**

# **CHƯƠNG 5**

## **LÀM VIỆC VỚI TẬP TIN**



# MỞ ĐẦU

**Bài toán:** Xây dựng chương trình nhập vào một danh sách sinh viên của một lớp, hiển thị ra màn hình danh sách sinh viên vừa nhập. Biết rằng mỗi sinh viên gồm các thông tin: Mã sinh viên, họ tên, lớp, quê quán.

## Thuật toán

Bước 1: Tạo danh sách trống để lưu thông tin của các sinh viên.

Bước 2: Nhập mã sinh viên.

Bước 3: Nếu mã sinh viên trống thì kết thúc việc nhập và chuyển đến Bước 6.

Bước 4: Nhập họ tên, lớp, quê quán của sinh viên.

Bước 5: Thêm sinh viên vào danh sách và quay lại Bước 2.

Bước 6: Hiển thị danh sách sinh viên vừa nhập ra màn hình.

Bước 7: Kết thúc.



# MỞ ĐẦU

- Thực hiện:

```
DSSV=[] #Bước1
#Nhập danh sách sinh viên từ bàn phím
while(True) :
    maSV = input("Mã sinh viên: ") #Bước 2
    if maSV=="": #Bước 3
        break
    tenSV = input("Tên sinh viên: ") #Bước 4
    Lop = input("Lớp: ") #Bước 4
    QueQuan = input("Quê quán: ") #Bước 4
    DSSV.append([maSV, tenSV, Lop, QueQuan]) #Bước 5
#Bước 6 Hiển thị danh sách sinh viên ra màn hình
print('Danh sách sinh viên vừa nhập')
print("\t".join([ "Mã SV", 'Họ tên SV', 'Lớp ', 'Quê quán' ]))
for sv in DSSV:
    print("\t".join(sv)) #Bước 7
```



# MỞ ĐẦU

## → Nhận xét:

- Chương trình cho phép người dùng nhập một danh sách sinh viên và hiển thị danh sách đó ra màn hình.
- Tuy nhiên, mỗi khi chạy lại chương trình thì danh sách sinh viên vừa nhập mất đi và phải nhập lại từ đầu.

→ **Câu hỏi:** làm thế nào để lưu thông tin vừa nhập để mỗi lần chạy lại chương trình dữ liệu nhập vào không bị mất đi

## → Giải pháp:

- Lưu dữ liệu đó vào tệp (tổng quát hơn là lưu trữ ở bộ nhớ ngoài);
- Hiển thị danh sách sinh viên chỉ cần đọc dữ liệu từ tệp.



# Nội dung

5.1 Làm việc với tệp tin văn bản

5.2 Làm việc với các kiểu tệp tin khác

5.2.1. Tệp tin có định dạng .csv

5.2.2. Làm việc với file hình ảnh



## 5.1. LÀM VIỆC VỚI TỆP TIN VĂN BẢN

- Giới thiệu:
- Tập (File): là một vùng nhớ trên ổ đĩa của máy tính được quản lý bởi tên; tập dùng để lưu trữ dữ liệu và dữ liệu lưu trong tập sẽ không bị mất khi chương trình kết thúc hoặc tắt máy tính.
- Phân loại: tập văn bản, tập nhị phân, ...
  - Tập văn bản có ưu điểm dễ đọc và phân tích, nhưng chạy khá chậm và dễ bị hack
  - Tập nhị phân: dữ liệu được lưu trữ hiệu quả hơn (tiết kiệm bộ nhớ), an toàn hơn, nhưng các thao tác trên nhị phân khó giải quyết hơn tập văn bản.



## 5.1. LÀM VIỆC VỚI TỆP TIN VĂN BẢN

- Thao tác làm việc với tệp dữ liệu:
  - Bước 1: Mở (tạo) tệp dữ liệu
  - Bước 2: Đọc (ghi) dữ liệu từ (vào) tệp
  - Bước 3: Đóng tệp để các tài nguyên gắn với tệp được giải phóng.



## 5.1. LÀM VIỆC VỚI TỆP TIN VĂN BẢN

5.1.1. Các thao tác cơ bản

5.1.2. Đọc tệp tin

5.1.3. Ghi dữ liệu lên tệp tin





## 5.1.1. CÁC THAO TÁC CƠ BẢN

- Tạo/mở tệp tin văn bản:

**<Tên biến tệp> = open(file\_name, mode, encoding)**

Trong đó:

- **<Tên biến tệp>**: là tên biến của đối tượng tệp do người dùng đặt theo quy tắc đặt tên của Python
- **File\_name**: là đường dẫn trỏ đến tệp cần mở (tạo mới nếu chưa có)
- **Mode**: là chế độ mở tệp
- **Encoding**: Kiểu mã hóa khi đọc (hoặc ghi) tệp, nếu muốn dùng tiếng Việt thì đặt encoding = “utf-8”



## 5.1.1. CÁC THAO TÁC CƠ BẢN

- Các chế độ mở tệp

Chế độ	Mô tả
'r'	Mở tệp để đọc dữ liệu (chế độ mặc định)
'w'	Mở tệp để ghi dữ liệu vào tệp; nếu tệp chưa tồn tại thì sẽ tạo tệp mới, nếu tệp đã tồn tại thì tất cả dữ liệu cũ trong tệp sẽ bị xóa
'x'	Mở một tệp đã tồn tại trên ổ đĩa, nếu tệp chưa tồn tại sẽ phát sinh lỗi
'a'	Mở tệp để ghi dữ liệu vào tệp; nếu tệp chưa tồn tại thì sẽ tạo tệp, nếu tệp đã tồn tại thì sẽ ghi dữ liệu vào cuối tệp (ghi nối đuôi)
't'	Mở tệp ở chế độ văn bản
'b'	Mở tệp ở chế độ nhị phân (binary)
'+'	Mở tệp để đọc + ghi dữ liệu vào file



## 5.1.1. CÁC THAO TÁC CƠ BẢN

- Ví dụ 1: Tại ổ D tạo tệp tin văn bản có tên test.txt để ghi dữ liệu

*f = open("D:/test.txt", 'wt')*

Hoặc

*f = open("D:/test.txt", 'at')*

- Ví dụ 2: Mở tệp tin có tên test.txt để đọc dữ liệu và chỉ rõ nội dung được mã hóa dạng utf-8

*f1 = open("D:/test.txt", 'rt', encoding='utf-8')*



## 5.1.1. CÁC THAO TÁC CƠ BẢN

- Ví dụ 3: Mở tệp tin có tên “image.jpg” để đọc và ghi dữ liệu ở chế độ nhị phân

*fp = open(“image.jpg”, ‘r+b’)*

- Ví dụ 4: Đọc file nhị phân “data.bin”

*fp = open(“data.bin”)*



## 5.1.1. CÁC THAO TÁC CƠ BẢN

- **Đóng tập tin:**

**<Tên biến tệp>.close()**

- **Lưu ý:** Trong quá trình thao tác với tập tin có thể phát sinh ngoại lệ khiến tập tin không đóng khi kết thúc thao tác.
- Giải pháp giải quyết là sử dụng khối **try .... finally** hoặc từ khóa **with** (tập tin tự động đóng khi không làm việc với file).



## 5.1.1. CÁC THAO TÁC CƠ BẢN

- Ví dụ 5: Sử dụng try ... except trong mở/đóng tập tin

```
if __name__ == "__main__":  
    try:  
        f= open("datda.txt",mode='r',encoding="utf-8")  
        print(f.read(256))  
    except :  
        print(" open file error ")
```



## 5.1.1. CÁC THAO TÁC CƠ BẢN

- Ví dụ 5: Sử dụng try ... except trong mở/đóng tập tin

try:

```
f= open("datda.txt",mode='r',encoding="utf-8")
```

```
print(f.read(256))
```

except NameError:

```
print(NameError)
```



## 5.1.1. CÁC THAO TÁC CƠ BẢN

- Ví dụ 5: Sử dụng try ... finally trong mở/đóng tập tin

try:

```
logFile= "E:\\log.txt"
```

```
f= open(logFile,mode='w',encoding="utf-8")
```

```
f.write(" some message ")
```

```
f.close()
```

except Exception as e:

```
f.write(" an error message ")
```

```
f.close()
```

finally:

```
print("finish ")
```





## 5.1.1. CÁC THAO TÁC CƠ BẢN

- Ví dụ 5: Sử dụng try ... finally trong mở/đóng tập tin

```
if __name__ == "__main__":  
    try:  
        logFile= "E:\\filename.txt"  
        f= open(logFile,mode='r',encoding="utf-8")  
    except Exception as e:  
        print(" an error message ", e)  
    finally:  
        print(" function is not performed ")
```



## 5.1.1. CÁC THAO TÁC CƠ BẢN

- Trong quá trình mở dữ liệu trên file đôi khi bị lỗi do nội dung được mã hóa dưới dạng utf-8 hoặc ascii, ... bạn có thể bỏ qua lỗi hoặc thay thế ký tự lỗi.
- Cú pháp:  
`open('filename', 'mode', errors= 'replace/ignore')`



## 5.1.1. CÁC THAO TÁC CƠ BẢN

- Ví dụ:

```
>>> # Replace bad chars with Unicode U+fffd replacement char
>>> f = open('sample.txt', 'rt', encoding='ascii', errors='replace')
>>> f.read()
'Spicy Jalape?o!'
>>> # Ignore bad chars entirely
>>> g = open('sample.txt', 'rt', encoding='ascii', errors='ignore')
>>> g.read()
'Spicy Jalapeo!'
>>>
```



## 5.1.2. ĐỌC TỆP TIN

- Trình tự đọc dữ liệu từ tệp tin:
  - **Bước 1:** Mở tệp ở chế độ đọc
    - Ví dụ: `f = open("D:/test.txt", 'rt', encoding='utf-8')`
  - **Bước 2:** Đọc dữ liệu
    - Thực hiện các thao tác đọc dữ liệu theo mục đích bài toán
  - **Bước 3:** Đóng tệp
    - Ví dụ: `f.close()`



## 5.1.2. ĐỌC TỆP TIN

- **Thực hiện:** *bước 2 là bước quan trọng nhất.*
  - Bước 2: Sử dụng phương thức **read()** hoặc **readlines()** hoặc **readline()** của đối tượng tệp để đọc dữ liệu từ tệp theo cú pháp:

<Tên biến tệp>.read(n)  
<Tên biến tệp>.readline()  
<Tên biến tệp>.readlines()

- Trong đó:
  - **Read(n)**: đọc n ký tự từ tệp, nếu n=0 hoặc n>kích thước của tệp thì sẽ đọc toàn bộ tệp.
  - **Readline()**: đọc và trả về 1 dòng dữ liệu từ tệp
  - **Readlines()**: Đọc và trả về 1 danh sách, mỗi phần tử của danh sách là một dòng dữ liệu từ tệp.



## 5.1.2. ĐỌC TỆP TIN

- Ví dụ 7: Mở file “data.bin” để đọc và ghi dữ liệu dưới dạng mã utf-8

```
f = open('E:/data.bin', 'wb')
text = 'Ngôn ngữ lập trình Python'
f.write(text.encode('utf - 8'))
f.close()
f = open("E:/data.bin", 'rb')
databin = f.read(256)
datatext = databin.decode('utf - 8')
f.close()
print(" Dữ liệu mã hóa: ", databin)
print(" Dữ liệu giải mã: ", datatext)
```

```
Dữ liệu mã hóa: b'Ng\xc3\xb4n ng\xe1p tr\xednh Python'
Dữ liệu giải mã: Ngôn ngữ lập trình Python
```



## 5.1.2. ĐỌC TỆP TIN

- Ví dụ 7: dùng hàm `readline()`

```
f = open("E:/data.bin", 'rb')
databin = f.readline()
datatext = databin.decode('utf - 8')
f.close()
print(" Dữ liệu mã hóa: ", databin)
print(" Dữ liệu giải mã: ", datatext)
```

```
Dữ liệu mã hóa:  b'Ng\xc3\xb4n ng\xe1\xbb\xaf l\xe1\xba\xadp tr\xc3\xacnh Python'
Dữ liệu giải mã:  Ngôn ngữ lập trình Python
```



## 5.1.2. ĐỌC TỆP TIN

- Ví dụ 8: Ghi câu sau vào tệp “test1.bin”:  
“Không có gì quý hơn độc lập tự do hạnh phúc”  
Sau đó **mở file, đọc và hiển thị** nội dung đó ra màn hình

- **Thực hiện:**

```
f = open('d:/test1.bin','wb')
```

```
f.write(repr("Không có gì quý hơn độc lập tự do hạnh phúc").encode('utf-8'))
```

```
f.close()
```

```
with open('d:/test1.bin','rb') as fr:
```

```
    for line in fr:
```

```
        print(line.decode('utf-8'))
```





## 5.1.2. ĐỌC TỆP TIN

- Ví dụ 9: Mở file văn bản và duyệt nội dung file theo từng dòng:

```
with open('text_line.txt') as lines:  
    for line in lines:  
        line = line[0:len(line)-1]  
    print(line)
```



## 5.1.3. GHI DỮ LIỆU VÀO TỆP

- **Thực hiện:**
  - **Bước 1:** Mở tệp ở chế độ '**w**' hoặc '**x**' hoặc '**+**' hoặc '**a**'
  - **Bước 2:** Sử dụng phương thức **write()** hoặc **writelines()** để ghi dữ liệu vào tệp theo cú pháp:

```
<Tên biến tệp>.write(str)  
<Tên biến tệp>.writelines(lines)
```

- Trong đó:
  - str: là một chuỗi ký tự chứa dữ liệu cần ghi vào tệp
  - lines: là một danh sách, mỗi phần tử của danh sách là một chuỗi ký tự
- **Bước 3:** Đóng tệp bằng câu lệnh <tên biến tệp>.close()



## 5.1.3. GHI DỮ LIỆU VÀO TỆP

- Ví dụ 6: Ghi nội dung sau vào tệp văn bản test.txt

Đường vô xứ Huế loanh quanh  
Non xanh nước biếc như tranh họa đồ

→ Thực hiện:

- Mở tệp tin để ghi:

```
f = open('D:\\test.txt', 'wt')
```

- Ghi dữ liệu vào tệp:

```
f.writelines(['Đường vô xứ Huế loanh quanh \n', 'Non  
xanh nước biếc như tranh họa đồ'])
```

- Đóng tệp tin:

```
f.close()
```



## 5.1.3. GHI DỮ LIỆU VÀO TỆP

- Ví dụ 6:

```
f = open('E:\\test.txt', 'wt', encoding="utf-8")
f.writelines(["Đường vô xứ Huế quanh \n", "Non xanh nước biếc
như tranh họa đồ "])
f.close()
with open('E:\\test.txt', encoding="utf-8") as lines:
    for line in lines:
        line = line[0:len(line)-1]
        print(line)
f.close()
```



## 5.1.3. GHI DỮ LIỆU VÀO TỆP

- Ví dụ: Viết chương trình Python thực hiện các công việc sau:
  - Nhập thông tin về một danh sách sinh viên, mỗi sinh viên gồm: Mã sinh viên, họ tên, quê quán.
  - Lưu thông tin về danh sách sinh viên này vào tệp `sinhvien.txt`
  - Đọc và hiển thị thông tin về sinh viên lưu trong tệp `sinhvien.txt` ra màn hình

## 5.1.3. GHI DỮ LIỆU VÀO TẬP

### Thực hiện:

#Nhập ghi thông tin về DSSV vào tập văn bản sinh viên.txt

```
file = open('d:/sinhvien.txt','a',encoding='utf-8')
# Nhập ds sinh viên từ bàn phím
while (True):
    maSV = input("Mã sinh viên: ")
    if (maSV == ""):
        break
    tenSV = input("Tên sinh viên: ")
    Lop = input("Lớp: ")
    QueQuan = input("Que quan: ")
    file.write('\t'.join([maSV,tenSV,Lop,QueQuan])+'\n')
file.close()
```

#Hiển thị danh sách sinh viên ra màn hình

```
print('Danh sách sinh viên đã nhập')
print('\t'.join(['maSV','tenSV','\t','Lop','QueQuan']))
file = open('d:/sinhvien.txt','r',encoding='utf-8')
for sv in file.readlines():
    print(sv.replace('\n',''))
file.close()
```



## 5.2. LÀM VIỆC VỚI CÁC KIỂU TỆP TIN KHÁC

5.2.1. Tập tin có định dạng .csv

5.2.2. Làm việc với file hình ảnh



## 5.2.1. Tập tin có định dạng .csv

- Cho file stock.csv với nội dung sau:

```
Symbol,Price,Date,Time,Change,Volume  
"AA",39.48,"6/11/2007","9:36am",-0.18,181800  
"AIG",71.38,"6/11/2007","9:36am",-0.15,195500  
"AXP",62.58,"6/11/2007","9:36am",-0.46,935000  
"BA",98.31,"6/11/2007","9:36am",+0.12,104800
```

- CSV là định dạng dữ liệu có cấu trúc.
- Trong đó: có dòng tiêu đề và các dòng dữ liệu; các giá trị phân cách bởi dấu phẩy.





## 5.2.1. Tập tin có định dạng .csv

- Đọc file .csv
  - Bước 1: Nạp thư viện csv với lệnh **import csv**
  - Bước 2: Mở file csv bằng lệnh **open('file name', 'mode')**
  - Bước 3: Đọc file với **csv.reader()**
  - Bước 4: Duyệt file với vòng **for**



## 5.2.1. Tập tin có định dạng .csv

- **Ví dụ 10: Đọc và in dữ liệu stock.csv**

```
import csv
```

```
with open('stock.csv', 'r') as f:
```

```
    file_reader = csv.reader(f)
```

```
    for read in file_reader:
```

```
        print(read)
```

```
Symbol,Price,Date,Time,Change,Volume  
"AA",39.48,"6/11/2007","9:36am",-0.18,181800  
"AIG",71.38,"6/11/2007","9:36am",-0.15,195500  
"AXP",62.58,"6/11/2007","9:36am",-0.46,935000  
"BA",98.31,"6/11/2007","9:36am",+0.12,104800
```



## 5.2.1. Tập tin có định dạng .csv

- Để truy cập 1 trường trong file, người dùng sử dụng chỉ số  $[0, 1, \dots]$  ứng với trường dữ liệu: Symbol, Price,...
- Ví dụ:

```
print(read[1]) # Hiển thị thông tin trường Price (tại cột thứ 2)
```

```
Price  
9.48,  
71.38  
62.58  
98.31
```



## 5.2.1. Tập tin có định dạng .csv

- Để hiển thị dữ liệu bỏ qua dòng tiêu đề dùng lệnh `next()`

```
import csv
```

```
with open('stock.csv', 'r') as f:
```

```
    file_reader = csv.reader(f)
```

```
    next(file_reader)
```

```
    for read in file_reader:
```

```
        print(read)
```

```
"AA",39.48,"6/11/2007","9:36am",-0.18,181800  
"AIG",71.38,"6/11/2007","9:36am",-0.15,195500  
"AXP",62.58,"6/11/2007","9:36am",-0.46,935000  
"BA",98.31,"6/11/2007","9:36am",+0.12,104800
```



## 5.2.1. Tập tin có định dạng .csv

- Ghi dữ liệu vào file csv:

```
csv.writer(tên_file.csv)
writerow(dữ liệu)
writerows(dữ liệu)
```

Trong đó:

- Lớp `csv.writer()`: trả về đối tượng `writer` giúp chuyển đổi dữ liệu người dùng vào các chuỗi được phân biệt bởi dấu phẩy.
- Đối tượng `writerow()`: Ghi từng dòng vào file (mỗi dòng là 1 list)
- Đối tượng `writerows()`: Ghi nhiều dòng dữ liệu vào file (list nhiều chiều)



## 5.2.1. Tập tin có định dạng .csv

- Ví dụ : Ghi thêm dữ liệu vào file stock.csv

```
import csv
```

```
with open('stock.csv', 'a') as f:
```

```
    writer = csv.writer(f)
```

```
    writer.writerow(["CAT",20.5,"8/11/2007","9:00am",+0.15,104900])
```

```
    writer.writerow(["PSG",100,"2/10/2007","6:00am",+5.15,105909])
```

```
csv_rowlist = [ ["HNI",2.5,"8/10/2007","8:00am",+0.18,102000],
```

```
                ["JLK",10.7,"2/12/2007","3:00am",+2.16,105409] ]
```

```
with open('stock.csv', 'w') as f:
```

```
    writer = csv.writer(f)
```

```
    writer.writerows(csv_rowlist)
```



## 5.2.1. Tập tin có định dạng .csv

- Ngoài `csv.reader()` và `writerow()` Python còn hỗ trợ lớp đọc ghi file:

```
csv.DictReader()  
csv.DictWriter(file, fieldnames)
```

Trong đó:

- Đối tượng của lớp `csv.DictReader()` cho phép đọc 1 file csv thành một từ điển, trong đó mặc định các giá trị dòng đầu trở thành khóa, các dòng còn lại là các giá trị dữ liệu.
- Đối tượng của lớp `csv.DictWriter()` cho phép ghi 1 từ điển vào file csv



## 5.2.1. Tập tin có định dạng .csv

- Ví dụ 12: Sử dụng csv.DictReader để đọc dữ liệu tập tin stock.csv

```
import csv
with open("stock.csv", 'r') as file:
    csv_file = csv.DictReader(file)
    for row in csv_file:
        print(dict(row))
```

```
{'Symbol': 'CAT', 'Price': '20.5', 'Date': '8 / 11 / 2007', 'Time': '9:00 am', 'Change': '0.15', 'Volume': '104900'}
{'Symbol': 'PSG', 'Price': '100', 'Date': '2 / 10 / 2007', 'Time': '6:00 am', 'Change': '5.15', 'Volume': '105909'}
```





## 5.2.1. Tập tin có định dạng .csv

- Ví dụ 13: Sử dụng **csv.DictWriter()** để ghi dữ liệu vào file stock\_new.csv

```
import csv
with open('d:/stock_new.csv','w') as f:
    fieldnames = ['Symbol','Price','Date','Time','Change','Volume']
    writer = csv.DictWriter(f, fieldnames=fieldnames)
    writer.writeheader()
    writer.writerow({'Symbol': "CAT", 'Price': 20.5, 'Date': "8 / 11 / 2007", 'Time': "10:00 \
        am", 'Change': +0.15, 'Volume': 104900})
    writer.writerow({'Symbol': "PSG", 'Price': 22.5, 'Date': "18 / 11 / 2007", 'Time': "9:00 \
        am", 'Change': +1.15, 'Volume': 234900})
    data_dict=({'Symbol': "ATA", 'Price': 50.5, 'Date': "20 / 11 / 2007", 'Time': "7:00 am",
        'Change': +5.34, 'Volume': 200400}, {\
        'Symbol': "TMT", 'Price': 150.5, 'Date': "28 / 11 / 2007", 'Time': "9:00 \
        am", 'Change': +3.25, 'Volume': 211400})
    writer.writerows(data_dict)
```



## 5.2.1. Tập tin có định dạng .csv

- Sử dụng thư viện Pandas.
- Ưu điểm: Pandas là một thư viện mã nguồn mở được xây dựng dựa trên NumPy, sử dụng thao tác và phân tích dữ liệu, cung cấp khả năng:
  - Khả năng import dữ liệu từ nhiều nguồn khác nhau như CSV
  - Xử lý, phân tích dữ liệu tốt như mô hình hoá và thống kê
  - Tích hợp tốt với các thư viện khác của python
  - Cung cấp hiệu suất tốt



## 5.2.1. Tập tin có định dạng .csv

- Đọc file .csv sử dụng thư viện Pandas.
- Ví dụ:

```
import pandas as pd  
df = pd.read_csv('data.csv')  
print(df)
```

```
      Duration  Pulse  Maxpulse  Calories  
0           60    110      130      409.1  
1           60    117      145      479.0  
2           60    103      135      340.0  
3           45    109      175      282.4  
4           45    117      148      406.0  
..          ...    ...      ...      ...  
164          60    105      140      290.8  
165          60    110      145      300.4  
166          60    115      145      310.2  
167          75    120      150      320.4  
168          75    125      150      330.4
```

```
[169 rows x 4 columns]
```



## 5.2.1. Tập tin có định dạng .csv

- Ghi file .csv sử dụng thư viện Pandas.
- Ví dụ:

```
import pandas as pd
# Dữ liệu 3 cột
nme = ["aparna", "pankaj", "sudhir", "Geeku"]
deg = ["MBA", "BCA", "M.Tech", "MBA"]
scr = [90, 40, 80, 98]
# tạo dữ liệu dạng từ điển
dict = {'name': nme, 'degree': deg, 'score': scr}
# tạo dataframe
df = pd.DataFrame(dict)
# lưu file định dạng .csv
df.to_csv('file1.csv')
print(df)
```



## 5.2.2. Làm việc với file hình ảnh

- **Yêu cầu:** Cài package PIL (Python Imaging Library)
  - Mở Run gõ cmd chạy lệnh **pip install pillow**
- **Mở file ảnh**
  - Bước 1: Nạp các module trong gói IPL  
**from PIL import Image**
  - Bước 2: Mở file bằng lệnh **Image.open(filename)**
- **Ví dụ:** Mở file ảnh photo.jpg trong ổ D  
**from PIL import Image**  
**Ima = Image.open('D:\photo.jpg')**



## 5.2.2. Làm việc với file hình ảnh

- **Lưu file** vào máy tính bằng phương thức:

```
save (path,type)
```

- Ví dụ: Đổi đuôi mở rộng file photo.jpg sang photo.png  
from PIL import Image  
im = Image.open('D:/photo.jpg')  
im.save('D:/photo.jpg', '.png')



## 5.2.2. Làm việc với file hình ảnh

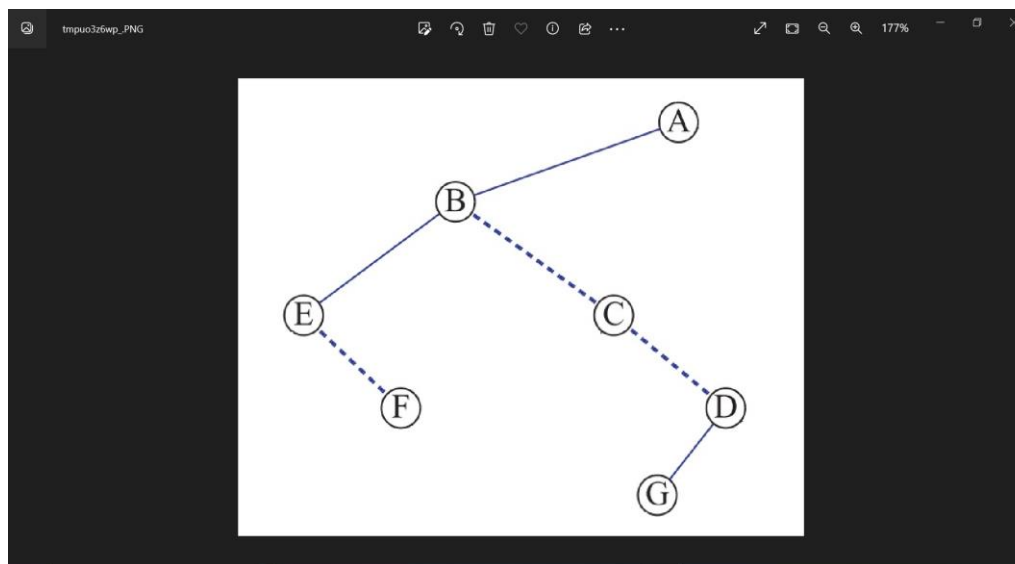
- **Hiển thị file ảnh:** bằng phương thức **show()**
- Ví dụ: Mở file ảnh photo.jpg và hiển thị lên màn hình  
from PIL import Image  
img = Image.open('D/photo.jpg')  
img.show()



## 5.2.2. Làm việc với file hình ảnh

- **Hiển thị file ảnh:** bằng phương thức **show()**

```
from PIL import Image  
img= Image.open('C:\\Users\\GIAPCN\\Pictures\\bt_gt.jpg')  
img.show()
```







## 5.2.2. Làm việc với file hình ảnh

- **Thumbnail:** là hình ảnh thu nhỏ có kích thước giảm so với ảnh gốc nên thumbnail giúp tăng tốc độ tải hình ảnh trên web và các thiết bị di động.

- **Tạo thumbnail:**

`Image.thumbnail(size, resample)`

Trong đó:

- Size: kích thước ảnh thumbnail
- Resample: lấy lại mẫu (tùy chọn): `PIL.Image.NEAREST`, `PIL.Image.BILINEAR`, `PIL.Image.BICUBIC`, `PIL.Image.LANCZOS` mặc định là `PIL.Image.BICUBIC`



## 5.2.2. Làm việc với file hình ảnh

- Ví dụ về thumbnail:

```
from PIL import Image
image = Image.open('images/cat.jpg')
image.thumbnail((90,90))
image.save('images/thumbnail.jpg')
image1 = Image.open('images/thumbnail.jpg')
image1.show()
```



## 5.2.2. Làm việc với file hình ảnh

- **Cắt ảnh:** sử dụng hàm crop(kích thước)
- Ví dụ:

```
from PIL import Image  
img = Image.open('D:/photo.jpg')  
box = (100, 100, 400, 400)  
region = img.crop(box)  
region.show()  
region.save('D:/photo_crop.jpg')
```



## 5.2.2. Làm việc với file hình ảnh

- **Biến đổi màu:** Python có 2 chế độ màu thông dụng
  - Màu đen trắng (L)
  - Màu RGB
- **Hàm biến đổi màu: `convert()`**
- **Ví dụ:**

```
from PIL import Image  
img = Image.open('D:/photo.jpg')  
converted_img = img.convert("L")  
converted_img.show()
```



## 5.2.2. Làm việc với file hình ảnh

- Sử dụng thư viện: opencv
- Ưu điểm: opencv hỗ trợ nhiều phép toán xử lý ảnh.
- Cài đặt
  - `pip install opencv-python`
- Webside: <https://pypi.org/project/opencv-python/>



## 5.2.2. Làm việc với file hình ảnh

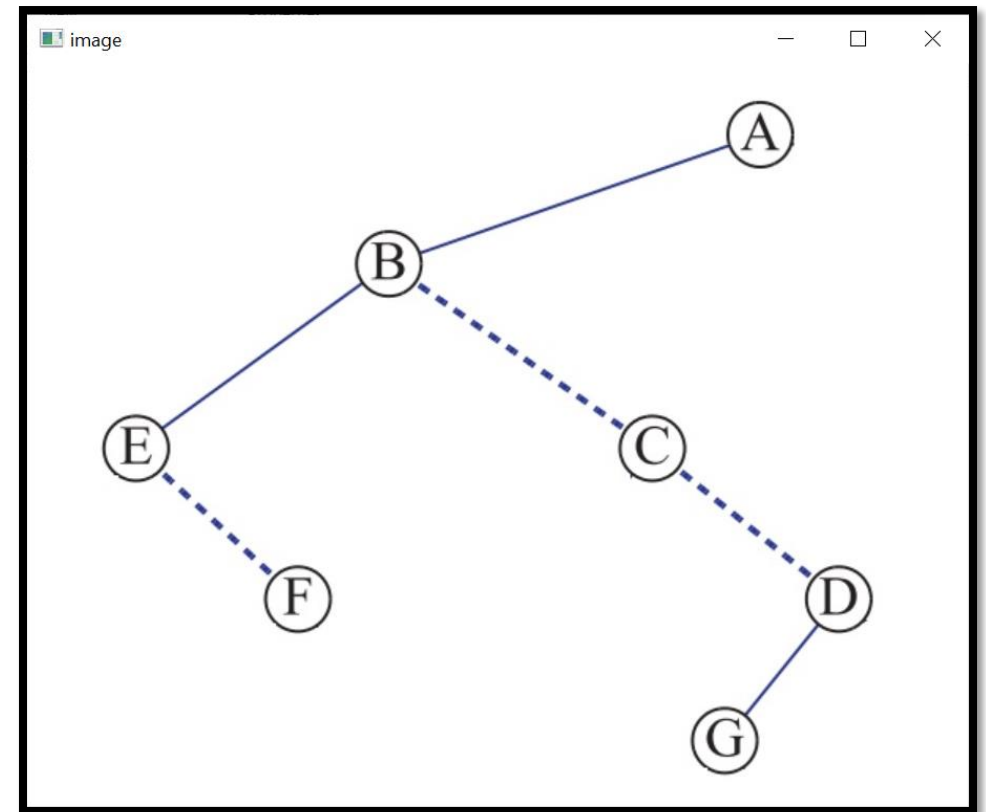
- Một số hàm thường dùng:
  - Đọc ghi: `imread`, `imwrite`
  - Hiển thị: `imshow`
  - Xử lý ảnh: `rotate`, `warpAffine`, `undistort`, ....



## 5.2.2. Làm việc với file hình ảnh

- Ví dụ: Mở và hiện thị file ảnh

```
# Mở file
import cv2
img =
cv2.imread("C:\\Users\\GIAPCN\\Pictures\\b
t_gt.jpg", cv2.IMREAD_COLOR)
cv2.imshow("image", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```





## 5.2.2. Làm việc với file hình ảnh

- Ví dụ: xoay ảnh

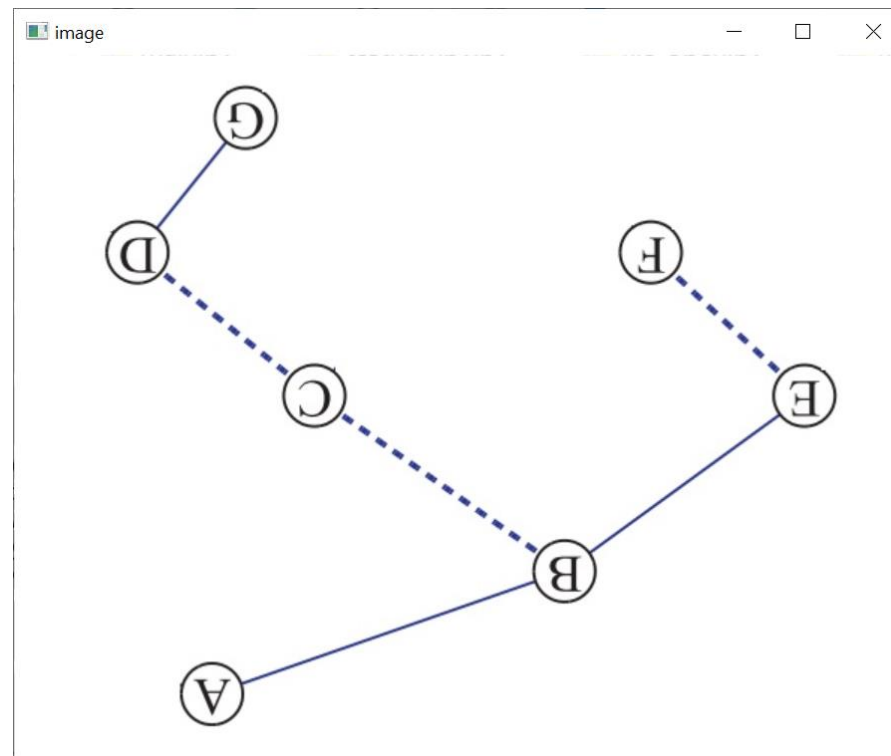
*# Mở file*

```
import cv2  
img_org =  
cv2.imread("C:\\Users\\GIAPCN\\Pictures\\bt_g  
t.jpg", cv2.IMREAD_COLOR)
```

*# xoay ảnh*

```
image = cv2.rotate(img_org, cv2.ROTATE_180)
```

```
cv2.imshow("image", image)  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```







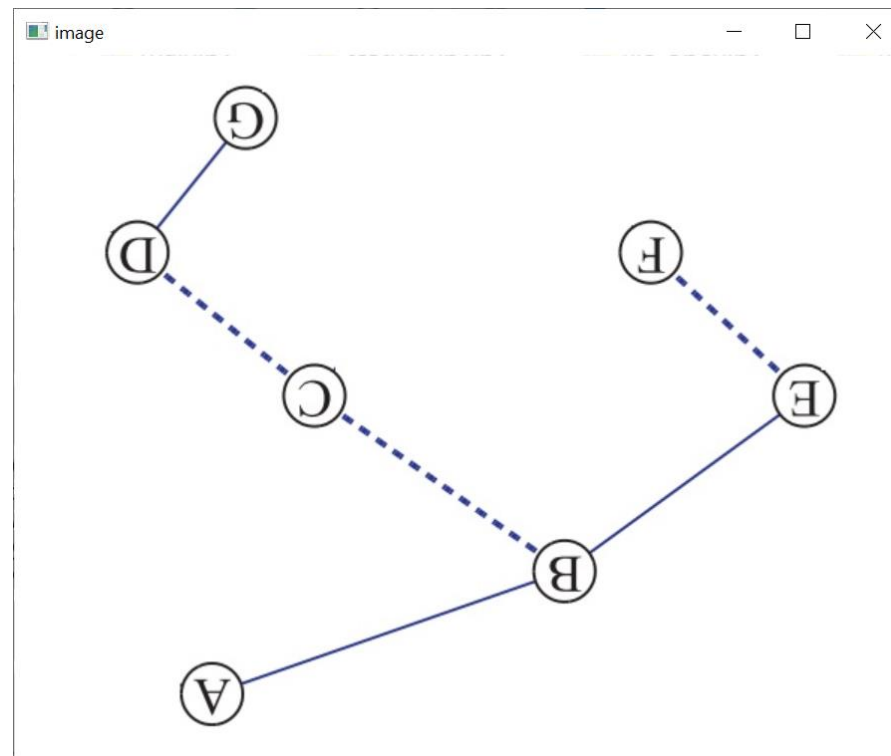
## 5.2.2. Làm việc với file hình ảnh

- Ví dụ: lưu ảnh

```
import cv2
img_org =
cv2.imread("C:\\Users\\GIAPCN\\Pictures\\bt_gt
.jpg", cv2.IMREAD_COLOR)
image = cv2.rotate(img_org, cv2.ROTATE_180)
cv2.imshow("image", image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

*# save file*

```
filename = 'savedImage.jpg'
cv2.imwrite(filename, image)
```





# Tổng kết CHƯƠNG 5

## LÀM VIỆC VỚI TẬP TIN:

5.1 Làm việc với tệp tin văn bản

5.2 Làm việc với các kiểu tệp tin khác

5.2.1. Tệp tin có định dạng .csv

5.2.2. Làm việc với file hình ảnh



**TRƯỜNG ĐẠI HỌC THƯƠNG MẠI**  
**THUONGMAI UNIVERSITY**

**CHÚC CÁC EM HỌC TỐT!**