



Bài giảng môn học:

**Khoa Học Dữ Liệu (7080509)**

# CHƯƠNG 3

## Lập trình Python căn bản

### (Phần 03)

# Nội dung chương 3

---



**3.1 Biến, toán tử trong Python**

**3.2 Các kiểu dữ liệu cơ bản**

**3.3 Cấu trúc điều khiển, vòng lặp**

**3.4 Hàm, lớp, đối tượng trong Python**

**3.5 Làm việc với tập tin**

**3.6 Bài tập chương 3**

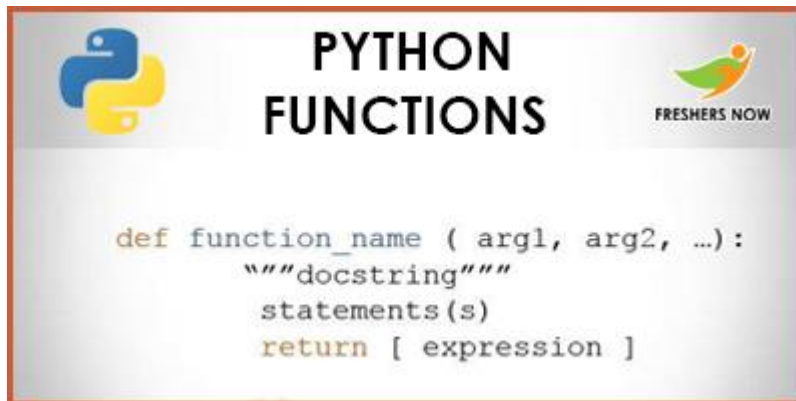


# HÀM TRONG PYTHON



# Cơ bản về hàm trong Python

- Hàm trong Python là một nhóm các câu lệnh trong chương trình được tổ chức chung với nhau để thực hiện một chức năng hay một nhiệm vụ cụ thể nào đó.
- Sử dụng hàm giúp phân rã chương trình từ một chương trình lớn, phức tạp thành các phần cụ thể nhỏ hơn giúp dễ quản lý, tổ chức, nâng cao khả năng tái sử dụng và chia sẻ công việc.



```
1  #xây dựng hàm trong python  
2  def hello_MDC(str):  
3      #Hiển thị câu chào  
4      print('Hi ', str, ', How are you?')  
5      print('Have a nice day!')
```

```
1  #xử dụng hàm đã xây dựng  
2  hello_MDC('Tùng Dương')
```

Hi Tùng Dương , How are you?  
Have a nice day!

**EXAMPLE**

# Cơ bản về hàm trong Python



## 1. Cú pháp:


```
def tên_hàm(các_tham_số):  
    "function_docstring"  
    Các câu lệnh xử lý bên trong hàm  
    return [kết quả trả về]
```

- Từ khóa **def** được sử dụng để bắt đầu phần định nghĩa hàm.
- sau đó là **tên\_hàm**, tên hàm được đặt theo quy tắc như tên biến.
- Các tham số được truyền vào bên trong các dấu ngoặc đơn.
- Ở cuối là dấu hai chấm “:”.
- Sau đó là lệnh để được thực thi.
- Kết quả trả về cho hàm được thực hiện thông qua lệnh return

**Lưu ý: Hàm không bắt buộc phải có tham số truyền vào hay kết quả trả về**

# Cơ bản về hàm trong Python

- Trước khi bắt tay vào xây dựng 1 hàm trong Python, cần phải tự **trả lời các câu hỏi**:
  - Hàm này sử dụng nhằm mục đích gì?
  - Hàm này nhận đầu vào là gì?
  - Hàm này trả kết quả ra là gì?



```
1  #Xây dựng hàm tính n!
2  #1)Hàm này dùng để làm gì? - Để tính n!
3  #2)Hàm này nhận dữ liệu vào là gì? - Một số nguyên dương N
4  #3)Hàm trả kết quả là gì? - Một số nguyên dương là tích của 1*2*...*N
5  def giai_thua(n):
6      #Nhóm câu lệnh xử lý bên trong hàm
7      tích=1
8      for i in range(1,n+1):
9          tích=tích*i
10     #kết quả trả về cho hàm
11     return tích
```

```
1  n = int(input('Nhập vào một số nguyên N:'))
2  print(n,'!=',giai_thua(n))
```

Nhập vào một số nguyên N:10  
10 != 3628800

# Cơ bản về hàm trong Python



## Gọi hàm:

- Hàm sau khi được xây dựng, có thể thực hiện lời gọi hàm ở nơi nào cần dùng đến.

```
1 n = int(input('Nhập vào một số nguyên N:'))
2 print(n, '!=', giai_thua(n))
```

Nhập vào một số nguyên N:10  
10 != 3628800



```
1 #Gọi hàm giai_thua đã xây dựng
2 #Tính 12!
3 print('12! = ', giai_thua(12))
```

12! = 479001600

- Các lệnh mà chúng ta đã được học và sử dụng trước đây như: `print()`, `input()`, `type()`, `int()`, `float()`, `str()`... Đây thực chất là các **hàm được Python định nghĩa sẵn**.

# Cơ bản về hàm trong Python



## Lệnh return:

- Lệnh **return <kết quả trả về>** được sử dụng để trả kết quả xử lý thông qua tên hàm.
- Lệnh return có thể có hoặc không.
- Trong trường hợp **không cung cấp <kết quả trả về>**, thì hàm **return này sẽ trả về None**. Nói cách khác, lệnh return được sử dụng để thoát khỏi định nghĩa hàm.

```
1  #xây dựng hàm trong python
2  def hello_MDC(str):
3      #Hiển thị câu chào
4      print('Hi ', str, ', How are you?')
5      print('Have a nice day!')
```

**EXAMPLE**

```
1  #Xây dựng hàm tính n!
2  def giai_thua(n):
3      tích=1
4      for i in range(1,n+1):
5          tích=tích*i
6      #kết quả trả về cho hàm
7      return tích
```



# Cơ bản về hàm trong Python



Lệnh `return()` có thể trả về 1 hay nhiều kết quả, nếu có nhiều hơn một kết quả thì ngăn cách nhau bởi **dấu phẩy**.

```
1  #Hàm tính tổng, hiệu, tích và thương
2  def all_ab(a,b):
3      tong=a+b
4      hieu=a-b
5      tich=a*b
6      thuong=a/b
7      #hàm trả về
8      return tong
```

```
1  a=10
2  b=6
3  #Lấy kết quả trả về khi thực hiện hàm
4  tong,hieu,tich,thuong = all_ab(a,b)
5
6  #Lưu ý: Thứ tự trả về theo đúng thứ tự đã viết trong
7  #câu lệnh return
8  print('Tổng ',a,'+',b,'=',tong)
9  print('Hiệu ',a,'-',b,'=',hieu)
10 print('Tích ',a,'*',b,'=',tich)
11 print('Thương ',a,'/',b,'=',thuong)
```

```
Tổng   10 + 6 = 16
Hiệu   10 - 6 = 4
Tích   10 * 6 = 60
Thương 10 / 6 = 1.6666666666666667
```

**EXAMPLE**

# Cơ bản về hàm trong Python



## Tham số truyền vào hàm:

- Tham số bắt buộc
- Tham số có mặc định (Default parameter)
- Tham số có độ dài biến (Variable-Length Parameter)

### ➤ Tham số bắt buộc

```
1 #Xây dựng hàm
2 #Hàm giai_thua
3 def giai_thua
4     tích=1
5     for i in
6         tích=
7     #kết quả
8     return ti
```

```
1 #Gọi hàm giai_thua đã xây dựng
2 print('12! = ', giai_thua(12))
```

12! = 479001600

```
1 #Gọi hàm giai_thua đã xây dựng
2 #Khi không truyền vào tham số
3 print('12! = ', giai_thua())
```

**EXAMPLE**

TypeError

Traceback (most recent call last)

<ipython-input-14-01226097b9b9> in <module>

```
1 #Gọi hàm giai_thua đã xây dựng
```

```
2 #Khi không truyền vào tham số
```

```
----> 3 print('12! = ', giai_thua())
```

TypeError: giai\_thua() missing 1 required positional argument: 'n'

# Cơ bản về hàm trong Python



## ➤ Tham số mặc định cho hàm:

- Để hạn chế trường hợp báo lỗi khi gọi hàm không cung cấp tham số thì trong Python cũng cung cấp cho chúng ta **thiết lập giá trị mặc định của tham số** khi khai báo hàm. Bằng cách sử dụng dấu **=** với cú pháp như sau:

```
def ten_ham(param = defaultValue):  
    # code
```

- Trong đó: **defaultValue** là giá trị mặc định của tham số đó mà bạn muốn gán.

# Cơ bản về hàm trong Python



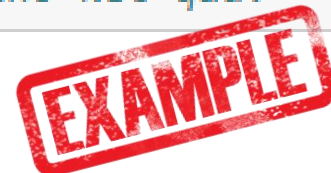
## ➤ Ví dụ:

- Hàm **sum()** gọi khi không truyền giá trị và có truyền giá trị

```
1  # Phan dinh nghia ham o day
2  def sum( arg1=5, arg2=7 ):
3      # Cong hai tham so va tra ve ket qua."
4      total = arg1 + arg2
5      return total;
6
7  # Goi ham sum() co đủ 2 tham số truyền vào
8  print(sum( 10, 20 ))
9
10 # Goi ham sum() không có tham số truyền vào
11 print(sum())
12
13 #Khi code trên được thực thi sẽ cho kết quả:
```

30

12



# Cơ bản về hàm trong Python



## ➤ Tham số có độ dài biến (Variable-Length Parameter)

- Trên thực tế, không phải lúc nào chúng ta cũng biết được chính xác số lượng biến truyền vào trong hàm. Chính vì thế trong Python có cũng cấp cho chúng ta khai báo một **param** đại diện cho các biến truyền vào hàm bằng cách thêm dấu **\*** vào trước param đó.

### ▪ Ví dụ:

```
1  def get_sum(*num):
2      tmp = 0
3      # duyệt các tham số
4      for i in num:
5          tmp += i
6      return tmp
7
8  result = get_sum(1, 2, 3, 4, 5)
9  print(result)
10
11 #Khi code trên được thực thi sẽ cho kết quả:
```

**EXAMPLE**

# Cơ bản về hàm trong Python



## Phạm vi của biến trong hàm.

- Khi một biến được khai báo ở trong hàm thì nó chỉ có thể được sử dụng ở trong hàm đó thôi.

### ➤ Ví dụ:

```
1 def say_hello():
2     a = "Hello"
3     print(a)
4
5 print(a)
6 # Lỗi: name 'a' is not defined
```

-----  
NameError

Traceback (most recent call last)

<ipython-input-10-b25341c3aba9> in <module>

3 print(a)

4

----> 5 print(a)

6 # Lỗi: name 'a' is not defined

NameError: name 'a' is not defined



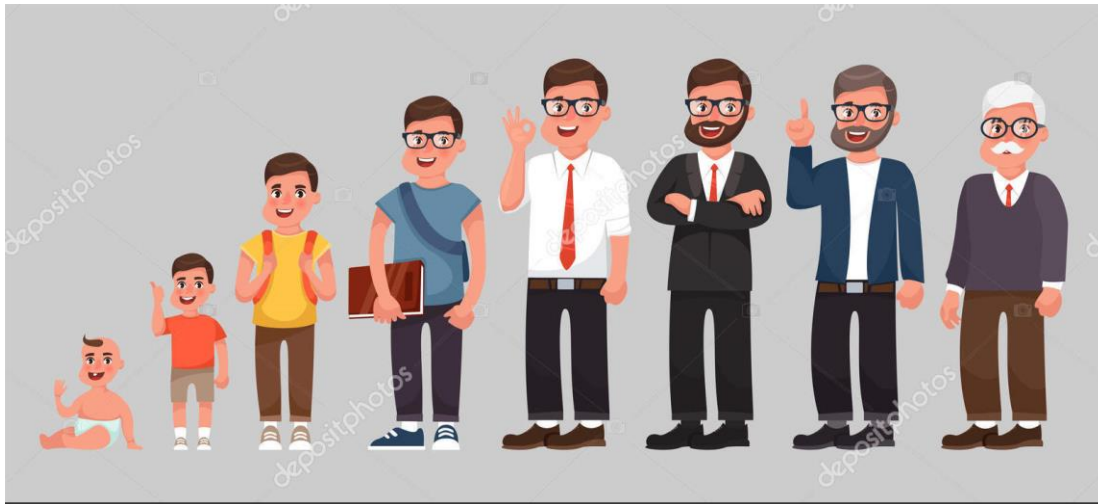


# Thực hành

# Bài 15\_a: Viết hàm cho các bài đã thực hiện



- 1) Viết hàm **greeting()**: Trả về câu chào với tham số truyền vào là chuỗi họ tên và năm sinh (**Xem lại bài tập số 2**)
- 2) Viết hàm **rabbit\_count()**: tính số thỏ trong rừng khi truyền vào số tháng (**Xem lại bài tập số 3**)
- 3) Viết hàm **count\_mark()**: trả về số sinh viên học lại và tổng số sinh viên trong lớp với tham số truyền vào là một danh sách bảng điểm (**Xem lại bài tập số 5 ý 1, 2**)

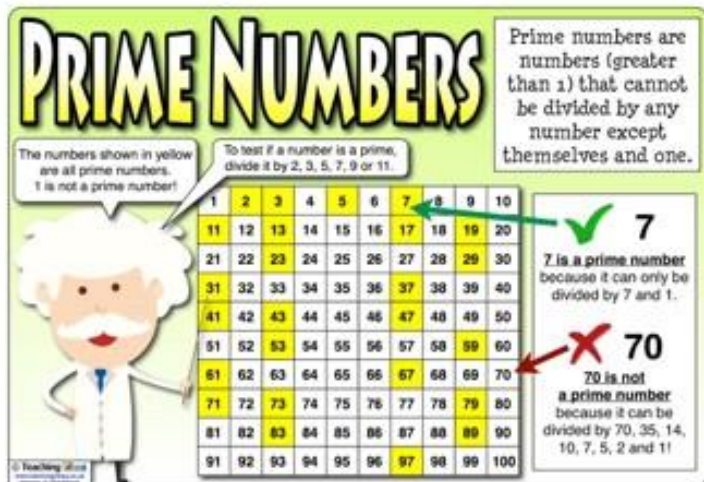




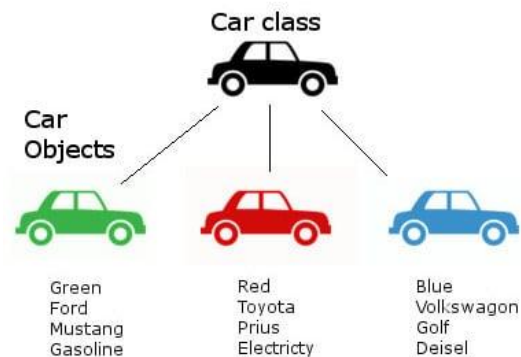
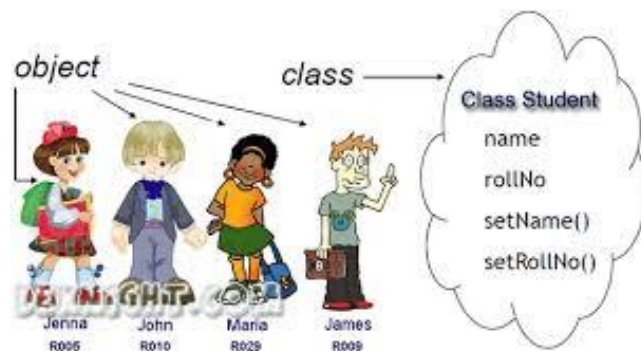
# Bài 15\_b: Viết hàm cho các bài đã thực hiện



- 4) Viết hàm **bmi\_show()**: Trả về nhận xét dựa vào chỉ số BMI đã tính với 2 tham số truyền vào là chiều cao, cân nặng (**Xem lại bài tập số 7**)
- 5) Viết hàm **cal\_point()**: Trả về điểm trung bình hệ 10 và hệ 4 của một học sinh khi truyền vào danh sách điểm (**Xem lại bài tập số 10 ý 2**)
- 6) Viết hàm **list\_prime()**: trả danh sách các số nguyên tố trong khoảng từ 1 đến n với tham số truyền vào là n (**Xem lại bài tập số 12**)



# LỚP, ĐỐI TƯỢNG TRONG PYTHON



# Lớp, đối tượng trong Python



## 1. Giới thiệu và một số khái niệm

- Python là một ngôn ngữ lập trình hướng đối tượng

### ➤ Một số khái niệm hướng đối tượng

- Lớp:** được định nghĩa bởi người dùng cho một đối tượng gồm một tập hợp các thuộc tính mà xác định rõ bất kỳ đối tượng nào của lớp đó. Các thuộc tính là các thành viên dữ liệu (các biến class và biến instance) và các phương thức được truy cập thông qua toán tử dot (dấu chấm .).
- Biến lớp - class:** Đây là một biến được chia sẻ bởi tất cả các instance (sự thể hiện) của một lớp. Các biến class được định nghĩa bên trong một lớp nhưng ở bên ngoài bất cứ phương thức nào của lớp đó. Biến class không được sử dụng thường xuyên như biến instance.
- Thành viên dữ liệu:** Là một biến class hoặc biến instance mà giữ dữ liệu được liên kết với một lớp và các đối tượng của nó.

# Lớp, đối tượng trong Python

## ➤ Một số khái niệm hướng đối tượng

- **Trình khởi tạo:** Là trình tạo một sự thể hiện của một lớp.
- **Phương thức:** Một loại hàm đặc biệt mà được định nghĩa trong một phần định nghĩa lớp.

## ➤ Khai báo Class trong Python

```
class ClassName:  
    'Gồm các thuộc tính, phương thức'  
    # Code ...
```

- Trong đó, **className** là tên của class cần khai báo.

# Lớp, đối tượng trong Python



## ➤ Ví dụ 1:

```
class Person:
    # thuộc tính
    name = "Vũ Thanh Tài";
    age = 22;
    male = "Nam"
    # phương thức
    def setName(self, name):
        self.name = name

    def getName(self):
        return self.name

    def setAge(self, age):
        self.age = age

    def getAge(self):
        return self.age

    def setMale(self, male):
        self.male = male

    def getMale(self):
        return self.male
```

**EXAMPLE**

# Lớp, đối tượng trong Python



## ➤ Ví dụ 2:

```
1 class Sinhvien:
2     'Class co so chung cho tat ca sinh vien'
3     svCount = 0
4
5     def __init__(self, ten, hocphi):
6         self.ten = ten
7         self.hocphi = hocphi
8         Sinhvien.svCount += 1
9
10    def displayCount(self):
11        print ("Tong so Sinh vien %d" % Sinhvien.svCount)
12
13    def displaySinhvien(self):
14        print ("Ten : ", self.ten, ", Hoc phi: ", self.hocphi)
```



# Lớp, đối tượng trong Python



## ➤ Khởi tạo class.

Sau khi đã khai báo được class trong Python rồi, thì để khởi tạo nó sử dụng cú pháp sau:

```
variableName = className()
```

Trong đó:

- **variableName** là biến thể hiện lại đối tượng.
- **className** là class muốn khởi tạo.

## ➤ Ví dụ: Khởi tạo class person ở trên.

```
# instance  
person = Person()
```

# Lớp, đối tượng trong Python



- Sau khi đã khởi tạo được class rồi thì biến sẽ có thể truy cập được các phần tử được cho phép trong class đó.
- Bằng cách sử dụng dấu `.` theo cú pháp sau:

```
# truy cập den thuoc tinh
object.propertyName

#truy cap den phuong thuc
object.methodName()
```

Trong đó:

- `object` là biến thể hiện lại object.
- `propertyName` là tên thuộc tính muốn truy xuất.
- `methodName` là tên phương thức muốn truy xuất.



# Lớp, đối tượng trong Python



- **Ví dụ:** sẽ truy xuất đến các thuộc tính và phương thức trong class Person

```
# properties
print(person.name) # Vũ Thanh Tài
print(person.age) # 22
print(person.male) # True
# methods
person.setName("Nguyễn Thị HH")
print(person.getName()) # Nguyễn Thị HH

person.setAge(22)
print(person.getAge()) # 22

person.setMale("Nữ")
print(person.getMale()) # False
```





Thực hành

# Bài 16: Xây dựng lớp phân số



**Xây dựng lớp phân số gồm các thành phần:**

- **Thuộc tính:** Tử số, Mẫu số
- **Phương thức:** Nhập phân số, in phân số, tính tổng 2 phân số.

**Yêu cầu: Nhập 2 phân số, tính tổng và in ra kết quả**

	F	G	H	I	J	K
3		<b>Hàm LCM trong Excel</b>				
4		<u>Hàm lượng giác và toán học trong Excel</u>				
5						
6		<b>Tính tổng các phân số</b>				
7			Phân số 1	Phân số 2	Kết quả	
8		Tử số	25	96	?	
9		Mẫu số	19	17		
10						
11						

Nha\_SX Sheet2 T ... Excel.net.vn

# Làm việc với tập tin trong Python

Excel Data (Print) - Notepad

1400-57-0201AB01	w7	v5/v14	28	15-03-2010	P11
1400-57-0202AB01	w4	v33/v1	28	13-03-2010	P11
1400-57-0202AB01	w7	v33/v1	28	13-03-2010	P11
1400-57-0301AB01	w8	v1/v33	28	16-03-2010	P11
1400-57-0402AH05	w5	v17	3	04-01-2010	KCS
1400-57-0509JA01	w11	v7	12	29-12-2009	KCS
1400-57-0517AL01	w15	v23	6	15-01-2010	KCS
1400-57-0518JA01	w4	v21	12	04-01-2010	KCS
1400-57-0610AL01	w11	v24	10	08-01-2010	KCS
1400-57-0610AL01	w4	v23	10	11-01-2010	KCS
1400-57-0619AH01	w11	v5	16	30-12-2009	KCS
1400-57-0623JA01	w2	v6	10	05-01-2010	KCS
1400-57-0628JA01	w2	v9	16	08-01-2010	KCS
1400-57-0901AM03	w11	v13	8		
1400-57-0902AM03	w13	v5	12		



	A	B	C	D	E
1	id	title	timing	genre	rating
2		1 Dog with a	17:30-18:0	Comedy	4.7
3		2 Liv and Ma	18:00-18:3	Comedy	6.3
4		3 Girl Meets	18:30-19:0	Comedy	7.2
5		4 KC Underc	19:00-19:3	Comedy	6.1
6		5 Austin and	19:30-20:0	Comedy	6
7					

# Làm việc với tập tin (File)



## 1. File là gì?

- File hay còn gọi là tệp, tập tin. File là tập hợp của các thông tin được đặt tên và lưu trữ trên bộ nhớ máy tính như đĩa cứng, đĩa mềm, CD, DVD,...
- Khi muốn **đọc** hoặc **ghi file**, chúng ta cần phải **mở file** trước. **Khi hoàn thành**, file cần **phải được đóng** lại để các tài nguyên được gắn với file được giải phóng.
- Do đó, trong Python, một thao tác với file diễn ra theo thứ tự sau:
  - ✓ **Mở tệp tin**
  - ✓ **Đọc hoặc ghi**
  - ✓ **Đóng tệp**

# Làm việc với tập tin (File)



## 2. Mở File

Để mở file trong Python chúng ta sử dụng hàm **open** với cú pháp như sau:

```
open(filePath, mode, buffer)
```

Trong đó:

- **filePath** là đường dẫn đến địa chỉ của file.
- **mode** là thông số thiết lập chế độ chúng ta mở file được cấp những quyền gì? Mặc định mode sẽ bằng **r** (xem các mode ở dưới).
- **buffer** là thông số đệm cho file mặc định thì nó sẽ là 0.

# Làm việc với tập tin (File)



## 2. Mở File

### Các chế độ

Mode	Chú thích
r	Chế độ chỉ được phép đọc.
rb	Chế độ chỉ được phép đọc nhưng cho định dạng nhị phân.
r+	Chế độ này cho phép đọc và ghi file, con trỏ nó sẽ nằm ở đầu file.
rb+	Chế độ này cho phép đọc và ghi file ở dạng nhị phân, con trỏ sẽ nằm ở đầu file.
w	Chế độ ghi file, nếu như file không tồn tại thì nó sẽ tạo mới file và ghi nội dung, còn nếu như file đã tồn tại nó sẽ ghi đè nội dung lên file cũ.
wb	Tương tự chế độ w nhưng đối với nhị phân.
w+	Mở file trong chế độ đọc và ghi. còn lại như w.
wb+	Giống chế độ w+ nhưng đối với nhị phân
a	Mở file trong chế độ ghi tiếp. Nếu file đã tồn tại rồi thì nó sẽ ghi tiếp nội dung, và nếu như file chưa tồn tại thì nó sẽ tạo một file mới và ghi nội dung vào đó.
ab	Tương tự a nhưng đối với nhị phân.
a+	Mở file trong chế độ đọc và ghi tiếp nội dung, còn lại cơ chế giống chế độ a.
ab+	Tương tự chế độ a+ nhưng đối với nhị phân.

# Làm việc với tập tin (File)

## 2. Mở File

➤ Ví dụ:

```
f=open("test.txt") #mở file mode 'r' hoặc 'rt' để đọc
```

```
f=open("test.txt",'w') #mở file mode 'w' để ghi
```

```
import csv  
f=open('data.csv','rt') #mở file mode 'r' hoặc 'rt' để đọc file csv
```

## 3. Đóng File

- Việc đóng file được xây dựng trong Python bằng hàm `close()` với cú pháp như sau:

```
fileObject.close()
```

- Trong đó, `fileObject` là đối tượng mà chúng ta thu được khi sử dụng hàm `open()`.



# Làm việc với tập tin (File)



## 4. Đọc file

- Sau khi đã mở được file ra rồi, để đọc được file thì chúng ta sử dụng phương thức **read** với cú pháp:

```
fileObject.read(length);
```

Trong đó:

- **fileObject** là đối tượng mà chúng ta thu được khi sử dụng hàm **open()**.
- **length** là dung lượng của dữ liệu mà chúng ta muốn đọc, nếu để trống tham số này thì nó sẽ đọc hết file hoặc nếu file lớn quá thì nó sẽ đọc đến khi giới hạn của bộ nhớ cho phép.

# Làm việc với tập tin (File)

## 4. Đọc file

- Ví dụ: đọc và xuất ra dữ liệu đọc được trong file readme ở trên

```
# Mở file để đọc dữ liệu
fo = open("E:\Python\File\data.txt", "r+")
# Đọc một chuỗi trong file
str = fo.read(20)
print("Chuỗi được đọc là: ", str)
str = fo.read()
# In ra chuỗi được đọc
print("Chuỗi được đọc là: ", str)
# Đóng file lại
fo.close()
```

# Làm việc với tập tin (File)



## 5. Ghi file

- Để ghi được file thì bạn phải chắc chắn là đang mở file ở các chế độ cho phép ghi. Và sử dụng phương thức **write** với cú pháp sau:

```
fileObject.write(data)
```

Trong đó:

- **fileObject** là đối tượng mà chúng ta thu được khi sử dụng hàm **open()**.
- **data** là dữ liệu mà chúng ta muốn ghi vào trong file.

➤ Ví dụ:

```
# Mở file để đọc dữ liệu  
fo = open('E:/Python/File/readme.txt', 'w')  
#Ghi file  
fo.write('Ta Quang Chieu - khoa CNTT - HUMG')  
# Đóng file lại  
fo.close()
```

# Làm việc với tập tin (File)

## 5. Ghi file

### ➤ Các thuộc tính trong file.

Thuộc tính	Chú thích
<code>file.name</code>	Trả về tên của file đang được mở.
<code>file.mode</code>	Trả về chế độ mode của file đang được mở.
<code>file.closed</code>	Trả về true nếu file đã được đóng, và false nếu file chưa đóng.

### ➤ Ví dụ: In ra thông số của file readme.md ở trên

```
1 # Mở file để đọc dữ liệu
2 fo = open('E:/Python/File/readme.txt', 'w')
3 #Trả về tên của file đang mở
4 print(fo.name)
5 #Trả về chế độ mode của file đang được mở.
6 print(fo.mode)
7 #Trả về true nếu file đã được đóng, và false nếu file chưa đóng.
8 print(fo.closed)
9 fo.close()
```

```
E:/Python/File/readme.txt
w
False
```

# Ví dụ với đọc/ghi tệp tin

## Bài 1: Ghi dữ liệu vào File “data.txt”

```
# Mở file để ghi
fo = open("data.txt", "w")
# Ghi dữ liệu lên file
fo.write("Tobe or not tobe. \n Nghi lon de thanh cong ! \n");
# Close opened file
fo.close()
print("Ghi file thanh cong !")
```

## Bài 2: Đọc và ghi dữ liệu từ một File

```
obj=open("test.txt","w")
obj.write("Chao mung cac ban den voi khoa CNTT")
obj.close()
obj1=open("test.txt","r")
s=obj1.read()
print (s)
obj1.close()
obj2=open("test.txt","r")
s1=obj2.read(20)
print (s1)
obj2.close()
```



# Thực hành

# Bài 17: Đọc/Ghi file



## Đọc dữ liệu trong file dayso1\_bai17.txt:

- Tìm phần tử lớn nhất và nhỏ nhất trong dãy, sau đó thực hiện đổi chỗ phần tử lớn nhất xuất hiện đầu tiên trong dãy cho phần tử nhỏ nhất xuất hiện đầu tiên trong dãy. Lưu dãy mới đã đổi chỗ sang file dayso2\_bai17.txt

```
dayso1_bai17 - Notepad
File Edit Format View Help
5 9 8 14 6 18 21 2 19 26 -1 17 8 25 0 14 -1 1 30 11 5 30 -1
```

```
dayso2_bai17 - Notepad
File Edit Format View Help
5 9 8 14 6 18 21 2 19 26 30 17 8 25 0 14 -1 1 -1 11 5 30 -1
```