Bài 03:
Lập trìnhPython cơ bản (tiếp)

Nội dung bài 3

1. Hàm trong Python
    a. Giới thiệu Hàm
    b. Xây dựng hàm trong Python
    c. Return
    d. Tham số của hàm
    e. Phạm vi của biến
    f. Hàm ẩn danh (Lambda)
2. Lớp và đối tượng trong Python
3. Module và Package
4. Đọc và ghi file với Python

1. Hàm trong Python

Python Programming

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.


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

```python
#xử dụng hàm đã xây dựng
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?**

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

```python
n = int(input('Nhập vào một số nguyên N:'))
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 sẽtrả 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!')
```


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

8

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 ngăncách nhau bởi **dấu phẩy**.

```python
#Hàm tính tổng, hiệu, tích, thương:
def all_ab(a,b):
    add = a+b
    sub = a-b
    multi = a*b
    div = a/b
    #hàm trả về kết quả là 4 giá trị
    return add, sub, multi, div
```


```python
a=10
b=6
#Lấy kết quả trả về khi thực hiện hàm
tong,hieu,tich,thuong = all_ab(a,b)

#Lưu ý: Thứ tự trả về theo đúng thứ tự đã viết trong
#câu lệnh return
print('Tổng ',a,'+',b,'=',tong)
print('Hiệu ',a,'-',b,'=',hieu)
print('Tich ',a,'*',b,'=',tich)
print('Thương ',a,'/',b,'=',thuong)
```

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

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 tính n!
2  #Hàm giai_thua có 1 tham số bắt buộc n
3  def giai_thua(n):
4      tich=1
5      for i in range(1,n+1):
6          tich=tich*i
7      #kết quả trả về cho hàm
8      return tich
```


```
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())
```

```
---------------------------------------------------------------------------
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 khi gọi hàm không cung cấp tham số thì trong Python cũng cung cấp cho chúng ta **thiết lập giá trị mặc định của thamsố** 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_ab(a,b)** gọi khi truyền tham số và và khi không truyền tham số:

```
1  #Hàm tính tổng
2  def sum_ab(a=5, b =7):
3      total = a + b
4      return total
```

```
1  #Gọi hàm sum_ab() truyền vào 2 tham số
2  print(sum_ab(8,13))
```

```
21
```

```
1  #Gọi hàm sum_ab() không truyền vào tham số
2  #Sử dụng tham số mặc định
3  print(sum_ab())
```

```
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 mộtparam đại diện cho các biến truyền vào hàm bằng cách thêm dấu * vào trước param đó.

- Ví dụ:

```
1  #Xây dựng hàm tính tổng các số đưa vào
2  def get_sum(*num):
3      tmp=0
4      #duyệt các tham số
5      for i in num:
6          tmp = tmp+i
7      return tmp
8
9  #Gọi hàm và truyền các tham số cho hàm
10 result = get_sum(1,2,3,4,5)
11 print('Kết quả:', result)
```

```
Kết quả: 15
```

Cơ bản về hàm trong Python

Phạm vi của biến.

- Một biến chỉ có tác dụng trong phạm vi mà nó khai báo (global – local).
- Khi một biến được khai báo ở trong hàm thì nó chỉ có thể được sử dụng ở trong hàmđó.

```
1  x = 300 #Biến toàn cục, có tác dụng trong toàn bộ chương trình
2  y = 800
3  def myfunc():
4      #Biến địa phương, chỉ có tác dụng trong thân hàm
5      x = 200
6      total = x + y
7      print('(Local) x :', x)
8      print('total :', total)
9  #Gọi hàm
10 myfunc()
11
12 print('----------------------')
13 print('(global) x:', x)
14
```

```
(Local) x : 200
total : 1000
----------------------
(global) x: 300
```

```
1  #Thiết lập một biến local thành global
2  def myfunc():
3      global k #Thiết lập biến k là biến global
4      k = 300
5      print('Inside func: k = ',k)
6
7  myfunc()
8
9  print('Outside func: k =', k)
```

```
Inside func: k =  300
Outside func: k = 300
```

Hàm ẩn danh - lambda

- Một hàm ẩn danh là một hàm được định nghĩa mà không có tên
- Các hàm bình thường được định nghĩa bằng từ khóa def; hàm ẩn danh được định nghĩa bằng từ khóa lambda

Cú pháp:

```
lambda arguments :  Expression
```

- Các hàm lambda có thể có bất kỳ đối số nào nhưng chỉ có một biểu thức.

```
#Ví dụ hàm ẩn danh: 1 tham số
x = lambda a : a + 10
#lambda a: a + 10 là hàm lambda
#Trong đó: a là tham số truyền vào
#          a+10 là biểu thức

print(x(5))
print(x(7))
```

```
15
17
```

```
#Hàm ẩn danh 2 tham số
x = lambda a, b : a * b

print(x(5, 6))
```

```
30
```

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)**

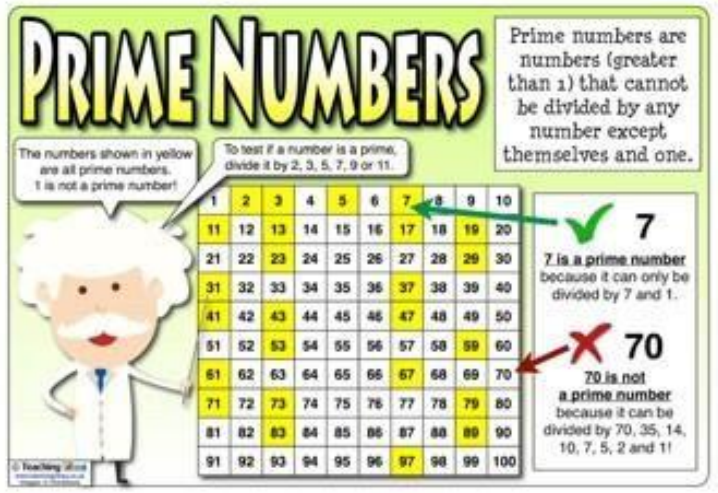
3) Viết hàm **count_mark():** trả về số sinh viên học lại và tổng số sinh viên trong lớp với tham số truyền vào là một danh sách bảng điểm **(Xem lại bài tập số 5 ý 1, 2)**

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 tử 1 đến n với tham số truyền vào là n **(Xem lại bài tập số 12)**

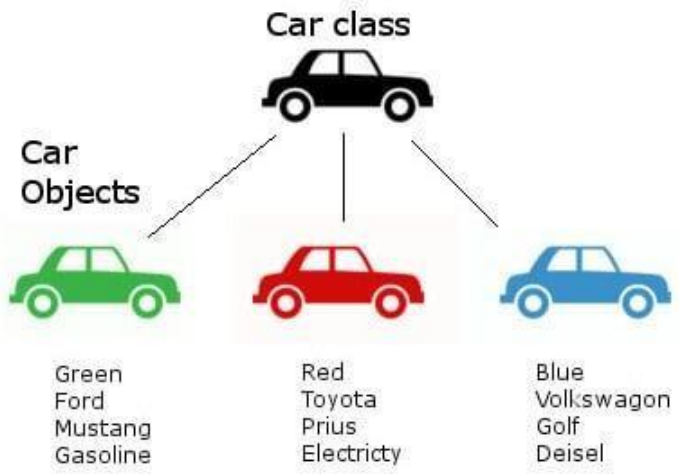


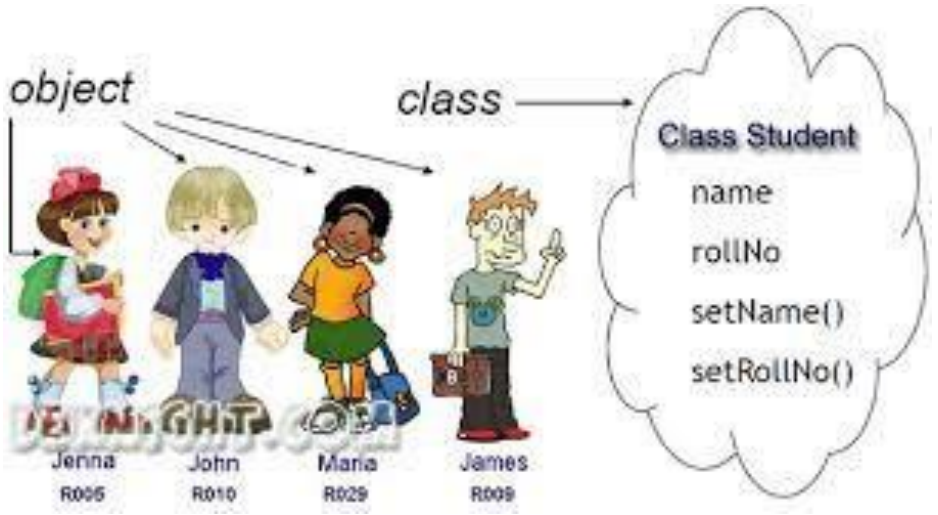
2. Lớp, đối tượng trong Python

Lớp, đối tượng trong Python

1. Giới thiệu lớp

- Python là một ngôn ngữ lập trình hướng đối tượng





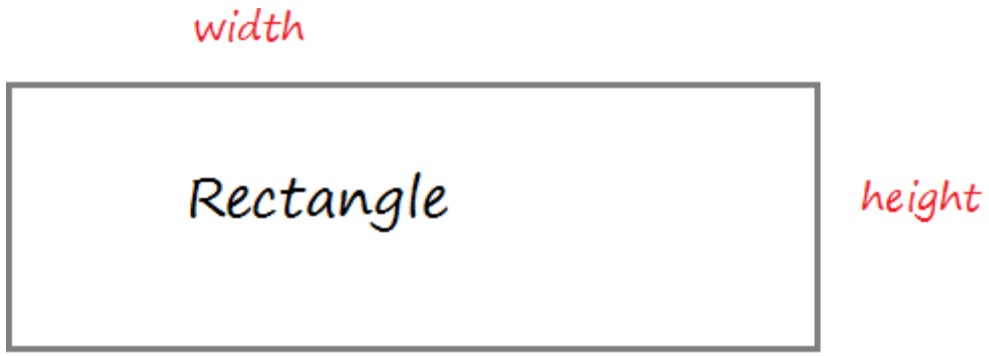
➢ Khai báo Class trong Python

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

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

Lớp, đối tượng trong Python

1. Ví dụ




```python
#Tạo một lớp Rectangle

class Rectangle:
    #Lớp Rectangle có 2 thuộc tính: width, height

    #Phương thức khởi tạo đối tượng (Contructor)
    def __init__(self, width, height):
        self.width = width
        self.height = height

    #Phương thức tính diện tích
    def getArea(self):
        area = round(self.width * self.height,1)
        return area

    #Phương thức tính chu vi
    def getPerimeter(self):
        perimeter = round((self.width + self.height)*2,1)
        return perimeter
```

Lớp, đối tượng trong Python

➢ Một số khái niệm hướng đối tượng

- **Thuộc tính (Attribute):** Thuộc tính là một thành viên của lớp, Hình chữ nhật có 2 thuộc tính width và height
- **Phương thức (Method):**
  - Phương thức của class tương tự như một hàm thông thường, nhưng nó là mộthàm của class. Để sử dụng được cần phải gọi thông qua đối tượng.
  - Tham số đầu tiên của phương thức luôn là self (Một từ khóa ám chỉ chính classđó)
- **Phương thức khởi tạo (Constructor):**
  - Là một phương thức đặc biệt của lớp, luôn có tên là__init__. Tham số đầu tiên luôn là self. Chỉ có thể định nghĩa một contructor trong class
  - Constructor được sử dụng để tạo ra một đối tượng.
  - Constructor gán các giá trị từ tham số vào các thuộc tính của đối tượng sẽ được tạo ra

Lớp, đối tượng trong Python

➢ Khởi tạo đối tượng từ lớp

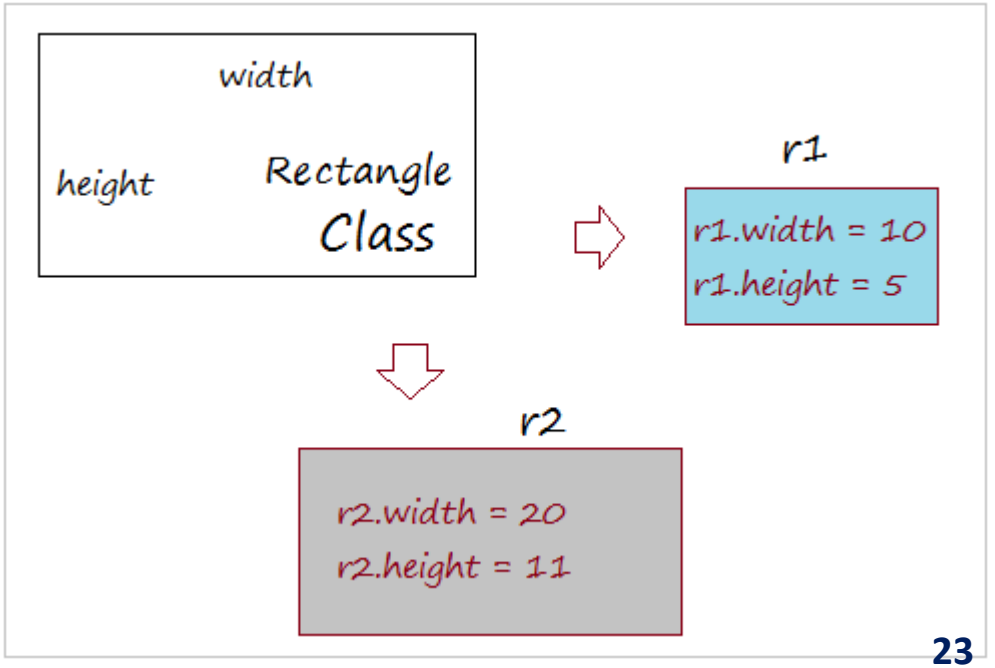
Sau khi đã khai báo được class trong Python rồi, thì để khởi tạo các đối tượngsửdụng cú pháp sau:

```
variableName = className()
```

Trong đó:

- **variableName** là tên đối tượng.
- **className** là class muốn khởi tạo.

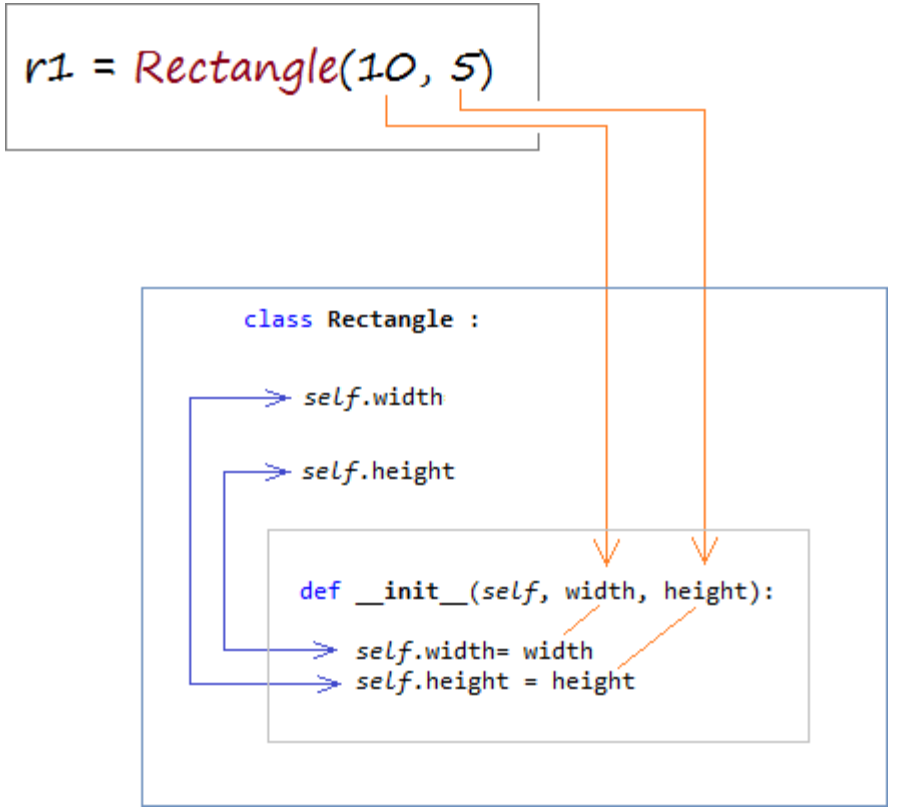
```
1  #Ví dụ: Tạo 2 đối tượng r1, r2 từ class Rectangle
2  r1 = Rectangle(10,5)
3
4  r2 = Rectangle(20,11)
```



Lớp, đối tượng trong Python

➢ Điều gì xẩy ra khi khởi tạo một đối tượng

Khi tạo một đối tượng của lớp **Rectangle**, phương thức khởi tạo (constructor) của class đó sẽ được gọi để tạo một đối tượng, và các thuộc tính của đối tượng sẽ được gán giá trị từ tham số



Lớp, đối tượng trong Python

- Sau khi đã khởi tạo được đối tượng sẽ có thể truy cập được các thuộc tính và phương thức trong class đó.
- Bằng cách sử dụng dấu . theo cú pháp sau:

```
# truy cap 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 Rectangle

```
#Lấy thuộc tính width, height  của đối tượng rec1
x = r1.width
y = r1.height
print('----Thuộc tính----------------')
print('1. Thuộc tính Chiều rộng: ', x)
print('2. Thuộc tính Chiều dài: ', y)
#Gọi phương thức getArea, getPerimeter của đối tượng rec1
dt = r1.getArea()
cv = r1.getPerimeter()
print('---------Phương thức----------')
print('1. Phương thức tính Diện tích:', dt)
print('2. Phương thức tính Chu vi:',cv)
```

```
----Thuộc tính----------------
1. Thuộc tính Chiều rộng:  10
2. Thuộc tính Chiều dài:  5
---------Phương thức----------
1. Phương thức tính Diện tích: 50
2. Phương thức tính Chu vi: 30
```

Thực hành

Bài 16: Xây dựng lớp Person

Xây dựng lớp Person:

- **Gồm 4 Thuộc tính:**
  - họ tên (name), năm sinh (year), chiều cao (height), cân nặng (weight)
  - Giá trị mặc định của các thuộc tính là thông tin của bạn
- **Gồm 2 Phương thức:**
  - Geeting(): Hiển thị thông tin của Person
  - Bmi(): Tính toán chỉ số BMI của Person

```
1  #Tạo đối tượng p1 thuộc lớp Person
2  p1 = Person('Đặng Văn Nam', 1985, 1.65, 59)
```

```
1  #Gọi phương thức Geeting()
2  p1.Geeting()
```

```
My name is  Đặng Văn Nam
I am  36  years old. Nice to meet you!
```

```
1  #Gọi phương thức BMI
2  print('Chỉ số BMI:',p1.BMI())
```

```
Chỉ số BMI: 21.7
```

3. Module và Package

Module và Package

- Module hiểu ngắn gọn là một file python mà trong đó chứa các khai báo và định nghĩa về hàm số và biến. Các chương trình python sẽ được thiết kế sao cho nội dung được chia nhỏ về các files module để dễ dàng quản lý. Chúng ta có thể dễ dàng import lại cáckhai báo và định nghĩa từ module này sang module khác.
- Một package sẽ bao gồm nhiều file module

```
1  #gọi package để sử dụng
2  import math
3
4  #Liệt kê tất cả các phương thức, thuộc tính của package math
5  print (dir(math))
```

```
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'a
tan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'exp
m1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclos
e', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan',
'pi', 'pow', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
```

3. Làm việc với tập tin trong Python

Excel Data (Print) - Notepad

File Edit Format View Help

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	22-01-2010	KCS
1400-57-0902AM03	w13	V5	12	19-01-2010	KCS

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 tênvà 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 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

Mode	Chú thích
r	Chế độ chỉ được phép đọc.
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 nósẽ ghi đè nội dung lên file cũ.
a	Mở file trong chế độ ghi tiếp. Nếu file đã tồn tại rồi thì nósẽ ghi tiếp nội dung, và nếu như file chưa tồn tại thì nósẽ tạo một file mới và ghi nội dung vào đó.
a+	Mở file trong chế độ đọc và ghi tiếp nội dung, còn lại cơchế giống chế độ a.

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 phươngthức **read** với cú pháp:

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

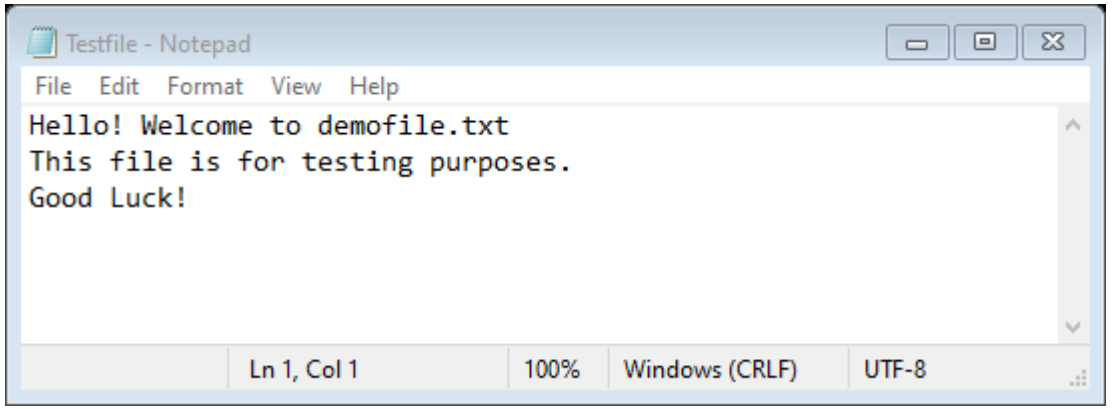
Trong đó:

- 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:




```
1  #Mở file để đọc dữ liệu
2  f = open('Testfile.txt')
3
4  #Đọc nội dung của file vào biến st
5  st = f.read()
6
7  print('Nội dung file:')
8  print(st)
```

```
Nội dung file:
Hello! Welcome to demofile.txt
This file is for testing purposes.
Good Luck!
```

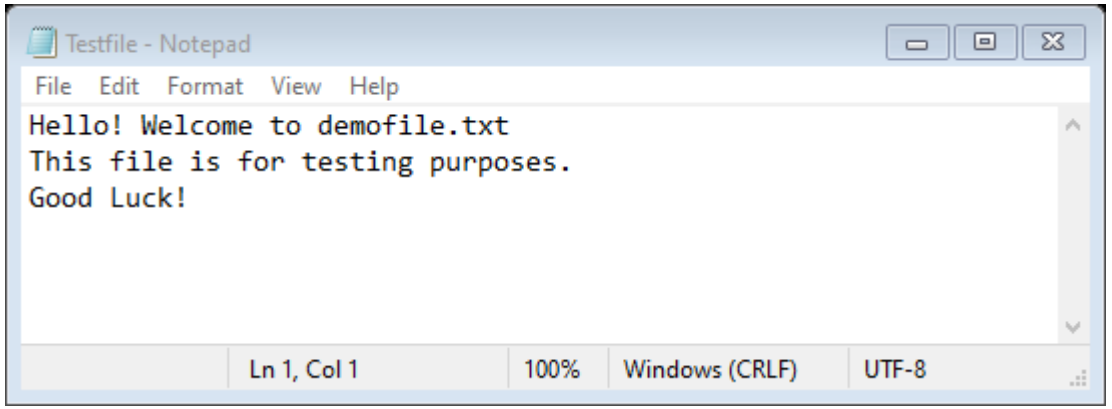
```
1  #Mở file để đọc dữ liệu
2  f = open('Testfile.txt',"r")
3
4  #Đọc 10 ký tự đầu tiên của file
5  st1 = f.read(15)
6
7  print(st1, ' -- Số ký tự là: ', len(st1))
```

```
Nội dung file:
Hello! Welcome   -- Số ký tự là:  15
```

Làm việc với tập tin (File)

4. Đọc file

- Ví dụ:




```
1  #Mở file để đọc dữ liệu
2  f = open('Testfile.txt')
3
4  #Đọc từng dòng dữ liệu của file
5  print(f.readline())
6  print(f.readline())
7
8  f.close() #Đóng file dữ liệu
```

```
Hello! Welcome to demofile.txt

This file is for testing purposes.
```

```
1  #Mở file để đọc dữ liệu
2  f = open("Testfile.txt", "r")
3
4  #đọc tất cả các dòng của file
5  for x in f:
6      print(x)
7
8  f.close() #Đóng file dữ liệu
```

```
Hello! Welcome to demofile.txt

This file is for testing purposes.

Good Luck!
```

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 phépghi. 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 với chế độ ghi đè (w):
#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.
f1 = open('Ghifile.txt', 'w')

#Dữ liệu muốn ghi vào file
st = 'Welcome to Python for Analysis!'

f1.write(st) #Ghi dữ liệu vào file
f1.close() #Đóng file
```

Làm việc với tập tin (File)

5. Ghi file

➢ Ví dụ:

```
1  #Mở file với chế độ ghi tiếp (a):
2  # Nếu file đã tồn tại rồi thì nó sẽ ghi tiếp nội dung,
3  # và nếu như file chưa tồn tại thì nó sẽ tạo một file mới và ghi nội dung vào đó.
4  f1 = open('Ghifile.txt', 'a+')
5
6  #Dữ liệu muốn ghi vào file
7  st = 'This is new line.....'
8
9  f1.write(st) #Ghi tiếp dữ liệu vào file
10 f1.close() #Đóng file
11
12 #open and read the file after the appending:
13 f = open("Ghifile.txt", "r")
14 print(f.read())
```

```
Welcome to Python for Analysis!This is new line.....
```

Làm việc với tập tin (File)

- Các thuộc tính trong file.

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

- Ví dụ:

In ra thông số của file Ghifile.txt ở trên

```
#Lấy các thông số của file
f2 = open('Ghifile.txt')

print('1.Tên file:',f2.name)
print('2.Chế độ mở file:',f2.mode)
print('3.Trạng thái đóng file:',f2.closed)
```

```
1.Tên file: Ghifile.txt
2.Chế độ mở file: r
3.Trạng thái đóng file: 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

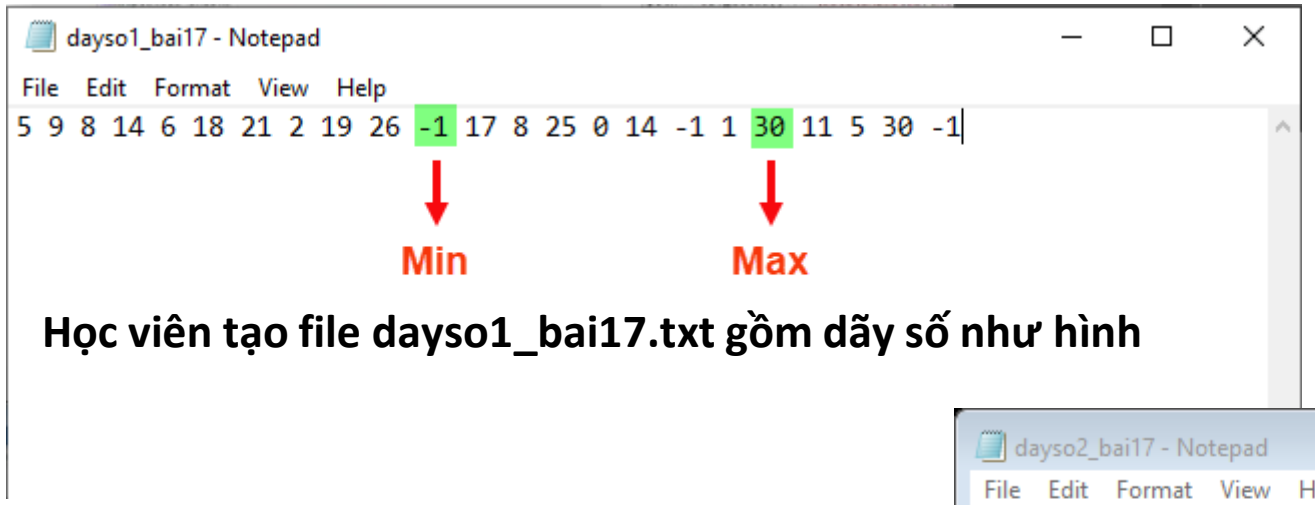
```
obj=open("test.txt","w")
obj.write("Chao mung cac ban den voi khoa CNTT")
obj.close()
obj1=open("test.txt","r")
s=obj1.read()
print (s)
obj1.close()
obj2=open("test.txt","r")
s1=obj2.read(20)
print (s1)
obj2.close()
```

Thực hành

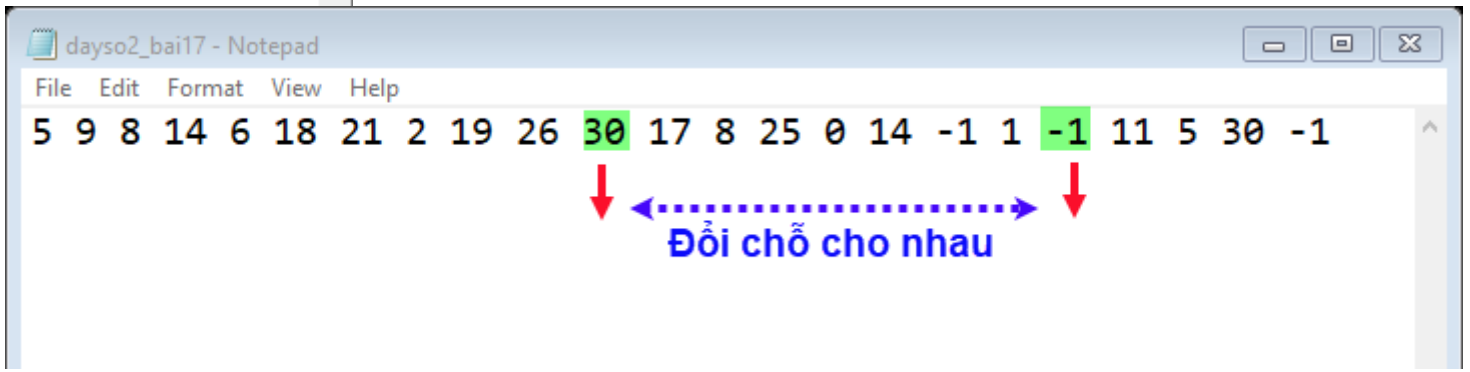
Bài 17: Đọc/Ghi file

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



Học viên tạo file dayso1_bai17.txt gồm dãy số như hình



TEST– w3schools.com

https://www.w3schools.com/python/exercise.asp?filename=exercise_syntax1