**I.CÂU ĐIỀU KIỆN**

print ("Nhap a :")

a=int(input())

print ("Nhap b :")

b=int(input())

print ("Nhap c :")

c=int(input())

if (b<c) and (c>a):

print ("max=c")

elif (b>c) and (b>a):

print ("max=b")

else:

print ("max a")

**II.VÒNG LẶP FOR**

A = [1, 2, 5, 22, 11, 4, 6, 8, 10]

tong = 0

for a in A:

tong = tong+a

print('tong cac so la:',tong)

**III.VÒNG LẶP WHILE**

i= 0

n = 0

while (i< 10):

print (“so thu nhat la:”, i)

n = n + 1

i = i+ 1

**IV.LIST**

list = [1,’Luu Uy’,2];

print "list[1]: ", list[1]

**V.CLASS**

* **Class**: Được định nghĩa cho một đối tượng bao gồm một tập các attributes đặc trưng cho tất cả các đối tượng của lớp
* **Cú pháp khởi tạo class**

Class ClassName(object):

Có thể mô tả class(không bắt buộc)

#Code…

* **Thuộc tính (Attributes):**
  + - Thuộc tính là thành viên của lớp
    - Thuộc tính gồm: data members và methods được gọi thông qua ký hiệu dấu chấm.
    - Để khai báo một thuộc tính trong class thì ta chỉ cần khai báo như khai báo một biến bình thường và lưu ý nó phải nằm trong phạm vi của class
* **Phương thức (Method):**
  + - Phương thức của class nó tương tự như một hàm thông thường, nhưng nó là một hàm của class, để sử dụng nó bạn cần phải gọi thông qua đối tượng.
    - Tham số đầu tiên của phương thức luôn là **self**(Một từ khóa ám chỉ chính class đó).

### Phương thức khởi tạo (Constructor):

* Phương thức khởi tạo (Constructor) là một phương thức đặc biệt của lớp (class), nó luôn có tên là **\_\_init\_\_**
* Tham số đầu tiên của constructor luôn là **self** (Một từ khóa ám chỉ chính class đó).
* 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.
* Bạn chỉ có thể định nghĩa nhiều nhất một phương thức khởi tạo (constructor) trong class.
* Nếu class không được định nghĩa constructor, **Python** mặc định coi rằng nó thừa kết từ constructor của lớp cha.

**VI.FUCTION**

Hàm (function) là một khối lệnh đặc biệt, nó được đặt tên, giúp mã chương trình dễ đọc hơn, và có thể gọi để sử dụng ở các nơi khác nhau trong chương trình. Hàm chính là khối lệnh có thể tái sử dụng (reuseable).

* **Cú pháp**
* Một hàm bắt đầu bởi từ khóa def (Viết tắt của từ define), theo sau đó là tên của hàm.
* Tiếp theo là danh sách các tham số nằm trong cặp đấu ngặc ( ) và dấu hai chấm ( : ), hàm có thể có 0, 1 hoặc nhiều tham số, các tham số cách nhau bởi dấu phẩy.
* Dòng đầu tiên của thân hàm (function body) là một chuỗi mô tả ngắn về hàm (Không bắt buộc).

#Ví dụ:

def functionName( parameters ):

     "Mô tả ngắn về hàm"

     codes ...

    return [expression]

* **Lệnh return:**

Lệnh **return** dùng để trả về một giá trị (hoặc một biểu thức), hoặc đơn giản là trả về **"không gì cả"**. Khi lệnh **return** được thực thi, hàm sẽ kết thúc. **return** là lệnh không bắt buộc phải có trong thân hàm.

* **Hàm với tham số mặc định**:

def ten\_ham(thamso = defaultValue):

# code

defaultValue là tham số có giá trị mặc định

* **Hàm với tham số bắt buộc:**

def ten\_ham(thamso1,thamso2):

#code

Thamso1,thamso2 là các tham số bắt buộc

* **Hàm nặc danh:**
* Hàm nặc danh có thể có 0 hoặc nhiều tham số, nhưng trong thân hàm chỉ có duy nhất một biểu thức (expression). Giá trị của biểu thức chính là giá trị trả về của hàm. Nhưng không được sử dụng từ khóa **'return'** ngay trước biểu thức.
* Danh sách các tham số cách nhau bởi dấu phẩy, và không được đặt trong cặp dấu ngặc tròn ( ).
* Trong thân của hàm nặc danh, bạn không thể truy cập các biến bên ngoài, bạn chỉ có thể truy cập các tham số của nó.
* Hàm nặc danh không thể gọi trực tiếp hàm print, bởi vì **lambda** đòi hỏi một biểu thức.