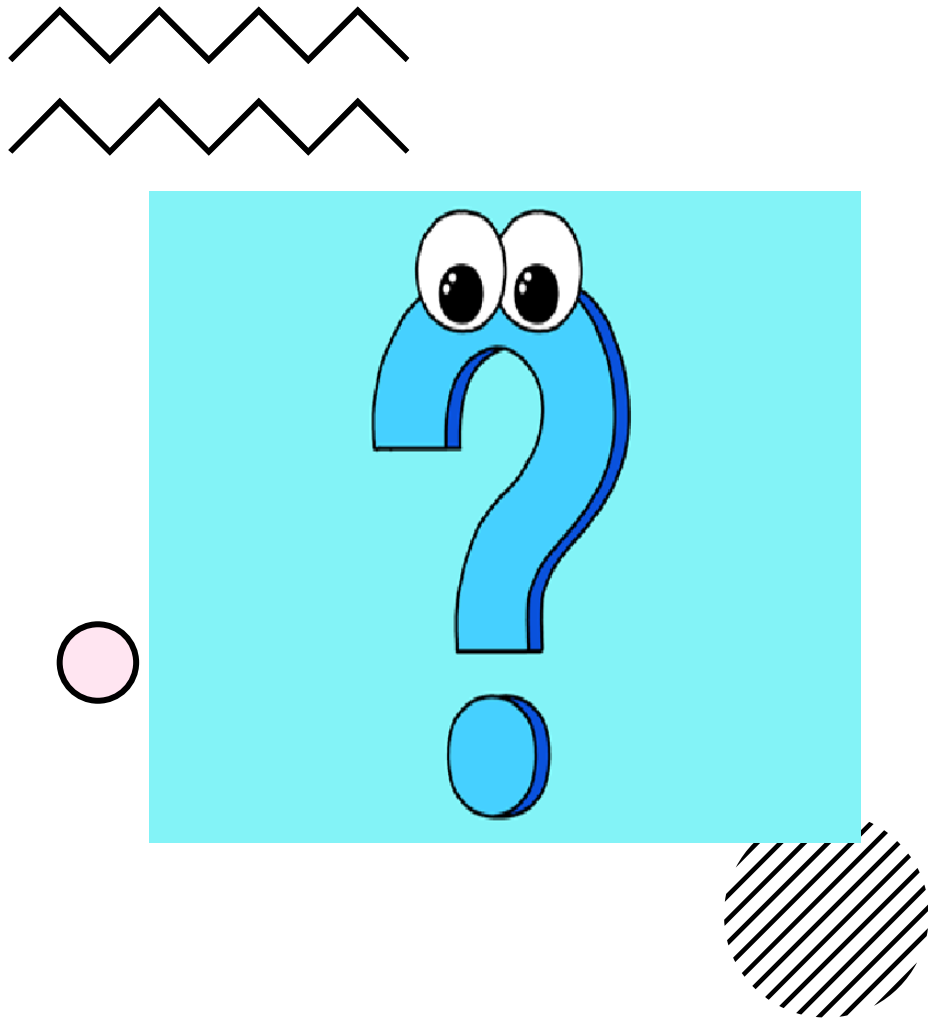


# BÀI 28

## PHẠM VI CỦA BIẾN





1. Một biến được định nghĩa trong chương trình chính (bên ngoài các hàm) thì sẽ được sử dụng như thế nào bên trong các hàm ?
2. Một biến được khai báo bên trong một hàm thì có sử dụng được ở bên ngoài hàm đó hay không?

# 1. PHẠM VI CỦA BIẾN KHAI BÁO TRONG HÀM

Các biến được khai báo bên trong một hàm chỉ được sử dụng bên trong hàm đó. Chương trình chính không sử dụng được.

```
HoangThithanhTam_TL.py ×  
1  def func(a,b) :  
2      n = 10  
3      a = a * 2  
4      b = a+b  
5      return a + b + n  
6  a = 1  
7  b = 2  
8  print(func(a,b))  
9  print(a,b)  
10 print(n)
```

Các biến n, a, b đang hoạt động trong hàm

Các biến bên ngoài hàm a,b

Shell x

```
>>> %Run HoangThithanhTam_TL.py
```

```
16
```

```
1 2
```

```
Traceback (most recent call last):
```

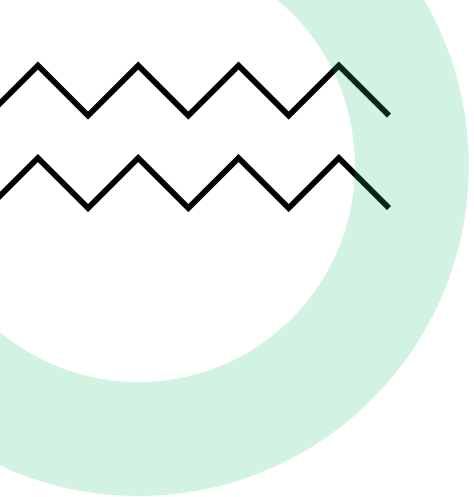
```
  File "C:\Users\Admin\Downloads\HoangThithanhTam_TL.py", line  
10, in <module>
```

```
    print(n)
```

```
NameError: name 'n' is not defined
```

Sau khi chạy hàm, các biến a,  
b vẫn không thay đổi giá trị

Biến n gọi bên ngoài hàm sẽ bị báo lỗi



**Như vậy:**

Trong Python tất cả các biến khai báo bên trong hàm đều có tính địa phương (cục bộ), không có hiệu lực ở bên ngoài hàm.





1. Giả sử có các lệnh sau:

```
>>> a, b = 1, 2
```

```
>>> def f(a, b):
```

```
    a = a+b
```

```
    b = b*a
```

```
    return a + b
```

Giá trị của a, b bằng bao nhiêu sau khi thực hiện lệnh sau?

a) f(1, 2)      b) f(10, 20)

2. Ta có thể khai báo một biến bên trong hàm trùng tên với biến đã khai báo trước đó bên ngoài hàm không?

## 2. PHẠM VI CỦA BIẾN KHAI BÁO NGOÀI HÀM

**Ví dụ 1.** Biến khai báo bên ngoài hàm không có tác dụng bên trong hàm.

```
>>> def f(n):  
    t = n + 1  
    return t
```

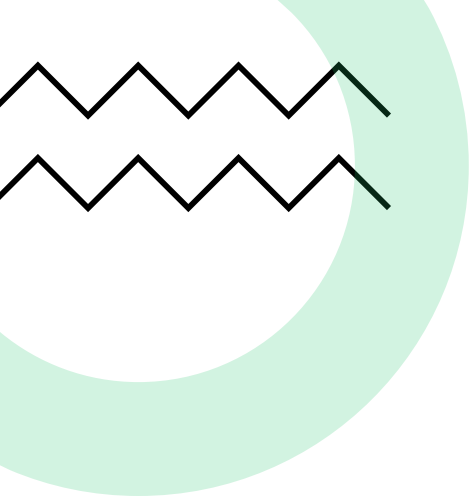
```
>>> t = 10
```

```
>>> f(5)
```

```
6
```

```
>>> t
```

```
10
```



**Ví dụ 2.** Bên trong hàm có thể truy cập để sử dụng giá trị của biến đã khai báo trước đó ở bên ngoài hàm.

```
>>> def f(a, b):  
    return a + b + N
```

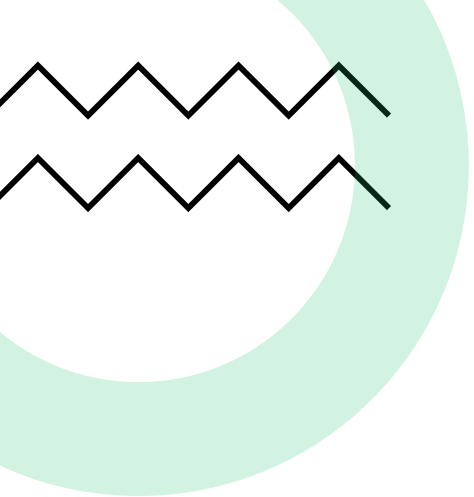
```
>>> N = 10
```

```
>>> f(1, 2)
```

```
13
```







**Lưu ý:** nếu muốn biến bên ngoài vẫn có tác dụng bên trong hàm thì chỉ cần khai báo lại biến này bên trong hàm với từ khóa **global**

```
>>> def f(n) :  
    global t  
    t = 2*n + 1  
    return t
```

```
>>> t = 10
```

```
>>> f(1)
```

```
3
```

```
>>> t
```

```
3
```

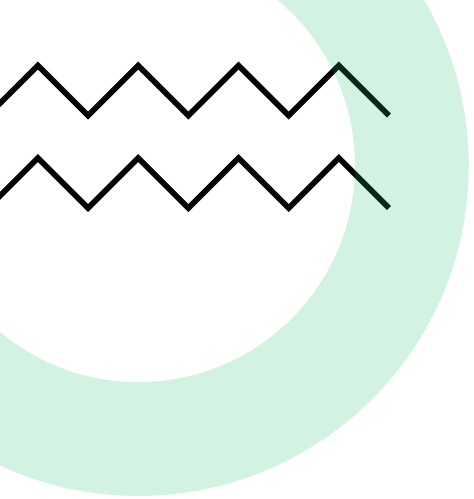




## Tóm lại:

Biến đã khai báo bên ngoài hàm chỉ có thể truy cập giá trị để sử dụng bên trong hàm mà không làm thay đổi được giá trị của biến đó (trừ trường hợp với từ khóa **global**)





Giả sử hàm  $f(x, y)$  được định nghĩa như sau:

```
>>> def f (x, y) :  
    a = 2* (x + y)  
    print(a + n)
```

Kết quả nào được in ra khi thực hiện các lệnh sau?

```
n = 10
```

```
f(1, 2)
```





## 3. Thực Hành

### Phạm vi của biến

**Nhiệm vụ 1.** Viết hàm với đầu vào là danh sách A chứa các số và số thực x. Hàm trả lại một danh sách kết quả B từ danh sách A bằng cách chỉ giữ lại các phần tử lớn hơn hoặc bằng x.

**Hướng dẫn.** Biến B kiểu danh sách cần được định nghĩa trong hàm và được bổ sung thêm các phần tử từ A nếu thỏa mãn điều kiện lớn hơn hoặc bằng x.

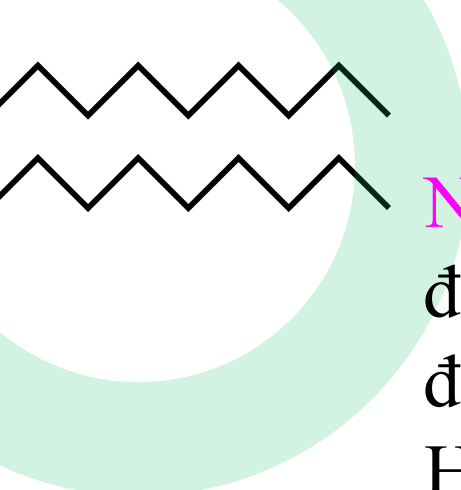




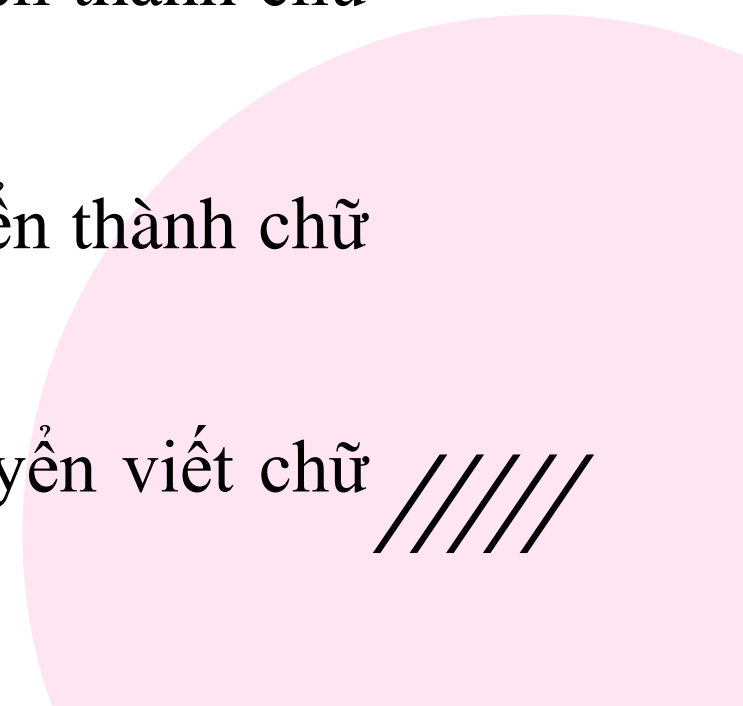
HoangThithanhTam\_TL.py \* x

```
1  def Select( A, x ) :  
2      B = [ ]  
3      for k in range(len(A)):  
4          if A[k] >= x :  
5              B.append(A[k])  
6      return B  
7
```





**Nhiệm vụ 2.** Viết hàm với đầu vào là xâu kí tự Str và số c, đầu ra là danh sách các từ được tách ra từ xâu Str nhưng đã được chuyển thành chữ in hoa hoặc chữ in thường. Hoặc chỉ chuyển kí tự đầu các từ thành chữ in hoa tùy thuộc vào tham số đầu vào c như sau :

- Nếu  $c = 0$ , danh sách B là các từ được chuyển thành chữ in hoa.
  - Nếu  $c = 1$ , danh sách B là các từ được chuyển thành chữ in thường.
  - Nếu  $c = 2$ , danh sách B là các từ được chuyển viết chữ hoa kí tự đầu của mỗi từ.
- 



**Hướng dẫn.** Chúng ta cần sử dụng các lệnh sau:

**Str.upper()** – chuyển kí tự của xâu thành chữ in hoa.

**Str.lower()** – chuyển kí tự của xâu thành chữ in thường.

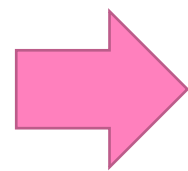
**Str.title()** – chuyển kí tự đầu mỗi từ của xâu thành chữ in hoa, các kí tự khác chuyển về chữ thường

- Hàm được định nghĩa có dạng **Tach\_tu(Str, c)**. Đầu tiên xâu Str cần được tách từ bằng lệnh `split()`. Sau đó danh sách kết quả sẽ được chuyển đổi chữ in hoa, in thường sử dụng một trong các lệnh trên tùy thuộc vào giá trị của đối số c.



HoangThithanhTam\_TL.py ×

```
1 def Tach_tu(Str, c):
2     A = Str.split()
3     for k in range(len(A)):
4         if c == 0:
5             A[k] = A[k].upper()
6         if c == 1:
7             A[k] = A[k].lower()
8         if c == 2:
9             A[k] = A[k].title()
10    return A
11 s="hoàng thị thanh tâm"
12 c=int(input("nhập c = 0,1,2: "))
13 a=Tach_tu(s,c)
14 print(a)
```

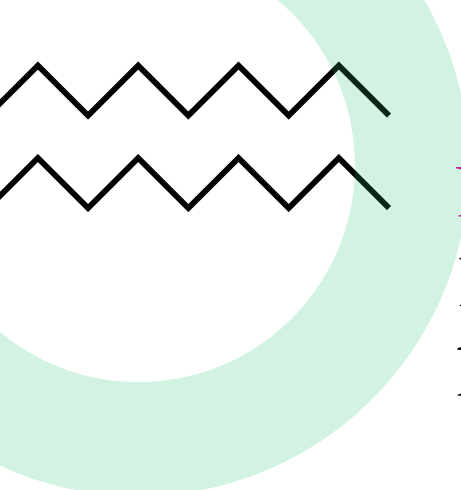


Shell ×

```
>>> %Run HoangThithanhTam_TL.py
nhập c = 0,1,2: 0
['HOÀNG', 'THỊ', 'THANH', 'TÂM']
>>> %Run HoangThithanhTam_TL.py
nhập c = 0,1,2: 1
['hoàng', 'thị', 'thanh', 'tâm']
>>> %Run HoangThithanhTam_TL.py
nhập c = 0,1,2: 2
['Hoàng', 'Thị', 'Thanh', 'Tâm']
```







**Nhiệm vụ 3.** Viết chương trình yêu cầu thực hiện lần lượt các việc sau, mỗi việc cần được thực hiện bởi một hàm:

1. Nhập từ bàn phím một dãy các số nguyên, mỗi số cách nhau bởi dấu cách. Chuyển các số này vào danh sách A và in danh sách A ra màn hình.
2. Trích từ danh sách A ra một danh sách B gồm các phần tử lớn hơn 0. In danh sách B ra màn hình.
3. Trích từ danh sách A ra một danh sách C gồm các phần tử nhỏ hơn 0. In danh sách C ra màn hình.

**Hướng dẫn.** Với mỗi việc trên được viết thành một hàm. Toàn bộ chương trình có thể như sau:



```
1 t = 0
2 def Nhap_Dulieu():
3     s = input("Nhập các số nguyên cách nhau bởi dấu cách: ")
4     A = s.split()
5     for k in range(len(A)): A[k] = int(A[k])
6     return A
7 def getB(A):
8     B = []
9     for x in A:
10         if x > 0: B.append(x)
11     return B
12 def getC(A):
13     C = []
14     for x in A:
15         if x < 0: C.append(x)
16     return C
17 # Chương trình chính
18 A = Nhap_Dulieu()
19 print("Danh sách A:", A)
20 B = getB(A); C = getC(A)
21 print("Danh sách B:", B)
22 print("Danh sách C:", C)
```



Shell x

Nhập các số nguyên cách nhau bởi dấu cách: 2 5 -9 6 -7 8 -4 3

Danh sách A: [2, 5, -9, 6, -7, 8, -4, 3]

Danh sách B: [2, 5, 6, 8, 3]

Danh sách C: [-9, -7, -4]





# LUYỆN TẬP

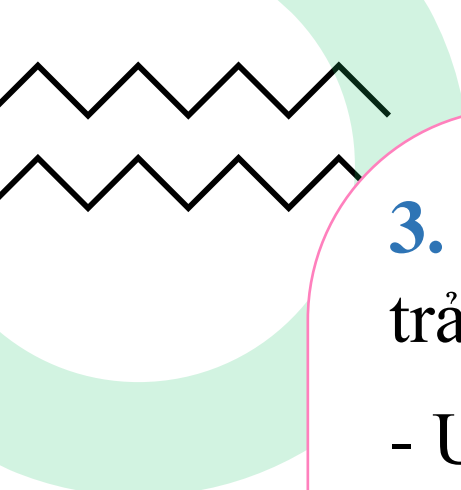
**1.** Viết hàm với đầu vào, đầu ra như sau:

- Đầu vào là danh sách slist, các phần tử là xâu kí tự.
- Đầu ra là danh sách clist, các phần tử là kí tự đầu tiên của các xâu kí tự tương ứng trong danh sách slist.

**2.** Viết hàm Tach\_day() với đầu vào là danh sách A, đầu ra là hai danh sách B, C được mô tả như sau:

- Danh sách B thu được từ A bằng cách lấy ra các phần tử có chỉ số chẵn.
- Danh sách B thu được từ A bằng cách lấy ra các phần tử có chỉ số lẻ.





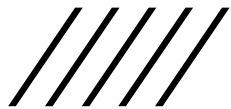
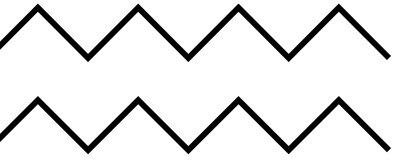
**3.** Viết hàm có hai tham số đầu vào là  $m$ ,  $n$ . Đầu ra trả lại hai giá trị là:

- ƯCLN của  $m$ ,  $n$ .
- Bội chung nhỏ nhất (BCNN) của  $m$ ,  $n$ .

*Gợi ý:* Sử dụng công thức  $\text{ƯCLN}(m, n) \times \text{BCNN}(m, n) = m \times n$

**4.** Viết chương trình nhập ba số tự nhiên từ bàn phím day, month, year, các số cách nhau bởi dấu cách. Các số này biểu diễn giá trị của ngày, tháng, năm nào đó. Chương trình cần kiểm tra và in ra thông báo số liệu đã nhập vào đó có hợp lệ hay không.





# THANK YOU

PRESENTER NAME  
EMAIL  
WEBSITE

# TOPIC ONE

SUBTITLE

