



1



1. Một số quy tắc



- Python không dùng cặp “{}” để bao khối lệnh (nhiều lệnh) mà áp dụng quy tắc thụt dòng
- Các lệnh đồng cấp phải thụt dòng như nhau, nếu không sẽ gây lỗi
- VD: Khối lệnh bên trong các lệnh điều khiển if, lệnh lặp, hay hàm phải thụt dòng sâu hơn so với các lệnh điều khiển/hàm này.

```

a=10
b=100
if a<b:
    →c=b
else:
    →c=a
print('max({0}, {1})={2}'.format(a, b, c))
  
```

2

2

1. Một số quy tắc

- 1 lệnh viết trên nhiều dòng thì phải dùng “\” để nối các dòng.


```
sum = a + b + c \
      + e + f
```
- Khai báo các biến có dùng [], (), {} trên nhiều dòng thì không cần “\” để nối các dòng.


```
giaCam = [“Gà”, “Vịt”,
          “Chim bồ câu”]
```

3

2. Biểu thức logic

- Biểu thức Boolean (Boolean Expression) còn được gọi là Predicate.
- Kết quả của biểu thức là giá trị là **True** hoặc **False**, dựa vào các giá trị này mà ta điều hướng các công việc trong phần mềm.

Biểu thức	Ý nghĩa
$x == y$	True nếu $x=y$, False nếu x khác y
$x < y$	True nếu $x < y$, False nếu $x \geq y$
$x \leq y$	True nếu $x \leq y$, False nếu $x > y$
$x > y$	True nếu $x > y$, False nếu $x \leq y$
$x \geq y$	True nếu $x \geq y$, False nếu $x < y$
$x != y$	True nếu x khác y , False nếu $x=y$

4

3. Cấu trúc rẽ nhánh

```

hoten=input("Nhập họ tên:")
dtb=float(input("Nhập điểm tb:"))
if dtb>=5:
    print("Bạn %s đã đậu! Chúc mừng bạn" %hoten)
else:
    print("Ban %s đã rớt! Cố gắng lần sau"%hoten)

```

Nhập họ tên:Trần Hùng
 Nhập điểm tb:7
 Bạn Trần Hùng đã đậu! Chúc mừng bạn

Nhập họ tên:Nguyễn Lan Hương
 Nhập điểm tb:3.5
 Bạn Nguyễn Lan Hương đã rớt! Cố gắng lần sau

if condition :

if-block

else:

else-block

```

graph TD
    Start(( )) --> Condition{Condition}
    Condition -- true --> IfBlock[/If-block/]
    Condition -- false --> ElseBlock[/else-block/]
    IfBlock --> Merge(( ))
    ElseBlock --> Merge
    Merge --> Exit[Exit]

```

5

5

3. Cấu trúc rẽ nhánh

```

if condition1:
    Block1
elif condition2:
    Block2
elif condition3:
    Block3
...
else:
    Blockn

```

```

hoten=input("Nhập họ tên:")
dtb=float(input("Nhập điểm tb:"))
if dtb>=8.5:
    xeploai='Xuất sắc'
elif dtb>=7:
    xeploai='Khá'
elif dtb>=5.5:
    xeploai='Trung bình'
else:
    xeploai='Yếu'
print('Bạn {0} xếp loại: {1}'.format(hoten, xeploai))

```

Nhập họ tên:Trần Hùng
 Nhập điểm tb:9
 Bạn Trần Hùng xếp loại: Xuất sắc

Nhập họ tên:Nguyễn Văn Cảnh
 Nhập điểm tb:6
 Bạn Nguyễn Văn Cảnh xếp loại: Trung bình

if condition₁:

Block₁

elif condition₂:

Block₂

elif condition₃:

Block₃

...

else:

Block_n

```

graph TD
    Start(( )) --> Cond1{condition1}
    Cond1 -- Đ --> B1[Block1]
    Cond1 -- S --> Cond2{condition2}
    Cond2 -- Đ --> B2[Block2]
    Cond2 -- S --> Cond3{condition3}
    Cond3 -- Đ --> B3[Block3]
    Cond3 -- S --> Bn[Blockn]
    B1 --> Merge(( ))
    B2 --> Merge
    B3 --> Merge
    Bn --> Merge
    Merge --> Exit(( ))

```

6

6

3. Cấu trúc rẽ nhánh

■ Biểu thức **pass**

```
'''
    Giai phương trình bậc nhất
    ax + b = 0
'''
print('Giai phương trình bậc nhất ax + b = 0')
a=float(input('Nhap a:'))
b=float(input('Nhap b:'))
if a==0:
    else:
        x=(-b)/a
        print('{0}x+{1}=0: nghiem x={2}'.format(a, b, x))
```

```
'''
    Giai phương trình bậc nhất
    ax + b = 0
'''
print('Giai phương trình bậc nhất ax + b = 0')
a=float(input('Nhap a:'))
b=float(input('Nhap b:'))
if a==0:
    pass
else:
    x=(-b)/a
    print('{0}x+{1}=0: nghiem x={2}'.format(a, b, x))
```

} {

Lỗi

7

3. Cấu trúc rẽ nhánh

■ So sánh số thực

```
a=3.10 - 1.10
b=5.10 - 3.10
d=math.fabs(a-b)
if a==b:
    print('{0} == {1}'.format(a, b))
else:
    print('{0} != {1}'.format(a, b))

2.0 != 1.9999999999999996
```

```
import math
a=3.10 - 1.10
b=5.10 - 3.10
d=math.fabs(a-b)
if d<0.000001:
    print('{0} == {1}'.format(a, b))
else:
    print('{0} != {1}'.format(a, b))

2.0 == 1.9999999999999996
```

8

3. Cấu trúc rẽ nhánh

■ Sử dụng cấu trúc rẽ nhánh bằng phép gán

```

a=10
b=100
if a<b:
    c=b
else:
    c=a
print('max({0}, {1})={2}'.format(a, b, c))

```

→

```

a=10
b=100
c=b if a<b else a
print('max({0}, {1})={2}'.format(a, b, c))


```

9


Bài tập

1. Tìm min 2 số. Vd: a=10, b=20 → 10
 - min=a
 - Nếu a>b: min=b
2. Xếp loại điểm trung bình:
 - Nhập họ tên, điểm trung bình
 - Xếp loại:
 - dtb ∈ [8.5, ..., 10]: xuất sắc
 - dtb ∈ [7.0, ..., 8.5): khá
 - dtb ∈ [5.5, ..., 7.0): trung bình
 - Còn lại: yếu

10



Bài tập




3. Giải phương trình bậc 1: $ax + b = 0$


- Nếu $a=0$:
 - ✓ Nếu $b=0$: $0x + 0 = 0 \rightarrow$ Phương trình vô số nghiệm
 - ✓ Nếu $b \neq 0$: $0x + b = 0 \rightarrow$ Phương trình vô nghiệm
- Nếu $a \neq 0$:
 - ✓ $x = -b/a$

11

11



Bài tập



4. Giải phương trình bậc 2: $ax^2 + bx + c = 0$

Tính **$\Delta = b^2 - 4ac$**

- Nếu $\Delta < 0$: phương trình vô nghiệm
- Nếu $\Delta = 0$: 1 nghiệm kép $\rightarrow x = -b/2a$
- Nếu $\Delta > 0$: 2 nghiệm phân biệt
 - $X1 = (-b + \sqrt{\Delta})/2a$
 - $X2 = (-b - \sqrt{\Delta})/2a$
- Lưu ý: hàm tính căn bậc 2 của a:


```
import math
a=4
c=math.sqrt(a) #2
```

12

12

4. Cấu trúc lặp

1 while

2 for

3 break

4 continue

5 while/else

6 for/else

7 Vòng lặp lồng nhau

13

13

4.1. Cấu trúc lặp while

■ Cú pháp:

```
while condition :  
    block
```

- block sẽ không được thực hiện nếu *condition==False* từ đầu
- Kết thúc vòng while khi:
 - *condition=False* hoặc
 - Sử dụng lệnh **break**

```

graph TD
    Entry(( )) --> Condition{Condition}
    Condition -- true --> Block[/block/]
    Block --> Condition
    Condition -- false --> Exit[Exit]
    
```

14

14

4.1. Cấu trúc lặp while

- Ví dụ 1: Yêu cầu nhập vào số nguyên dương $n \in [1..100]$

```

n=0
while n<1 or n>100:
    n=int(input('Nhập giá trị [1..100]:'))
print('n=', n)

Nhập giá trị [1..100]:0
Nhập giá trị [1..100]:199
Nhập giá trị [1..100]:50
n= 50

```

15

4.1. Cấu trúc lặp while

- Ví dụ 2: tính tổng n số nguyên đầu tiên:
 $1+2+3+\dots+n$
- $i=1 \rightarrow n: s=s+i$
- $N=5, s=0$
 - $i=1 \quad s=s+i=s+1=1$
 - $i=2 \quad s=s+i=s+2=1+2=3$
 - $i=3 \quad s=s+i=s+3=3+3=6$
 - $i=4 \quad s=s+i=s+4=6+4=10$
 - $i=5 \quad s=s+i=s+5=10+5=15$

16

4.1. Cấu trúc lặp while

- Ví dụ 2: tính tổng n số nguyên đầu tiên: $1+2+3+\dots+n$

```
#Tính tổng: s=1+2+...+n
n=int(input('Nhập n:'))
s=0
i=1
while i<=n:
    s+=i
    i=i+1
print('1+...+{0}={1}'.format(n, s))

Nhập n:5
1+...+5=15
```

17

17

4.2. Cấu trúc lặp for

- Cấu trúc lặp for thường sử dụng range để định nghĩa vùng dữ liệu lặp và bước nhảy.
- Cú pháp *range*:

range(begin, end, step)

 - begin: giá trị bắt đầu
 - end: giá trị cuối
 - step: bước nhảy

Ví dụ

 - `range(10)` ➔ 0; 1; 2; 3; 4; 5; 6; 7; 8; 9
 - `range(1, 10)` ➔ 1; 2; 3; 4; 5; 6; 7; 8; 9
 - `range(1, 10, 2)` ➔ 1; 3; 5; 7; 9
 - `range(10, 0, -1)` ➔ 10; 9; 8; 7; 6; 5; 4; 3; 2; 1
 - `range(10, 0, -2)` ➔ 10; 8; 6; 4; 2
 - `range(2, 11, 2)` ➔ 2; 4; 6; 8; 10
- Cú pháp *for* sử dụng *range*:

for bt in range():

block

18

18

4.2. Cấu trúc lặp for

Ví dụ:

```

for n in range(10): → 0 1 2 3 4 5 6 7 8 9
    print(n, end=' ')
for n in range(1, 10): → 1 2 3 4 5 6 7 8 9
    print(n, end=' ')
for n in range(1, 10, 2): → 1 3 5 7 9
    print(n, end=' ')
for n in range(10, 0, -1): → 10 9 8 7 6 5 4 3 2 1
    print(n, end=' ')
for n in range(10, 0, -2): → 10 8 6 4 2
    print(n, end=' ')
for n in range(2, 11, 2): → 2 4 6 8 10
    print(n, end=' ')

```

19

19

4.2. Cấu trúc lặp for

Ví dụ: In danh sách số chẵn < n

```

#In danh sách số chẵn < n
n=int(input('Nhập n:'))
s=""
for i in range(2, n, 2):
    s=s+str(i)+'\t'
print('Số chẵn < {0} là \n{1}'.format(n, s))

```

Nhập n:11
Số chẵn < 11 là
2 4 6 8 10

20

20

4.3. Câu lệnh break

- **break** là từ khóa đặc biệt dùng để thoát khỏi vòng lặp chứa nó.
- Khi gặp lệnh break, chương trình sẽ không thực hiện bất cứ lệnh nào bên dưới nó, mà thoát luôn khỏi vòng lặp.
- Ví dụ: Yêu cầu nhập vào số nguyên dương $n \in [1..100]$

```
while True:
    n = int(input('Nhập giá trị [1..100]:'))
    if 1<=n and n<=100:
        break
    print('Giá trị n: {0}'.format(n))

    Nhập giá trị [1..100]:0
    Nhập giá trị [1..100]:200
    Nhập giá trị [1..100]:80
    Giá trị n: 80
```

21

21

4.4. Câu lệnh continue

- **continue** là từ khóa đặc biệt dùng để nhảy tới lần lặp kế tiếp, các lệnh sau continue sẽ không được thực thi.
- Khác với lệnh break: gặp **break** thì dừng luôn vòng lặp còn **continue** chỉ dừng lần lặp hiện tại để chuyển qua lần lặp tiếp theo.
- Ví dụ: Tính tổng số lẻ $\leq n$ trừ các số lẻ chia hết cho 3

```
#Tổng số lẻ <=n trừ số lẻ chia hết cho 3
s=0
n = int(input('Nhập n:'))
for i in range(1, n+1, 2):
    if i%3==0:
        continue
    print(i)
    s=s+i
print('Tổng các số lẻ <={0} trừ các số lẻ chia hết cho 3 là: {1}'.format(n, s))
```

Nhập n:11

1

5

7

11

Tổng các số lẻ <=11 trừ các số lẻ chia hết cho 3 là: 24

22

22

4.5. while/else

- Cú pháp:


```
while condition:
    while-block
else:
    else-block
```
- else-block trong trường hợp while kết thúc 1 cách bình thường (tức là không phải dùng break để kết thúc).
- Nếu while kết thúc bình thường thì else-block sẽ tự động được thực hiện ngay sau đó.

23

23

4.5. while/else

- Ví dụ: Tính trung bình cộng của n số nguyên dương được nhập từ bàn phím


```

#Tính trung bình cộng của n số được nhập từ bàn
phím
d=0
s=0
n=int(input('Nhập n:'))
while d<n:
    di=float(input('Nhập số thứ {0}:'.format(d+1)))
    if di < 0:
        print('Nhập sai, thoát chương trình')
        break
    d+=1
    s+=di
else:
    print('Trung bình: {0}'.format(s/d))
  
```


Nhập n:3
 Nhập số thứ 1:5
 Nhập số thứ 2:4
 Nhập số thứ 3:10
 Trung bình: 6.333333333333333

24

24



4.6. for/else




- Cú pháp:


- else-block trong trường hợp while kết thúc 1 cách bình thường (tức là không phải dùng break để kết thúc).
- Nếu while kết thúc bình thường thì else-block sẽ tự động được thực hiện ngay sau đó.

25

25



4.5. for/else



- Cú pháp:


```
for expression:
    for-block
else:
    else-block
```


- else-block trong trường hợp for kết thúc 1 cách bình thường (tức là không phải dùng break để kết thúc).
- Nếu for kết thúc bình thường thì else-block sẽ tự động được thực hiện ngay sau đó.

26

26

4.6. for/else

■ Ví dụ: Tính trung bình cộng của n số nguyên dương được nhập từ bàn phím

```
#Tính trung bình cộng của n số được nhập từ bàn phím
d=0
s=0
n=int(input('Nhập n:'))
for d in range(0,n):
    di=float(input('Nhập số thứ {0}:'.format(d+1)))
    if di < 0:
        print('Nhập sai, thoát chương trình')
        break
    s+=di
else:
    print('Trung bình: %.2f'%(s/n))
```

Nhập n:3
 Nhập số thứ 1:1
 Nhập số thứ 2:8
 Nhập số thứ 3:20
 Trung bình: 9.67

Nhập n:3
 Nhập số thứ 1:10
 Nhập số thứ 2:-9
 Nhập sai, thoát chương trình

27

27

Bài tập

1. Tính $n! = 1 * 2 * 3 * \dots * n$, $0! = 1$
2. Tính tổng n số thực được nhập từ bàn phím.
3. In danh sách các số chẵn $< n$ không chia hết cho 12.
4. Nhập số nguyên dương $n \in [-m, m]$.

28

28

4.7. Vòng lặp lồng nhau

```

n=int(input('Nhập bậc ma trận:'))
print('Ma trận đơn vị cấp %s'%n)
for i in range(n):
    for j in range(n):
        if i==j:
            print('1', end=' ')
        else:
            print('0', end=' ')
    print()

```

Nhập bậc ma trận:4
Ma trận đơn vị cấp 4

```

1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1

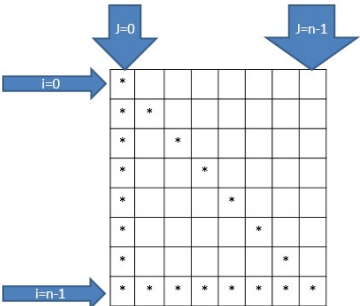
```

29

29

5. Bài tập

- Viết chương trình nhập hai số a, b và tính:
 - $(a + b)$,
 - a/b ,
 - a^b .
- Giải phương trình bậc 2: $ax^2 + bx + c = 0$
- Tính $n!$
- In * dạng tam giác dưới như hình bên.
- Đổi giờ - phút - giây:
 - Viết chương trình cho phép nhập vào số giây (ký hiệu là n) và sau đó quy đổi thời gian giây thành giờ, phút, giây. Xuất kết quả ra màn hình dưới dạng: giờ:phút:giây.
 - Ví dụ: người dùng nhập $n = 3770$ thì xuất ra màn hình 1:2:50.

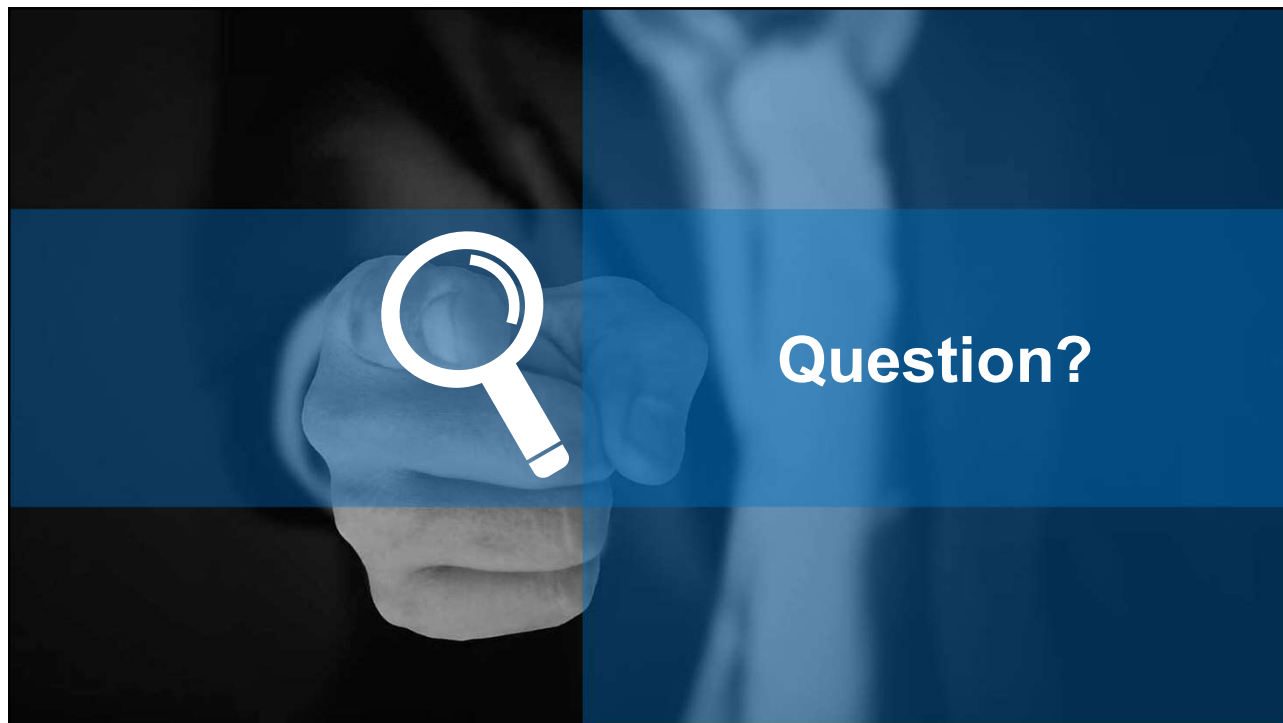


30

30



31



32