



# TIN HỌC CƠ SỞ

## Phần 2: Ngôn ngữ lập trình Python

### CHƯƠNG 8.

### ĐIỀU KHIỂN LƯỠNG VÀ VÒNG LẶP

**Tin cơ sở (LT):** 010100229802 - DHKL16A1HN, - DHKL16A2HN

GV. Cao Diệp Thắng

# Mục tiêu chương

## Nắm vững:

- ☐ Cách sử dụng câu lệnh đơn và câu lệnh ghép.
- ☐ Cú pháp lệnh gán, lệnh rẽ nhánh, lệnh điều khiển
- ☐ Cú pháp và cách sử dụng vòng lặp for, while với break, continue.
- ☐ Cách sử dụng lệnh pass



# Nội dung

---

- 1 Cấu trúc if điều kiện
- 2 Cấu trúc rẽ nhánh if ...else
- 3 Cấu trúc if ...elif...else
- 4 Cấu trúc lặp while
- 5 Cấu trúc lặp for

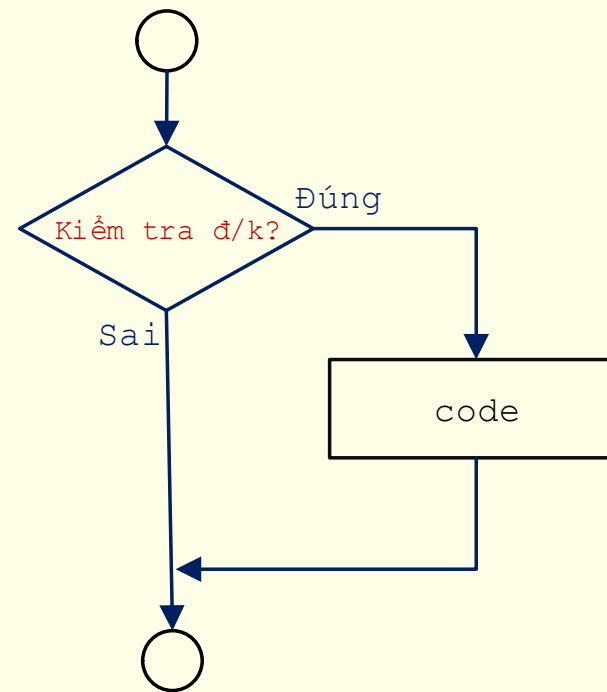


## 8.1. CẤU TRÚC IF

Cú pháp: **if** điều kiện : ← Bắt buộc phải có dấu “:”  
#khối lệnh

Ở đây, chương trình đánh giá **điều kiện** và sẽ thực hiện các **lệnh** khi **điều kiện** là **True**. Nếu **điều kiện False** thì **lệnh** sẽ không được thực hiện.

- ❑ Khối lệnh của lệnh **if** được viết **thụt lề** vào trong. Khối lệnh của **if** bắt đầu với một khoảng **thụt lề** và **dòng không thụt lề** đầu tiên sẽ được hiểu là kết thúc lệnh **if**.



Hình 8.1. Sơ đồ lệnh if



## 8.1. CẤU TRÚC IF,...

Ví dụ 8.1.

```
1  #Xét kết quả học tập như sau
2  diem_TB = eval(input("Nhập điểm trung bình: "))
3  if diem_TB >=5:
4      print("Sinh viên đỗ!!!")
5  -
```

```
Nhập điểm trung bình: 6.5
Sinh viên đỗ!!!
```



## 8.2. CẤU TRÚC Rẽ NHÁNH IF...ELSE

□ Được dùng để lựa chọn một trong hai nhánh thực hiện của chương trình.

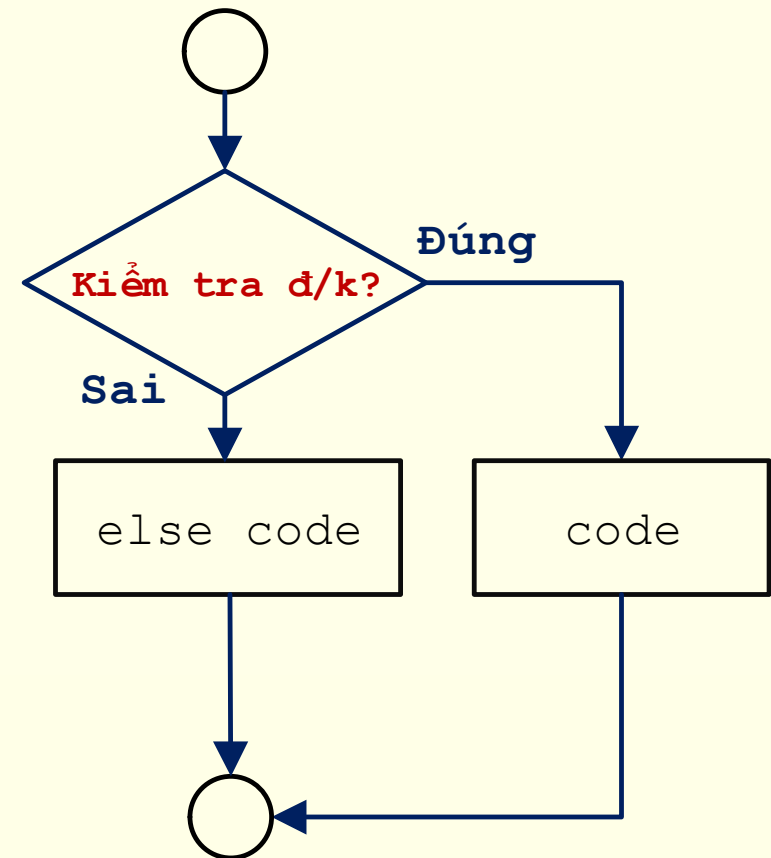
Cú pháp:

```
if điều kiện:  
    #khởi_lệnh_if
```

```
else:  
    #khởi_lệnh_else
```

Bắt buộc phải có dấu “:”

sau else  
bắt buộc  
phải có dấu  
“.”



Hình 8.2. Sơ đồ if ....else

## 8.2. CẤU TRÚC RỄ NHÁNH IF...ELSE,...

**Ví dụ 8.2:** Xét kết quả học tập như sau

Nếu điểm trung bình  $\geq 5$ : kết quả là “thi đỗ”, ngược lại kết quả là “thi trượt”

```
1  #Xét kết quả học tập sử dụng lệnh if...else
2  diem_TB = eval(input("Nhập điểm trung bình: "))
3  if diem_TB >=5:
4      print("Sinh viên đã thi đỗ!!!")
5  else:
6      print("Thi trượt")
```

Kết quả:

```
Nhập điểm trung bình: 4
Thi trượt
```



### 8.3. CẤU TRÚC if ...elif ...else

- Là các cấu lệnh if elif nối đuôi
- Các điều kiện sẽ được kiểm tra theo thứ tự từ đầu đến cuối. Nếu có một điều kiện thỏa mãn thì ngừng.
- Có thể lồng cấu trúc điều kiện với nhau

#### Cú pháp:

```
if điều_kiện_1:  
    #khởi_lệnh_if  
elif điều_kiện_2:  
    #khởi_lệnh_elif  
...  
elif điều_kiện_n:  
    #khởi_lệnh_elif  
else:  
    #khởi_lệnh_else
```

sau mỗi lệnh điều kiện phải có dấu ":"

sau else bắt buộc phải có dấu ":"





### Ví dụ 8.3:

Xếp loại sinh viên dựa trên điểm trung bình nhập vào

```
1  #Xét kết quả học tập sử dụng lệnh if...elif...else
2  diem_TB = eval(input("Nhập điểm trung bình: "))
3  if diem_TB >=0 and diem_TB <=10:
4      if diem_TB < 5:
5          print("Yếu/Kém!!!")
6      elif diem_TB <6:
7          print("Trung bình !!")
8      elif diem_TB<7:
9          print("Trung bình - Khá!")
10     elif diem_TB <8:
11         print("Khá!!!")
12     elif diem_TB <9:
13         print("Giỏi!!!")
14     else:
15         print("Xuất sắc !!!!!")
16 else:
17     print("Điểm nhập vào không hợp lệ !")
```

Thang điểm	Loại học tập
Điểm < 5	Yếu kém
5<=Điểm < =6	Trung bình
6<=Điểm < 7	Trung bình – Khá
7<=Điểm < 8	Khá
8<=Điểm < 9	Giỏi
Điểm >=9	Xuất sắc



Nhập điểm trung bình: 8.8  
Giỏi

## 8.4. CẤU TRÚC LẶP

Vòng lặp (tiếng anh là loop) là một thuật ngữ dùng để diễn tả một hành động hay một cụm hành động được lặp đi lặp lại nhiều lần.

Ví dụ: Xem xét, hoạt động bán vé là một chuỗi thao tác được lặp đi lặp lại ở quầy vé.

**Tình huống 1:** Nhân viên được giao bán hết 1000 vé thì nghỉ (Giả định một lần chỉ thực hiện bán 1 vé). Như vậy thao tác bán vé được thực hiện lặp đi lặp lại với số lần lặp biết trước là 1000 lần.

**Tình huống 2:** Nhân viên được giao bán vé từ 8 giờ sáng đến 17 giờ chiều (nghỉ trưa 1 giờ). Trong tình huống 2, số lần thao tác bán vé không xác định là bao nhiêu lần (số lần lặp không biết trước).

**Trong lập trình:** *Một cấu trúc lặp gồm một câu lệnh hay một khối lệnh sẽ thi hành lặp lại cho tới khi biểu thức điều kiện sai.*

Có hai loại cấu trúc lặp trong Python

- **while**
- **for**



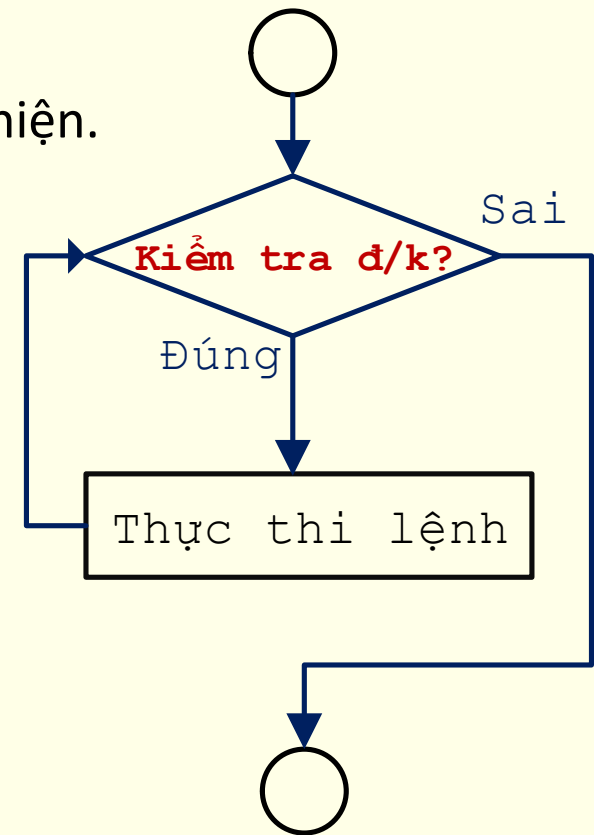
## 8.4.1. Cấu trúc lặp while

- Cấu trúc while dùng để thực hiện một lệnh hay một khối lệnh trong khi biểu thức điều kiện còn đúng.
- Điều kiện được kiểm tra trước khi các lệnh được thực hiện.

Cú pháp:

```
while điều kiện lặp :  
    #khối lệnh lặp
```

Bắt buộc phải có dấu ":"



Hình 8.3. Sơ đồ vòng lặp while

### Ví dụ 8.4. In ra các số từ 1 đến 10

```
i=1  
while i<=10 :  
    print("i=",  
        i)  
    i+=1
```

Kết quả:

```
i=1  
i=2  
i=3  
i=4  
i=5  
i=6  
i=7  
i=8  
i=9  
i=10
```



### Ví dụ 8.5. In ra bảng cửu chương 3

```
1 print("Chương trình in bang cuu chuong 3")
2 i=1
3 while i<=10:
4     print("3 x", i, " = ", 3*i)
5     i+=1
```

Kết quả:

```
Chương trình in bang cuu chuong 3
3 x 1  =  3
3 x 2  =  6
3 x 3  =  9
3 x 4  =  12
3 x 5  =  15
3 x 6  =  18
3 x 7  =  21
3 x 8  =  24
3 x 9  =  27
3 x 10 =  30
```



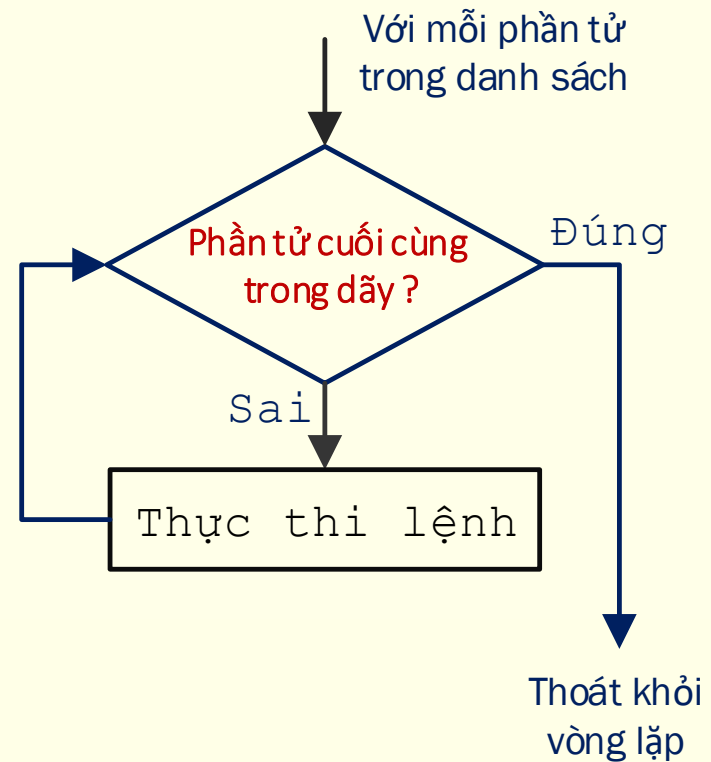
## 8.4.2. Cấu trúc lặp for

Cấu trúc lặp for được sử dụng để duyệt các phần tử trong một dãy, danh sách, ví dụ như list hoặc string

Cú pháp:

Bắt buộc phải có dấu ":"

```
for bien_lap in chuoi_lap:  
    khối lệnh của for
```



Hình 8.4. Sơ đồ khối cấu trúc lặp for

### Ví dụ 8.6.

```
1  for char in "UNETI":  
2      print("Character: " + char)
```

Kết quả:

Character: U

Character: N

Character: E

Character: T

Character: I



## 8.4. CẤU TRÚC LẶP,...

Ví dụ 8.7. In tổng  $S=1+2+3+4+5$

```
1  sum=0
2  numbers = [1,2,3,4,5]
3  for number in numbers:
4      |    sum+=number
5  print("Sum = ", sum)
```

Kết quả:

Sum = 15





### Ví dụ 8.8.

```
1 numbers=[1,2,3,4,5]
2 animals=["dog","cat", "elephant", "bird", "lion"]
3 for index in range(len(numbers)):
4     print("animal["+str(index)+ "] is "+ animals[index])
```

Kết quả:

```
animal[0] is dog
animal[1] is cat
animal[2] is elephant
animal[3] is bird
animal[4] is lion
```



## 8.4. CẤU TRÚC LẶP ,...

### Ví dụ 8.9.

```
1  animals=["dog","cat", "elephant", "bird", "lion"]
2  for animal in animals:
3      print("animal["+str(animals.index(animal))+ "] is "+ animal)
```

Kết quả:

```
animal[0] is dog
animal[1] is cat
animal[2] is elephant
animal[3] is bird
animal[4] is lion
```



## 8.5. HÀM range()

### 8.5.1. Khái niệm

- ❑ Hàm Range trong Python là một kiểu dữ liệu được tích hợp sẵn được sử dụng chủ yếu với các vòng lặp trong lập trình Python.
- ❑ Về cơ bản, hàm range được tích hợp sẵn trong Python, hàm này cho phép trả về một chuỗi số bắt đầu từ 0 và tăng dần lên 1 cho đến khi đạt được một số cụ thể. Nói cách khác, hàm này nhận một số nguyên và trả lại một range object kiểu (iterable). Hàm này thường được sử dụng để lặp lại kiểu trình tự, phổ biến nhất là for và loops.



## 8.5.2. Cách sử dụng hàm range trong python

Cú pháp: **range**(start, stop, step)

Tham số của hàm range (range parameters)

- **start** - Đây là tham số bắt đầu, chuỗi sẽ bắt đầu với tham số này
- **stop** - Là điểm kết thúc của dãy số, dãy số sẽ dừng ngay khi đạt đến tham số dừng
- **step** - Số nguyên xác định khoảng cách của mỗi số trong chuỗi

**Ví dụ 8.10:** tạo dãy số với hàm range(start=0, stop, step=1)

Sử dụng Python shell:

```
>>> print('Dang 1:', list(range(10)))    #start =0, stop=10, step =1
Dang 1: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> print('Dang 2:', list(range(1,10)))  #start =1, stop=10, step =1
Dang 2: [1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> print('Dang 3:', list(range(0,10,2)))#start =0, stop=10, step =2
Dang 3: [0, 2, 4, 6, 8]
>>>
>>> for i in range(0,10,2):
        print(i,end=', ')

0, 2, 4, 6, 8,
```



**Ví dụ 8.11.** Tạo danh sách số lẻ giữa tham số truyền vào bằng cách sử dụng **range()**

```
>>> start = 1
>>> stop = 10
>>> step = 2
>>> print(list(range(start, stop, step)))
[1, 3, 5, 7, 9]
```

**Ví dụ 8.12.** Tạo danh sách số chẵn giữa tham số truyền vào bằng cách sử dụng **range()**

```
>>> start=2
>>> stop=10
>>> step=2
>>> print(list(range(start, stop, step)))
[2, 4, 6, 8]
```



## 8.5.2. Cách sử dụng hàm range trong python,...

**Ví dụ 8.13.** range() hoạt động với step là số âm

```
>>> start=2
>>> stop=-14
>>> step=-2
>>> print(list(range(start,stop,step)))
[2, 0, -2, -4, -6, -8, -10, -12]
```



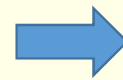
## 8.6. DÙNG ELSE VỚI VÒNG LẶP

Python hỗ trợ việc sử dụng **else** trong cấu trúc lặp

- ❑ Nếu **else** được sử dụng trong **for**, khối lệnh **else** sẽ được thực hiện sau khi for đã duyệt xong danh sách.
- ❑ Nếu **else** được sử dụng trong **while**, khối lệnh **else** sẽ được thực hiện khi điều kiện lặp trở thành **False**.

**Ví dụ 8.14.** In ra màn hình các số chẵn trong phạm vi 10 số tự nhiên đầu tiên, khi hoàn thành vòng lặp in ra thông báo “Finish loop!”

```
1  for item in range(1,10):  
2      |  if item%2==0:  
3      |      |  print("item", item)  
4  else:  
5      |  print("Finish Loop !")
```

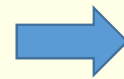


```
item 2  
item 4  
item 6  
item 8  
Finish Loop !
```

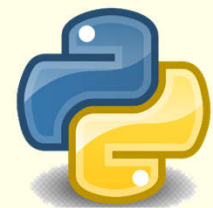


**Ví dụ 8.15.** In ra màn hình các giá trị nguyên  $\leq 5$  và giá trị nguyên đầu tiên lớn hơn 5.

```
1  count = 1
2  while count <=5:
3      print(count, "<= 5")
4      count +=1
5  else:
6      print(count, "> 5")
```



```
1 <= 5
2 <= 5
3 <= 5
4 <= 5
5 <= 5
6 > 5
```





## 8.7. SỬ DỤNG BREAK, CONTINUE, PASS VỚI VÒNG LẶP

### 8.7.1. Lệnh break

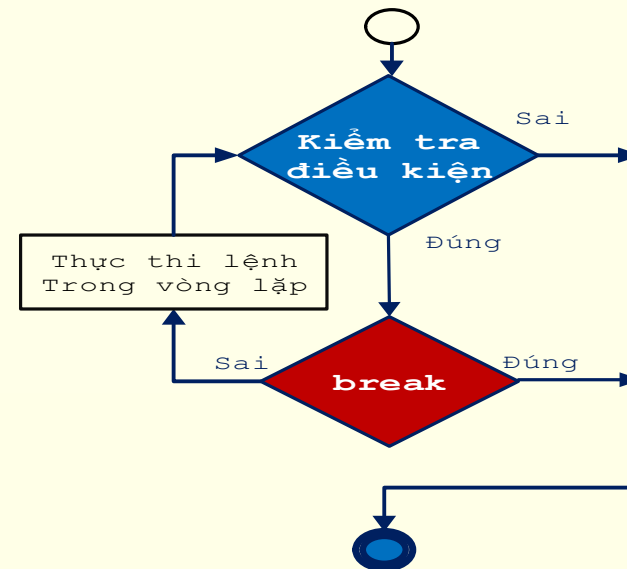
- ❑ **break** trong Python là giống như lệnh break trong C. Lệnh này kết thúc vòng lặp hiện tại và truyền điều khiển tới cuối vòng lặp.
- ❑ **break** có thể được sử dụng trong vòng lặp while và vòng lặp for. Nếu chúng ta đang sử dụng lồng vòng lặp, thì lệnh break kết thúc sự thực thi của vòng lặp bên trong và bắt đầu thực thi dòng code tiếp theo của khối. Thoát khỏi vòng lặp khi thỏa một điều kiện.

#### Ví dụ 8.16.

```
1 for ch in "Hello Python":
2     if ch == 'y':
3         break
4     print("Ký tự hiện thời: ",ch)
```

kết quả :

```
Ký tự hiện thời: H
Ký tự hiện thời: e
Ký tự hiện thời: l
Ký tự hiện thời: l
Ký tự hiện thời: o
Ký tự hiện thời: 
Ký tự hiện thời: P
```



Hình 8.5. Sơ đồ khối vòng lặp và lệnh break



**Ví dụ 8.17** Cho giá trị `var = 10`; in ra màn hình các số chẵn lớn hơn 5 và  $\leq 10$ . , sau đó in ra màn hình thông báo "Kết thúc lặp !"

```
1  var = 10
2  while var > 0:
3      if var %2 == 0:
4          print("Giá trị : ", var)
5          var -=1
6          if var ==5:
7              break
8  print("Kết thúc vòng lặp !")
```

kết quả

```
Giá trị : 10
Giá trị : 8
Giá trị : 6
Kết thúc vòng lặp !
```



### 8.7.2. Lệnh continue

- Bỏ qua lần lặp hiện hành và quay về đầu vòng lặp .
- Kiểm tra lại điều kiện, nếu thỏa mãn thì tiếp tục lặp,
- Nếu không thỏa điều kiện thì thoát.

#### Ví dụ 8.18

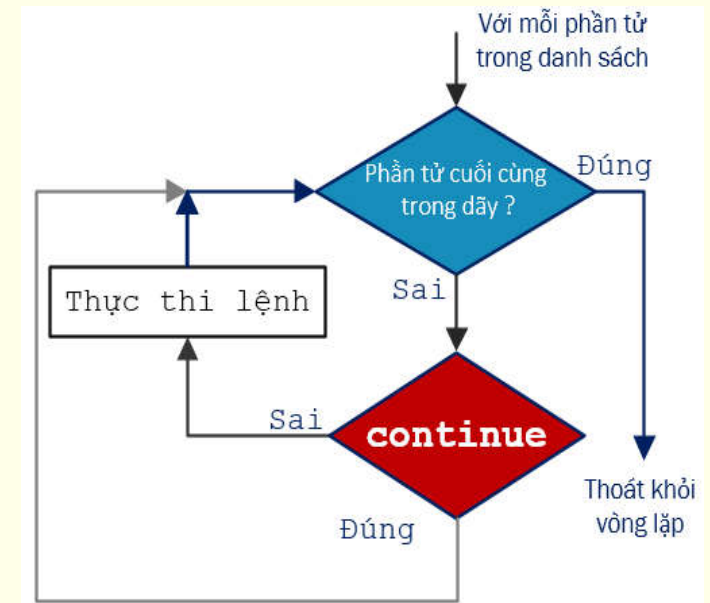
```
1 for ch in "Hello Python":
2     if ch == 'y':
3         continue
4     print("Ký tự hiện thời: ",ch)
```

Ký tự hiện thời: H  
Ký tự hiện thời: e  
Ký tự hiện thời: l  
Ký tự hiện thời: l  
Ký tự hiện thời: o  
Ký tự hiện thời: P  
Ký tự hiện thời: t  
Ký tự hiện thời: h  
Ký tự hiện thời: o  
Ký tự hiện thời: n

#### Ví dụ 8.19.

```
1 var = 10
2 while var > 0:
3     if var == 6:
4         continue
5     if var % 2 == 0:
6         print("Giá trị : ", var)
7     var -= 1
```

Giá trị : 10  
Giá trị : 8



Hình 8.6. Sơ đồ khối vòng for và continue



### 8.7.3. pass

- ❑ Lệnh **pass** trong Python, dùng như một **placeholder** (trình giữ chỗ) cho việc thực thi các hàm, vòng lặp,... trong tương lai.
- ❑ Giả sử, lập trình viên có một vòng lặp, hoặc một hàm, nhưng chưa biết nên xây dựng nó như thế nào, chưa biết nên code sao cho tối ưu và muốn để lại làm sau. Nhưng hàm, lệnh đó không thể có một khối lệnh rỗng, trình biên dịch sẽ báo lỗi, vì thế, chỉ cần sử dụng lệnh **pass** để xây dựng một khối lệnh rỗng, lúc này trình biên dịch sẽ hiểu và không ***“báo lỗi/thắc mắc”*** gì với người lập trình.

#### **pass**

- Được sử dụng khi một câu lệnh được yêu cầu nhưng ta không muốn bất cứ lệnh hoặc code nào được thực hiện.
- Là một toán tử **null**, không có gì xảy ra khi nó được thực hiện.
- Được sử dụng hữu ích cho những nơi code sẽ đi qua nhưng chưa được viết.



### 8.7.3. pass,...

#### Ví dụ 8.20

*# pass chỉ giữ chỗ cho for*

```
1  for ch in "Hello Python":
2      if ch == 'y':
3          pass
4          print("Pass block")
5      print("Ký tự hiện thời: ", ch)
6  print("Kết thúc vòng lặp for !")
```



```
Ký tự hiện thời: H
Ký tự hiện thời: e
Ký tự hiện thời: l
Ký tự hiện thời: l
Ký tự hiện thời: o
Ký tự hiện thời: 
Ký tự hiện thời: P
Pass block
Ký tự hiện thời: y
Ký tự hiện thời: t
Ký tự hiện thời: h
Ký tự hiện thời: o
Ký tự hiện thời: n
Kết thúc vòng lặp for !
```



# Câu hỏi củng cố kiến thức

1. Câu lệnh if trong chương trình Python có dạng:

- A.** if
- B.** if <điều kiện>
- C.** <điều kiện>:
- D.** if <điều kiện>:

## Câu hỏi củng cố kiến thức

2. Trong quá trình thực hiện thuật toán, khi nào cần dùng cấu trúc rẽ nhánh?

**B.** Khi có các phép tính toán.

**A.** Khi phải dựa trên một điều kiện cụ thể nào đó để xác định bước thực hiện tiếp theo.

**C.** Khi lặp đi lặp lại một công việc nào đó.

**D.** Khi sử dụng các hàm toán học.

# Câu hỏi củng cố kiến thức

3. Câu lệnh rẽ nhánh if-else trong chương trình Python có dạng:

- A.** `if <điều kiện>:`  
    `-----<khối lệnh>`  
`else`  
    `-----<khối lệnh>`
- B.** `if <điều kiện>:`  
    `else`
- C.** `if <điều kiện>:`  
    `-----<khối lệnh>`  
    `else:`  
    `-----<khối lệnh>`
- D.** `if`  
    `else:`



## Câu hỏi củng cố kiến thức

4. Trong Python, đối với cấu trúc if hoặc if-else thì câu lệnh hoặc nhóm câu lệnh phải được viết như thế nào?

- A.** Viết thẳng hàng so với điều kiện.
- B.** Lùi vào trong một số vị trí so với dòng chứa điều kiện và không cần viết thẳng hàng với nhau.
- C.** Chỉ lùi vào trong một số vị trí so với dòng chứa điều kiện khi có nhóm lệnh.
- D.** Lùi vào trong một số vị trí so với dòng chứa điều kiện và viết thẳng hàng với nhau.

## Câu hỏi củng cố kiến thức

5. Cho đoạn chương trình sau:

```
a=2
b=3
if a>b:
    a=a*2
else:
    b=b*2
```

Sau khi thực hiện đoạn chương trình trên giá trị của b là:

- A. 4**
- B. 2**
- C. 6**
- D. Không xác định**

# Câu hỏi củng cố kiến thức

6. Cấu trúc lặp với số lần biết trước là:

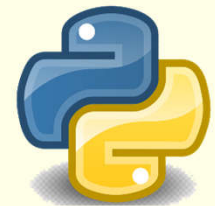
- A.** `for <Biến chạy> in range(m,n)`  
    `<Khối lệnh cần lặp>`
- B.** `while <Điều kiện>:`  
    `<Câu lệnh hay khối lệnh>`
- C.** `for <Biến chạy> in range(m,n):`
- D.** `for <Biến chạy> in range(m,n) :`  
    `<Khối lệnh cần lặp>`



# Câu hỏi củng cố kiến thức

7. Cấu trúc lặp với số lần không biết trước là:

- A.** `for <Biến chạy> in range(m,n)`  
    <Khối lệnh cần lặp>
- B.** `while <Điều kiện>:`  
    <Câu lệnh hay khối lệnh>
- C.** `while <Điều kiện>:`
- D.** `for <Biến chạy> in range(m,n) :`  
    <Khối lệnh cần lặp>



# Câu hỏi củng cố kiến thức

8 . Cho đoạn lệnh sau:

```
for i in range(5):  
    print(i)
```

Trên màn hình i có các giá trị là:

**A.** 0 1 2 3 4 5

**B.** 1 2 3 4 5

**C.** 0 1 2 3 4

**D.** 1 2 3 4



## Câu hỏi củng cố kiến thức

9. Cho đoạn chương trình sau:

```
s=0  
for i in range(6):  
    s=s+i
```

Sau khi thực hiện đoạn chương trình trên giá trị của s là:

- A. 1**
- B. 15**
- C. 6**
- D. 21**



## Câu hỏi củng cố kiến thức

10. Cho đoạn chương trình sau:

```
s=0
i=1
while i<=5:
    s=s+1
    i=i+1
```

Sau khi thực hiện đoạn chương trình trên giá trị của s là:

- A. 9**
- B. 15**
- C. 5**
- D. 10**



## Câu hỏi củng cố kiến thức

11. Cho đoạn chương trình sau:

```
i=1  
s=0  
while <điều kiện>:  
    s=s+i  
    i=i+1
```

Đoạn chương trình trên tính tổng  $s=1+2+3+\dots+10$ , <điều kiện> là:

- A.  $i \leq 10$
- B.  $i == 10$
- C.  $i \geq 10$
- D.  $i > 10$





## Câu hỏi củng cố kiến thức

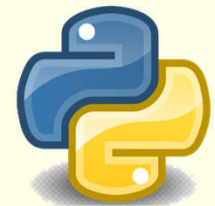
12. Phát biểu nào sau đây là **sai**?

- A.** Dùng câu lệnh while ta cũng thể hiện được cấu trúc lặp với số lần biết trước.
- B.** Dùng câu lệnh for ta cũng thể hiện được cấu trúc lặp với số lần không biết trước.
- C.** Trong Python có 2 dạng lặp: Lặp với số lần biết trước và lặp với số lần không biết trước.
- D.** Câu lệnh thể hiện lặp với số lần không biết trước phải sử dụng một biểu thức logic làm điều kiện lặp.



## Câu hỏi thảo luận

1. Trình bày cú pháp và công dụng của lệnh **if**, **if...else**, **if....elif...else**.
2. Trình bày thế nào và câu lệnh đơn, câu lệnh ghép cho 2 ví dụ minh họa.
3. So sánh lưu đồ của các dạng lệnh if , if ....else và if...elif...else.
4. Sự khác biệt giữa vòng lặp for và wihle
5. Cú pháp và chức năng các lệnh break, continue, pass?



# **Bài tập vận dụng**

Các bài tập trong TLHT chương 8.