

Bài giảng môn học:

Khoa Học Dữ Liệu (7080509)

CHƯƠNG 3: Lập trình Python cơ bản (Phần 2)

FIT.HUMG

Nội dung bài học:

1. Các toán tử trong Python

Toán tử số học | Toán tử gán | Toán tử so sánh | Toán tử logic | Toán tử membership

2. Cấu trúc điều khiển

Dạng 1, dạng 2, dạng 3, dạng 4

3. Cấu trúc vòng lặp

Vòng lặp While | Vòng lặp for | Break, continue

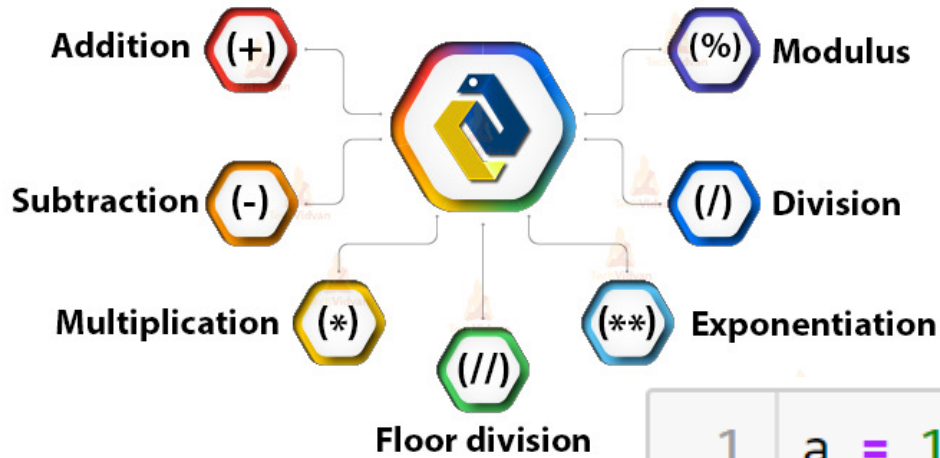
4. Ngoại lệ (SV tham khảo thêm)



1. Các toán tử trong Python

Các toán tử số học

Python Arithmetic Operators



```
1  a = 10
2  b = 8
3  #-----
4  tong = a + b           # Tổng của hai số (+)
5  hieu = a - b           # Hiệu của hai số (-)
6  tich = a*b             # Tích của hai số (*)
7  thuong = a/b           # Thương của hai số (/)
8  thuong_nguyen = a//b   # Phép chia lấy phần nguyên (//)
9  thuong_du = a % b      # Phép chia lấy phần dư (%)
10 mu = a**b              # Tính giá trị a lũy thừa b (**)
```

Các toán tử gán

Toán tử	Mô tả	Ví dụ
=	Toán tử này dùng để gán giá trị của một đối tượng cho một giá trị	$c = a$ (lúc này c sẽ có giá trị $= a$)
+=	Toán tử này cộng rồi gán giá trị cho đối tượng	$c += a$ (tương đương với $c = c + a$)
-=	Toán tử này trừ rồi gán giá trị cho đối tượng	$c -= a$ (tương đương với $c = c - a$)
*=	Toán tử này nhân rồi gán giá trị cho đối tượng	$c *= a$ (tương đương với $c = c * a$)
/=	Toán tử này chia rồi gán giá trị cho đối tượng	$c /= a$ (tương đương với $c = c / a$)
%	Toán tử này chia hết rồi gán giá trị cho đối tượng	$c \% = a$ (tương đương với $c = c \% a$)
**=	Toán tử này lũy thừa rồi gán giá trị cho đối tượng	$c ** = a$ (tương đương với $c = c ** a$)
//=	Toán tử này chia làm tròn rồi GÁN giá trị cho đối tượng	$c //= a$ (tương đương với $c = c // a$)

Các toán tử so sánh

Toán tử	Mô tả	Ví Dụ (a =8, b=10)
==	So sánh giá trị của các đối số xem có bằng nhau hay không. Nếu bằng nhau thì kết quả trả về sẽ là True và ngược lại sẽ là False.	a == b // False
!=	So sánh giá trị của các đối số xem có khác nhau hay không. Nếu khác nhau thì kết quả trả về sẽ là True và ngược lại sẽ là False.	a != b //True
<	Dấu < đại diện cho phép toán nhỏ hơn, nếu đối số 1 nhỏ hơn đối số 2 thì kết quả sẽ trả về là True và ngược lại sẽ là False.	a < b //True
>	Dấu > đại diện cho phép toán lớn hơn, nếu đối số 1 lớn hơn đối số 2 thì kết quả sẽ trả về là True và ngược lại sẽ là False.	a > b //False
<=	Dấu > đại diện cho phép toán nhỏ hơn hoặc bằng, nếu đối số 1 nhỏ hơn hoặc bằng đối số 2 thì kết quả sẽ trả về là True và ngược lại sẽ là False.	a <= b //True
>=	Dấu > đại diện cho phép toán lớn hơn hoặc bằng, nếu đối số 1 lớn hơn hoặc bằng đối số 2 thì kết quả sẽ trả về là True và ngược lại sẽ là False.	a>= b //False

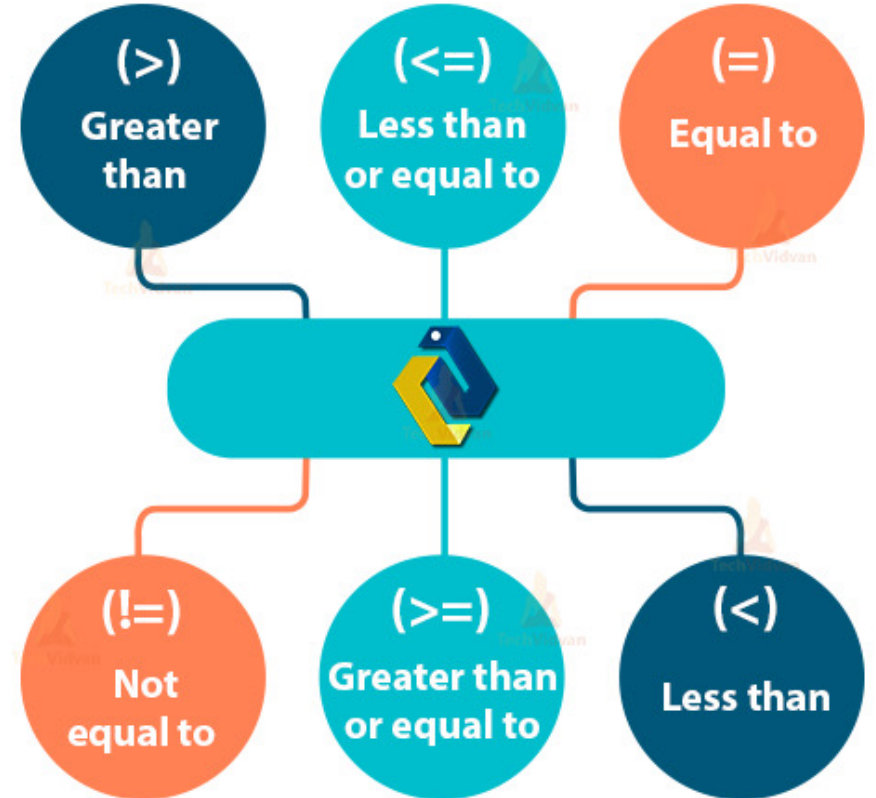
Các toán tử so sánh (t)

```
1  #Kết quả của các phép so sánh có kiểu dữ liệu Boolean
2  a = 8
3  b = 10
4  kt = a>b
5  print(type(kt))
6  #-----
7  print('1) SS lớn hơn (a>b):', a>b)
8  print('2) SS nhỏ hơn (a<b):', a<b)
9  print('3) SS bằng (a=b):', a==b)
10 print('4) SS lớn hơn hoặc bằng (a>=b):', a>=b)
11 print('5) SS nhỏ hơn hoặc bằng (a<=b):', a<=b)
12 print('6) SS khác (a!=b):', a!=b)
```

```
<class 'bool'>
```

```
1) SS lớn hơn (a>b): False
2) SS nhỏ hơn (a<b): True
3) SS bằng (a=b): False
4) SS lớn hơn hoặc bằng (a>=b): False
5) SS nhỏ hơn hoặc bằng (a<=b): True
6) SS khác (a!=b): True
```

PYTHON RELATIONAL OPERATORS



Học viên nhập code và đọc kết quả của các phép so sánh ở trên!

Các toán tử logic, xác thực

Các toán tử logic

Toán tử	Mô tả	Ví dụ
and	Nếu 2 vế của toán tử này đều là True thì kết quả sẽ là True và ngược lại nếu 1 trong 2 vế là False thì kết quả trả về sẽ là False.	$x < 5$ and $x < 10$
or	Nếu 1 trong 2 vế là True thì kết quả trả về sẽ là True và ngược lại nếu cả 2 vế là False thì kết quả trả về sẽ là False.	$x < 5$ or $x < 4$
not	Đây là dạng phủ định, nếu biểu thức là True thì nó sẽ trả về là False và ngược lại.	$\text{not}(x < 5 \text{ and } x < 10)$

Các toán tử xác thực

Toán tử	Mô tả	Ví dụ: $a=4$, $b=5$
is	Toán tử này sẽ trả về True nếu $a == b$ và ngược lại	$a \text{ is } b$ //False
is not	Toán tử này sẽ trả về True nếu $a != b$ và ngược lại	$a \text{ is not } b$ //True

Toán tử membership

Toán tử	Mô tả	Ví dụ
in	Nếu 1 đối số thuộc một tập đối số nó sẽ trả về True và ngược lại.	a in b
not in	Nếu 1 đối số không thuộc một tập đối số nó sẽ trả về True và ngược lại.	a not in b

```
1  #D/ Các toán tử membership
2  a = [4, 6, 9 , 0]
3  kt1 = 4 in a          #Kiểm tra 1 phần tử có trong danh sách không?
4  kt2 = 0 not in a      #Kiểm tra 1 phần tử không có trong danh sách không?
5
6  print('Kết quả 1: ', kt1)
7  print('Kết quả 2:', kt2)
```

Kết quả 1: True

Kết quả 2: False

Thứ tự ưu tiên các phép toán

Python Operator Precedence

Precedence	Operator Sign	Operator Name
Highest	**	Exponentiation
	+X, -X, ~X	Unary positive, unary negative, bitwise negation
	*, /, //, %	Multiplication, division, floor, division, modulus
	+, -	Addition, subtraction
	<<, >>	Left-shift, right-shift
	&	Bitwise AND
	^	Bitwise XOR
		Bitwise OR
	==, !=, <, <=, >, >=, is, is not	Comparison, Identity
	not	Boolean NOT
	and	Boolean AND
Lowest	or	Boolean OR

2. Cấu trúc điều khiển (if)

Câu lệnh điều kiện

- Câu lệnh điều kiện là một trong những câu lệnh cơ bản của bất cứ ngôn ngữ lập trình nào.

Ngôn ngữ tự nhiên	Ngôn ngữ lập trình
Nếu Bạn đủ 18 tuổi thì <i>“Bạn được kết hôn”,</i> Ngược lại thì <i>“Bạn chưa được kết hôn”</i>	if (age > = 18): <i>print(‘Bạn được kết hôn!’)</i> else: <i>print(‘Bạn chưa được kết hôn!’)</i>

```
1  #Câu Lệnh điều kiện
2  so_tien = input ('Nhập vào số tiền bạn có: ')
3  so_tien = int(so_tien)
4  if (so_tien >= 1000000000):
5      print('Bạn đã là một tỷ phú!')
6  else:
7      print('Bạn còn phải kiếm nhiều tiền hơn!')
```

Các dạng câu lệnh điều kiện

- Dạng 1:

if (điều kiện1):

Nhóm lệnh 1

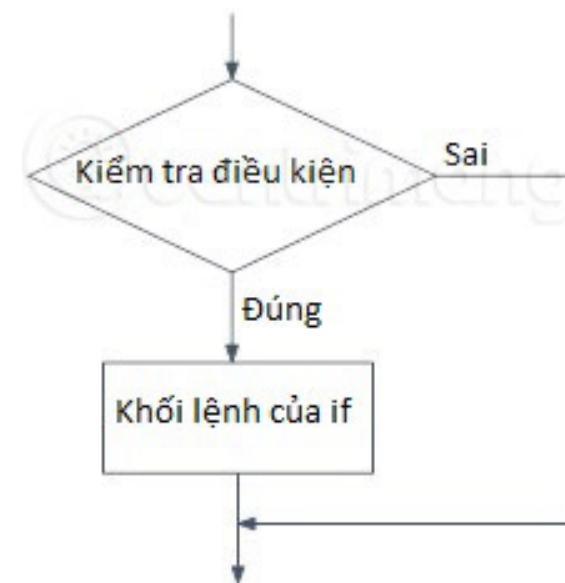
(Nếu điều kiện1 đúng thì thực hiện nhóm lệnh 1)

```
In [2]: num = 3
        if num > 0:
            print(num, "là số dương.")
        print("Thông điệp này luôn được in.")
```

```
3 là số dương.
Thông điệp này luôn được in.
```

```
In [3]: num = -1
        if num > 0:
            print(num, "là số dương.")
        print("Thông điệp này luôn được in.")
```

```
Thông điệp này luôn được in.
```



Ví dụ: Kiểm tra số chẵn

Nhập vào 1 số nguyên N, kiểm tra nếu N là số chẵn hiển thị thông báo.

- **Yêu cầu:**
Nhập vào một số: 12
- **Kiểm tra và hiển thị thông báo:**
Số 12 là số chẵn!

The number 12 is displayed in white inside a green rounded square.

Odd

The number 13 is displayed in black inside an orange circle.

Even

Các dạng câu lệnh điều kiện (t)

- Dạng 2:

if (điều kiện1):

Nhóm lệnh 1

else:

Nhóm lệnh 2

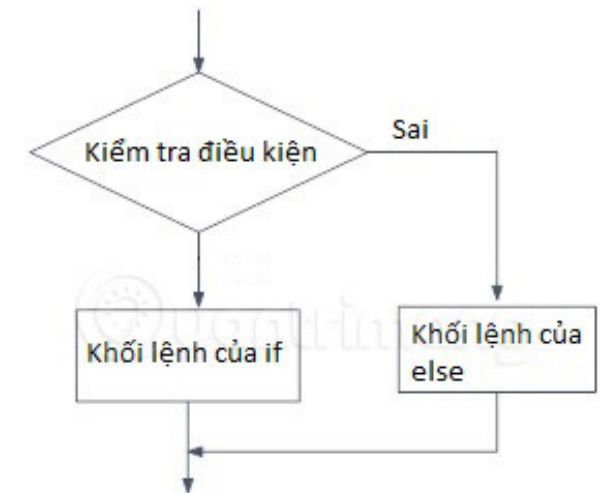
(Nếu điều kiện1 đúng thì thực hiện nhóm lệnh 1, nếu sai thực hiện nhóm lệnh 2)

```
In [5]: num = 3
        if num >= 0:
            print("Số dương")
        else:
            print("Số âm")
```

Số dương

```
In [6]: num = -1
        if num >= 0:
            print("Số dương")
        else:
            print("Số âm")
```

Số âm



Ví dụ: Kiểm tra số chẵn – lẻ

Nhập vào 1 số nguyên N, kiểm tra nếu N là số chẵn hiển thị thông báo “ Đây là số chẵn!”, ngược lại thông báo “Đây là số lẻ!”



Odd

Nhập vào một số:12
Đây là số chẵn!



Even

Nhập vào một số:13
Đây là số lẻ!

Các dạng câu lệnh điều kiện (t)

- Dạng 3 (if lồng nhau):

if (điều kiện1):

Nhóm lệnh 1

elif (điều kiện 2):

Nhóm lệnh 2

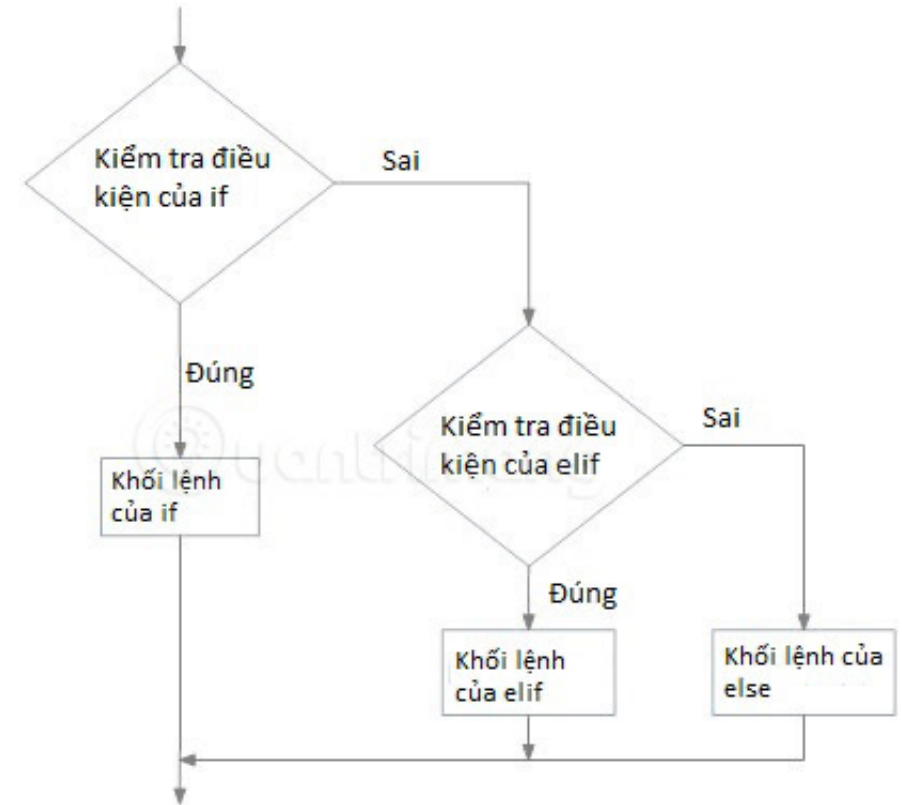
...

elif (điều kiện n):

Nhóm lệnh n

else:

Nhóm lệnh x

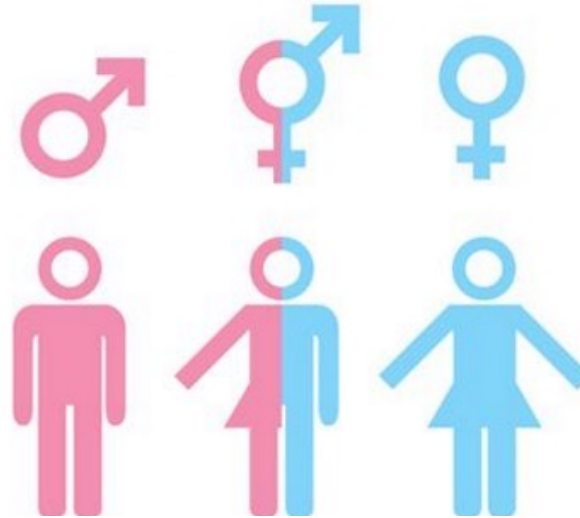


(Kiểm tra điều kiện1 đúng thì thực hiện nhóm lệnh 1, nếu sai kiểm tra điều kiện 2 đúng thực hiện nhóm lệnh 2 ...nếu tất cả các điều kiện sai thực hiện nhóm lệnh x)

Ví dụ: Chào theo giới tính

Nhập vào giới tính (0=Nam | 1=Nữ)

- Nếu nhập vào 0 → Chào anh đẹp trai!
- Nếu nhập vào 1 → Chào chị xinh gái!
- Nếu nhập khác 0 hoặc 1 → Cảnh báo: Giới tính không xác định!



Nhập giới tính (0:Nam - 1:Nữ):0
Chào anh đẹp trai!

Nhập giới tính (0:Nam - 1:Nữ):1
Chào chị xinh gái!

Nhập giới tính (0:Nam - 1:Nữ):2
Cảnh báo: Giới tính không xác định!

Các dạng câu lệnh điều kiện (t)

- Dạng 4 (if lồng nhau – Nested if):

if (điều kiện1):

 if (điều kiện 1.1):

Nhóm lệnh 1.1

 elif (điều kiện 1.2):

Nhóm lệnh 1.2

 else:

Nhóm lệnh 1.x

else:

nhóm lệnh 2

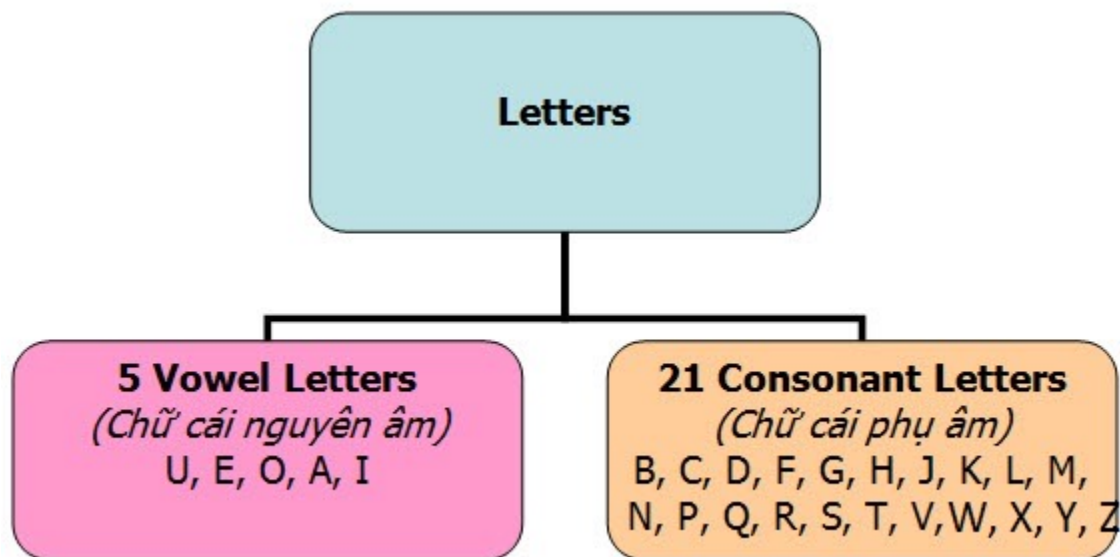
```
num = float(input("Nhập một số: "))
if num >= 0:
    if num == 0:
        print("Số Không")
    else:
        print("Số dương")
else:
    print("Số âm")
```

Nhập một số: 0
Số Không

Thực hành

Bài 6: Xác định nguyên âm | Phụ âm

Nhập vào một ký tự chữ cái bất kỳ(không phân biệt chữ hoa-chữ thường), cho biết nó là nguyên âm hay phụ âm trong tiếng anh.



Nhập vào một chữ cái:a
Đây là nguyên âm!

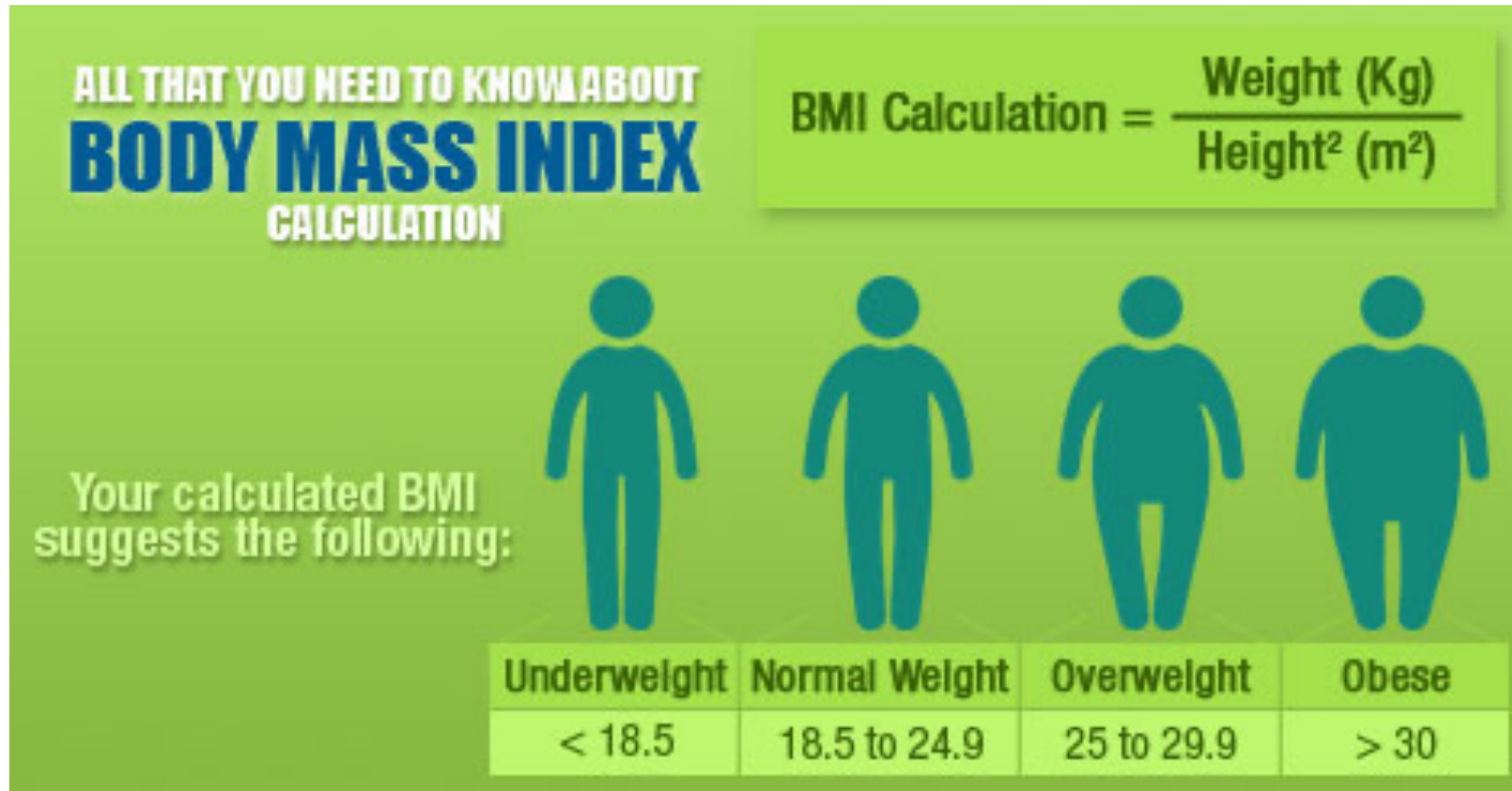
Nhập vào một chữ cái:E
Đây là nguyên âm!

Nhập vào một chữ cái:y
Đây là phụ âm!

Nhập vào một chữ cái:B
Đây là phụ âm!

Bài 7: Tính chỉ số BMI

Nhập vào chiều cao (m) và cân nặng (kg) và tính chỉ số BMI của người đó, Dựa vào chỉ số BMI tính được đưa ra nhận xét về cơ thể của người đó.



Bài 8: Xác định mùa trong năm

Nhập vào tháng sinh của bạn, và cho biết bạn sinh vào mùa nào biết:

- **Yêu cầu:**
 - Nếu 1, 2, 3: Bạn sinh vào Mùa xuân
 - Nếu 4, 5, 6: Bạn sinh vào Mùa hạ
 - Nếu 7, 8, 9: Bạn sinh vào Mùa thu
 - Nếu 10, 11, 12: Bạn sinh vào Mùa đông
 - <1 hoặc >12: Tháng sinh nhập vào không đúng



3. Cấu trúc vòng lặp trong Python

Câu lệnh vòng lặp

- Cũng như câu lệnh điều kiện, Câu lệnh vòng lặp là một trong những câu lệnh cơ bản của bất cứ ngôn ngữ lập trình nào.
- Để giải quyết bài toán, chúng ta cần thực hiện một công việc nào đó lặp đi lặp lại rất nhiều lần. Số lần lặp đó có thể biết trước hoặc không biết trước.

Ngôn ngữ tự nhiên	Ngôn ngữ lập trình
Tính tổng các số từ 1 đến 10: <i>1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10</i>	tong = 0 for i in range(1,11): <i>tong = tong + i</i> <i>print('Tổng từ 1 đến 10 là:', tong)</i>

Có 2 kiểu vòng lặp trong python:

- * Vòng lặp while
- * Vòng lặp for

Vòng lặp While

- Vòng lặp while sử dụng khi **không biết trước số lần lặp**.
- Cú pháp:

While <điều kiện>:
Nhóm lệnh 1

```
1 n = int(input('Em sinh tháng mấy?'))
2 i=1
3 while(i<=n):
4     print(i, ') I Love You!')
5     i=i+1
6
7 #Câu Lệnh Lặp ngoài vòng Lặp while
8 print('-----AIACADEMY-----')
```

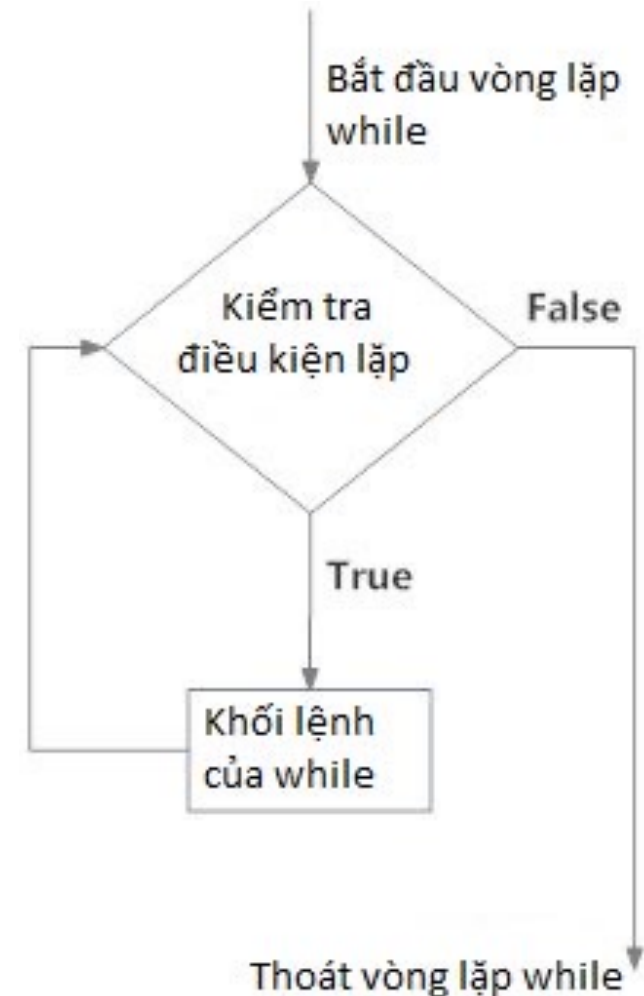
Em sinh tháng mấy?3

1) I Love You!

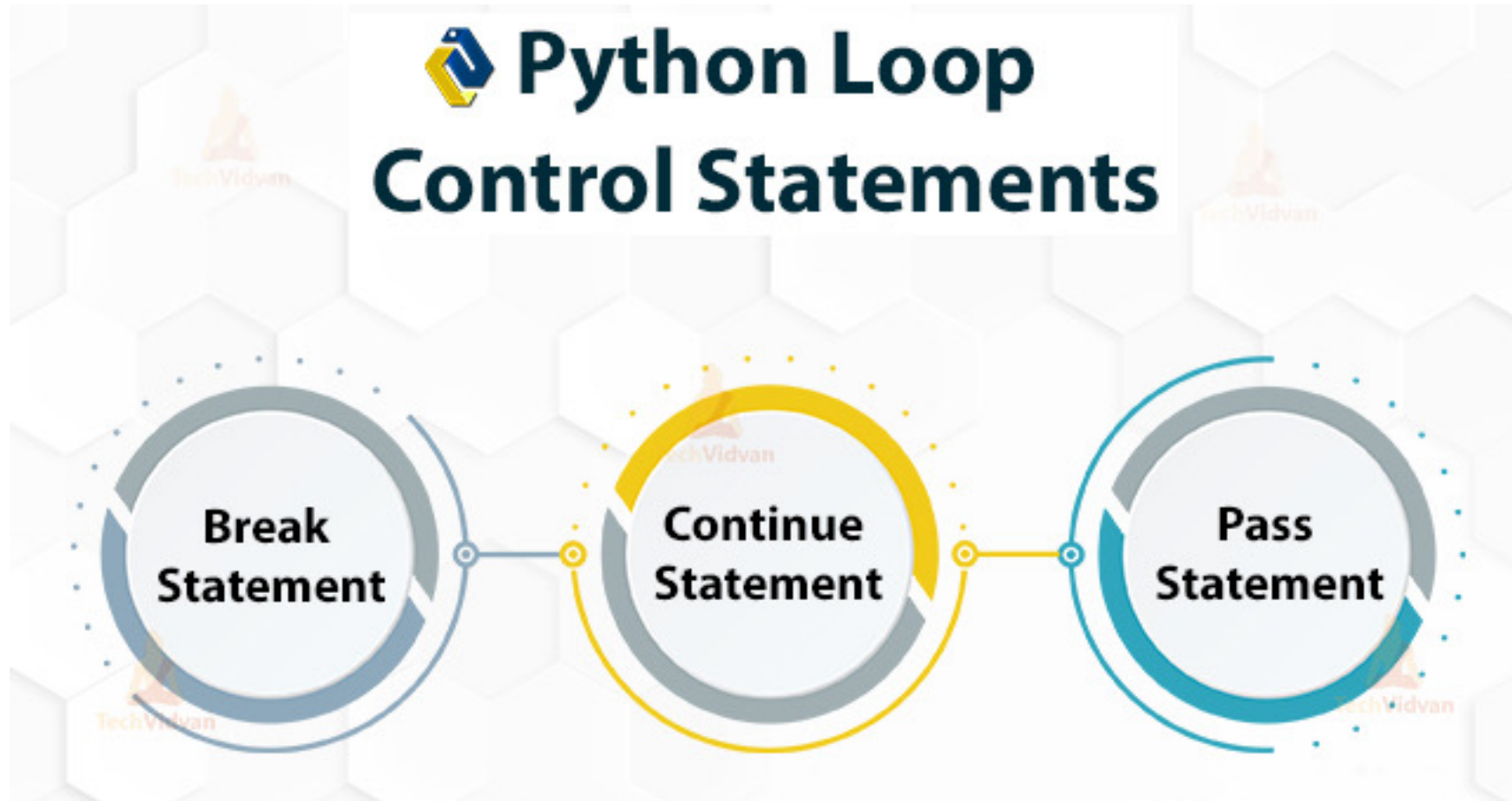
2) I Love You!

3) I Love You!

-----AIACADEMY-----



Kiểm soát vòng lặp



Lệnh break

- Lệnh **break kết thúc vòng lặp** chứa nó và truyền điều khiển đến lệnh tiếp theo sau khối lệnh của vòng lặp đó.

```
1 n = int(input('Em sinh tháng mấy? '))
2 i=1
3 while(i<=n):
4     print(i, ') I Love You!')
5
6     #Chỉ hiển thị tối đa 3 Lần
7     if (i==3):
8         break #Thoát ra khỏi vòng lặp while
9
10    i=i+1
11
12 #Câu Lệnh Lặp ngoài vòng Lặp while
13 print('-----AIACADEMY-----')
```

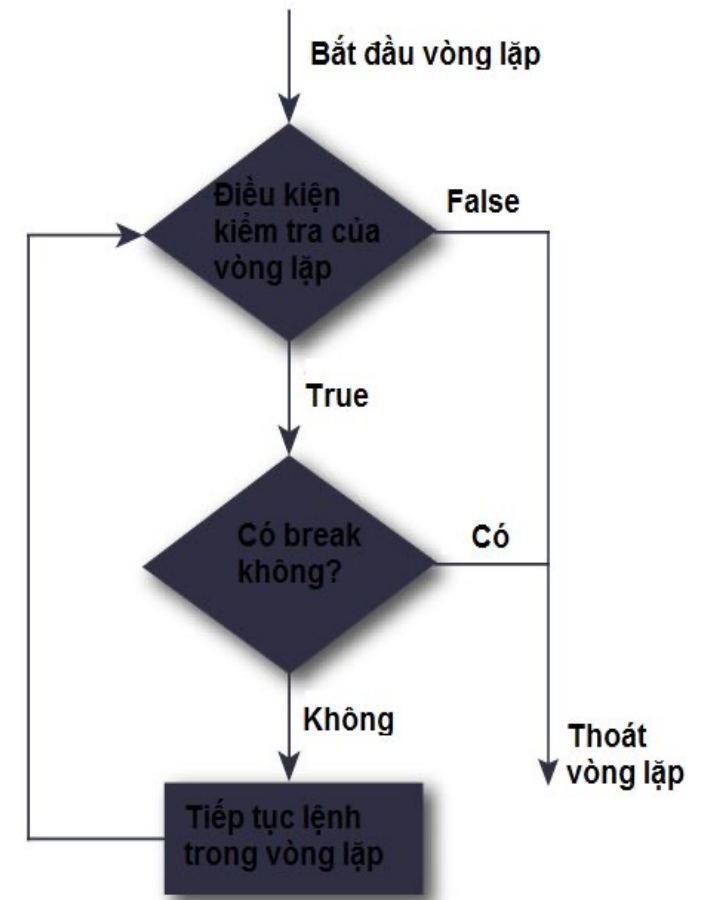
Em sinh tháng mấy? 8

1) I Love You!

2) I Love You!

3) I Love You!

-----AIACADEMY-----

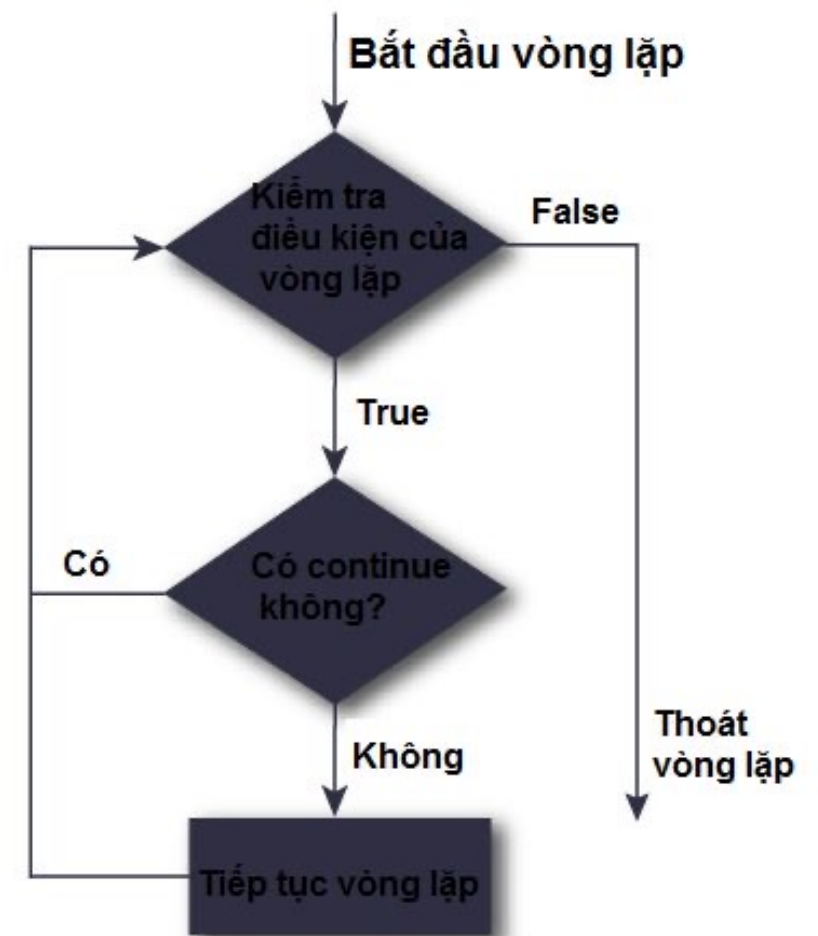


Lệnh continue

- Lệnh **continue** bỏ qua phần còn lại của khối lệnh bên trong vòng lặp, áp dụng cho **vòng lặp tiếp theo**. Nghĩa là vòng lặp không chấm dứt nó sẽ tiếp tục với số lần lặp kế tiếp

```
1 n = 20
2 i = 1
3 while (i<=n):
4     i = i+1
5     if (i%3!=0):
6         continue
7         #Bỏ qua các câu lệnh phía sau nếu ko chia hết cho 3
8     print(i)
9
10 #Câu Lệnh Lặp ngoài vòng Lặp while
11 print('-----AIACADEMY-----')
```

```
3
6
9
12
15
18
21
-----AIACADEMY-----
```



Lệnh while ...else

- Cấu trúc:

```
while <điều kiện>:  
    <Nhóm lệnh A>  
else:  
    <Nhóm lệnh B>
```

- Thực hiện nhóm lệnh A cho đến khi điều kiện còn đúng.
- Nếu điều kiện sai thực hiện nhóm lệnh B.
(ít nhất thực hiện nhóm lệnh B 1 lần)

```
1 counter = 4  
2 while counter < 3:  
3     print("Inside loop")  
4     counter = counter + 1  
5 else:  
6     print("Inside else")  
7 #-----  
8 print('----Bên ngoài vòng lặp while----')
```

```
Inside else  
----Bên ngoài vòng lặp while----
```

Lệnh while True

- Cấu trúc:

while True:

<Nhóm lệnh thực hiện khi điều kiện đúng>

if <điều kiện dừng>:

break

Với cấu trúc này các **câu lệnh sẽ thực hiện lặp đi lặp lại**, cho đến khi biểu thức <điều kiện dừng> thỏa mãn. Lúc đó câu lệnh **if** sẽ giúp cho lệnh **break** được thực thi và dừng vòng lặp.

```
1  #chỉ cho phép nhập tháng sinh 1 - 12
2  while True:
3      n = int(input('Em sinh tháng mấy? '))
4      if (1<= n <= 12):
5          #Tháng sinh nhập vào hợp lệ!
6          break;
7      print('Tháng không đúng, vui lòng nhập lại')
8  #Câu lệnh ngoài vòng lặp while
9  print('Chào em cô gái tháng ', n)
```

```
Em sinh tháng mấy? 15
Tháng không đúng, vui lòng nhập lại
Em sinh tháng mấy? 10
Chào em cô gái tháng 10
```

Vòng lặp for

- Vòng lặp for sử dụng khi **biết trước số lần lặp**.
- Cú pháp:

for <biến chạy> in <dãy>:

Nhóm lệnh 1

<Biến chạy> sẽ lần lượt nhận các giá trị của các thành phần có trong <dãy>. Dãy có thể là một danh sách (list), chuỗi ký tự (str), dãy số....

```
1  #Tích 10! = 1*2*3*4*5*6*7*8*9*10
2  #Tổng 10 = 1+2+3+4+5+6+7+8+9+10
3  n = 10
4  tích = 1
5  tong = 0
6  for i in range(1, n+1):
7      #Mỗi lần lặp biến i tăng lên 1
8      tích = tích*i
9      tong = tong+i
10
11  print('10! = ', tích)
12  print('10+ = ', tong)
```

```
10! = 3628800
10+ = 55
```


Vòng lặp for (2)

- **Vòng lặp for với chuỗi ký tự:** Biến chạy sẽ lần lượt nhận các giá trị là các ký tự trong chuỗi ký tự.

```
1  #Vòng lặp for với chuỗi ký tự:  
2  st = 'HUMG IN MY MIND'  
3  for i in st:  
4      print('ký tự: ', i)
```

1

ký tự: H
ký tự: U
ký tự: M
ký tự: G
ký tự: I
ký tự: N
ký tự: M
ký tự: Y
ký tự: M
ký tự: I
ký tự: N
ký tự: D

```
1  #Đếm số ký tự M trong chuỗi  
2  st = 'HUMG IN MY MIND'  
3  dem = 0  
4  for i in st:  
5      if (i=='M'): dem=dem+1  
6  print('Số ký tự M có trong chuỗi là: ', dem)
```

2

Số ký tự M có trong chuỗi là: 3

Vòng lặp for (3)

- **Vòng lặp for với danh sách:** Biến chạy sẽ lần lượt nhận các giá trị là các phần tử trong danh sách.

```
1  #Vòng lặp for với danh sách
2  hoc_sinh = ['Lê Thùy Dung', 'Trần Đức Hùng',
3              'Nguyễn Lan Anh', 'Mai Phương Thúy',
4              'Trần Thanh Thủy', 'Kiều Thành Công']
5
6  print('Danh sách học sinh bao gồm:')
7  tt = 1
8  for i in hoc_sinh:
9      print( tt, ') ', i)
10     tt = tt+1
```

Danh sách học sinh bao gồm:

- 1) Lê Thùy Dung
- 2) Trần Đức Hùng
- 3) Nguyễn Lan Anh
- 4) Mai Phương Thúy
- 5) Trần Thanh Thủy
- 6) Kiều Thành Công

Vòng lặp for (4)

- **Vòng lặp for với lệnh range():** Lệnh range() trong Python kết hợp với vòng lặp for sẽ trở nên rất hữu hiệu trong việc kiểm soát giá trị bắt đầu, kết thúc và bước nhảy của biến chạy.
- Cú pháp:

```
for <biếnchạy> in range(<bắtđầu>,<kếtthúc>,<bướcnhảy>) :  
    nhóm lệnh thực hiện
```

- **<bắt đầu>** là giá trị khởi gán ban đầu cho biến chạy (mặc định = 0)
- **<kết thúc>** là giá trị kết thúc cho biến chạy, nhưng không bao gồm chính nó (< kết thúc)
- **<bước nhảy>** là giá trị mà biến nhảy tăng thêm sau mỗi lần lặp (mặc định = 1)

Vòng lặp for (4)

- Vòng lặp for với lệnh range():

```
1 #Lệnh for với range()
2 for i in range(5):
3     #Giá trị khởi đầu mặc định = 0
4     #Bước nhảy mặc định = 1
5     print('i = ',i)
```

1

```
i = 0
i = 1
i = 2
i = 3
i = 4
```

```
1 #Lệnh for với range(m,n,d)
2 for i in range(2,11,2):
3     #Giá trị khởi đầu m = 2
4     #Giá trị kết thúc n = 11
5     #Bước nhảy d = 2
6     print('i = ',i)
```

3

```
i = 2
i = 4
i = 6
i = 8
i = 10
```

```
1 #Lệnh for với range(m,n)
2 for i in range(5,10):
3     #Giá trị khởi đầu m = 5
4     #Giá trị kết thúc n = 10
5     #Bước nhảy mặc định = 1
6     print('i = ',i)
```

2

```
i = 5
i = 6
i = 7
i = 8
i = 9
```

Vòng lặp for lồng nhau

- Trong một số bài toán chúng ta cần kết hợp và sử dụng nhiều câu lệnh lặp đặt lồng nhau để giải quyết.


```
1  #Hiển thị bảng cửu chương từ 2 -> 9
2  for i in range(2,10):
3      print('Bảng cửu chương ', i)
4      for j in range(1,11):
5          print (i , ' x ', j, ' = ', i*j)
6      print('-----')
```

Bảng cửu chương 2


```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

Break, continue cho cả while và for


```
for var in sequence:  
    # codes inside for loop  
    if condition:  
        break  
    # codes inside for loop  
# codes outside for loop
```




```
while test expression:  
    # codes inside while loop  
    if condition:  
        break  
    # codes inside while loop  
# codes outside while loop
```



```
for var in sequence:  
    # codes inside for loop  
    if condition:  
        continue  
    # codes inside for loop  
# codes outside for loop
```



```
while test expression:  
    # codes inside while loop  
    if condition:  
        continue  
    # codes inside while loop  
# codes outside while loop
```



Thực hành

Bài 9: Hiển thị bảng cửu chương

Nhập vào bảng cửu chương muốn in (1 – 10), hiển thị bảng cửu chương tương ứng với số vừa nhập. (Thực hiện kiểm tra giá trị nhập, chỉ thỏa mãn khi giá trị nhập vào trong khoảng [1,10])

Nhập vào bảng cửu chương muốn hiển thị [1-10]:12

Nhập sai, Vui lòng nhập số từ 1 đến 10:6

BẢNG CỬU CHƯƠNG 6

$$6 \times 1 = 6$$

$$6 \times 2 = 12$$

$$6 \times 3 = 18$$

$$6 \times 4 = 24$$

$$6 \times 5 = 30$$

$$6 \times 6 = 36$$

$$6 \times 7 = 42$$

$$6 \times 8 = 48$$

$$6 \times 9 = 54$$

$$6 \times 10 = 60$$

1	2	3	4	5
1x1 = 1 1x2 = 2 1x3 = 3 1x4 = 4 1x5 = 5 1x6 = 6 1x7 = 7 1x8 = 8 1x9 = 9 1x10 = 10	2x1 = 2 2x2 = 4 2x3 = 6 2x4 = 8 2x5 = 10 2x6 = 12 2x7 = 14 2x8 = 16 2x9 = 18 2x10 = 20	3x1 = 3 3x2 = 6 3x3 = 9 3x4 = 12 3x5 = 15 3x6 = 18 3x7 = 21 3x8 = 24 3x9 = 27 3x10 = 30	4x1 = 4 4x2 = 8 4x3 = 12 4x4 = 16 4x5 = 20 4x6 = 24 4x7 = 28 4x8 = 32 4x9 = 36 4x10 = 40	5x1 = 5 5x2 = 10 5x3 = 15 5x4 = 20 5x5 = 25 5x6 = 30 5x7 = 35 5x8 = 40 5x9 = 45 5x10 = 50
6	7	8	9	10
6x1 = 6 6x2 = 12 6x3 = 18 6x4 = 24 6x5 = 30 6x6 = 36 6x7 = 42 6x8 = 48 6x9 = 54 6x10 = 60	7x1 = 7 7x2 = 14 7x3 = 21 7x4 = 28 7x5 = 35 7x6 = 42 7x7 = 49 7x8 = 56 7x9 = 63 7x10 = 70	8x1 = 8 8x2 = 16 8x3 = 24 8x4 = 32 8x5 = 40 8x6 = 48 8x7 = 56 8x8 = 64 8x9 = 72 8x10 = 80	9x1 = 9 9x2 = 18 9x3 = 27 9x4 = 36 9x5 = 45 9x6 = 54 9x7 = 63 9x8 = 72 9x9 = 81 9x10 = 90	10x1 = 10 10x2 = 20 10x3 = 30 10x4 = 40 10x5 = 50 10x6 = 60 10x7 = 70 10x8 = 80 10x9 = 90 10x10 = 100

Bài 10: Tính điểm học tập của bạn

Khởi tạo một danh sách bao gồm các phần tử là điểm của bạn (hệ 10). Hãy thực hiện:

- 1) Tạo một danh sách là các điểm chữ, điểm hệ 4 tương ứng với điểm hệ 10 ở trên.
- 2) Tính điểm trung bình Hệ 10 và Hệ 4

Loại	Thang điểm 10	Thang điểm 4	
		Điểm số	Điểm chữ
Đạt	Từ 9,0 đến 10	4,0	A ⁺
	Từ 8,5 đến 8,9	3,7	A
	Từ 8,0 đến 8,4	3,5	B ⁺
	Từ 7,0 đến 7,9	3,0	B
	Từ 6,5 đến 6,9	2,5	C ⁺
	Từ 5,5 đến 6,4	2,0	C
	Từ 5,0 đến 5,4	1,5	D ⁺
	Từ 4,0 đến 4,9	1,0	D
Không đạt	Dưới 4,0	0	F

```
Điểm hệ 10: [8.4, 6.5, 7.3, 2.6, 9.0, 5.8, 6.0, 9.7, 8.1]
Điểm chữ tương ứng: ['B+', 'C+', 'B', 'F', 'A+', 'C', 'C', 'A+', 'B+']
-----Điểm Trung Bình-----
Tổng số môn học: 9
ĐTB hệ 10: 7.044444444444444
ĐTB hệ 4: 2.7222222222222223
```

Bài 11: Kiểm tra N có là số nguyên tố?

Nhập vào số tự nhiên N, cho biết N có phải là số nguyên tố hay không? (Số nguyên tố là số lớn hơn 1, chỉ chia hết cho 1 và chính nó)

Nhập vào một số nguyên dương N ($N > 1$): 25
Số 25 không phải là số nguyên tố!

Nhập vào một số nguyên dương N ($N > 1$): 43
Số 43 là số nguyên tố!

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Bài 12: dãy số nguyên tố.

Nhập vào một số N, hiển thị các số nguyên tố từ 2 tới N:

Nhập vào một số nguyên dương N ($N > 1$): 250
Các số nguyên tố từ 1 tới 250 :
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241,

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Bài 13: Đổi số từ thập phân sang nhị phân.

Nhập vào một số tự nhiên N ($N > 0$), chuyển đổi số N sang hệ nhị phân.

Nhập vào một số tự nhiên (>0):30

30 (hệ 10) = 11110 (hệ 2)

Nhập vào một số tự nhiên (>0):100

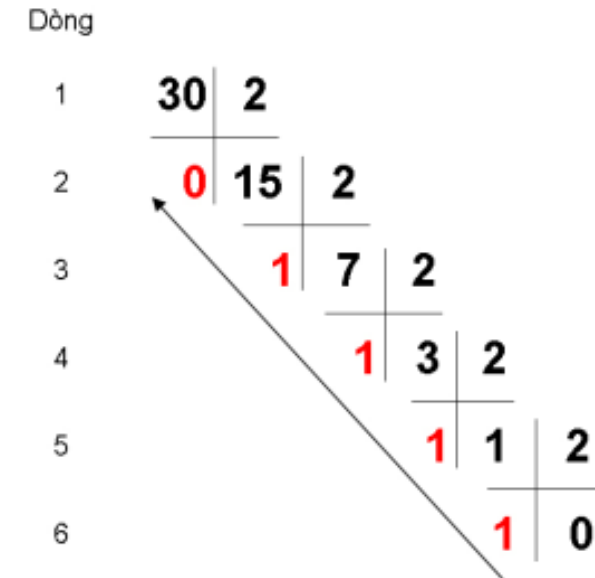
100 (hệ 10) = 1100100 (hệ 2)

Nhập vào một số tự nhiên (>0):1024

1024 (hệ 10) = 10000000000 (hệ 2)

Nhập vào một số tự nhiên (>0):2019

2019 (hệ 10) = 11111100011 (hệ 2)



5. Ngoại lệ (Exception)

Ngoại lệ

- Ngoại lệ là một sự kiện, xảy ra trong quá trình thực thi chương trình làm gián đoạn luồng hướng dẫn bình thường của chương trình.
- Khi một tập lệnh trong python tạo ra một ngoại lệ, nó phải xử lý ngoại lệ đó ngay lập tức nếu không nó sẽ kết thúc và thoát.
- Một số ngoại lệ phổ biến trong Python:
 - ZeroDivisionError:
 - NameError:
 - IndentationError:
 - IOError:
 - EOFError:
 - ...

```
-----  
ZeroDivisionError                                Traceback (most recent call last)  
<ipython-input-4-7fec9ef9d1af> in <module>  
    1 a = 10  
    2 b = 0  
----> 3 c = a/b  
    4 print('a/b = ',c)  
  
ZeroDivisionError: division by zero
```

```
-----  
NameError                                         Traceback (most recent call last)  
<ipython-input-2-780a7d5891ae> in <module>  
----> 1 print (A)  
  
NameError: name 'A' is not defined
```

```
File "<ipython-input-3-6f06c5334b49>", line 3  
    print(---)  
      ^  
IndentationError: unexpected indent
```

Xử lý ngoại lệ

- Xử lý ngoại lệ:

Python try-except Error & Exceptions Handling

try :

{ Execute/Run this code

except :

{ Execute this block when
exception occurred

else :

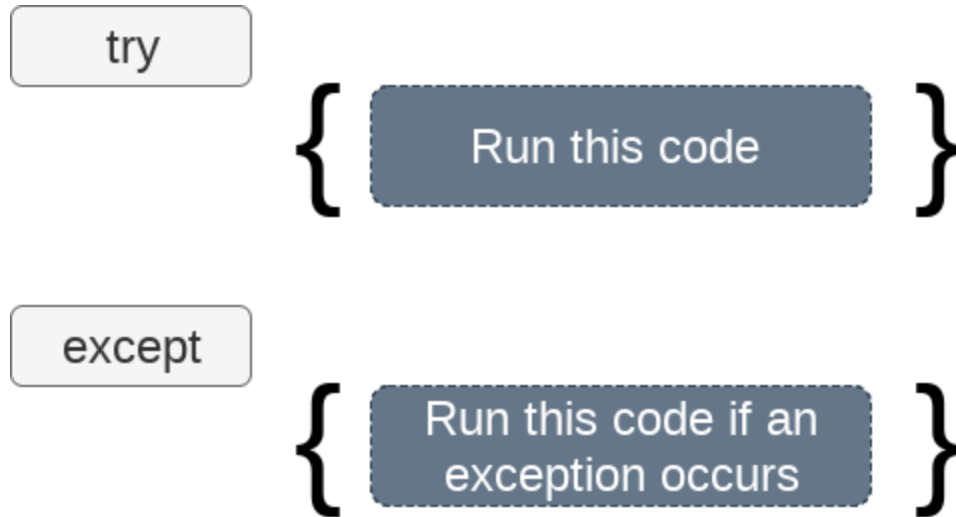
{ If no exception run this
code

finally :

{ Always run this block of
code

Xử lý ngoại lệ

- Xử lý ngoại lệ:



```
1 #try....except
2 try:
3     a = int(input("Enter a:"))
4     b = int(input("Enter b:"))
5     c = a/b
6 except:
7     print("Can't divide with zero")
```

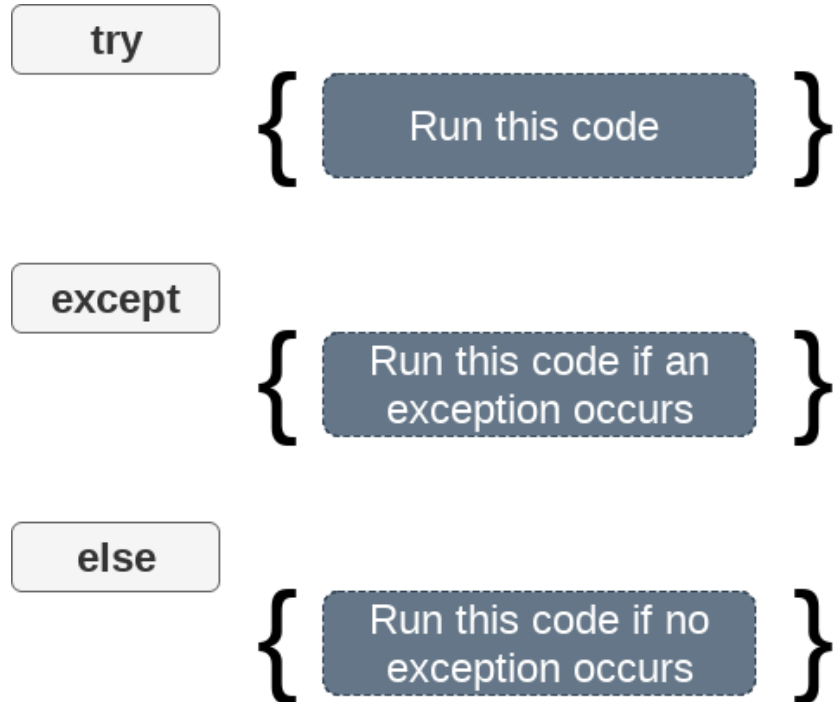
Enter a:10
Enter b:0
Can't divide with zero

```
1 #Hiển thị ngoại lệ:
2 try:
3     fh = open("testfile", "r")
4     fh.write("This is my test file for exception handling!!")
5 except Exception as e:
6     print(e)
```

[Errno 2] No such file or directory: 'testfile'

Xử lý ngoại lệ

- Xử lý ngoại lệ:



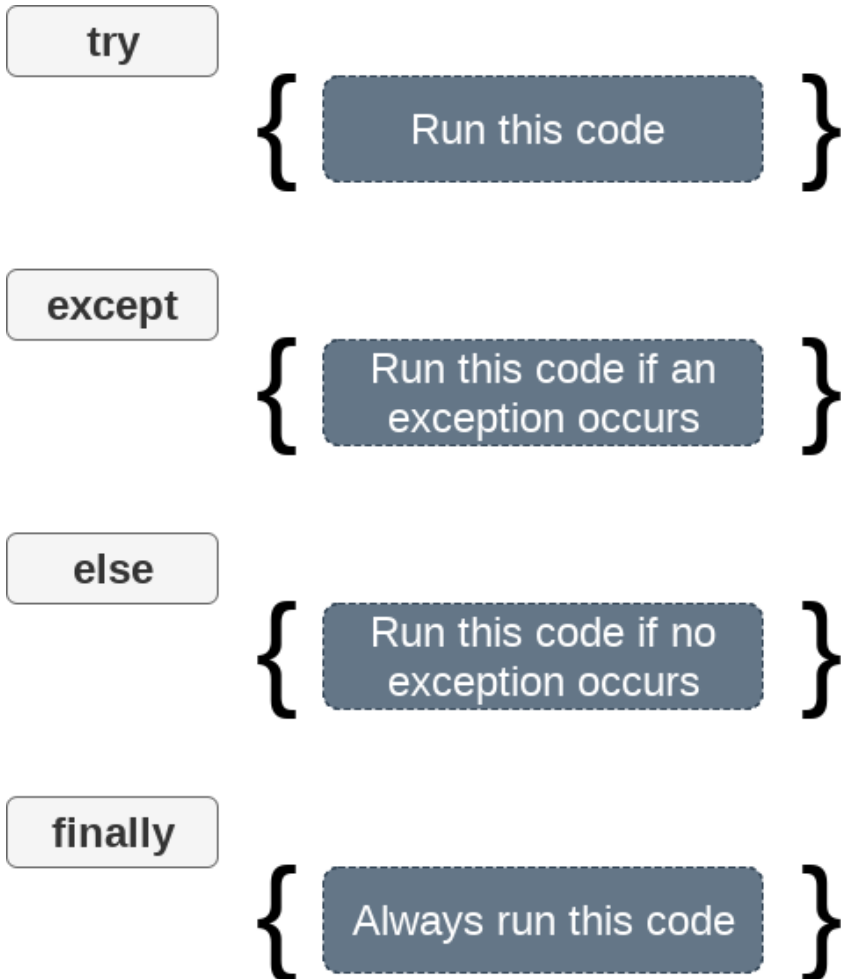
```
1 #Try....except...else:
2 try:
3     fh = open("testfile.txt", "r")
4     fh.write("This is my test file for exception handling!!")
5 except Exception as e:
6     print("Error: can't find file or read data")
7     print(e)
8 else:
9     print("Written content in the file successfully")
```

Error: can't find file or read data
not writable

- Một số lưu ý:
 - Một câu lệnh try duy nhất có thể có nhiều câu lệnh except.
 - Mã trong khối else thực thi nếu mã trong khối try không phát sinh ngoại lệ.
 - Khối lệnh else là không lệnh không cần sự bảo vệ của khối try

Xử lý ngoại lệ


- Xử lý ngoại lệ:



- Khối finally là nơi để đặt khối lệnh cần phải được thực thi, cho dù khối try có đưa ra ngoại lệ hay không.

```
1  ##Try....except...else...finally
2  try:
3      fh = open("testfile.txt", "r")
4      fh.write("This is my test file for exception handling!!")
5  except IOError as e:
6      print("Error: can't find file or read data")
7      print(e)
8  else:
9      print("Written content in the file successfully")
10 finally:
11     fh.close()
12     print("End process....")
```

```
Error: can't find file or read data
not writable
End process....
```



Q & A
Thank you!