



VÒNG LẶP



1. Vòng lặp for và hàm range():

a) Vòng lặp for:



Trong ngôn ngữ lập trình Python có chút khác so với các ngôn ngữ lập trình như C/C++, Java,...

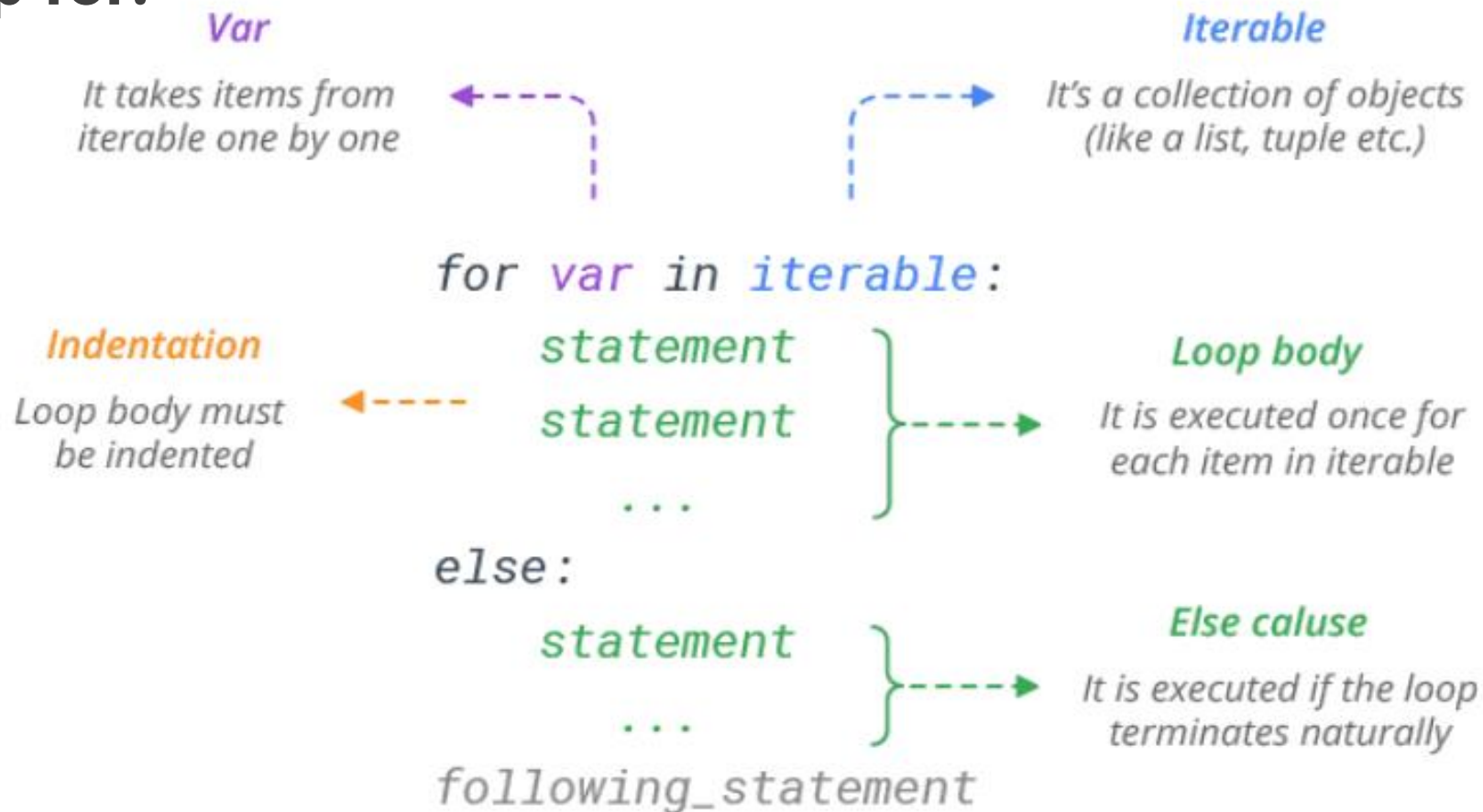


Thay vì duyệt qua các giá trị số thì Python sẽ lặp trên các iterable (list, tuple, string, set...). Thứ tự duyệt sẽ theo thứ tự xuất hiện trong iterable. Để thực hiện vòng lặp for ta sử dụng built-in function là range().



1. Vòng lặp for và hàm range():

a) Vòng lặp for:



1. Vòng lặp for và hàm range():

b) Hàm range():



Hàm range() sẽ sinh ra một dãy số và bạn sẽ sử dụng vòng for để duyệt qua từng số trong dãy đã sinh ra.

Cú pháp: `range (start, stop, step)`

Các tham số:

- start:** Giá trị bắt đầu của dãy số (mặc định là 0).
- stop:** Giá trị cuối cùng của dãy số (cận này không được lấy).
- step:** Bước nhảy của dãy số (mặc định là 1).

1. Vòng lặp for và hàm range():

b) Hàm range():

EXAMPLE

start = 0, stop = 5, step = 1

```
for i in range(5):  
    print('28tech', end = ' ')
```

OUTPUT

28tech 28tech 28tech 28 tech 28tech

EXAMPLE

start = 0, stop = 5, step = 1

```
for i in range(0, 5, 1):  
    print('28tech', end = ' ')
```

OUTPUT

28tech 28tech 28tech 28 tech 28tech

1. Vòng lặp for và hàm range():

b) Hàm range():

EXAMPLE

start = 2, stop = 5, step = 1

```
for i in range(2, 5):  
    print('28tech', end = ' ')
```

OUTPUT

28tech 28tech 28tech

EXAMPLE

start = 1, stop = 11, step = 2

```
for i in range(1, 11, 2):  
    print(i, end = ' ')
```

OUTPUT

1 3 5 7 9

1. Vòng lặp for và hàm range():

b) Hàm range():

EXAMPLE

In ra các ước của N

```
n = 100
for i in range(1, 100, 1):
    if n % i == 0:
        print(i, end = ' ')
```

OUTPUT

1 2 4 5 10 20 25 50

EXAMPLE

Duyệt các bội không âm của 3 nhỏ hơn hoặc bằng N


```
n = 31
for i in range(0, 31, 3):
    print(i, end = ' ')
print()
# Cách khác
for i in range(0, 31):
    if i % 3 == 0 : print(i, end = ' ')
```

OUTPUT

0 3 6 9 12 15 18 21 24 27 30
0 3 6 9 12 15 18 21 24 27 30

1. Vòng lặp for và hàm range():


b) Hàm range():


QUIZ

```
for i in range(1, 10, 3):  
    print(i, end = ' ')
```

OUTPUT

???


QUIZ

```
for i in range(1, 10, 0):  
    print(i, end = ' ')
```

OUTPUT

???

1. Vòng lặp for và hàm range():

c) Vòng for lồng nhau:



Vòng for lồng nhau (Nested loop) xuất hiện khi một câu lệnh bên trong vòng for này lại là một vòng for khác. Để hiểu được cách hoạt động của vòng for các bạn chỉ cần nhớ rằng mỗi vòng lặp của vòng for bên ngoài thì toàn bộ vòng for con bên trong sẽ được thực hiện.

EXAMPLE

```
for i in range(3):  
    print('Vong for ngoai khi i = ', i)  
    for j in range(2):  
        print(i, j)
```

OUTPUT

```
Vong for ngoai khi i = 0  
0 0  
0 1  
Vong for ngoai khi i = 1  
1 0  
1 1  
Vong for ngoai khi i = 2  
2 0  
2 1
```

2. Câu lệnh break và continue:

a) Câu lệnh break:



Câu lệnh break được sử dụng để kết thúc vòng lặp ngay lập tức, vòng lặp for sẽ kết thúc ngay tại thời điểm gặp câu lệnh break và tiếp tục các câu lệnh bên dưới vòng for. Thông thường thì câu lệnh break sẽ đi kèm theo một điều kiện kích hoạt.

EXAMPLE

```
for i in range(5):  
    print(i, end = ' ')  
    if i == 3: break  
    print('28tech', end = ' ')  
print('Ket thuc')
```

OUTPUT

0 28tech 1 28tech 2 28tech 3 Ket thuc

2. Câu lệnh break và continue:

b) Câu lệnh continue:



Câu lệnh continue được dùng để bỏ qua lần lặp hiện tại và quay trở lại luôn vòng lặp tiếp theo. Các câu lệnh bên dưới continue ở trong vòng lặp sẽ được bỏ qua.

**EXAMPLE**

```
for i in range(5):  
    print('28tech')  
    continue  
    print('python')
```

OUTPUT

```
28tech  
28tech  
28tech  
28tech  
28tech
```

3. Vòng lặp while:

a) Vòng lặp while, break, continue :



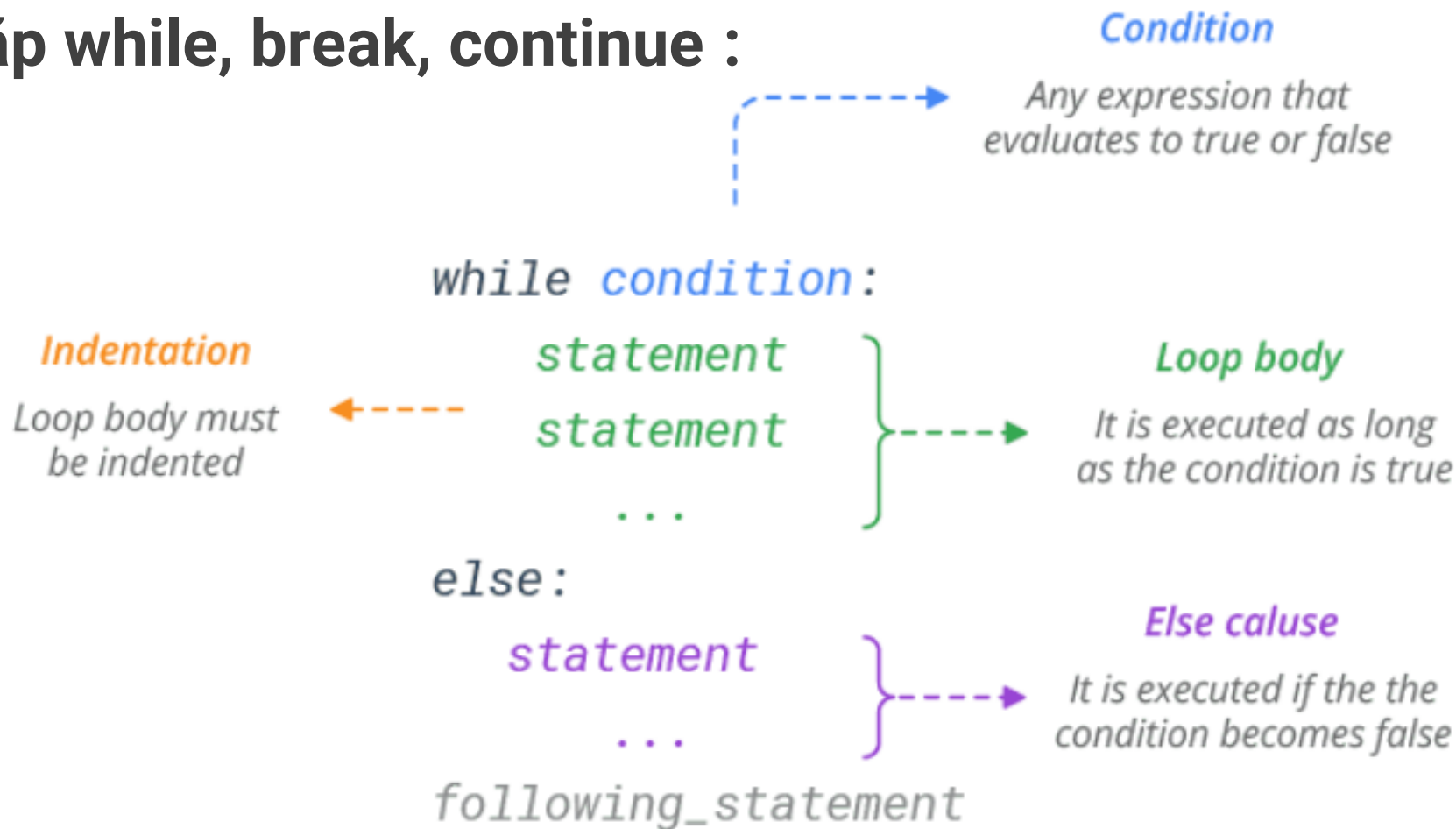
While được sử dụng khi bạn muốn thực hiện một tác vụ vô thời hạn, cho đến khi một điều kiện cụ thể được đáp ứng. Đây là một vòng lặp được kiểm soát theo điều kiện và thường sử dụng dụng khi chưa biết trước số lượng vòng lặp.

CÚ PHÁP

```
while condition:  
    #code when condtion is True  
else:  
    #code while condition is False
```

3. Vòng lặp while:

a) Vòng lặp while, break, continue :



3. Vòng lặp while:

a) Vòng lặp while, break, continue :

EXAMPLE

```
n = 1
while n < 5:
    print(n)
    n += 1
print('Following statement')
```

OUTPUT

```
1
2
3
4
Following statement
```

EXAMPLE

```
n = 1
while n < 5:
    print(n)
    n += 1
else:
    print('End')
print('Following statement')
```

OUTPUT

```
1
2
3
4
End
Following statement
```

3. Vòng lặp while:

a) Vòng lặp while, break, continue :

EXAMPLE

Tính tổng chữ số của số nguyên

```
n = 12345
sum_digit = 0
while n != 0:
    sum_digit += n % 10
    n //= 10
print('Sum :', sum_digit)
```

OUTPUT

Sum : 15

EXAMPLE

Lật ngược một số nguyên

```
n = 12345
rev = 0
while n != 0:
    rev = rev * 10 + n % 10
    n //= 10
print('rev :', rev)
```

OUTPUT

rev : 54321

3. Vòng lặp while:

a) Vòng lặp while, break, continue :

EXAMPLE

Câu lệnh break sẽ làm else trong while không được thực hiện

```
i = 1
while i <= 4:
    print(i)
    if i == 3: break
    i += 1
else:
    print('End')
print('Following statement')
```

OUTPUT

```
1
2
3
Following statement
```


3. Vòng lặp while:

a) Vòng lặp while, break, continue :

EXAMPLE

Câu lệnh continue

```
i = 1
while i <= 4:
    print('28tech')
    i += 1
    continue
    print('python')
else:
    print('End')
print('Following statement')
```

OUTPUT

```
28tech
28tech
28tech
28tech
End
Following statement
```

3. Vòng lặp while:

b) Vòng lặp vĩnh viễn:



Để tạo 1 vòng lặp vĩnh viễn thì bạn chỉ cần để điều kiện là True.

EXAMPLE

Vòng lặp sẽ lặp vĩnh viễn cho tới khi bạn nhập xâu là 28tech

```
while True:
    str = input('Enter your
password :')
    if str == '28tech':
        print('OK')
        break
    else:
        print('NOT OK')
```

EXAMPLE

Yêu cầu nhập lại khi người dùng nhập 1 số nguyên âm hoặc số 0

```
n = -1
while True:
    n = int(input('Enter a positive number :'))
    if n > 0:
        break
print(n)
```