Lecture 2

Truong Quang Nhat

Ngày 3 tháng 9 năm 2020

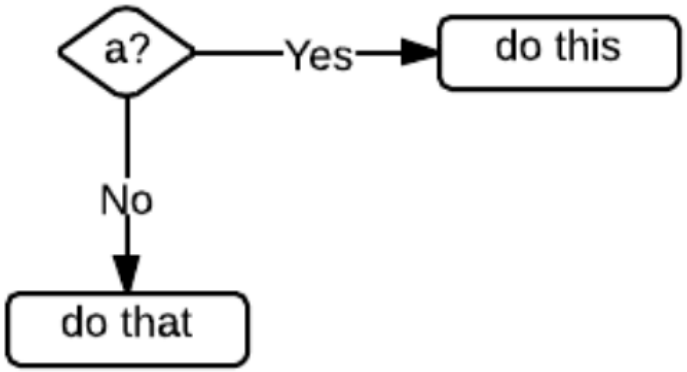
Nội dung

Các cấu trúc điều kiển trong Python

- Cấu trúc `if - elif - else`
- Vòng lặp
  - Vòng lặp `for`
  - Vòng lặp `while`

Cấu trúc `if - else`

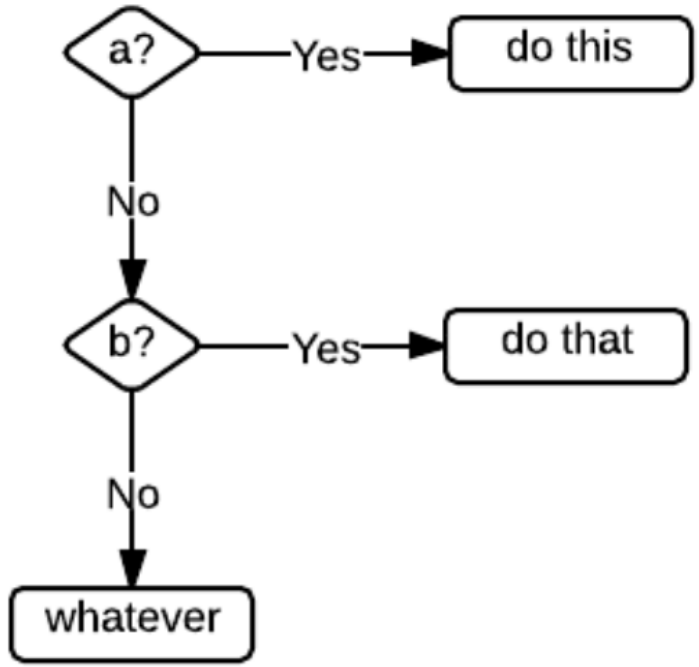
```
if a:
    do this
else:
    do that
```



Hình: Cấu trúc `if - else`

Cấu trúc if - elif - else

```
if a:
    do this
elif b:
    do that
else:
    whatever
```

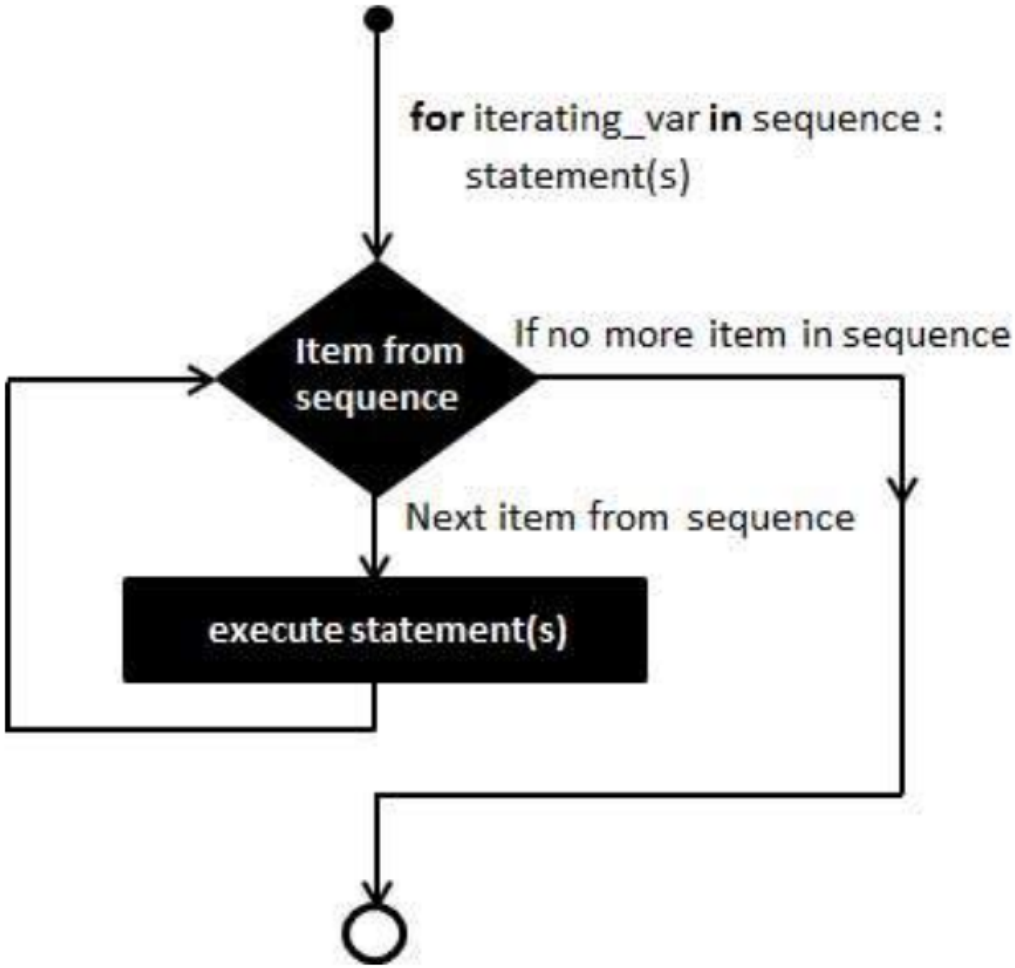


Hình: Cấu trúc `if - elif - else`

Ví dụ

```
x = 5
y = 4
if x == y:
  print("x bang y")
elif x > y:
  print("x lon hon y")
else:
  print("x nho hon y")


#ket qua
x lon hon y
```

Vòng lặp for



Hình: Vòng lặp for

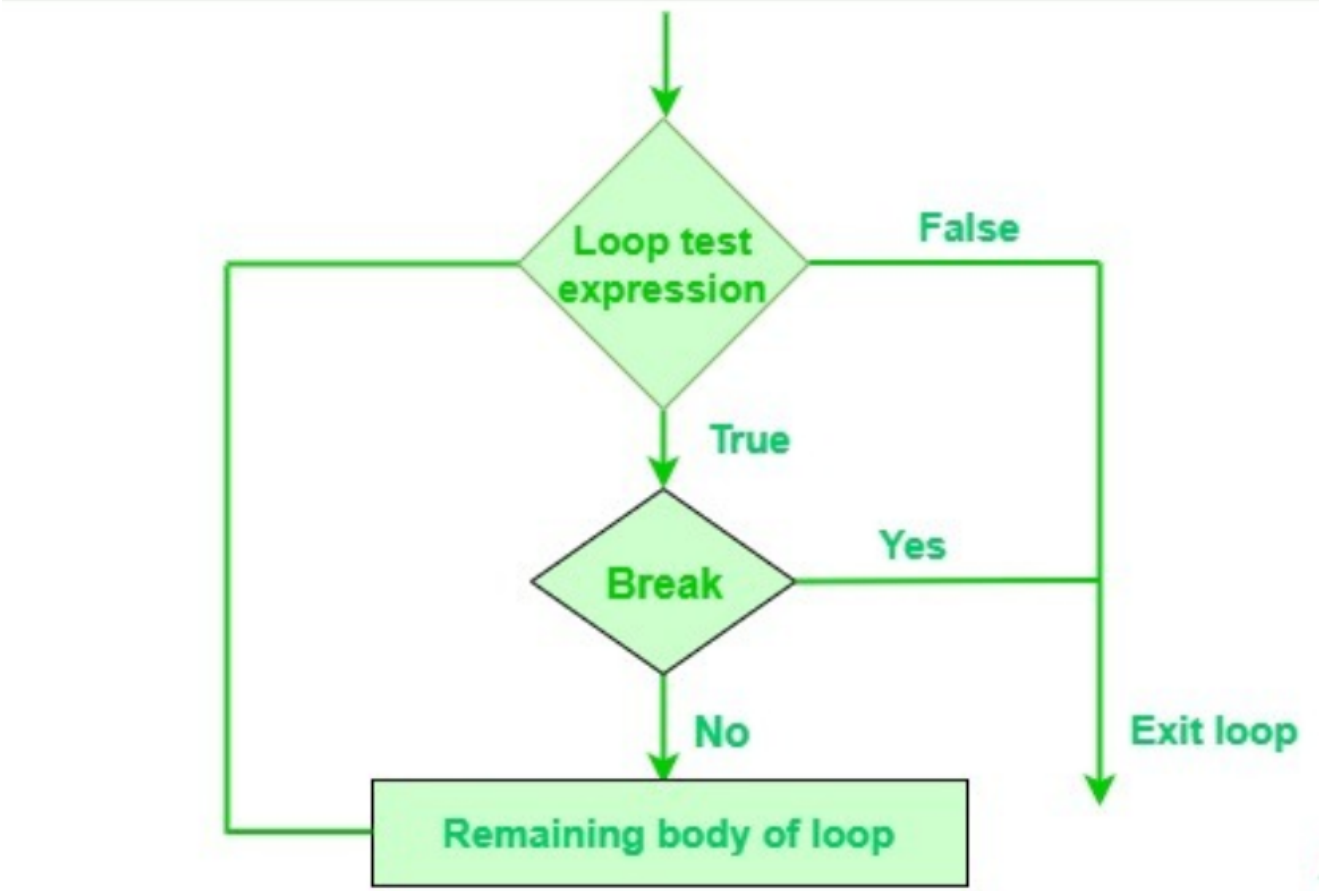
Ví dụ

```
#In ra cac phan tu trong list
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)

# Ket qua
apple
banana
cherry
```

Ví dụ

```
#In ra cac ki tu trong chuoi
for x in "banana":
  print(x)

# Ket qua:
b
a
n
a
n
a
```

Lệnh break

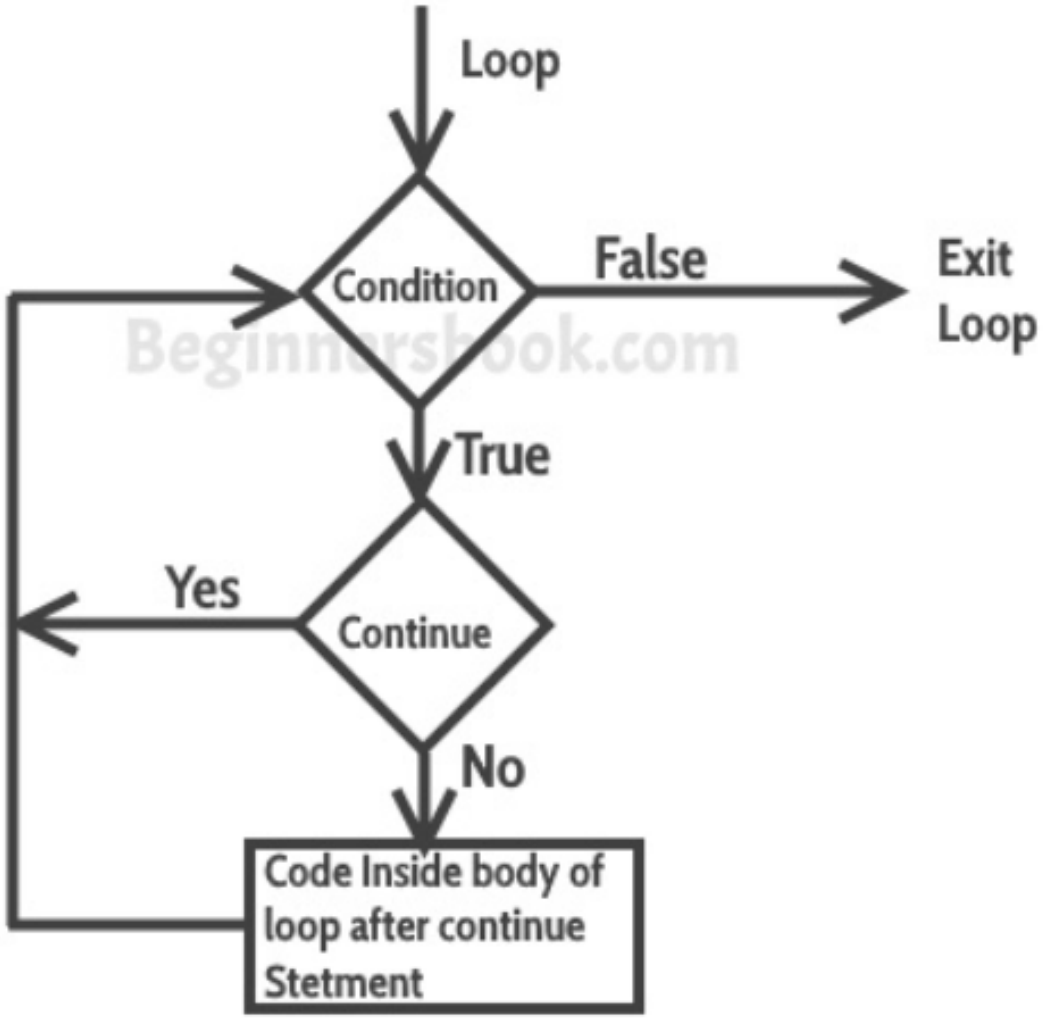


Hình: Lệnh break

Ví dụ

```
#Dung vong lap khi x = banana
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
  if x == "banana":
    break

# Ket qua
apple
banana
```

Lệnh continue



Hình: Lệnh continue

Ví dụ

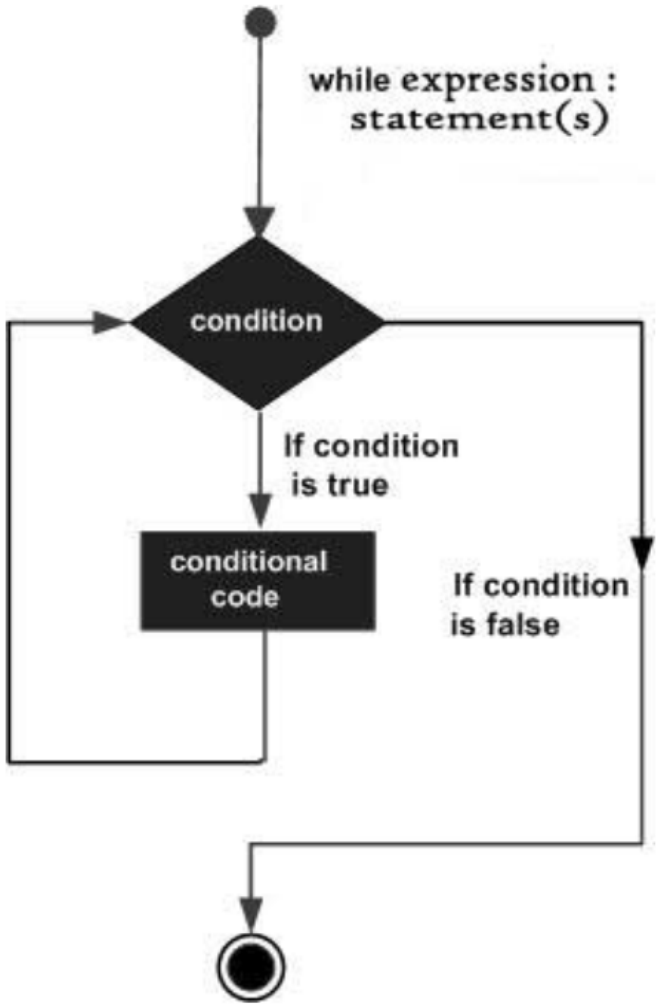
```
#Khong in ra gia tri banana
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  if x == "banana":
    continue
  print(x)

# Ket qua
apple
cherry
```

Lệnh range()

Để tiến hành lặp với một số lần nhất định cho trước, ta có thể sử dụng lệnh range. Hàm range(n) trả về một dãy các số, bắt đầu từ 0, kết thúc ở n - 1. Hàm range(m, n) sẽ trả về dãy số bắt đầu từ m, kết thúc ở n-1. Ta xem xét ví dụ dưới đây:

```
for x in range(2, 6):
  print(x)

#Ket qua
2
3
4
5
```

Vòng lặp while



Hình: Vòng lặp while

Ví dụ

```
x = 0
while x < 5:
  print('x is currently: ',x)
  print(' x is still less than 5, adding 1 to x')
  x+=1
else:
  print('All Done!')

#Ket qua
x is currently:  0
x is still less than 5, adding 1 to x
x is currently:  1
x is still less than 5, adding 1 to x
x is currently:  2
x is still less than 5, adding 1 to x
x is currently:  3
x is still less than 5, adding 1 to x
x is currently:  4
x is still less than 5, adding 1 to x
All Done!
```

Vòng lặp while

```
x = 0
while x < 10:
  print('x is currently: ',x)
  print(' x is still less than 10, adding 1 to x')
  x+=1
  if x==2:
    print('Breaking because x==2')
    break
  else:
    print('continuing...')
    continue

#Ket qua
x is currently:  0
x is still less than 10, adding 1 to x
continuing...
x is currently:  1
x is still less than 10, adding 1 to x
Breaking because x==2
```

Chú ý

Trong vòng lặp while, cần kiểm soát điều kiện dừng, vì có thể dẫn tới tình trạng lặp vô hạn lần. Chẳng hạn như ví dụ dưới đây:

```
# DO NOT RUN THIS CODE!!!!
while True:
  print("I'm stuck in an infinite loop!")
```

Xử lý một dataFrame

Đầu tiên, ta tiến hành import các thư viện cần thiết:

```
# Import packages and set visualization style
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
%matplotlib inline
```

Xử lý một dataFrame

```
#Tao dataFrame ngau nhien
df = pd.DataFrame(np.random.randn(10, 4))
df
```

	0	1	2	3
0	-0.845414	0.366514	0.729946	-0.605936
1	2.377305	-0.355096	-0.396565	-0.419888
2	-1.207315	0.505627	-0.129222	0.055994
3	-0.777510	0.462492	-0.124346	-1.853105
4	-0.390814	-0.337384	-0.315292	-0.045710
5	0.212578	0.150169	-0.217418	0.567705
6	0.547657	0.071664	-0.264882	-1.964389
7	0.689891	0.688959	-2.778321	0.335685
8	0.426844	-0.092391	-0.022044	0.849567
9	-0.400668	-0.417913	0.350527	1.798075

Tách và gộp dataFrame

```
pieces = [df[:3], df[5:7]]
#tach dataFrame thanh 2 phan dong 0--> 2 va 5-->6

pd.concat(pieces)
#noi 2 phan vua tach
```

	0	1	2	3
0	-0.845414	0.366514	0.729946	-0.605936
1	2.377305	-0.355096	-0.396565	-0.419888
2	-1.207315	0.505627	-0.129222	0.055994
5	0.212578	0.150169	-0.217418	0.567705
6	0.547657	0.071664	-0.264882	-1.964389

Hình: DataFrame mới

Lệnh merge

Ngoài ra, ta có thể nối 2 dataFrame bằng lệnh merge như sau:

```
left = pd.DataFrame({'key': ['foo', 'foo'], 'lval': [1, 2]})
right=pd.DataFrame({'key': ['foo', 'foo'], 'rval': [4, 5]})
pd.merge(left, right, on='key')
```

	key	lval
0	foo	1
1	foo	2

	key	rval
0	foo	4
1	foo	5

Hình: dataFrame left and right

Kết quả

	key	lval	rval
0	foo	1	4
1	foo	1	5
2	foo	2	4
3	foo	2	5

Hình: DataFrame mới

Group by trong Python

Khi đề cập đến Group by, tức là ta đang nhắc đến một quy trình gồm nhiều bước dưới đây:

- Chia dữ liệu thành các nhóm dựa trên một số tiêu chí
- Áp dụng các hàm tính toán cho từng nhóm một cách độc lập
- Kết hợp các kết quả thành một cấu trúc dữ liệu

Ví dụ

	A	B	C	D
0	foo	one	1.306349	1.646776
1	bar	one	-0.535545	0.085232
2	foo	two	0.906716	-0.779865
3	bar	three	0.194231	0.814457
4	foo	two	-0.265594	-0.584221
5	bar	two	0.264933	-0.256854
6	foo	one	-0.091046	-0.596817
7	foo	three	-0.574646	-0.136811

Hình: DataFrame khởi tạo

Ví dụ

Tiến hành gom nhóm theo cột A, và tính tổng của từng nhóm ta dùng lệnh sau:

```
df.groupby('A').sum()
```

Kết quả:

A	C	D
bar	-0.076381	0.642835
foo	1.281780	-0.450939

Hình: Kết quả

Ví dụ

Ngoài ra, ta cũng có thể group by nhiều dữ liệu một lúc như sau:

```
df.groupby(['A', 'B']).sum()
```

		C	D
A	**B**		
bar	**one**	-0.535545	0.085232
	three	0.194231	0.814457
	two	0.264933	-0.256854
foo	**one**	1.215304	1.049958
	three	-0.574646	-0.136811
	two	0.641122	-1.364086

Bài tập

1. Dùng lệnh để for và while tính giai thừa:

$$n! = 1 \times 2 \times ... \times (n-1) \times n$$

2. Tính tổng sau bằng cả for và while:

$$\sum_{k=5}^{20} \frac{k^2}{k-1}$$

3. Đặt:

$$s_n = \sum_{k=1}^{n} k$$

Tính tổng dưới đây bằng cả for và while:

$$\sum_{n=1}^{10} s_n$$

4. Viết chương trình tính tổng các số chia hết cho 3 và nhỏ hơn 50 bằng hai cách.

Bài tập

5. Nhập vào một số nguyên dương n, dùng lệnh while để tính tổng dưới đây:

$$\sum_{k=1}^{n} \frac{1}{k}$$

6. Tìm giá trị lớn nhất của n để:

$$\sum_{k=1}^{n} k^2 < 100$$

7. Tạo một dataFrame 10×4 thực, ngẫu nhiên. Sau đó thực hiện các yêu cầu sau:

- Vẽ histogram của cột 2.
- Tách và gộp các cột 2, 3, 6 thành một dataFrame mới. Đặt tên các cột mới là A, B, C.
- Phân cụm dữ liệu ở cột A thành 2 cụm $< 0, \geq 0$.

BTVN

1. Nhập vào một số nguyên dương bất kỳ, kiểm tra xem số đó có phải là số nguyên tố hay không.
2. Tính tổ hợp n chập k ($k \leq n$), với n, k được nhập từ bàn phím.
3. Tính tổng của 10 số Fibonaci đầu tiên. (Dãy Fibonaci là dãy có 2 phần tử đầu tiên là 1, các phần tử sau đó là tổng của 2 phần tử trước đó: 1, 1, 2, 3, 5, ...).
4. Nhập vào một vectơ dương. Tính các tổng dưới đây:
   - Tổng các phần tử chẵn
   - Tổng các phần tử lẻ
   - Tổng các số chính phương
   - Tổng các số nguyên tố.

BTVN

5. Cho một vecto bất kì, đếm xem trong đó có bao nhiêu phần tử âm, bao nhiêu phần tử dương, bao nhiêu phần tử bằng 0.
6. Tải bộ dữ liệu của Netflix ở đây. Tiến hành group by bộ data trên theo cột *release_year*. Sau đó vẽ biểu đồ scatter, với x là *release_year*, y là median rating.