



---

# Câu lệnh lặp

Khóa học: Python căn bản

---

# Kiểm tra bài trước

Hỏi và trao đổi về các khó khăn gặp phải trong bài “Cấu trúc điều kiện”

Tóm tắt lại các phần đã học từ bài “Cấu trúc điều kiện”

# Mục tiêu

---

- Giải thích được khái niệm lặp
- Mô tả được cú pháp của vòng lặp for, while, do/while
- Giải thích được cách sử dụng lệnh break, continue
- Trình bày được vòng lặp lồng nhau
- Sử dụng được vòng lặp for, while, do/while
- Sử dụng được lệnh break, continue, pass
- So sánh được các loại vòng lặp
- Sử dụng được vòng lặp lồng nhau



---

# Thảo luận

Vòng lặp for



# Vòng lặp (loop)

---

- Vòng lặp cho phép tự động thực hiện một khối lệnh lặp đi lặp lại nhiều lần dựa vào một điều kiện cho trước
- Vòng lặp giúp cho lập trình viên viết được các mã nguồn ngắn gọn hơn so với việc phải viết lặp lại những dòng mã tương tự nhau
- Vòng lặp bao gồm:
  - Một điều kiện để đánh giá (lúc nào thì lặp?)
  - Một khối lệnh để thực thi (lặp cái gì?)
- Các vòng lặp hỗ trợ bởi Python: *for*, *while*

# Vòng lặp: Ví dụ



Customers who bought this product

CopyPrintExcelCSVPDFReset column order

Show10orders

Search

Date	Order	Billing First name	Billing Last name	Billing E-mail	Billing Phone	Billing Address 1	
2017-05-14	<a href="#">2712</a>	John	Doe	<a href="mailto:john@johndoe.com">john@johndoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	20 John avenue	
2017-05-06	<a href="#">2645</a>	Jane	Doe	<a href="mailto:jane@janedoe.com">jane@janedoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	72 Jane street	
2017-05-06	<a href="#">2645</a>	James	Doe	<a href="mailto:james@jamesdoe.com">james@jamesdoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	433 James plaza	
2017-07-21	<a href="#">2919</a>	Jeremy	Doe	<a href="mailto:jeremy@jeremydoe.com">jeremy@jeremydoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	154 Jeremy lane	Creel
2017-07-21	<a href="#">2919</a>	Janis	Doe	<a href="mailto:janis@janisdoe.com">janis@janisdoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	747 Janis road	Creel
2017-06-22	<a href="#">2905</a>	Jolene	Doe	<a href="mailto:janis@janisdoe.com">janis@janisdoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	4830 Jolene circle	
2017-05-14	<a href="#">2710</a>	Jasper	Doe	<a href="mailto:jasper@jasperdoe.com">jasper@jasperdoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	765 Jasper route	
2017-05-13	<a href="#">2709</a>	Jackie	Doe	<a href="mailto:iackie@jackiedoe.com">iackie@jackiedoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	7373 Jackie crescent	406

Showing 1 to 8 of 8 entries1 row selected

Email all customersEmail selected customers

Total : 10

# Vòng lặp: Ví dụ



Screenshot of a Facebook group page titled "MLM Affiliate Marketing World". The page shows a list of members, including:

- Nikki Rosemain**: Radio Talk Show Host at Christian Bible Talk. Joined. Added about 9 months ago.
- Mary Ann LaCombe**: Legal Shield Independent Associate at LegalShield. Joined. Added by Dennis J. duffeck about 9 months ago.
- Dave Manning**: U.S. Army. Joined. Added by Gerald Reynolds about 10 months ago.
- Julia Ramirez**: Learn more about me at Your Social Media Angel. Joined. Added by Dianne Cavanagh Gardner about 9 months ago.
- Carlos Vera**: Joined. Added about 8 months ago.
- Adrian Armstrong**: Works at Self Employed Entrepreneur. Joined. Added about 9 months ago.
- April Clark**: Independent Distributor at MONAT Canada Official. Joined. Added about 9 months ago.
- Angela Gower-Johnson**: Business Energy Strategist at. Added Friend.
- D.a. Dupree**: Works at Self Employed and Loving It! Joined. Added about 8 months ago.
- Andy Van Dyke**: Joined. Added by Leon Jenkins about 4 months ago.
- Wiley Brenneca**: Independent Distributor at Jeunesse Global. Joined. Added about 7 months ago.
- Dee Anne Hall**: Mentor/Coach at Self-Employed. Joined. Added by Enrique Salinas about 8 months ago.
- Karen Monasingh**: Works at Karen Monasingh Marketing Training. Joined. Added about 9 months ago.
- Jimmy Ezzell**: Founder at Entrepreneur. Joined. Added about 8 months ago.
- Tony Collinson**: Owner at Self Employed Entrepreneur. Joined. Added about 8 months ago.
- Mujibur Rahman**: Chief Executive Officer at.

The right sidebar shows a list of people who liked the page, including Brian Halcomb, Ida Crowder, Reginald Brazier, Maureen Chady Spatho, Sonya Ravji, Catalina Maese-Ragsdale, Tu Tri Huynh, Nicole Sabath, Nicholas Carmona Segura, Raquel Jackson, Luis Cortes, Steve Howell, Larry Borges, Lissette Wagner, Jason R Kraft, Keith Slice Thompson, Sara Carson, Roberto Gonzalez, Sandie Vill, and Katie Marie.

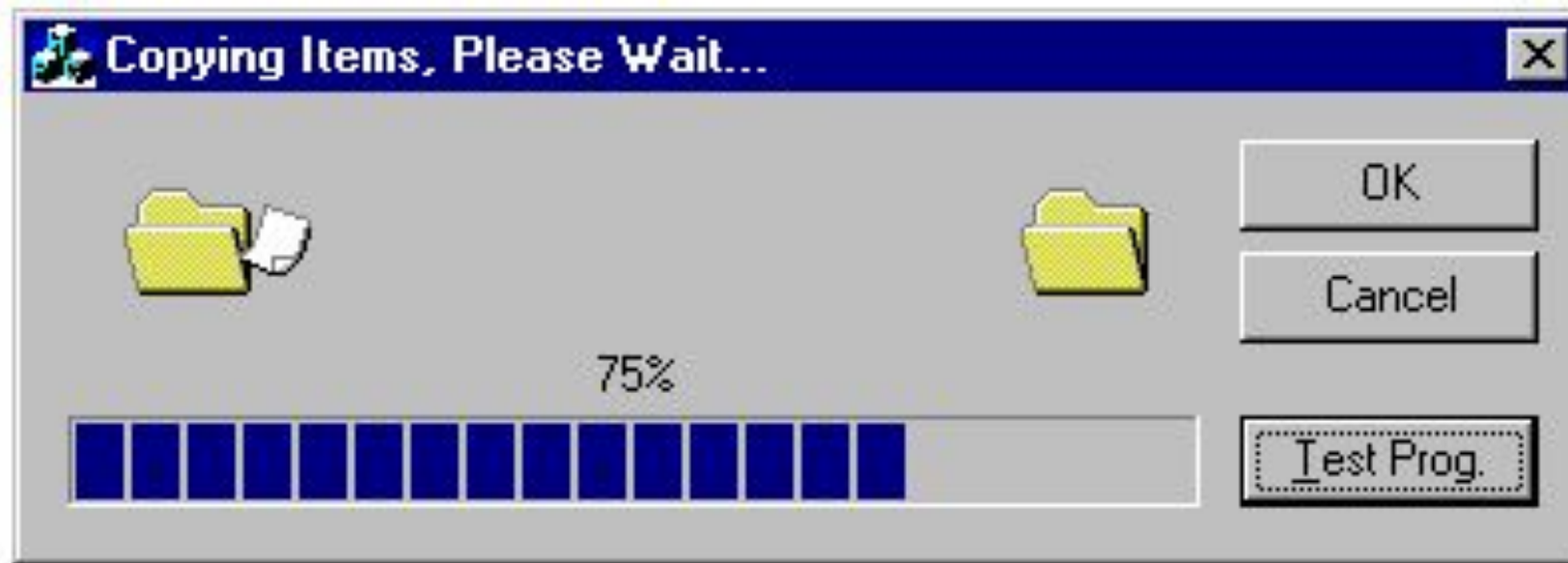
# Vòng lặp: Ví dụ

---





# Vòng lặp: Ví dụ





---

# Thảo luận

Vòng lặp while

# Vòng lặp while

- Vòng lặp *while* thực thi lặp lại một khối lệnh nếu biểu thức điều kiện trả về giá trị đúng
- Cú pháp:

***while*** (*loop-continuation-condition*) :  
*statement(s)*;

Trong đó:

- *loop-continuation-condition* : là biểu thức điều kiện
- *statement(s)*: là các câu lệnh được thực thi trong mỗi lần lặp

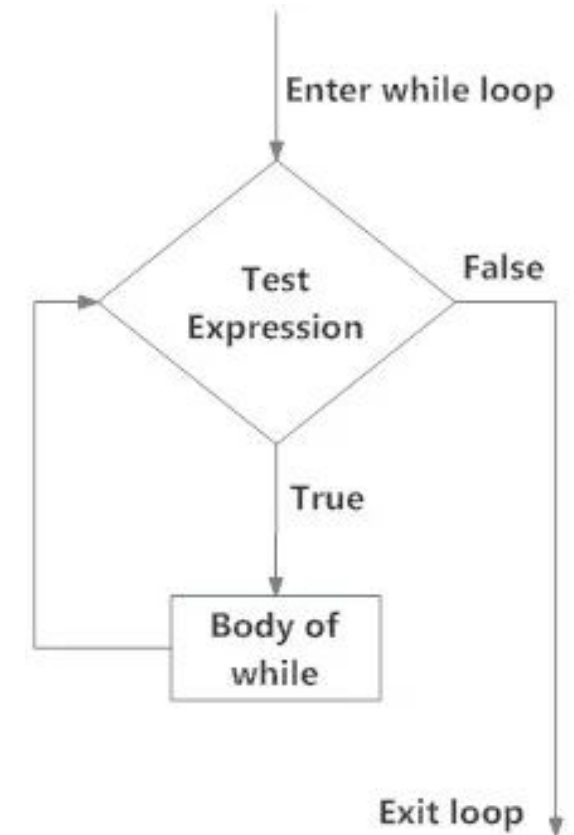


Fig: operation of while loop

# While: Ví dụ

---



```
# Program to add natural
# numbers up to
# sum = 1+2+3+...+n
n = 10
# initialize sum and counter
sum = 0
i = 1
while i <= n:
    sum = sum + i
    i = i+1    # update counter
# print the sum
print("The sum is", sum)
```

```
Enter n: 10
The sum is 55
```



# While với else: Ví dụ

---

```
counter = 0
```

```
while counter < 3:  
    print("Inside loop")  
    counter = counter + 1  
else:  
    print("Inside else")
```

```
Inside loop  
Inside loop  
Inside loop  
Inside else
```



# Lặp với điều kiện phức tạp

---

- Nhập vào một chuỗi để in ra trang web
- Nhập vào ký tự "q" hoặc ký tự rỗng để thoát vòng lặp

# Lặp vô hạn với while

---



```
while True:  
    #Statements
```

```
while 1:  
    #Statements
```

**Lưu ý:** Khi vòng lặp thực thi vô tận thì có thể sẽ hao tốn rất nhiều tài nguyên. Cần thiết kể *điều kiện dừng* cẩn thận để tránh xảy ra trường hợp này.



---

# Demo

Vòng lặp while



# Vòng lặp for

Với vòng lặp for chúng ta có thể thực hiện 1 tập hợp các câu lệnh một lần cho từng mục trong danh sách hoặc chuỗi.

## Cú pháp:

```
for val in list:  
    print(val)
```

Trong đó:

- val: biến nhận giá trị của từng mục trong chuỗi trên mỗi lần lặp.
- list: danh sách những phần tử cần lặp

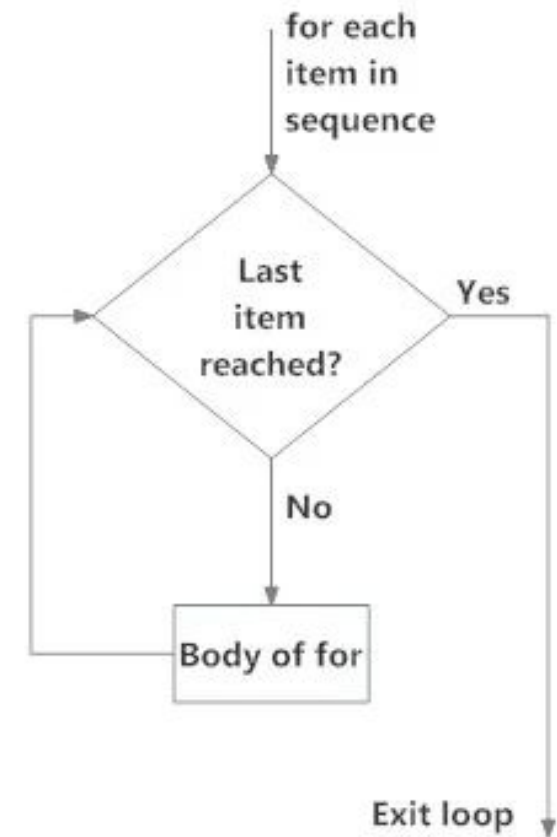


Fig: operation of for loop



# Vòng lặp for: Ví dụ 1

---

```
# List of numbers  
numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]
```

```
# variable to store the sum  
sum = 0
```

```
# iterate over the list  
for val in numbers:  
    sum = sum+val
```

```
print("The sum is", sum)
```

```
The sum is 48
```

# Vòng lặp for: Ví dụ

---



## Ví dụ: Lặp qua 1 chuỗi

```
for val in "apple" :  
    print(val)
```

**Kết quả: a, p, p, l, e**

Vòng lặp sẽ trả ra từng phần tử trong chuỗi đã cho

## Ví dụ: Lặp qua 1 danh sách

```
list = [ "apple", "banana", "cherry" ]  
for val in list  
    print(val)
```

**Kết quả: apple, banana, cherry**

Vòng lặp trả ra từng phần tử trong mảng danh sách đã cho



---

# Demo

Vòng lặp for



# Hàm range() với vòng lặp

---

Hàm range trả về một mảng trong đó tổng số phần tử sẽ phụ thuộc vào các tham số truyền vào.

## Cú pháp:

```
range(start, end, step)
```

Trong đó:

- start: giá trị bắt đầu
- end: giá trị kết thúc
- step: khoảng cách giữa các phần tử hay còn gọi là bước nhảy



# Hàm range(): Ví dụ 1

---

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

Kết quả sẽ tạo một mảng gồm 5 phần tử có giá trị lần lượt từ 0  
-> 5.

0, 1, 2, 3, 4,



# Hàm range(): Ví dụ 2

---

```
for i in range(5, 10):  
    print(i, end=', ')
```

Kết quả sẽ tạo một mảng gồm 5 phần tử có giá trị lần lượt là 5  
-> 9

5, 6, 7, 8, 9,



# Hàm range(): Ví dụ 3

---

```
for i in range(1, 10, 2):  
    print(i, end=', ')
```

Kết quả trả về một mảng 5 phần tử có giá lần lượt là 1, 3, 5, 7, 9 vì bước nhảy là 2.

1, 3, 5, 7, 9,



---

# Một số từ khóa dùng với vòng lặp

=



# Từ khóa **break**

---

**break** : là từ khóa dùng để dừng và thoát khỏi vòng lặp

**Ví dụ :**

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)  
    if x == "banana":  
        break
```

**Kết quả: apple, banana**



# Từ khóa **continue**

---

**continue** : là từ khóa bỏ qua lần lặp 1 hàng hoặc các hàng cụ thể, và chuyển sang bước tiếp theo

**Ví dụ :**

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    if x == "banana":  
        continue  
    print(x)
```

**Kết quả: apple, cherry**



# Từ khóa pass

---

**pass:** Vòng lặp for không được để trống, nhưng trong trường hợp vì lý do nào đó vòng lặp không có nội dung, để tránh xảy ra lỗi thì ta dùng pass

## Cú pháp:

```
for i in s:
```

```
    # Không có lỗi xảy ra
```

```
    pass
```

# Từ khóa pass: Ví dụ

---



## Ví dụ :

```
for letter in 'Python':  
    if letter == 'h':  
        pass  
    print ('This is pass block')  
    print ('Current Letter :',letter)  
print ("Good bye!")
```

## Kết quả:

```
Current Letter : P  
Current Letter : y  
Current Letter : t  
This is pass block  
Current Letter : h  
Current Letter : o  
Current Letter : n  
Good bye!
```



# Từ khóa else

---

**else** : là từ khóa trong vòng lặp chỉ định 1 đoạn lệnh sẽ thực hiện khi kết thúc vòng lặp.

**Ví dụ :**

```
for x in [1, 2 ,3]:  
    print(x)  
else:  
    print("Finally finished!")
```

**Kết quả:**

1, 2 ,3 , Finally finished!



# Từ khóa else: Ví dụ

---

**Chú ý :** Else sẽ không thực hiện nếu vòng lặp kết thúc bằng break

**Ví dụ :**

```
for i in [1,2,3,4,5]:  
    if i==3:  
        break  
    print (i)  
else:  
    print ("for loop is done")  
print ("Outside the for loop")
```

**Kết quả:** 1 2 Outside the for loop



# Lựa chọn for hay while?

---

Đối với vòng lặp for:

- Câu lệnh for lặp qua một tập hợp (collection), đối tượng có thể lặp (iterable object) hoặc kết quả của một hàm khởi tạo (generator function) (ví dụ range()).
- Vòng lặp for có cấu trúc đơn giản dễ viết, là cách viết tốt nhất khi chúng ta biết số lần lặp lại
- Khi chúng ta có một tập hợp được tạo sẵn: set, tuple, list, map hoặc một chuỗi văn bản thì đơn giản, chúng ta sẽ dùng vòng lặp for.



# Lựa chọn while hay for?

---



Đối với vòng lặp while:

- Vòng lặp while là vòng lặp phụ thuộc điều kiện True hoặc False, vòng lặp chỉ dừng lại khi điều kiện là Sai
- Vòng lặp while sử dụng khi chúng ta không biết chính xác số lần lặp qua khối lệnh
- Nếu chúng ta không có cấu trúc dữ liệu gọn gàng để lặp lại hoặc chúng ta không có một hàm khởi tạo, bạn phải sử dụng vòng lặp while



# Lựa chọn for hay while: Ví dụ

---

## Ví dụ:

Dùng vòng lặp for  
for i in range(11)  
print i

Dùng vòng lặp while  
i = 0  
while i <= 10:  
 print (i)  
 i += 1

**Cách viết vòng lặp for đơn giản, dễ đọc hơn.**



# Vòng lặp lồng nhau (nested loops)

---

- Một vòng lặp có thể được đặt bên trong một vòng lặp khác
- Không hạn chế số lượng vòng lặp lồng nhau

## Cú pháp:

```
for [biến lặp đầu tiên] in [vòng lặp ngoài]: # Vòng lặp ngoài  
    for [biến lặp thứ hai] in [vòng lặp lồng]: # Vòng lặp lồng nhau
```

**Lưu ý:** Sử dụng quá nhiều vòng lặp lồng nhau sẽ làm tăng độ phức tạp của mã nguồn lên rất nhiều.



# Vòng lặp for lồng nhau: Ví dụ 1

---

**Ví dụ :**

```
num_list = [1, 2, 3]
```

```
alpha_list = ['a', 'b', 'c']
```

```
for number in num_list:
```

```
    print(number)
```

```
    for letter in alpha_list:
```

```
        print(letter)
```

**Kết quả: 1 a b c 2 a b c 3 a b c**



# Vòng lặp for lồng nhau: Ví dụ 2

**Ví dụ:**

```
num_list = [1, 2, 3]
alpha_list = ['a', 'b', 'c']

for number in num_list:
    print(number)
    for letter in alpha_list:
        print(letter)
```

**Kết quả: 1,a,b,c,2,a,b,c,3,a,b,c**

Chương trình sẽ hoàn thành lần lặp đầu tiên của vòng lặp bên ngoài và in ra 1, sau đó kích hoạt vòng lặp bên trong in ra a,b,c .

Khi vòng lặp bên trong hoàn thành, sẽ quay lại vòng lặp bên ngoài in ra 2, sau đó lại kích hoạt in ra vòng lặp a,b,c

Và cứ tiếp tục in ra kết quả cho đến khi kết thúc vòng lặp



# Vòng lặp for lồng nhau: Ví dụ 3

**Ví dụ:** In ra từng phần tử trong mảng lồng nhau

```
list_of_lists = [  
    ['hammerhead', 'great white', 'dogfish'],  
    [0, 1, 2],  
    [9.9, 8.8, 7.7]]
```

```
for list in list_of_lists:  
    for item in list:  
        print(item)
```

**Kết quả:**

```
hammerhead  
great white  
dogfish  
0  
1  
2  
9.9  
8.8  
7.7
```

- Khi sử dụng vòng lặp for lồng nhau, ta có thể in ra từng phần tử riêng lẻ trong mảng



# Tóm tắt bài học

---

- Vòng lặp cho phép thực hiện các câu lệnh lặp đi lặp lại nhiều lần
- Các vòng lặp được Python hỗ trợ: for, while
- Vòng lặp for thường được sử dụng khi đã biết trước số lần lặp
- Vòng lặp while thường được sử dụng khi không biết trước số lần lặp
- Câu lệnh break sẽ kết thúc vòng lặp
- Câu lệnh continue sẽ bỏ qua lần lặp hiện tại
- Nhiều câu lệnh lặp lồng nhau sẽ làm tăng độ phức tạp của thuật toán

# Hướng dẫn

Hướng dẫn làm bài thực hành và bài tập

Chuẩn bị bài tiếp theo