

TÁC GIẢ: 8 SYNC

CỘNG ĐỒNG



[Kevin Nguyễn](#)



[Nhóm Chia Sẻ Công Nghệ](#)



[Nhóm BlockChain](#)



[Tiktok: 8 Sync](#)



[Youtube: 8 Sync Dev](#)



[Zalo](#)

KHÓA HỌC:



[Fullstack Python](#)



[Fullstack Nextjs](#)



[Fullstack Android-IOs](#)

Tài liệu sẽ được cập nhật định kì và thông báo trong group nên các bạn chú ý nhen .



BÀI 9. PYTHON - VÒNG LẶP WHILE

Thông thường, luồng thực thi các bước trong một chương trình máy tính diễn ra từ đầu đến cuối. Tuy nhiên, thay vì chuyển sang bước tiếp theo, nếu luồng được chuyển hướng lại đến bất kỳ bước nào trước đó, đó là một vòng lặp.

VÒNG LẶP WHILE TRONG PYTHON

Một câu lệnh vòng lặp while trong ngôn ngữ lập trình Python thực hiện một câu lệnh mục tiêu lặp đi lặp lại miễn là một biểu thức boolean cụ thể là đúng.

CÚ PHÁP CỦA VÒNG LẶP WHILE

Cú pháp của vòng lặp while trong ngôn ngữ lập trình Python là –



< 8 Sync Dev />

```
while biểu_thức:  
    các câu lệnh
```

Từ khóa while được theo sau bởi một biểu thức boolean, và sau đó là dấu hai chấm, để bắt đầu một khối câu lệnh đã được lùi vào bên trong. Ở đây, các câu lệnh có thể là một câu lệnh duy nhất hoặc một khối câu lệnh với định dạng thụt đồng nhất. Điều kiện có thể là bất kỳ biểu thức nào, và true là bất kỳ giá trị không-zero nào. Vòng lặp lặp lại trong khi biểu thức boolean là true.

Ngay khi biểu thức trở thành false, điều khiển chương trình chuyển đến dòng lệnh ngay sau vòng lặp.

Nếu nó không thể trở thành false, vòng lặp tiếp tục chạy và không dừng lại trừ khi bị dừng một cách buộc bằng cách sử dụng dừng buộc. Một vòng lặp như vậy được gọi là vòng lặp vô hạn, mà là không mong muốn trong một chương trình máy tính.

VÍ DỤ

VÍ DỤ 1:

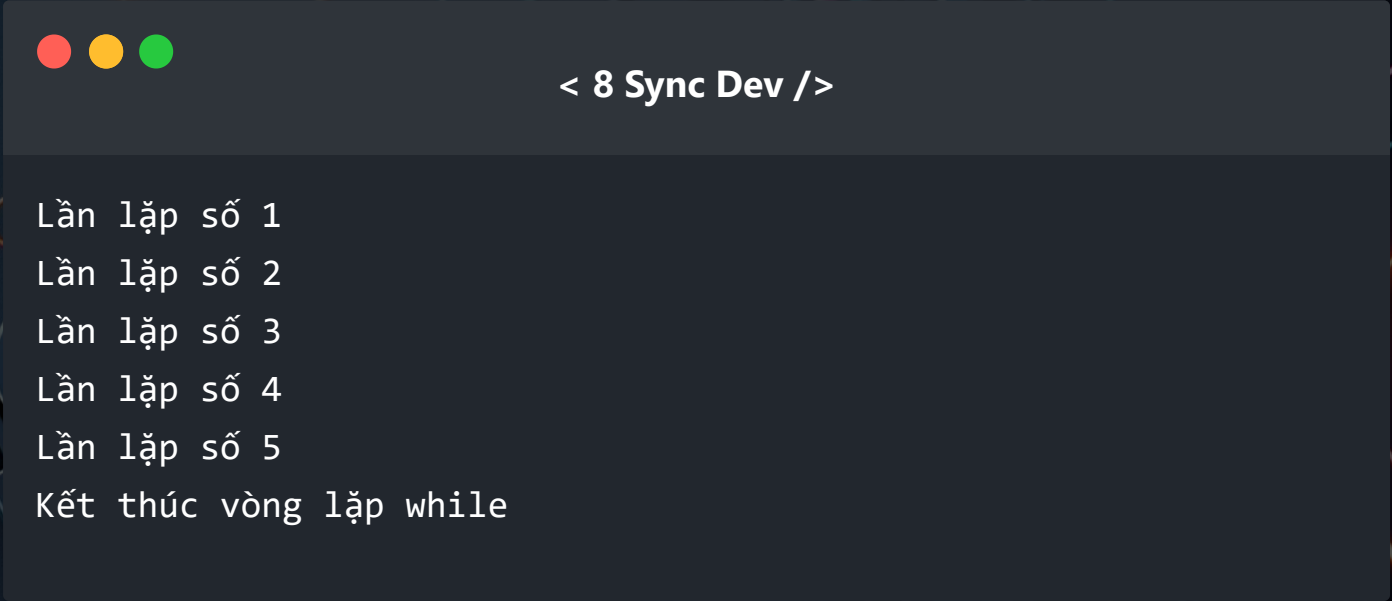


< 8 Sync Dev />

```
count = 0
while count < 5:
    count += 1
    print("Lần lặp số {}".format(count))

print("Kết thúc vòng lặp while")
```

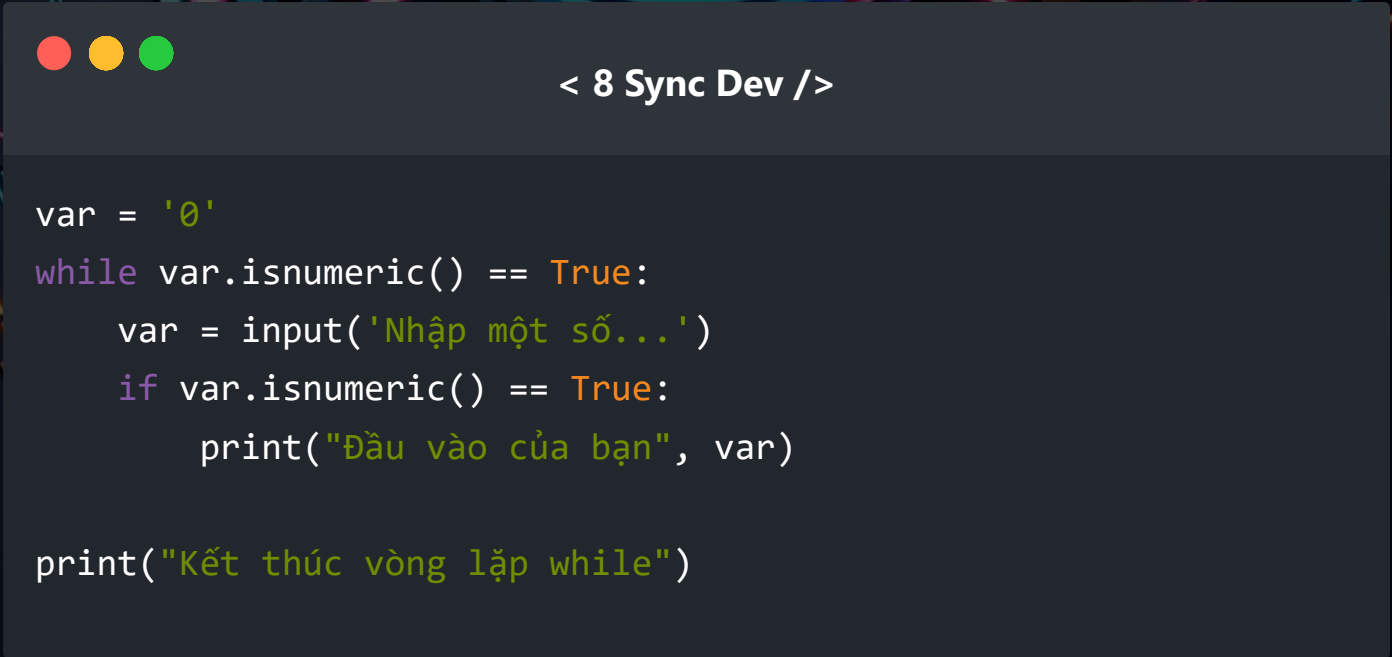
Kết quả khi chạy mã trên sẽ là:



A terminal window with a dark background and a title bar containing three colored circles (red, yellow, green) and the text "< 8 Sync Dev />". The terminal displays the output of the first code block: "Lần lặp số 1", "Lần lặp số 2", "Lần lặp số 3", "Lần lặp số 4", "Lần lặp số 5", and "Kết thúc vòng lặp while".

```
Lần lặp số 1
Lần lặp số 2
Lần lặp số 3
Lần lặp số 4
Lần lặp số 5
Kết thúc vòng lặp while
```

VÍ DỤ 2:



A terminal window with a dark background and a title bar containing three colored circles (red, yellow, green) and the text "< 8 Sync Dev />". The terminal displays the code for the second example, which uses a while loop to repeatedly prompt the user for input until it is numeric.

```
var = '0'
while var.isnumeric() == True:
    var = input('Nhập một số...')
    if var.isnumeric() == True:
        print("Đầu vào của bạn", var)

print("Kết thúc vòng lặp while")
```

Kết quả khi chạy mã trên sẽ là:



< 8 Sync Dev />

```
Nhập một số...10
Đầu vào của bạn 10
Nhập một số...100
Đầu vào của bạn 100
Nhập một số...543
Đầu vào của bạn 543
Nhập một số...qwer
Kết thúc vòng lặp while
```

VÒNG LẶP VÔ HẠN TRONG PYTHON

Một vòng lặp trở thành vòng lặp vô hạn nếu một điều kiện không bao giờ trở thành FALSE. Bạn phải cẩn thận khi sử dụng các vòng lặp while vì có khả năng rằng điều kiện này không bao giờ trở thành giá trị FALSE. Điều này dẫn đến một vòng lặp không bao giờ kết thúc. Một vòng lặp như vậy được gọi là một vòng lặp vô hạn.

Một vòng lặp vô hạn có thể hữu ích trong lập trình máy chủ/máy khách nơi máy chủ cần chạy liên tục để các chương trình máy khách có thể giao tiếp với nó khi cần thiết.

VÍ DỤ



< 8 Sync Dev />

```
var = 1
while var == 1: # Đây tạo ra một vòng lặp vô hạn
    num = int(input("Nhập một số:"))
    print("Bạn đã nhập: ", num)

print("Tạm biệt!")
```

Kết quả khi chạy mã trên sẽ là:



< 8 Sync Dev />

```
Đầu vào số: 20
Bạn đã nhập: 20
Đầu vào số: 29
Bạn đã nhập: 29
Đầu vào số: 3
Bạn đã nhập: 3
Đầu vào số: 11
Bạn đã nhập: 11
Đầu vào số: 22
Bạn đã nhập: 22
Đầu vào số: Nhập vào số nguyên...
```

VÒNG LẶP WHILE-ELSE TRONG PYTHON

Python hỗ trợ việc sử dụng câu lệnh "else" kết hợp với câu lệnh vòng lặp while.

Nếu câu lệnh "else" được sử dụng với một vòng lặp while, câu lệnh "else" được thực thi khi điều kiện trở thành false trước khi điều khiển chuyển sang dòng lệnh chính của chương trình.

VÍ DỤ



< 8 Sync Dev />

```
count = 0
while count < 5:
    count += 1
    print("Lần lặp số {}".format(count))
else:
    print("Vòng lặp while đã kết thúc. Bây giờ đang trong khối
```

```
else")
```

```
print("Kết thúc vòng lặp while")
```

Kết quả khi chạy mã trên sẽ là:



< 8 Sync Dev />

Lần lặp số 1

Lần lặp số 2

Lần lặp số 3

Lần lặp số 4

Lần lặp số 5

Vòng lặp while đã kết thúc. Bây giờ đang trong khối else

Kết thúc vòng lặp while

Ở đây, khi điều kiện của vòng lặp trở thành false, câu lệnh trong khối else được thực thi trước khi điều khiển chuyển sang dòng lệnh chính của chương trình.