

# 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 .



## PYTHON - LUỒNG ĐIỀU KHIỂN

---

Trong Python, luồng điều khiển của chương trình được điều chỉnh bằng các loại câu lệnh điều kiện, vòng lặp và cuộc gọi hàm khác nhau. Mặc định, các chỉ thị trong một chương trình máy tính được thực thi theo cách tuần tự, từ trên xuống dưới hoặc từ đầu đến cuối. Tuy nhiên, các chương trình thực thi theo cách tuần tự như


vậy chỉ có thể thực hiện các nhiệm vụ đơn giản. Chúng ta muốn chương trình có khả năng ra quyết định, để nó thực hiện các bước khác nhau tùy thuộc vào các điều kiện khác nhau.

Hầu hết các ngôn ngữ lập trình bao gồm cả Python đều cung cấp chức năng để kiểm soát luồng thực thi của các chỉ thị. Thông thường, có hai loại câu lệnh kiểm soát luồng trong bất kỳ ngôn ngữ lập trình nào và Python cũng hỗ trợ chúng.

## CÂU LỆNH RA QUYẾT ĐỊNH

Câu lệnh ra quyết định được sử dụng trong các chương trình Python để làm cho chúng có khả năng quyết định xem nhóm các chỉ thị thay thế nào sẽ được thực thi, tùy thuộc vào giá trị của một biểu thức Boolean nhất định.

Sơ đồ dưới đây minh họa cách các câu lệnh ra quyết định hoạt động -

 Câu lệnh ra quyết định

## CÂU LỆNH IF

Python cung cấp các câu lệnh điều khiển if..elif..else như một phần của quyết định. Dưới đây là một ví dụ đơn giản sử dụng if..elif..else. Bạn có thể thử chạy chương trình này với các điểm khác nhau và xác minh kết quả.



< 8 Sync Dev />

```
marks = 80
result = ""
if marks < 30:
    result = "Failed"
elif marks > 75:
    result = "Passed with distinction"
else:
    result = "Passed"

print(result)
```

Kết quả sẽ là:



< 8 Sync Dev />

Passed with distinction

## CÂU LỆNH MATCH

Python hỗ trợ câu lệnh Match-Case, cũng có thể được sử dụng như một phần của quyết định. Dưới đây là một ví dụ đơn giản sử dụng câu lệnh match.



< 8 Sync Dev />

```
def checkVowel(n):  
    match n:  
        case 'a': return "Vowel alphabet"  
        case 'e': return "Vowel alphabet"  
        case 'i': return "Vowel alphabet"  
        case 'o': return "Vowel alphabet"  
        case 'u': return "Vowel alphabet"  
        case _: return "Simple alphabet"  
  
print (checkVowel('a'))  
print (checkVowel('m'))  
print (checkVowel('o'))
```

Kết quả sẽ là:



< 8 Sync Dev />

```
Vowel alphabet  
Simple alphabet  
Vowel alphabet
```



# CÂU LỆNH LẶP HOẶC LẶP ĐIỀU KHIỂN

Hầu hết các quy trình đòi hỏi một nhóm các chỉ thị được lặp đi lặp lại. Trong thuật ngữ lập trình, điều này được gọi là vòng lặp. Thay vì bước tiếp theo, nếu luồng được chuyển hướng về bất kỳ bước trước đó nào, nó tạo thành một vòng lặp.

Sơ đồ dưới đây minh họa cách vòng lặp hoạt động -

## Vòng lặp

Nếu điều khiển quay lại mà không có điều kiện, nó tạo thành một vòng lặp vô hạn không mong muốn vì phần còn lại của mã sẽ không bao giờ được thực thi.

Trong một vòng lặp có điều kiện, việc lặp lại lặp đi lặp lại của một nhóm các chỉ thị tiếp tục cho đến khi một điều kiện nhất định được đáp ứng. Python hỗ trợ một số vòng lặp như vòng lặp for, vòng lặp while mà chúng ta sẽ nghiên cứu trong các chương tiếp theo.

## VÒNG LẶP FOR

Vòng lặp for lặp lại qua các mục của bất kỳ chuỗi nào, chẳng hạn như một danh sách, tuple hoặc một chuỗi.

Dưới đây là một ví dụ sử dụng Vòng lặp For để lặp qua một mảng trong Python:



< 8 Sync Dev />

```
words = ["one", "two", "three"]  
for x in words:  
    print(x)
```

Kết quả sẽ là:



< 8 Sync Dev />

one  
two  
three

## VÒNG LẶP WHILE

Vòng lặp while lặp lại một câu lệnh mục tiêu liên tục miễn là một biểu thức boolean cụ thể được đánh giá là đúng.

Dưới đây là một ví dụ sử dụng Vòng lặp While để in ra 5 số đầu tiên trong Python:



< 8 Sync Dev />

```
i = 1
while i < 6:
    print(i)
    i += 1
```

Kết quả sẽ là:



< 8 Sync Dev />

```
1
2
3
4
5
```

## CÂU LỆNH NHẢY

nhảy

Các câu lệnh nhảy được sử dụng để nhảy đến một câu lệnh cụ thể bằng cách phá vỡ luồng thực thi hiện tại của chương trình. Trong Python, có hai câu lệnh nhảy là break và continue.

## CÂU LỆNH BREAK

Nó chấm dứt vòng lặp hiện tại và tiếp tục thực thi tại câu lệnh tiếp theo.

Ví dụ dưới đây minh họa việc sử dụng câu lệnh break -

```
< 8 Sync Dev />

x = 0

while x < 10:
    print("x:", x)
    if x == 5:
        print("Breaking...")
        break
    x += 1

print("End")
```

Kết quả sẽ là:

```
< 8 Sync Dev />

x: 0
x: 1
x: 2
x: 3
x: 4
x: 5
Breaking...
End
```



# CÂU LỆNH CONTINUE

Nó bỏ qua việc thực thi của khối chương trình và trả lại điều khiển về đầu vòng lặp hiện tại để bắt đầu vòng lặp kế tiếp.

Ví dụ dưới đây minh họa việc sử dụng câu lệnh continue -



< 8 Sync Dev />

```
for letter in "Python":  
    # Bỏ qua khi chữ cái là 'h'  
    if letter == "h":  
        continue  
    print("Current Letter :", letter)
```

Kết quả sẽ là:



< 8 Sync Dev />

```
Current Letter : P  
Current Letter : y  
Current Letter : t  
Current Letter : o  
Current Letter : n
```

Như vậy, các câu lệnh điều khiển và vòng lặp là các công cụ quan trọng giúp kiểm soát luồng thực thi của chương trình Python và tạo điều kiện cho quyết định và lặp lại các phần mã cần thiết.