

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 CONTINUE STATEMENT

Câu lệnh **continue** trong Python được sử dụng để bỏ qua việc thực thi của khối mã và trả lại quyền điều khiển cho đầu vòng lặp hiện tại để bắt đầu vòng lặp tiếp theo. Khi gặp phải, vòng lặp bắt đầu vòng lặp tiếp theo mà không thực thi các câu lệnh còn lại trong vòng lặp hiện tại.

Câu lệnh `continue` có thể được sử dụng cả trong vòng lặp `while` và `for`.

CÚ PHÁP CỦA CÂU LỆNH CONTINUE



< 8 Sync Dev />

```
continue
```

VÍ DỤ VỀ CÂU LỆNH CONTINUE

Bây giờ chúng ta hãy xem một ví dụ để hiểu cách câu lệnh `continue` hoạt động trong Python –



< 8 Sync Dev />

```
for letter in 'Python': # Ví dụ 1
    if letter == 'h':
        continue
    print ('Chữ hiện tại :', letter)

var = 10 # Ví dụ 2
while var > 0:
    var = var -1
    if var == 5:
        continue
    print ('Giá trị biến hiện tại :', var)
print ("Tạm biệt!")
```

Khi mã trên được thực thi, nó sẽ tạo ra kết quả như sau:



< 8 Sync Dev />

```
Chữ hiện tại : P
Chữ hiện tại : y
Chữ hiện tại : t
Chữ hiện tại : o
Chữ hiện tại : n
Giá trị biến hiện tại : 9
Giá trị biến hiện tại : 8
Giá trị biến hiện tại : 7
Giá trị biến hiện tại : 6
Giá trị biến hiện tại : 4
Giá trị biến hiện tại : 3
Giá trị biến hiện tại : 2
Giá trị biến hiện tại : 1
Giá trị biến hiện tại : 0
Tạm biệt!
```

SỬ DỤNG CÂU LỆNH CONTINUE TRONG VÒNG LẶP WHILE TRONG PYTHON

Câu lệnh `continue` trong Python được sử dụng cả trong vòng lặp `for` cũng như vòng lặp `while` để bỏ qua việc thực thi của vòng lặp hiện tại và chuyển quyền điều khiển của chương trình sang vòng lặp tiếp theo.

VÍ DỤ: KIỂM TRA CÁC ƯỚC SỐ NGUYÊN TỐ

Đoạn mã sau sử dụng `continue` để tìm các ước số nguyên tố của một số được cung cấp. Để tìm các ước số nguyên tố, chúng ta cần lặp lại việc chia số đã cho bắt đầu từ 2, tăng giá trị của ước số và tiếp tục quá trình tương tự cho đến khi đầu vào giảm xuống còn 1.

Thuật toán để tìm các ước số nguyên tố như sau:

- Chấp nhận đầu vào từ người dùng (n)
- Đặt ước số (d) thành 2

- Thực hiện các bước sau cho đến khi $n > 1$:
 - Kiểm tra xem số đã cho (n) có chia hết cho ước số (d) hay không.
 - Nếu $n \% d == 0$:
 - In dưới dạng một ước số
 - Đặt giá trị mới của n là n/d
 - Tiếp tục từ bước 4
 - Nếu không:
 - Tăng giá trị của d lên 1
 - Tiếp tục từ bước 3

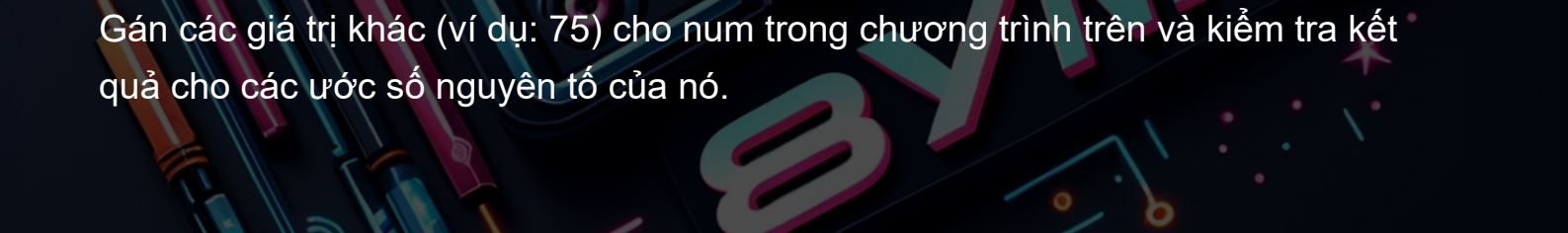
Dưới đây là đoạn mã Python cho mục đích đó:

```
num = 60
print ("Các ước số nguyên tố cho: ", num)
d = 2
while num > 1:
    if num % d == 0:
        print (d)
        num = num / d
        continue
    d = d + 1
```

Khi thực thi, mã này sẽ tạo ra kết quả sau:

```
< 8 Sync Dev />

Các ước số nguyên tố cho: 60
2
2
3
5
```



Gán các giá trị khác (ví dụ: 75) cho num trong chương trình trên và kiểm tra kết quả cho các ước số nguyên tố của nó.