

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 6. PYTHON - LIST COMPREHENSION

LIST COMPREHENSION TRONG PYTHON

List comprehension là một công cụ lập trình rất mạnh mẽ. Nó tương tự như cách xây dựng tập hợp trong toán học. Đây là cách ngắn gọn để tạo ra một danh sách

mới bằng cách thực hiện một loại quy trình nào đó trên từng mục trong danh sách hiện tại. List comprehension nhanh hơn đáng kể so với việc xử lý một danh sách bằng vòng lặp for.

VÍ DỤ VỀ LIST COMPREHENSION TRONG PYTHON

Giả sử chúng ta muốn tách mỗi chữ cái trong một chuỗi và đưa tất cả các chữ cái không phải là nguyên âm vào một đối tượng danh sách. Chúng ta có thể làm điều này bằng một vòng lặp for như sau:

```
< 8 Sync Dev />

chars=[]
for ch in '8SyncDev':
    if ch not in 'aeiou':
        chars.append(ch)
print(chars)
```

Đối tượng danh sách chars sẽ được hiển thị như sau:

```
< 8 Sync Dev />

['8', 'S', 'y', 'n', 'c', 'D', 'e', 'v']
```

KỸ THUẬT LIST COMPREHENSION TRONG PYTHON

Chúng ta có thể dễ dàng có được kết quả tương tự bằng một kỹ thuật list comprehension. Một cách sử dụng chung của list comprehension là như sau:

```
listObj = [x for x in iterable]
```

Áp dụng điều này, đối tượng danh sách chars có thể được xây dựng bằng câu lệnh sau:



< 8 Sync Dev />

```
chars = [char for char in '8SyncDev' if char not in 'kkkk']  
print(chars)
```

Danh sách chars sẽ được hiển thị như trước:



< 8 Sync Dev />

```
['8', 'S', 'y', 'n', 'c', 'D', 'e', 'v']
```

VÍ DỤ

Ví dụ sau sử dụng list comprehension để xây dựng một danh sách các bình phương của các số từ 1 đến 10



< 8 Sync Dev />

```
squares = [x*x for x in range(1,11)]  
print(squares)
```

Đối tượng danh sách squares là:



< 8 Sync Dev />

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

NESTED LOOPS TRONG LIST COMPREHENSION CỦA PYTHON

Trong ví dụ sau, tất cả các kết hợp của các mục từ hai danh sách dưới dạng một tuple được thêm vào danh sách thứ ba.



< 8 Sync Dev />

```
list1=[1,2,3]
list2=[4,5,6]
CombLst=[(x,y) for x in list1 for y in list2]
print(CombLst)
```

Nó sẽ tạo ra đầu ra sau:



< 8 Sync Dev />

```
[(1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5),
(3, 6)]
```

ĐIỀU KIỆN TRONG LIST COMPREHENSION CỦA PYTHON

Câu lệnh sau sẽ tạo ra một danh sách các số chẵn từ 1 đến 20.



< 8 Sync Dev />

```
list1=[x for x in range(1,21) if x%2==0]  
print(list1)
```

Nó sẽ tạo ra đầu ra sau:



< 8 Sync Dev />

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```