

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 3. PYTHON - GIẢI NÉN CÁC PHẦN TỬ CỦA TUPLE

Thuật ngữ "giải nén" đề cập đến quá trình phân tích các phần tử tuple thành các biến riêng lẻ. Trong Python, dấu ngoặc đơn là các dấu phân cách mặc định cho một biểu diễn chữ của đối tượng chuỗi.

Các câu lệnh sau để khai báo một tuple là giống nhau.



< 8 Sync Dev />

```
t1 = (x,y)
t1 = x,y
type (t1)
<class 'tuple'>
```

Ví dụ:

Để lưu trữ các phần tử tuple trong các biến riêng lẻ, hãy sử dụng nhiều biến trên bên trái của toán tử gán, như được thể hiện trong ví dụ sau:



< 8 Sync Dev />

```
tup1 = (10,20,30)
x, y, z = tup1
print ("x: ", x, "y: ", y, "z: ", z)
```

Sẽ cho ra kết quả sau:



< 8 Sync Dev />

```
x: 10 y: 20 z: 30
```

Đó là cách mà tuple được giải nén thành các biến riêng lẻ.

Trong ví dụ trên, số lượng biến bên trái của toán tử gán bằng số phần tử trong tuple. Điều gì sẽ xảy ra nếu số lượng không bằng số phần tử?

VALUEERROR KHI GIẢI NÉN MỘT TUPLE

Nếu số lượng biến nhiều hơn hoặc ít hơn so với độ dài của tuple, Python sẽ ném ra một ValueError.

Ví dụ:



< 8 Sync Dev />

```
tup1 = (10,20,30)
x, y = tup1
x, y, p, q = tup1
```

Sẽ cho ra kết quả sau:



< 8 Sync Dev />

```
x, y = tup1
^^^^
ValueError: too many values to unpack (expected 2)
x, y, p, q = tup1
^^^^^^^^^^
ValueError: not enough values to unpack (expected 4, got 3)
```

GIẢI NÉN CÁC PHẦN TỬ CỦA TUPLE SỬ DỤNG DẤU * (ASTERISK)

Trong trường hợp như vậy, ký hiệu "*" được sử dụng để giải nén. Tiền tố "*" cho "y", như được thể hiện dưới đây –

VÍ DỤ 1



< 8 Sync Dev />


```
tup1 = (10,20,30)
x, *y = tup1
print ("x: ", x, "y: ", y)
```

Sẽ cho ra kết quả sau:



< 8 Sync Dev />

```
x: 10 y: [20, 30]
```

Giá trị đầu tiên trong tuple được gán cho "x", và phần còn lại được gán cho "y" và trở thành một danh sách.

VÍ DỤ 2

Trong ví dụ này, tuple chứa 6 giá trị và các biến để giải nén là 3. Chúng ta thêm "*" vào biến thứ hai.



< 8 Sync Dev />

```
tup1 = (10,20,30, 40, 50, 60)
x, *y, z = tup1
print ("x: ", x, "y: ", y, "z: ", z)
```

Sẽ cho ra kết quả sau:



< 8 Sync Dev />

```
x: 10 y: [20, 30, 40, 50] z: 60
```

Ở đây, giá trị được giải nén trong "x" và "z" trước, và sau đó các giá trị còn lại được gán cho "y" dưới dạng một danh sách.

VÍ DỤ 3

Điều gì sẽ xảy ra nếu chúng ta thêm "*" vào biến đầu tiên?



< 8 Sync Dev />

```
tup1 = (10,20,30, 40, 50, 60)
*x, y, z = tup1
print ("x: ", x, "y: ", y, "z: ", z)
```

Sẽ cho ra kết quả sau:



< 8 Sync Dev />

```
x: [10, 20, 30, 40] y: 50 z: 60
```

Ở đây, một lần nữa, tuple được giải nén sao cho các biến riêng lẻ lấy giá trị trước, để lại các giá trị còn lại cho danh sách "x".