

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 .



SAO CHÉP TỪ ĐIỂN TRONG PYTHON

Trong Python, vì một biến chỉ là một nhãn hoặc tham chiếu đến một đối tượng trong bộ nhớ, toán tử gán đơn giản không tạo ra bản sao của đối tượng.

Ví dụ 1:



Trong ví dụ này, chúng ta có một từ điển "d1" và chúng ta gán nó cho một biến khác "d2". Nếu "d1" được cập nhật, các thay đổi cũng phản ánh trong "d2".



< 8 Sync Dev />

```
d1 = {"a":11, "b":22, "c":33}
d2 = d1
print ("id:", id(d1), "dict: ",d1)
print ("id:", id(d2), "dict: ",d2)

d1["b"] = 100
print ("id:", id(d1), "dict: ",d1)
print ("id:", id(d2), "dict: ",d2)
```

Kết quả:



< 8 Sync Dev />

```
id: 2215278891200 dict: {'a': 11, 'b': 22, 'c': 33}
id: 2215278891200 dict: {'a': 11, 'b': 22, 'c': 33}
id: 2215278891200 dict: {'a': 11, 'b': 100, 'c': 33}
id: 2215278891200 dict: {'a': 11, 'b': 100, 'c': 33}
```

Để tránh điều này và tạo một bản sao nông (shallow copy) của một từ điển, sử dụng phương thức `copy()` thay vì toán tử gán.

Ví dụ 2:



< 8 Sync Dev />

```
d1 = {"a":11, "b":22, "c":33}
d2 = d1.copy()
print ("id:", id(d1), "dict: ",d1)
print ("id:", id(d2), "dict: ",d2)
d1["b"] = 100
```

```
print ("id:", id(d1), "dict: ",d1)
print ("id:", id(d2), "dict: ",d2)
```

Kết quả:



< 8 Sync Dev />

Khi "d1" được cập nhật, "d2" sẽ không thay đổi bây giờ vì "d2" là bản sao của đối tượng từ điển, không chỉ là một tham chiếu.

```
id: 1586671734976 dict: {'a': 11, 'b': 22, 'c': 33}
id: 1586673973632 dict: {'a': 11, 'b': 22, 'c': 33}
id: 1586671734976 dict: {'a': 11, 'b': 100, 'c': 33}
id: 1586673973632 dict: {'a': 11, 'b': 22, 'c': 33}
```

Trong ví dụ thứ hai, khi "d1" được cập nhật, "d2" không thay đổi, vì "d2" là một bản sao của đối tượng từ điển, không chỉ là một tham chiếu.