

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 5. ĐỐI TƯỢNG XEM TỪ ĐIỂN TRONG PYTHON

Trong Python, các phương thức `items()`, `keys()` và `values()` của lớp `dict` trả về các đối tượng xem (view objects). Các xem này được cập nhật động mỗi khi có bất kỳ thay đổi nào trong nội dung của đối tượng từ điển gốc của chúng.

PHƯƠNG THỨC ITEMS()

Phương thức items() trả về một đối tượng dict_items xem. Nó chứa một danh sách các bộ, mỗi bộ được tạo thành từ các cặp key, value tương ứng.

Cú Pháp:



< 8 Sync Dev />

```
obj = dict.items()
```

Giá Trị Trả Về:

Phương thức items() trả về đối tượng dict_items, đó là một xem động của các bộ (key, value).

Ví dụ:



< 8 Sync Dev />

```
numbers = {10: "Ten", 20: "Twenty", 30: "Thirty", 40: "Forty"}
obj = numbers.items()
print('Kiểu của obj: ', type(obj))
print(obj)
print("Cập nhật từ điển numbers")
numbers.update({50: "Fifty"})
print("Xem được cập nhật tự động")
print(obj)
```

Kết quả:



< 8 Sync Dev />

```
Kiểu của obj: <class 'dict_items'>
dict_items([(10, 'Ten'), (20, 'Twenty'), (30, 'Thirty'), (40, 'Forty')])
```

Cập nhật từ điển numbers

Xem được cập nhật tự động

```
dict_items([(10, 'Ten'), (20, 'Twenty'), (30, 'Thirty'), (40, 'Forty'), (50, 'Fifty')])
```

PHƯƠNG THỨC KEYS()

Phương thức `keys()` của lớp `dict` trả về đối tượng `dict_keys`, đó là một danh sách của tất cả các key được xác định trong từ điển. Đây là một đối tượng xem, vì nó được cập nhật tự động mỗi khi có bất kỳ hành động cập nhật nào được thực hiện trên đối tượng từ điển.

Cú Pháp:



< 8 Sync Dev />

```
obj = dict.keys()
```

Giá Trị Trả Về:

Phương thức `keys()` trả về đối tượng `dict_keys`, đó là một xem của các keys trong từ điển.

Ví dụ:



< 8 Sync Dev />

```
numbers = {10: "Ten", 20: "Twenty", 30: "Thirty", 40: "Forty"}
obj = numbers.keys()
print('Kiểu của obj: ', type(obj))
print(obj)
print("Cập nhật từ điển numbers")
numbers.update({50: "Fifty"})
print("Xem được cập nhật tự động")
print(obj)
```


Kết quả:



< 8 Sync Dev />

```
Kiểu của obj: <class 'dict_keys'>
dict_keys([10, 20, 30, 40])
Cập nhật từ điển numbers
Xem được cập nhật tự động
dict_keys([10, 20, 30, 40, 50])
```

PHƯƠNG THỨC VALUES()

Phương thức values() trả về một xem của tất cả các giá trị có trong từ điển. Đối tượng là kiểu dict_values, được cập nhật tự động.

Cú Pháp:



< 8 Sync Dev />

```
obj = dict.values()
```

Giá Trị Trả Về:

Phương thức values() trả về một xem dict_values của tất cả các giá trị có trong từ điển.

Ví dụ:

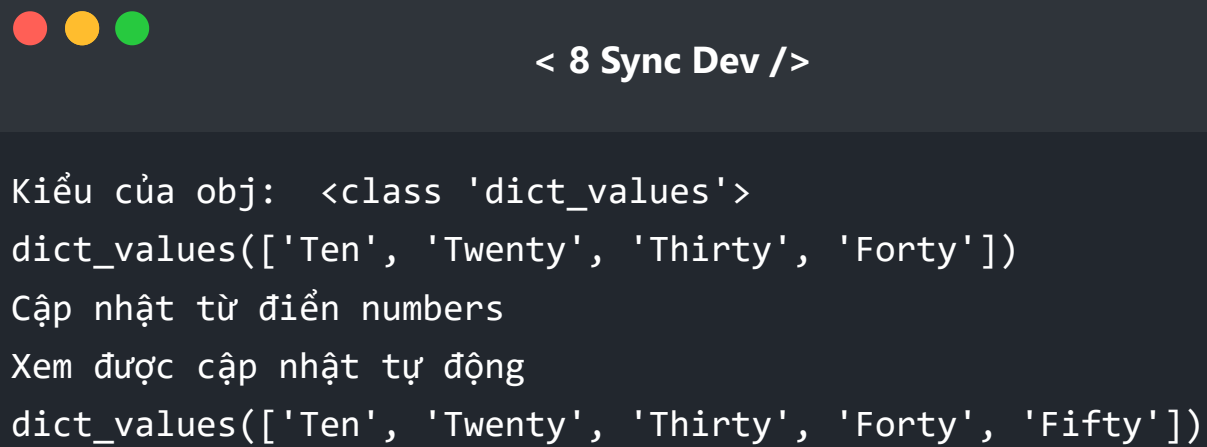


< 8 Sync Dev />

```
numbers = {10: "Ten", 20: "Twenty", 30: "Thirty", 40: "Forty"}
obj = numbers.values()
print('Kiểu của obj: ', type(obj))
print(obj)
print("Cập nhật từ điển numbers")
numbers.update({50: "Fifty"})
```

```
print("Xem được cập nhật tự động")  
print(obj)
```

Kết quả:



```
Kiểu của obj: <class 'dict_values'>  
dict_values(['Ten', 'Twenty', 'Thirty', 'Forty'])  
Cập nhật từ điển numbers  
Xem được cập nhật tự động  
dict_values(['Ten', 'Twenty', 'Thirty', 'Forty', 'Fifty'])
```

Các phương thức `items()`, `keys()` và `values()` cho phép bạn truy cập vào dữ liệu trong từ điển một cách linh hoạt và tiện lợi, đồng thời tự động cập nhật các xem của chúng khi có sự thay đổi trong từ điển gốc.