

# Kiểu dữ liệu Từ điển

TS. Nguyễn Tuấn Anh

Trường Đại học Công nghệ Thông tin & Truyền thông

*Khoa Công nghệ Thông tin*

Tháng 2 năm 2023



# Khái niệm Từ điển

Từ điển (**Dict**) là một danh sách các phần tử, mỗi phần tử gồm có hai thành phần, một thành phần được gọi là khóa (**key**), thành phần còn lại gọi là giá trị (**value**) của thành phần đó.

Các phần tử trong từ điển có **khóa** khác nhau, còn giá trị thì có thể bằng hoặc giống nhau.

**Khóa** của mỗi phần tử thì không thể thay đổi được, giá trị của phần tử có thể thay đổi được.



# Khái niệm Từ điển

## Khai báo:

```
<biến từ điển> = {  
<key 0>:<Data 0>,  
<key 1>:<Data 1>, ...,  
<key n-1>:<Data n-1> }
```

## Ví dụ:

```
myDict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964}
```



# Khái niệm Từ điển

Hàm `len()` trả về số phần tử trong từ điển:

```
n = len(myDict)
```

Kiểu dữ liệu của từ điển:

```
myDict = {  
    "brand": "Ford",  
    "electric": False,  
    "year": 1964,  
    "colors": ["red", "white", "blue"]  
}
```



# Khái niệm Từ điển

Phương thức type trả về kiểu dữ liệu:

```
myDict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
  
print(type(myDict))
```



# Khái niệm Từ điển

Phương thức khởi tạo từ điển dict():

```
myDict = dict(name = "Nguyễn Văn A", age = 44)  
print(myDict)
```

Kết quả:

```
{'name': 'Nguyễn Văn A', 'age': 44}
```



# Truy cập các phần tử trong Từ điển

Truy cập các phần tử thông qua khóa:

```
myDict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
x = thisdict["model"]
```

Phương thức `get()` trả về giá trị theo khóa:

```
x = myDict.get("model")
```



# Truy cập các phần tử trong Từ điển

Phương thức `keys()` trả về danh sách toàn bộ khóa trong từ điển:

```
x = myDict.keys()
```

Phương thức `values()` trả về danh sách toàn bộ các giá trị trong từ điển:

```
x = myDict.values()
```

Phương thức `items()` trả về danh sách các phần tử trong Từ điển:

```
x = myDict.items()
```





# Truy cập các phần tử trong Từ điển

- Hàm kiểm tra khóa có tồn tại không

```
if "model" in myDict:  
    print("'model' là key trong myDict")
```



# Thay đổi giá trị trong Từ điển

Có thể thay đổi giá trị của các phần tử thông qua khóa:

Ví dụ:

```
myDict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
myDict["year"] = 2018
```



# Thêm một phần tử vào từ điển:

Để thêm một phần tử vào từ điển, bạn sử dụng cặp khóa và giá trị, với khóa là chỉ số mới chưa tồn tại trong từ điển.

```
myDict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
myDict["color"] = "red"
```



# Xóa một phần tử ra khỏi từ điển

- Sử dụng phương thức pop kèm theo khóa:

```
myDict.pop("model")
```

- Phương thức popitem() xóa phần tử được thêm vào sau cùng:

```
myDict.popitem()
```

- Sử dụng từ khóa del để loại bỏ một phần tử ra khỏi từ điển:

```
del myDict["model"]
```



# Xóa một phần tử ra khỏi từ điển

- Phương thức `clear()` cho phép xóa tất cả các phần tử của từ điển, lúc này từ điển vẫn tồn tại nhưng không có phần tử nào.

```
myDict.clear()
```



# Duyệt các phần tử trong từ điển

In toàn bộ khóa trong từ điển

```
for x in myDict:  
    print(x)
```

In toàn bộ giá trị trong từ điển

```
for x in myDict:  
    print(myDict[x])
```



# Duyệt các phần tử trong từ điển

Sử dụng phương thức `values()` trả về toàn bộ giá trị:

```
for x in myDict.values():  
    print(x)
```

Sử dụng phương thức `keys()` trả về toàn bộ khóa:

```
for x in myDict.keys():  
    print(x)
```

Phương thức `items()` trả về toàn bộ các phần tử trong từ điển bao gồm khóa và giá trị:

```
for x, y in myDict.items():  
    print(x, y)
```



# Sao chép từ điển

Phương thức `copy()` tạo bản sao của từ điển đã có

```
myDict1 = myDict.copy()  
myDict2 = dict(myDict)
```

