(Ghi chú)

Ngôn ngữ lập trình Python

**Thực hiện: Nguyễn Văn Hào**

**Email:** [hao2205tb@gmail.com](mailto:hao2205tb@gmail.com)

(Tài liệu được biên xoạn lại từ nhiều nguồn và không dùng trong mục đích kinh doanh kiếm tiền)

# Mục lục

[Mục lục 1](#_Toc128349192)

[Dictionary 2](#_Toc128349193)

[Hàm khác 2](#_Toc128349194)

[Min() 2](#_Toc128349195)

[Max() 2](#_Toc128349196)

[Sorted() 3](#_Toc128349197)

[Tham khảo 3](#_Toc128349198)

# Dictionary

Khái quát

<https://laptrinhcanban.com/python/nhap-mon-lap-trinh-python/dictionary-trong-python/dictionary-trong-python-la-gi/>

## Khởi tạo

Cấu trúc dict:

{ key1 : value1 , key2 : value2,… }

VD:

# Tạo dict bằng cách thông thường --------------------------------------

dict1 = {"name": "hao", "age":25}

dict2 = {1: "hello", 2: "world", 3: "!"}

dict3 = {"0": dict1, "1": dict2}

dict4 = {"0": "line1",

         "1": "line2",

         "3": "line3"}

# tạo dict bằng dict() ---------------------------------------------------

dict5 = dict(HaNoi=900, NinhBinh=370, ThanhHoa=250, NamDinh=230)

print(dict5)

#>> {'HaNoi': 900, 'NinhBinh': 370, 'ThanhHoa': 250, 'NamDinh': 230}

list1=[ ("HaNoi",900), ("NinhBinh",370), ("ThanhHoa",250), ("NamDinh",230)]

dict6 = dict(list1)

print(dict6)

#>> {'HaNoi': 900, 'NinhBinh': 370, 'ThanhHoa': 250, 'NamDinh': 230}

# tạo dict bằng dict() và zip() -------------------------------------------

keys = ['HaNoi', 'NinhBinh', 'ThanhHoa', 'NamDinh']

values = [900, 370, 250, 230]

dict7 = dict(zip(keys, values))

print(dict7)

#>> {'HaNoi': 900, 'NinhBinh': 370, 'ThanhHoa': 250, 'NamDinh': 230}

# tạo dict chỉ chứa keys với dict.fromeKeys --------------------------------

# values = None

dict8 = dict.fromkeys(['HaNoi', 'NinhBinh', 'ThanhHoa', 'NamDinh'])

print(dict8)

#>> {'HaNoi': None, 'NinhBinh': None, 'ThanhHoa': None, 'NamDinh': None}

# Đặt values mặc định

dict9 = dict.fromkeys(['HaNoi', 'NinhBinh', 'ThanhHoa', 'NamDinh'], "OK")

print(dict9)

#>> {'HaNoi': 'OK', 'NinhBinh': 'OK', 'ThanhHoa': 'OK', 'NamDinh': 'OK'}

## Đọc và thay đổi giá trị

### Sử dụng key - value

# Tạo dict bằng cách thông thường --------------------------------------

dict1 = {1: "name", 2: "age", "0": "this"}

print(dict1["0"])

#>> this

print(dict1[1])

#>> name

dict1[1] = "new\_name"

print(dict1)

#>> {1: 'new\_name', 2: 'age', '0': 'this'}

dict1[3] = "salary"

print(dict1)

#>> {1: 'new\_name', 2: 'age', '0': 'this', 3: 'salary'}

### Phương thức get()

Get() xử lý trường hợp key không tồn tại.

VD:

mydict = {"NB":"Ninh Bình", "ND":"Nam Định", "TH":"Thanh Hóa"}

print(mydict.get("NB"))

#>>Ninh Bình

# Chỉ định default

print(mydict.get("TK","Key không tồn tại"))

#>>Key không tồn tại

# Lược bỏ default

print(mydict.get("TK"))

#>>None

Có thể thấy kể cả khi key không tồn tại, cách lấy giá trị bằng cách chỉ định khóa trong dictionary python bằng phương thức get() cũng sẽ không trả về lỗi.

### Phương thức keys() | lấy tất cả các khóa

mydict = {"L":"Lemon", "O":"Orage", "G":"Grapes"}

print(mydict.keys())

#>> dict\_keys(['L', 'O', 'G'])

Cách tương tự để lấy tất cả các keys | sử dụng list.

mydict = {"L":"Lemon", "O":"Orage", "G":"Grapes"}

keys=list(mydict)

print(keys)

#>> ['L', 'O', 'G']

### Phương thức values() | lấy tất cả các giá trị

mydict = {"L":"Lemon", "O":"Orage", "G":"Grapes"}

print(mydict.values())

#>> dict\_values(['Lemon', 'Orage', 'Grapes'])

### Phương thức items() | lấy tất cả khoá – giá trị

mydict = {"L":"Lemon", "O":"Orage", "G":"Grapes"}

print(mydict.items())

#>> dict\_items([('L', 'Lemon'), ('O', 'Orage'), ('G', 'Grapes')])

## Thêm phần tử

### Setdefault()

Cú pháp:

dict.setdefault(key, value)

setdefault dùng thêm cặp key – value mới vào dict.

Trong trường hợp key tồn tại trong dict, sẽ không có gì thay đổi với dict.

VD:

dict1 = {0:"0", 1:"1"}

dict1.setdefault(2, "2")

print(dict1)

#>> {0: '0', 1: '1', 2: '2'}

dict1.setdefault(0, "0000")

print(dict1)

#>> {0: '0', 1: '1', 2: '2'}

### Update()

Thêm một dict và dict khác

dict1 = {0:"0", 1:"1"}

dict1.update({2:"2"})

print(dict1)

#>> {0: '0', 1: '1', 2: '2'}

dict1.update({2:"3"})

print(dict1)

#>> {0: '0', 1: '1', 2: '3'}

dict1.update({3:"3", 4:"4"})

print(dict1)

#>> {0: '0', 1: '1', 2: '3', 3: '3', 4: '4'}

## Thay đổi giá trị

C1: sử dụng key-value (xem phần đọc và thay đôi giá trị)

C2: sử dụng update() (xem phần thêm phần tử)

C3: sử dụng \*\* và tạo một dict mới.

VD:

my\_dict = { 'HN': 'Hà Nội' , 'NB': "Ninh Bình", 'ND': "Nam Định"}

my\_dict = { \*\*my\_dict, 'NB': 'ninh bình'}

print(my\_dict)

#>> {'HN': 'Hà Nội', 'NB': 'ninh bình', 'ND': 'Nam Định'}

Phương pháp này cho phép bạn thay đổi giá trị của từ điển bằng cách giải nén từ điển đó bằng toán tử \* và thêm một hoặc nhiều cặp khóa-giá trị mà bạn muốn thay đổi từ điển.

## Len(): đếm số lượng phần tử

mydict = {"NB":"Ninh Bình", "ND":"Nam Định", "TH":"Thanh Hóa"}

print(len(mydict))

#>> 3

## Kiểm tra key - value có tồn tại trong dictionary hay không

### Kiểm tra key

Cú pháp:

key in dict

Hoặc

key in dict.keys()

Nếu khóa tồn tại trong dictionary, kết quả True sẽ được trả về và ngược lại, nếu khóa không tồn tại, False sẽ được trả về.

VD:

mydict = {"L":"Lemon", "O":"Orage", "G":"Grapes"}

print("O" in mydict)

#>> True

print("O" in mydict.keys())

#>> True

print("P" in mydict)

#>> False

print("P" in mydict.keys())

#>> False

### Kiểm tra value

Cú pháp:

key in dict.values()

Nếu value tồn tại trong dictionary, kết quả True sẽ được trả về và ngược lại, nếu khóa không tồn tại, False sẽ được trả về.

mydict = {"L":"Lemon", "O":"Orage", "G":"Grapes"}

print("Orage" in mydict.values())

#>> True

print("Melon" in mydict.values())

#>> False

## Xoá phần tử

### Pop()

Lấy ra một phần tử và xoá nó khỏi dict

Cú pháp:

dict.pop(key[, default])

mydict = {"L":"Lemon", "O":"Orage", "G":"Grapes"}

print(mydict.pop("L"))

#>> Lemon

print(mydict)

#>> {'O': 'Orage', 'G': 'Grapes'}

# khoá không tồn tại

print(mydict.pop("L", "No Key"))

#>> No Key

print(mydict)

#>> {'O': 'Orage', 'G': 'Grapes'}

### Del

Xoá phần tử

Cú pháp:

del dic[key]

mydict = {"A":"Apple", "L":"Lemon", "O":"Orange"}

print(mydict)

#>> {"A":"Apple", "L":"Lemon", "O":"Orange"}

del mydict["L"]

print(mydict)

#>> {'A': 'Apple', 'O': 'Orange'}

# Xoá nhiều phần tử

dict = {'red':1, 'blue':2, 'yellow':3}

del dict['red'], dict['yellow']

print(dict)

#>>{'blue': 2}

### Popitem()

Lấy và xoá phần tử cuối cùng

Cú pháp:

dict.popitem()

dict = {'red':1, 'blue':2, 'yellow':3}

popped = dict.popitem()

print(popped)

#>> ('yellow', 3)

print(dict)

#>> {'red': 1, 'blue': 2}

### Clear()

Xoá toàn bộ phần tử

Cú pháp:

Dict.clear()

mydict = {"A":"Apple", "L":"Lemon", "O":"Orange"}

mydict.clear()

print(mydict)

#>> {}

# Tuple

# Hàm khác

## Min()

Tìm phần tử có giá trị nhỏ nhất.

Cú pháp:

min(iterable)

VD:

list1 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

print(min(list1))

#>> 0

list2 = ["hello", "world", "there", "are"]

print(min(list2))

#>> are

Với chuỗi, so sánh giá trị theo bảng mã UNICODE

## Max()

Tìm phần tử có giá trị lớn nhất

Cú pháp:

max(iterable)

VD:

list1 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

print(max(list1))

#>> 9

list2 = ["hello", "world", "there", "are"]

print(max(list2))

#>> world

Với chuỗi, so sánh giá trị theo bảng mã UNICODE

## Sorted()

Xắp xếp các phần tử của iterable.

Cú pháp:

sorted(iterable, reverse=False)

Trong đó:

* reverse = False: Xắp xếp theo chiều tăng dần
* reverse = True: Xắp xếp theo chiều giảm dần

VD:

list1 = [0, 2, 3, 1, 2, 4, 6, 4]

print(sorted(list1))

#>> [0, 1, 2, 2, 3, 4, 4, 6]

print(sorted(list1, reverse=True))

#>> [6, 4, 4, 3, 2, 2, 1, 0]

# Tham khảo

Lập trình căn bản:

<https://laptrinhcanban.com/cpp/lap-trinh-cpp-co-ban/>