DICTIONARY

Dictionaries là một cấu trúc dữ liệu thuộc dạng associative array hay hashmap.

Bạn có thể tưởng tượng dict (viết tắt) là một cấu trúc dữ liệu mã mỗi phần tử của nó sẽ là một ánh xạ từ một khóa (key) sang một giá trị tương ứng (value).

Ví dụ:

Keys	Values
'name'	'28tech'
'job'	'dev'
'city'	'HCM'
'salary'	'500'
'web'	'28tech.com.vn'

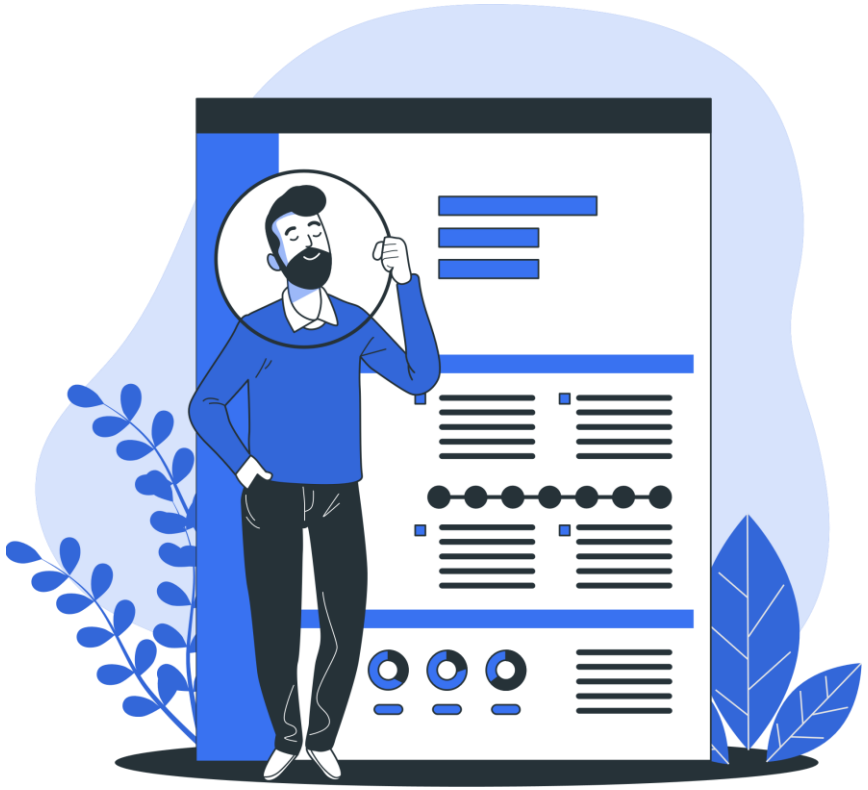
1. Tạo dict:

Để tạo dict bạn liệt kê các cặp key:value phân cách nhau bởi dấu phẩy và đặt giữa đóng mở ngoặc nhọn. Mỗi cặp key:value được đặt cách nhau dấu :

Tạo dict lưu thông tin

EXAMPLE

```
infor = {
    'name' : '28tech',
    'job' : 'dev',
    'salary' : 500,
    'web' : '28tech.com.vn',
    'city' : 'HCM'
}
```



1. Tạo dict:

Tạo dict bằng dict() constructor

Bạn có thể tạo ra dict bằng cách convert cặp 2 giá trị thành một item của dict, khi đó giá trị thứ nhất làm key, giá trị thứ 2 làm value.

EXAMPLE

```
a = [['name', '28tech'], ['job', 'dev'], ('web', '28tech.com.vn')]
b = dict(a)
print(b)
```

OUTPUT

```
{'name': '28tech', 'job': 'dev', 'web': '28tech.com.vn'}
```

1. Tạo dict:

Tạo dict bằng hàm zip()

EXAMPLE

```
a = ['name', 'job', 'web']
b = ['28tech', 'dev', '28tech.com.vn']
c = dict(zip(a, b))
print(c)
```

OUTPUT

```
{'name': '28tech', 'job': 'dev', 'web': '28tech.com.vn'}
```

Tạo dict bằng hàm fronkey

EXAMPLE

```
a = ['a', 'b', 'c']
defaultValue = 0
b = dict.fromkeys(a, defaultValue)
print(b)
```

OUTPUT

```
{'a': 0, 'b': 0, 'c': 0}
```

2. Các tính chất quan trọng của dict:

Key trong dict là duy nhất: Dict không thể chứa 2 key giống nhau, trong trường hợp bạn gán nhiều value cho cùng 1 key thì dict sẽ giữ lại value cuối cùng bạn gán cho key đó.


```
a = {'name' : '28tech', 'job' : 'dev', 'name' : '27tech'}
print(a)
```

OUTPUT

```
{'name': '27tech', 'job': 'dev'}
```

2. Các tính chất quan trọng của dict:

Key phải là object không thể thay đổi (immutable): Bạn có thể lựa chọn tuple, str, int… làm key còn value có thể là bất kì object nào.


```
a = {(1, 2) : 'abc', 'name' : '28tech', 3 : 4}
print(a)
```

OUTPUT

```
{(1, 2): 'abc', 'name': '28tech', 3: 4}
```


```
a = {[1, 2] : 'abc', 'name' : '28tech', 3 : 4}
print(a)
```

OUTPUT

```
TypeError: unhashable type: 'list'
```

3. Truy cập phần tử:

Để truy cập value thông qua key ta dùng cú pháp : **dict[key]**, tuy nhiên nếu bạn sử dụng một giá trị không phải là key trong dict sẽ dẫn đến lỗi.


```
infor = {
    'name' : '28tech',
    'job' : 'dev',
    'salary' : 500,
    'web' : '28tech.com.vn',
    'city' : 'HCM'
}
print(infor['name'])
print(infor['web'])
```

OUTPUT

```
28tech
28tech.com.vn
```

3. Truy cập phần tử:

Để tránh xảy ra lỗi khi truy cập vào một key không nằm trong dict các bạn có thể sử dụng **hàm get(),** hàm này sẽ trả về None nếu key bạn truy cập không nằm trong dict.


```
infor = {
    'name' : '28tech',
    'job' : 'dev',
    'salary' : 500,
    'web' : '28tech.com.vn',
    'city' : 'HCM'
}
print(infor.get('name'))
print(infor.get('address'))
```

OUTPUT

```
28tech
None
```

4.Duyệt dict():

Lấy key, value, item trong list: Bạn có thể sử dụng phương thức keys(), values(), items() trong dict để lấy về bộ key, value, hoặc items() trong dict. 3 phương thức này đều trả về iterable object, bạn có thể convert nó sang list.


```
infor = {
    'name' : '28tech',
    'job' : 'dev',
}
a = list(infor.keys())
b = list(infor.values())
c = list(infor.items())
print(a)
print(b)
print(c)
```

OUTPUT

```
['name', 'job']
['28tech', 'dev']
[('name', '28tech'), ('job', 'dev')]
```

4. Duyệt dict():

Duyệt dict mặc định

EXAMPLE

```
infor = {
    'name' : '28tech',
    'job' : 'dev',
    'salary' : 500,
    'web' : '28tech.com.vn',
    'city' : 'HCM'
}
for x in infor:
    print(x, end = ' ')
```

OUTPUT

name job salary web city

4. Duyệt dict():

Duyệt các cặp key, value trong dict

EXAMPLE

```
infor = {
    'name' : '28tech',
    'job' : 'dev',
    'salary' : 500,
    'web' : '28tech.com.vn',
    'city' : 'HCM'
}
for x in infor:
    print(x, infor[x], sep = '->')
```

OUTPUT

```
name->28tech
job->dev
salary->500
web->28tech.com.vn
city->HCM
```

EXAMPLE

```
infor = {
    'name' : '28tech',
    'job' : 'dev',
    'salary' : 500,
    'web' : '28tech.com.vn',
    'city' : 'HCM'
}
for (key, value) in infor.items():
    print(key + '->' + str(value))
```

OUTPUT

```
name->28tech
job->dev
salary->500
web->28tech.com.vn
city->HCM
```

4. Duyệt dict():

Duyệt các value trong dict

EXAMPLE

```
infor = {
    'name' : '28tech',
    'job' : 'dev',
    'salary' : 500,
    'web' : '28tech.com.vn',
    'city' : 'HCM'
}
for x in infor.values():
    print(x)
```

OUTPUT

```
28tech
dev
500
28tech.com.vn
HCM
```

5. Thêm và sửa dict:

Để thêm cặp key:value vào dict bạn sử dụng cú pháp dict[key] = value. Nếu key đã nằm trong dict thì câu lệnh trên sẽ cập nhật value mới cho key.

Thêm item

```
infor = {
    'name' : '28tech',
    'job' : 'dev',
}
infor['web'] = '28tech.com.vn'
print(infor)
```

OUTPUT

```
{'name': '28tech', 'job': 'dev', 'web': '28tech.com.vn'}
```

Thay đổi giá trị của value thông qua key

```
infor = {
    'name' : '28tech',
    'job' : 'dev',
}
infor['name'] = '27tech'
print(infor)
```

OUTPUT

```
{'name': '27tech', 'job': 'dev'}
```

6. Xóa phần tử trong dict:

Xóa phần tử thông qua key bằng **hàm pop().**

EXAMPLE

```
infor = {
    'name' : '28tech',
    'job' : 'dev',
    'web' : '28tech.com.vn'
}
infor.pop('job')
print(infor)
```

OUTPUT

```
{'name': '28tech', 'web': '28tech.com.vn'}
```

Chú ý: Sử dụng **hàm pop()** với key không tồn tại trong dict sẽ gây lỗi Keyerror.

Nếu bạn không cần giá trị value của item cần xóa bạn có thể sử dụng **hàm del.**

EXAMPLE

```
infor = {
    'name' : '28tech',
    'job' : 'dev',
    'web' : '28tech.com.vn'
}
del infor['job']
print(infor)
```

OUTPUT

```
{'name': '28tech', 'web': '28tech.com.vn'}
```

Chú ý: Sử dụng **hàm del()** với key không tồn tại trong dict sẽ gây lỗi Keyerror.

6. Xóa phần tử trong dict:

Xóa phần tử ngẫu nhiên bằng **hàm popitem().**

EXAMPLE

```
infor = {
    'name' : '28tech',
    'job' : 'dev',
    'web' : '28tech.com.vn'
}
a = infor.popitem()
print(a)
print(infor)
```

OUTPUT

```
('web', '28tech.com.vn')
{'name': '28tech', 'job': 'dev'}
```

Sử dụng **hàm clear()** để xóa mọi phần tử trong dict.

EXAMPLE

```
infor = {
    'name' : '28tech',
    'job' : 'dev',
    'web' : '28tech.com.vn'
}
infor.clear()
print(len(infor))
```

OUTPUT

```
0
```

7. Kiểm tra sự tồn tại của key, value:

Các bạn có thể sử dụng toán tử in để kiểm tra sự tồn tại của một key hoặc value nào đó trong dict.

EXAMPLE

```
infor = {
    'name' : '28tech',
    'job' : 'dev',
    'web' : '28tech.com.vn'
}
print('job' in infor)
print('28tech' in
infor.values())
```

OUTPUT

```
True
True
```

8. Trộn 2 dict:

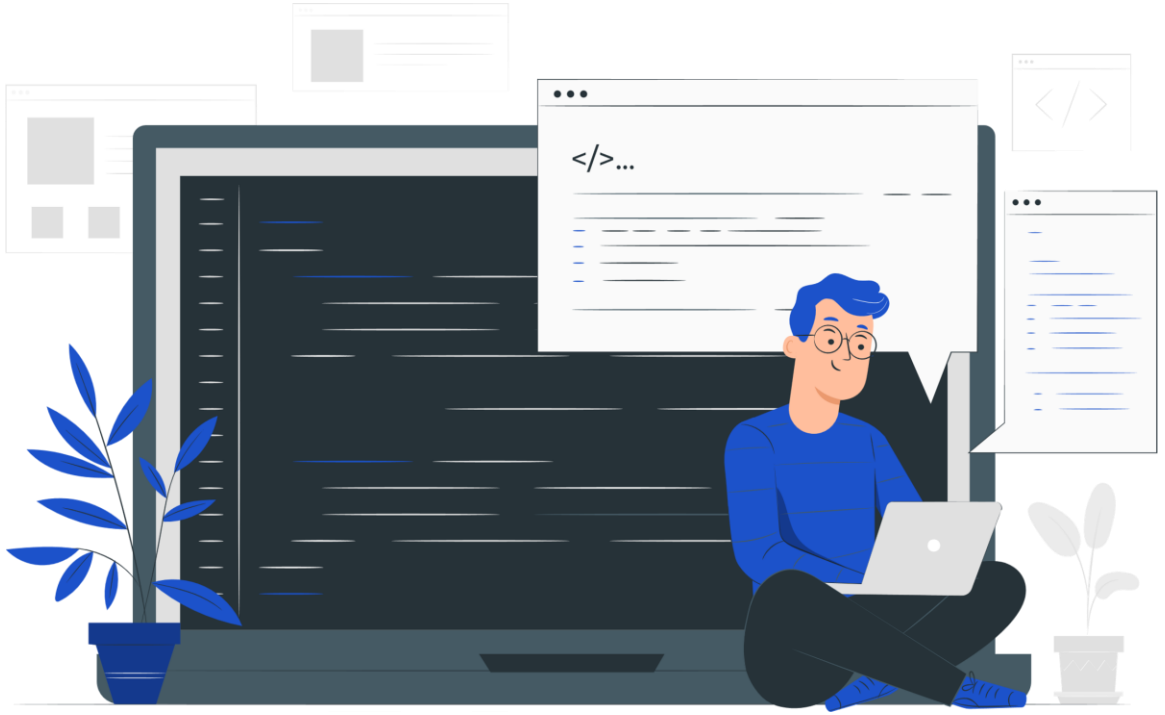
Để trộn 2 dict với nhau bạn có thể sử dụng **hàm update().**

EXAMPLE

```
a = {
    'name' : '28tech',
    'job' : 'dev'
}
b = {'web' : '28tech.com.vn', 'salary' : 500}
a.update(b)
for key, value in a.items():
    print(key, value)
```

OUTPUT

```
name 28tech
job dev
web
28tech.com.vn
salary 500
```



9. Một số ví dụ với dict:

Với dict bạn có thể sử dụng vào những bài toán liên quan tới tần suất, đếm, đánh dấu...

Đếm tần suất xuất hiện của phần tử trong mảng

```
a = [1, 3, 2, 1, 2, 3, 1, 0]
d = dict({})
for x in a :
    if x in d:
        d[x] += 1
    else:
        d[x] = 1
for val, fre in d.items():
    print(val, fre)
```

OUTPUT

```
1 3
3 2
2 2
0 1
```

Tìm từ xuất hiện nhiều nhất trong câu, nếu có nhiều từ có cùng số lần xuất hiện thì in ra từ có thứ tự từ điển nhỏ nhất.

```
s = "28tech 28tech abc abc python c++ java"
d = dict({})
for x in s.split():
    if x in d: d[x] += 1
    else: d[x] = 1
b = list(d.items())
b.sort(key = lambda x : (-x[1], x[0]))
print(b[0][0], b[0][1])
```

OUTPUT

```
28tech 2
```

10. Dict comprehension:

Tương tự như list comp, dict comp là một cách gọn nhẹ giúp tạo ra một dict giúp code của bạn chuẩn Pythonic hơn.

CÚ PHÁP: {key : value for var in iterable}

EXAMPLE

```
a = [1, 2, 3, 4]
d1 = dict({})
for x in a:
    d1[x] = x ** 2

#dict comp
d2 = {x : x ** 2 for x in a}
print(d1)
print(d2)
```

OUTPUT

```
{1: 1, 2: 4, 3: 9, 4: 16}
{1: 1, 2: 4, 3: 9, 4: 16}
```

EXAMPLE

```
s = 'abc'
d = {x : x * 3 for x in s}
print(d)
```

OUTPUT

```
{'a': 'aaa', 'b': 'bbb', 'c': 'ccc'}
```

10. Dict comprehension:

CÚ PHÁP: {key : value for var in iterable}

EXAMPLE

```
s = ["28tech", "PyThon", "JaVA", "C++"]
d = {x.lower() : x.upper() for x in s}
for key, val in d.items():
    print(key, val)
```

OUTPUT

```
28tech 28TECH
python PYTHON
java JAVA
c++ C++
```

EXAMPLE

```
s = {'job' : 'dev', 'name' : '28tech', 'web' : '28tech', 'salary' : 500}
selected = ['name', 'job']
d = {x : s[x] for x in selected}
print(d)
```

OUTPUT

```
{'name': '28tech', 'job': 'dev'}
```