

Dict

ภาควิชาวิศวกรรมคอมพิวเตอร์  
จุฬาลงกรณ์มหาวิทยาลัย

๒๕๖๒

# list กับ dict

- list: เก็บข้อมูลเป็นรายการเรียงจากซ้ายไปขวา
  - ใช้จำนวนเต็มเป็นตัวระบุตำแหน่ง ในการใช้ข้อมูลใน list

```
x = ["MO", "TU", "WE", "TH", "FR", "SA", "SU"]
```

↑  
x[2]

↑  
x[4]

↑  
x[-1]

- dict: เก็บข้อมูลเป็นคู่ ๆ *key*: *value*
  - ใช้ *key* เป็นตัวระบุ *value* ที่ต้องการใน dict  
(*key* เป็นได้ทั้ง int, float, str, ...)

d["WE"]



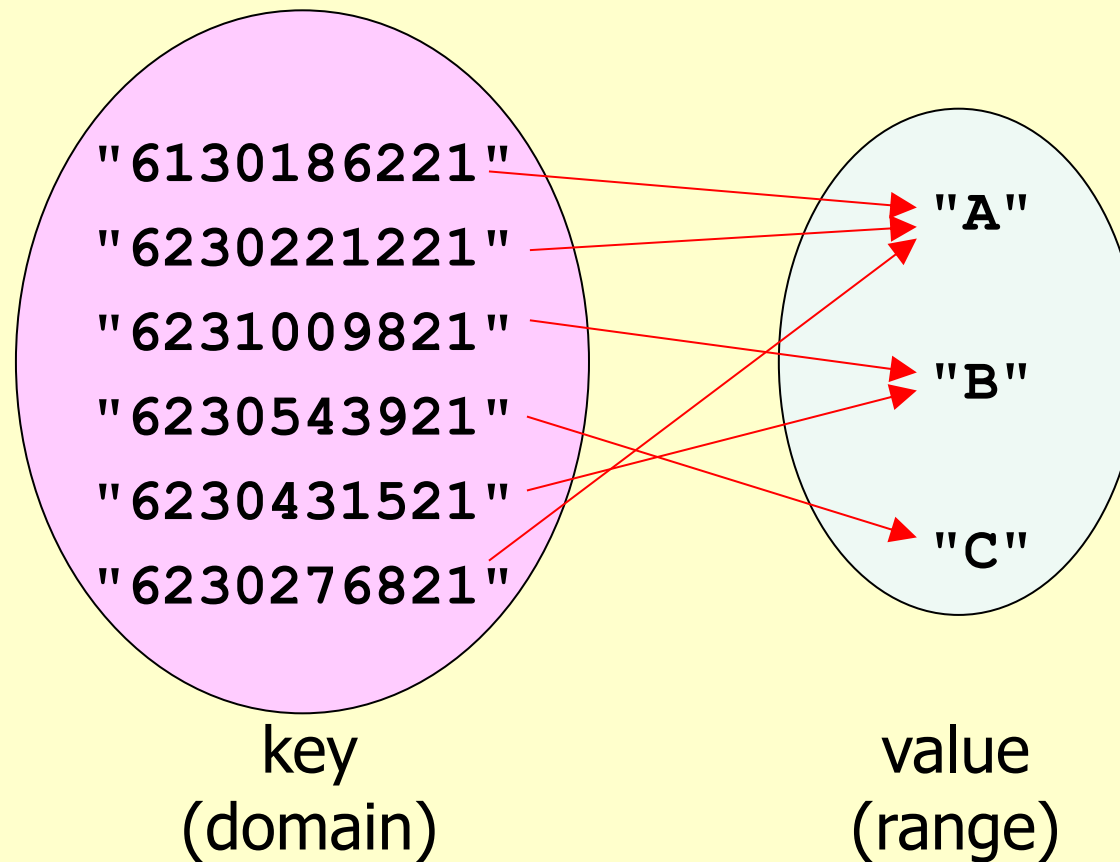
```
d = {"MO": "จ", "TU": "อ", "WE": "พ",  
      "TH": "พฤ", "FR": "ศ", "SA": "ส", "SU": "อา"}
```

↑  
d["TH"]

↑  
d["SA"]

# dict เหมือน mapping function

```
grade = {"6130186221": "A", "6230221221": "A",  
         "6231009821": "B", "6230543921": "C",  
         "6230431521": "B", "6230276821": "A"}
```



keys ห้ามซ้ำกัน  
values ซ้ำกันได้

ให้ *key* กับ dict แล้วได้ *value*

```
v = grade["6231010621"]
```

*key*      *value*

```
grade = {"6130186221": "A", "6230221221": "A",
         "6231009821": "B", "6230543921": "C",
         "6230431521": "B", "6230276821": "A"}
```

```
ID = input()
```

```
while ID != "q":
    print(ID, "-->", grade[ ID ])
    ID = input()
```

ถ้าใช้คีย์ที่ไม่มีอยู่ใน dict  
จะเกิดความผิดพลาด

## Input

6231009821  
6130186221  
6230221221  
9

output

6231009821 --> B  
6130186221 --> A  
6230221221 --> A

มีแต่ให้ key ได้ value  
ไม่มีแบบให้ value ได้ key

# การเพิ่ม/การเปลี่ยน value ใน dict

```
grade["6231010621"] = "A"
```

```
grade = { }  
grade["6231010621"] = "A"           # {"6231010621": "A"}  
  
grade["6231009821"] = "B"           # {"6231010621": "A",  
                                     "6231009821": "B"}  
  
grade["6231009821"] = "C"           # {"6231010621": "A",  
                                     "6231009821": "C"}
```

# การเพิ่ม/การเปลี่ยน value ใน dict

```
grade = { }    # dict ว่าง
n = int(input())
for k in range(n):
    ID, g = input().split()
    grade[ ID ] = g
print(grade)
```

Input

4		
6130186221	A	เพิ่ม
6230221221	A	เพิ่ม
6231009821	F	เพิ่ม
6231009821	B	เปลี่ยน

Output

```
{ '6130186221': 'A', '6230221221': 'A', '6231009821': 'B' }
```

## ตัวอย่าง: นับโหวต

```
BLACKPINK = ["Jisoo", "Jennie", "Rosé", "Lisa"]
votes = {} # สร้าง dict ว่าง
for name in BLACKPINK:
    votes[name] = 0 # เพิ่ม key: value ใหม่ใน dict
# {"Jisoo": 0, "Jennie": 0, "Rosé": 0, "Lisa": 0}

name = input()
while name != 'q':
    if name in BLACKPINK:
        votes[name] += 1 # เพิ่ม value
    name = input()

for name in BLACKPINK:
    print(name, "-->", votes[name]) # หยิบ value มาใช้
```

Lisa  
Lisa  
Lisa  
Lisa  
Jennie  
Aum  
Jisoo  
Lisa  
q

# แบบฝึกหัด: นับจำนวนของตัวอักษรแต่ละตัว

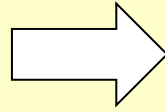
```
alphabets = "abcdefghijklmnopqrstuvwxyz"
```

<i>Input</i>	ABCaBcABZ
<i>Output</i>	a -> 3
	b -> 3
	c -> 3
	z -> 1



*for each\_key in a\_dict*

```
for k in dic:  
    if k == t:  
        v = dic[k]
```



```
v = dic[t]
```

## for each\_key in a\_dict

เมื่อต้องการหยิบ key แต่ละตัวออกมาใช้งาน

```
for k in dic:  
    if k == t:  
        v = dic[k]
```

```
v = dic[k]
```

```
ordinal = {"first": 1, "second": 2, "third": 3,  
           "fourth": 4, "fifth": 5, "sixth": 6,  
           "seventh": 7, "eighth": 8,  
           "ninth": 9, "tenth": 10 }
```

```
for key in ordinal:  
    print(key, "-->", ordinal[key] )
```

ข้อสังเกต: key ที่ได้จาก  
for มีลำดับไม่แน่นอน

```
tenth --> 10  
second --> 2  
fourth --> 4  
third --> 3  
sixth --> 6  
ninth --> 9  
seventh --> 7  
eighth --> 8  
fifth --> 5  
first --> 1
```

ถ้า run โปรแกรมนี้ด้วย Python 3.7 จะได้ key เรียงตามตอน  
สร้าง แต่ไม่รับประกันว่าพฤติกรรมจะเป็นแบบนี้ในรุ่นถัด ๆ ไป  
(Python ใน Grader ก็เป็นแบบที่ได้ลำดับไม่แน่นอน)

## ตัวอย่าง: ฟังก์ชันหาค่าเฉลี่ยของ value ใน dict

```
def average( d ):    # d เป็น dict ที่มี value เป็นจำนวน
    total = 0
    for key in d:    # ลุยหยิบทุก key มาใช้
        total += d[key]
    return total / len(d)
```

len(d) คือจำนวนคู่  
key:value ใน dict d

```
gpa = {"6130186221": 3.15, "6230221221": 2.85,
        "6231009821": 2.90, "6230543921": 3.20,
        "6230431521": 3.35, "6230276821": 3.42}

print( average(gpa) ) # ได้ 3.145
```

# แบบฝึกหัด: สองฟังก์ชัน

```
def reverse( d ): # d เป็น dict ที่ประกันว่า value ไม่ซ้ำกัน
    r = { }
```

รับ { "A": "เอ", "B": "บี", "C": "ซี" }

คืน { "เอ": "A", "บี": "B", "ซี": "C" }

```
    return r
```

```
def keys( d, v ): # คืนลิสต์ของ keys ที่มี value เท่ากับ v
    x = []
```

รับ d = { 2:33, 4:55, 9:33, 7:33, 8:55 }

v = 33

คืน [ 2, 7, 9 ]

```
    return x
```

*if key in dict* หรือ *if key not in dict*

เมื่ออยากรู้ว่ามี *key* หรือไม่มี *key* ใน *dict* ไหม ?

```
grade = {"6130186221": "A", "6230221221": "A",  
         "6231009821": "B", "6230543921": "C",  
         "6230431521": "B", "6230276821": "A"}
```

```
ID = input()  
while ID != "q":
```

```
    if ID in grade:
```

```
        print(ID, "-->", grade[ ID ])
```

```
    else:
```

```
        print("Not found")
```

```
    ID = input()
```

ถ้าไปหยิบ value ของ key ที่ไม่มีใน dict  
จะทำงานผิดพลาด จึงต้องทดสอบก่อน

# *if key in dict* ทำงานเร็วกว่า *if elem in list*

```
import time
def search_all(X):
    b = time.time()
    n = len(X)
    for i in range(n):
        if i in X:      # True
            pass
    for i in range(n):
        if (n+1) in X:  # False
            pass
    print(time.time() - b)
```

```
n = 50000
L = []
for i in range(n):
    L.append(i)
D = {}
for i in range(n):
    D[i] = i

search_all(L)
search_all(D)
```

การค้นใน dict  
เร็วกว่า list

**มาก**

40.90408134460449

0.0156097412109375

หน่วยเป็นวินาที

# แบบฝึกหัด: ชื่อเล่นอะไร ชื่อจริงอะไร

เขียนโปรแกรม รับชื่อจริงแสดงชื่อเล่น  
รับชื่อเล่นแสดงชื่อจริง โดยใช้ dict

Robert  $\longleftrightarrow$  Dick

William  $\longleftrightarrow$  Bill

James  $\longleftrightarrow$  Jim

John  $\longleftrightarrow$  Jack

Margaret  $\longleftrightarrow$  Peggy

Edward  $\longleftrightarrow$  Ed

Sarah  $\longleftrightarrow$  Sally

Andrew  $\longleftrightarrow$  Andy

<http://www.cc.kyoto-su.ac.jp/~trobb/nicklist.html>

## ตัวอย่าง: นับโหวต

```
BLACKPINK = ["Jisoo", "Jennie", "Rosé", "Lisa"]
votes = {"Jisoo":0, "Jennie":0, "Rosé":0, "Lisa":0}

name = input()
while name != 'q':
    if name in BLACKPINK:
        votes[name] += 1
        name = input()
    for name in BLACKPINK:
        print(name, "-->", votes[name])
```

ตรวจจาก key ใน votes ก็ได้  
หยิบจาก key ใน votes ก็ได้  
(ดูหน้าถัดไป)

Lisa  
Lisa  
Lisa  
Lisa  
Jennie  
Aum  
Jisoo  
Lisa  
q



## ตัวอย่าง: นับโหวต (เขียนอีกแบบ)

```
votes = {"Jisoo":0, "Jennie":0, "Rosé":0, "Lisa":0}
```

```
name = input()
```

```
while name != 'q':
```

```
    if name in votes:
```

```
        votes[name] += 1
```

```
    name = input()
```

```
for name in votes:
```

```
    print(name, "-->", votes[name])
```

ตรวจสอบก่อนว่าเป็นชื่อที่ถูกต้อง  
แล้วค่อยเพิ่มคะแนน

Lisa  
Lisa  
Lisa  
Lisa  
Jennie  
Aum  
Jisoo  
Lisa  
q

# ตัวอย่าง: นับโหวต (แบบนับทุกชื่อ)

```
votes = {}
```

```
name = input()
```

```
while name != 'q':
```

```
    if name in votes:
```

```
        votes[name] += 1 # เป็นชื่อที่มีอยู่แล้ว เพิ่ม 1 คะแนน
```

```
    else:
```

```
        votes[name] = 1 # เป็นชื่อใหม่ ให้ 1 คะแนน
```

```
    name = input()
```

```
for name in votes:
```

```
    print(name, "-->", votes[name])
```

เราไม่รู้ว่า จะมีชื่อใครบ้าง  
ก็ต้องอ่านไป สร้างไป นับไป

Lisa  
Lisa  
Lisa  
Lisa  
Jennie  
Aum  
Jisoo  
Lisa  
q

# แบบฝึกหัด: ยอดขายไอศกรีม

	Input	Output
	5	Total ice cream sales = 725.0
ราคา	Magnum 50	
	Cornetto 25	
	PaddlePop 15	
	AsianDelight 20	
	Calippo 15	
	9	ยอดขาย =
	Magnum 5	5x50 +
	Cookie 4	
	MamaTomYum 3	
จำนวน	Cornetto 5	5x25 +
ขาย	AsianDelight 4	4x20 +
	Calippo 4	4x15 +
	Cornetto 6	6x25 +
	MangoSheet 4	
	Calippo 4	4x15 = 725

# ย้ำอีกครั้ง

d เป็น dict, e เป็น key

```
if e in d :  
    ...
```

เร็ว

```
d[e] = z
```

เร็ว

```
z = d[e]
```

เร็ว

x เป็น list, e เป็นข้อมูล

```
if e in x :  
    ...
```

ช้า

```
k = x.index(e)
```

ช้า

k เป็นจำนวนเต็ม

```
x[k] = z
```

เร็ว

```
z = x[k]
```

เร็ว

