

$\hookrightarrow \text{dict} = \{\}$
 $\text{pt}(\text{dict}, \text{goal}(\text{dict}))$

Dicit

{} , False

```
# [1] FALSE  
#> dict = {'fruit': "banana", 'animal': "monkey",  
#>           "two": 2}
```

dict_keys()
items()

dic.items()

dict. values')

clci "apple"

↳ `dict['service'] = {
 ...
 'set', 'item', 'keys'}`

...^g(dic)

`dict(dic)`, `copy`, `fromkeys`, `getdefault`, `update`, `values`

~~dir~~, copy, update),
#clear, setdefault,
pop; then, ~~copy~~

pop, pop;

↳ dict.clear()

dic1.clear()
↳ { }
+ ('fruit'), #banana
+ None

1. dict. set($\{s\}$) #
t('Frodo')

↳ dict. set('Fruit')
{'apple', 'banana', 'cherry', 'orange', 'juice'}

dict. sing, "keys (keys)

↳ Keys = 2^q ,
 $\text{Keys} = \text{dict - from Keys}$
 $\text{Keys} = \text{dict - from Keys}$
 $\text{Keys} = \text{dict - from Keys}$

$$\begin{aligned} J_{12} &= \left[\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix} \right]^2 = \left[\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix} \right] \cdot \left[\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix} \right] = \left[\begin{smallmatrix} 1+3 & 2+4 \\ 3+4 & 2+16 \end{smallmatrix} \right] = \left[\begin{smallmatrix} 4 & 6 \\ 7 & 18 \end{smallmatrix} \right] \\ J_{12} &= \left[\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix} \right] \cdot \left[\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix} \right] = \left[\begin{smallmatrix} 1+3 & 2+4 \\ 3+4 & 2+16 \end{smallmatrix} \right] = \left[\begin{smallmatrix} 4 & 6 \\ 7 & 18 \end{smallmatrix} \right] \end{aligned}$$

$\text{dice}^2 = \text{apple}$, e
 $s^{\text{"ba"}}$

Set - ~~set~~ "get", "set", "remove", "keys",
"fromKeys" "true"

Set "a" contains "set", "copy", "from", "keys", "get", "items",
"clear", "popitem", "setdefault", "update", "values".

dict = {"swig": "banana", "animal": "monkey"}
1: "one", "two")

dict.keys()
dict.values()
dict.items()

dic1["fruit"] # banana
dic1["three"] # key error
dic1.set("three") # none

$\hookrightarrow \text{dic}^1 \cdot \left(\begin{smallmatrix} \text{con} \\ \text{dic}^1 \end{smallmatrix} \right) \stackrel{?}{=} 3$

↳ Did C "travel" 3²³, since: ③
"Home"
"Home"

→ `dic["three"] = 3`
 `{'fruit': "banana", "three": 3}`

 # same key
`dic["three"] = 56` `"three": 56}`

(GPs) $\text{dic}_2 = \text{dic}^1$, $\text{id}(\text{dic}^2))$
 $\text{pt}(\text{id}(\text{dic}^1), \dots)$ same

#include <iostream>

Solution
↳ $\text{dic}^3 = \text{dic}^1. \text{copy}()$
 $\text{id}(\text{dic}^1), \text{id}(\text{dic}^3)$
They are different

// Set Default

dic1 = { }
dic1.setdefault("k", 45)

{ "k": 45 }
dic1.setdefault("m", 46)

{ "k": 45, "m": 46 }

dic1.setdefault("k", 47) // Any value present
with "k" will not add
"k": 45, "m": 46
some ordered.

// Dictionary now

ABDUL

di = { 1: "one", 2: "two" }

di[1] # one

di[5] # key error

di[3] = "three"
{ 1: "one", 2: "two", 3: "three", 4: "four" }

di[5] = "five"

di[5] = "five"

5: "five" }

di = {

for i in di:
pt(i, di[i]) } { 1: "one",
2: "two",
3: "three",
4: "four",
5: "five" }

// Heterogeneous

$d_1 = [1: 3, 5, 2: 5: \text{True}, 5+6j: "abc"]$

$d_2 = \sum_1: [10, 11], 2: (4, 5), 3: 88: 93,$
 $4: \{1: 1, 2: 2\}$

$\underline{d_2[4]} = \{1: 1, 2: 2\}$

$d_3 = \{1, 2\}: "hi"$

$d_4 = \{[1, 2, 3]: "hi"\}$

// List is mutable.

Creation Method

$d_2 = \{1: "one", 2:$

$(3, "three"), 4: "four"\}$

$d_2 = [1: "one", 2: "two", 3: "three", 4: "four"]$

$\text{dict} = \text{dict}[d_2]$

$\text{dict} = \{1: "one", 2: "two", 3: "three", 4: "four"\}$

ZIP

$L_1 = [1, 2, 3, 4], 1: "one", 2: "two", 3: "three", 4: "four", 5: "five"]$

$L_2 = ["one", "two", "three", "four", "five"]$

$L_2 = \text{zip}(L_1, L_2)$

$d_2 = \text{dict}(\text{zip}(L_1, L_2))$

$\{1: "one", 2: "two", 3: "three", 4: "four"\}$

// enumerate (L1) - One
L1 = ["one", "two", "three", "four"]

L1 = ["one", "two", "three", "four"]
d1 = dict(enumerate(L1, start=1))

enumerate (L1, start=1)

d2 = dict(enumerate(L1, start=1))

{1: "one", 2: "two", 3: "three", 4: "four"}

Comprehensions:

L1 = [(1, "one"), (2, "two"), (3, "three"), (4, "four")]
list of tuples
x,y in L1

d3 = {x:y for x,y in L1}
{1: "one", 2: "two", 3: "three", 4: "four"}

L1 = [1, 2, 3, 4]

L2 = ["one", "two", "three", "four"]

d4 = dict(zip(L1, L2))

for x,y in zip(L1, L2)

d5 = {x:y for x,y in zip(L1, L2)}

{1: "one", 2: "two", 3: "three", 4: "four"}

L1 = ["one", "two", "three", "four"]

d6 = dict(enumerate(L1, start=1))

for x,y in enumerate(L1, start=1)

d7 = {x:y for x,y in enumerate(L1, start=1)}

{1: "one", 2: "two", 3: "three", 4: "four"}

dic4

Loops

$d1 = \{4: "four", 5: "five", 6: "six", 7: "sevens"\}$

for i in d1: || 4 four
 pt(i, d1[7]) || 7 sevens

for i in d1.keys(): || 4, 5, 6, 7
 pt(i, end="")

for i in d1.values(): || four
 pt(i, end="") || seven

for i in d1.items(): || (7, "seven")
 pt(i, end="")

(4, "four") (5, "five")

for x, y in d1.items(): || 4 four
 pt(x, y) || 5 five
 6 six
 7 seven

↳ d1.set(4) # four

d1[4] # four

d1.set(16) # None

d1[16] # out scope

d1[16] # keyError

d1.set(5) # 5 missing

d1.set(16, "missing") # missing

1/ setdefault

d1. setdefault(s)
key already there

d1. setdefault(t)
None
 $s: "five", 6: "six",$
 $t: "four", 5: "five", 16: "none"$

$\hookrightarrow \{4: "four", 5: "five", 6: "six", 7: "seven", 16: "none", 17: "undetermined"\}$

$\hookrightarrow d1. setdefault(17, "undetermined")$
 $\{4: "four", 5: "five", 6: "six", 7: "seven", 16: "none", 17: "undetermined"\}$

dict methods

dict (dictionary)

a. update(sequence, $default$)

b. fromkeys(sequence, $default$)

c. copy()

d. pop(key, default)

e. popitem()

f. clear()

updated

$d1 = \{1: "one", 2: "two"\}$

$d2 = \{5: "five"\}$

$d1. update(d2)$

$p + (d1) \Rightarrow \{1: "one", 2: "two", 3: "three", 4: "four", 5: "five"\}$

fromkeys

(210)

- ↳ $L_1 = [1, 2, 3, 4]$ dict from keys (L_1)
 $d_2 = \{1: \text{none}, 2: \text{none}, 3: \text{none}, 4: \text{none}\}$
 $\{1: \text{none}, 2: \text{none}, 3: \text{none}, 4: \text{none}\}$
↳ $d_3 = \{1: \text{"unknown"}, 2: \text{"unknown"}, 3: \text{"unknown"}, 4: \text{"unknown"}\}$

copy
 $d_5 = d_1.\text{copy}()$
 $d_5 = \{1: \text{"one"}, 2: \text{"two"}, 3: \text{"three"}, 4: \text{"four"}\}$
 $d_1 = \{1: \text{"one"}, 2: \text{"two"}, 3: \text{"three"}, 4: \text{"four"}\}$
 $d_5 = \{1: \text{"one"}, 2: \text{"two"}, 3: \text{"three"}, 4: \text{"four"}\}$

Shallow copy

pop
 $d_1 = \{1: \text{"one"}, 2: \text{"two"}, 3: \text{"three"}, 4: \text{"four"}\}$
 $d_1.\text{pop}(2)$
two
 $d_1 = \{1: \text{"one"}, 3: \text{"three"}, 4: \text{"four"}\}$
 $d_1.\text{pop}(5)$
Key Error
missing
 $d_1.\text{pop}(5, \text{"missing"})$
missing

#PopItem
 d1 = {1: "one", 2: "two", 3: "three", 4: "four"}
 d1.popitem()
 (4, "four")
 ↳ d1.create() |
 #NameError
 #{} { }

Programs
 users = [{"john": "bob", "alex": "john"}, {"bob": "richard", "alex": "richard"}]

d1 = {}
 for i in users:
 if i in d1:
 d1[i] += 1
 else:
 d1[i] = 1

Lossy
 users:
 [{"john": "bob", "alex": "john"}, {"bob": "richard", "alex": "richard"}]

// original = {"a": 1, "b": 2, "c": 1, "d": 2, "e": 3}
 inverted = {}
 for x, y in original.items():
 if y not in inverted:
 inverted[y] = {}
 else:
 inverted[y].update(x)

inverted[2] = {"a": 1, "b": 1, "c": 1, "d": 1, "e": 1}

{1: {"a": 1, "c": 1}, 2: {"a": 1, "b": 1, "c": 1, "d": 1, "e": 1}, 3: {}}

$Ser_1 = "abccbab"$
 $Ser_2 = "cdeecbc"$

playful
title

Die 6

```

    flag = True
if len(str1) != len(str2):
    flag = False

```

```

else:
    map1, map2 = {}, {}

    for c1, c2 in zip(str1, str2):
        if c1 in map1:
            if map1[c1] != c2:
                flag = False
                break
        map1[c1] = c2

```

$$\begin{array}{c} \text{a} \quad \text{b} \\ \text{c} \quad \text{d} \\ \text{e} \quad \text{f} \end{array}$$

```

else: map1[c1] = c2
if c2 in map2:
    if map2[c2] != c1:
        flag = False
        break
else:
    map2[c2] = c1

```

Nested

Nested [{"name": "James", "age": 25, "city": "NY"}, {"name": "Eduard", "age": 30, "city": "DEL"}]

```
data = [{"name": "James", "age": 30, "city": "DELHI"},  
        {"name": "Killer", "age": 24, "city": "PARIS"}]
```

["Killer", 30, "PAC"],
[" ", 24, " "]

"Kild", 24, "PAQ", 177

[1855-1861, 24, PA, 1861, 1]]

[15m] 13, 14, 15, DEL]]

115m
Aug 27, 1962

PERAIS, 21, 1981. 8

[K A] - 'age',

frame, ast,

= [same]

$$= \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} x^n$$

```

length = len(students)
result = []
for i in range(length):
    newdict = {}
    for row in data:
        if row[i] not in newdict:
            newdict[row[i]] = [row]
        else:
            newdict[row[i]].append(row)
    result.append(newdict)
result = append(result)

# name      : [C'james', 25, "NY"]
# James     : [C'Karen', 30, "DEL"]
# Karen
# ' '
# age      : [C'james', 25, "NY"]
# 25       : [C'Ras', 27, "DEL"]
# ' '
# 27       : [C'james', 25, "NY"], [C'Ras', 27, "DEL"]
# city      : [C'james', 25, "NY"], [C'Karen', 30, "DEL"], [C'Ras", 27, "DEL"]
# NY       : [C'Karen', 30, "PAR"]
# DEL
# PAR

```

```
import uuid
items = [
    ["laptop", 1200],
    ["mouse", 20],
    ["key", 30],
    ["tablet", 200]
]

item_data = {}

for item in items:
    id = uuid.uuid5(uuid.NAMESPACE_EPOCH, item)
    key = id.hex[-6]
    item_data[key] = item
    print(item_data)

# { 'bd0220': '56831a', 'laptop': [ 'laptop', 1200], 'mouse': [ 'mouse', 20], 'key': [ 'key', 30], 'tablet': [ 'tablet', 200] }.
```