(1) Dictionary

In [ ]:
```
'''
Q1: Write a program that:

Creates a dictionary, dic_1, mapping the following keys to values:

key-value pairs:
1) apple and fruit
2) strawberry and fruit
3) brocolli and vegetable
4) carrot and vegetable
5) tomato and fruit

>>> dic_1
{'apple': 'fruit',
 'strwberry': 'fruit',
 'brocolli': 'vegetable',
 'carrot': 'vegetable',
 'tomato': 'fruit'}
'''
```

In [1]:
```
dic_1 = {'apple': 'fruit', 'strwberry': 'fruit', 'brocolli': 'vegetable', 'carrot': 've
```

In [2]:
```
dic_1
```

Out[2]:
```
{'apple': 'fruit',
 'strwberry': 'fruit',
 'brocolli': 'vegetable',
 'carrot': 'vegetable',
 'tomato': 'fruit'}
```

In [ ]:
```
'''
Q2: Write a program that:

Changes dic_1 by the following:

from tomato and fruit
to tomato and vegetable

>>> dic_1
{'apple': 'fruit',
 'strwberry': 'fruit',
 'brocolli': 'vegetable',
 'carrot': 'vegetable',
 'tomato': 'vegetable'}
'''
```

In [3]:
```
dic_1['tomato'] = 'vegetable'
```

In [4]:
```
dic_1
```

Out[4]:
```
{'apple': 'fruit',
 'strwberry': 'fruit',
 'brocolli': 'vegetable',
 'carrot': 'vegetable',
 'tomato': 'vegetable'}
```

In [ ]:
```
'''
Q3: Write a program that:

(1) Counts a number of occurences of each value in the dic_1; and
(2) Prints following messages:

>>>
fruit appears 2 times.
vegetable appears 3 times.
'''
```

In [5]:
```
counters = {}

for i in dic_1.values():
    if i in counters:
        counters[i] += 1
    else:
        counters[i] = 1

for i in counters:
    print('{} appears {} times.'.format(i, counters[i]))
```

```
fruit appears 2 times.
vegetable appears 3 times.
```

In [ ]:
```
'''
Q4: Write a function lookup_dic() that:

Implements an employee information management application.

The function should take, as inputs, a dictionary representing a employee table,
mapping strings (containing an employee number) as a key
to tuples (containing first name and last name) as a value

It repeats 1) requesting an user to input employee number, and
           2) printing out one's first and last name.

If an user inputs wrong number, it prints out "Wrong number!"
If an user inputs nothing, stop the iteration and print out "Goodbye."

>>> employee = {'901234': ('Anna', 'Karenina'),
                '321908': ('Yu', 'Tsun'),
                '123456': ('Hans', 'Castorp')}

>>> lookup_dic(employee)
Enter the employee number: 321908
Name: Yu Tsun
Enter the employee number: 123456
Name: Hans Castorp
Enter the employee number: 001234
Name: Wrong number!
```

```
Enter the employee number:
Goodbye.
'''
```

In [6]:
```
employee = {'901234': ('Anna', 'Karenina'),
            '321908': ('Yu', 'Tsun'),
            '123456': ('Hans', 'Castorp')}
```

In [7]:
```
def lookup_dic(employee):
    while True:
        emp_no = input('Enter the employee number: ')
        a = employee.values()
        b = list(a)
        if emp_no in employee:
            print('Name:', * employee[emp_no])
        elif emp_no == '':
            print('Goodbye.')
            break
        else:
            print('Name: Wrong number!')
```

In [5]:
```
lookup_dic(employee)
```

```
Name: Yu Tsun

Name: Hans Castorp

Name: Wrong number!

Goodbye.
```

In [ ]:
```
'''
Q5: Write a function create_dic() that:

Implements an employee information management application.

The function takes nothing,
mapping strings (containing an employee number) as a key
to tuples (containing first name and last name) as a value

It 1) repeats requesting an user to input employee number, firt name, and last name
   2) returns the employee table (i.e. dictionary) and stops the iteration, as an user

>>> create_dic()
Enter the employee number: 901234
Enter the first name: Anna
Enter the last name: Karenina
Enter the employee number: 321908
Enter the first name: Yu
Enter the last name: Tsun
Enter the employee number: 123456
Enter the first name: Hans
Enter the last name: Castorp
Enter the employee number:
{'901234': ('Anna', 'Karenina'), '321908': ('Yu', 'Tsun'), '123456': ('Hans', 'Castorp'
'''
```

In [8]:
```
def create_dic():
    dict1 = {}
    dict2 = {}
    dict3 = {}
    while True:
        emp_no = input('Enter the employee number:')
        if emp_no == '':
            break
        fname = input('Enter the first name:')
        lname = input('Enter the last name:')
        emps = {emp_no:(fname, lname)}
        dict1.update(emps)
        dict2.update(dict1)
        dict3.update(dict2)
    print(dict3)
```

In [9]:
```
create_dic()
```

```
Enter the employee number:901234
Enter the first name:Anna
Enter the last name:Karenina
Enter the employee number:321908
Enter the first name:Yu
Enter the last name:Tsun
Enter the employee number:123456
Enter the first name:Hans
Enter the last name:Castorp
Enter the employee number:
{'901234': ('Anna', 'Karenina'), '321908': ('Yu', 'Tsun'), '123456': ('Hans', 'Castor
p')}
```

(2) Random Module

In [ ]:
```
'''
Q6: Write a function rand_n() that:

(1) Takes a bound as an input; and
(2) Returns a random positive integer number lower than the bound.

>>> rand_n(10)
3
>>> rand_n(100)
13
'''
```

In [10]:
```
def rand_n(bound):
    import random
    t = random.randrange(bound)
    return t
```

In [11]:
```
rand_n(10)
```

Out[11]: 2

In [12]:
```
rand_n(100)
```

Out[12]: 46

In [ ]:
```
'''
Q7: Write a function random_choice() that:

(1) Takes a list as an input; and
(2) Randomly selects an item in the list and returns it.

>>> random_choice(['apple', 'pear', 'strawberry', 'rasberry'])
'apple'
>>> random_choice(['apple', 'pear', 'strawberry', 'rasberry'])
'rasberry'
>>> random_choice(['apple', 'pear', 'strawberry', 'rasberry'])
'pear'
'''
```

In [13]:
```
def random_choice(lst):
    import random
    return random.choice(lst)
```

In [14]:
```
random_choice(['apple', 'pear', 'strawberry', 'rasberry'])
```

Out[14]: 'strawberry'

In [15]:
```
random_choice(['apple', 'pear', 'strawberry', 'rasberry'])
```

Out[15]: 'rasberry'

In [16]:
```
random_choice(['apple', 'pear', 'strawberry', 'rasberry'])
```

Out[16]: 'apple'

In [ ]:
```
'''
Q8: Write a function dice() that returns an integer from 1 to 6 with equal probability.

>>> dice()
3
>>> dice()
1
>>> dice()
5
'''
```

In [17]:
```
def dice():
    import random
```

```
    a = random.randrange(1, 7)
    return a
```

In [18]:
```
dice()
```

Out[18]: 5

In [19]:
```
dice()
```

Out[19]: 5

In [20]:
```
dice()
```

Out[20]: 4

In [ ]:
```
'''
Q9: Write function dice_game() that:

(1) Has two players (i.e. A and B) who roll a dice ranging from 1 to 6; and
(2) Prints out following messages that show:
    1) Each player's dice value;
    2) The winner; and
    3) If the dice values are same, prints out "Draw!"."

>>> dice_game()
A:3, B:4
B is the winner!

>>> dice_game()
A:5, B:2
A is the winner!

>>> dice_game()
A:4, B:4
Draw!
'''
```

In [21]:
```
def dice_game():
    import random
    a = random.randrange(1, 7)
    b = random.randrange(1, 7)
    print('A:{}, B:{}'.format(a, b))

    if a > b:
        print('A is the winner!')
    elif a < b:
        print('B is the winner!')
    else:
        print('Draw!')
```

In [22]:
```
dice_game()
```

```
A:2, B:2
Draw!
```

In [23]:
```
dice_game()
```

```
A:5, B:4
A is the winner!
```

In [24]:
```
dice_game()
```

```
A:4, B:5
B is the winner!
```

In [ ]:
```
'''
Q10: Write function dice_game_summary() that:

(1) Takes a number of games (i.e. rounds) as an input;
(2) Has two players (i.e. A and B) who roll a dice ranging from 1 to 6
(3) It repeatedly prints out the result of each round that shows:
    1) Each player's dice value;
    2) The winner; and
    3) If the dice values are same, prints out "Draw!"; and
(4) After running all rounds, it prints out the summary by following:
    1) A number of winning games of A;
    2) A number of winning games of B;
    3) The final champion; and
    4) If the number of winning games are same, prints out "Unbelievable!".

>>> dice_game_summary(3)
A:3, B:4
B is the winner!
A:5, B:2
A is the winner!
A:4, B:4
Draw!
Winning games of A: 1
Winning games of B: 1
Unbelievable!

>>> dice_game_summary(4)
A:2, B:6
B is the winner!
A:3, B:4
B is the winner!
A:5, B:2
A is the winner!
A:4, B:4
Draw!
Winning games of A: 2
Winning games of B: 1
Final Champion is A!
'''
```

In [25]:
```
def dice_game_summary(n):

        def dice_game():
            import random
```

```
            a = random.randrange(1, 7)
            b = random.randrange(1, 7)
            print('A:{}, B:{}'.format(a, b))

            if a > b:
                return 'A is the winner!'
            elif a < b:
                return 'B is the winner!'
            else:
                return'Draw!'

        lst1 = 0
        lst2 = 0
        lst3 = 0
        for i in range(n):
            import random
            a_rand = random.randrange(1, 7)
            b_rand = random.randrange(1, 7)
            if a_rand > b_rand:
                lst1 += 1
            elif a_rand < b_rand:
                lst2 += 1
            else:
                lst3 += 1
            print(dice_game())

        print('Winning games of A: {}'.format(lst1))
        print('Winning games of B: {}'.format(lst2))
        if lst1 > lst2:
            print('Final champion is A!')
        elif lst1 < lst2:
            print('Final champion is B!')
        else:
            print('Unbelievable!')
```

In [26]:

```
dice_game_summary(10)
```

```
A:5, B:3
A is the winner!
A:6, B:3
A is the winner!
A:6, B:6
Draw!
A:1, B:4
B is the winner!
A:5, B:2
A is the winner!
A:6, B:5
A is the winner!
A:3, B:1
A is the winner!
A:4, B:2
A is the winner!
A:1, B:2
B is the winner!
A:6, B:5
A is the winner!
Winning games of A: 2
Winning games of B: 6
Final champion is B!
```