

Ch7: Lists, Tuples, Maps, Sets

305172 Computer Programming
Laboratory
Jiraporn Pooksook
Naresuan University

What is a List?

- List is a sequence of data which can be added or updated its information.

```
fruits_str = 'mango, orange, lychee, apple, banana'
```

```
print(fruits_str)
```

```
fruits_list = ['mango', 'orange', 'lychee', 'apple', 'banana']
```

```
print(fruits_list)
```

Try Lists

```
fruits_list = ['mango', 'orange', 'lychee', 'apple', 'banana']  
  
print(fruits_list)  
  
print(fruits_list[1])  
  
fruits_list[3]='grape'  
  
print(fruits_list)  
  
print(fruits_list[2:3])  
print(fruits_list[2:4])
```

Data Types

- Any primitive types: one type or mix types
- A List of lists.

```
string_list = ['mango', 'orange', 'lychee', 'apple', 'banana']
```

```
num_list = [1.2, 4 , 7.6 , 8.9 ]
```

```
num_and_string_list = ['mon', 'tue', 'wed', 4, 5, 'sat', 'sun']
```

```
list_of_list = [[1,2,3], [8,9] , [2.3, 5.5,8.9,7.6] ]
```

```
fruits_num_list = [ string_list, num_list]
```

Data Types

```
string_list = ['mango', 'orange', 'lychee','apple','banana']

num_list = [1.2, 4 , 7.6 , 8.9 ]

num_and_string_list = ['mon', 'tue', 'wed', 4, 5, 'sat','sun']

list_of_list = [[1,2,3], [8,9] , [2.3, 5.5,8.9,7.6] ]

fruits_num_list = [ string_list, num_list]

print(string_list)
print('str[1]= ',string_list[1])

print(num_list)
print('num[1]= ', num_list[1])

print(num_and_string_list)
print('num&string[1]= ', num_and_string_list[1])

print(list_of_list)
print('list_of_list[1]= ',list_of_list[1])

print(fruits_num_list)
print('fruits_num_list[1]= ',fruits_num_list[1])
```

Operations on List

```
fruits_list = ['mango', 'orange', 'lychee', 'apple', 'banana']  
print(fruits_list)
```

```
fruits_list.append('melon')  
print(fruits_list)
```

```
del fruits_list[3]  
print (fruits_list)
```

```
num_list = [2.4, 5.6, 4.3, 7.8]
```

```
print(fruits_list+num_list)
```

```
new_list = fruits_list + num_list  
print(new_list)
```

```
print(num_list*2)
```

What is a Tuple?

- A tuple is a collection of data which cannot be changed once you have created.

```
fruits_tuple = ('mango', 'orange', 'lychee', 'apple', 'banana')  
print(fruits_tuple)  
print(fruits_tuple[2])
```

```
fruits_tuple[3]=grape  
print(fruits_tuple)
```

Error !!!

What is a Map?

- A map or a dictionary is a collection of data that each item has its own key and value.
- We cannot merge two maps as we do with lists.

Key	Value
Peter	Mango
Bob	Orange
Alice	lychee
Frank	Apple
Kathy	Banana
Kate	Apple

Try Maps?

```
favorite_fruits = {'Peter': 'mango',  
                  'Bob': 'orange',  
                  'Alice': 'lychee',  
                  'Frank': 'apple',  
                  'Kathy': 'banana',  
                  'Kate': 'apple'}
```

```
print(favorite_fruits)
```

```
print(favorite_fruits['Frank'])
```

```
print(favorite_fruits['Ken'])
```

← Error !!!

```
favorite_fruits['Alice'] = 'mango'
```

```
print(favorite_fruits)
```

```
del favorite_fruits['Bob']
```

```
print(favorite_fruits)
```

```
del favorite_fruits
```

Built-in Lists Functions & Methods

```
fruits_list = ['mango', 'orange', 'lychee', 'apple', 'banana']
```

```
num_list = [1, 4, 5, 8, 1, 3]
```

```
print(len(fruits_list))
```

```
print(len(num_list))
```

```
print(max(num_list))
```

```
print(min(num_list))
```

```
print(fruits_list.count('mango'))
```

```
print(num_list.count(1))
```

```
print(fruits_list.index('lychee'))
```

```
print(fruits_list.index('banana'))
```

```
fruits_list.pop()
```

```
print(fruits_list)
```

```
fruits_list.reverse()
```

```
print(fruits_list)
```

```
num_list.sort()
```

```
print(num_list)
```

Lists with Loop

```
fruits = ['mango', 'orange', 'lychee', 'apple', 'banana']
```

```
for i in fruits:  
    print(i)
```

```
print('-----Try double loop \n')
```

```
for i in fruits:  
    print(i)  
    for j in fruits:  
        print(j)  
    print('-----')
```

Exercise

- ใช้ลิสรับค่าตัวเลข n จำนวนใดๆ เก็บในลิส แล้วทำการหาค่าเฉลี่ยของตัวเลขทั้งหมด
- ทำการรับค่าตัวเลขมาหนึ่งชุด n จำนวน แล้วรับค่า ตัวเลขที่ต้องการค้นหา แล้วทำการแสดงผลว่า ตัวเลขที่ต้องการค้นหานั้น อยู่ที่ตำแหน่งเท่าไหร่ในลิส (หรือไม่มี)
- รับค่าตัวเลข n จำนวนใดๆ เก็บในลิส แล้วทำการแสดงค่า เฉพาะเลขที่เป็นเลขจำนวนเฉพาะเท่านั้น
- เขียนโปรแกรมที่ทำการเรียงตัวเลข n จำนวน จากน้อยไปหามาก ด้วยวิธีการ **bubble sort**

Exercise

- เขียนโปรแกรมที่ทำการบันทึกสินค้าที่ซื้อ ซึ่งมี ชื่อสินค้า และ จำนวน เก็บข้อมูลไว้ในลิสต์ เสร็จแล้วให้โปรแกรม ระบุยอดการใช้จ่ายทั้งหมด รวมถึงเรียงลำดับรายการสินค้าที่ซื้อบ่อยครั้งที่สุด จากมากไปน้อย และแสดงลำดับสินค้าที่ใช้เงินซื้อมากที่สุด จากมากไปน้อย เช่นกัน
 - สำหรับราคาของสินค้า ให้เก็บข้อมูลผูกไว้กับชื่อสินค้า (ตรงนี้ต้องใช้ **Map** ชื่อสินค้ากับราคาผูกกันไว้)
- ตัวอย่างเช่น

บันทึกการใช้จ่าย

สบู่ 2 ก้อน

กาแฟ 1 แก้ว

แชมพู 2 ขวด

กาแฟ 2 แก้ว

จากบันทึกพบว่า คุณใช้เงินไปทั้งหมด **380** บาท

รายการสินค้าที่ซื้อบ่อยที่สุด คือ กาแฟ สบู่ แชมพู

รายการสินค้าที่ใช้เงินมากที่สุด คือ แชมพู กาแฟ สบู่

สบู่	35
กาแฟ	50
แชมพู	80

What is a Set?

- A set is an unordered collection of data.

```
fruits = {'mango', 'orange', 'lychee', 'banana', 'banana'}
```

```
print(fruits)
```

```
numbers1 = {1,2,3,4,5,2,3}
```

```
print(numbers1)
```

```
numbers2 = set('1234423')
```

```
print(numbers2)
```

Methods for Set

```
fruits = {'mango', 'orange', 'lychee', 'banana', 'banana'}
print(fruits)

numbers = {1,2,3,4,5,2,3}
print(numbers)

print(len(fruits))

if 1 in numbers:
    print("number 1 is in the set")
elif 1 not in numbers:
    print("number 1 is Not in the set")

emp_set=set()

if fruits.issuperset(emp_set):
    print("fruits is a superset of an empty set.")

if emp_set.issubset(numbers):
    print("an empty set is a subset of numbers.")

print(fruits.union(numbers))
print(fruits.intersection(numbers))
print(fruits.difference(numbers))
print(fruits.symmetric_difference(numbers))

cop_fruits=fruits.copy()
print(cop_fruits)
```

Lists Comprehensions

```
num_list = []

for i in range(10):
    num_list.append(i)

print(num_list)

num_list2 = list(range(10))
print(num_list2)

num_list3 = [i for i in range(10)]
print(num_list3)
```

Lists Comprehensions

```
cart = []
```

```
for i in range(4):  
    for j in [1,3]:  
        if i != j:  
            cart.append( (i,j))
```

```
print(cart)
```

```
cart2=[(i,j) for i in range(4) for j in [1,3] if i!=j]  
print(cart2)
```

Nested List Comprehensions

```
A = [  
    [1,2,3],  
    [4,5,6],  
    [7,8,9],  
    ]
```

```
transpose_A = []  
for i in range(3):  
    transpose_A.append([row[i] for row in A])
```

```
print(transpose_A)
```

```
transpose_A2 = [ [row[i] for row in A] for i in range(3) ]  
print(transpose_A2)
```

Looping Techniques

```
fav_fruits={ 'Peter':'mango',  
             'Bob': 'orange',  
             'Alice': 'lychee',  
             'Frank': 'apple',  
             'Kathy': 'banana',  
             'Kate': 'apple' }
```

```
for i,j in fav_fruits.items():  
    print(i,j)
```

```
fruits_list = ['mango', 'orange', 'lychee', 'banana']  
for i,j in enumerate(fruits_list):  
    print(i,j)
```