

Unit 15.

Dictionary Method 2

Learning objectives

- ✓ ສາມາດແປງ list ແລະ tuple ເປັນ dictionary ໂດຍນໍາໃຊ້ຝັງຊັນ dic.
- ✓ ສາມາດໃຊ້ dict.defaultdict method ເພື່ອສ້າງ key ໃຫ້ຄ່າເລີ່ມຕົ້ນ (default value) ກໍລະນີບໍ່ມີ key.
- ✓ ສາມາດພິມ item ຂອງ dictionary ໂດຍນໍາໃຊ້ statement ການວິນຊີ້າ.
- ✓ ສາມາດສ້າງ dictionary ພາຍໃນ dictionary ພ້ອມທັງ assign ແລະ copy ບັນດາ items ພາຍໃນ dictionary ດັ່ງກ່າວໃຫ້ກັບຕົວປ່ຽນອື່ນໆ.

Learning overview

- ✓ ຮຽນຮູ້ການແປງ lists ແລະ tuples ເປັນ dictionaries.
- ✓ ຮຽນຮູ້ການພິມ dictionaries ໂດຍນຳໃຊ້ statements ວິນຊື້.
- ✓ ຫຼັງຈາກສ້າງ dictionary (double dictionary) ພາຍໃນ dictionary, ທຽນຮູ້ assign ຫຼື copy ບັນດາ items ໃນ dictionary ໃຫ້ກັບຄົວປຸງ ອື່ນ.

Concepts You Will Need to Know From Previous Units

- ✓ ວິທີຈັດການກັບ values ໂດຍການເຂົ້າເຖິງ dictionary key ແລະ ການເຮັດວຽກຂັ້ນພື້ນຖານຂອງ dictionary.
- ✓ ການເພີ່ມ ຫຼື ການບັນທຶກ values ໂດຍໃຊ້ setdefault ແລະ get.
- ✓ ເຂົ້າໃຈການອ້າງອີງຂອງ list ແລະ copy ບັນດາ object.

Keywords

`defaultdict`

`Print Dictionary`

`Double Dictionary`

`Copy Dictionary`

Mission

1. Real world problem

1.1. ການພັດທະນາເຕັກໂນໂລຊີຂ່າວສານ, ການສື່ສານ ແລະ telephone dictionary



- ▶ ເນື່ອງຈາກການພັດທະນາເຕັກໂນໂລຊີຂ່າວສານ, ການສື່ສານ ແລະ mobile phones ກໍາລັງຖືກນຳໄປໃຊ້ວຽກຕ່າງໆ.
- ▶ ຍ້ອນການແຜ່ຫຼາຍຂອງ COVID-19 ແລະ ການເພີ່ມຂຶ້ນຂອງຜູ້ສັກຢາວັກຊີນ, ສະນັ້ນຈຶ່ງເຮັດໃຫ້ການໃຊ້ໂທລະສັບຄ່ອບງໍຂະໜາຍຕົວ, ໂດຍມີການແນະນຳ COVID-19 passports ດັ່ງສະແດງໃນຮູບ.
- ▶ ເຖິງຢ່າງໃດກໍຕາມ, ພັງຊັນພື້ນຖານຂອງ mobile phone ຍັງແມ່ນພັງຊັນສຽງ ແລະ ໃນໂທລະສັບຍັງບັນຈຸຊື່ ແລະ ເປີໂທຂອງຄົນຮູ້ຈັກ.
- ▶ ຕອນນີ້ກໍາລັງຈະສ້າງ telephone dictionary ແບບ່າຍ .

1. Real world problem

1.2. What is phone book?



- ▶ telephone dictionary เป็นบัญชีประจำบ้านโดย เบอร์ และ ชื่อของผู้ใช้บริการเป็นระบบماຊิก ห្មី បំលិតុដ និង ទីយៈ ជំនួយ.
- ▶ ខាងសອរប្រចាំឆ្នាំ, ឱ្យអ្នកស្វែងនៅក្នុងការប្រើប្រាស់ ជំនួយការងារសំសាលាត្រូវបានដាក់ឡើង ដោយ មិនត្រូវបានបញ្ជាក់ថា គ្មានមិនមែន ពី ការប្រើប្រាស់ (telephone dictionary) ទេ ប៉ុណ្ណោះ គឺជាប្រព័ន្ធដែលបានបង្កើតឡើង។
- ▶ បច្ចុប្បន្ន, បំនុះការប្រើប្រាស់ នឹងត្រូវបានដាក់ឡើង ដោយការងារសំសាលាត្រូវបានបង្កើតឡើង។

1. Real world problem

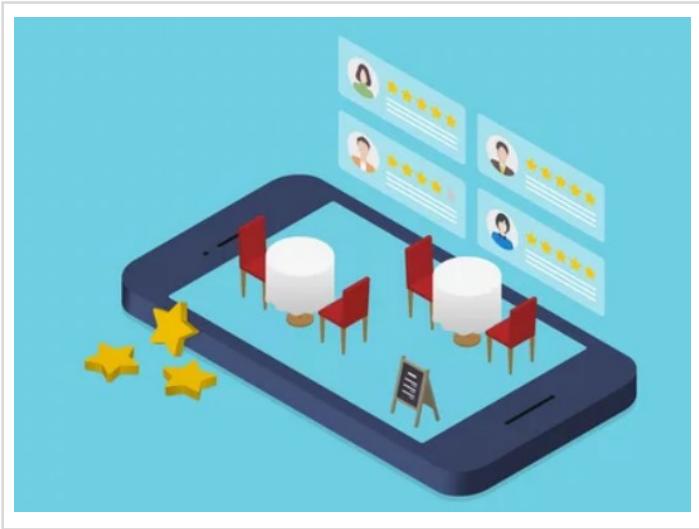
1.3. ຂໍ້ມູນເພີ່ມຕົມກ່ຽວກັບ telephone dictionary

| To learn more about telephone directory, check Wikipedia, https://en.wikipedia.org/wiki/Telephone_directory .



2. Solution

2.1. Telephone directory



- ▶ ใน mission นี้, จะเรียนรู้ว่าในการป้อนข้อมูล review สำหรับร้านอาหารที่สะดวกเวลาค้นหาในอินเตอร์เน็ต.
- ▶ เพื่อให้บันทึกข้อมูล สมมุติสั่ง telephone directory สำหรับร้านอาหารสามร้าน.
- ▶ ในนี้ให้มี ชื่อ และ เบอร์ ของห้างสามร้านอาหารนี้.
- ▶ สรุปโปรแกรม เพื่อเพิ่มที่อยู่ และ review ในร้านอาหารนี้.
- ▶ (ข้อจำกัด: พิจารณาใช้ภาษาไทยที่ได้รับรองเท่านั้น ไม่สามารถใช้ภาษาอังกฤษได้ แต่ reviews จะบ่งบอกว่าภาษาไทยใน strings.)

3. Mission

3.1. ການເຮັດວຽກຂອງ telephone directory

```
1 tele_dir_dic = {'Chinese restaurant':('010-1234-4500'),'Japanese restaurant': ('010-2230-6540'), 'Italian restaurant':('010-3232-7788')}
2 print("previous number")
3 for key, value in tele_dir_dic.items():
4     print(key+ " : ", value)
5 for key, value in tele_dir_dic.items():
6     adr, review = input(key+ 'Please write address and review of: ').split()
7     update = (value, adr, review)
8     tele_dir_dic[key] = update
9 print("Updated number")
10 for key, value in tele_dir_dic.items():
11     print(key+ " : ", value)
```

Previous number

Chinese restaurant:

Japanese restaurant:

Italian restaurant:

Please write address and review for the Chinese restaurant: Northern Busan It was ok

Please write address and review for the Japanese restaurant: Sasang Busan It was delicious

Please write address and review for the Italian restaurant: Donglae Busan Very delicious

Updated number

Chinese restaurant: ('010-1234-4500', 'Northern Busan', 'It was ok')

Japanese restaurant: (010-2230-6540', 'Sasang Busan', 'It was delicious')

Italian restaurant: (010-3232-7788', 'Donglae Busan'. 'Very delicious')

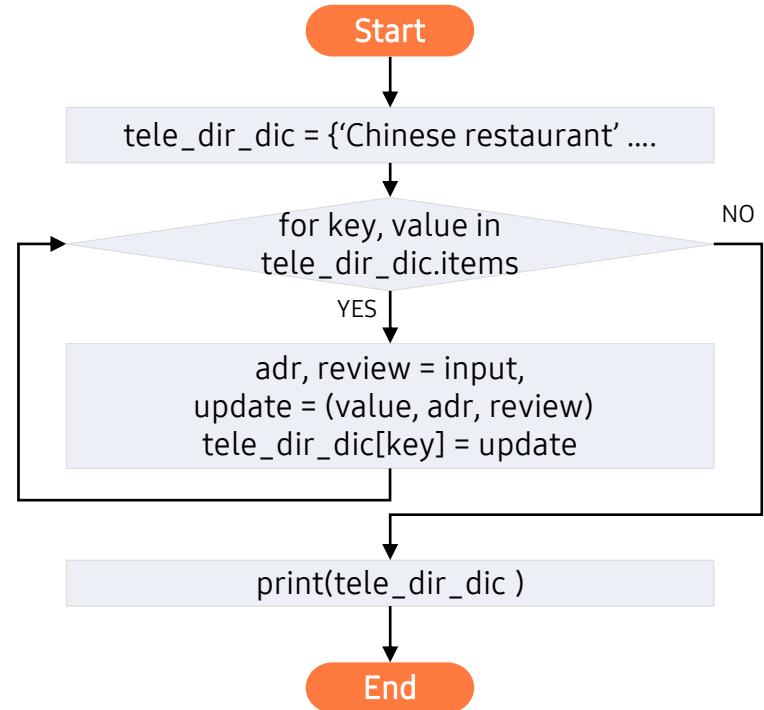
3. Mission

3.2. Programming Plan

Pseudocode

```
[1] Start  
[2] The name and phone number of the restaurant are stored in  
the telephone dictionary.  
[3] for I in list in the telephone dictionary do{  
    [4] Input address and review.  
    Add address and review to the corresponding restaurant.  
}  
[5] Print the updated telephone directory.  
[6] End
```

Flowchart



3. Mission

3.3. Telephone directory final code

```
1 tele_dir_dic = {"Chinese restaurant":("010-1234-4500"),'Japanese restaurant': ('010-2230-6540'), 'Italian restaurant':('010-3232-7788')}\n2 print("previous number")\n3 for key, value in tele_dir_dic.items():\n4     print(key+ " : ", value)\n5 for key, value in tele_dir_dic.items():\n6     adr, review = input(key+ 'Please write address and review of: ').split()\n7     update = (value, adr, review)\n8     tele_dir_dic[key] = update\n9 print("Updated number")\n10 for key, value in tele_dir_dic.items():\n11     print(key+ " : ", value)
```

| Key concept

1. ແປ່ງ List ແລະ Tuple ເປັນ Dictionary

1.1. ມີຫຼາຍວິທີໃນການສ້າງ dictionary

- | ມາສ້າງ dictionary ຈາກ list ແລະ tuple ທີ່ມີແລ້ວ.
- | ສາມາດສ້າງ dictionary ດ້ວຍອີງປະກອບຂອງ list ແລະ tuple ດ້ວຍ keys ໂດຍນຳໃຊ້ fromkeys method ໃນ dictionary class dict.
- | ໃນກໍລະນີ້ນີ້, ທັງໝົດຄໍາເລີ່ມຕົ້ນເປັນ None.

1. ແປງ List ແລະ Tuple ເປັນ Dictionary

1.2. ສ້າງ dictionary ໂດຍນຳໃຊ້ fromkeys method

| ຊອກວິທີນຳໃຊ້ dict.fromkeys method ຈາກຕົວຢ່າງລຸ່ມນີ້.

List Example

```
1 keys = ['a', 'b', 'c', 'd']
2 x = dict.fromkeys(keys)
3 x
{'a': None, 'b': None, 'c': None, 'd': None}
```

| ທຳອິດ ກະກຽມ keys ດັ່ງນີ້ keys=['a', 'b', 'c', 'd'].

| ເວລາເປັນ list ທີ່ປະກອບດ້ວຍ keys ໃນ dict.fromkeys, dictionary ຈະຖືກສ້າງຂຶ້ນ.

| dict.fromkeys ຈະສ້າງ dictionary ດ້ວຍ key list ແລະ ຫຼັງໝົດ values ຈະຖືກກຳນົດຄ່າເລີ່ມຕົ້ນເປັນ None.

Tuple Example

```
1 keys = ('a', 'b', 'c', 'd')
2 x = dict.fromkeys(keys)
3 x
{'a': None, 'b': None, 'c': None, 'd': None}
```

| ນອກຈາກນີ້, ມັນຢັງສາມາດສ້າງ dictionaries ຈາກ tuples ໄດ້.

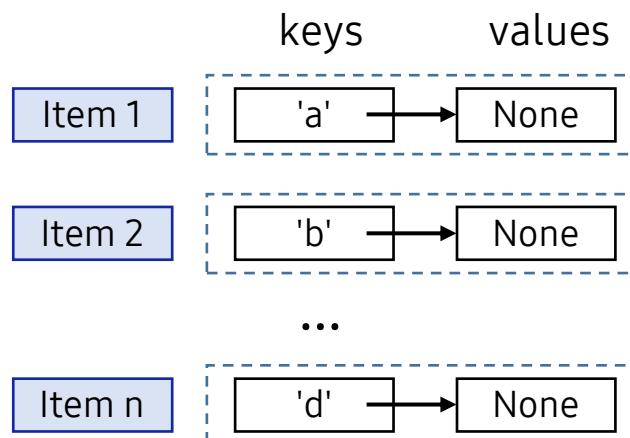
1. ແປງ List ແລະ Tuple ເປັນ Dictionary

1.2. ສ້າງ dictionary ໂດຍນໍາໃຊ້ fromkeys method

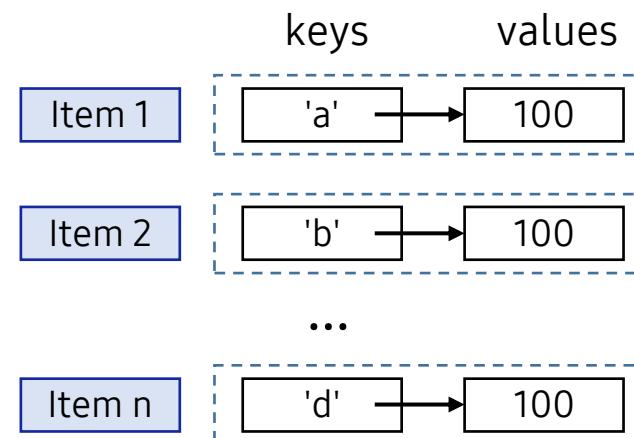
Focus ຖ້າວ່າ key ແລະ value ຕີກເພີ່ມ ແຊ້ນ: dict.fromkeys(key, value), value ແມ່ນບັນຈຸ default value. ມາສິນທຽບຄວາມແຕກຕ່າງຈາກແຜນວາດລຸ່ມນີ້.

```
1 | y = dict.fromkeys(keys, 100)
2 | y
```

```
{'a': 100, 'b': 100, 'c': 100, 'd': 100}
```



ຜົນໄດ້ຮັບຈາກ dict.fromkeys(keys)
key ແລະ default value ຂອງ dictionary



ຜົນໄດ້ຮັບຈາກ dict.fromkeys(keys, 100)
key ແລະ default value ຂອງ dictionary

1. ແປງ List ແລະ Tuple ເປັນ Dictionary

1.3. ເປັນຫຍັງຈຶ່ງຕ້ອງການ defaultdict method?

 ມາເບິ່ງກໍລະນີ KeyError ເກີດຂຶ້ນໃນ dictionary.

| ເກີດຂຶ້ນຜິດພາດຂຶ້ນ ເວລາເຮົາພະຍາມເຊົ້າເຖິງ key ທີ່ບໍ່ມີໃນ dictionary(dic).

```
1 x = {'a': 0, 'b': 0, 'c': 0, 'd': 0}
2 x['z']
```

```
KeyError Traceback (most recent call last)
<ipython-input-99-7867b3f4c074> in <module>
      1 x = {'a': 0, 'b': 0, 'c': 0, 'd': 0}
----> 2 x['z']

KeyError: 'z'
```

- ▶ ຈະຮັດແນວໄດ້ເພື່ອປ້ອງກັນບໍ່ໃຫ້ມີຄວາມຜິດພາດເກີດຂຶ້ນ? ໃນກໍລະນີນີ້ ຕ້ອງໃຊ້ defaultdict method.
- ▶ defaultdict ຈະບໍ່ພິມ error ຖ້າມີການພະຍາມເຊົ້າເຖິງ key ທີ່ບໍ່ມີໃນ dictionary ແລະ ຈະສົງຄ່າ default ອອກມາ.
- ▶ defaultdict ແມ່ນບັນຈຸຢູ່ໃນ collections module.

1. ແປງ List ແລະ Tuple ເປັນ Dictionary

1.4. ວິທີໃຊ້ defaultdict

- | ລຸ່ມນີ້ແມ່ນການສ້າງ default dictionary ດ້ວຍ default value.

```
1 from collections import defaultdict # bring defaultdict from collections module
2
3 keys = ('a', 'b', 'c', 'd')
4 y = dict.fromkeys(keys, 100)
5 y = defaultdict(int) # set default value as int
```

```
1 y['z']
```

```
0
```

- | ບໍ່ມີ key 'z' ໃນ dictionary ທີ່ຊື່ y, ແຕ່ຖ້າຕ້ອງການເຂົ້າເຖິງຄ່າຂອງ key ດັ່ງກ່າວ, ເຊັ່ນ: y['z'], ມັນຈະສິ່ງຄ່າເປັນສູນອອກມາ.
- | ເນື່ອງຈາກຄ່າ default ຂອງຝຶ່ງຊັ້ນ int() ແມ່ນເທົ່າ 0.

```
1 y
```

```
defaultdict(int, {'z': 0})
```

```
1 int()
```

```
0
```

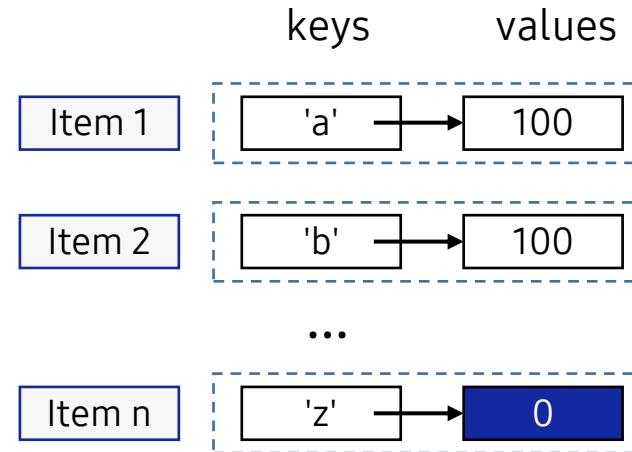
1. ແປ່ງ List ແລະ Tuple ເປັນ Dictionary

1.4. ການນຳໃຊ້ defaultdict

- | ລຸ່ມນີ້ແມ່ນການສ້າງ default dictionary ດ້ວຍ default value ເປັນສູນ.
 - ▶ ດັ່ງສະແດງໃນຮູບລຸ່ມນີ້, key ບໍ່ມີ item ສໍາລັບ 'z', ສະນັ້ນ default value ຈຶ່ງເຫຼົາ 0.

```
1 from collections import defaultdict
2
3 keys = ('a', 'b', 'c', 'd')
4 y = dict.fromkeys(keys, 100)
5 y = defaultdict(int) # int as set default value
6 y['z']
```

0



Result of `y = defaultdict(int)`
key and default value of dictionary

1. ແປງ List ແລະ Tuple ເປັນ Dictionary

1.4. ການນຳໃຊ້ defaultdict

- | ໃນ Unit 14, ພວກເຮົານຳໃຊ້ setdefault method ເພື່ອປະມວນຜົນກໍລະນີບໍ່ມີຄ່າໃນ key.
- | ເວລານີ້, ມາເບິ່ງກໍລະນີການປະມວນຜົນໂດຍນຳໃຊ້ defaultdict ເຊິ່ງເປັນ class ຂອງ Python built-in

```
1 from collections import defaultdict
2
3 word = 'abcbaabca'
4 counter = defaultdict(int)
5 for letter in word:
6     counter[letter] += 1
7
8 counter
defaultdict(int, {'a': 4, 'b': 3, 'c': 2})
```

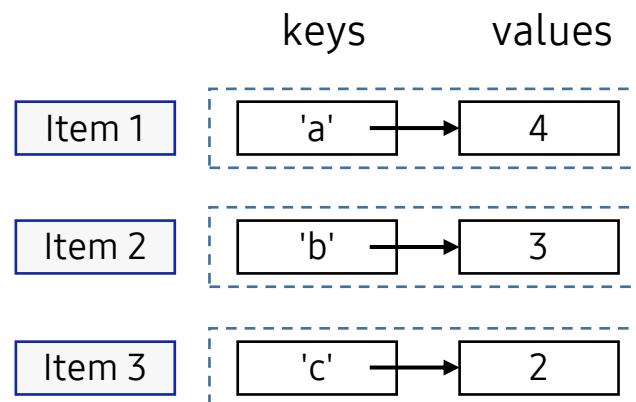
- | Code ນີ້ແມ່ນສະແດງຄວາມທີ່ຂອງຕົວອັກສອນໃນ string ດ້ວຍ key-value.
- | defaultdict class ຈະເອີ້ນ function (int) ແບບອັດຕະໂນມັດ ເພື່ອສິ່ງຄ່າອອກມາ .

1. ແບ່ງ List ແລະ Tuple ເປັນ Dictionary

1.5. ການຕີຄວາມໝາຍ defaultdict code

```
1 from collections import defaultdict  
2  
3 word = 'abcbaabca'  
4 counter = defaultdict(int)  
5 for letter in word:  
6     counter[letter] += 1  
7  
8 counter
```

defaultdict(int, {'a': 4, 'b': 3, 'c': 2})



word = 'abcbaabca'

The counter of this code stores the frequency of appearance of each alphabet.

1. ແປ່ງ List ແລະ Tuple ເປັນ Dictionary

1.6. ຂໍ້ຄວນລະວັງຂອງ defaultdict

 Focus ຖ້າເອີ້ນໂດຍປັດສະຈາກການ int, ມັນຈະສົ່ງຄ່າ ສູນ.

- | ອາດຈະສົງໄສວ່າ default value ເປັນສູນເມື່ອໃສ່ defaultdict(int).
- | int ຈະແປງຈຳນວນຈິງ ຫຼື string ເປັນຈຳນວນເຕັມ ແຕ່ຈະສົ່ງຄ່າເປັນສູນ ຖ້າການເອີ້ນບໍ່ໃສ່ຫຍັງເລີຍໃນ int ດັ່ງສະແດງລຸ່ມນີ້.

```
1 int()
```

0

- | defaultdict ອາດຈະມີຝັງຊັນທີ່ returns ຄ່າສະເພາະ.
- | defaultdict(int) ເປັນຝັງຊັນທີ່ສ້າງ default values ໃຫ້ສະເພາະ int, ສົ່ງເພີນໃຫ້ເປັນສູນ.
- | ຖ້າຕ້ອງການກຳນົດຄ່າ default value ບໍ່ເຫັນກັບສູນ, ສາມາດສ້າງ ແລະ ເພີ່ມຄ່າ default ໃຫ້ກັບຝັງຊັນດັ່ງກ່າວໄດ້ດັ່ງລຸ່ມນີ້.
- | ກໍລະນີລຸ່ມນີ້, list ຫວ່າງເປົ່າຈະກາຍມາເປັນ default value.

```
1 list_dict = defaultdict(list)
2 list_dict
```

```
defaultdict(list, {})
```

```
1 list_dict['list1']
```

```
[]
```

2. พิม Dictionary โดยนำໃຊ້ Loop Statements

2.1. ການพິມໂດຍນໍາໃຊ້ for iteration statement

- | ພິມຕົວປ່ຽນ i ໂດຍໃຊ້ຟັງຂັນ print ຫຼັງຈາກກຳນົດ dictionary ທີ່ຂຶ້ນ x ໃນຄໍາສັ່ງ for ດັ່ງນີ້ for i in x:.
- | ໃນກໍລະນີນີ້, value ຈະບໍ່ຖືກພິມອອກມາ ມີພຽງແຕ່ key ທີ່ຖືກພິມ.
- | ເຮົາຈະພິມທັງ key ແລະ value ໄດ້ແນວໃດ?

```
1 x = {'a': 10, 'b': 20, 'c': 30, 'd': 40}
2 for i in x:
3     print(i, end=' ')
```

a b c d

- | ດ້ວຍເຫດນີ້ຈຶ່ງຄວນກຳນົດ dictionary ຕາມຫຼັງ for in ແລະ items method ຄວນຈະຖືກນໍາໃຊ້ ດັ່ງສະແດງລຸ່ມນີ້.

```
1 for Key, value in dictionary.items():
2     Code to repeat
```

2. พิม Dictionary โดยนำໃຊ້ Loop Statements

2.2. ຕົວຢ່າງການพິມ ໂດຍນໍາໃຊ້ for iteration statement

Ex code ລຸ່ມນີ້ແມ່ນສະແດງຜົນຮັບທັງໝົດເປັນ keys ແລະ values ໃນ dictionary ໂດຍໃຊ້ for loop.

```
1 x = {'a': 10, 'b': 20, 'c': 30, 'd': 40}
2 for key, value in x.items():
3     print(key, value)
```

a 10
b 20
c 30
d 40

Line 1, 2

- for key, value in x.items(): ເອົາ key-value ຈາກ dictionary ທີ່ຂຶ້ນ x ແລະ ເກັບ key ໄວ້ນຕົວປັນ key ແລະ ເກັບ value ໄວ້ນຕົວປັນ value ແລ້ວປະຕິບັດຊ້າຈົນຈີບຈຶ່ງອອກຈາກ loop.
- ຢ່າງໄດ້ກຳຕາມ, ຖ້າ key ແລະ value ຖືກພິມ ໂດຍນໍາໃຊ້ຝັ້ງຊັ້ນ print, ທັງ key-value ກໍຈະພິມອອກມາ.

2. พิม Dictionary โดยนำໃຊ້ Loop Statements

2.2. ຕົວຢ່າງການพິມ ໂດຍນຳໃຊ້ for iteration statement

Ex ຄໍາຕັ້ງການພິມສະເພາະ key ໃນ dictionary ໃຫ້ປະຕິບັດດັ່ງລຸ່ມນີ້.

```
1 x = {'a': 10, 'b': 20, 'c': 30, 'd': 40}
2 for key in x.keys():
3     print(key, end=' ')
```

a b c d

Line 2

- Key ຂອງ dictionary ທີ່ຂຶ້ນ x ສາມາດນຳໃຊ້ x.keys method.

2. พิม Dictionary โดยนำໃຊ້ Loop Statements

2.2. ຕົວຢ່າງການพິມ ໂດຍນຳໃຊ້ for iteration statement

Ex ຖ້າຕ້ອງການພິມສະເພາະ value ໃນ dictionary ໃຫ້ປະຕິບັດດັ່ງລຸ່ມນີ້.

```
1 x = {'a': 10, 'b': 20, 'c': 30, 'd': 40}
2 for value in x.values():
3     print(value, end=' ')
```

10 20 30 40

Line 2

- Value ຂອງ dictionary ທີ່ໃຊ້ x ສາມາດນຳໃຊ້ x.values method.

3. Dictionary ใน Dictionary

3.1. ຕົວຢ່າງ code ຂອງ double dictionary

- | Dictionary ອາດຈະບັນຈຸບັນໃນ dictionary ດັ່ງນີ້ dictionary = {key1: {keyA: keyA}, key2: {keyB: keyB}}.
- | ຂຽນລັດສະໜີ (radius), ມວນສານ (mass), ແລະ ວົງໂຄຈອນ (orbital cycles) ຂອງດາວເຄາະໃນ dictionary ດັ່ງລຸ່ມນີ້.

```
1 terrestrial_planet = {
2     'Mercury': {
3         'mean_radius': 2439.7,
4         'mass': 3.3022E+23,
5         'orbital_period': 87.969
6     },
7     'Venus': {
8         'mean_radius': 6051.8,
9         'mass': 4.8676E+24,
10        'orbital_period': 224.70069,
11    },
12    'Earth': {
13        'mean_radius': 6371.0,
14        'mass': 5.97219E+24,
15        'orbital_period': 365.25641,
16    },
17    'Mars': {
18        'mean_radius': 3389.5,
19        'mass': 6.4185E+23,
20        'orbital_period': 686.9600,
21    }
22 }
23
24 print(terrestrial_planet['Venus']['mean_radius'])
```

6051.8

3. Dictionary ใน Dictionary

3.1. ຕົວຢ່າງ code ຂອງ double dictionary

- | ໃນ dictionary ທີ່ຂຶ້ນ terrestrial_planet ປະກອບດ້ວຍ keys 'Mercury', 'Venus', 'Earth', 'Mars'.
- | ພາຍໃນ key ເລີ່ມື້ຍັງມີ dictionary ພາຍໃນເປັນຄ່າ.
- | ນີ້ແມ່ນ double dictionary ທີ່ມີປະໂຫຍດຫຼາຍໃນການເກັບຂໍ້ມູນແບບລໍາດັບຊັ້ນ.
- | ການເຂົ້າເຖິງຂໍ້ມູນໃນ dictionary ທີ່ຊອນຢູ່ໃນ dictionary, ໃຫ້ວາງ [] (square) ຫຼັງຈາກ dictionary ພ້ອມທັງກຳມີດຄ່າໃຫ້ມັນ ໂດຍມີຂັ້ນຕອນດັ່ງສະແດງລຸ່ມນີ້.

```
dictionary[key][key]
```

```
Dictionary[key][key] = value
```

- | ທີ່ນີ້ dictionary ປະກອບດ້ວຍສອງຂັ້ນຕອນ, ໂດຍມີ ວົງຂໍ ຖືກນຳໃຊ້ສອງຄັ້ງ.
- | ລັດສະໜີຂອງ Venus ສະແດງອອກມາ, ທໍາອິດຫາ Venus ຈາກນັ້ນຈຶ່ງຫາຄ່າ mean radius.

```
1 print(terrestrial_planet['Venus']['mean_radius'])
```

6051.8

4. Assignment และ Copying ຂອງ Double dictionaries

4.1. Assignment และ copying

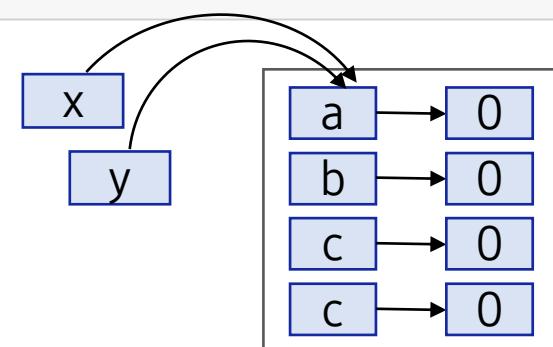
I ມາຮຸນກ່ຽວກັບ assignment และ copying.

- ▶ ຖ້າສ້າງ dictionary ແລະ ເກັບໄວ້ໃນຕົວປັບປຸງ, ເຊັ່ນ: $y = x$, ມັນຄ້າຍກັບວ່າ ມີສອງ dictionaries, ແຕ່ໃນຄວາມເປັນຈິງມັນມີພຽງແຕ່ dictionary ດຽວ.
- ▶ ຖ້າປຽບທຽບ x ແລະ y ກັບຕົວດຳເນີນການ `is` ຜົນໄດ້ຮັບເປັນ `True`. ມັນມີພຽງແຕ່ຕົວປັບປຸງເທົ່ານັ້ນທີ່ແຕກຕ່າງກັນ, ສ່ວນ dictionaries x ແລະ y ພັນນີ້ ທີ່ວ່າເປັນ object ດຽວກັນ.

```
1 x = {'a': 0, 'b': 0, 'c': 0, 'd': 0}  
2 y = x
```

```
1 x is y
```

True



`x` and `y` refer to the same object.

4. Assignment และ Copying ຂອງ Double dictionaries

4.1. Assignment และ copying

Focus ເຖິງແມ່ນວ່າຊື່ຂອງຕົວປັນຈະແຕກຕ່າງກັນ, ແຕ່ພວກມັນຈະຊື້ໄປ object ດຽວກັນ, ສະນັ້ນ ຖ້າມີການປັນແປງຄ່າໃນຕົວປັນໜຶ່ງ ດ່າວິກຕົວປັນໜຶ່ງຈະຖືກປັນຕາມ.

| ເນື້ອງຈາກ x ແລະ y ອ້າງອີງໄປທີ່ object ດຽວກັນ, ການປັນແປງຄ່າໃນ key 'a' ດ້ວຍ `y['a'] = 99` ແມ່ນຈະຖືກປັນຕາມທັງສອງ dictionaries ຄື: x ແລະ y.

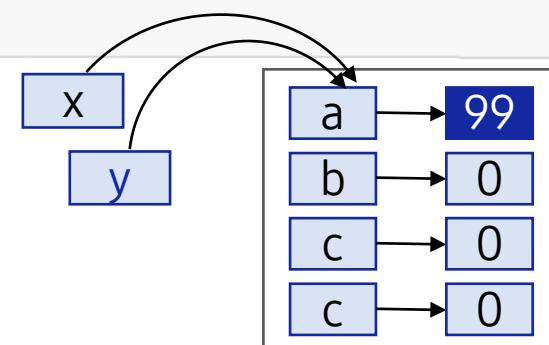
```
1 | y['a'] = 99
```

```
1 | y
```

```
{'a': 99, 'b': 0, 'c': 0, 'd': 0}
```

```
1 | x
```

```
{'a': 99, 'b': 0, 'c': 0, 'd': 0}
```



x and y refer to the same object.

4. Assignment และ Copying និង Double dictionaries

4.3. ជិតខ្សែការណ៍សំណើ ដើម្បី copy ផ្សេងៗ

- | មានការសំណើដើម្បីថា object ទាំងអស់នៅក្នុងក្រប់ក្រង់ ត្រូវបានក្រប់ក្រង់ជាលើសរើស object ទាំងអស់នៅក្នុងវាទែងទៀត។
- | ព័ត៌មានថា ខ្សែការណ៍សំណើនេះ គឺជាផ្សេងៗនៃក្រប់ក្រង់បញ្ហាដែលបានបង្ហាញនៅក្នុងកូដលើកម្មវិធីក្នុងកូដ Python ។
- | ដូច្នេះ ក្នុងកូដលើកម្មវិធីក្នុងកូដ Python ត្រូវបានក្រប់ក្រង់ជាលើសរើស ដើម្បីបានក្រប់ក្រង់បញ្ហាដែលបានបង្ហាញនៅក្នុងកូដលើកម្មវិធីក្នុងកូដ Python ។

```
1 x = {'a': 0, 'b': 0, 'c': 0, 'd': 0}
```

```
1 y = x.copy()
```

- | ពួនឯកសារនេះ ត្រូវបានក្រប់ក្រង់ជាលើសរើស និងយកឱ្យលើកម្មវិធីក្នុងកូដលើកម្មវិធីក្នុងកូដ Python ។

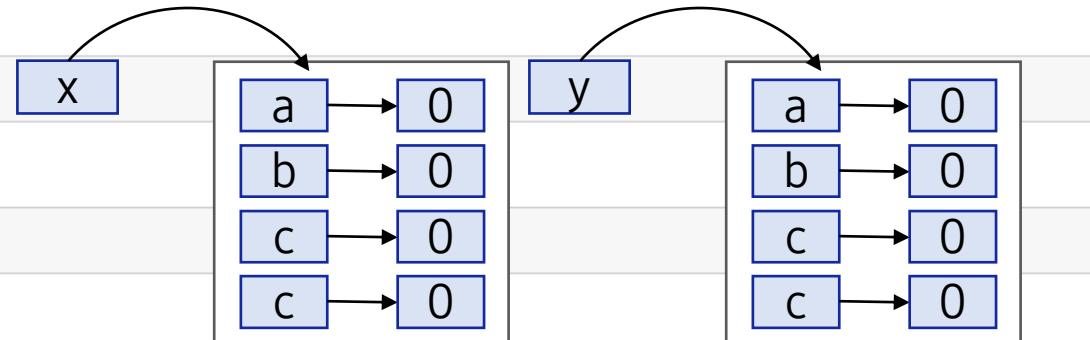
- | ត្រូវបានក្រប់ក្រង់ជាលើសរើស និងយកឱ្យលើកម្មវិធីក្នុងកូដលើកម្មវិធីក្នុងកូដ Python ។

```
1 x is y
```

False

```
1 x == y
```

True



As a result of performing `y = x.copy()`, `x` and `y` refer to different objects.
But the value of the element is the same.

4. Assignment และ Copying ຂອງ Double dictionaries

4.3. ຜົນຂອງການສໍາເນົາ ໂດຍໃຊ້ copy ພັງຊັນ

| ນໍາໃຊ້ພັງຊັນ copy ເພື່ອກຳນົດ objects ດ້ວຍສອງຕົວປຸງ ເປັນ object ທີ່ແຕກຕ່າງກັນ.

```
1 | x = {'a': 0, 'b': 0, 'c': 0, 'd': 0}
```

```
1 | y = x.copy()
2 | y['a'] = 99
```

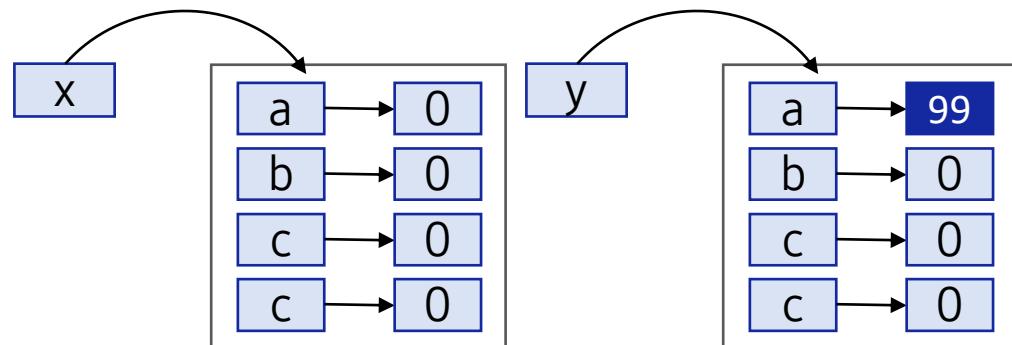
```
1 | x is y
```

False

```
1 | x == y
```

False

- ດຽວນີ້ກໍາລັງສິນທຽບ x ແລະ y ດ້ວຍຕົວດໍາເນີນການ is ຈະສົ່ງຄ່າ False ອອກມາ. ເນື່ອງຈາກໄດ້ກຳນົດ y['a'] = 99.
- ເນື່ອງຈາກສອງ dictionary ເປັນ object ທີ່ແຕກຕ່າງກັນຈຶ່ງຮັດ content ແຕກຕ່າງກັນ ແຊ້ນ: x is y ແລະ x==y ຈະສົ່ງຄ່າ False ອອກມາ.



As a result of performing `y = x.copy()`, x and y refer to different objects.
`y['a'] = 99`, x and y have different element values.

4. Assignment และ Copying ຂອງ Double dictionaries

4.3. ຜົນຂອງການສໍາເນົາ ໂດຍໃຊ້ copy ພັງຊັນ

| ບໍ່ສາມາດສໍາເນົາ ສອງ dictionaries ໄດ້ຢ່າງສີມບູນ ໂດຍໃຊ້ copy methods.

- ▶ ເປັນໄປໄດ້ບໍ່ທີ່ຈະສໍາເນົາ double dictionary ທີ່ບັນຈຸ dictionary ໃນ dictionary ດ້ວຍ copy method?
- ▶ ສ້າງ double dictionary ດັ່ງສະແດງລຸ່ມນີ້ ແລະ copy ມັນດ້ວຍ copy method.

```
1 | x = {'a': {'python': '2.7'}, 'b': {'python': '3.6'}}  
2 | y = x.copy()
```

- ▶ ບຽນ value ຂອງ y, ເຊັ່ນ: y['a']['python'] = '2.7.15' ທີ່ສະແດງໃຫ້ເຫັນທັງ x ແລະ y.
- ▶ ມັນສະແດງໃຫ້ເຫັນວ່າ ການ ສໍາເນົາ double dictionaries ດ້ວຍການໃຊ້ copy method ແມ່ນເປັນໄປບໍ່ໄດ້

```
1 | y['a']['python'] = '2.7.15'  
2 | x
```

```
{'a': {'python': '2.7.15'}, 'b': {'python': '3.6'}}
```

```
1 | y
```

```
{'a': {'python': '2.7.15'}, 'b': {'python': '3.6'}}}
```

4. Assignment และ Copying ຂອງ Double dictionaries

4.4. Deepcopy

 Focus ເພື່ອຮັດໃຫ້ການສໍາເນົາ double dictionary ສໍາເລັດ, ຕ້ອງໃຊ້ ພັງຊັນ deepcopy.

| ພັງຊັນ deepcopy ຈະສໍາເນົາ module ແກນ copy method.

```
1 x = {'a': {'python': '2.7'}, 'b': {'python': '3.6'}}
2 import copy           # call copy module
3 y = copy.deepcopy(x)  # deep copy using deepcopy function
4 y['a']['python'] = '2.7.15'
5 x
{'a': {'python': '2.7'}, 'b': {'python': '3.6'}}
```

```
1 y
{'a': {'python': '2.7.15'}, 'b': {'python': '3.6'}}
```

| ຕອນນີ້, ການປ່ຽນແປງ value ຂອງ dictionary ທີ່ຂຶ້ນ y ບໍ່ສິ່ງຜົນກະທິບຕໍ່ dictionary ທີ່ຂຶ້ນ x.

| copy.deepcopy ພັງຊັນ ໄດ້ແນະນຳ deep copy ທັງໝົດ dictionaries ໃນ double dictionary.

| Deep copy ແມ່ນເຕັກນິກການສໍາເນົາ ທີ່ບໍ່ພຽງແຕ່ແບ່ງປັນທີຢູ່ຂອງຂໍ້ມູນ, ແຕ່ມັນຍັງສ້າງ ແລະ ຈັດເກັບການຂໍ້ມູນທີ່ສໍາເນົາຂໍ້ມູນໃນທ່ວຍຄວາມຈໍາແຍກອອກຕ່າງໜາກ.

Paper coding

- ຕ້ອງເຂົ້າໃຈແນວຄິດພື້ນຖານຂອງຫຼັກສູດນີ້ໃຫ້ຄົບຖ້ວນກ່ອນຈະໄປສູ່ຂັ້ນຕອນຕໍ່ໄປ.
- ການທີ່ບໍ່ເຂົ້າໃຈແນວຄິດພື້ນຖານ ຈະເພີ່ມພາລະໃນການຮຽນຮູ້ຂອງຫຼັກສູດນີ້ ແລະ ຈະເຮັດໃຫ້ບໍ່ປະສົບຜົນສໍາເລັດ.
- ມັນອາດຈະເປັນເລື່ອງທີ່ຍາກຕອນນີ້, ແຕ່ຖ້າຢາກປະສົບຜົນສໍາເລັດໄດ້ນັ້ນ ພວກເຮົາຂໍແນະນຳໃຫ້ເຂົ້າໃຈແນວຄິດພື້ນຖານ ນີ້ຢ່າງລຶກເຊິ່ງ ແລະ ກ້າວໄປສູ່ຂັ້ນຕອນຕໍ່ໄປ.

Q1.

ໃນ tuple ທີ່ຂີ້ study_tup ມີສາມອີງປະກອບຄື: student ID number, name ແລະ phone number, ດັ່ງສະແດງລຸ່ມນີ້. ຈຶ່ງແກ້ໄຂ student_tup ລຸ່ມນີ້ ໂດຍການສ້າງເປັນ dictionary ແລະ ພິມມັນອອກມາໃຫ້ໄດ້ດັ່ງນີ້ {student ID number : [name, phone number]}

Condition for Execution	student_tup = ('211101', 'David Doe', '010-1234-4500'), ('211102', 'John Smith', '010-2230-6540'), ('211103', 'Jane Carter', '010-3232-7788')
Time	7 min

Output example

```
{'211101' : ['David Doe', '010-1234-4500'] }  
{'211102' : ['John Smith', '010-2230-6540'] }  
{'211103' : ['Jane Carter', '010-3232-7788'] }
```



Write the entire code and the expected output results in the note.

Q2. ຂຽນໂປຣແກຣມເພື່ອສະແດງຂໍ້ມູນຂອງນັກສຶກສາປະລິຍາຕີ ໂດຍນຳໃຊ້ student_tup ໃນ Q1 ເພື່ອຮັບ ລະຫັດນັກສຶກສາທີ່ປ້ອນເຂົ້າມາ ແລະ ພິມ ຊຶ່ງ ພ້ອມທັງເບີໂທຂອງນັກສຶກສາຄືນດັ່ງກ່າວອອກມາທາງໜ້າຈຳ.

Example program
execution results.

Enter student ID number : 211101
Name : David Doe
Phone number : 010-1234-4500

Time

5min



Write the entire code and the expected output results in the note.

| Let's code

1. ການຮັດຊ້າໂດຍໃຊ້ dictionary methods

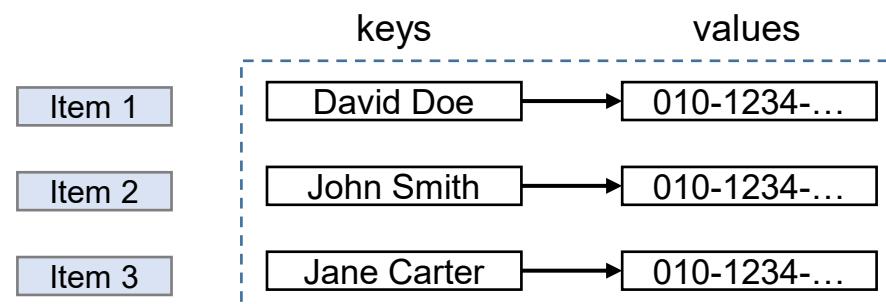
1.1. The return form of items

- | Items ທີ່ສິ່ງຜ່ານ phone_book.items method ແມ່ນຖືກແຍກໂດຍຈຸດ ດັ່ງນີ້ ('Name','Hong Gil-dong').
- | ການສິ່ງປະເພດຂໍ້ມູນຂອງ item ດ້ວຍ items method ແມ່ນ tuple.

```
1 phone_book = {"David Doe": "010-1234-5678", "John Smith": "010-1234-5679", "Jane Carter": "010-1234-5680"}
```

```
1 phone_book.items()
```

```
dict_items([('David Doe', '010-1234-5678'), ('John Smith', '010-1234-5679'), ('Jane Carter', '010-1234-5680')])
```



1. ການເຮັດຊໍາໂດຍໃຊ້ dictionary methods

1.2. keys method

| Keys ແມ່ນ method ຂອງການຮັບ keys ທັງໝົດໃນ dictionary.

- ▶ Items ສາມາດຄົ້ນຫາດ້ວຍ indexes ໃນ list, ແຕ່ values ສາມາດຄົ້ນຫາດ້ວຍ keys ໃນ dictionary.
- ▶ ພິມທັງໝົດ keys ໃນ dictionary, ນໍາໃຊ້ method keys ດັ່ງສະແດງລຸ່ມນີ້.

```
1 phone_book.keys()
```

```
dict_keys(['David Doe', 'John Smith', 'Jane Carter'])
```

1. ການຮັດຊ້າໂດຍໃຊ້ dictionary methods

1.3. values method

| Values ເປັນ method ທີ່ນຳເອົາຄ່າທັງໝົດໃນ dictionary.

- ▶ ນຳໃຊ້ values ເພື່ອພິມຄ່າທັງໝົດໃນ dictionary.

```
1 phone_book.values()
```

```
dict_values(['010-1234-5678', '010-1234-5679', '010-1234-5680'])
```

- ▶ ນອກຈາກນີ້, items ຕ່າງໆ ອາດຈະໃຊ້ເພື່ອພິມຄ່າທັງໝົດໃນ dictionary.

- ▶ ເມື່ອຝັງຊັນ phone_book.items ຖືກເອັນໃຊ້ໃນ for loop, ຈະສົ່ງ key, value ເປັນ tuple ອອກມາ, ດັ່ງສະແດງລຸ່ມນີ້.

```
1 phone_book.items()
```

```
dict_items([('David Doe', '010-1234-5678'), ('John Smith', '010-1234-5679'), ('Jane Carter', '010-1234-5680')])
```

```
1 for name, phone_num in phone_book.items():
2     print(name, ':', phone_num)
```

```
David Doe : 010-1234-5678
John Smith : 010-1234-5679
Jane Carter : 010-1234-5680
```

2. ການນິຍາມ Module ແລະ ການນຳເຂົ້າ Statement

2.1. module ແມ່ນຫຍັງ?

| Module

- ▶ ມັນແມ່ນ script file ຂອງຝຶງຊັ້ນ Python, variables ຫຼື classes.
- ▶ ໃນ Python ມີຫຼາຍ modules ທີ່ສ້າງຂຶ້ນໄດຍ ນັກພັດທະນາຈຳນວນຫຼາຍ.
- ▶ ເນື່ອນຳເອົາ module ທີ່ສ້າງແລ້ວມາໃຊ້, ໃຫ້ຂຽນຊື່ module ໄວດ້ານຫຼັງ 'import'.
- ▶ ເວລານຳໃຊ້ມັນ, ໃຫ້ຂັ້ນດ້ວຍເຄື່ອງໝາຍຈຳ (.) ໃນ module name ດັ່ງສະແດງລຸ່ມນີ້

```
import [module name].[class name].[method name]
```

| Keyword that brings module is import.

```
import module name 1[, module name2, ...]
```

2. ການນິຍາມ Module ແລະ Import Statement

2.1. module ແມ່ນຫຍັງ?

- | ມາເບິ່ງ modules ທີ່ຕ້ອງການຮຽນໃນ Python.
 - ▶ module ທີ່ມາພ້ອມກັບການຕິດຕັ້ງ Python ເຊັ່ນວ່າ Python Standard Library.
 - ▶ ມີຫຼາຍກວ່າ 100 standard libraries, ທີ່ປະກອບດ້ວຍ modules ສໍາລັບ string ແລະ text processing, binary data processing, date, time ແລະ arrangement, numerical operations ແລະ mathematical function modules, file ແລະ directory access, modules ສໍາລັບ Unix system database access, data compression ແລະ graphics modules.

2. ການນິຍາມ Module ແລະ Import Statement

2.2. datetime module ສາມາດຮັບ ແລະ ຈັດການ date ແລະ time module

datetime module

- ▶ ເປັນໜຶ່ງຝັງຊັນທີ່ສາມາດກະກຽມ ແລະ ຈັດການຝັງຊັນທີ່ກ່ຽວຂ້ອງກັບ date ແລະ time.

| ໃນ `datetime.datetime.now` ດ້ວນລຸ່ມນີ້, `datetime` ແມ່ນຊື່ຂອງ module ສ່ວນ `datetime` ຖັນມາແມ່ນຊື່ຂອງ class.

| ຕອນນີ້ method ຈະສົ່ງເວລາປະຈຸບັນອອກມາ.

```
1 import datetime      # The following sentence is omitted  
2 datetime.datetime()
```

```
datetime.datetime(2021, 8, 30, 8, 56, 9, 164997)
```

Line 2

- `datetime.now` method ຈະສົ່ງ ປີ, ເດືອນ, ວັນທີ, ຊົ່ວໂມງ, ນາທີ, ວິນາທີ ແລະ microsecond ປະຈຸບັນອອກມາ.

2. ການນິຍາມ Module ແລະ Import Statement

2.3. today method

I ມາພື້ນຄ່າໂດຍນໍາໃຊ້ today method.

- ▶ today method ຂອງ date class ຈະສິ່ງເວລາປະຈຸບັນຂອງທົ່ວຖິ່ນນັ້ນອອກມາ.
- ▶ today object ປະກອບດ້ວຍ year, month ແລະ day ຂອງເວລາປະຈຸບັນ, ເພື່ອໃຫ້ສາມາດພິມ attributes.

```
1 today = datetime.date.today()  
2 print(today)
```

2021-08-30

```
1 today  
datetime.date(2021, 8, 30)
```

```
1 today.year  
2021
```

```
1 today.month  
8
```

```
1 today.day  
30
```

2. ການນິຍາມ Module ແລະ Import Statement

2.4. dir function

| ມາພິມ ພັງຊັນ ແລະ attributes ຂອງ class ໂດຍໃຊ້ພັງຊັນ dir.

- ▶ ພັງຊັນ dir ຈະສິ່ງທັງໝົດ class lists ແລະ attributes available ໃນ datetime module.
- ▶ MAXYEAR ດີບືສູງສຸດທີ່ datetime object ສາມາດສະແດງໄດ້ ແລະ ມີຄ່າເປັນ 9999.
- ▶ MINYEAR ມີຄ່າເທົ່າກັບ 1.
- ▶ Classes ຕ່າງໆ date, datetime, datetime_CAPI, time, timedelta, timezone ແລະ tzinfo ມີພັງຊັນ date, time, time zone ແລະ time zone information.

```
1 print(dir(datetime))
```

```
['MAXYEAR', 'MINYEAR', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', 'spec__', 'date', 'datetime', 'datetime_CAPI', 'sys', 'time', 'timedelta', 'timezone', 'tzinfo']
```

2. ການນິຍາມ Module ແລະ Import Statement

2.5. replace method

I replace method ສາມາດປ່ຽນ date ຫຼື time.

- ▶ ໃຊ້ເພື່ອປ່ຽນຄ່າ date ຫຼື time ໃນ datetime.

```
1 start_time = datetime.datetime.now()  
2 start_time.replace(month = 12, day = 25)
```

```
datetime.datetime(2021, 12, 25, 9, 0, 35, 190210)
```

Line 1, 2

- ໃນ date object ປະຈຸບັນແມ່ນກຳນົດໂດຍ start_time.
- ນຳໃຊ້ replace method, month ແລະ day ຂອງເວລາປະຈຸບັນ ແມ່ນ ບ່ຽນເປັນ 12 ແລະ 25 ຕາມລຳດັບ.

2. ການນິຍາມ Module ແລະ Import Statement

2.6. as syntax

| ຖ້ານໍາໃຊ້ as syntax, ສາມາດໃຊ້ module name ດ້ວຍການຕັ້ງຊື່ສັນງົກໄດ້.

- ▶ ການຂຽນ "[Module name].[Class name].[Method name]" ໂດຍໃຊ້ຈຳເພື່ອແຍກແຕ່ລະອັນແມ່ນມີຄວາມຫຍຸງຍາກ.
- ▶ as syntax ສາມາດແກ້ໄຂບັນຫານີ້ໃຫ້ກ່າຍຂຶ້ນ ດັ່ງສະແດງລຸ່ມນີ້.

```
import [module name] as [module alias]
```

```
1 import datetime as dt          # use dt instead of datetime from now on
2 start_time = dt.datetime.now()
3 start_time.replace(month = 12, day = 25)

datetime.datetime(2021, 12, 25, 9, 1, 46, 113098)
```



One More Step

import ~ as

- ▶ ວິທີຕັ້ງຊື່ສັນໃຫ້ກັບ module ທີ່ຢ່າວ.
- ▶ ເວລານຳໃຊ້ classes ຫຼື methods ພາຍໃນ modul ຕ້ອງເຊື່ອມຕໍ່ມັນດ້ວຍຈໍາ ແລະ ຖ້າ ຊື່ຂອງ module ຍາວເກີນໄປ ມັນອາດຈະເປັນເລື່ອງທີ່ຫຍຸ້ງຍາກ ໃນການຂຽນ.
- ▶ ໃນກໍລະນີນີ້, ຊື່ໃໝ່ຂອງ module ທີ່ຢ່າວ ຕ້ອງນຳໃຊ້ as.
- ▶ ຊື່ສັນຂອງ module ເປັນທີ່ນີ້ຍົມໃຊ້ກັນເຊັ່ນ: m ສໍາລັບ math, dt ສໍາລັບ datetime, rd ສໍາລັບ random ແລະ t ສໍາລັບ turtle.

ex1) import datetime as dt

ex2) import random as rd

ex3) import math as m

ex3) import turtle as t

2. ການນິຍາມ Module ແລະ Import Statement

2.7. from ~ import ~ statement

- | ຄໍາສັ່ງ from~import ແມ່ນ method ຂອງ picking ທີ່ຕ້ອງການພາຍໃນ module ແລະ ນໍາມັນອອກມາ.
- | ມັນມີຮູບແບບ “from [module name] import [class name].[method name].”
- | ໃນ datetime module, ປິພຽງ class datetime ທີ່ມີ module ນີ້.
- | ການຂຽນ statement, ບໍ່ຈໍາເປັນຕ້ອງຂຽນ module name ພຽງແຕ່ຂຽນ import statement ກໍດັ່ງ

```
1 from datetime import datetime  
2 start_time = datetime.now()  
3 start_time.replace(month = 12, day = 25)
```

```
datetime.datetime(2021, 12, 25, 9, 8, 53, 800714)
```

- | “from [module name] import *” ມີຄວາມໝາຍວ່າເອົາທັງໝົດ variables, functions ແລະ classes ໃນ module.
- | ‘*’ ມີຄວາມໝາຍວ່າ ເອົາຫຼຸກຢ່າງທີ່ມີໃນ module.
- | code ຄໍາສັ່ງລຸ່ມນີ້ ທັງໝົດ classes ໃນ module ຖືກເອີ້ນໂດຍ datetime.
- | ວັນທີໃນມື້ນີ້ຖືກພິມໂດຍໃຊ້ຝັ້ງຊັ້ນ today ຂອງ dateclass (method ຈະສົ່ງວັນທີປະຈຸບັນອອກມາ).

```
1 from datetime import *  
2 today = date.today()  
3 today
```

```
datetime.date(2021, 9, 7)
```

2. ການນິຍາມ Module ແລະ Import Statement

2.8. time module

I time module ໄດ້ກຽມັງຊັນ time.

- ▶ ໃນ module ໄດ້ມີການກະກຽມັງຊັນ time ເພື່ອສະແດງເວລາ.
- ▶ UTC ເປັນເວລາເລີ່ມຕົ້ນຂອງ Unix system 0:0:0 on January 1, 1970 ເອັນວ່າ epoxy.
- ▶ time system ໃຊ້ເປັນມາດຕະຖານໃນລະບົບປະຕິບັດການ Unix ຍັງເອັນວ່າ epoxy time ຫຼື Unix time.
- ▶ code ດຳເສົ່າງລຸ່ມນີ້ ແຈ້ງເວລາເປັນວິນາທີ ຕາມເວລາຂອງ epoxy time (0:0:0 on January 1, 1970).

```
1 import time
2 seconds = time.time()
3 print('Time after epoxy = ', seconds)
```

Time after epoxy = 1630282234.5054421

2. ການນິຍາມ Module ແລະ Import Statement

2.8. time module

I ທີ່ດລອງໃຊ້ localtime ຂອງ time module.

- ▶ local time method ສື່ງຄ່າ epoxy time ໃນ form ຂອງ date ແລະ time.

```
1 import time  
2 unix_timestamp = time.time()  
3 local_time = time.localtime(unix_timestamp)  
4 local_time
```

```
time.struct_time(tm_year=2021, tm_mon=9, tm_mday=7, tm_hour=23, tm_min=16, tm_sec=42, tm_wday=1, tm_yday=250, tm_isdst=0)
```

- ▶ strftime method ພິມຄ່າເວລາທີ່ໄດ້ຮັບຈາກ localtime ໃນຮູບແບບ date/time ທີ່ຜູ້ໃຊ້ຕ້ອງການ.

```
1 import time  
2 unix_timestamp = time.time()  
3 local_time = time.localtime(unix_timestamp)  
4 print(time.strftime('%Y-%m-%d %H:%M:%S', local_time))
```

```
2021-08-30 09:11:42
```

- ▶ %Y ແມ່ນ ປີ, %m ແມ່ນເດືອນ, %d ແມ່ນວັນທີ ແລະ '%Y-%m-%d' ມີຄວາມໝາຍວ່າ 'year-month-day' .
- ▶ '%H:%M:%S': ແຕລະອັນເປັນຕົວແທນຂອງ time(24 hours), minutes (00 ~ 59) ແລະ second (00 ~ 59).

2. ການນິຍາມ Module ແລະ Import Statement

2.9. math module

- | Math module ມີຝັ້ງຊັ້ນທີ່ກ່ຽວກັບຄະນິດສາດ
- | Math module
 - ▶ ພາຍໃນ module ແມ່ນມີຝັ້ງຊັ້ນທີ່ກ່ຽວກັບຄະນິດສາດ.
 - ▶ ມີການກຳນົດຄ່າ Pi , ຄ່າຄົງທີ່ e ແລະ ອື່ນໆ.
 - ▶ ພັ້ນຖານ math ປະກອບມີ sin, cos, tan, log, pow, ceil, floor, trunc, fabs, copysign(x,y) ແລະ ອື່ນໆ.

2. ການນິຍາມ Module ແລະ Import Statement

2.10. ພິມັງຊັນ built-in ໃນ math module ໂດຍນໍາໃຊ້ dir

```
1 import math
2 print(dir(math))      # returns built-in functions of the math module
[ '__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'comb', 'copysign', 'cos', 'cosh', 'degrees', 'dist', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'isqrt', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'perm', 'pi', 'pow', 'prod', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
```

2. ການນິຍາມ Module ແລະ Import Statement

2.11. ຕຽບການນຳໃຊ້ ພັງຊັນ math ໃນ modules math

| ມາເປີ່ງພັງຊັນຕ່າງໆທີ່ໃຊ້ໃນ module.

```
1 import math as m  
2 m.pow(3, 3) # 3 to 3 squares
```

27.0

```
1 mfabs(-99) # The absolute value of -99
```

99.0

```
1 m.ceil(2.1) # 2.1 rounded up
```

3

```
1 m.ceil(-2.1) # -2.1 rounded up
```

-2

```
1 m.floor(2.1) # 2.1 rounded down
```

2

```
1 m.log(2.71828)
```

0.999999327347282

```
1 m.log(100, 10) # value of 100 with logarithm 10 as the bottom
```

2.0

2. ການນິຍາມ Module ແລະ Import Statement

2.11. ຮຽນການນໍາໃຊ້ ພັງຊັນ math ໃນ modules math

| ພັງຊັນທາງດ້ານຄະນິດສາດໃນ math module ດັ່ງສະແດງລຸ່ມນີ້

Function	Description	Example
fabs(x)	Returns the absolute value of real number x.	fabs(-2) → 2.0
ceil(x)	Raises x to a near integer and returns it.	ceil(2.1) → 3, ceil(-2.1) → -2
floor(x)	Rounds down x to a near integer and returns it.	floor(2.1) → 2, ceil(-2.1) → -3
exp(x)	Returns e^x .	exp(1) → 2.71828
log(x)	Returns natural logarithm x.	log(2.71828) → 1.0
log(x, base)	Returns the value of log x with base as base.	log(100, 10) → 2.0
sqrt(x)	Returns the square root of x.	sqrt(4.0) → 2.0
sin(x)	Returns the sine value of x. x in radians.	sin(3.14159/2) → 1, sin(3.14159) → 0
asin(x)	Returns asin of x.	asin(1.0) → 1.57, asin(0.5) → 0.523599
cos(x)	Returns the cos value of x. x in radians.	cos(3.14159/2) → 0, cos(3.14159) → -1
acos(x)	Returns acos of x.	acos(1.0) → 0, acos(0.5) → 1.0472
tan(x)	Returns the tan value of x. x in radians.	tan(3.14159/4) → 1, tan(0.0) → 0
degrees(x)	Changes the angle x from radians to degrees.	degrees(1.57) → 90
radians(x)	Changes the normal angle x to radians.	radians(90) → 1.57

2. ການນິຍາມ Module ແລະ Import Statement

2.12. ການນໍາໃຊ້ radian ໃນຝັງຊັນ Trigonometric

- | ມາຊອກຫາ ເປັນຫຍັງ $\sin(90)$ ບໍ່ເທົ່າ 1 ດັ່ງສະແດງໃນ code ຄໍາສັ່ງລຸ່ມນີ້.
- ▶ ຍ້ອນວ່າຝັງຊັນ trigonometric ໃນ Python's math module ໃຊ້ radian angle ເຊັ່ນ factor value.
 - ▶ ໃນການໃຊ້ມູມທົ່ວໄປ 90 ອີງສາ ໃນ math module, ມັນມີຄວາມຈຳເປັນຕ້ອງແປງ radian ເຊັ່ນ `math.pi/2.0`.

```
1 import math as m  
2  
3 m.sin(0.0)      # value of sin 0
```

0.0

```
1 m.sin(90.0)      # do not use general angle of 90.0
```

0.8939966636005579

```
1 m.sin(m.pi/2)    # radian notation
```

1.0

| Pair programming



Pair Programming Practice

| ແນວທາງ, ກິນໄກ ແລະ ແຜນສຸກເສີນ

ການຈັບຄຸ້ຂຽນໂປຣແກຣມ ເປັນການຈັບຄຸ້ຂອງນັກຮຽນເພື່ອຮັດວຽກມອບໝາຍ, ນັກຮຽນຄວນມີແຜນ ແລະ ສາມາດປ່ຽນແທນກັນໄດ້ ໃນ ກໍາລະນີມີຜູ້ໃຫ້ນີ້ບໍ່ສາມາດເຂົ້າຮ່ວມຮັດວຽກມອບໝາຍໄດ້ບໍ່ວ່າໃນກໍາລະນີໃດກຳຕາມ ເຊິ່ງບັນຫາເລື່ອນັ້ນຕ້ອງຮັດໃຫ້ຈະເຈັ້ງ ແລະ ກຳບໍ່ແມ່ນ ຄວາມຜິດຂອງນັກຮຽນທີ່ຈັບຄຸ້ບໍ່ດີ.

| ຈັບຄຸ້ທີ່ຄ້າຢັກນັ້ນ, ບໍ່ຈໍາເປັນເຫົ້າທຽມກັນ, ຄວາມສາມາດເປັນຄຸ້ຮ່ວມງານ

ການຈັບຄຸ້ຂຽນໂປຣແກຣມ ຈະໄດ້ຮັບຜົນທີ່ກຳຕໍ່ເນື້ອນັກຮຽນມີຄວາມສາມາດຄ້າຢັກນັ້ນ ຫາຍວ່າ ບໍ່ຈໍາເປັນຕ້ອງມີຄວາມສາມາດຄ້າຢັກນັ້ນກໍໄດ້, ແຕ່ວ່າ ການຈັບຄຸ້ ນັກຮຽນທີ່ມີຄວາມສາມາດແຕກຕ່າງກັນຫຼາຍ ກໍຈະຮັດໃຫ້ບໍ່ສົມດຸນກັນ. ຄຸສອນຮູ້ດີວ່າ ການຈັບຄຸ້ກັນບໍ່ແມ່ນ ຍຸດທະສາດ “ແບ່ງເພື່ອເອົາຊະນະ” ແຕ່ເປັນຄວາມ ພະຍາຍາມຮັດວຽກຮ່ວມກັນຂອງນັກຮຽນໃຫ້ປະສົບຜົນສໍາເລັດ. ຄຄວນທີ່ກາເວັ້ນການຈັບຄຸ້ກັນລະຫວ່າງນັກຮຽນອ່ອນ ແລະ ນັກຮຽນເກົ່າງ.

| ກະຕຸ້ນນັກຮຽນໂດຍການໃຫ້ສິ່ງຈຸງໃຈພິເສດ

ຂໍສະເໜີແຮງຈຸງໃຈທີ່ຮັດໃຫ້ນັກຮຽນຈັບຄຸ້, ໂດຍສະເພາະນັກຮຽນທີ່ມີຄວາມສາມາດສູງ. ບາງຄຸສອນໄດ້ພິບວ່າ ການຈັບຄຸ້ຮັດວຽກມອບໝາຍ ແມ່ນມີ ປະໂຫຍດ ສໍາລັບໜຶ່ງ ຫຼື ສອງວຽກມອບເທົ່ານັ້ນ



Pair Programming Practice

| ចំណាំការងារប់ព័ីននៃការសម្រេច

សៀវភៅទាយសំលាបត្វូរដែលមានគោលការណ៍ដែលត្រូវបានដោះស្រាយ និងបង្កើតឡើង ដើម្បីជួយអ្នករាយការ ដើម្បីបង្កើតឡើងការងារប់ព័ីននៃការសម្រេច។ ការងារប់ព័ីននៃការសម្រេច គឺជាការងារប់ព័ីនដែលត្រូវបានដោះស្រាយ និងបង្កើតឡើង ដើម្បីបង្កើតឡើងការងារប់ព័ីននៃការសម្រេច។ ការងារប់ព័ីននៃការសម្រេច គឺជាការងារប់ព័ីនដែលត្រូវបានដោះស្រាយ និងបង្កើតឡើង ដើម្បីបង្កើតឡើងការងារប់ព័ីននៃការសម្រេច។

| សម្រាប់របាយការងារប់ព័ីននៃការសម្រេច

សម្រាប់របាយការងារប់ព័ីននៃការសម្រេច គឺជាការងារប់ព័ីនដែលត្រូវបានដោះស្រាយ និងបង្កើតឡើង ដើម្បីបង្កើតឡើងការងារប់ព័ីននៃការសម្រេច។ ការងារប់ព័ីននៃការសម្រេច គឺជាការងារប់ព័ីនដែលត្រូវបានដោះស្រាយ និងបង្កើតឡើង ដើម្បីបង្កើតឡើងការងារប់ព័ីននៃការសម្រេច។ ការងារប់ព័ីននៃការសម្រេច គឺជាការងារប់ព័ីនដែលត្រូវបានដោះស្រាយ និងបង្កើតឡើង ដើម្បីបង្កើតឡើងការងារប់ព័ីននៃការសម្រេច។

Q1.

List ທີ່ນີ້ student_tuple ທີ່ມີອີງປະກອບເປັນ tuple ດັ່ງສະແດງລຸ່ມນີ້. ພາຍໃນ Tuple ມີອີງປະກອບພາຍໃນ ເຊັ່ນ ລະຫັດນັກສຶກສາ, ຊື່ ແລະ ເບີໂທ. ນຳໃຊ້ຂັ້ນຕຶງກ່າວ ໃຫ້ສ້າງ dictionary ໂດຍໃຫ້ ລະຫັດນັກສຶກສາເປັນ Key ແລະ ຊື່ ເປັນ value ພ້ອມທັງພິມມັນອອກມາຫາງໜ້າຈຳ. ເວລາປ້ອນລະຫັດນັກສຶກສາ ຕ້ອງໜັນໃຈວ່າ ລະຫັດນັ້ນ, ຊື່ ແລະ ເບີໂທ ຖືກພິມອອກມາ ດັ່ງສະແດງລຸ່ມນີ້.

```
| student_tuple = [('211101', 'David Doe', '010-123-1111'), ('211102', 'John Smith', '010-123-2222'),  
|   ('211103', 'Jane Carter', '010-123-3333')]
```

Example Output

```
211101 : David Doe  
211102 : John Smith  
211103 : Jane Carter  
Enter student ID number : 211103  
211103 student is Jane Carter and phone number is 010-123-333.
```