List Part 2

Recap:

List: ที่เก็บของ (อะไรก็ได้ int, string, etc.)
เพื่อ เอาของด้วย index

- Syntax: my_list = [0, 1, "Hello", 0.12]
  neg index: -4 -3 -2 -1
  index: 0 1 2 3 ← list
- index: - เริ่มจาก 0
  - อีกนัย: ตัวสุดท้ายของคือ -1 :
- update: my_list[0] = 10 ← อัพเดตค่าที่ตำแหน่ง 0
  ↳ [10, 1, "Hello", 0.12]

For loop ตาม items:

```
items = [10, 20, 30]
for item in items:
    print(item)
```

Output: 10
20
30

For loop ตาม index:

```
items = [10, 20, 30]
for i in range(len(items)):
    print(items[i])
```

range(3) → [0,1,2]

Output: 10
20
30

List Operators

- \+ : เอามาต่อกัน (concatenation)
  ex. [10,20] + [30,40] → [10,20,30,40]
  ex. [ ] + [10,20] → [10,20]
  empty list, list ว่าง
- * : เอามาต่อกัน n ครั้ง
  ex. [10,20] * 3 → [10,20,10,20,10,20]
- in operator
  ↳ ใช้เช็คว่า ตัว ของอยู่ใน list ไหม? → True / False
  ↳ Syntax: item in list
  True / False
  ex. 10 in [10,20,30] → True
  20 in [0,1,2] → False
  ex: เช็ค user id ในฐานข้อมูล

```
db = [1042, 1059, 2078, 3061]
user_id = int(input("Enter your ID:"))
if user_id in db:
    print("in our db")
else:
    print("Not in our db")
```

ถ้า user_id = 1059
if 1059 in [1042, 1059, 2078, 3061]
→ True
Output: in our db

List Methods

List Methods

- คำสั่ง (function) ที่เรียกใช้งานกับ list

Syntax: list.method()

ต้องมีทุกครั้ง — method ที่ต้องใช้

1. append method

- ใช้เพิ่ม item ลงไปท้ายสุดของ list เท่านั้น!!
- Syntax: list.append(item)

ex. [10,20,30].append(50)
↳ [10,20,30,50]

ex.
```
my_list = [10,20,30]
my_list.append(50)   ← อัพเดต my_list ด้วย!
print(my_list)  → [10,20,30,50]
```

EX: รับค่า list ด้วย user's input

```
n = int(input("Enter n:"))
ages = []   ← empty list
for i in range(n):
    age = int(input("age:"))
    ages.append(age)
print(ages) → [18,17,20]
```

user enters n=3
18
17
20

loop #1:
age = 18
ages.append(18)
↳ [18]

loop #2
age = 17
ages.append(17)
↳ [18,17]

loop #3
age = 20
ages.append(20)
↳ [18,17,20]

---

insert(index, item)

↳ เพิ่มค่า item ที่ตำแหน่ง index
↳ ทำให้ update ตำแหน่งใหม่ของ item ที่อยู่ตรงหรือหลัง index จะถูก push ไป 1 index

↳ EX [1,2,3].insert(1,50)
0 1 1
insert
[1,50,2,3]
0 1 2 3

↳ EX my_list = [10,20,50,70,80]
0 1 2 3 4

↳ Ex
```
my_list = [10, 20, 50, 70, 80]
my_list.insert(3, 99)
print(my_list) → [10, 20, 50, 99, 70, 80]
```

- remove(item) method
  - ↳ remove item ตัวแรก ใน list
    - ↳ error ถ้า item ที่จะ ลบ ไม่ อยู่ใน list

ex
```
[0, 2, 3, 0, 1].remove(0)
```
ตัวแรก
↳ [2, 3, 0, 1]

[0, 3, 2].remove(20) → error! เพราะ: 20 ไม่อยู่ใน list

ex
```
to_rm = int(input())
my_list = [10, 20, 30]
if to_rm in my_list:   ← ห้ามลืมเด็ดขาด!
    my_list.remove(to_rm)
```

- sort() method
  - ↳ เรียง ของ จาก น้อย ไป มาก
  - ↳
```
my_list = [10, 20, 1, 5, -3]
my_list.sort()   ← เรียง ใน ตัว เลย
print(my_list) → [-3, 1, 5, 10, 20]
```

- sorted() function
  - ↳ เรียง จาก น้อย ไป มาก แต่ เป็น function!
    - ↳ จะ ส่ง list ใหม่ ออกมา

ex
```
my_list = [10, 20, 1, 5]
my_list = sorted(my_list)   ← ส่ง list
print(my_list) → [1, 5, 10, 20]
```

---

del function

1. ลบ ของ ตาม ตำแหน่ง index

↳ ลบ ของ ตาม ตำแหน่ง index

↳ Syntax: del list[i]

ex: seq = [11, 22, 33]

del seq[1]

print(seq) → [11, 33]

Useful Built-in functions

- sum(list) → หาผลรวมของ list
  - ↳ e.x. sum([10, 20, 30]) → 60
- min(list) → หาค่าต่ำสุดใน list
  - ↳ e.x. min([5, 7, 1, 2]) → 1
- max(list) → หาค่าสูงสุดใน list
  - ↳ e.x. max([7, 5, 1, 2]) → 7

- หา variance: $\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$

  $\bar{x}$ (mean) $= \frac{1}{n}\sum x_i$ = sum(data) / len(data)

หรือ data มา

```
data = [1, 10, 2, 3, 5]
mean = sum(data) / len(data)
total = 0
for i in range(len(data)):
    total += (data[i] - mean)**2
var = total / len(data)
```

List Comprehension

↳ short-hand version ของ การสร้าง list ใหม่ ด้วย append

Syntax:

```
list1 = []
for item in items:
    list1.append(item)
```

short →

```
list1 = [item for item in items]
```

```
for item in items:
    list1.append(item)
```

list1 = [item for item in items]

↑ Expression ทำงานก่อน !!

ex squared ของ numbers

```
numbers = [10, 20, 30]
square = []
for num in numbers:
    Square.append(num**2)
print(square) → [100, 400, 900]
```

```
numbers = [10, 20, 30]
square = [(num**2) for num in numbers]
print(square) → [100, 400, 900]
```

append

ex: แปลง list ของ cel to F : $(cel \cdot \frac{9}{5}) + 32$

```
temp = [320, 212, 32, 70]
cel = [(t*9/5+32) for t in temp ]
```

List Comprehension with conditions

↳ คือ append เฉพาะ item ที่ตรงเงื่อนไข (ถ้าจริง)

```
numbers = [10, 7, 2, 9, 5, 6]
evens = []
for num in numbers:
    if num % 2 == 0:
        evens.append(num)
print(evens) → [10, 2, 6]
```

evens = [num for num in numbers if num % 2 == 0]

↑ True เท่านั้น ถึงจะ ทำงาน

2D list

↳ list ซ้อน list !!

list1 = [□ , △ , ○]

index 0 1 2

list1[0] → □

list1 = [[1, 2] , [3, 4]]

0 1

list1 = [[1,2], [3,4]]

print(list1[0]) → [1,2]

print(list1[1]) → [3,4]

print(list1[0][0]) → 1

print(list1[1][1]) → 4

list1[0].append(9)

print(list1) → [[1,2,9], [3,4]]

list_2d = [[1, 2, 3],
            [4, 5, 6],
            [7, 8, 9]]

list_2d[i][j] (row, col)

len(list_2d) → 3

print(list_2d[0]) → [1,2,3]

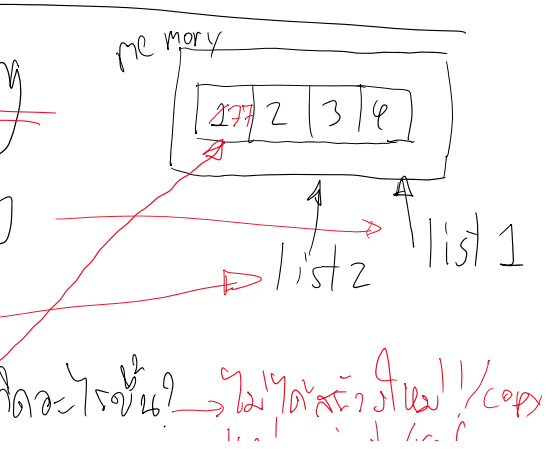
print(list_2d[0][2]) → 3

print(list_2d[2][0]) → 7

List Referencing

list1 = [1, 2, 3, 4]

list2 = list1



แล้วถ้าอยากได้ → ไม่ใช่การสร้างใหม่!! / copy!!

list1 = list2

list1[0] = 77 → เกิดอะไรขึ้น? → ไม่ได้สร้างใหม่!/copy!!
ต้อง point/ref ไปที่ memory เดียวกัน!

print(list1) → [77, 2, 3, 4]

print(list2) → ~~[1, 2, 3, 4]~~ [77, 2, 3, 4]

ถ้าเปลี่ยนอันใดอันหนึ่ง อีกอันก็เปลี่ยน!

Summary

- List basic: index, update
- List Operators: + (ต่อกัน), * (ต่อกันหลายๆ), in (เช็คในlist)
- List methods: append, insert, remove, sort
- Built-in functions: sum, min, max, del
- List comprehension: [x for x in seq]
  - ↳ สร้าง list ใหม่ จาก append
- 2D list ✓
- list reference → copy ด้วย = ไม่ได้