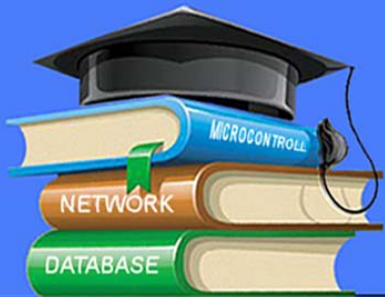


Chapter 8: List



Computer Engineering, Kasetsart University Kamphaeng Sean Campus

Outline

◆ Basic Loop

- Review: while Statement
- Review: for Statement
- Loop statement with if statement

◆ List

- Overview
- List Creation
- Accessing list element
- Python list Methods

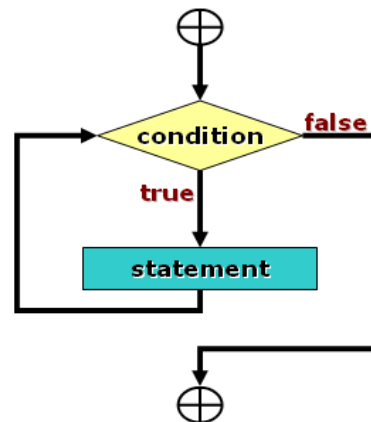
While statement

◆ ไวยากรณ์ (Syntax):

while (เงื่อนไข):
 คำสั่ง

◆ ตัวอย่าง

```
while (x >= 10):  
    x += 1
```



ทำงาน **คำสั่ง** ตราบเท่าที่เงื่อนไขเป็นจริง

For statement

◆ ไวยากรณ์ (Syntax):

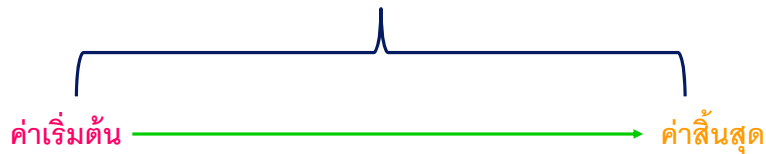
for **ตัวแปรสำหรับนับรอบ** in range(**ค่าเริ่มต้น**, **ค่าสิ้นสุด**, **ค่าส่งปรับค่า**):

- range() : ฟังก์ชันที่ทำให้ for loop ทำการนับค่าให้อยู่ในช่วงที่ของ **ค่าเริ่มต้น** และ **ค่าสิ้นสุด**
- **ตัวแปรสำหรับนับรอบ** เป็นตัวแปรนับรอบ
- **ค่าเริ่มต้น** กำหนดค่าเริ่มต้นให้กับตัวนับรอบ (หากไม่กำหนดจะมีค่าเป็น 0)
- **ค่าสิ้นสุด** กำหนดค่าสิ้นสุดให้กับตัวนับรอบโดยกำหนดให้มีค่าเป็นบวกหรือลบ
- **ค่าส่งปรับค่า** กำหนดค่าเพิ่มค่าและลดค่าตัวนับรอบ (หากไม่กำหนดจะมีค่าเป็น +1)
- **คำสั่ง** เป็นคำสั่งที่ให้อำนาจในแต่ละรอบซึ่งเป็นคำสั่งเดียว หรือคำสั่งประกอบภายใต้บล็อกถัดไปจากคำสั่ง for loop

For statement

◆ ไวยากรณ์ (Syntax):

```
for ตัวแปรสำหรับนับรอบ in range(ค่าเริ่มต้น,ค่าสิ้นสุด, ค่าสั่งปรับค่า):
```



Quick Check1

◆ จงบอกผลลัพธ์ของชุดคำสั่งต่อไปนี้

a. x=5

```
while True:
    x=x*x
    if(x>5):
        break
print(x)
```

c. x = 20

```
for i in range(20,0,-5):
    print(x-i)
```

d. x = 20

```
for i in range(x):
    print(i)
```

b. x=5

```
while(x<5):
    x=x*x
print(x)
```

e. n=5

```
j=1
while(n%j!=0) :
    j+=1
print(j)
```

Loop statement with if statement

◆ รูปแบบของการนำ if statement มารวมกับ loop statement มีได้ 2 แบบคือ

if (condition):	while (condition):
<pre>while(condition): ...</pre>	<pre>if(condition): </pre>

Loop statement อยู่ภายใน if statement if statement อยู่ภายใน loop statement

Loop statement with if statement

◆ Example1 โปรแกรมคำนวณผลรวมของเลขคู่ตั้งแต่ 1 จนถึง n

```
i = 1
sum=0
while (i <= n):
    if (i%2==0):
        sum=sum+i
    i = i + 1
```

n =6	
1	0
2	0+2=2
3	2
4	2+4=6
5	6
6	6+6=12
7	
End Loop!!	

Loop statement with if statement

- ◆ Example2 โปรแกรมบวกผลรวมตัวเลขในช่วงที่กำหนด

Enter number: **3 5**
The summation of 3 to 5 is 12

Enter number: **5 3**
The summation of 5 to 3 is 12

Enter number: **5 -3**
The summation of 5 to -3 is 9

Enter number: **4 4**
The summation of 4 to 4 is 4

Loop statement with if statement

- ◆ Example2 โปรแกรมบวกผลรวมตัวเลขในช่วงที่กำหนด

```
sum=0
m,n = input().split()
m = int(m)
n = int(n)
if(m < n):
    for i in range(m,n+1):
        sum+=i
else:
    for i in range(n,m+1):
        sum+=i

print(sum)
```

กรณีบวกจากน้อยไปมาก

กรณีบวกจากมากไปน้อย

List Overview

- ◆ ถ้าต้องการเขียนโปรแกรมคำนวณผลรวมตัวเลข 1000 จำนวน

- ประกาศตัวแปร 1000 ตัว

num1, num2, num3,num4,num5,...,num100,...,num1000

- กำหนดค่าตัวแปร 1000 ตัว

num,num2,num3,num4,...,num1000 = input().split()

- คำนวณผลรวมตัวแปร 1000 ตัว

total =num1+ num2+num3+ num4+...+num1000

List Overview

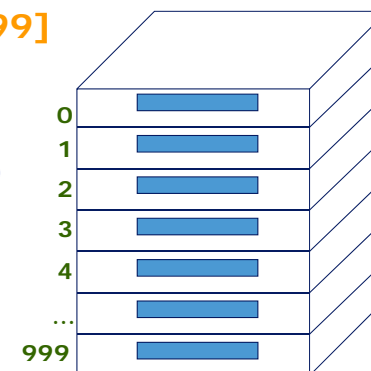
- ◆ ลิสต์ (list) ตัวแปรชุด ที่เก็บข้อมูลไว้ในตัวแปรเดียว

- ◆ ลิสต์ช่วยอำนวยความสะดวกในการประกาศและใช้งานตัวแปรจำนวนมาก

- ◆ ตัวอย่าง ประกาศตัวแปร list ชนิดจำนวนเต็ม เก็บค่า 0-999

num= [0,1,2,3,...,999]

num=list(range(0,100))



List Overview

- ◆ การเขียนโปรแกรมคำนวณผลรวมตัวเลข 1000 จำนวน

```
total=0;
num1,num2,num3,num4,...,
num1000 = input().split()
total =int(num1)+ int(num2)+
int(num3)+ int(num4)+...+
int(num1000)
print(total)
```

```
num = []
index = 1000
total = 0
for i in range(index):
    num.append(input())
for i in range(index):
    total += int(num[i])
print(total)
```

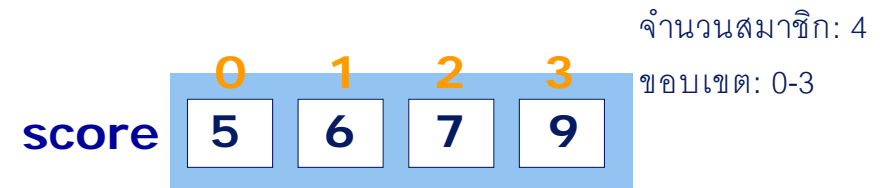
13

List Creation

- ◆ ไวยากรณ์ (Syntax):

ชื่อตัวแปร = [ค่าที่1,ค่าที่2, ...]

- ◆ ตัวอย่าง: **score = [5,6,7,9]**



14

More on List Creation

- ◆ **List1 = []**

ลิสต์ว่าง

- ◆ **List2 = [1.2, 3.5, 4.0]**

ลิสต์ที่เก็บค่าทศนิยม 3 ตัว

- ◆ **List3 = ['a','b','c','d']**

ลิสต์ที่เก็บค่าตัวอักษร 4 ตัว

15

More on List Creation

```
List1= [ ]
print(List1)

List2 = [1.2, 3.5, 4.0]
print(List2)

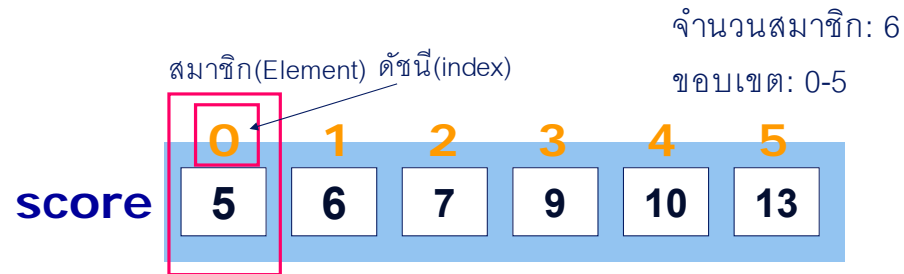
List3 = ['a','b','c','d']
print(List3)
```

```
[]
[1.2, 3.5, 4.0]
['a', 'b', 'c', 'd']
```

16

Accessing list elements

- ◆ สมาชิกของ list เรียกว่า Elements
- ◆ ตัวเลขที่ใช้ระบุสมาชิกของ list เรียกว่า **ดัชนี** (Indexes)
- ◆ ตัวอย่าง: **score = [5,6,7,9,10,13]**

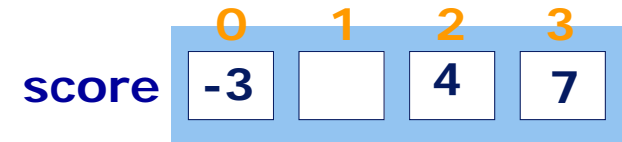


17

Accessing list elements

- ◆ การเข้าถึงตัวแปร list จะต้องระบุ ดัชนี ของสมาชิก list
- ◆ ดัชนีของ list จะเริ่มต้นที่ค่า 0 เสมอ

score = [-3, None, 4, 7]



```
score[3-2] = score[2]+score[3]  
print(score[1])
```

18

Quick check2

- ◆ จงบอกผลลัพธ์ทางจอภาพ เมื่อประกาศลิสต์ดังต่อไปนี้

```
x = [7,3,2,5,6]
```

```
y = [1.1,9.2,2.5,6.3]
```

```
print(x[0])
```

```
print(x[3]+y[3])
```

```
print(x[5])
```

19

Python list Methods

- ◆ คำสั่งต่างๆ ใน Python ที่ใช้บริหารจัดการข้อมูลในลิสต์

Method	Description
append()	เพิ่มสมาชิกเข้าไปในช่องท้ายสุดของลิสต์
insert()	เพิ่มสมาชิกเข้าสู่ลิสต์สู่ดัชนีที่กำหนด
len()	นับจำนวนสมาชิกภายในลิสต์
pop()	ลบสมาชิกภายในลิสต์ที่อยู่ภายในดัชนีที่กำหนด

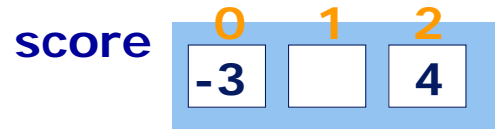
20

Python list Methods

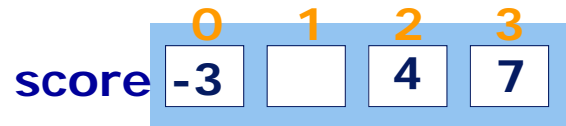
- ◆ ตัวอย่างการเพิ่มสมาชิกด้วยคำสั่ง `append()`

■ ไวยากรณ์ (Syntax): ตัวแปรลิสต์.append(ค่าที่ต้องการเพิ่ม)

`score = [-3, None, 4]`



`score.append(7)`



21

Python list Methods

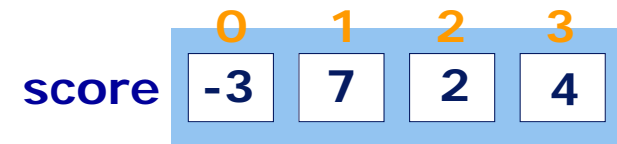
- ◆ ตัวอย่างการเพิ่มสมาชิกด้วยคำสั่งในตำแหน่งดัชนีที่กำหนด `insert()`

■ ไวยากรณ์ (Syntax): ตัวแปรลิสต์.insert(ดัชนี,ค่าที่ต้องการเพิ่ม)

`score = [-3, 2, 4]`



`score.insert(1, 7)`



22

Python list Methods

- ◆ การแสดงขนาดของลิสต์ด้วยคำสั่ง `len()`

■ ไวยากรณ์ (Syntax): `len(ตัวแปรลิสต์)`

`score = [-3, 4, 2, 7]`



`print(len(score))`

4

23

Python list Methods

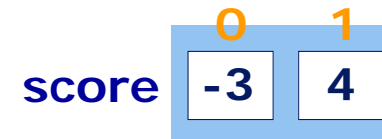
- ◆ ตัวอย่างการลบสมาชิกด้วยคำสั่งในตำแหน่งดัชนีที่กำหนด `pop()`

■ ไวยากรณ์ (Syntax): ตัวแปรลิสต์.pop(ดัชนีที่ต้องการลบ)

`score = [-3, 7, 4]`



`score.pop(1)`



24

Accessing lists elements

◆ ตัวอย่างการกำหนดค่าให้ตัวแปร list

num = [] ได้ลิสต์ว่าง ๆ มี len(num) เป็น 0

```
num.append(int(input()))
num.insert(1,5)
num.insert(2,num[0]+num[1])
print(num)
```

```
num[0]=73
print(num)
```



25

Accessing lists elements

◆ ตัวอย่างการกำหนดค่าให้ตัวแปร list

letter = []

```
for i in range(4):
    letter.insert(i,input())
```

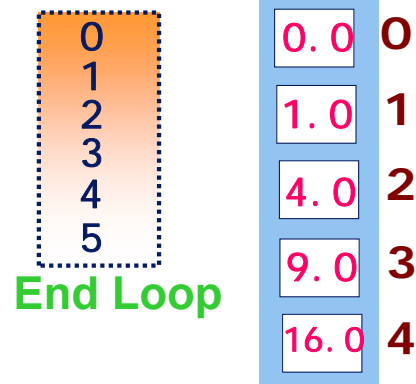


0	A	letter[0]='A'
1	B	letter[1]='B'
2	C	letter[2]='C'
3	D	letter[3]='D'

Accessing lists elements

◆ Example3 ตัวอย่างการใช้ค่าตัวแปร list

```
A = []
i = 0
while (i < 5):
    A.append(i ** 2.0)
    print(A[i])
    i = i + 1
```



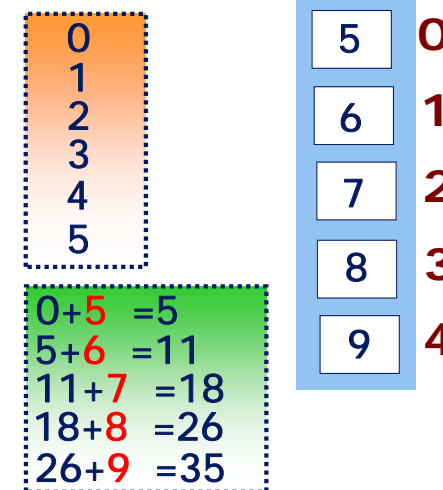
เขียนโปรแกรมนี้ใหม่ โดยใช้คำสั่ง for

27

Accessing lists elements

◆ Example4 การบวกผลรวม list

```
A = [5,6,7,8,9]
i = 0
sum=0
while (i < len(A)):
    sum=sum+A[i]
    i = i + 1
print(sum)
```

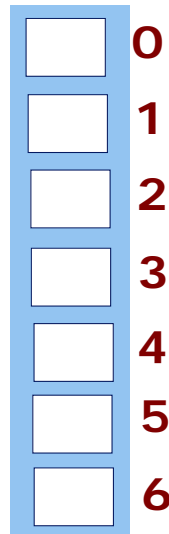


28

Quick Check3

- ◆ กำหนดให้โปรแกรมมีขั้นตอนการทำงานดังนี้

```
เริ่มต้นโปรแกรม
รับค่า count
ทำซ้ำ โดยให้ i = 0 ถึง count-1
x.append( i + 1)
จบทำซ้ำ
จบโปรแกรม
```



ถ้าโปรแกรมนี้ประมวลผลจนจบ กำหนดให้ count เป็น 7
ให้นักวาดโครงสร้างของลิสต์ x และค่าที่เก็บในลิสต์

Quick Check4

- ◆ ประกาศตัวแปร lists ชื่อ temp
- ◆ ใช้คำสั่ง for รับค่าจำนวนจริง เข้าตัวแปร temp จากคีย์บอร์ด ทั้ง 25 ค่า
- ◆ จากนั้นใช้คำสั่ง for คำนวณผลรวมของค่าตัวแปร temp

Quick check5

- ◆ ให้นักเขียนโปรแกรมเพื่อหาค่าที่มากที่สุดที่ป้อนเข้ามาทางคีย์บอร์ด N ตัว โดย N ก็รับมาจากคีย์บอร์ดเช่นกัน โดยให้นักเขียนทั้งแบบไม่ใช้ลิสต์และใช้ลิสต์

```
N: 5
number: 5
number: 9
number: 8
number: 12
number: 3
Maximum number is 12
```

Python lists Methods

- ◆ คำสั่งต่างๆ ใน Python ที่ใช้บริหารจัดการข้อมูลในลิสต์ (เพิ่มเติม)

Method	Description
index()	แสดงดัชนีของสมาชิกที่มีค่าเท่ากับค่าที่กำหนด
remove()	ลบสมาชิกตัวแรกที่มีค่าเท่ากับค่าที่กำหนด
sort()	เรียงลำดับสมาชิกในลิสต์จากน้อยไปมาก
count()	นับจำนวนสมาชิกในลิสต์ที่มีค่าเท่ากับค่าที่กำหนด
clear()	ลบสมาชิกทั้งหมดภายในลิสต์
copy()	คัดลอกสมาชิกภายในลิสต์

https://www.w3schools.com/python/python_lists_methods.asp