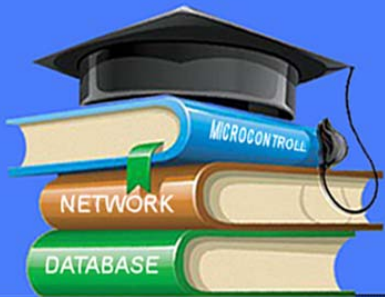


Computers & Programming

Chapter 7: Advance Control Flow



Computer Engineering, Kasetsart University Kamphaengsaen Campus

Contents

Python
Advance
Control flow

Review: condition statements

Review: loop statements

loop statement with if statement

2

1. Review: condition statements

- ◆ คำสั่งใช้เพื่อตัดสินใจว่า จะเลือกทำทางใดทางหนึ่ง ขึ้นอยู่กับเงื่อนไข ซึ่งมีผลลัพธ์เป็น boolean
- ◆ ประกอบด้วยสองส่วน คือ
 - ส่วนของการตรวจสอบเงื่อนไข
 - ส่วนของคำสั่งที่จะเลือกทำ
- ◆ ชนิดของคำสั่ง
 - if
 - if-else

3

1.1 If statement

◆ รูปแบบ

```
if (นิพจน์หรือเงื่อนไข) :  
    คำสั่ง
```

if (นิพจน์หรือเงื่อนไข) :

คำสั่ง
คำสั่ง

◆ ตัวอย่าง

```
i = 4  
if (i%4-1 < 0):  
    print(i, " is not ordinary")  
print("end.")
```

```
i = 4  
j = 2  
if (i-j%4 < 0):  
    print("extraordinary")  
print("end.")
```

4

1.2 If-else statement

◆ คำสั่ง if/else ใช้สำหรับกรณีที่ ต้องมีการตัดสินใจ 2 ทางเลือก

- ถ้าเงื่อนไขเป็นจริงจะทำคำสั่งภายในส่วนของ if
- ถ้าเงื่อนไขเป็นเท็จจะทำคำสั่งภายในส่วนของ else

◆ รูปแบบ

```
if (นิพจน์หรือเงื่อนไข) :  
    คำสั่งที่ 1  
else:  
    คำสั่งที่ 2
```

```
if (นิพจน์หรือเงื่อนไข) :  
    คำสั่ง  
    คำสั่ง  
else:  
    คำสั่ง  
    คำสั่ง
```

5

1.2 If-else statement

◆ ตัวอย่าง

```
x = bool(4)  
y = False  
if (not x or y):  
    print("x = ",x)  
    print("y = ",y)  
    print("seems to be different")  
else:  
    print("still cannot conclude")
```

```
x = 4  
y = False  
if (x > 0):  
    if (y):  
        print("true")  
    else:  
        print("false")
```

6

Exercise1

◆ จงเขียน flow chart เพื่อกำหนดความสมดุลของร่างกายระหว่างความสูงและน้ำหนักตัว ซึ่งเราเรียกว่า BMI (Body Mass Index) และมีสูตรการคำนวณดังนี้

$$\text{BMI} = \text{น้ำหนัก} * 703 / \text{ความสูง}^2$$

BMI	ความหมาย
< 20	ผอมเกินไป (underweight)
20 – 25	ปกติ (normal)
25.1 – 30	อ้วนเกินไป (overweight)
> 30	เป็นโรคอ้วน (obese)

หมายเหตุ :

ได้รับน้ำหนักและส่วนสูงทางคีย์บอร์ด โดยตรวจสอบว่า หากค่าใดค่าหนึ่งติดลบให้แสดง ข้อความ error

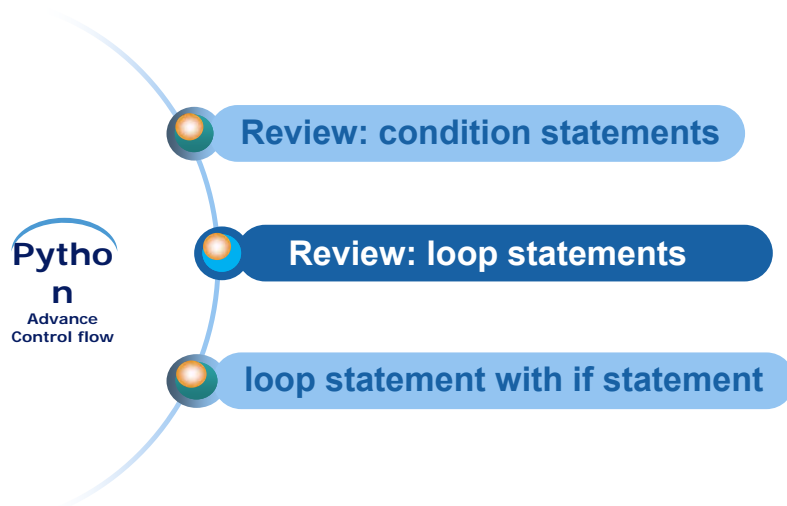
7



Exercise1

8

Contents



9

2. Review: loop statements

- ◆ เป็นคำสั่งของการทำซ้ำภายใต้เงื่อนไข โดยหากเงื่อนไขเป็นจริงก็จะวนกลับมาที่จุดเริ่มต้นของคำสั่งภายใต้ loop และทำไปเรื่อยๆ จนกว่าเงื่อนไขจะเป็นเท็จ
 - การทำงานจะเสร็จสิ้น ณ การตรวจสอบเงื่อนไขที่ต้องเป็นเท็จเท่านั้น
- ◆ ชนิดของคำสั่ง
 - while
 - for

10

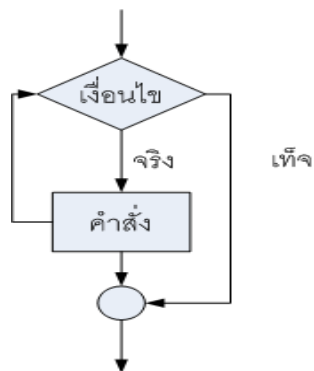
2.1 while statement

◆ Syntax

```
while(นิพจน์หรือเงื่อนไข) :  
    คำสั่ง
```

```
while(นิพจน์หรือเงื่อนไข) :  
    คำสั่ง  
    คำสั่ง
```

◆ การประมวลผล



11

2.1 while statement

◆ ตัวอย่าง

```
x = 4  
while (x > 0):  
    print(x, end=' ')  
    x = int(x/2)
```

```
letter = 97  
while( letter <= 101 ):  
    print(chr(letter))  
    letter = letter+ 1
```

12

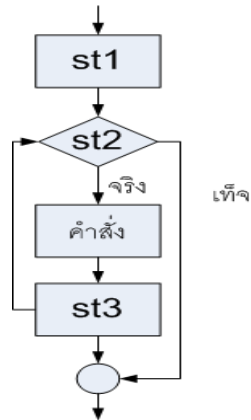
2.2 for statement

◆ Syntax

```
for st in range(st1,st2,st3)
    คำสั่ง
```

```
for st in range(st1,st2,st3)
    คำสั่ง
    คำสั่ง
```

◆ การประมวลผล



13

2.2 for statement

◆ ตัวอย่าง

```
for x in range(4,0,-2):
    print(x,end=' ')
```

```
for letter in range(97,102):
    print(chr(letter))
```

14

while vs for

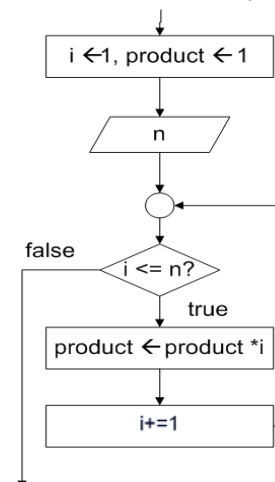
- ◆ for เป็นการวนลูปแบบกำหนดช่วง เหมาะสมสำหรับสถานการณ์ที่รู้จำนวนรอบที่แน่นอน
- ◆ while เป็นการวนลูปแบบใช้เงื่อนไขกำหนด เหมาะสมสำหรับกรณีที่ **ไม่** รู้จำนวนรอบที่แน่นอน

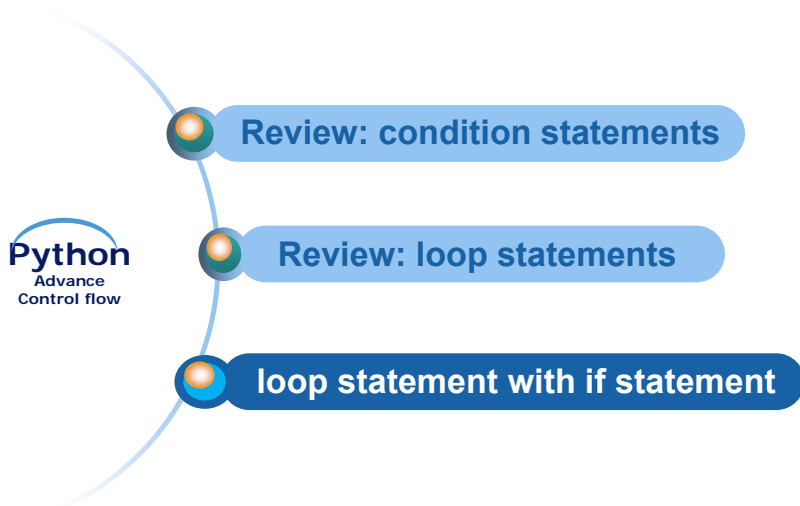
15



Exercise2

จาก flow chart ข้างล่าง โปรแกรมนี้ทำอะไร และแปลง flow chart เป็นคำสั่งในภาษา Python





3. Loop statement with if statement

- ◆ คำสั่งประเภทวนซ้ำ หรือ คำสั่งแบบเงื่อนไข ก็คือ คำสั่งหนึ่ง ซึ่งก็ไม่ได้ต่างกับคำสั่ง print() หรือ input() หรืออื่นๆ แต่มีความซับซ้อนมากกว่า
- ◆ ในการแก้ไขปัญหาหนึ่ง อาจต้องมีคำสั่งภายใต้ loop หรือ if ที่ซับซ้อนกว่าการ print() หรือ input() เช่น
 - โปรแกรมคำนวณผลรวมของเลขคู่ตั้งแต่ 1 จนถึง n
 - คำสั่ง if ภายใต้ loop
 - โปรแกรมแยกตัวประกอบของเลข n ใดๆ
 - คำสั่ง if ภายใต้ loop

3. Loop statement with if statement

- ◆ ตัวอย่าง โปรแกรมคำนวณผลรวมของเลขคู่ตั้งแต่ 1 จนถึง n

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

n = 6

1	1
2	1+3 = 4
3	4
4	4+5 = 9
5	9
6	
7	

End Loop!!

Exercise3

- ◆ จงเขียนโปรแกรมเพื่อคำนวณหาผลบวกของเลขจำนวนเต็มตั้งแต่ 0 - n โดยที่ $n \geq 0$ โดยให้รับค่า n ทางคีย์บอร์ด
 - $n \geq 0$ หาผลบวก
 - $n < 0$ แสดงข้อความ Invalid input!

3.1 break statement

◆ คำสั่ง break ใช้ในการออกจาก block ของคำสั่งเรื่องการวนซ้ำ

◆ โครงสร้าง

```
for (...):  
    ...  
    if(expression):  
        break  
    ...
```

```
while(...):  
    ...  
    if(expression):  
        break  
    ...
```

21

3.1 break statement

◆ ตัวอย่าง

```
letter = 'd'  
  
while( letter <= 'n' ):  
    print(letter)  
    if( letter == 'k' ):  
        break  
    letter = chr(ord(letter) + 1)
```

โปรแกรมนี้ทำอะไร?

22

Exercise4

จงบอกผลลัพธ์ที่ได้จากโปรแกรมข้างล่าง

```
for i in range(1,100):  
  
    print(i)  
    if (i == 4):  
        break
```

23

3.2 Continue statement

◆ คำสั่ง continue ใช้ในการยกเลิกการวนรอบนั้น แล้วขึ้น loop รอบใหม่ แทนที่จะเป็นการออกจาก loop เหมือนอย่าง break

◆ โครงสร้าง

```
for (...):  
    ...  
    if(expression):  
        continue;  
    ...
```

```
while(...):  
    ...  
    if(expression):  
        continue;  
    ...
```

24

3.2 Continue statement

◆ ตัวอย่าง

```
for x in range(7):
    if(x == 5):
        continue
    print(x, end=' ')
```

25

Exercise5

โปรแกรมข้างล่างทำอะไร และได้ผลลัพธ์อย่างไร

i **sum**

```
i = 0
sum = 0
while (i<10):
    i+=1
    if (i%3 != 0):
        continue
    sum = sum + i

print(sum)
```

26

Exercise 6

◆ จงเขียนโปรแกรมที่รับจำนวนเต็ม 2 ตัวเรื่อยๆ แล้วตรวจสอบว่าเลขที่ใส่มีเครื่องหมายเดียวกันหรือไม่ (เลขบวกทั้งคู่ หรือ ลบทั้งคู่) หากเหมือนกันให้แสดงผลคูณของเลขทั้งสอง แต่หากต่างกันไม่ต้องทำอะไร และให้หยุดการทำงานเมื่อผู้ใช้ป้อนตัวที่ 2 เป็น 0

ตัวอย่าง (ตัวหนาเอียงแทนค่าจากคีย์บอร์ด)

0 5	-3 -5
10 -3	15
-3 -5	2 -4
15	-3 1
5 0	10 2
	20
	10 0

27

Exercise 6

◆ จงเขียนโปรแกรมที่รับจำนวนเต็ม 2 ตัวเรื่อยๆ แล้วตรวจสอบว่าเลขที่ใส่มีเครื่องหมายเดียวกันหรือไม่ (เลขบวกทั้งคู่ หรือ ลบทั้งคู่) หากเหมือนกันให้แสดงผลคูณของเลขทั้งสอง แต่หากต่างกันไม่ต้องทำอะไร และให้หยุดการทำงานเมื่อผู้ใช้ป้อนตัวที่ 2 เป็น 0

```
while (_____):
    a,b = input().split()
    a = int(a)
    b = int(b)
    if(_____):
        break
    elif (_____):
        _____
    print(a*b)
```

28