

04 คำสั่งควบคุมทิศทางการทำงานของโปรแกรม (Control Statement)

Loop

คำสั่งวนซ้ำในภาษา Python เราจะพูดถึงการควบคุมการทำงานโดยใช้คำสั่ง

- while loop
- for loop

คำสั่งเหล่านี้สามารถควบคุมโปรแกรมให้ทำงานซ้ำๆ ในเงื่อนไขที่กำหนดและเพิ่มความสามารถของการเขียนโปรแกรม

while loop

while loop เป็นคำสั่งวนซ้ำที่ง่ายและพื้นฐานที่สุดในภาษา Python คำสั่ง while loop นั้นใช้ควบคุมโปรแกรมให้ทำงานบางอย่างซ้ำๆ ในขณะที่เงื่อนไขของลูปนั้นยังคงเป็นจริงอยู่

รูปแบบของการใช้งานคำสั่ง while loop ในภาษา Python :

while expression:

statements

In [1]:

```
i = 1 #ประกาศตัวแปร i และกำหนดค่าให้กับตัวแปรเป็น 1

while i <= 10: # สร้างเงื่อนไขสำหรับ while loop เป็น i <= 10 นั้นหมายความว่าโปรแกรมจะทำงานในขณะที่ค่าในตัวแปร i ยังคงน้อยกว่าหรือเท่ากับ 10
    print(i, end = ', ') # และเราแสดงผลค่าของ i
    i = i + 1 # เพิ่มค่าของตัวแปรขึ้นทุกครั้งหลังจากที่แสดงผลเสร็จ ถ้าหากคุณไม่เพิ่มค่าของ i ลูปจะทำงานไม่มีวันหยุดหรือเรียกว่า Infinite loop
```

1, 2, 3, 4, 5, 6, 7, 8, 9, 10,

The break Statement

In [2]:

```
i = 1
while i < 6:
    print(i)
    if i == 3:
        break
    i += 1
```

```
1
2
3
```

In [3]:

```
number = []
MAX_INPUT = 10

print('Enter %d numbers to the list' % MAX_INPUT)  # %d: แสดงค่าจำนวนเต็ม; %f: แสดงค่าจุดทศนิยม; %s:
แสดงผล string.

#รับค่า เราได้กำหนดเงื่อนไขให้กับ loop เป็น i <= MAX_INPUT
#เพื่อให้โปรแกรมรับค่าจากผู้ใช้เป็นตัวเลขจำนวน 10 ตัว หลังจากนั้นเราเก็บตัวเลขที่อ่านมาได้ลงไปใน List number
โดยใช้เมธอด append()
i = 1
while i <= MAX_INPUT:
    print('Number %d: ' % i, end = "")
    n = int(input())
    number.append(n)
    i += 1

# แสดงผลตัวเลขใน list
print('Your numbers in the list')

sum = 0

i = 1
while i <= MAX_INPUT:
    print(number[i - 1], end = ', ')
    sum += number[i - 1]
    i += 1

print('\nSum = %d' % sum)
print('Average = %f' % (sum / MAX_INPUT))
```

Enter 10 numbers to the list

Number 1: 3

Number 2: 4

Number 3: 6

Number 4: 6

Number 5: 8

Number 6: 2

Number 7: 78

Number 8: 85

Number 9: 66

Number 10: 3

Your numbers in the list

3, 4, 6, 6, 8, 2, 78, 85, 66, 3,

Sum = 261

Average = 26.100000

คำสั่ง for loop

In [4]:

```
# loop through string
site = 'marcuscode'
for n in site:
    print(n)
```

m
a
r
c
u
s
c
o
d
e

In [5]:

```
# loop through list
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
```

apple
banana
cherry

In [6]:

```
numbers = [10, 20, 30, 40, 50, 60, 70, 80]
for n in numbers:
    print(n)
```

10
20
30
40
50
60
70
80

คำสั่ง for loop กับฟังก์ชัน range()

ในภาษา Python เรามักจะใช้คำสั่ง for loop กับฟังก์ชัน range() ในการวนอ่านค่าของตัวเลข ฟังก์ชัน range() นั้นเป็น built-in ฟังก์ชันใช้สำหรับสร้างออบเจกต์ของตัวเลข โดยมีพารามิเตอร์ 3 ตัว คือตัวเลขเริ่มต้น ตัวเลขสุดท้าย และค่าที่เปลี่ยนแปลงในลำดับของตัวเลข

In [7]:

```
a = list(range(10))
b = list(range(1, 11))
c = list(range(0, 30, 5))
d = list(range(0, -10, -1))

print(a)
print(b)
print(c)
print(d)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[0, 5, 10, 15, 20, 25]
[0, -1, -2, -3, -4, -5, -6, -7, -8, -9]
```

In []:

In []:

break

คำสั่งให้ออกจากการทำงานใน loop ทันที

In [8]:

```
## Break
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
    if x == "banana":
        break
```

```
apple
banana
```

In [9]:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    if x == "banana":
        break
    print(x)
```

```
apple
```

continue

คำสั่งให้ข้ามไปทำงานรอบถัดไปที่

In [10]:

```
for x in fruits:  
    if x == "banana":  
        continue  
    print(x)
```

apple
cherry

In []: