



**Mahidol University**  
*Wisdom of the Land*



# Lecture 05: Repetition - I

Lecturers:

Aj. Jidapa Kraisangka

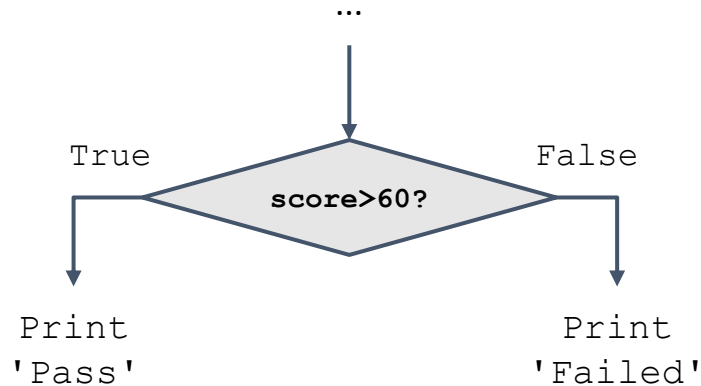
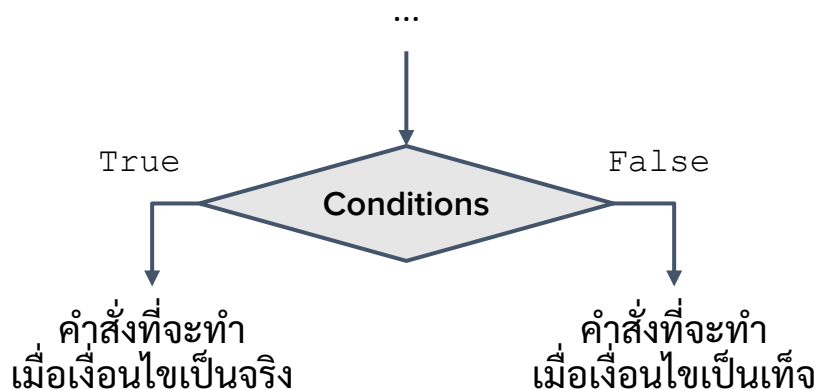
Aj. Akara Supratak

Aj. Tipajin Thaipisutikul



- **RECAP:** relational and logical expressions, `if-else` statement, and `elif` statement
- **Repetition:**
  - `while` statement
  - `for` statement
  - Loop control statements: `continue` and `break`

- **Selection:** รูปแบบที่การเลือกทำชุดคำสั่งบางชุด โดยพิจารณาจากค่าที่เงื่อนไข (Conditions) เหล่านั้นเป็นจริง(True) หรือเท็จ (False)





**Relational expression** คือ expressions ที่ใช้ตัวดำเนินการเชิงเปรียบเทียบ (Comparison operators) โดยจะเปรียบเทียบค่าสองค่าและประมวลผลลัพธ์เป็นค่า Boolean

Table 2-1: Comparison Operators

Operator	Meaning
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

เครื่องหมาย == ใช้เปรียบเทียบว่าค่าสองค่าเท่ากันหรือไม่ ซึ่งไม่ใช่ Assignment operator (=) ที่นำค่ามาใส่ตัวแปร

ใช้เปรียบเทียบเฉพาะ int และ float เท่านั้น



**Logical expression** คือ expressions ที่ใช้ตัวดำเนินการด้านตรรกศาสตร์ (Boolean operators ได้แก่ **and**, **or** และ **not**) โดยจะเปรียบเทียบค่า Boolean และประมวลผลลัพธ์เป็นค่า **Boolean (True or False)**

Table 2-2: The and Operator's Truth Table

Expression	Evaluates to ...
True and True	True
True and False	False
False and True	False
False and False	False

Table 2-3: The or Operator's Truth Table

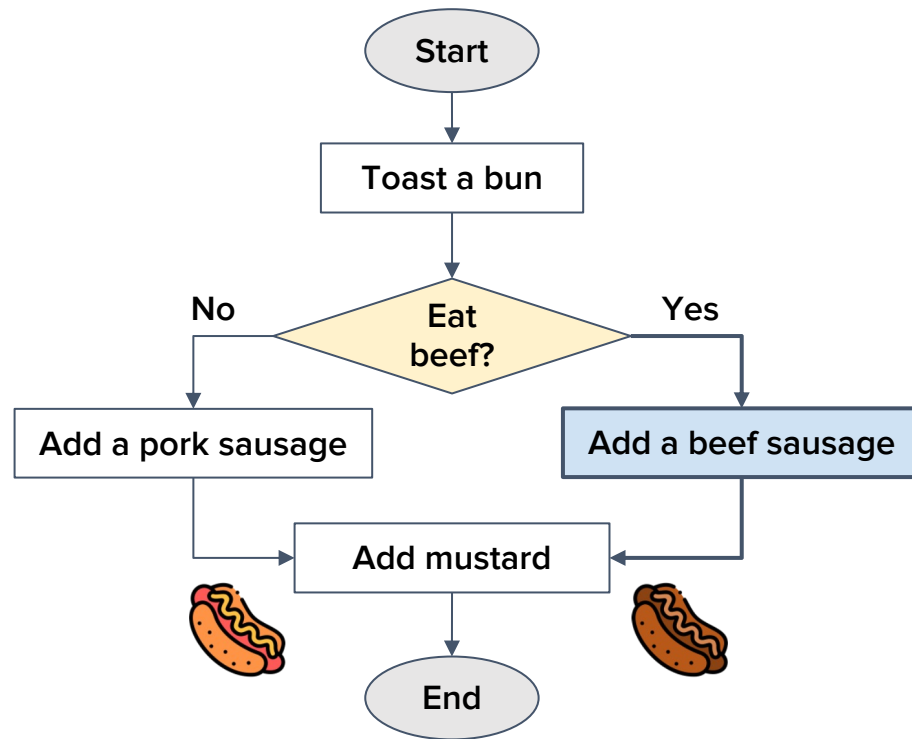
Expression	Evaluates to ...
True or True	True
True or False	True
False or True	True
False or False	False

Table 2-4: The not Operator's Truth Table

Expression	Evaluates to ...
not True	False
not False	True

Tables from <https://automatetheboringstuff.com/2e/chapter2/>

- Relational Expression และ Logical Expression เป็นพื้นฐานสำหรับการเขียนเงื่อนไข (Conditions)
- Conditions จะถูกประมวลผลเป็นค่า Boolean เสมอ
- ผลของ Conditions (ได้แก่ True, False) จะเป็นตัวกำหนดชุดคำสั่งที่จะกระทำ





- **if** เป็นคำสั่งที่ใช้ควบคุมการทำงานของโปรแกรมให้ทำงานตามเงื่อนไขที่กำหนดเมื่อเงื่อนไขนั้น ๆ เป็นจริง
- **else** เป็นคำสั่งที่ใช้ต่อเนื่องจากคำสั่ง **if** เพื่อควบคุมการทำงานของโปรแกรมให้ทำงานตามเงื่อนไขที่กำหนดเมื่อเงื่อนไขนั้น ๆ เป็นเท็จ

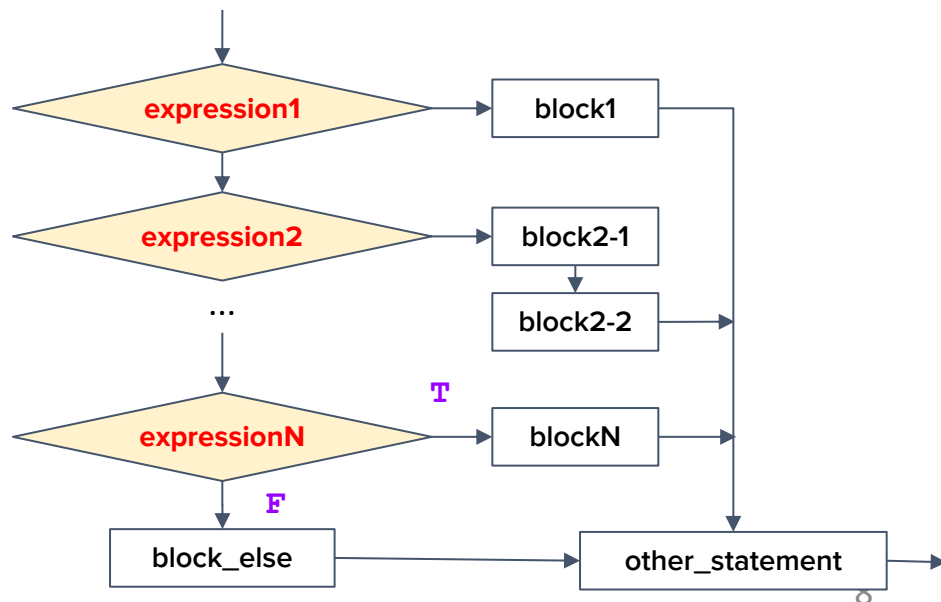
```
normal_statementX  
  
if expression:  
    statement1  
    ...  
    statementN  
  
normal_statementY
```

```
normal_statementX  
  
if expression:  
    statement_when_T  
else :  
    statement_when_F  
  
normal_statementY
```



- **elif** (ย่อจาก else if ) เป็นคำสั่งที่ใช้ต่อเนื่องจากคำสั่ง **if** เพื่อควบคุมการทำงานของโปรแกรมให้ทำงานตามหลาย ๆ เงื่อนไขอย่างต่อเนื่อง
- ใช้ร่วมกับ **if** และ **else**

```
if expression1:  
    statement_block1  
elif expression2:  
    statement_block2-1  
    statement_block2-2  
...  
elif expressionN:  
    statement_blockN  
else:  
    statement_block_else  
other_statement
```



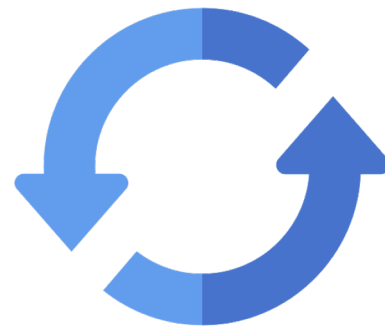




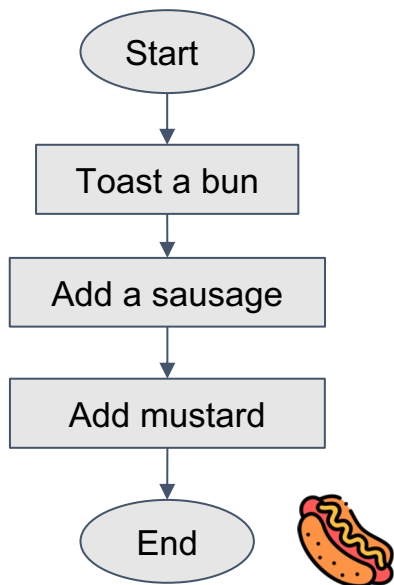
- **Flow controls (in programming)** เป็นลักษณะการเขียนโปรแกรม ที่มีลำดับการประมวลผลแตกต่างกันไป
- รูปแบบของ Flow controls ได้แก่
  - **Sequential:** รูปแบบปกติที่เป็นลำดับขั้นตอนเรียงกัน
  - **Selection:** รูปแบบที่การเลือกทำชุดคำสั่งบางชุดตามเงื่อนไข
  - **Repetition:** รูปแบบปกติที่ทำซ้ำในบางชุดคำสั่ง
  - **Invocation:** รูปแบบการเรียกใช้ชุดคำสั่งอื่น ๆ หรือฟังก์ชัน



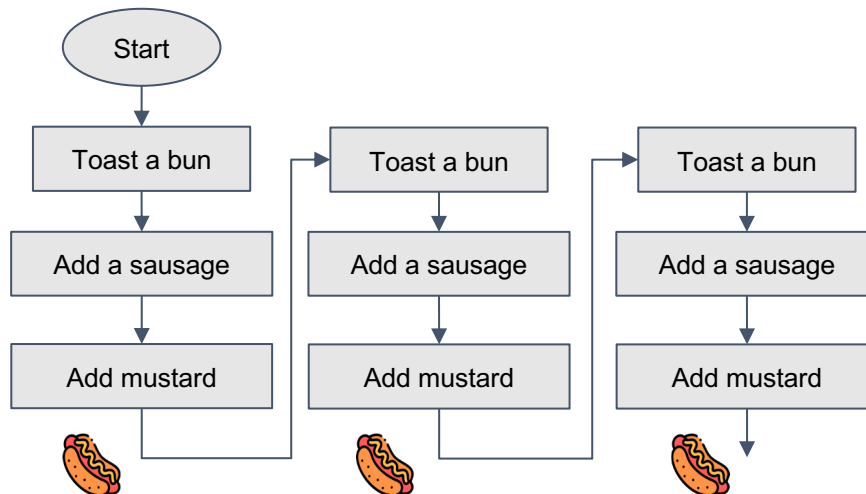
- **Repetition** หรือการเขียนโปรแกรมแบบวนซ้ำ
- เป็นการนำชุดคำสั่งมาทำงานซ้ำหลายๆ รอบ
- จะทำงานที่รอบขึ้นอยู่กับการเงื่อนไขที่ผู้เขียนโปรแกรมกำหนดไว้
- ชุดคำสั่งที่มีการวนใช้ซ้ำๆ เรียกว่า **loop**



ขั้นตอนการทำ hotdog 1 อัน

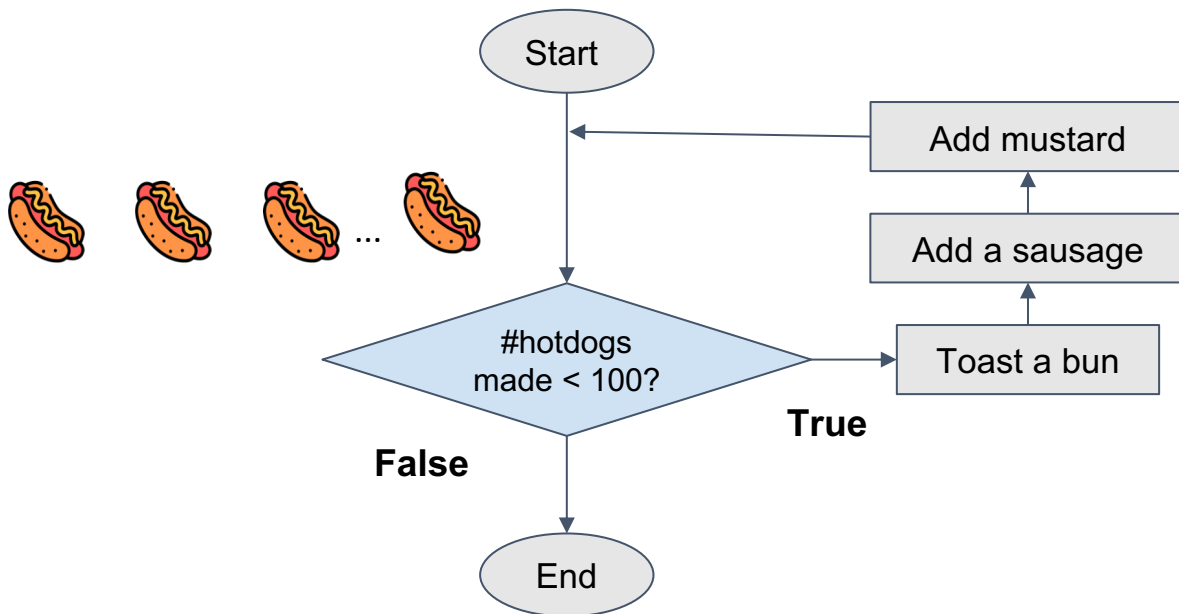


ถ้าต้องการทำ hotdog 100 หรือ 1000 อัน  
เราจะต้องเขียนชุดคำสั่งนี้ 100 หรือ 1000 ครั้ง?



•  
•  
•

การใช้ loop ช่วยให้การเขียน code มีความ flexible และสั้นลง



Loop จะวนไปเรื่อยๆ ถ้า  
condition ที่กำหนดเป็นจริง



คำสั่งที่ใช้ในการวน loop

- **while** statement
- **for** statement



- ควบคุมโปรแกรมให้ทำงานบางอย่างซ้ำๆ ในขณะที่เงื่อนไข (expression) ของ loop นั้นยังคงเป็นจริง (ไม่เท่ากับ 0) อยู่
- มักถูกใช้เมื่อจำนวนรอบในการวน loop ไม่ทราบล่วงหน้า

```
while expression:  
    statement(s)
```

ขั้นตอนการทำงานของ while loop:

1. Test the expression
2. If the expression is TRUE
  - a. Execute the statement(s)
  - b. Go to step 1

Else exit the while loop



## 1. Conditions

เงื่อนไขที่ใช้การวน loop ถ้าเป็นจริง loop จะทำงานซ้ำ โดยใช้ (Relational/Logical) Expression

## 2. Initialization:

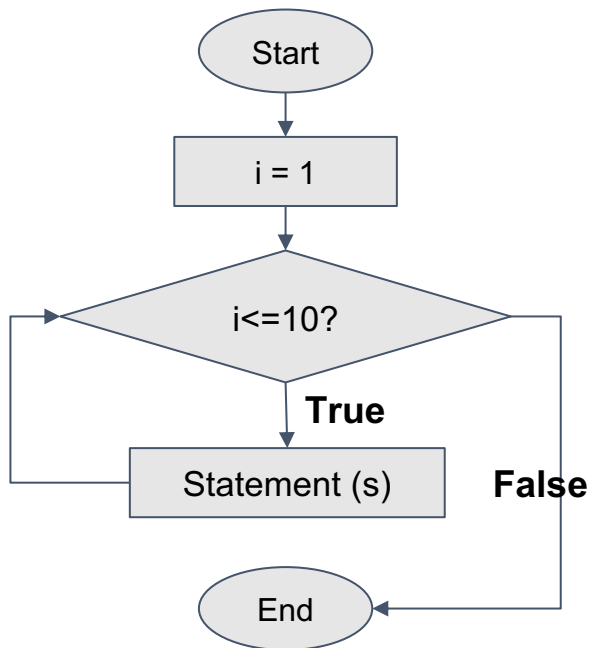
การกำหนดค่าเริ่มต้นของตัวแปรที่จะเป็นเงื่อนไขในการ loop

## 3. Alteration:

- การเปลี่ยนแปลงค่าของตัวแปรที่ใช้เป็นเงื่อนไขในการ loop
- ผลลัพธ์ของ expression จะถูกเปลี่ยนเป็น **False** เพื่อหยุดการทำงานของ loop



Print เลข 1 ถึง 10



Initialization

Conditions จะต้องเป็น **True**  
เพื่อให้ loop ทำงานต่อไป

```
i = 1
while i <= 10:
    print(i)
    i = i + 1
```

Alteration

Output:

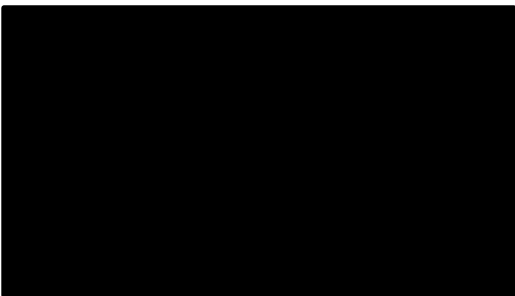
1  
2  
3  
4  
5  
...  
10





```
1. i=1
2. while i<=5:
3.     print(i)
4.     i=i+1
5. print( 'End' )
```

Output:



รอบที่	ค่า i ที่เงื่อนไข (Line#2)	i<=5 (Line#2)	ค่า i ที่ Line#4



```
1. i=1
2. while i<=5:
3.     print(i)
4.     i=i+1
5. print( 'End' )
```

Output:

1

รอบที่	ค่า i ที่เงื่อนไข (Line#2)	i<=5 (Line#2)	ค่า i ที่ Line#4
1	1	True	2



```
1. i=1
2. while i<=5:
3.     print(i)
4.     i=i+1
5. print( 'End' )
```

Output:

```
1
2
```

รอบที่	ค่า i ที่เงื่อนไข (Line#2)	i<=5 (Line#2)	ค่า i ที่ Line#4
1	1	True	2
2	2	True	3



```
1. i=1
2. while i<=5:
3.     print(i)
4.     i=i+1
5. print( 'End' )
```

Output:

```
1
2
3
```

รอบที่	ค่า i ที่เงื่อนไข (Line#2)	i<=5 (Line#2)	ค่า i ที่ Line#4
1	1	True	2
2	2	True	3
3	3	True	4



```
1. i=1
2. while i<=5:
3.     print(i)
4.     i=i+1
5. print('End')
```

Output:

```
1
2
3
4
```

รอบที่	ค่า i ที่เงื่อนไข (Line#2)	i<=5 (Line#2)	ค่า i ที่ Line#4
1	1	True	2
2	2	True	3
3	3	True	4
4	4	True	5



```
1. i=1
2. while i<=5:
3.     print(i)
4.     i=i+1
5. print('End')
```

Output:

```
1
2
3
4
5
```

รอบที่	ค่า i ที่เงื่อนไข (Line#2)	i<=5 (Line#2)	ค่า i ที่ Line#4
1	1	True	2
2	2	True	3
3	3	True	4
4	4	True	5
5	5	True	6



```
1. i=1
2. while i<=5:
3.     print(i)
4.     i=i+1
5. print( 'End' )
```

Output:

```
1
2
3
4
5
End
```

รอบที่	ค่า i ที่เงื่อนไข (Line#2)	i<=5 (Line#2)	ค่า i ที่ Line#4
1	1	True	2
2	2	True	3
3	3	True	4
4	4	True	5
5	5	True	6
6	6	False	-

เมื่อมีการตรวจสอบเงื่อนไขในรอบที่ 6 เงื่อนไข  $i \leq 5$  มีค่าเป็น **False** ทำให้การ while loop หยุดการทำงาน แล้วทำคำสั่งต่อไปนอก while block ใน Line#6



จงระบุจำนวนรอบ (Loop) และผลลัพธ์ในแต่ละรอบที่โปรแกรมทำงาน จากส่วนของโปรแกรมต่อไปนี้

ส่วนของโปรแกรม	จำนวนรอบ (Loop)	ผลลัพธ์ในแต่ละรอบ
<pre>i=1 while i&lt;=3:     print(1)     i=i+1</pre>	3	Loop#1: 1 (i=1) Loop#2: 1 (i=2) Loop#3: 1 (i=3)
<pre>i=1 while i&lt;=10:     print(i)     i=i+1</pre>		
<pre>i=0 while i&lt;10:     print(i)     i=i+1</pre>		





จงระบุจำนวนรอบ (Loop) และผลลัพธ์ในแต่ละรอบที่โปรแกรมทำงาน จากส่วนของโปรแกรมต่อไปนี้

ส่วนของโปรแกรม	จำนวนรอบ (Loop)	ผลลัพธ์ในแต่ละรอบ
<pre>i=0 x=0 while i&lt;20:     print(x, end=' ')     x=x+1     i=i+5</pre>		
<pre>i=8 x=5 while i&gt;2:     print(x, end=' ')     x=x+1     i=i-2</pre>		



จงเขียนเงื่อนไขใน while ที่ทำให้โปรแกรมแสดงผลลัพธ์ตามต้องการดังต่อไปนี้

ส่วนของโปรแกรม	ผลลัพธ์ที่ต้องการ	เงื่อนไข
<pre>i=1 while [ ]:     print(1, end=' ')     i=i+1</pre>	1 1 1 1 1	$i \leq 5$
<pre>x=-2 while [ ]:     print(x, end=' ')     x=x+1</pre>	-2 -1 0 1 2 3 4	
<pre>x=0 while [ ]:     if x%10==0:         print(x, end=' ')     x=x+1</pre>	0 10 20 30 40 50	



จงเขียน while ในการแสดงผลลัพท์ตัวเลขต่อไปนี้

- 100 99 98 ... 1
- 9 7 5 3 1
- 1 2 4 8 16 32 64
- 1 2 4 7 11 16 22 29 37
- 10 x 9 y 8 x 7 y 6 x 5 y 4 x 3 y 2 x 1 y



Q1: 100 99 98 ... 1

Q2: 1 3 5 7 9



Q3: 1 2 4 8 16 32 64

A large, empty rectangular box with a thin black border, intended for the student to write their solution for Question 3.

Q4: 1 2 4 7 11 16 22 29 37

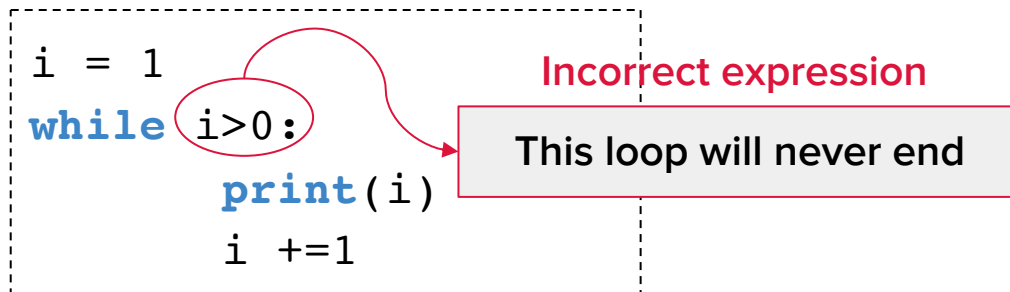
A large, empty rectangular box with a thin black border, intended for the student to write their solution for Question 4.



Q5: 10 x 9 y 8 x 7 y 6 x 5 y 4 x 3 y 2 x 1 y



- การใช้ **while** statement ควรใช้ด้วยความระมัดระวัง เนื่องจากสามารถทำให้เกิด loop ที่ไม่สิ้นสุดได้
- เกิดจากการใช้ expression ที่ผิด หรือการลืมเปลี่ยนแปลงค่าตัวแปรที่ใช้เป็นเงื่อนไขในการ loop → condition ไม่ถูกเปลี่ยนให้เป็น False
- ในกรณีที่โปรแกรมวน loop ไม่หยุด กด **ctrl +c**





Mahidol University  
*Wisdom of the Land*



# for statement





- เป็นคำสั่งวนซ้ำที่ใช้ควบคุมการทำงานซ้ำๆ มักใช้สำหรับการวนอ่านค่าใน **iterable object** เช่น list, string, tuple
- มักใช้เมื่อ**ทราบจำนวนรอบ**ในการวน loop แน่แน่นอน

```
for var in <sequence>:  
    statement(s)
```

## ขั้นตอนการทำงาน for loop:

1. Item แรกใน sequence ถูก assign เก็บใน **var**
  2. ชุดคำสั่งใน for loop ถูก execute
  3. If item ใน sequence ยังมีอยู่
    - a. item ถัดมาใน sequence จะถูก assign ให้ **var**
    - b. กลับไป step 2
- Else ออกจาก for loop



- **range()** เป็น built-in function ที่มักใช้ร่วมกับ for loop
- ใช้ในการสร้าง sequence ของตัวเลข ภายใน range ที่กำหนด
- Default: เริ่มต้นที่ 0, เพิ่มขึ้นทีละ 1, และจบที่ตัวเลขสุดท้ายที่กำหนด

## Syntax:

**range(stop)**

**range(start, stop)**

**range(start, stop, step)**

มี parameter 3 ตัว เป็น type integer ได้แก่

- **start:** ตัวเลขเริ่มต้น
- **stop:** ตัวเลขสุดท้าย (ไม่รวม)
- **step:** ค่าที่เปลี่ยนแปลง

ไม่ support float numbers



ไม่รวมตัวสุดท้าย

```
for x in range(10):  
    print(x, end=' ')
```

Output:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

```
for x in range(5, 12):  
    print(x, end=' ')
```

5	6	7	8	9	10	11
---	---	---	---	---	----	----



```
for x in range(1, 9, 1):  
    print(x, end=' ')
```

1 2 3 4 5 6 7 8

increment/เพิ่ม

```
for x in range(1, 15, 3):  
    print(x, end=' ')
```

1 4 7 10 13

```
for x in range(8, 0, -1):  
    print(x, end=' ')
```

8 7 6 5 4 3 2 1

decrement/ลด



ทำ output ของ code ต่อไปนี้:

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

```
for x in range(1, 11):  
    print(x, end=' ')
```

```
for x in range(-5, 10, 2):  
    print(x, end=' ')
```

Output



ทำ output ของ code ต่อไปนี้:

```
for x in range(-5):  
    print(x, end=' ')
```

```
for x in range(5, -4, -1):  
    print(x, end=' ')
```

Output



จงระบุจำนวนรอบ (Loop) และผลลัพธ์ในแต่ละรอบที่โปรแกรมทำงาน จากส่วนของโปรแกรมต่อไปนี้

ส่วนของโปรแกรม	จำนวนรอบ (Loop)	ผลลัพธ์ในแต่ละรอบ
<pre>for x in range(3):     print(x, end=' ')</pre>	3	Loop#1: 1 (x=1) Loop#2: 1 (x=2) Loop#3: 1 (x=3)
<pre>q=1; for x in range(0, 6):     print(q, end=' ')     q=q*2;</pre>		
<pre>for x in range(10):     if x%2==1:         print(x, end=' ')     else:         print('-', end=' ')</pre>		



เขียน **for** loop เพื่อ print output ดังต่อไปนี้

- 100 99 98 ... 1
- 9 7 5 3 1
- 1 2 4 8 16 32
- 20 16 12 8 4 0 -4 -8 -12 -16 -20
- a b a b a b a b





Q1: 100 99 98 ... 1

A large, empty rectangular box with a thin black border, intended for the user to write their answer to Question 1.

Q2: 9 7 5 3 1

A large, empty rectangular box with a thin black border, intended for the user to write their answer to Question 2.



Q3: 1 2 4 8 16 32

Q4: 20 16 12 8 0 4 -4 -8 -12 -16 -20



Q5: a b a b a b a b



Mahidol University  
*Wisdom of the Land*



# ตัวอย่างการใช้ Loop



เขียนโปรแกรมรับ input ตัวเลข 10 จำนวน และ print ผลรวมของตัวเลข

**Output:**

Enter a number : 10

10

Enter a number : 5

15

Enter a number : 3

18

Enter a number : 12

30

...



เขียนโปรแกรมรับ input ตัวเลข 10 จำนวน และ print ผลรวมของตัวเลข

**Output:**

Enter a number : 10

10

Enter a number : 5

15

Enter a number : 3

18

Enter a number : 12

30

...



ไม่ทราบจำนวนการวนซ้ำล่วงหน้า

จงเขียนโปรแกรมเพื่อรับ input ตัวเลข จนกว่าตัวเลขที่รับจะเป็นจำนวนลบ จากนั้น  
แสดงผลรวมของ input ที่เป็นจำนวนบวกทั้งหมด

**Output:**

Enter a number : 10

10.0

Enter a number : 5

15.0

Enter a number : 3

18.0

Enter a number : 0

18.0

Enter a number : -1

**Output:**

Enter a number : -5

**Output:**

Enter a number : 5

5.0

Enter a number : 5

10.0

Enter a number : -3



**Output:**

Enter a number : 10

10.0

Enter a number : 5

15.0

Enter a number : 3

18.0

Enter a number : 0

18.0

Enter a number : -1





- **for:** มักใช้เมื่อทราบจำนวนรอบในการ loop อย่างแน่นอน
  - ใช้ร่วมกับ function `range ( )` และ iterable objects เช่น list, string, tuple
- **while:** มักใช้เมื่อจำนวนรอบในการ loop ไม่ทราบล่วงหน้า
  - เช่น โปรแกรมที่จำนวน loop ขึ้นกับ user input
  - แต่สามารถใช้กับกรณีทราบจำนวนรอบในการ loop ล่วงหน้าแล้วเช่นกัน ขึ้นอยู่กับ style การ code ของ programmer



**Mahidol University**  
*Wisdom of the Land*



KEEP  
CALM  
AND  
CODE  
ON

[keep-calm.net](https://keep-calm.net)

Thanks to: <https://keep-calm.net/keep-calm-and-code-on.html>

**Happy Coding :)  
More Loop next week**