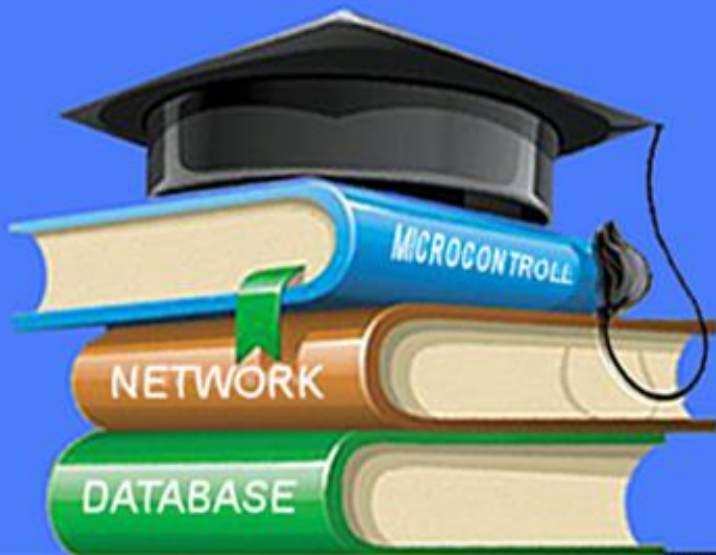


Computers & Programming 

Chapter 6: Python Basic Loop





Outline

Basic Loop

Loop Statement in Python

- while Statement
- for Statement

Python
Basic Loop



Basic Loop

พิจารณาโปรแกรมต่อไปนี้ ให้เขียนโปรแกรมแสดงคำว่า “Hello, Kitty”
2,000 ครั้ง

```
print("Hello, Kitty")  
print("Hello, Kitty")  
print("Hello, Kitty")  
print("Hello, Kitty")  
print("Hello, Kitty")  
print("Hello, Kitty")  
print("Hello, Kitty")  
print("Hello, Kitty")
```

Hello, Kitty 8 times

เราจะทราบได้อย่างไรว่าแสดงถึง 2000 ครั้งแล้ว?

Basic Loop

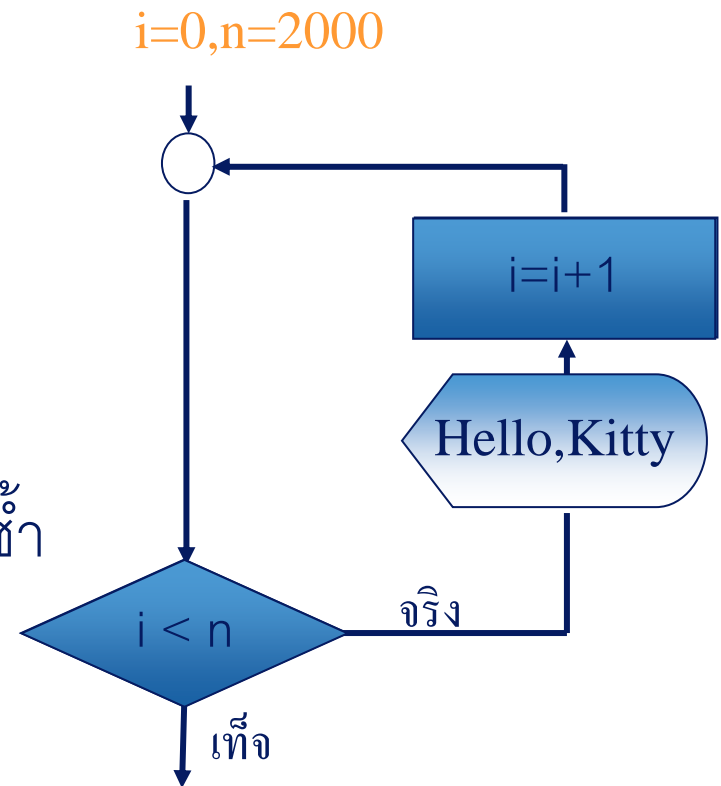
ตัวอย่าง flowchart ของปัญหาข้างต้น

◆ การทำซ้ำ (Looping or iteration)

การเขียนโปรแกรมที่ขั้นตอนการทำงานบาง
ขั้นตอนได้รับการประมวลผลมากกว่า 1 ครั้ง

ทั้งนี้ขึ้นอยู่กับเงื่อนไขบางประการ ซึ่งจะ

พิจารณาเงื่อนไขและตัดสินใจในการทำงานซ้ำ





Basic Loop: Looping Type

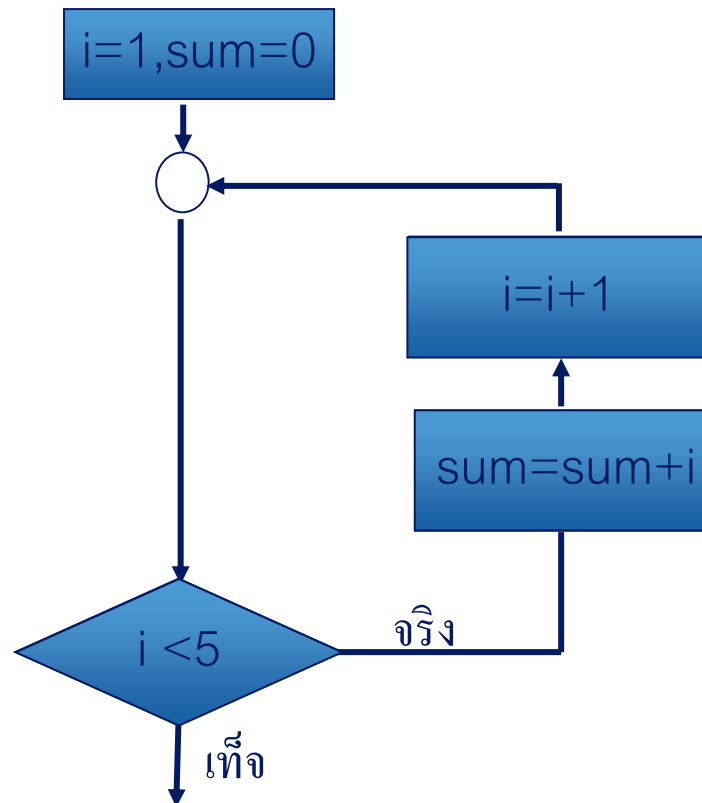
คำสั่งวนซ้ำแบบใช้ตัวนับ (Counting Loops)

คำสั่งวนซ้ำแบบใช้ตัวหยุด (Sentinel Loops)

Python
Basic Loop

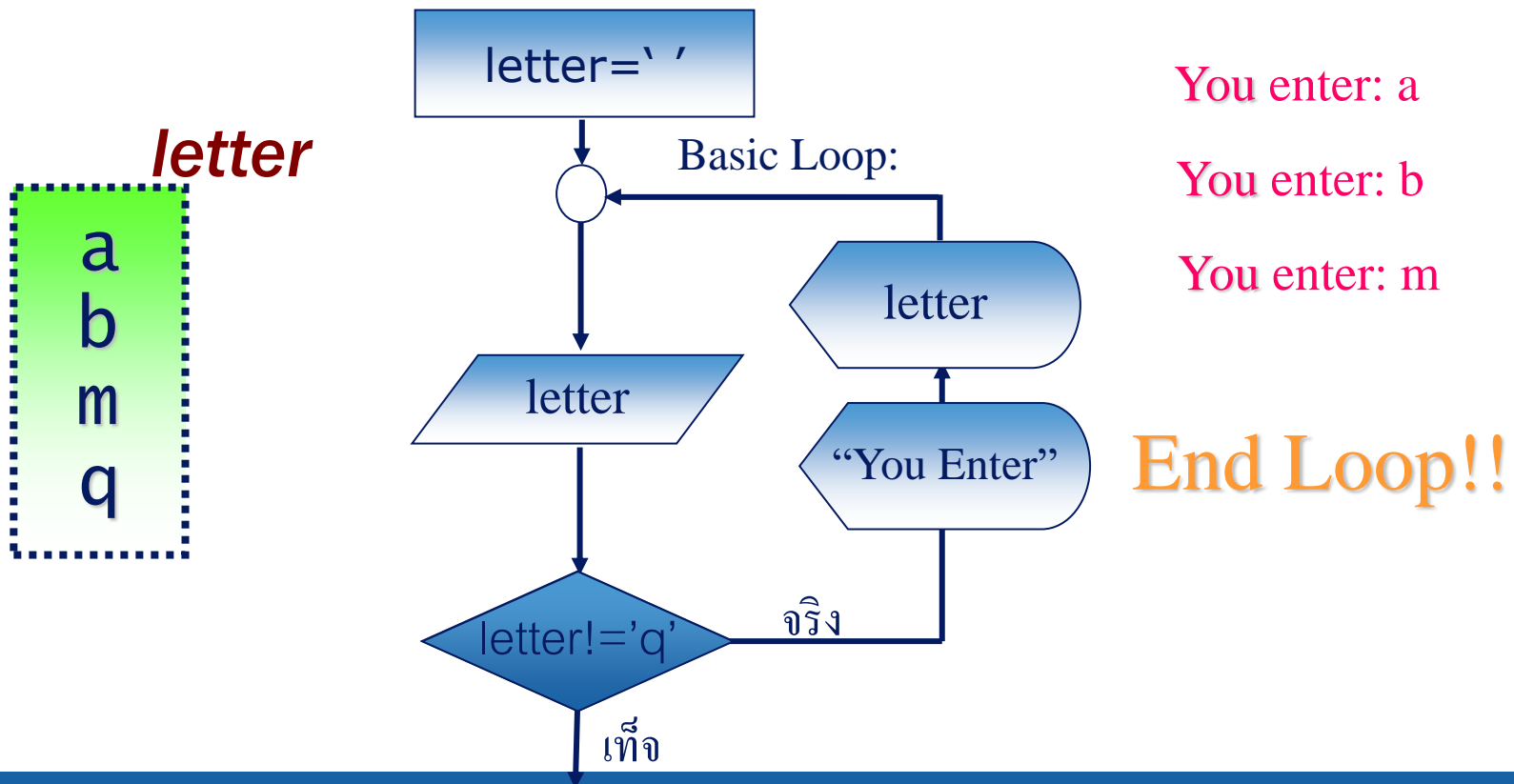
Basic Loop: Counting Loops

- ◆ ใช้ตัวแปรนับว่าชุดคำสั่งที่ต้องการทำซ้ำ ทำงานไปแล้วกี่รอบ
- ◆ การวนซ้ำจะหยุดเมื่อโปรแกรมทำงานครบตามจำนวนรอบที่ต้องการ



Basic Loop: Sentinel Loops

- ◆ ใช้การกำหนดเงื่อนไขของตัวแปร
- ◆ จะหยุดการทำงานเมื่อพบค่าบางอย่าง เช่น เมื่อค่าของตัวแปร letter เท่ากับ q



Looping or Iteration in Python

```
graph BT; while((while)) --> Iteration[Iteration]; for((for)) --> Iteration;
```

Iteration

while

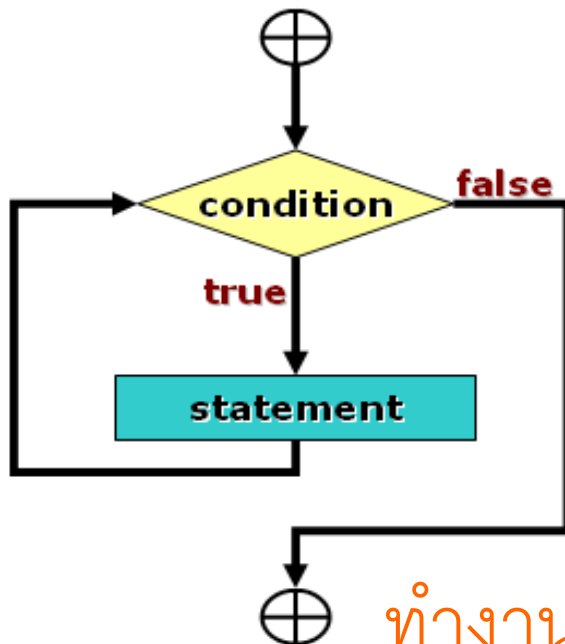
for

While statement

ไวยากรณ์ (Syntax):

while (เงื่อนไข):

Tab → คำสั่ง



while (เงื่อนไข):

คำสั่ง 1

คำสั่ง 2

...

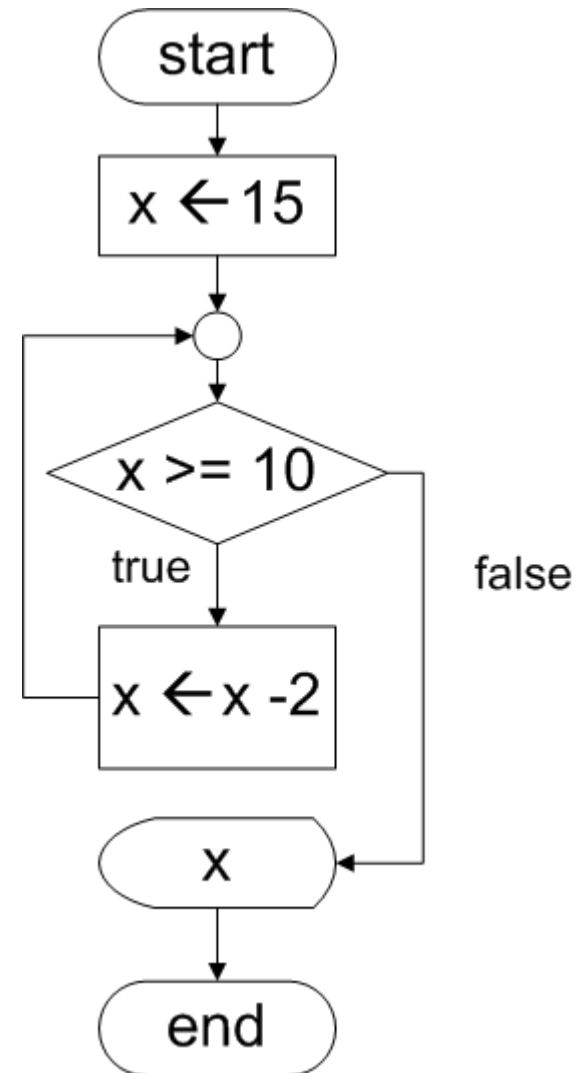
ทำงาน คำสั่ง ตราบเท่าที่เงื่อนไขเป็นจริง

While statement

- ตัวอย่าง 1

```
x = 15
while (x >= 10) :
    x = x - 2
print(x)
```

9



While statement

ตัวอย่าง 2

```
a = 0
```

```
while (a < 7):  
    print("Hello World")  
    a = a + 1;
```

a: 0

a: 1

a: 2

a: 3

a: 4

a: 5

a: 6

a: 7

Hello World

Hello World

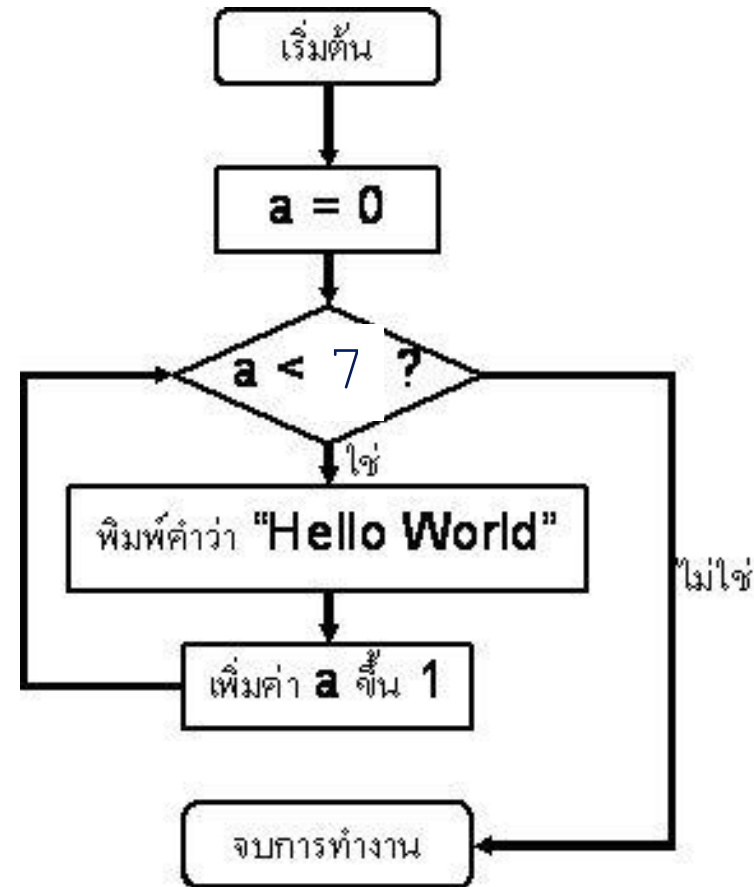
Hello World

Hello World

Hello World

Hello World

Hello World





While statement (Example)

```
i = 0
```

```
while (i < 5):  
    print("Kitty-",i)  
    i+=1
```

0
1
2
3
4
5

Kitty-0
Kitty-1
Kitty-2
Kitty-3
Kitty-4

While statement (Example)

```
i = 0  
n = input()
```

n=20

```
while (i < int(n)):  
    print("Kitty-",i)  
    i+=1
```

0
1
2
...
19
20

Kitty-0
Kitty-1
Kitty-2
...
Kitty-19

End Loop!!

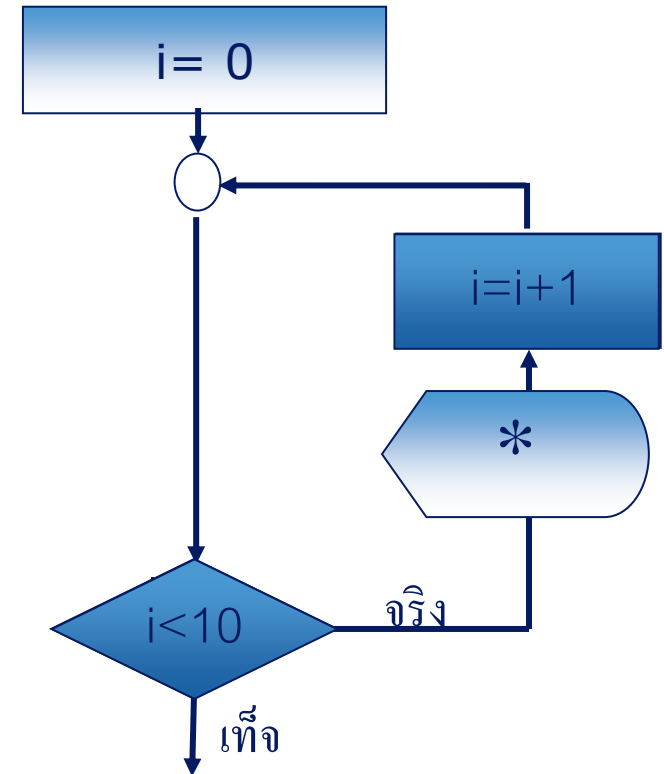
While statement (Example)

คำนวณหาผลรวมของเลขจำนวนเต็มบวกที่ไม่เกิน 35 ที่หาร 3 ลงตัว?

```
N = 1      #ตัวนับ
SUM = 0    #เก็บผลบวก
while ( N <= 35 ) :
    if (N%3 == 0):
        SUM = SUM+N;
    N+=1
print("SUM =", SUM)
```

While statement (Example)

```
i = 0
while ( i < 10 ):
    print("*")
    i += 1
```



ถ้าต้องการให้แสดง ***N (*)stars?***



Looping or Iteration in Python

Iteration

while

for

```
i = start  
while (i < end):  
    ...  
    i = i + 1
```

```
for i in range(start,end):
```


For statement

ไวยากรณ์ (Syntax):

```
for ตัวแปรสำหรับนับรอบ in range(ค่าเริ่มต้น,ค่าสิ้นสุด, คำสั่งปรับค่า):  
    คำสั่ง  
    คำสั่ง
```

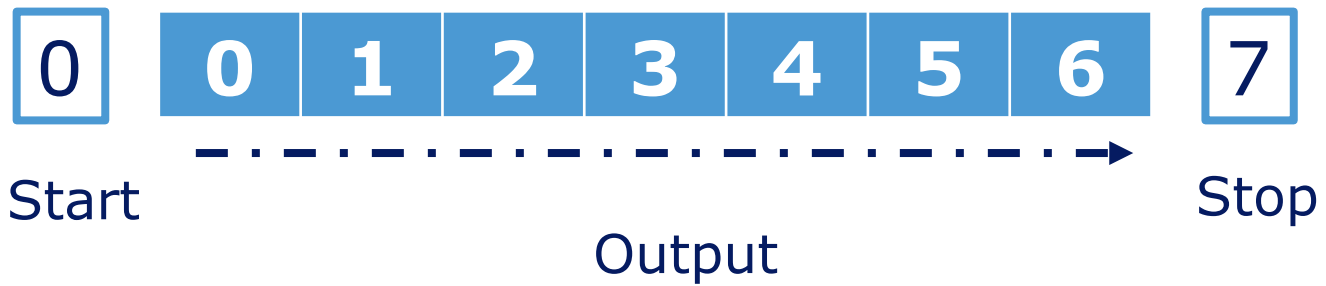
- **range()** : ฟังก์ชันที่ทำให้ for loop ทำการนับค่าให้อยู่ในช่วงที่ของ **ค่าเริ่มต้น** และ **ค่าสิ้นสุด**
- **ตัวแปรสำหรับนับรอบ** เป็นตัวแปรนับรอบ
- **ค่าเริ่มต้น** กำหนดค่าเริ่มต้นให้กับตัวนับรอบ (หากไม่กำหนดจะมีค่าเป็น 0)
- **ค่าสิ้นสุด** กำหนดค่าสิ้นสุดให้กับตัวนับรอบโดยกำหนดให้มีค่าเป็นบวกหรือลบ
- **คำสั่งปรับค่า** กำหนดค่าเพิ่มค่าและลดค่าตัวนับรอบ (หากไม่กำหนดจะมีค่าเป็น +1)
- คำสั่ง เป็นคำสั่งที่ให้ทำในแต่ละรอบซึ่งเป็นคำสั่งเดี่ยว หรือคำสั่งประกอบ ภายใต้บล็อกถัดไปจากคำสั่ง for loop



For statement

	Syntax	Output
Single Argument	<code>range(j)</code> Ex: <code>range(10)</code>	$0, 1, 2, 3, 4, \dots, j-1$ $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$
Double Argument	<code>range(i, j)</code> Ex: <code>range(1, 10)</code>	$i, i+1, i+2, i+3, \dots, j-1$ $1, 2, 3, 4, 5, 6, 7, 8, 9$
Triple Argument	<code>range(i, j, k)</code> Ex: <code>range(1, 10, 2)</code>	$i, i+k, i+2k, i+3k, \dots, j-1$ $1, 3, 5, 7, 9$

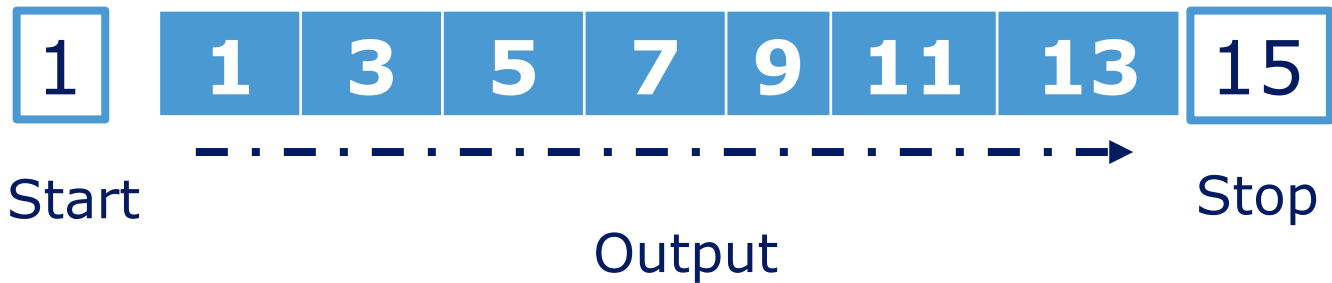

```
for i in range(7):
```



For statement

◆ ไวยากรณ์ (Syntax):

```
for i in range(1, 15, 2):
```





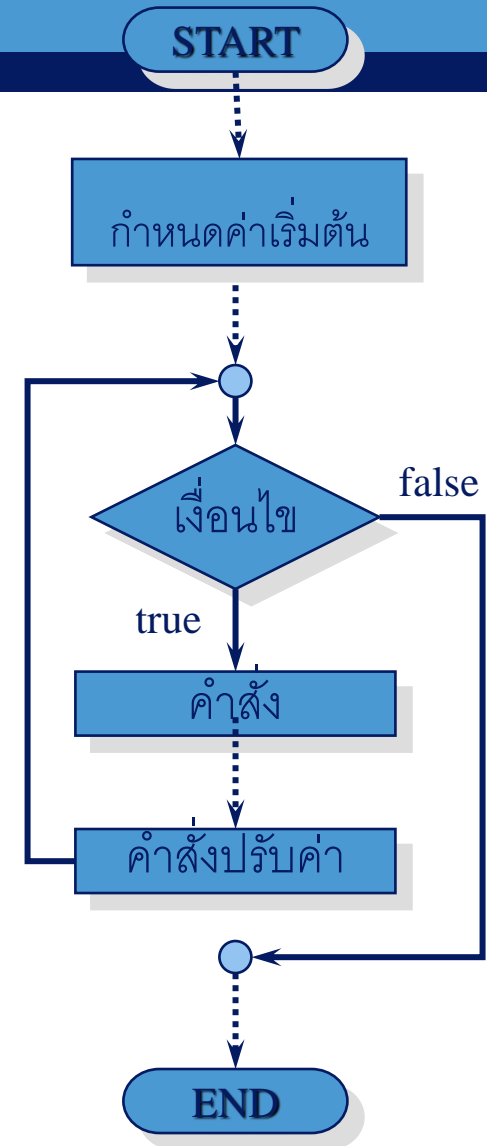
QuickCheck

For statement	Result
for i in range(5)	
for i in range(1,5)	
for i in range(15,1,-1)	
for i in range(15,0,-2)	
for i in range(1,5,10)	

For statement

- การทำงาน

1. กำหนดค่าเริ่มต้นให้กับตัวแปรนับรอบ
2. ทดสอบเงื่อนไขของตัวนับรอบ ถ้าเงื่อนไขเท็จให้ทำข้อ 5 ถ้าเงื่อนไขจริงให้ทำข้อ 3
3. ทำคำสั่งในลูป
4. เพิ่มค่า (หรือลดค่า) ให้กับตัวนับ แล้วไปทำในข้อ 2
5. จบการวนซ้ำ



For statement

ตัวอย่าง

```
a=0
```

```
while (a < 10):
```

```
    print("Hello World")
```

```
    a = a + 1
```

คำสั่งเริ่มต้น เงื่อนไข คำสั่งปรับค่า*

```
for a in range(0,10,1):  
    print("Hello World")
```

เริ่มต้นทำงาน กำหนดให้ a=0

ตรวจสอบเท่าที่ a มีค่าน้อยกว่า 10

จะพิมพ์คำว่า Hello World และ เพิ่มค่า a ขึ้นทีละ 1



For statement

- ตัวอย่าง : การหาผลรวมตัวเลขตั้งแต่ 1 ถึง 10

```
sum=0
for a in range(10):
    sum=sum+a

print(sum)
```

```
sum=0
for a in range(9,0,-1):
    sum=sum+a

print(sum)
```

```
a=1
sum=0
while(a < 10):
    sum=sum+a
    a+=1

print(sum)
```


For statement

ตัวอย่าง : การการคำนวณแฟคทอเรียลของเลข N

$$\blacklozenge N! = N * (N-1) * (N-2) * (N-3) * \dots * 1$$

$$\text{Example } 5! = 5*4*3*2*1$$

```
N = input()
N = int(N)
fact=1
for a in range(N,0,-1)
    fact=fact*a
print(fact)
```

N =5

5
4
3
2
1
0

5*1=5
4*5=20
3*20=60
2*60=120
1*120=120

End Loop!!



for statement

จงแสดงเลขตั้งแต่ 0 ถึง 100 เฉพาะเลขที่หารด้วย **N** ลงตัว
โดยรับค่า **N** จากทางคีย์บอร์ด ** ไม่ให้ใช้ **if**

```
N = input()
N = int(N)
for a in range(0, 100+1, N):
    print(a)
```

QuickCheck

◆ จงบอกผลลัพธ์ของชุดคำสั่งต่อไปนี้

a. `j= 1`

```
while(j <= 9):  
    print(j)  
    j += 2
```

b. `j= 1`

```
while(j <= 9):  
    j += 2  
    print(j)
```



QuickCheck

◆ จงบอกผลลัพธ์ของชุดคำสั่งต่อไปนี้

c. `for j in range(1,10,+2):`
 `print(j)`

d. `for j in range(9,0,-2):`
 `print(j)`

Thank You !

