



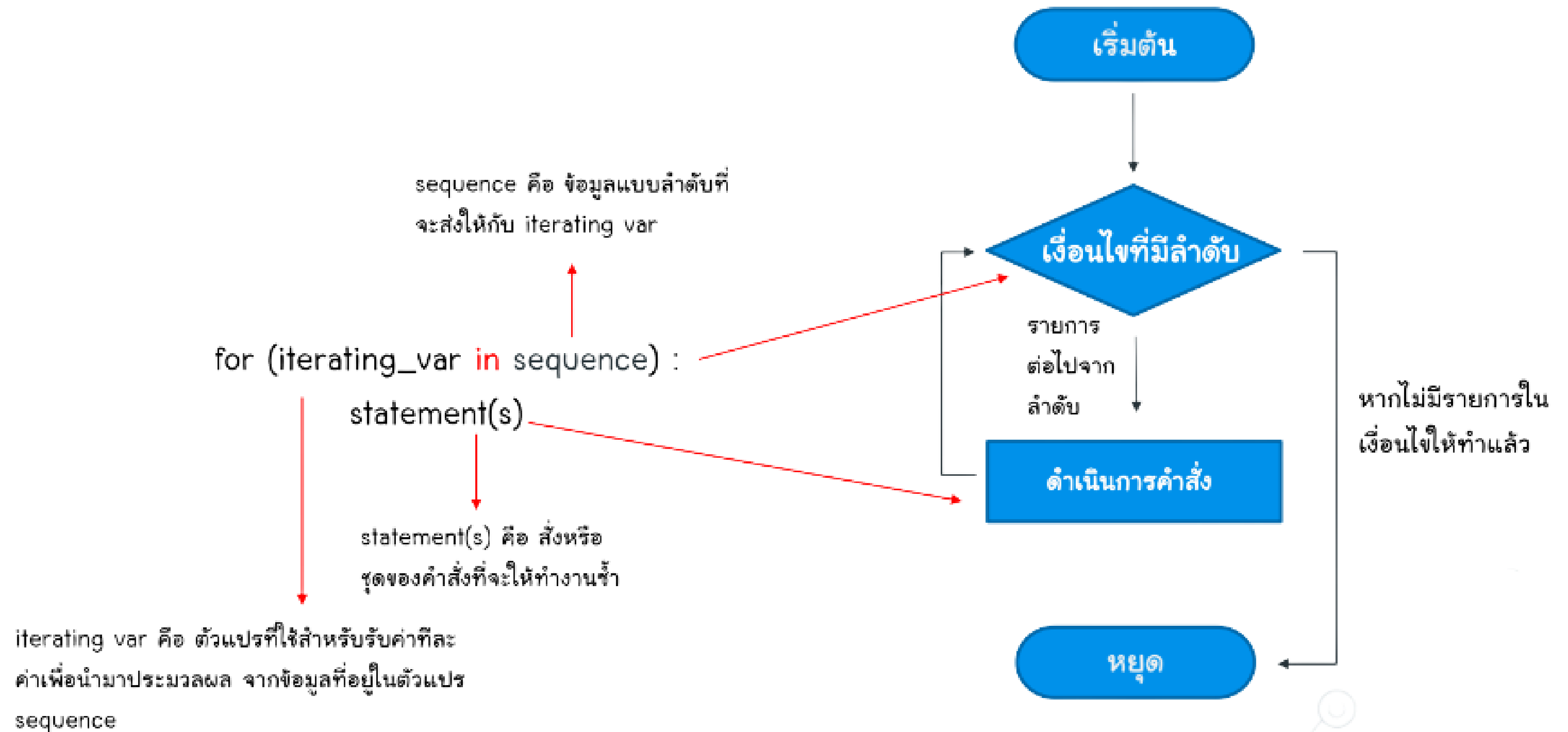
คำสั่งทำซ้ำ **for** **Python**

คำสั่งทำซ้ำ **for**

for เป็นคำสั่งที่ใช้สำหรับการทำซ้ำเช่นเดียวกับ while และต้องมีการตรวจสอบเงื่อนไขก่อนเข้าสู่
เหมือนกัน แต่แตกต่างกันตรงที่ for จะตรวจสอบรายการแบบลำดับแทน

คำสั่งทำซ้ำ for

รูปแบบการเขียนคำสั่งทำซ้ำ for



ตัวอย่างคำสั่งทำซ้ำ `for`

โปรแกรมแสดงตัวอักษรในข้อความ

```
1 #โปรแกรมแสดงตัวอักษรในข้อความ
2 ▼ for letter in "Welcome":
3     print(letter,end='-')
4 print("จบการทำงาน")
```



```
W-e-l-c-o-m-e-จบการทำงาน
> |
```

ตัวอย่างคำสั่งทำซ้ำ `for`

แปลโปรแกรมแสดงตัวอักษรในข้อความ

```
1 #โปรแกรมแสดงตัวอักษรในข้อความ
2 for letter in "Welcome":
3     print(letter,end=' - ')
4 print("จบการทำงาน")
```

บรรทัด 2 ส่งข้อมูลคำว่า "Welcome" ไปไว้ใน ตัวแปร letter (โดยจะส่งข้อมูลไปที่ละตัวอักษร ในการวนซ้ำแต่ละครั้ง จนกว่าจะครบทุกตัวอักษร)

บรรทัด 3 แสดงผลในตัวแปร letter (keyword อาร์กิวเมนต์ `end=' '` จะทำให้ข้อมูลที่แสดงผลอยู่ในแถวเดียวกัน และนำเครื่องหมาย – ต่อท้ายในทุกข้อมูลที่จะแสดงผล ในการวนซ้ำ)

เมื่อดำเนินการตามคำสั่งในบล็อกครบแล้ว ให้กลับไปตรวจสอบรายการ ในบรรทัด 2 (ทำซ้ำไปเรื่อยๆจนกว่าจะแสดงคำว่า Welcome จนครบ)

บรรทัด 4 หากแสดงรายการตามเงื่อนไขในบรรทัด 2 ครบแล้วให้แสดงคำว่า จบการทำงาน

ตัวอย่างคำสั่งทำซ้ำ `for`

โปรแกรมแสดงชื่อใน list

```
1 #โปรแกรมแสดงชื่อใน List
2 name_list = ["Nitiyaporn","Pinbangon","Mata"]
3 ▼ for index in name_list:
4     start = "Hi" + index
5     print(start)
6 print("จบการทำงาน")
```



```
Hi Nitiyaporn
Hi Pinbangon
Hi Mata
จบการทำงาน
> █
```

ตัวอย่างคำสั่งทำซ้ำ `for`

แปลโปรแกรมแสดงชื่อใน list

```
1 #โปรแกรมแสดงชื่อใน List
2 name_list = ["Nitiyaporn","Pinbangon","Mata"]
3 for index in name_list:
4     start = "Hi" + index
5     print(start)
6 print("จบการทำงาน")
```

บรรทัด 2 ประกาศตัวแปร list เก็บชื่อข้อมูลชื่อ 3 ชื่อ

บรรทัด 3 ส่งข้อมูลจากตัวแปร name_list ไปไว้ใน ตัวแปร index(โดยจะส่งข้อมูลไปที่ละชื่อ ในการวนซ้ำแต่ละครั้งจนกว่าจะครบทุกชื่อ)

บรรทัด 4 ประกาศตัวแปร start เก็บคำว่า "Hi " และนำไปต่อหน้า (+) ข้อมูลในตัวแปร index

บรรทัด 5 แสดงข้อมูลในตัวแปร start แล้วกลับไปบรรทัด 3

บรรทัด 6 หากแสดงรายการตามเงื่อนไขในบรรทัด 3 ครบแล้วให้แสดงคำว่า จบการทำงาน

การใช้ คำสั่งทำซ้ำ **for** ร่วมกับคำสั่งขอบเขต **range**

คำสั่งขอบเขตหรือ **range** เป็นคำสั่งที่จะกำหนดขอบเขตการทำงานว่าจะวนซ้ำการทำงานที่รอบ มักจะใช้คู่กับ **for** เสมอ โดยมี 4 แบบดังนี้

1. **range(x)** จะสร้างชุดของข้อมูลเริ่มจาก 0 ถึง $(x - 1)$ โดยเพิ่มขึ้นครั้งละ 1 เช่น เมื่อเรียกใช้คำสั่ง **range(6)** ข้อมูลที่ถูกสร้างขึ้นคือ $[0, 1, 2, 3, 4, 5]$
2. **range(x, y)** จะสร้างชุดของข้อมูลเริ่มต้นจาก x ถึง $(y - 1)$ โดยเพิ่มขึ้นครั้งละ 1 เช่น เมื่อเรียกใช้คำสั่ง **range(3, 10)** ข้อมูลที่ถูกสร้างขึ้นคือ $[3, 4, 5, 6, 7, 8, 9]$

การใช้ คำสั่งทำซ้ำ **for** ร่วมกับคำสั่งขอบเขต **range**

3. `range(x, y, i)` จะสร้างชุดข้อมูลเริ่มต้นจาก x ถึง $(y - 1)$ โดยเพิ่มขึ้นครั้งละ i เช่น เมื่อเรียกใช้คำสั่ง `range(3, 15, 2)` ข้อมูลที่ถูกสร้างขึ้นคือ `[3, 5, 7, 9, 11, 13]` มีข้อสังเกตสำหรับค่า y ตัวสุดท้ายคือ 15 จะไม่ถูกนำมาใส่ไว้ในรายการด้วย เนื่องจากติดเงื่อนไขที่ค่า $y = y - 1$ ดังนั้นค่า y ที่ได้คือ 14 โปรแกรมจึงตัดทั้ง 14 และ 15 ทิ้งไปด้วย เพราะไม่อยู่ในเงื่อนไขทั้งคู่

4. `range(y, x, -i)` จะสร้างชุดของข้อมูลแบบย้อนหลัง จาก y ถึง $(x - 1)$ โดย ลดลงครั้งละ $-i$ เช่น เมื่อเรียกใช้คำสั่ง `range(15, 3, -2)` ข้อมูลที่ถูกสร้างขึ้นคือ `[15, 13, 11, 9, 7, 5]` ตามลำดับ ถ้าต้องการให้ลดค่าทีละ 1 โดยเริ่มตั้งแต่ 15 ถอยหลังไปจนถึง 1 มีรูปแบบคือ `range(15, 0, -1)` หรือถ้าต้องการให้เป็นค่าที่ติดลบ โดยเริ่มตั้งแต่ 0 ลงไปถึง -4 และลดค่าครั้งละ 1 มีรูปแบบคือ `range(0, -5, -1)`

ตัวอย่างการใช้ คำสั่งทำซ้ำ `for` ร่วมกับคำสั่งขอบเขต `range`

แบบที่ 1 `range(x)`

```
1 ▼ for i in range(10):  
2     print(i)
```



```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
▶
```

ตัวอย่างการใช้ คำสั่งทำซ้ำ `for` ร่วมกับคำสั่งขอบเขต `range`

แบบที่ 2 `range(x, y)`

```
1 ▼ for i in range(3,9):  
2     print(i)
```



```
3  
4  
5  
6  
7  
8  
9
```

ตัวอย่างการใช้ คำสั่งทำซ้ำ `for` ร่วมกับคำสั่งขอบเขต `range`

แบบที่ 3 `range(x, y, i)`

```
1 ▼ for i in range(2,30,2):  
2   print(i)
```



```
2  
4  
6  
8  
10  
12  
14  
16  
18  
20  
22  
24  
26  
28  
> █
```

ตัวอย่างการใช้ คำสั่งทำซ้ำ `for` ร่วมกับคำสั่งขอบเขต `range`

แบบที่ 4 `range(x, y, -i)`

```
1 ▼ for i in range(30,2,-2):  
2   print(i)
```



```
30  
28  
26  
24  
22  
20  
18  
16  
14  
12  
10  
8  
6  
4  
> □
```

ตัวอย่างการใช้ คำสั่งทำซ้ำ **for** ร่วมกับคำสั่งขอบเขต **range** และ **input**

โปรแกรมกรอกชื่อ

```
1 nickname = input("ระบุชื่อเล่นของคุณคือ :")  
2 for a in range(3):  
3     print(a , nickname)  
4 print("จบการทำงาน")
```

ระบุชื่อเล่นของคุณคือ :



```
ระบุชื่อเล่นของคุณคือ : กอล์ฟ  
0 กอล์ฟ  
1 กอล์ฟ  
2 กอล์ฟ  
จบการทำงาน  
> 
```