

บทที่ 4

การโปรแกรมแบบทำซ้ำในภาษา Python

Chapter 4 Python and Iteration Statement



Asst.Prof.Dr.Nutthapat Kaewrattanapat
Suan Sunandha Rajabhat University

Course: Computer Programming and Developing Applications for Education
Department of Digital Technology for Education
Faculty of Education, Suan Sunandha Rajabhat University

อาจารย์บรรยาย



ผู้ช่วยศาสตราจารย์ ดร.ณัฐภัทร แก้วรัตนภัทร

การศึกษา: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

- 2565 ปริญญาเอก ปรัชญาดุสิตบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและการสื่อสารเพื่อการศึกษา (GPA. 4.00)
- 2551 ปริญญาโท วิทยาศาสตรมหาบัณฑิต สาขาวิชาระบบสารสนเทศเพื่อการจัดการ (GPA. 3.58)
- 2549 ปริญญาตรี วิทยาศาสตรบัณฑิต เกียรตินิยมอันดับ 1 สาขาวิชาวิทยาการคอมพิวเตอร์ (GPA. 3.66)

คุณวุฒิวิชาชีพและประกาศนียบัตร

- คุณวุฒิวิชาชีพ จาก สถาบันคุณวุฒิวิชาชีพ (องค์การมหาชน) สาขาวิชาชีพอุตสาหกรรมดิจิทัล สาขาธุรกิจดิจิทัลและพาณิชย์อิเล็กทรอนิกส์ อาชีพนักจัดการเทคโนโลยีสารสนเทศสำหรับธุรกิจ ระดับ 6 เลขที่หนังสือรับรอง PQCN-ICT-ECM-0-251100-B-64/000029
- วิทยาศาสตรข้อมูลด้วยภาษาไพทอน, มหาวิทยาลัยธรรมศาสตร์
- การโปรแกรมสำหรับนักภาษาศาสตร์, มหาวิทยาลัยมหิดล
- การโปรแกรมภาษาไพทอน, มหาวิทยาลัยเพนซิลวาเนีย, สหรัฐอเมริกา
- การโปรแกรมสำหรับทุกคน, มหาวิทยาลัยมิชิแกน, สหรัฐอเมริกา

ติดต่อ: nutthapat.ke@ssru.ac.th

Course Description

หลักการ ทฤษฎี ที่เกี่ยวข้องกับการเขียนโปรแกรมและพัฒนาแอปพลิเคชัน หลักการ พัฒนาโปรแกรมคอมพิวเตอร์ คุณสมบัติของโปรแกรมภาษาชนิดต่าง ๆ หลักการ เบื้องต้นเกี่ยวกับองค์ประกอบ ลักษณะคำสั่ง การเขียนโปรแกรม ขั้นตอนวิธี การ วิเคราะห์ การออกแบบ แอปพลิเคชันเพื่อการศึกษา การประเมินซอฟต์แวร์ สามารถพัฒนาแอปพลิเคชันเพื่อการศึกษา

Principles, theories associated with computer programming and development applications, computer programming principles, computer language, elements of computer language, syntax, computer programming, algorithms, analysis and design application for education, software evaluation, candidate teachers able to developing applications for education.

Algorithm Design

Python Programming Language

System Development Life Cycle: SDLC

Reference: <https://edu.ssru.ac.th/useruploads/files/20230724/1772131ed638786bc8d19918b37249af72c36be4.pdf>

Course Outline

- Chapter 1 – Computer Programming
- Chapter 2 – Introduction to Python Programming
- Chapter 3 – Conditional Statement
- **Chapter 4 – Iteration Statement**
- Chapter 5 – Strings
- Chapter 6 – Lists, Tuples, Sets, Dictionaries
- Chapter 7 – Functions
- Chapter 8 – Object-Oriented Programming: OOP
- Chapter 9 – Python and Database Management System
- Chapter 10 – System Development Life Cycle: SDLC

Measurement and Evaluation

การวัดและประเมินผล

1. ระหว่างการจัดการเรียนรู้

- สอบ Pre-test
- การมอบหมายงาน
- สอบ Post-test
- การมีส่วนร่วมในชั้นเรียน

0%
24%
12%
4%

2. การสอบกลางภาค (Midterm Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

20%

3. โครงการประจำภาคเรียน (Term Project)

- โครงการและการนำเสนอ

20%

4. การสอบปลายภาค (Final Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

20%

ร้อยละ	ระดับผลการเรียน	ความหมาย
86 – 100	A	ดีเยี่ยม
82 – 85	A-	ดีเยี่ยม
78 – 81	B+	ดีมาก
74 – 77	B	ดี
70 – 73	B-	ค่อนข้างดี
66 – 69	C+	ปานกลางค่อนข้างดี
62 – 65	C	ปานกลาง
58 – 61	C-	ปานกลางค่อนข้างอ่อน
54 – 57	D+	ค่อนข้างอ่อน
50 – 53	D	อ่อน
46 – 49	D-	อ่อนมาก
0 – 45	F	ตก

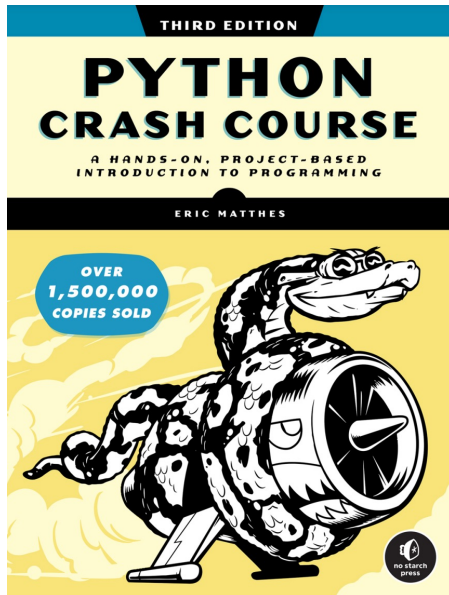
Measurement and Evaluation

ครั้งที่/วันที่	บทเรียน/หัวข้อ	มอบหมายงาน (24%)	สอบ Post-test (12%)	การมีส่วนร่วมในชั้นเรียน (4%)
1 พฤษ 7 รศ 66	แนะนำรายวิชา การวัดและการประเมินผล หัวข้อเรียนรู้ บทที่ 1 การโปรแกรมคอมพิวเตอร์	-	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
2 พฤษ 14 รศ 66	บทที่ 2 พื้นฐานการโปรแกรมภาษาไพทอน	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
3 พฤษ 21 รศ 66	บทที่ 3 การโปรแกรมแบบเงื่อนไข (Conditional Statement)	-	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
4 พฤษ 28 รศ 66 ONLINE	บทที่ 3 การโปรแกรมแบบเงื่อนไข (Conditional Statement) (ต่อ)	3 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
5 พฤษ 4 มค 67	บทที่ 4 การโปรแกรมแบบทำซ้ำ (Iteration Statement)	3 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
6 พฤษ 11 มค 67	บทที่ 5 การโปรแกรมสายอักขระ (String)	3 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
7 พฤษ 18 มค 67	บทที่ 6 ลิสต์ ทิวเพิล เซ็ต และดิกชันนารี	3 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
8 พฤษ 25 มค 67	สอบกลางภาค ปรนัย 35 ข้อ (35 คะแนน) และอัตนัย 1 ข้อ (5 คะแนน) 20%			

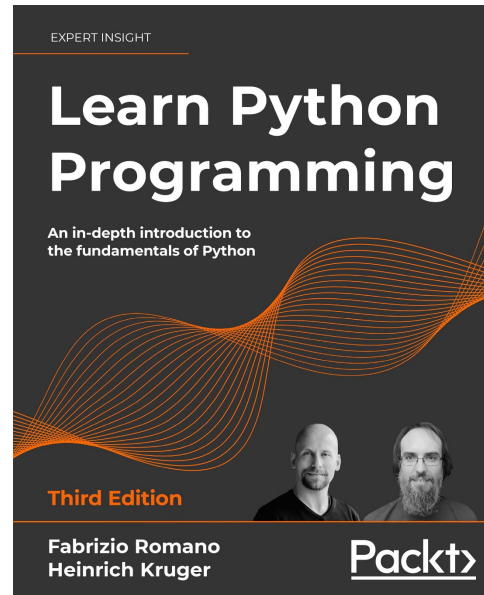
Measurement and Evaluation

ครั้งที่/วันที่	บทเรียน/หัวข้อ	มอบหมายงาน (24%)	สอบ Post-test (12%)	การมีส่วนร่วมในชั้นเรียน (4%)
9 พฤษ 1 กพ 67	บทที่ 7 ฟังก์ชัน	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
10 พฤษ 8 กพ 67	บทที่ 8 การโปรแกรมเชิงวัตถุ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
12 พฤษ 15 กพ 67	บทที่ 9 ภาษาไพทอนกับระบบการจัดการฐานข้อมูล	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
13 พฤษ 22 กพ 67	บทที่ 9 ภาษาไพทอนกับระบบการจัดการฐานข้อมูล ต่อ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
14 พฤษ 29 กพ 67	บทที่ 10 วงจรการพัฒนาระบบ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
15 พฤษ 7 มีค 67	สอบปลายภาค ปรนัย 35 ข้อ (35 คะแนน) และอัตนัย 1 ข้อ (5 คะแนน) 20%			
16 พฤษ 14 มีค 67	ส่งโครงงาน และนำเสนอ 20%			
17 พฤษ 21 มีค 67				

Learning Materials – References



Python Crash Course: A Hands-On, Project-Based Introduction to Programming [3 ed.]



Learn Python Programming
An in-depth introduction to the fundamentals of Python [3 ed.]

Website

- <https://www.w3schools.com/python/>
- <https://realpython.com/>
- <https://docs.python.org/3/tutorial/index.html>
- <https://pythontutor.com/>

Learning Materials – References

ตัวอย่าง Code

<https://cutt.ly/dwGEabVy>

✓ การโปรแกรมแบบวนซ้ำ (Iteration Statement)

Asst.Prof.Dr.Nutthapat Kaewrattanapat email: nutthapat.ke@ssru.ac.th

<https://cutt.ly/dwGEabVy>

การโปรแกรมแบบวนซ้ำในภาษา Python มี 2 ลักษณะหลักๆ คือ

1. while loop
2. for loop and for range

✓ การโปรแกรมวนซ้ำแบบ while loop

1. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 1 ถึง 5

```
[2]  1  i = 1           #ให้ตัวแปร i มีค่าเท่ากับ 1
      2  while i<=5:    #ตรวจสอบเงื่อนไขการวนทำงานซ้ำ เมื่อ i ยังน้อยกว่าหรือ
      3      print(i)   #แสดงค่า i
      4      i+=1       #เพิ่มค่า i ไปอีก 1 หากไม่เพิ่มเงื่อนไขในการวนซ้ำจะเป็นส
```

2. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 1 ถึง 1000

```
[3]  1  i = 1           #ให้ตัวแปร i มีค่าเท่ากับ 1
      2  while i<=1000: #ตรวจสอบเงื่อนไขการวนทำงานซ้ำ เมื่อ i ยังน้อยกว่าหรือ
      3      print(i)   #แสดงค่า i
      4      i+=1       #เพิ่มค่า i ไปอีก 1 หากไม่เพิ่มเงื่อนไขในการวนซ้ำจะเป็นส
```

โครงสร้างการควบคุมโปรแกรม (Program Control Structures)

โครงสร้างการควบคุมโปรแกรม (Program Control Structures) ในการเขียนโปรแกรม มี 3 ลักษณะหลัก ได้แก่

- การโปรแกรมแบบลำดับ (Sequential Statement)
- การโปรแกรมแบบเงื่อนไข (Conditional Statement)
- การโปรแกรมแบบทำซ้ำ (Iteration Statement)

แต่ละลักษณะมีความสำคัญและทำหน้าที่ที่แตกต่างกันในการควบคุมการทำงานของโปรแกรม

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

หากต้องการแสดงผลคำว่า “Hello, Name” ทั้งห้อง จะต้องเขียนโปรแกรมที่บรรทัด

```
print("Hello, Nutthapat")
print("Hello, Thanadol")
print("Hello, Natthapol")
print("Hello, Vilasinee")
print("Hello, Wichuda")
print("Hello, Chinchira")
print("Hello, Phachaya")
print("Hello, Manop")
print("Hello, Panita")
print("Hello, Naphasri")
```

```
Hello, Nutthapat
Hello, Thanadol
Hello, Natthapol
Hello, Vilasinee
Hello, Wichuda
Hello, Chinchira
Hello, Phachaya
Hello, Manop
Hello, Panita
Hello, Naphasri
```

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

หากต้องการเปลี่ยนคำว่า "Hello" เป็นคำว่า "สวัสดี," ทั้งห้อง จะต้องแก้ไขโปรแกรมที่บรรทัด

```
1. print("Hello, Nutthapat")
2. print("Hello, Thanadol")
3. print("Hello, Natthapol")
4. print("Hello, Vilasinee")
5. print("Hello, Wichuda")
6. print("Hello, Chinchira")
7. print("Hello, Phachaya")
8. print("Hello, Manop")
9. print("Hello, Panita")
10. print("Hello, Naphasri")
```



```
1. print("สวัสดี, Nutthapat")
2. print("สวัสดี, Thanadol")
3. print("สวัสดี, Natthapol")
4. print("สวัสดี, Vilasinee")
5. print("สวัสดี, Wichuda")
6. print("สวัสดี, Chinchira")
7. print("สวัสดี, Phachaya")
8. print("สวัสดี, Manop")
9. print("สวัสดี, Panita")
10. print("สวัสดี, Naphasri")
```

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

หากต้องการแสดงจำนวนตั้งแต่ 1 ถึง 10,000 จะต้องเขียนโปรแกรมที่บรรทัด

1. print("1")	14.print("14")	9999.print("9999")
2. print("2")	15.print("15")	10000.print("10000")
3. print("3")	16.print("16")	
4. print("4")	17.print("17")	
5. print("5")	18.print("18")	
6. print("6")	19.print("19")	
7. print("7")	20.print("20")	
8. print("8")	21.print("21")	
9. print("9")	22.print("22")	
10.print("10")	23.print("23")	
11.print("11")	24.print("24")	
12.print("12")	25.print("25")	
13.print("13")	26.print("26")	

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

การโปรแกรมแบบทำซ้ำ (Iteration Statement) เป็นเทคนิคในการเขียนโปรแกรมให้ทำงานซ้ำๆ กันหลายรอบ มีประโยชน์ในการเขียนโปรแกรมที่ต้องทำซ้ำ เช่น

1. การพิมพ์ข้อมูลที่ละรายการ
2. การคำนวณจำนวนที่มีหลายค่า
3. การวนซ้ำผ่านสมาชิกของชุดข้อมูล
4. การวนซ้ำทำงานตามเงื่อนไขที่กำหนดเพื่อหาข้อมูล หรือแก้ปัญหาที่ต้องมีการไล่เรียง

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

while loop syntax:

while condition:

คำสั่งที่ต้องการทำงานเมื่อเงื่อนไขเป็นจริง

การใช้ **while loop** เพื่อทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ (False)

for loop syntax:

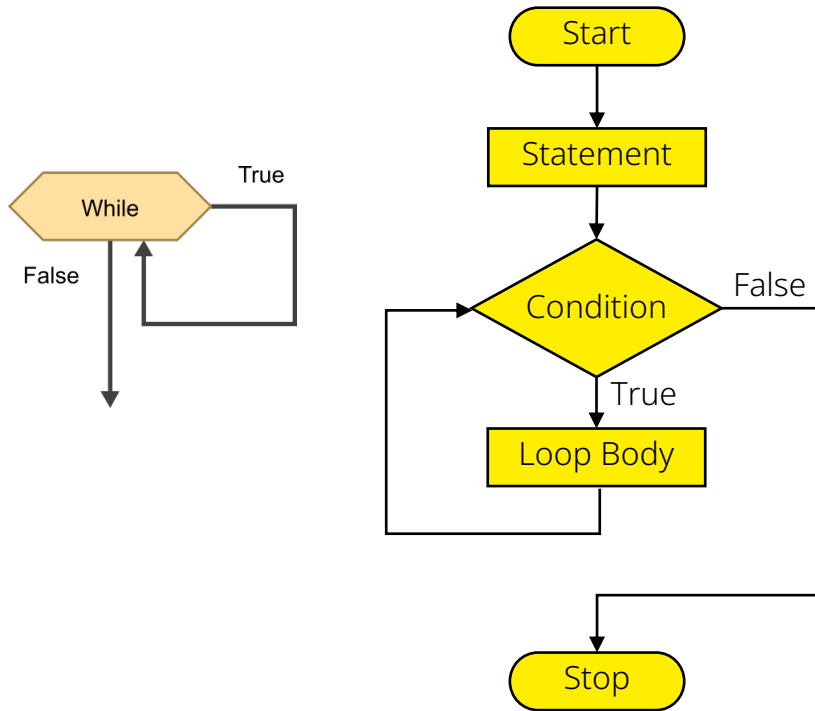
for item in collection:

คำสั่งที่ต้องการทำงานกับ item

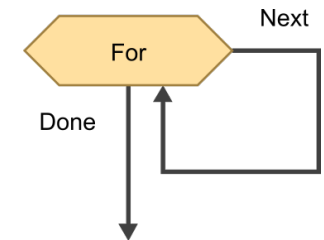
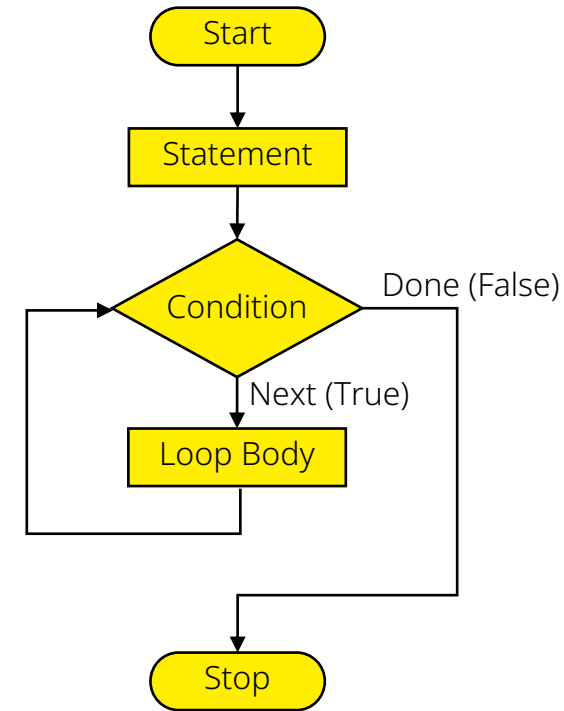
การใช้ **for loop** สำหรับการวนลูปผ่านคอลเลกชันข้อมูล เช่น รายการ, สตริง, คอลเลกชันอื่นๆ และทำงานกับแต่ละรายการในคอลเลกชัน

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

โครงสร้างการควบคุมทำซ้ำแบบ while
While Statement

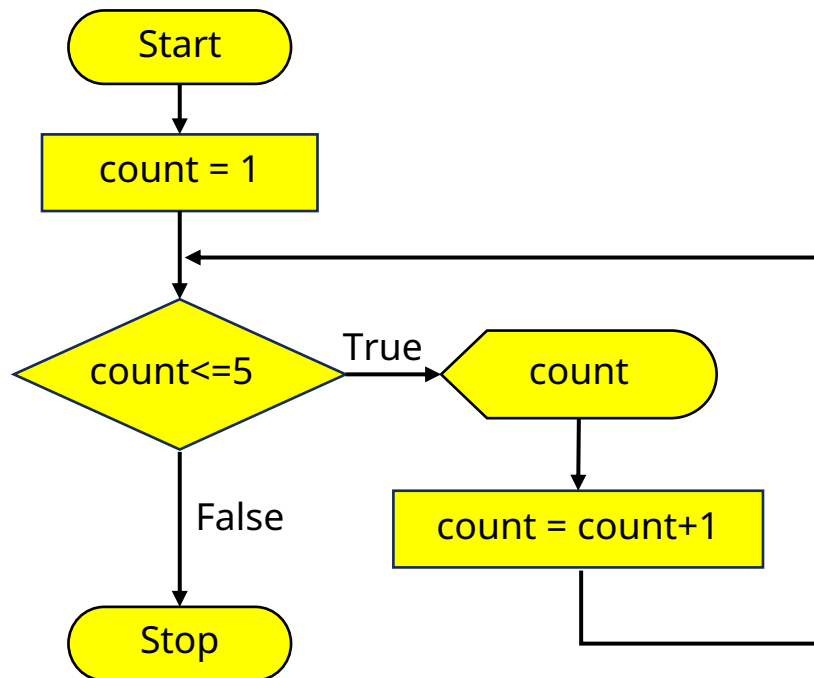
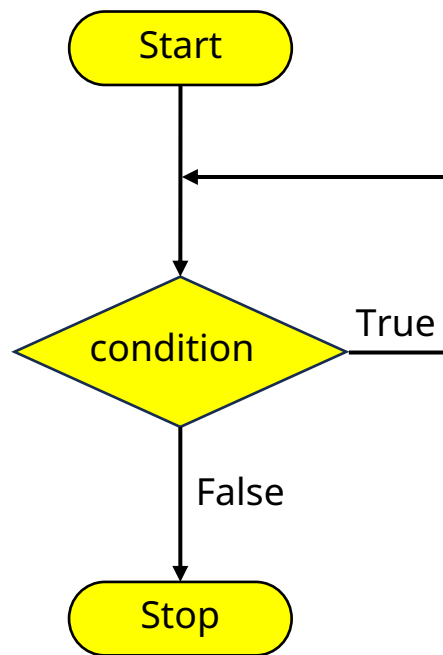


โครงสร้างการควบคุมทำซ้ำแบบ for
For Statement



การโปรแกรมแบบทำซ้ำ (Iteration Statement)

การใช้ while loop เพื่อทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ (False)

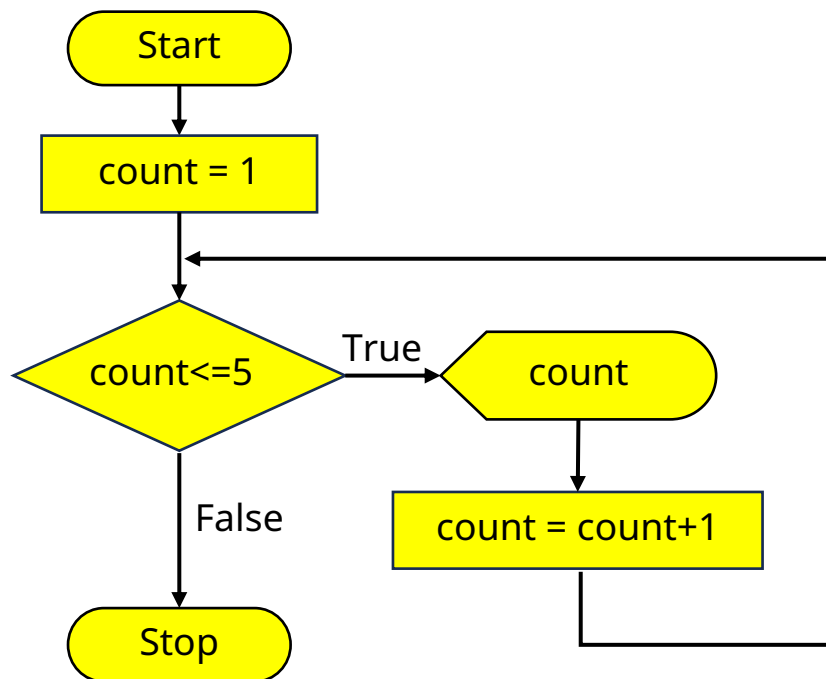


Python

```
1. count = 1
2. while count <= 5:
3.     print(count)
4.     count += 1
```

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

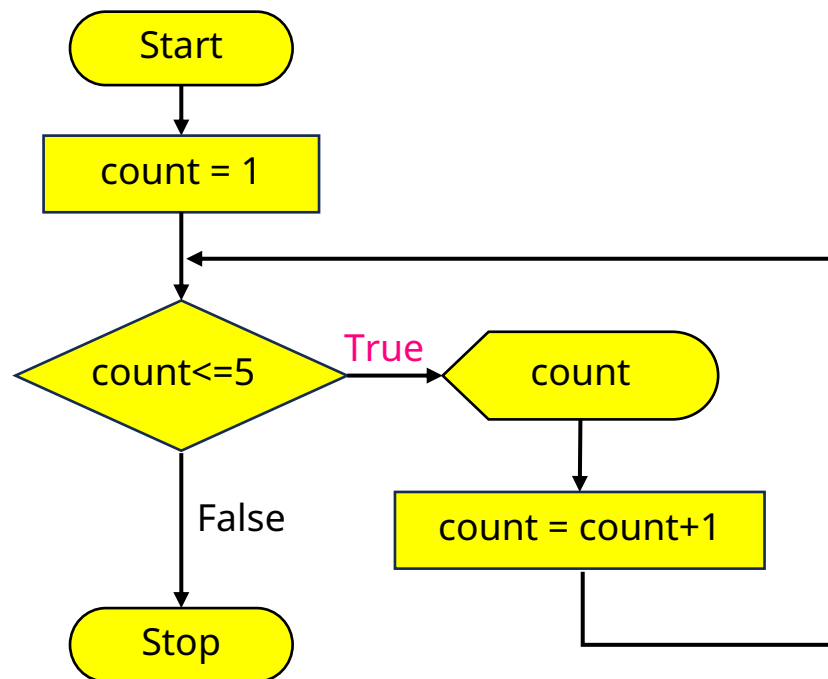
การใช้ while loop เพื่อทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ (False)



รอบที่	ค่าของตัวแปร count	เงื่อนไข (True, False)	แสดงผล

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

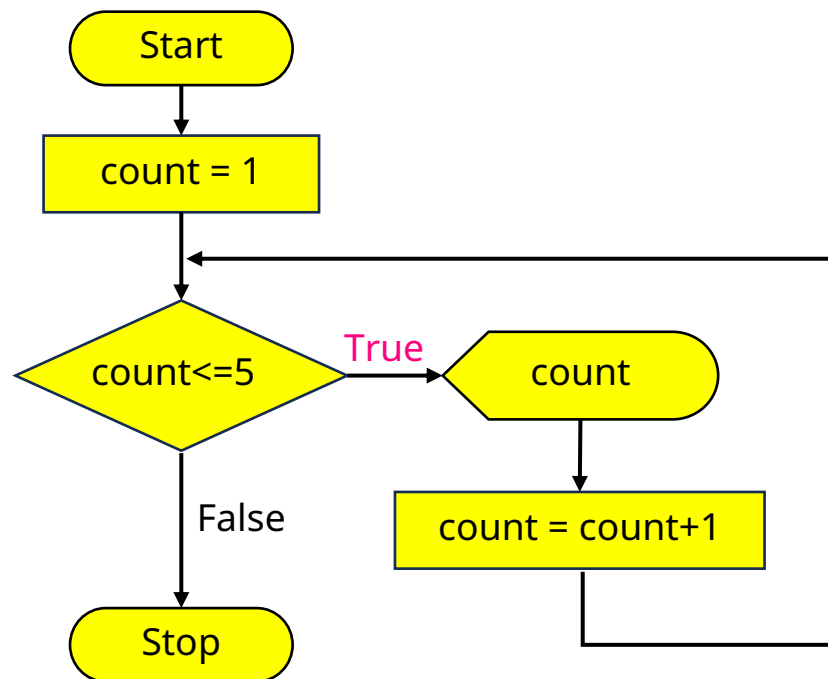
การใช้ while loop เพื่อทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ (False)



รอบที่	ค่าของตัวแปร count	เงื่อนไข (True, False)	แสดงผล
1	1	True	1

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

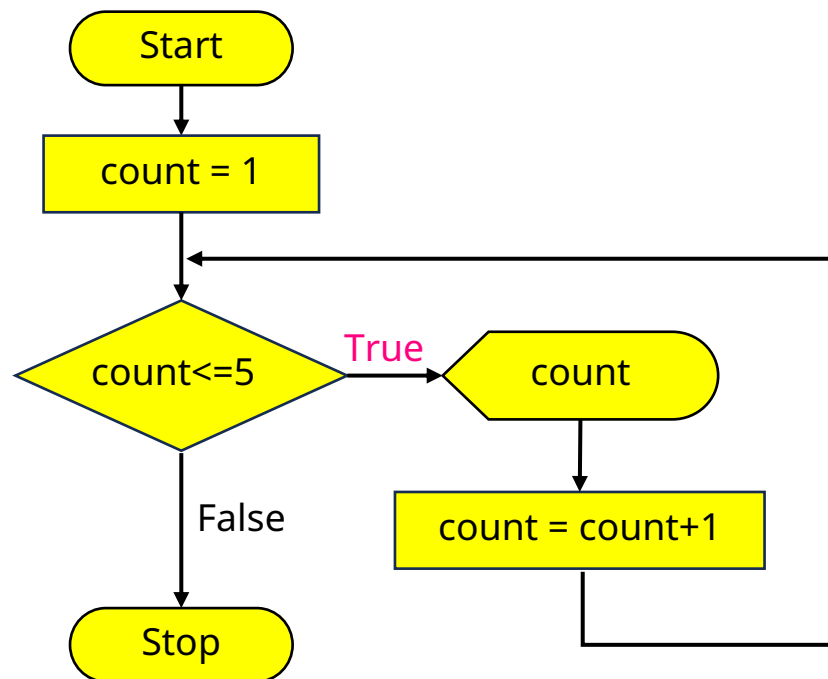
การใช้ while loop เพื่อทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ (False)



รอบที่	ค่าของตัวแปร count	เงื่อนไข (True, False)	แสดงผล
1	1	True	1
2	2	True	2

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

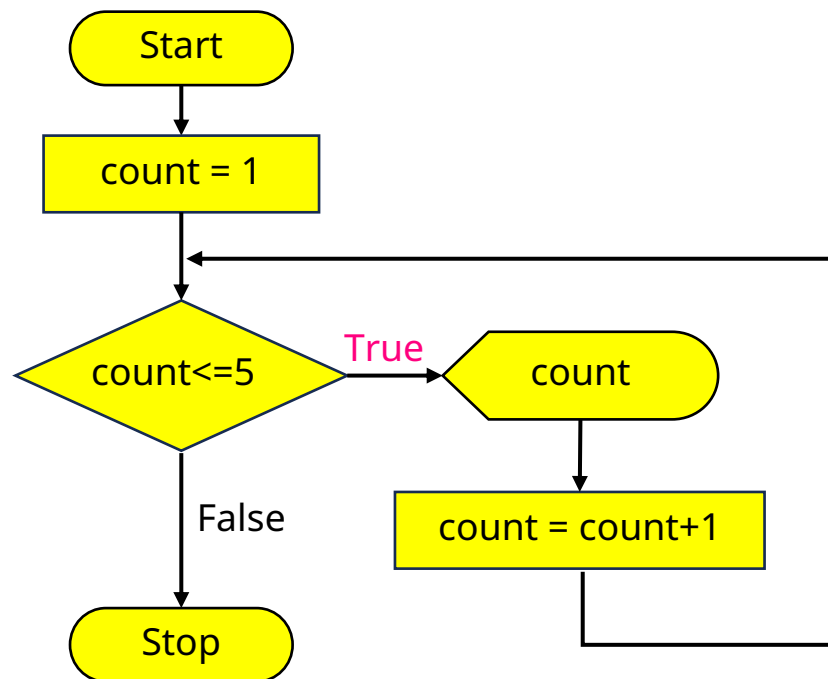
การใช้ while loop เพื่อทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ (False)



รอบที่	ค่าของตัวแปร count	เงื่อนไข (True, False)	แสดงผล
1	1	True	1
2	2	True	2
3	3	True	3

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

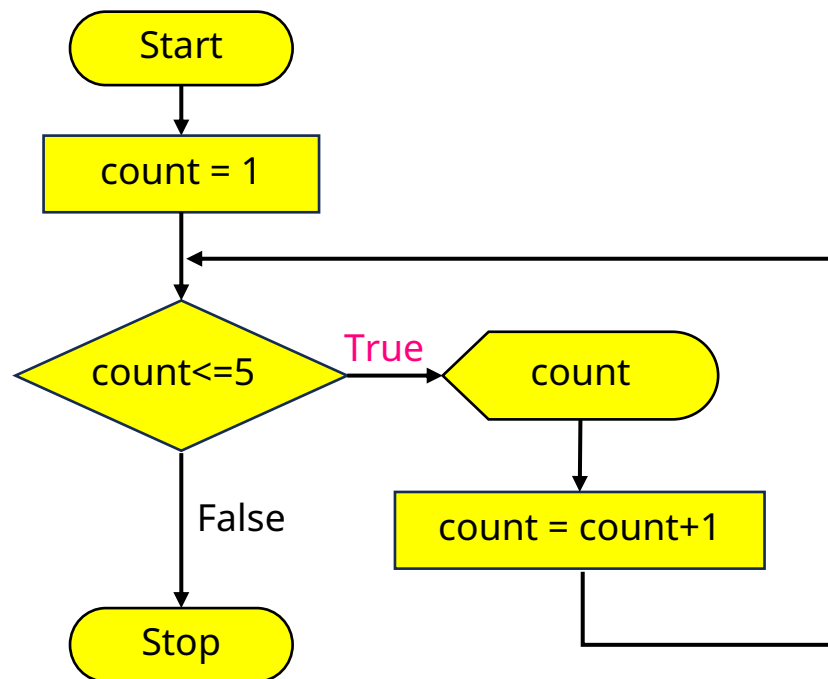
การใช้ while loop เพื่อทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ (False)



รอบที่	ค่าของตัวแปร count	เงื่อนไข (True, False)	แสดงผล
1	1	True	1
2	2	True	2
3	3	True	3
4	4	True	4

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

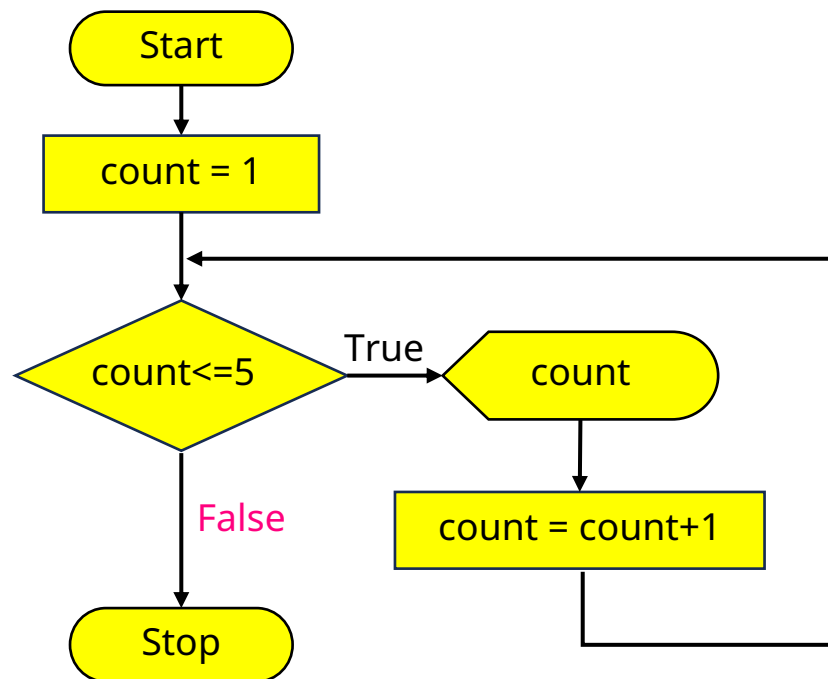
การใช้ while loop เพื่อทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ (False)



รอบที่	ค่าของตัวแปร count	เงื่อนไข (True, False)	แสดงผล
1	1	True	1
2	2	True	2
3	3	True	3
4	4	True	4
5	5	True	5

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

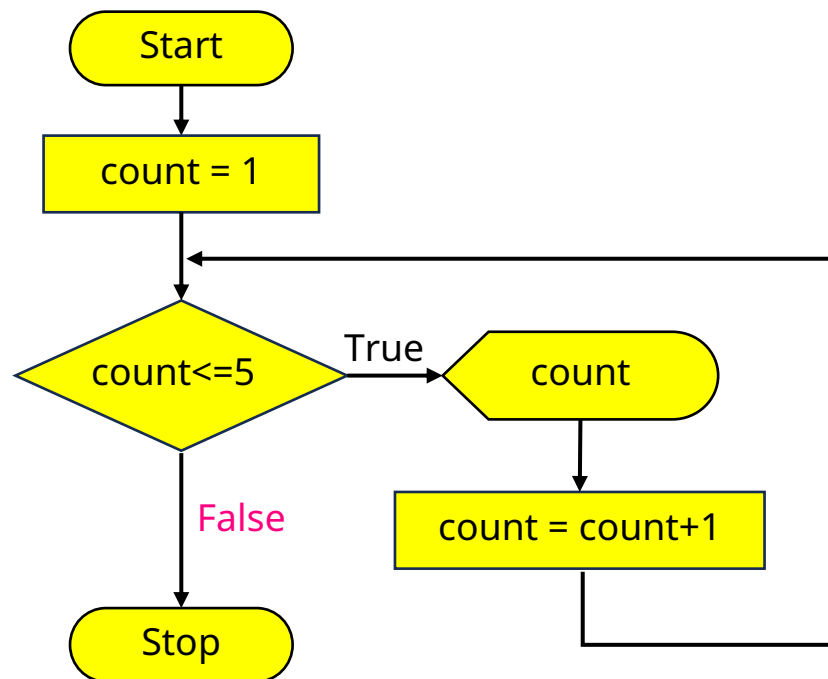
การใช้ while loop เพื่อทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ (False)



รอบที่	ค่าของตัวแปร count	เงื่อนไข (True, False)	แสดงผล
1	1	True	1
2	2	True	2
3	3	True	3
4	4	True	4
5	5	True	5
6	6	False	-

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

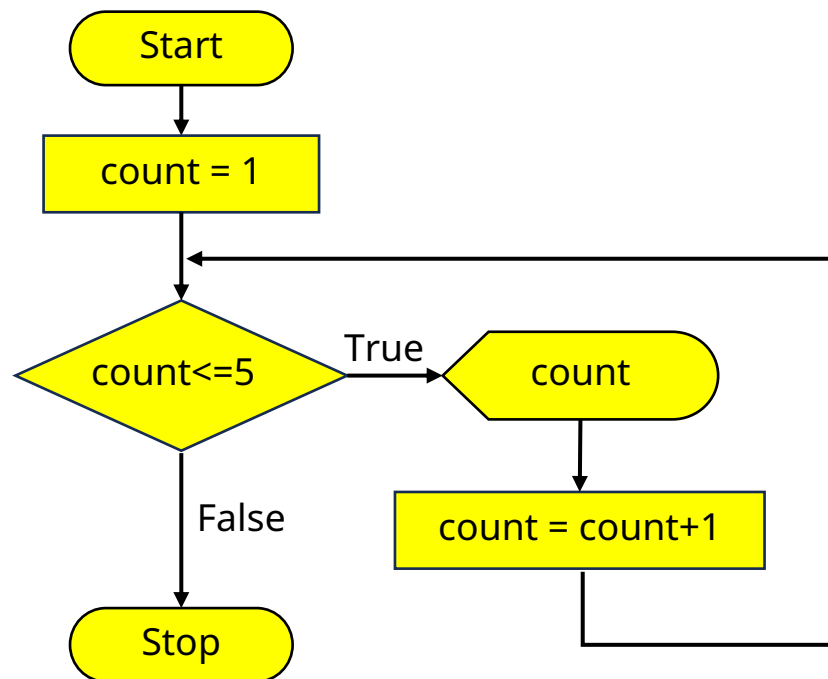
การใช้ while loop เพื่อทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ (False)



รอบที่	ค่าของตัวแปร count	เงื่อนไข (True, False)	แสดงผล
1	1	True	1
2	2	True	2
3	3	True	3
4	4	True	4
5	5	True	5
6	6	False	-

การโปรแกรมแบบทำซ้ำ (Iteration Statement)

1. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 1 ถึง 5

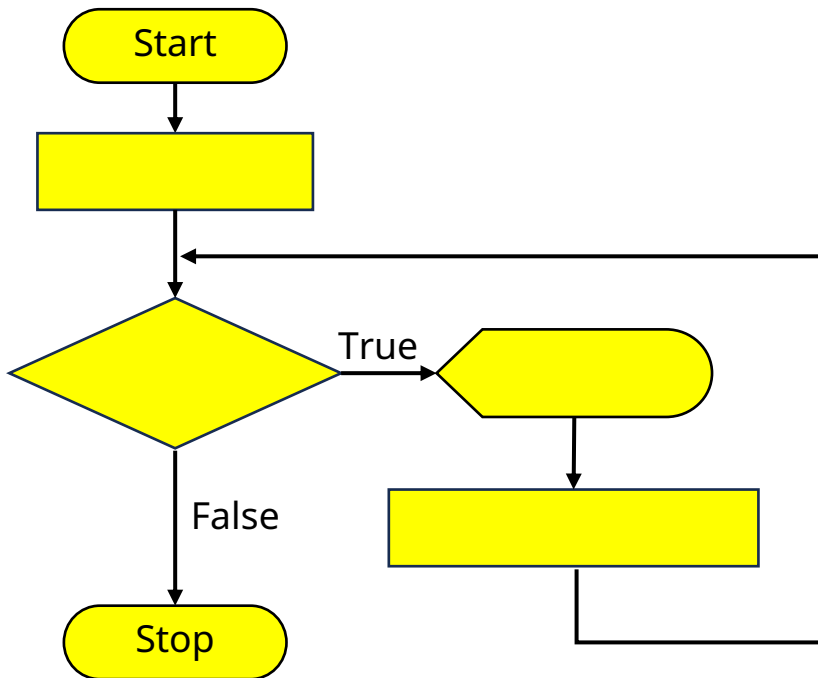


```
1. count = 1
2. while count <= 5:
3.     print(count)
4.     count += 1
```

Python Tutor
<https://cutt.ly/DwGxR4R2>

Iteration Statement – while

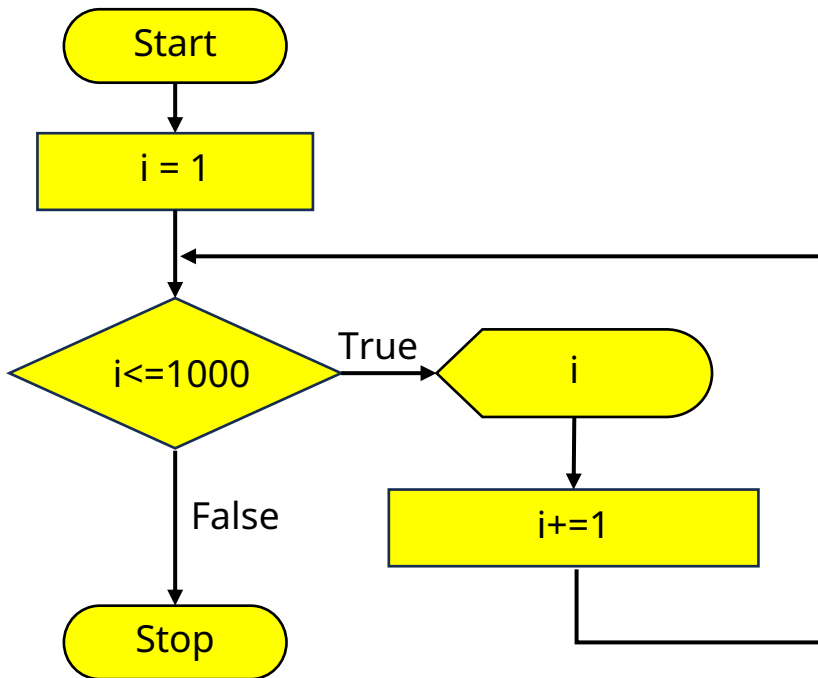
2. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 1 ถึง 1000



```
1.      =  
2.  while      :  
3.      print ( )  
4.      +=1
```

Iteration Statement – while

2. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 1 ถึง 1000



```
1. i = 1
2. while i <= 1000:
3.     print(i)
4.     i += 1
```

Python Tutor
<https://cutt.ly/MwGxAiG8>

Iteration Statement – while

3. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 5 ถึง 10 ให้เติมโปรแกรมให้ถูกต้อง

```
1.  i =  
2.  while i :  
3.      print ( )  
4.      i+=
```

Iteration Statement – while

3. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 5 ถึง 10 ให้เติมโปรแกรมให้ถูกต้อง

```
1.  i = 5
2.  while i<=10 :
3.      print(i)
4.      i+=1
```

Iteration Statement – while

4. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 2, 4, 6, ... ถึง 20 ให้เติมโปรแกรมให้ถูกต้อง

```
1.  i =  
2.  while i      :  
3.      print (   )  
4.      i+=
```

Iteration Statement – while

4. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 2, 4, 6, ... ถึง 20 ให้เต็มโปรแกรมให้ถูกต้อง

```
1. i = 2
2. while i<=20:
3.     print(i)
4.     i+=2
```

Iteration Statement – while

5. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 1, 3, 5, 7, ... ถึง 19 ให้เติมโปรแกรมให้ถูกต้อง

```
1.  i =  
2.  while      :  
3.      print(i)  
4.      i
```

Iteration Statement – while

5. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 1, 3, 5, 7, ... ถึง 19 ให้เติมโปรแกรมให้ถูกต้อง

```
1. i = 1
2. while i<=19:
3.     print(i)
4.     i+=2
```

Iteration Statement – while

6. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 50, 49, 48, ... 0

```
1.  i =  
2.  while      :  
3.      print(i)  
4.      i
```

Iteration Statement – while

6. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 50, 49, 48, ... 0

```
1. i = 50
2. while i >= 0:
3.     print(i)
4.     i -= 1
```

Iteration Statement – while

7. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50

```
1. i =  
2. while i<=50:  
3.     if i == 0:  
4.         print(i)  
5.     i
```

Iteration Statement – while

7. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50

```
1. i = 1
2. while i<=50:
3.     if i%2 == 0:
4.         print(i)
5.     i+=1
```

Iteration Statement – while

8. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50 และสรุปว่ามีทั้งหมดกี่จำนวน

```
1.  i = 1
2.  count = 0
3.  while i<=50:
4.      if i%2 == 0:
5.          print(i)
6.          count+=1
7.      i+=1
8.  print("มีทั้งหมด", count, "จำนวน")
```

Iteration Statement – while

8. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50 และสรุปว่ามีทั้งหมดกี่จำนวน

```
1.  i =
2.  count =
3.  while i      :
4.      if      == 0:
5.          print(i)
6.          count
7.          i
8.  print("มีทั้งหมด", count, "จำนวน")
```

Iteration Statement – while

8. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50 และสรุปว่ามีทั้งหมดกี่จำนวน

```
1. i = 1
2. count = 0
3. while i<=50:
4.     if i%2 == 0:
5.         print(i)
6.         count+=1
7.     i+=1
8. print("มีทั้งหมด", count, "จำนวน")
```

Iteration Statement – while

9. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนที่หาร 5 ลงตัว ตั้งแต่ 1 ถึง 100 และสรุปว่ามีทั้งหมดกี่จำนวน

```
1.  i =
2.  count = 0
3.  while i      :
4.      if      :
5.          print ( )
6.          count
7.      i
8.  print ("มีทั้งหมด", count, "จำนวน")
```

Iteration Statement – while

9. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนที่หาร 5 ลงตัว ตั้งแต่ 1 ถึง 100 และสรุปว่ามีทั้งหมดกี่จำนวน

```
1.  i = 1
2.  count = 0
3.  while i<=100:
4.      if i%5==0:
5.          print(i)
6.          count+=1
7.      i+=1
8.  print("มีทั้งหมด", count, "จำนวน")
```

Iteration Statement – while

10. การโปรแกรมเพื่อหาผลรวมของจำนวนตั้งแต่ 1, 2, 3, ... ถึง 10

```
1.  i =  
2.  n =  
3.  summation =  
4.  while      :  
5.      summation  
6.      i  
7.  print(summation)
```

Iteration Statement – while

10. การโปรแกรมเพื่อหาผลรวมของจำนวนตั้งแต่ 1, 2, 3, ... ถึง 10

```
1.  i = 1
2.  n = 10
3.  summation = 0
4.  while i<=n:
5.      summation += i
6.      i+=1
7.  print(summation)
```

Iteration Statement – while

11. การโปรแกรมเพื่อหาค่าเฉลี่ย (average) โดยค่าเฉลี่ยมาจาก ผลรวมของจำนวนทั้งหมด หารด้วยจำนวนทั้งหมด หรือ

$$\text{average} = \text{summation} / \text{count}$$

```
1  # กำหนดค่าเริ่มต้นของตัวแปร
2  i = 1
3  n = 10
4  summation = 0
5  count = 0
6  average = 0
7
8  # ใช้ลูป while เพื่อหาผลรวมของตัวเลข 1 ถึง 10
9  while i <= n:
10     summation += i  # เพิ่มค่า i เข้าไปใน summation
11     i += 1          # เพิ่มค่า i ไป 1
12     count += 1      # เพิ่มค่า count ไป 1 เพื่อบันทึกจำนวนตัวเลขที่ถูกรวบรวม
13
14  # คำนวณค่าเฉลี่ย
15  average = summation / count
16
17  # แสดงผลรวมของตัวเลข, จำนวนตัวเลขที่ถูกรวบรวม, และค่าเฉลี่ย
18  print(f"ผลรวมของตัวเลข 1 ถึง {n} คือ {summation}")
19  print(f"จำนวนตัวเลขทั้งหมดคือ {count}")
20  print(f"ค่าเฉลี่ยของตัวเลข 1 ถึง {n} คือ {average}")
```

การใช้ f string เบื้องต้น

```
print (f"สินค้า{product} มีราคา{price} บาท")
```

↑
เพิ่ม f

↑
{ตัวแปร}

↑
{ตัวแปร}

```
product = "Apple"
```

```
price = 50
```

```
print (f"สินค้า{product} มีราคา{price} บาท")
```

Iteration Statement – while

12. การโปรแกรมเพื่อแสดงจำนวน * เท่ากับจำนวนของตัวเลขตั้งแต่ 1 ถึง 10

```
1.  i = 1
2.  star = 1
3.  while i<=10:
4.      while star<=i:
5.          print("*", end=" ")
6.          star+=1
7.      i+=1
8.      star = 1
9.      print()
```

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

Iteration Statement – while

12. การโปรแกรมเพื่อแสดงจำนวน * เท่ากับจำนวนของตัวเลขตั้งแต่ 1 ถึง 10

```
1.  i = 1
2.  star = 1
3.  while i<=10:
4.      while star<=i:
5.          print("*", end="")
6.          star+=1
7.      i+=1
8.      star = 1
9.      print()
```

iteration: i	star<=i	print
1	1<=1	*
2	1<=2	**
3	1<=3	***
4	1<=4	****
5	1<=5	*****
6	1<=6	*****
7	1<=7	*****
8	1<=8	*****
9	1<=9	*****
10	1<=10	*****

Iteration Statement – while

12. การโปรแกรมเพื่อแสดงจำนวน * เท่ากับจำนวนของตัวเลขตั้งแต่ 1 ถึง 10

```
1.      # กำหนดค่าเริ่มต้นของตัวแปร i และ star
2.      i = 1                # ตัวแปร i ใช้ในลูปเพื่อควบคุมจำนวนบรรทัด
3.      star = 1            # ตัวแปร star ใช้ในลูปภายในเพื่อควบคุมจำนวนดาว

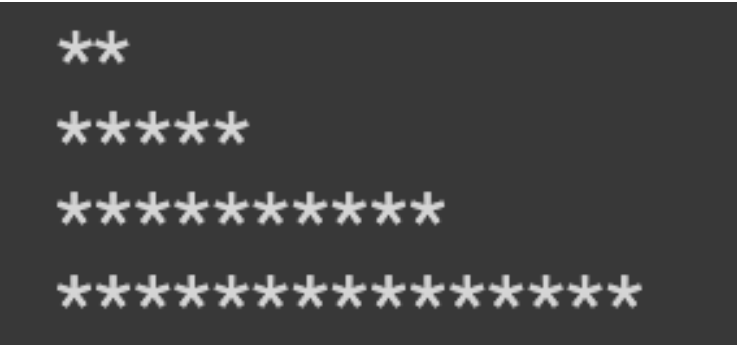
4.      # ใช้ลูป while เพื่อสร้างรูปทรงสามเหลี่ยมดาว
5.      while i <= 10:
6.          while star <= i:
7.              print("*", end="")    # แสดงดาวและไม่ขึ้นบรรทัดใหม่ด้วย end=""
8.              star += 1             # เพิ่มค่า star ไป 1 เพื่อเพิ่มจำนวนดาวในแต่ละบรรทัด
9.          i += 1                  # เพิ่มค่า i ไป 1 เพื่อเลื่อนจาก 1 ไปเรื่อยๆ ถึง 10
10.         star = 1                # กำหนดค่าเริ่มต้นของตัวแปร star ใหม่
11.         print()                 # ขึ้นบรรทัดใหม่
```

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

คำสั่งทำซ้ำ string

13. ในภาษา Python มีความพิเศษเกี่ยวกับคำสั่งที่ใช้ในการเพิ่มจำนวนข้อความ

```
print ("*" * 2)
print ("*" * 5)
print ("*" * 10)
print ("*" * 15)
```



```
**
*****
*****
*****
```

Iteration Statement – while

14. การโปรแกรมเพื่อแสดงจำนวน * เท่ากับจำนวนของตัวเลขตั้งแต่ 1 ถึง 10 โดยใช้การคูณข้อความ

```
1. i = 1
2. while i<=10:
3.     print ("*" * i)
4.     i+=1
```

iteration: i	star
1	print("*" * 1)
2	print("*" * 2)
3	print("*" * 3)
4	print("*" * 4)
5	print("*" * 5)
6	print("*" * 6)
7	print("*" * 7)
8	print("*" * 8)
9	print("*" * 9)
10	print("*" * 10)

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

Iteration Statement – while

15. การโปรแกรมเพื่อหาค่า Factorial 3! โดยที่ Factorial คือการหาผลคูณของจำนวนที่ต้องการ เช่น

- 3! มีค่าเท่ากับ $3 \times 2 \times 1$ ได้ผลลัพธ์เท่ากับ 6
- 5! มีค่าเท่ากับ $5 \times 4 \times 3 \times 2 \times 1$ ได้ผลลัพธ์เท่ากับ 120

```
1  # กำหนดค่าตัวแปร fac เพื่อระบุตัวเลขที่ต้องการหาค่า Factorial
2  fac = 3
3
4  # กำหนดค่าเริ่มต้นของตัวแปร i และ result
5  i = 1          # ตัวแปร i ใช้ในการวนลูปเพื่อคูณตัวเลขต่อเนื่อง
6  result = 1     # ตัวแปร result ใช้เก็บผลคูณของตัวเลขต่อเนื่อง
7
8  # ใช้ลูป while เพื่อคำนวณ Factorial ของตัวเลข fac
9  while i <= fac:
10     result = result * i  # คูณตัวแปร result ด้วยตัวแปร i เพื่อคำนวณ Factorial
11     i += 1              # เพิ่มค่า i ไป 1 เพื่อคูณตัวเลขถัดไป
12
13  # แสดงผลลัพธ์ Factorial
14  print(f"{fac}! = {result}")
```

Iteration Statement – while

16. การโปรแกรมเพื่อแสดงจำนวนที่ได้จาก Factorial 3! เช่น

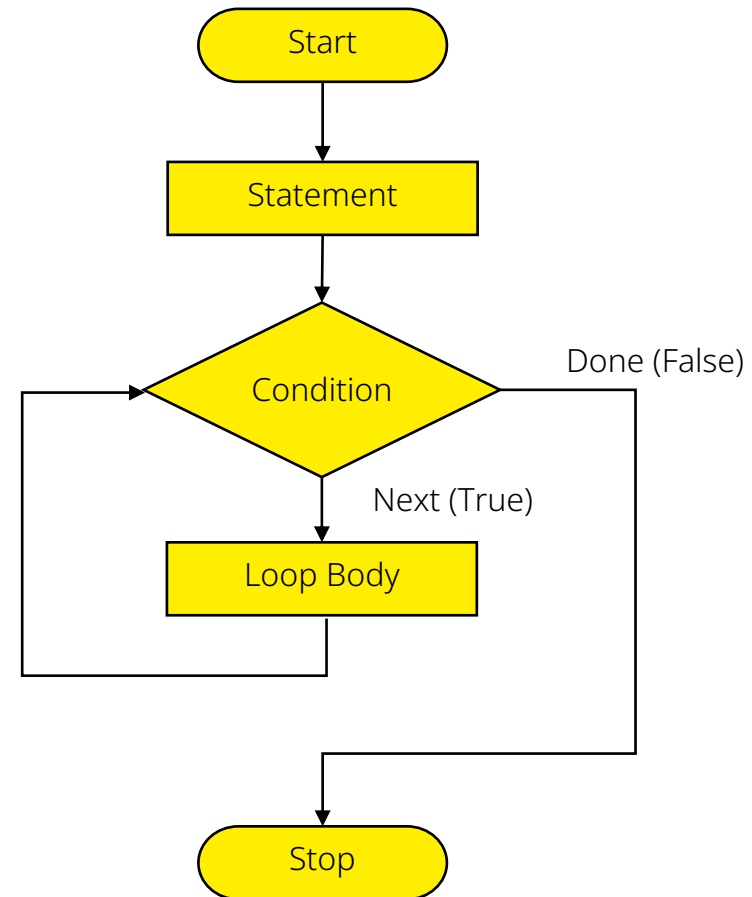
- 3! แสดงผล $3! = 3 \times 2 \times 1 = 6$

```
1  # กำหนดค่าตัวแปร fac และ result เริ่มต้น
2  fac = 3
3  result = 1
4
5  # แสดงข้อความและตัวแปร fac และตัวแปร result ที่เริ่มต้น
6  print(fac, "! = ", sep="", end="")
7
8  # ใช้ลูป while เพื่อคำนวณ Factorial
9  while fac > 0:
10     if fac != 1:
11         # แสดงตัวเลขและสัญลักษณ์คูณถ้าไม่ใช่ตัวสุดท้าย
12         print(fac, "x", sep="", end="")
13     else:
14         # แสดงตัวสุดท้ายโดยไม่มีสัญลักษณ์คูณ
15         print(fac, end="")
16
17     # คำนวณผลลัพธ์ Factorial
18     result = result * fac
19
20     # ลดค่า fac ลง 1 ในทุกรอบ
21     fac -= 1
22
23 # แสดงผลลัพธ์ของ Factorial
24 print(" = ", result, sep="")
```

Iteration Statement – for

การใช้ for loop สำหรับการวนลูปผ่านคอลเลกชันข้อมูล เช่น ระยะ, รายการ, สตริง, คอลเลกชันอื่นๆ และทำงานกับแต่ละรายการในคอลเลกชัน

โครงสร้างการควบคุมทำซ้ำแบบ for
For Statement



ฟังก์ชัน range()

range() เป็นฟังก์ชันในภาษา Python เพื่อสร้างชุดของตัวเลขแบบติดต่อกัน (sequence of numbers) โดยมักใช้ในการสร้างลูป (loop) เพื่อทำงานกับข้อมูลที่มีลำดับหรือเป็นตัวเลขที่เรียงต่อกัน

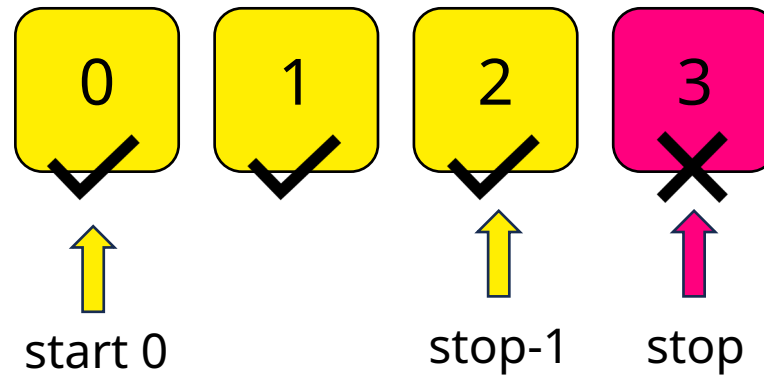
range() มีรูปแบบการใช้งาน 3 ลักษณะ ดังนี้

1. **range(stop):** สร้างชุดตัวเลขตั้งแต่ 0 ถึง stop - 1
2. **range(start, stop):** สร้างชุดตัวเลขตั้งแต่ start ถึง stop - 1
3. **range(start, stop, step):** สร้างชุดตัวเลขตั้งแต่ start ถึง stop - 1 โดยที่ลำดับของตัวเลขมีการเพิ่มขึ้นทีละ step

ฟังก์ชัน range()

1. **range(stop):** สร้างชุดตัวเลขตั้งแต่ 0 ถึง stop - 1

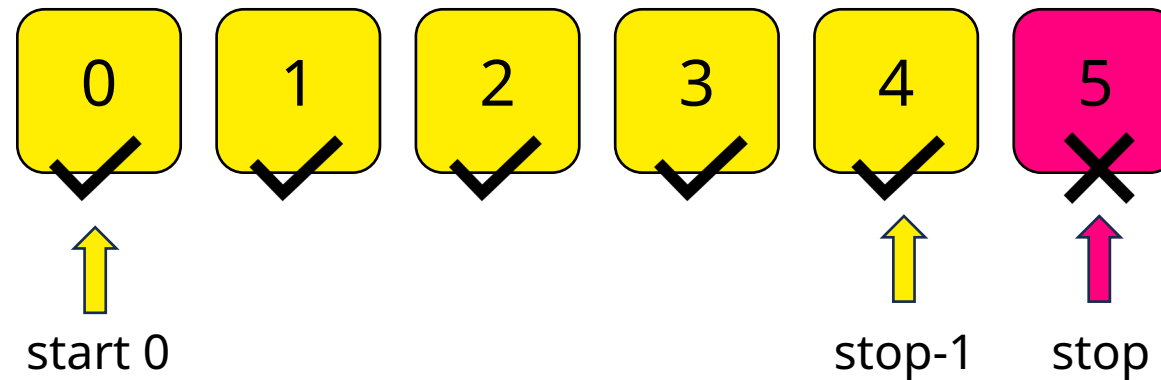
range (3)



ฟังก์ชัน range()

1. **range(stop):** สร้างชุดตัวเลขตั้งแต่ 0 ถึง stop - 1

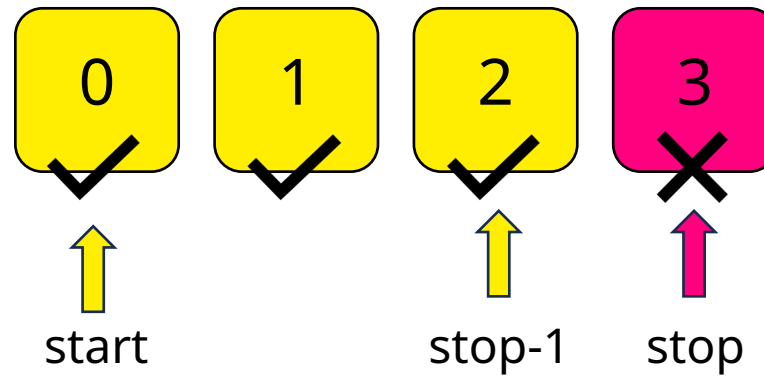
range (5)



ฟังก์ชัน range()

2. range(start, stop): สร้างชุดตัวเลขตั้งแต่ start ถึง stop - 1

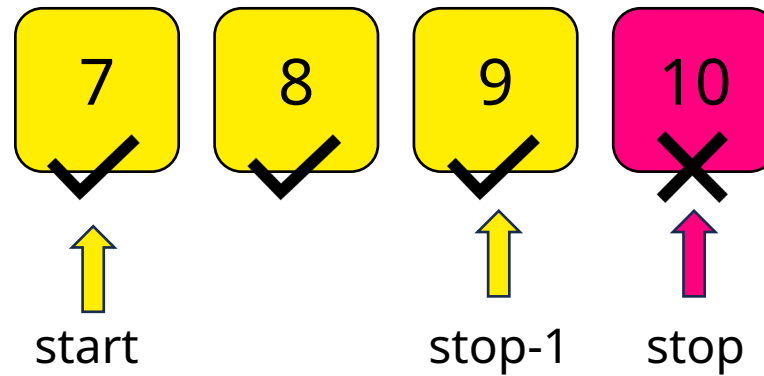
range (0 , 3)



ฟังก์ชัน range()

2. range(start, stop): สร้างชุดตัวเลขตั้งแต่ start ถึง stop - 1

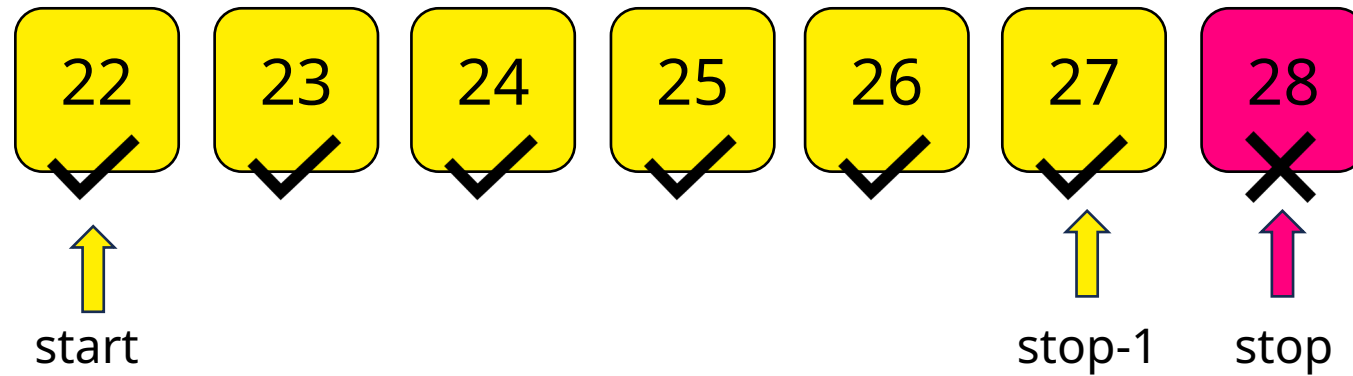
`range ( 7 , 10 )`



ฟังก์ชัน range()

2. range(start, stop): สร้างชุดตัวเลขตั้งแต่ start ถึง stop - 1

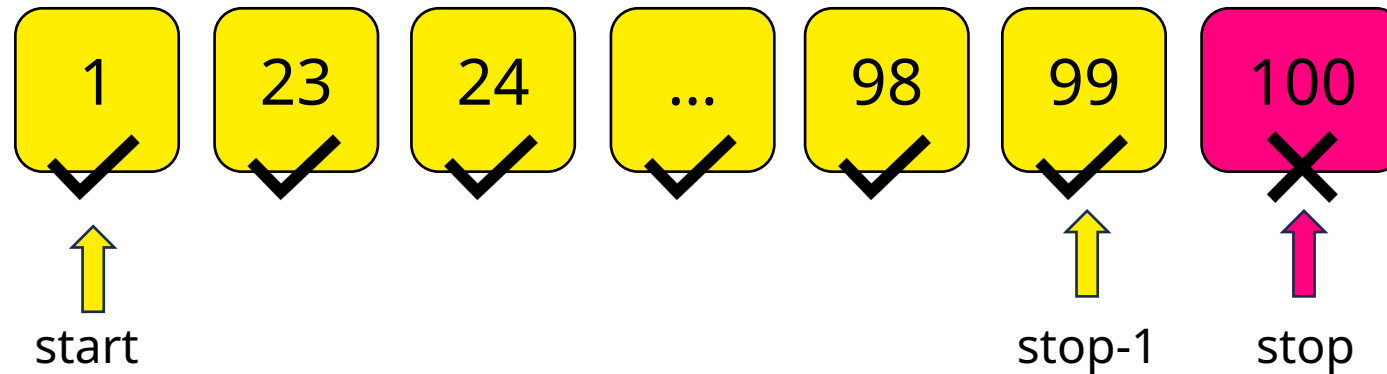
range (22 , 28)



ฟังก์ชัน range()

2. range(start, stop): สร้างชุดตัวเลขตั้งแต่ start ถึง stop - 1

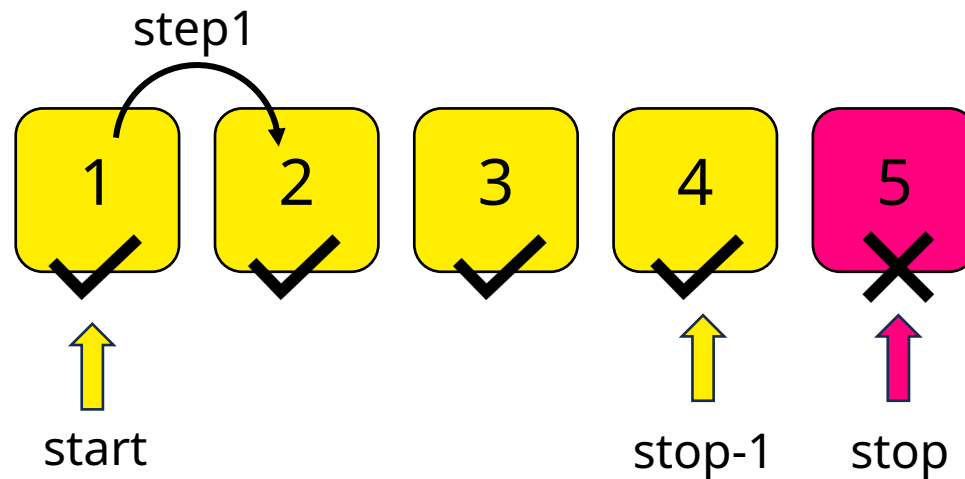
range (1, 100)



ฟังก์ชัน range()

3. range(start, stop, step): สร้างชุดตัวเลขตั้งแต่ start ถึง stop - 1 โดยที่ลำดับของตัวเลขมีการเพิ่มขึ้นทีละ step

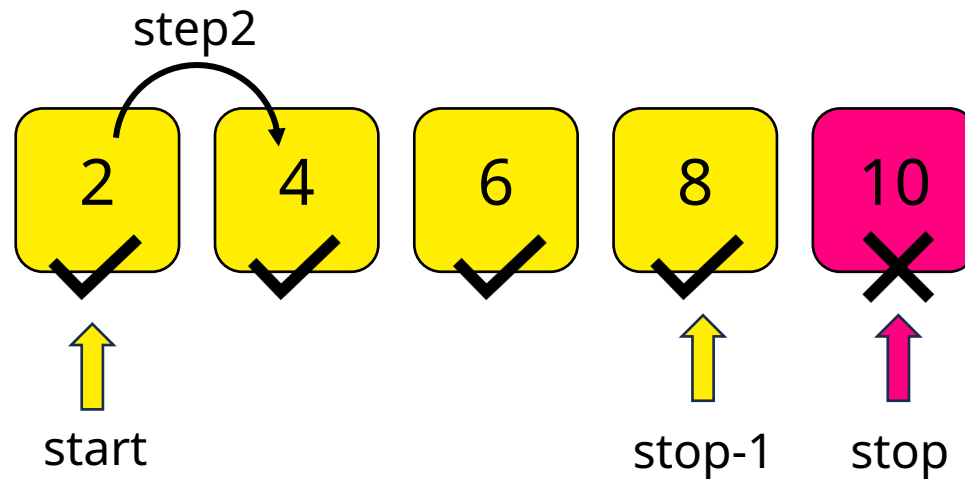
`range (1, 5, 1)`



ฟังก์ชัน range()

3. range(start, stop, step): สร้างชุดตัวเลขตั้งแต่ start ถึง stop - 1 โดยที่ลำดับของตัวเลขมีการเพิ่มขึ้นทีละ step

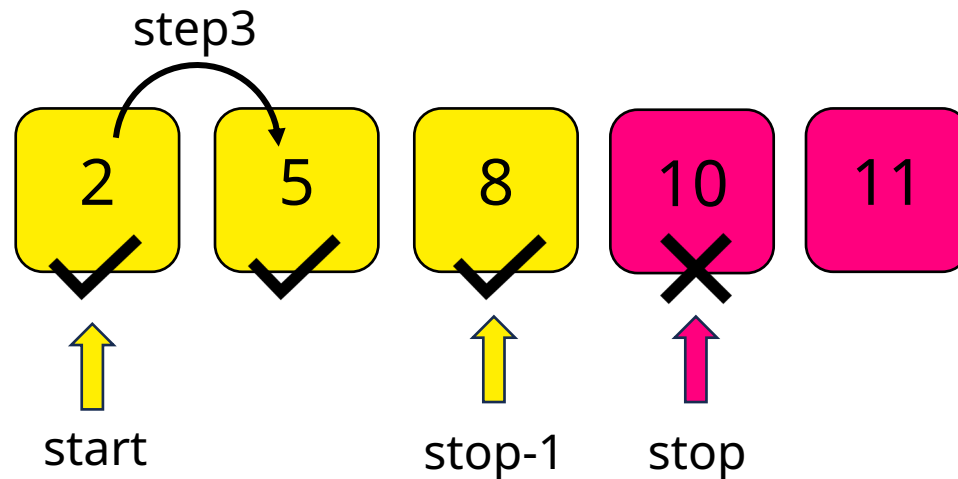
range (2, 10, 2)



ฟังก์ชัน range()

3. range(start, stop, step): สร้างชุดตัวเลขตั้งแต่ start ถึง stop - 1 โดยที่ลำดับของตัวเลขมีการเพิ่มขึ้นทีละ step

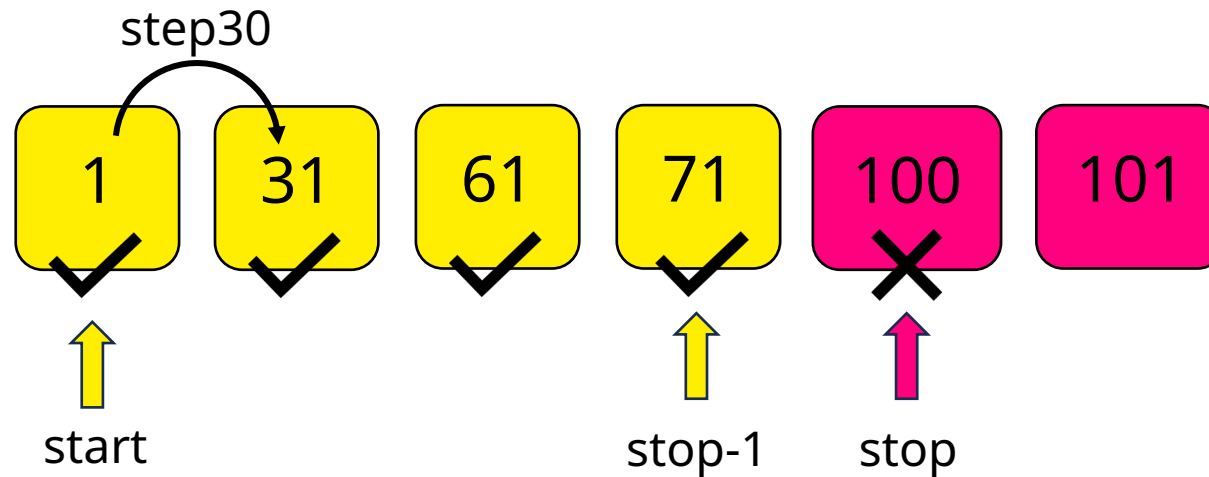
range (2, 10, 3)



ฟังก์ชัน range()

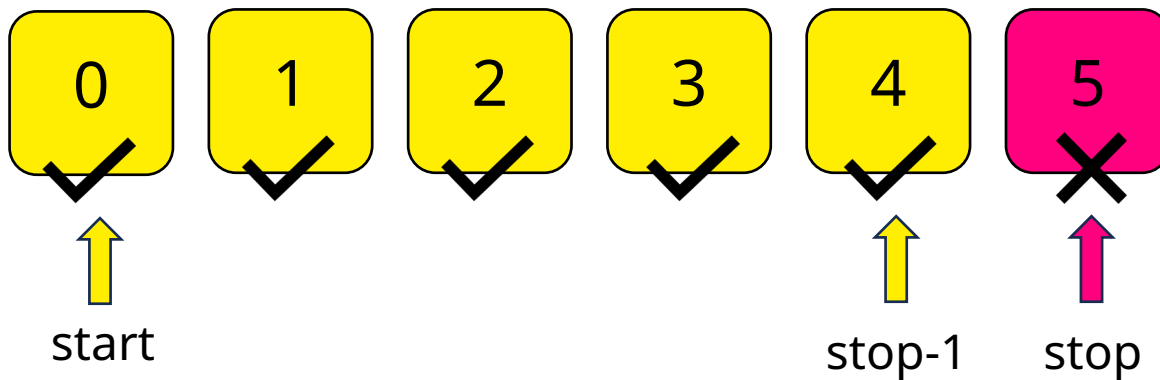
3. range(start, stop, step): สร้างชุดตัวเลขตั้งแต่ start ถึง stop - 1 โดยที่ลำดับของตัวเลขมีการเพิ่มขึ้นทีละ step

`range(1, 100, 30)`



Iteration Statement – for

1. ต้องการแสดงค่าข้อมูลใน range() จาก 0 ถึง 4



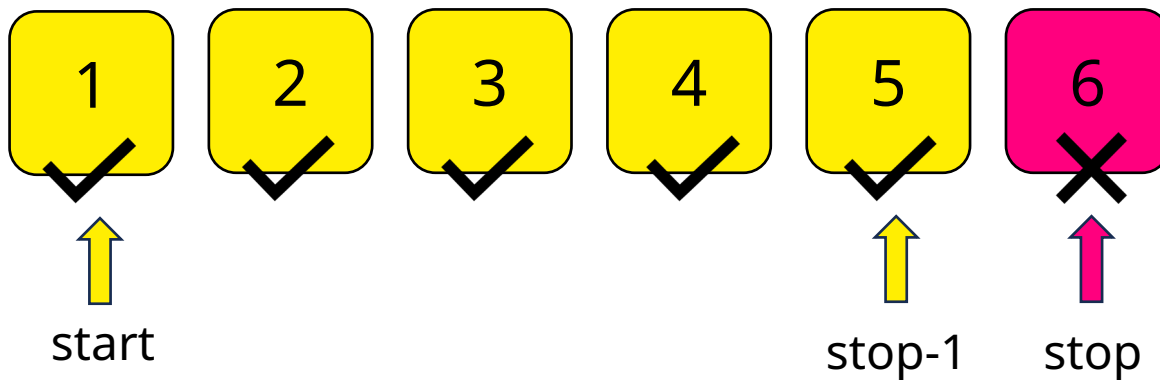
```
for i in range(5):  
    print(i)
```

```
1  for i in range(5):  
2      print(i)
```

```
0  
1  
2  
3  
4
```

Iteration Statement – for

2. ต้องการแสดงค่าข้อมูลใน range() จาก 1 ถึง 5



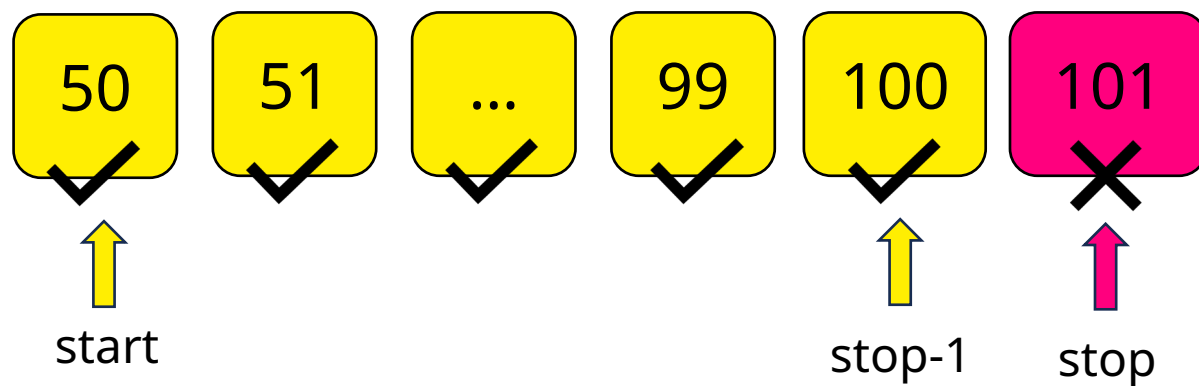
```
for i in range(1,6):  
    print(i)
```

```
1  for i in range(1,6):  
2  print(i)
```

```
1  
2  
3  
4  
5
```

Iteration Statement – for

3. ต้องการแสดงค่าข้อมูลใน range() จาก 50 ถึง 100



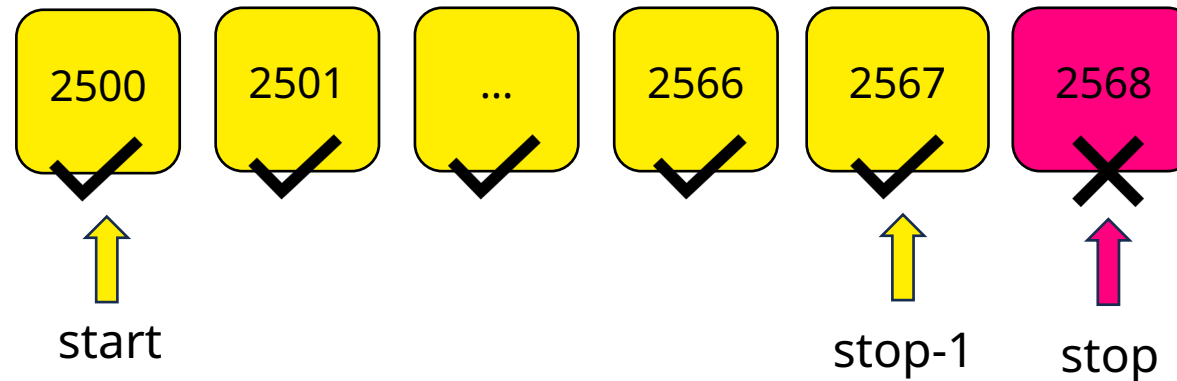
```
for i in range(50, ____):  
    print(i)
```

50	80
51	81
52	82
53	83
54	84
55	85
56	86
57	87
58	88
59	89
60	90
61	91
62	92
63	93
64	94
65	95
66	96
67	97
68	98
69	99
	100

Iteration Statement – for

4. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ปี พ.ศ.2500 ถึง พ.ศ.2567

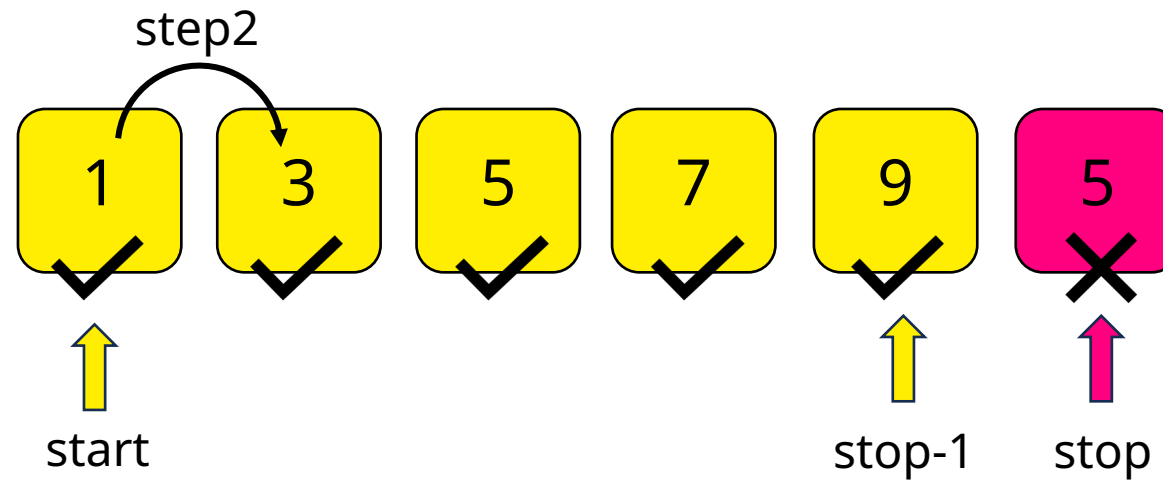
```
for i in range(_____, _____) :  
    print(i)
```



Iteration Statement – for

5. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 1 ถึง 10 เพิ่มทีละ 2

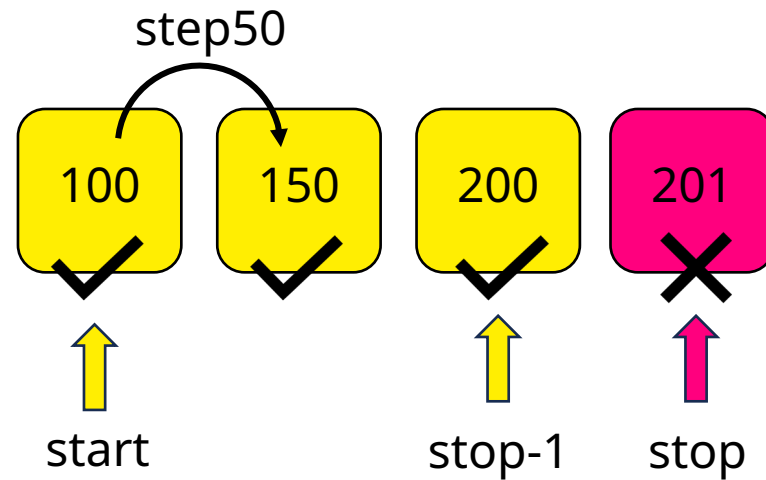
```
for i in range(_____, _____, _____):  
    print(i)
```



Iteration Statement – for

6. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 100 ถึง 200 เพิ่มขึ้น 50

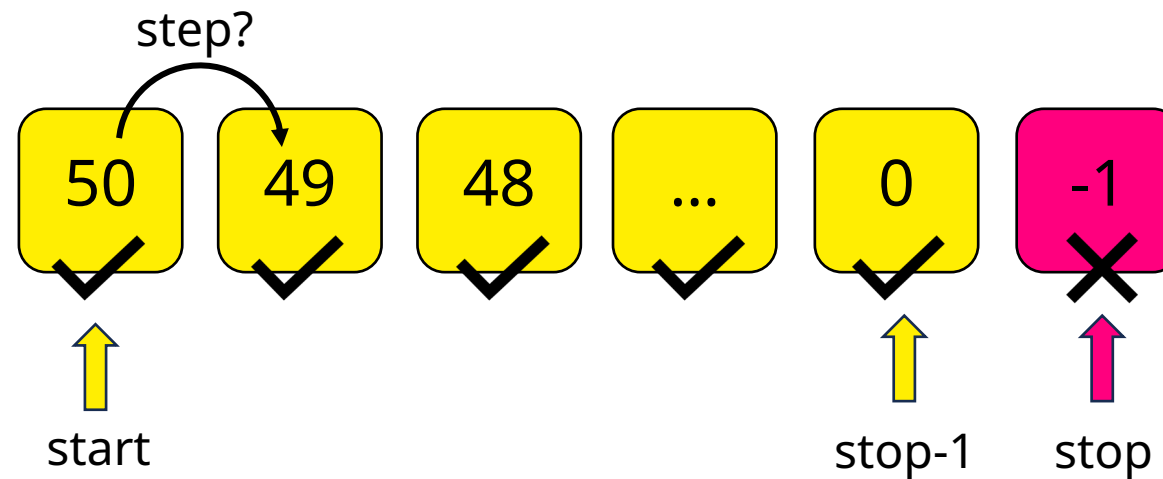
```
for i in range(_____, _____, _____):  
    print(i)
```



Iteration Statement – for

7. การโปรแกรมเพื่อแสดงผลจำนวนตั้งแต่ 50, 49, 48, ... 0

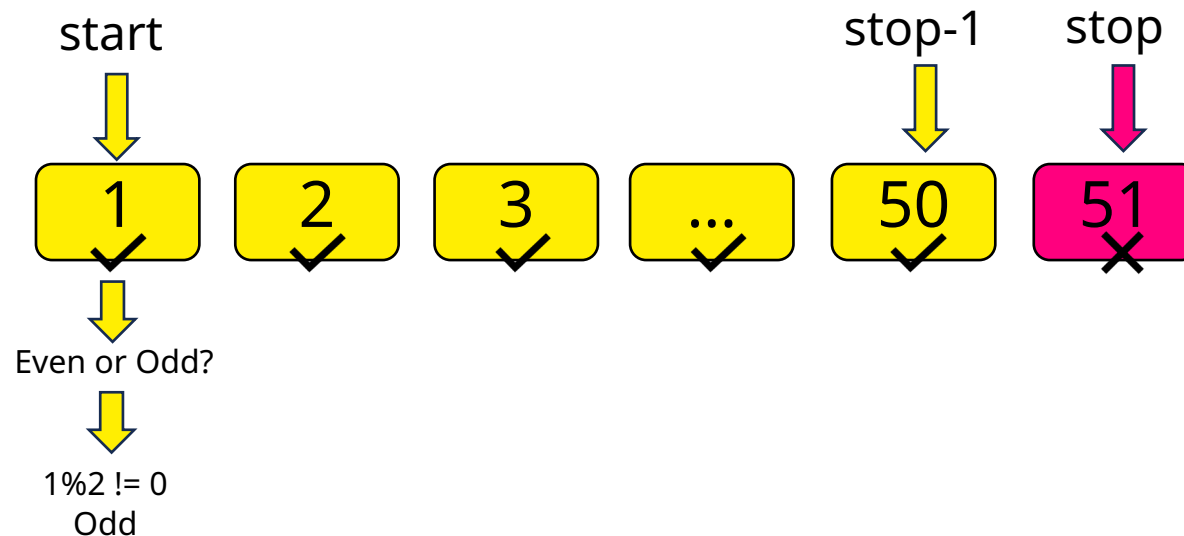
```
for i in range(_____, _____, _____) :  
    print(i)
```



Iteration Statement – for

8. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50

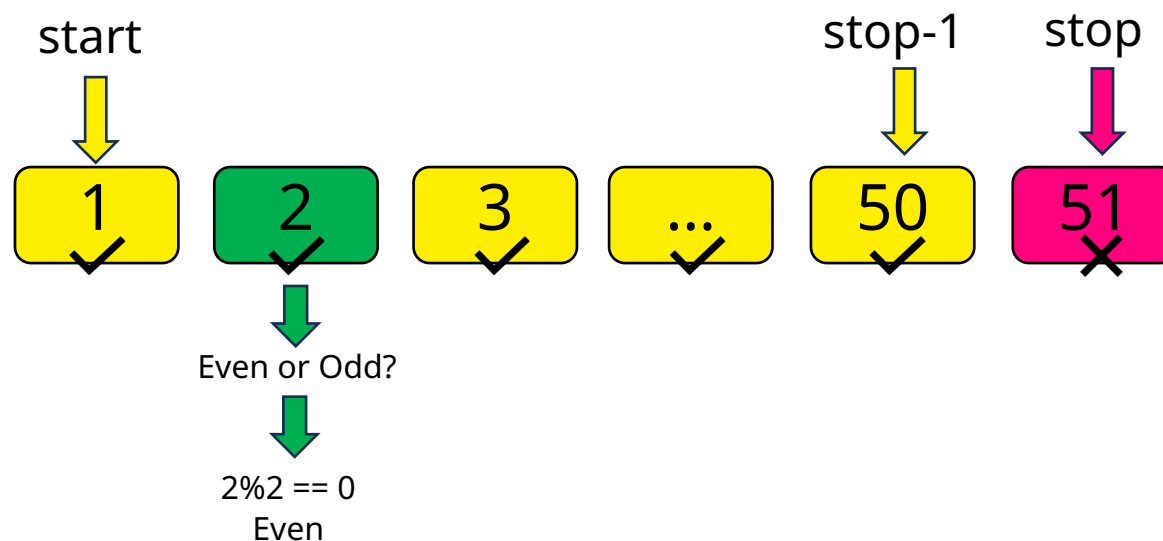
```
for i in range(_____, _____):  
    if _____:  
        print(i)
```



Iteration Statement – for

8. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50

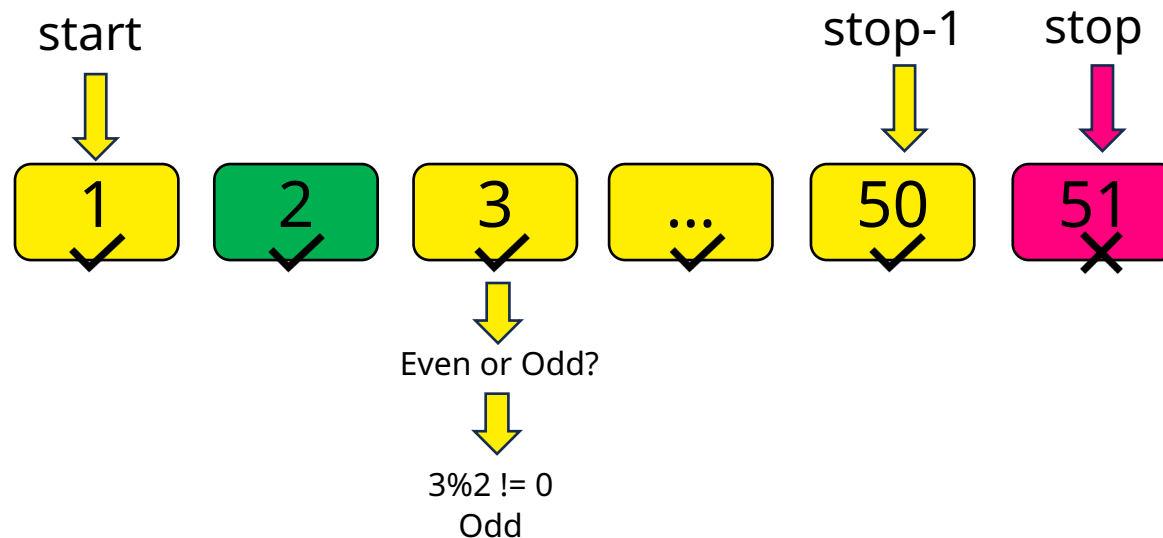
```
for i in range(_____, _____):  
    if _____:  
        print(i)
```



Iteration Statement – for

8. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50

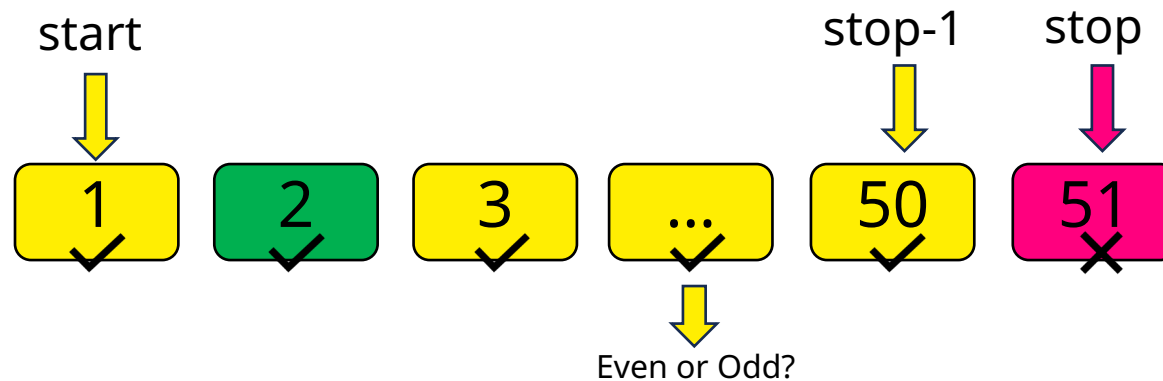
```
for i in range(_____, _____):  
    if _____:  
        print(i)
```



Iteration Statement – for

8. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50

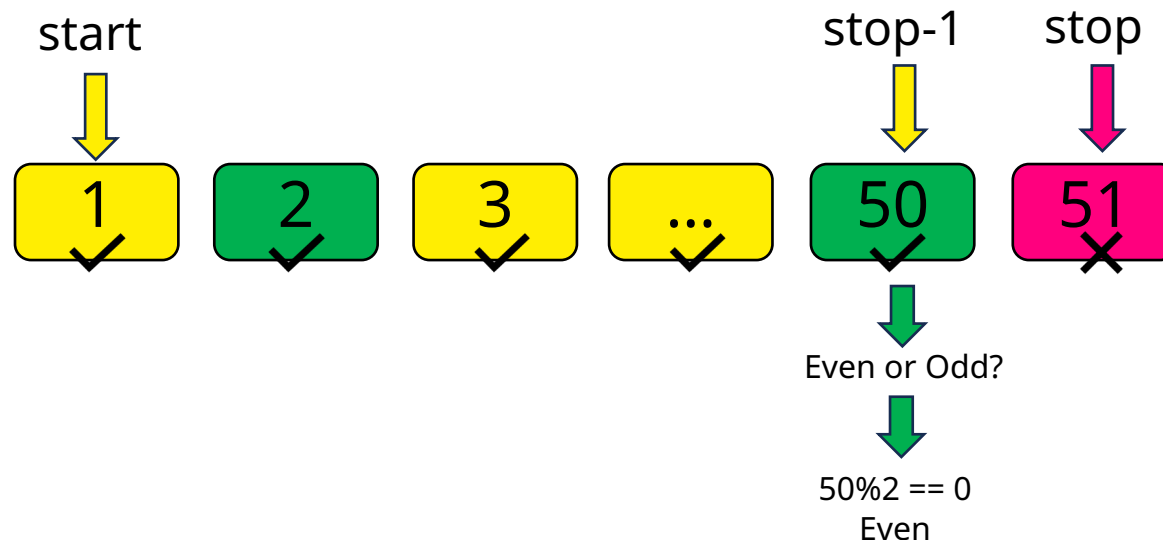
```
for i in range(_____, _____) :  
    if _____ :  
        print(i)
```



Iteration Statement – for

8. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50

```
for i in range(_____, _____) :  
    if _____ :  
        print(i)
```

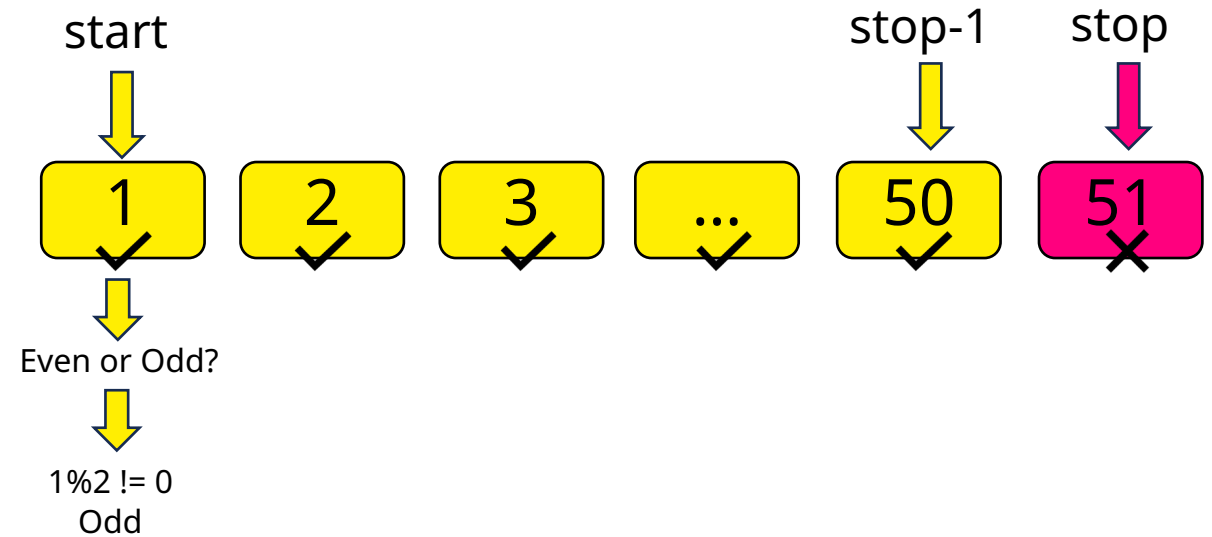


Iteration Statement – for

9. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50 และสรุปว่ามีทั้งหมดกี่จำนวน

```
_____ = _____  
for i in range(1, 50+1):  
    if i%2==0:  
        print(i)  
    _____ = _____
```

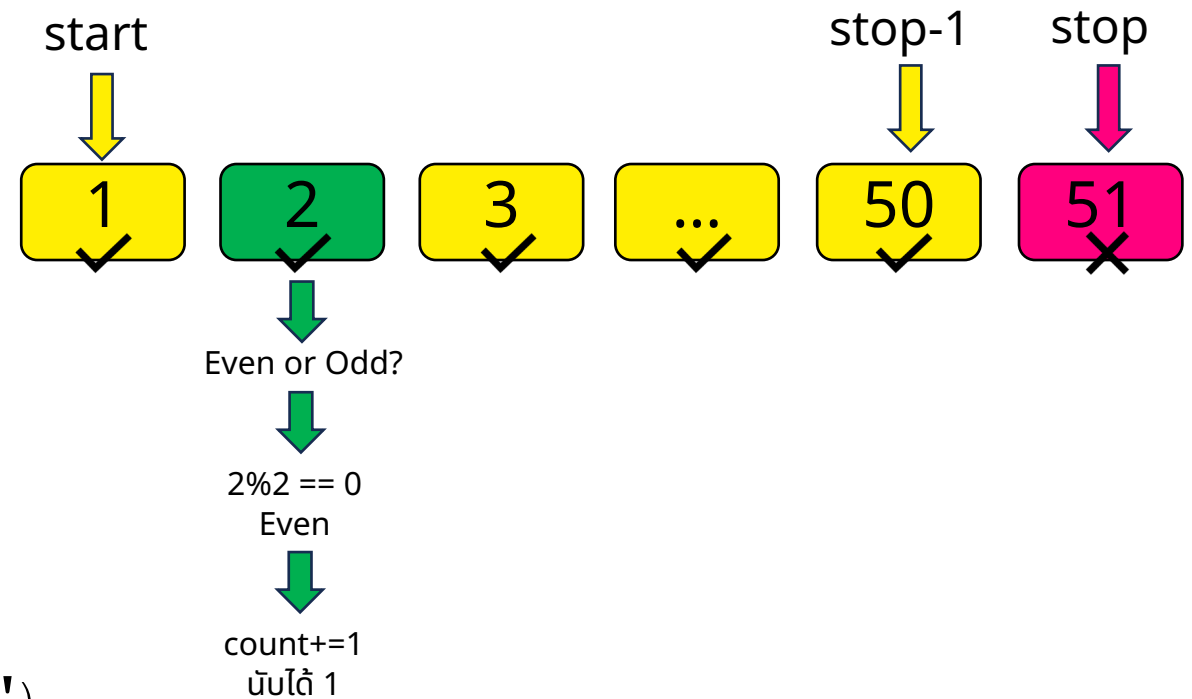
```
print(f"มีจำนวนคู่ทั้งหมด: {count} จำนวน")
```



Iteration Statement – for

9. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50 และสรุปว่ามีทั้งหมดกี่จำนวน

```
_____ = _____  
for i in range(1, 50+1):  
    if i%2==0:  
        print(i)  
    _____ = _____  
  
print(f"มีจำนวนคู่ทั้งหมด: {count} จำนวน")
```



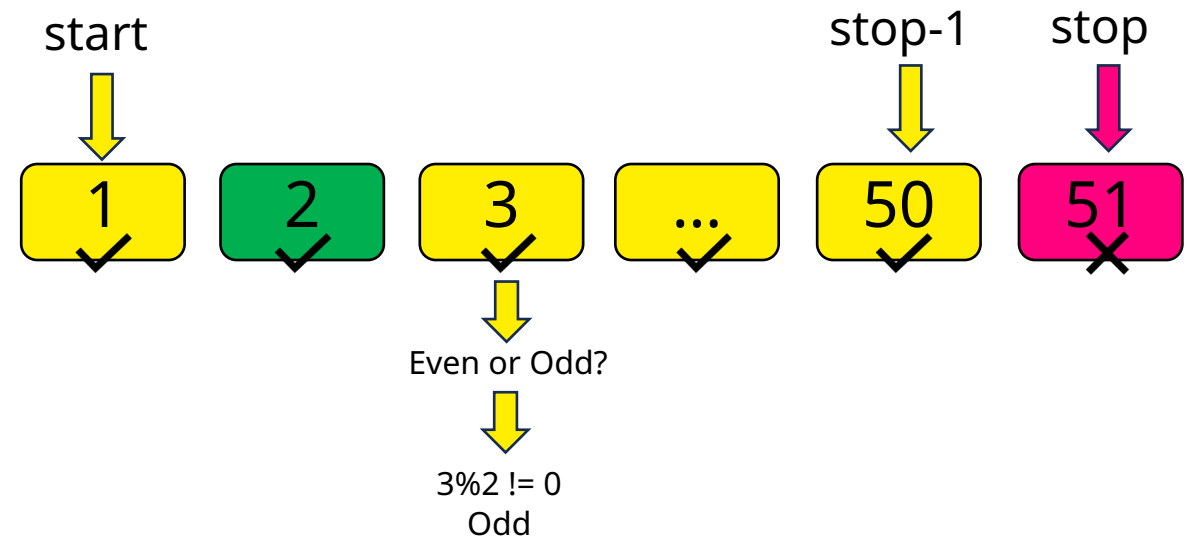
Iteration Statement – for

9. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50 และสรุปว่ามีทั้งหมดกี่จำนวน

```
_____ = _____  
for i in range(1, 50+1):  
    if i%2==0:  
        print(i)
```

```
_____ = _____
```

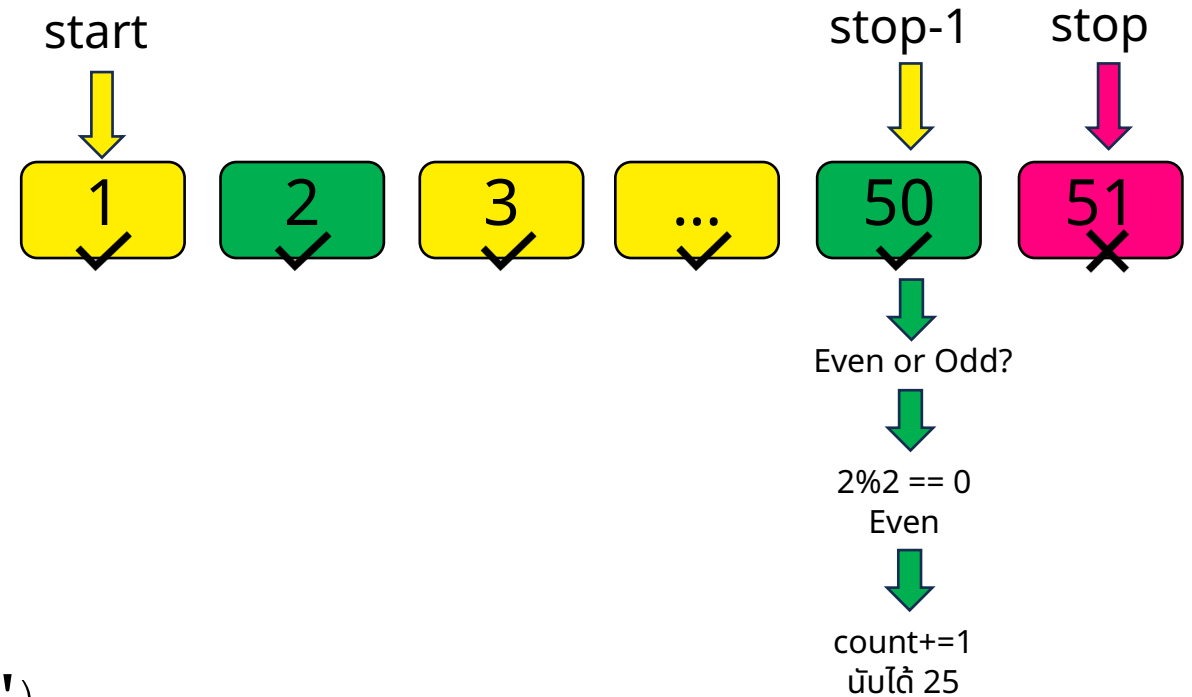
```
print(f"มีจำนวนคู่ทั้งหมด: {count} จำนวน")
```



Iteration Statement – for

9. การโปรแกรมเพื่อแสดงผลจำนวนที่เป็นจำนวนคู่ (Even) ตั้งแต่ 1 ถึง 50 และสรุปว่ามีทั้งหมดกี่จำนวน

```
_____ = _____  
for i in range(1, 50+1):  
    if i%2==0:  
        print(i)  
    _____ = _____  
  
print(f"มีจำนวนคู่ทั้งหมด: {count} จำนวน")
```



Iteration Statement – for

10. การโปรแกรมเพื่อหาผลรวมของจำนวนตั้งแต่ 1, 2, 3, ... ถึง 10

```
summation = 0
for i in range(1,10+1):
    summation = summation+i

print(f"Summation of 1 to 10 : {summation}")
```

```
Summation of 1 to 10 : 55
```

Iteration Statement – for

11. การโปรแกรมเพื่อหาผลรวมของจำนวนตั้งแต่ 2,4,6,..., 18, 20

```
summation = _____  
for i in range(_____, _____, _____):  
    summation = _____  
  
print(f"Summation of 2,4,6,...,18, 20 : {summation}")
```

```
Summation of 2,4,6, ... ,18, 20 : 110
```

Iteration Statement – for

11. การโปรแกรมเพื่อหาผลรวมของจำนวนตั้งแต่ 2,4,6,..., 18, 20

```
summation = 0
for i in range(2, 20+1, 2):
    summation = summation+i

print(f"Summation of 2,4,6,...,18, 20 : {summation}")
```

```
Summation of 2,4,6, ... ,18, 20 : 110
```

Iteration Statement – for

12. การโปรแกรมเพื่อหาค่าเฉลี่ย (average) ของจำนวนตั้งแต่ 1 ถึง 10 โดยค่าเฉลี่ยมาจาก ผลรวมของจำนวนทั้งหมดหารด้วย จำนวนทั้งหมด หรือ

$$\text{average} = \text{summation} / \text{count}$$

summation = _____

_____ = _____

_____ = _____

for i in range(_____, _____):

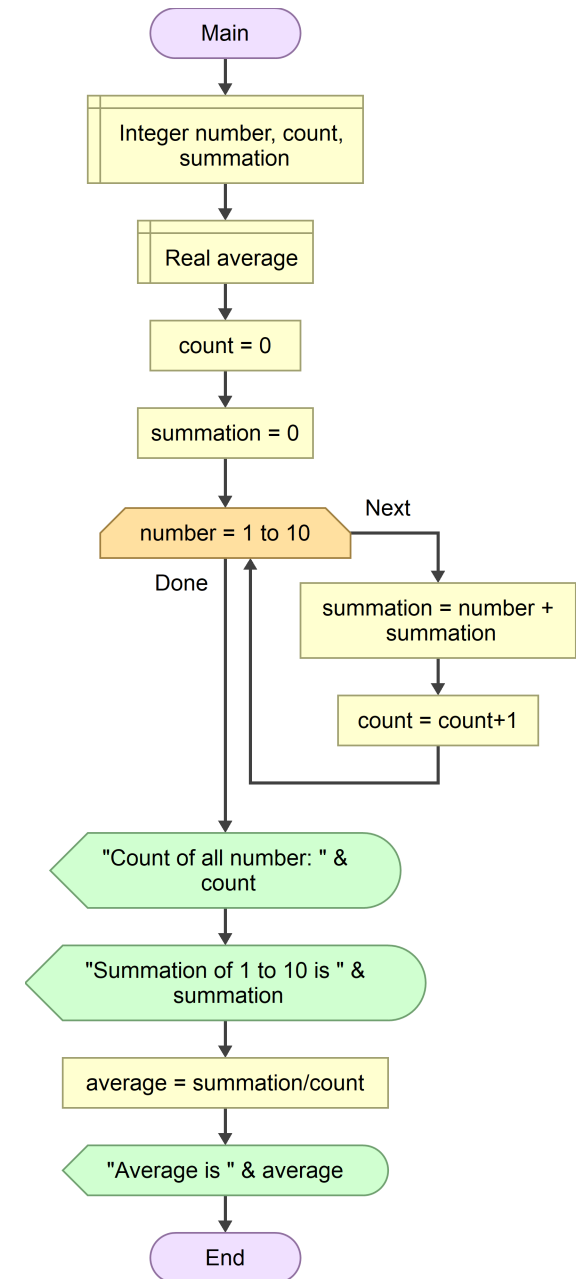
 summation = _____

 count = _____

average = _____

print(f"Average 1 to 10 : {_____}")

การเขียนโปรแกรมคอมพิวเตอร์และพัฒนาแอปพลิเคชันเพื่อการศึกษา
ผู้ช่วยศาสตราจารย์ ดร.ณัฐภัทร แก้วรัตนภัทร์



Iteration Statement – for

12. การโปรแกรมเพื่อหาค่าเฉลี่ย (average) ของจำนวนตั้งแต่ 1 ถึง 10 โดยค่าเฉลี่ยมาจาก ผลรวมของจำนวนทั้งหมดหารด้วย จำนวนทั้งหมด หรือ

$$\text{average} = \text{summation} / \text{count}$$

```
summation = 0
```

```
count = 0
```

```
average = 0
```

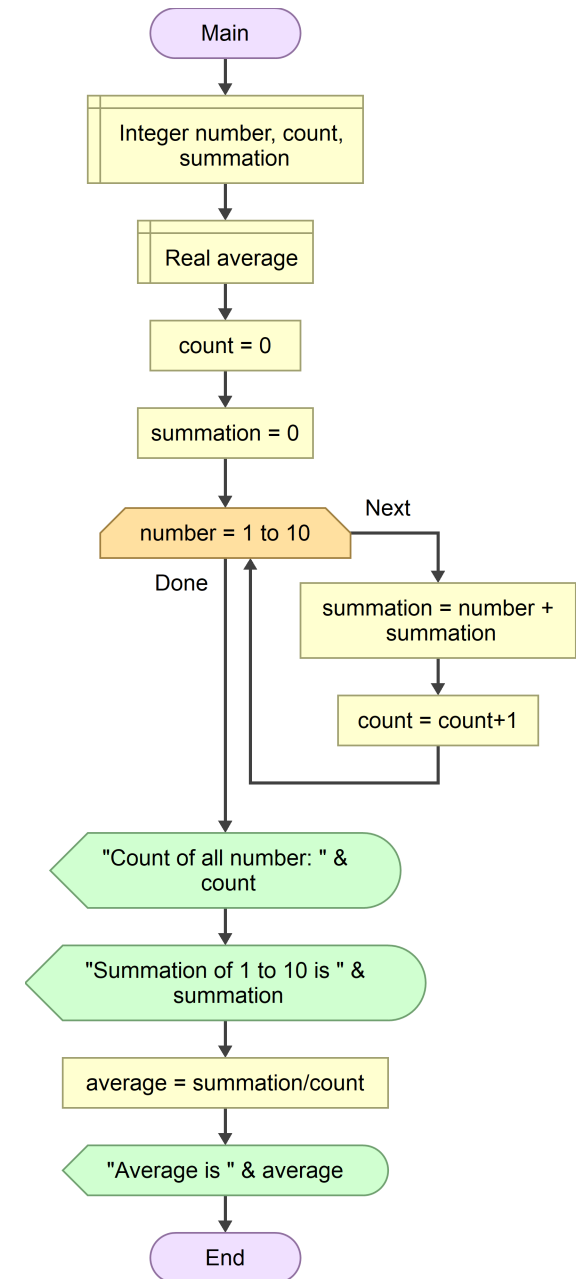
```
for i in range(1,10+1):
```

```
    summation = summation+i
```

```
    count = count+1
```

```
average = summation/count
```

```
print(f"Average 1 to 10 : {average}")
```



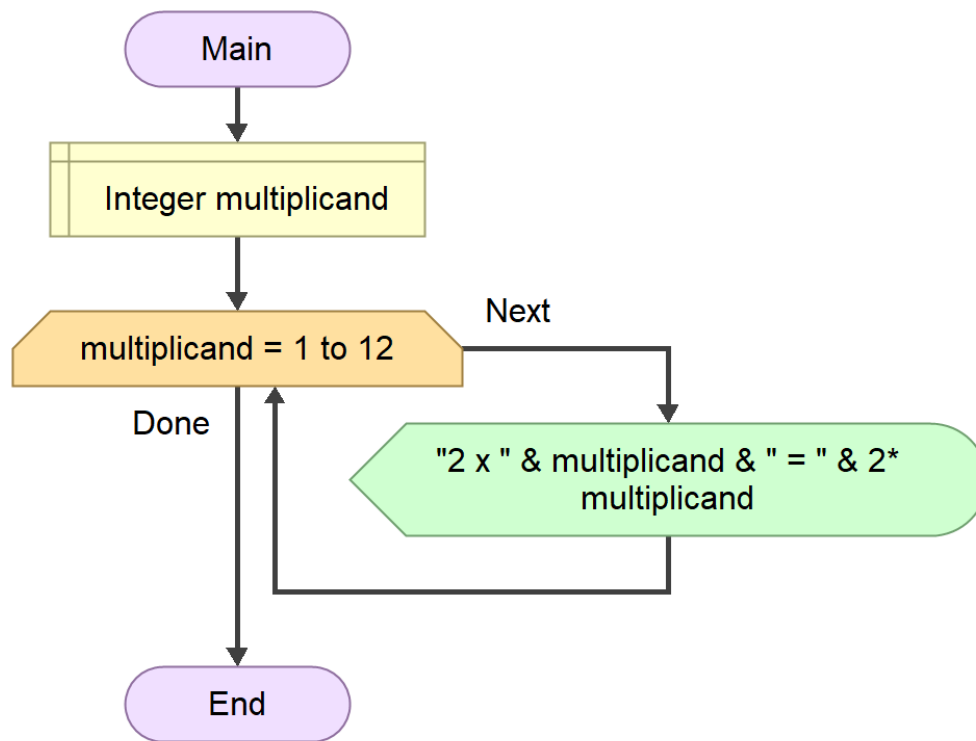
Iteration Statement – for

13. แสดงสูตรคูณแม่ 2 โดยเริ่มตั้งแต่ $2 \times 1 = 2$ ถึง $2 \times 12 = 24$ ด้วย for loop

Multiplier ตัวคูณเสมอ	Multiply การคูณ	Multiplicand ตัวตั้ง		Product ผลการคูณ	
2	*	1	=	2	$2 \times 1 = 2$
2	*	2	=	4	$2 \times 2 = 4$
2	*	3	=	6	$2 \times 3 = 6$
					$2 \times 4 = 8$
					$2 \times 5 = 10$
					$2 \times 6 = 12$
					$2 \times 7 = 14$
					$2 \times 8 = 16$
					$2 \times 9 = 18$
					$2 \times 10 = 20$
					$2 \times 11 = 22$
					$2 \times 12 = 24$

Iteration Statement – for

13. แสดงสูตรคูณแม่ 2 โดยเริ่มตั้งแต่ $2 \times 1 = 2$ ถึง $2 \times 12 = 24$ ด้วย for loop



```
for i in range(1,12+1):  
    print(f"2 x {i} = {2*i}")
```

```
2 x 1 = 2  
2 x 2 = 4  
2 x 3 = 6  
2 x 4 = 8  
2 x 5 = 10  
2 x 6 = 12  
2 x 7 = 14  
2 x 8 = 16  
2 x 9 = 18  
2 x 10 = 20  
2 x 11 = 22  
2 x 12 = 24
```

Iteration Statement – for

สูตรคูณแม่ 2 ถึงแม่ 5

```
for multiplier in range(2,5+1):  
    print(f"สูตรคูณแม่{multiplier}")  
    for multiplicand in range(1,12+1):  
        print(f"{multiplier}x{multiplicand}={multiplier*multiplicand}")  
    print("-----")
```

```
สูตรคูณแม่ 2  
2x1=2  
2x2=4  
2x3=6  
2x4=8  
2x5=10  
2x6=12  
2x7=14  
2x8=16  
2x9=18  
2x10=20  
2x11=22  
2x12=24  
-----  
สูตรคูณแม่ 3  
3x1=3  
3x2=6  
3x3=9  
3x4=12  
3x5=15  
3x6=18  
3x7=21  
3x8=24  
3x9=27  
3x10=30  
3x11=33  
3x12=36  
-----
```

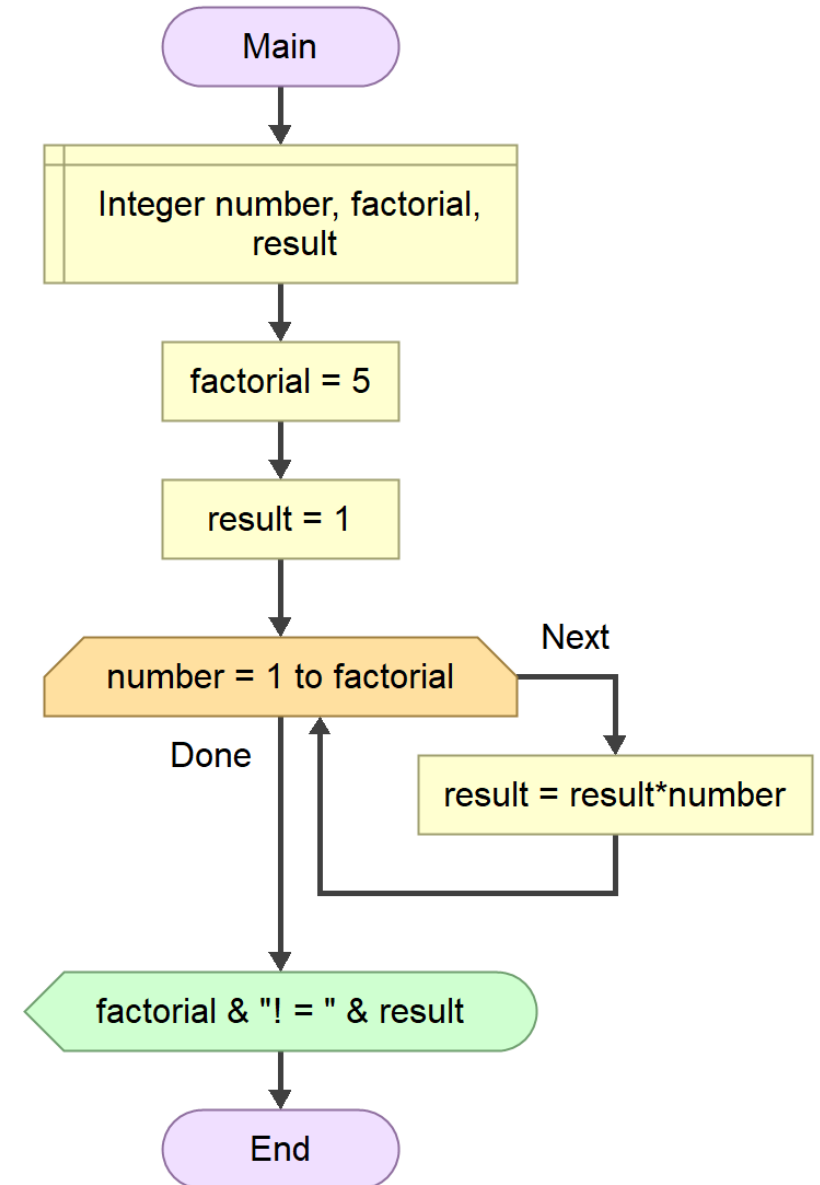
```
สูตรคูณแม่ 4  
4x1=4  
4x2=8  
4x3=12  
4x4=16  
4x5=20  
4x6=24  
4x7=28  
4x8=32  
4x9=36  
4x10=40  
4x11=44  
4x12=48  
-----  
สูตรคูณแม่ 5  
5x1=5  
5x2=10  
5x3=15  
5x4=20  
5x5=25  
5x6=30  
5x7=35  
5x8=40  
5x9=45  
5x10=50  
5x11=55  
5x12=60  
-----
```

Iteration Statement – for

14. การโปรแกรมเพื่อหาค่า Factorial ของ 5 หรือ 5!
Factorial (n!) คือ ผลคูณของจำนวนเต็มบวกตั้งแต่ 1 ถึง n

เช่น

- $1! = 1$
- $2! = 2 \times 1 = 2$
- $3! = 3 \times 2 \times 1 = 6$
- $4! = 4 \times 3 \times 2 \times 1 = 24$
- $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$



Iteration Statement – for

14. การโปรแกรมเพื่อหาค่า Factorial ของ 5 หรือ 5!
Factorial (n!) คือ ผลคูณของจำนวนเต็มบวกตั้งแต่ 1 ถึง n

เช่น

- $1! = 1$
- $2! = 2 \times 1 = 2$
- $3! = 3 \times 2 \times 1 = 6$
- $4! = 4 \times 3 \times 2 \times 1 = 24$
- $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

```
result = 1
for i in range(5, 0, -1):
    result = result*i

print(f"5! = {result}")
```

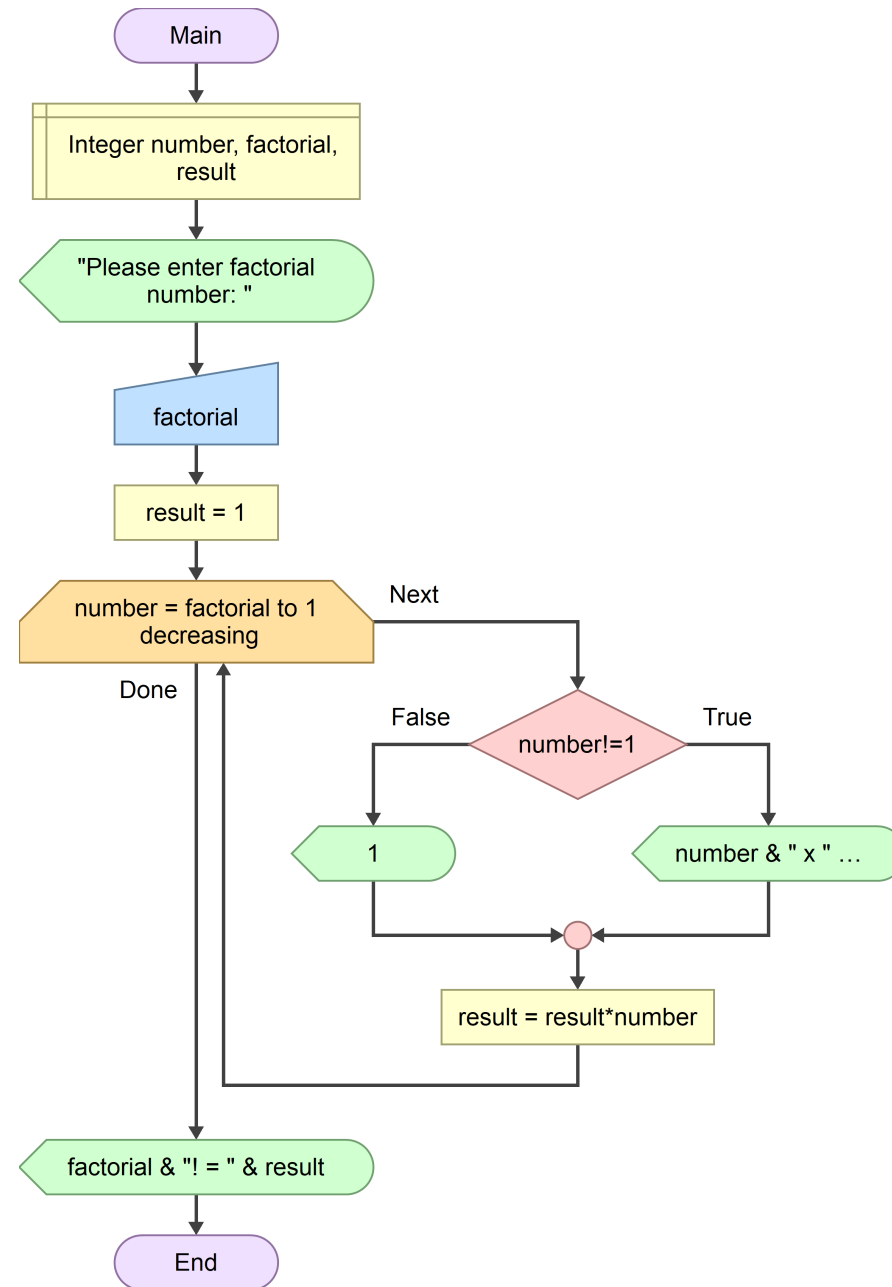
5! = 120

Iteration Statement – for

15. การโปรแกรมเพื่อหาค่า Factorial ของ 5 หรือ 5!

โดยแสดงเป็น

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$



Iteration Statement – for

15. การโปรแกรมเพื่อหาค่า Factorial ของ 5 หรือ 5!

โดยแสดงเป็น

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

```
result = _____  
print("5! = ", end="")  
for i in range(_____, _____, _____):  
    if i!=1:  
        print(f"{_____}x", end="")  
    else:  
        print("_____", end="")  
    result = _____  
  
print(f" = {_____}")
```

Iteration Statement – for

15. การโปรแกรมเพื่อหาค่า Factorial ของ 5 หรือ 5!

โดยแสดงเป็น

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

```
result = 1
print("5! = ", end="")
for i in range(5, 0, -1):
    if i!=1:
        print(f"{i}x", end="")
    else:
        print("1", end="")
    result = result*i

print(f" = {result}")
```

Iteration Statement – for

15. การโปรแกรมเพื่อหาค่า Factorial ของ 5 หรือ 5! โดยแสดงเป็น $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

```
# กำหนดค่าเริ่มต้นของตัวแปร result เพื่อเก็บผลคูณของตัวเลข
result = 1
# แสดงข้อความเริ่มต้นของการแสดงผลลัพธ์
print("5! = ", end="")
# ใช้ลูป for เพื่อคำนวณ Factorial ของ 5 และแสดงผลลัพธ์
for i in range(5, 0, -1):
    if i != 1:
        # แสดงตัวเลขและสัญลักษณ์คูณถ้าไม่ใช่ตัวสุดท้าย
        print(f"{i}x", end="")
    else:
        # แสดงตัวสุดท้ายโดยไม่มีสัญลักษณ์คูณ
        print("1", end="")
    result = result * i # คำนวณผลคูณของตัวเลขทุกตัว
# แสดงผลลัพธ์ของ Factorial
print(f" = {result}")
```

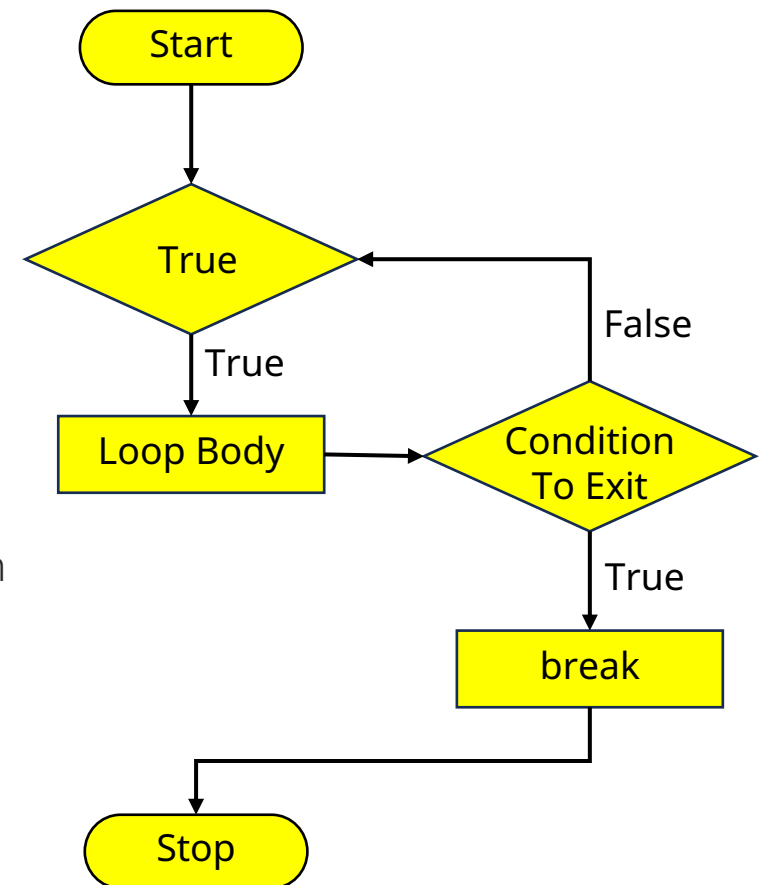
Iteration Statement – do-while

ในภาษา Python ไม่มี Syntax do-while เหมือนกับภาษาอื่นๆ เช่น ภาษา C, C++, Java แต่ก็ยังสามารถประยุกต์ใช้ while loop ในการสร้าง do-while ได้เช่นกัน ดังนี้

while True:

if condition_to_exit:

break # Exit the loop when the condition is match



Iteration Statement – do-while

ต้องการรับรหัสผ่านเข้าประตู (Digital Door Lock) กรณีกรอกผิดให้ทำการถามใหม่อีกครั้ง
ไป จนกว่าจะตอบถูก กรณีตอบถูกขึ้นคำว่า "Correct"

```
while True:
    | userinput = input("Please input password: ")
    | if userinput=="1234":
    |     | print("Correct")
    |     | break
```

```
Please input password: 2233333
Please input password: 22211
Please input password: 4456
Please input password: 65432
Please input password: 199[
Please input password: 1234
Correct
```

Assignment unit 4 Iteration Statement (3 คะแนน)

1. เขียนโปรแกรมเพื่อแสดงจำนวนตั้งแต่ 1 ถึง 10,000 ที่หาร 3 และหาร 5 ลงตัว และสรุปด้วยว่ามีกี่จำนวน
2. เขียนโปรแกรมหาค่าเฉลี่ยของจำนวนตั้งแต่ 500 ถึง 1,001
3. เขียนโปรแกรมสูตรคูณแม่ 2 ถึง สูตรคูณแม่ 12 ในการรันโปรแกรมครั้งเดียว
4. เขียนโปรแกรมรับค่ารหัสผ่าน
หากรหัสผ่านไม่ตรงตามที่กำหนดไว้ ให้แจ้งว่า **รหัสผ่านผิด** และวนรับค่ารหัสผ่านใหม่
หากรหัสผ่านตรงตามที่กำหนดไว้ ให้แจ้งว่า **รหัสผ่านถูกต้อง** และหยุดรับรหัสผ่าน