

Python

for New Gen New Game

Day 4

While

```
1  i = 0
2  while i < 5: ————— ทำคำสั่งซ้ำตราบเท่าที่เงื่อนไขเป็นจริง
3      print(i)
4      i += 1
5  print("End")
6
```

Challenges

<i>โจทย์</i>	<i>ตัวอย่างอินพุต</i>	<i>ตัวอย่างเอาต์พุต</i>
รับค่าตัวเลขจำนวนเต็มบวก 1 ตัว แสดงเลข 1 ถึงจำนวนนั้นๆ	3	1 2 3
รับค่าตัวเลข 1 ตัว แสดงเครื่องหมายดอกจันตามจำนวนตัวเลขที่ได้รับ	2	**
	10	*****
รับค่าตัวเลขจำนวนเต็มบวก 2 ตัว แสดงจำนวนที่น้อยที่สุดที่หารตัวเลขตัวแรกลงตัว เป็นจำนวนตัวที่สอง (ดูตัวอย่าง)	3	3
	2	6
	5	5
	3	10 15

If in while

```
1  i = 0
2  while i < 5:
3      print(i)
4      if i == 3:
5          print("Three")
6      i += 1
7  print("End")
8
```

Debug challenge

นี่คือโปรแกรมที่จะแสดงเฉพาะเลขคี่ตั้งแต่
1 - 21

```
1  i = 1
2  while i < 21:
3      i += 1
4      if i % 2 == 0:
5          print(i)
6
```

Debug challenge

นี่คือโปรแกรมที่จะรับตัวเลขไปเรื่อยๆ จนกว่าจะได้รับค่า 10 แล้วจึงออกจากโปรแกรม

```
1  number = input("Enter number: ")
2  while number == 10:
3      number = input("Enter number: ")
4  print("Program will now exit.")
5
```

Challenges

<i>โจทย์</i>	<i>ตัวอย่างอินพุต</i>	<i>ตัวอย่างเอาต์พุต</i>
รับค่าตัวเลขจำนวนเต็มบวก 1 ตัว แสดงเลขคี่ตั้งแต่ 1 ถึงไม่เกินจำนวนนั้นๆ	6	1 3 5
รับค่าตัวเลข 1 ตัว แสดงเครื่องหมายดอกจันสลับกับขีดกลางตามจำนวนตัวเลขที่ได้รับ	5	*-*-*
	8	*-*-*-*-
รับค่าตัวเลขจำนวนเต็มบวก 2 ตัว แสดงเลขตั้งแต่ 1 ถึงตัวแรก เฉพาะตัวที่สามารถหารด้วยตัวที่ 2 ลงตัว	16 5	5 10 15
	11 4	4 8

Reverse while

```
1  i = 4
2  while i >= 0:————— ทำคำสั่งซ้ำตราบเท่าที่เงื่อนไขเป็นจริง
3      print(i)
4      i -= 1
5  print("End")
6
```


Challenges

<i>โจทย์</i>	<i>ตัวอย่างอินพุต</i>	<i>ตัวอย่างเอาต์พุต</i>
รับค่าตัวเลขจำนวนเต็มบวก 1 ตัว แสดงเลขจำนวนนั้นๆ ถึง 1	3	3 2 1
รับค่าตัวเลข 2 ตัว โดยตัวแรกน้อยกว่าตัวที่สองเสมอ แสดงเลขคู่ที่อยู่ระหว่างเลข 2 ตัวนี้จากมากไปน้อย (รวมเลข 2 ตัวนี้ด้วย)	13 20	20 18 16 14
รับค่าตัวเลข 3 ตัว โดยตัวแรกน้อยกว่าตัวที่สองเสมอ แสดงเลขที่อยู่ระหว่างเลข 2 ตัวนี้จากมากไปน้อย เฉพาะตัวที่หารตัวที่สามลงตัว (ไม่รวมเลข 2 ตัวแรก)	11 21 3	18 15 12

Break

```
1  i = 0
2  while i < 5:
3      print(i)
4      if i == 3:
5          break ————— หยุดทำซ้ำ
6          i += 1
7  print("End")
8
```

Continue

```
1  i = 0
2  while i < 5:
3      print(i)
4      if i == 3:
5          continue ————ข้ามไปรอบถัดไป
6          i += 1
7  print("End")
8
```

Print without new-line

```
1  i = 0
2  while i < 5:
3      print(i, end=" ") ————— เปลี่ยนจากขึ้นบรรทัดใหม่เป็นเว้นวรรค
4      i += 1
5  print("End")
6
```

Challenges

โจทย์	ตัวอย่างอินพุต	ตัวอย่างเอาต์พุต
รับค่าตัวเลขจำนวนเต็มบวก แสดงค่าตั้งแต่ 1 ถึงจำนวนนั้นๆ ที่เป็นเลขคู่ คั่นด้วยเว้นวรรค	8	2 4 6 8
	12	2 4 6 8 10 12
รับเลขจำนวนเต็ม 5 ตัว ให้คะแนนตามผลรวมคะแนนดังนี้ 40-50 ได้เกรด A, 35-39 ได้เกรด B, 30-34 ได้เกรด C, 25-29 ได้เกรด D, 0-24 ได้เกรด F หากผลรวมตัวเลขน้อยกว่า 0 หรือมากกว่า 50 หรือมีเลขโดดเลขหนึ่งติดลบให้ขึ้นว่า <i>Error</i>	2 9 10 12 -8	Error

Expert Challenges

<i>โจทย์</i>	<i>ตัวอย่างอินพุต</i>	<i>ตัวอย่างเอาต์พุต</i>
รับค่าตัวเลขจำนวนเต็มบวก 1 ตัว แสดงค่าตัวเลขทั้งหมดที่สามารถหารจำนวนนั้นๆ ได้ลงตัว ยกเว้นเลข 1 และตัวมันเอง โดยเรียงลำดับจากน้อยไปมาก	12	2 3 4 6
	20	2 4 5 10
	32	2 4 8 16
รับค่าตัวเลขจำนวนเต็มบวก 1 ตัว แสดงตัวประกอบทั้งหมดออกมา โดยเรียงลำดับจากน้อยไปมาก	12	2 2 3
	20	2 2 5
	32	2 2 2 2 2

List declaration

```
1  fruits = ["Apple", "Orange", "Mango"]
2  print(fruits)
3
```

List reference by index

```
1  fruits = ["Apple", "Orange", "Mango"]
2  print(fruits[0])
3  print(fruits[1])
4  print(fruits[2])
5  fruits[2] = "Banana"
6  print(fruits[2])
7
```


length = 5

index

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

List length

```
1  fruits = ["Apple", "Orange", "Mango"]
2  print(len(fruits))
3
```

Add new item to the list

```
1  fruits = ["Apple", "Orange", "Mango"]
2  fruits.append("Banana")
3  print(fruits)
4
```

Debug challenge

นี่คือโปรแกรมที่รับข้อความไปเรื่อยๆ จนกว่าจะได้รับข้อความว่า "exit" แล้วจึงหยุด และแสดงข้อความทั้งหมดที่เคยรับไปแล้ว

```
1 list = []
2 while True:
3     newItem = input("Enter: ")
4     if (newItem != "exit"):
5         break
6     list.append(newItem)
7 print(newItem)
8
```

Challenges

<i>โจทย์</i>	<i>ตัวอย่างอินพุต</i>	<i>ตัวอย่างเอาต์พุต</i>
รับข้อความไปเรื่อยๆ จนกว่าจะพบข้อความ "exit" แล้วจึงแสดงจำนวนของข้อความที่ได้รับมาก่อนหน้า	this is example exit	3
รับข้อความไปเรื่อยๆ จนกว่าจะพบข้อความ "exit" แล้วจึงรับตัวเลขจำนวนเต็มบวก 1 ตัว แสดง ข้อความในลำดับนั้นๆ โดยตัวแรกนับเป็น 1 หรือ แสดงข้อความว่า "Error" หากตัวเลขลำดับมากกว่า จำนวนข้อความที่ได้รับก่อนหน้านี้	this is longer example exit 4	example
	hello world exit 5	Error