

Programming Fundamentals I Lab.

2. คำสั่งควบคุม

ชื่อ _____ รหัสนิสิต _____

ในปฏิบัติการนี้ คุณจะรู้จักคำสั่งควบคุมสำหรับการเขียนโปรแกรมโดยใช้คำสั่งควบคุม ได้แก่ การเขียนโปรแกรมแบบมีเงื่อนไข และวงวน

2.1 เงื่อนไขและตรรกศาสตร์

สร้างไฟล์ .py ใหม่ และพิมพ์โค้ดโปรแกรมต่อไปนี้

```
n = input("Enter a number:")
if n < 10:
    print("The number is less than 10.")
print("Thank you.")
print("Good bye.")
```

(คำเตือน: หัวบรรทัดที่สาม ใช้การกดปุ่ม `tab` บนคีย์บอร์ด)

หลังจากเซฟและทดลองรันโปรแกรม ให้ทดลองพิมพ์จำนวนเต็มอะไรก็ได้ และกดปุ่ม `enter` จะเกิดข้อผิดพลาดขึ้น ให้คุณบันทึกข้อผิดพลาดในช่องด้านล่าง

ให้คุณลองหาทางแก้ไขข้อผิดพลาดในโปรแกรมโดย **แก้ไขเฉพาะบรรทัดแรกเท่านั้น** จนสามารถทำงานได้ บันทึกโค้ดบรรทัดแรกหลังจากแก้ไขเสร็จแล้วในช่องด้านล่าง

หลังจากแก้ไขโปรแกรมแล้วเมื่อรันโปรแกรมและป้อนจำนวนเต็ม 5 และกดปุ่ม **enter** บันทึกผลลัพธ์ที่เกิดขึ้น

ทดลองรันโปรแกรมใหม่ ป้อนจำนวนเต็ม 10 เป็นข้อมูลนำเข้า บันทึกผลลัพธ์ที่เกิดขึ้น

ทดลองแก้ไขโค้ดตั้งแต่บรรทัดที่สองเป็นต้นไปให้เป็นดังนี้

```
if n < 10:
    print("The number is less than 10.")
    print("Thank you.")
print("Good bye.")
```

(คำเตือน: ใช้เครื่องหมาย **tab** เช่นเดียวกัน)

ทดลองรันโปรแกรมโดยป้อนจำนวนเต็ม 10 เป็นข้อมูลนำเข้า บันทึกผลลัพธ์ที่เกิดขึ้น

คุณคิดว่าเครื่องหมาย **tab** มีหน้าที่อย่างไรในภาษา Python

จงทดลองรันโปรแกรมดังกล่าวด้วยข้อมูลนำเข้าเป็นจำนวนเต็มต่าง ๆ และอธิบายว่าข้อมูลนำเข้าต้องมีคุณสมบัติเช่นไร โปรแกรมถึงแสดงข้อความ **The number is less than 10.**

จงเปลี่ยนเครื่องหมาย < ในบรรทัดที่สองเป็นเครื่องหมายต่อไปนี้ ทดลองด้วยข้อมูลนำเข้าต่าง ๆ และบันทึกว่าแต่ละเครื่องหมายมีความหมายอย่างไร

<	
<=	
>	
>=	
==	
!=	

จงแก้ไขโค้ดโปรแกรมในไฟล์ .py เป็นโค้ดต่อไปนี้

```
x = int(input('Enter 1st number:'))
y = int(input('Enter 2nd number:'))
if x<10 or y<10:
    print('The numbers are OK.')
print('Good bye.')
```

ทดลองรันโปรแกรม ป้อนข้อมูลนำเข้าเป็นจำนวนเต็มสองค่า สังเกตผลลัพธ์ หลังจากนั้นให้ทดลองด้วยข้อมูลนำเข้าเป็นจำนวนเต็มที่แตกต่างกันหลาย ๆ คู่ พิจารณาผลลัพธ์ที่เกิดขึ้น

ค่าของตัวแปร x และ y ต้องเป็นอย่างไรโปรแกรมจึงแสดงข้อความ **The numbers are OK.**

เปลี่ยนโค้ดบรรทัดที่สาม จาก **or** เป็น **and** ทดลองกับข้อมูลนำเข้าหลาย ๆ คู่ พิจารณาผลลัพธ์ที่เกิดขึ้น ค่าของตัวแปร x และ y ต้องเป็นอย่างไรโปรแกรมจึงแสดงข้อความ **The numbers are OK.**

ถึงตรงนี้ เราจะมาทำความเข้าใจการตรวจสอบเงื่อนไขมากขึ้น ให้คุณย้ายมาที่หน้าต่าง Python Shell ซึ่งมีเครื่องหมาย >>> ค้างอยู่ ทดลองพิมพ์คำสั่งต่อไปนี้ และบันทึกผลที่เกิดขึ้น

<code>2 < 3</code>	
<code>not (2 <= 3)</code>	
<code>4 > 10 or 4 < 5</code>	
<code>3 != 10 or 4 >= 5 and 10 == 100</code>	
<code>type(2 < 3)</code>	
<code>not (3 != 10) or (4 >= 5)</code>	
<code>type(not (3 != 10) or (4 >= 5))</code>	
<code>'AA' < 'AB'</code>	
<code>'AAA' < 'AAAAA'</code>	
<code>'B' < 'AAA'</code>	
<code>'BB' < 'b'</code>	
<code>'Aab' < 10</code>	

จากการทดลอง ค่าของผลลัพธ์ของการเปรียบเทียบจำนวน เป็นชนิดข้อมูลที่ชื่อว่าอะไร?

ชนิดข้อมูลดังกล่าว มีค่าที่ค่า ได้แก่อะไรบ้าง?

จงเรียงลำดับความสำคัญระหว่างตัวดำเนินการ `and`, `or`, และ `not`

การเปรียบเทียบ มากกว่า/น้อยกว่า ระหว่างสตริง มีหลักการอย่างไร?

2.2 การเขียนโปรแกรมแบบมีเงื่อนไข

แก้ไขไฟล์ .py เป็นดังนี้

```
age = int(input('Enter your age:'))
if age >= 20:
    print('You are old.')
print('Good bye')
```

หากรันโปรแกรมโดยป้อน 20 เป็นข้อมูลนำเข้า จะได้ผลลัพธ์อย่างไร?

หากรันโปรแกรมโดยป้อน 18 เป็นข้อมูลนำเข้า จะได้ผลลัพธ์อย่างไร?

แก้ไขโค้ดโปรแกรมเป็นดังนี้

```
age = int(input('Enter your age:'))
if age >= 20:
    print('You are old.')
else:
    print('You are young.')
print('Good bye.')
```

หากรันโปรแกรมโดยป้อน 20 เป็นข้อมูลนำเข้า จะได้ผลลัพธ์อย่างไร?

หากรันโปรแกรมโดยป้อน 18 เป็นข้อมูลนำเข้า จะได้ผลลัพธ์อย่างไร?

คำสั่ง `if...else...` มีการทำงานอย่างไร

แก้ไขโค้ดโปรแกรมเป็นดังนี้

```
age = int(input('Enter your age:'))
if age <= 10:
    print('You are young.')
else:
    if age < 20:
        print('You are a teenager.')
    else:
        print('You are old.')
print('Good bye.')
```

ทดลองรันโปรแกรมด้วยข้อมูลนำเข้าเป็นจำนวนเต็มค่าต่าง ๆ

ข้อมูลนำเข้าต้องมีค่าเท่าใดโปรแกรมจึงแสดงข้อความ `You are young.`

ข้อมูลนำเข้าต้องมีค่าเท่าใดโปรแกรมจึงแสดงข้อความ You are a teenager.

ข้อมูลนำเข้าต้องมีค่าเท่าใดโปรแกรมจึงแสดงข้อความ You are old.

จงอธิบายกระบวนการคิดของโปรแกรมโดยละเอียด

แก้ไขโค้ดโปรแกรมเป็นดังนี้

```
age = int(input('Enter your age:'))
if age <= 10:
    print('You are young.')
elif age < 20:
    print('You are a teenager.')
else:
    print('You are old.')
print('Good bye.')
```

ทดลองรันโปรแกรมด้วยข้อมูลนำเข้าต่าง ๆ วิเคราะห์ว่าการทำงานของโปรแกรมเหมือนหรือแตกต่างจากโปรแกรมที่แล้วอย่างไร

2.3 การเขียนโปรแกรมแบบวงวน

แก้ไขโค้ดโปรแกรมเป็นดังนี้

```
n = int(input('Please enter a number:'))
while n > 0:
    print('Too much.')
    n = int(input('Please enter a number again:'))
print('Program terminated.')
```

ทดลองรันโปรแกรม ป้อนข้อมูลนำเข้าเป็นจำนวนเต็ม 15 และกด **enter** บันทึกผลที่เกิดขึ้น

โปรแกรมสิ้นสุดการทำงานหรือยัง? สังเกตจากอะไร?

ทดลองป้อนข้อมูลนำเข้าไปเรื่อย ๆ จนกว่าโปรแกรมจะแสดงข้อความ **Program terminated.** ข้อมูลนำเข้าต้องมีคุณสมบัติอย่างไรโปรแกรมจึงสิ้นสุดการทำงาน

แก้ไขโปรแกรมเป็นดังนี้

```
n = int(input('Please enter a number:'))
while n > 0:
```



```
print('Too much.')
print('Program terminated.')
```

ทดลองรันโดยป้อนข้อมูลนำเข้าเป็นเลข -10 บันทึกผลที่ได้

ทดลองรันอีกครั้งโดยป้อนข้อมูลนำเข้าเป็นเลข 19 บันทึกผลที่ได้ (หากโปรแกรมทำงานไม่หยุด คุณสามารถสั่งหยุดได้โดยกด Control+C)

คุณคิดว่าเหตุใดโปรแกรมจึงทำงานไม่สิ้นสุด

จากการทดลอง คำสั่ง **while** มีการทำงานอย่างไร?

แก้ไขโค้ดโปรแกรมเป็นดังนี้

```
n = int(input('Please enter a number:'))
while n > 0:
```

```

    print(n)
    n = n - 1
print('Program terminated.')
```

รันโปรแกรมโดยป้อนข้อมูลนำเข้าเป็น 5 บันทึกผลที่ได้

จงอธิบายกระบวนการทำงานของโปรแกรกดังกล่าวโดยละเอียด

หากเปลี่ยนคำสั่ง $n = n - 1$ เป็น $n = n + 1$ จะได้ผลเช่นไร (คุณอาจลองคิดคำตอบโดยไม่ต้องทดลองจริง)

แก้ไขโค้ดโปรแกรมเป็นดังนี้

```

m = int(input('Enter the maximum number:'))
d = int(input('Enter the divisor:'))
i = 1
while i <= m:
    if i%d == 0:
        print(i)
    i = i + 1
print('Program terminated.')
```

หากรันแล้วใส่ข้อมูลป้อนเข้าเป็น 30 และ 7 ตามลำดับ จะได้ผลลัพธ์เช่นไร?

จงอธิบายการทำงานของโปรแกรม เมื่อใส่ข้อมูลป้อนเข้าเป็นจำนวนเต็มใด ๆ (สมมติว่าเป็น x และ y ตามลำดับ)

แก้ไขโค้ดโปรแกรมเป็นดังนี้

```
a = int(input('Enter 1st number:'))
b = int(input('Enter 2nd number:'))
i = 1
while i <= a:
    j = 1
    while j <= b:
        print(i,j)
        j = j + 1
    i = i + 1
```

ทดลองรันโปรแกรมโดยใส่ข้อมูลนำเข้าเป็นจำนวนเต็มสองตัวหลาย ๆ ครั้ง อธิบายว่าผลลัพธ์ของโปรแกรมนี้เป็นอย่างไร

หากแก้ไขบรรทัดที่ 6 จาก `while j <= b:` เป็น `while j <= i:` ผลลัพธ์ของโปรแกรมจะเป็นอย่างไร

2.4 โจทย์ปัญหา

1. จงเขียนโปรแกรมที่รับราคาสินค้ากับจำนวนเงินที่ลูกค้าจ่าย และแสดงจำนวนเงินที่ต้องทอน พร้อมกับแจกแจงด้วยว่าต้องทอนธนบัตรแต่ละชนิดเป็นจำนวนเท่าใด โดยธนบัตรชนิดใดไม่ต้องใช้ก็ไม่ต้องแสดงออกมา ดูรูปแบบจากตัวอย่าง

```
Welcome to Change Calculator.
```

```
Price: 247
```

```
Amount tendered: 1000
```

```
Change: 753
```

```
500: 1
```

```
100: 2
```

```
50: 1
```

```
2: 1
```

```
1: 1
```

```
Thank you.
```

2. จงเขียนโปรแกรมที่รับรหัสผ่านจากผู้ใช้ หากผู้ใช้กรอกรหัสเป็น k7e15 ให้แสดงข้อความว่า The password is correct. และสิ้นสุดการทำงาน หากผู้ใช้กรอกรหัสไม่ถูกต้อง ให้แสดงข้อความว่า The password is wrong, please try again และรับรหัสใหม่ แต่หากผู้ใช้กรอกรหัสครบสามครั้งแล้วยังไม่ถูกต้อง ให้แสดงข้อความว่า The system is disable. ดูตัวอย่างประกอบ

```
Enter the password: s78n2y
```

```
The password is wrong, please try again: k7e15
```

```
The password is correct.
```

```
Enter the password: s78n2y
```

```
The password is wrong, please try again: k7ed1er67
```

```
The password is wrong, please try again: 7e5d60
The system is disable.
```

3. จงเขียนโปรแกรมทายตัวเลข ที่วนรอบรับจำนวนเต็มในช่วง 1-1,000,000 จากผู้ใช้ และแสดงข้อความเพื่อบอกว่าน้อยไปหรือมากไป และให้ผู้เล่นทายใหม่ไปเรื่อย ๆ จนกว่าจะถูก และบอกจำนวนครั้งที่ผู้เล่นใช้ในการทายตัวเลขด้วย ดูตัวอย่างประกอบ

```
Welcome to THE SECRET NUMBER
Enter a number: 10
10 is too high.
Enter a number: 2
2 is too low.
Enter a number: 4
Congratulations!!!
You found the secret number in 3 times.
```

คำแนะนำ: คุณสามารถสั่งให้คอมพิวเตอร์ทำการสุ่มจำนวนเต็มตั้งแต่ 1-1,000,000 มาใส่ตัวแปรได้ โดยใช้คำสั่ง

```
x = random.randint(1, 1000000)
```

(สมมติว่าตัวแปรชื่อ x) แต่คุณจะใช้คำสั่งนี้ได้ต้องเพิ่มคำสั่ง

```
import random
```

ไว้ที่หัวโปรแกรม เพื่อให้ Python รู้จักคำสั่งเกี่ยวกับการสุ่มก่อน (จะกล่าวถึงโดยละเอียดอีกที)

4. จากข้อที่แล้ว ให้คุณพัฒนาเกมเพิ่มโดยเมื่อผู้เล่นเล่นจบแต่ละเกม ให้โปรแกรมถามว่าอยากเล่นต่อหรือไม่ หากผู้เล่นตอบว่าอยากเล่นต่อก็ให้เริ่มเกมใหม่ไปเรื่อย ๆ จนกว่าผู้เล่นจะตอบว่าไม่อยากเล่นแล้ว จึงค่อยสิ้นสุดการทำงาน และแสดงผลการเล่นที่ดีที่สุด ดูตัวอย่างประกอบ

```
Welcome to THE SECRET NUMBER
Enter a number: 10
10 is too high.
Enter a number: 2
2 is too low.
Enter a number: 4
Congratulations!!!
You found the secret number in 3 times.
Do you want to play again? (y/n): y
Enter a number: 200
200 is too high.
Enter a number: 100
Congraulations!!!
```

```
You found the secret number in 2 times.  
Do you want to play again? (y/n): n  
BEST: 2  
GOOD BYE.
```