





การเขียนโปรแกรมไพทอน ในโหมดสคริปต์













์ข้อดีของการทำงาน โดยใช้โหมดสคริปต์

ผู้ใช้สามารถบันทึกเก็บไว้เป็นไฟล์ได้เพื่อ จะได้นำมาแก้ไขเพิ่มเติมได้ในภายหลัง และยัง สามารถสั่งรันโปรแกรมที่บันทึกไว้ได้หลายครั้ง





การเขียนโปรแกรมไพทอน ใน PyCharm EDU ทำได้ดังนี้

1. สร้างโปรเจกต์

โดยเลือกเมนู File -> New Project... แล้วตั้งชื่อ ให้กับโปรเจกต์ จากนั้นคลิก Create ซึ่งโปรเจกต์จะ เป็นแหล่งรวมโปรแกรมไพทอนที่นักเรียนสร้างขึ้น โดยอาจมีหลายโปรแกรมสำหรับงานขนาดใหญ่ก็ได้









การเขียนโปรแกรมไพทอน ใน PyCharm EDU ทำได้ดังนี้



โดยเลือกเมนู File -> New... -> Python File ป้อนชื่อไฟล์โปรแกรมไพทอนในช่อง Name: แล้วคลิก ปุ่ม OK จะได้ไฟล์ใหม่ที่อยู่ภายใต้โปรเจกต์ที่สร้างขึ้น หลังจากนั้นนักเรียนสามารถเขียนโปรแกรมไพทอนได้











ตัวอย่างที่ 3.9 ร่วมด้วยช่วยแชร์ 2 Re-run

ให้นักเรียนสร้างโปรเจกต์และไฟล์โปรแกรม ไพทอน แล้วตั้งชื่อ หลังจากนั้นป้อนคำสั่งไพทอน ตามลำดับเหมือนในตัวอย่างที่ 3.8 พร้อมกับทดลอง รันด้วยการคลิกปุ่ม Execute แล้วป้อนราคาอาหาร และจำนวนผู้รับประทานอาหารที่แตกต่างกัน









8

ตัวอย่างที่ 3.9 ร่วมด้วยช่วยแชร์ 2 Re-run

้ป้อนคำสั่งไพทอน

```
🛵 main.py
      totalPrice = int(input('ค่าอาหารทั้งหมด'))
      number = int(input('จำนวนผู้รับประหานอาหาร'))
      avg = totalPrice / number
      print('จายคำอาหารคนละ', avg, 'บาท')
```



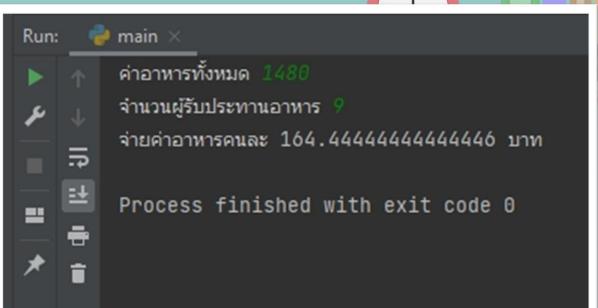






ตัวอย่างที่ 3.9 ร่วมด้วยช่วยแชร์ 2 Re-run

รูปต่อไปนี้แสดงผลที่ได้ จากการรัน หนึ่งครั้ง ที่ป้อน ราคาอาหารเป็น 1,480 บาท และจำนวนผู้รับประทาน อาหารเป็น 9 คน





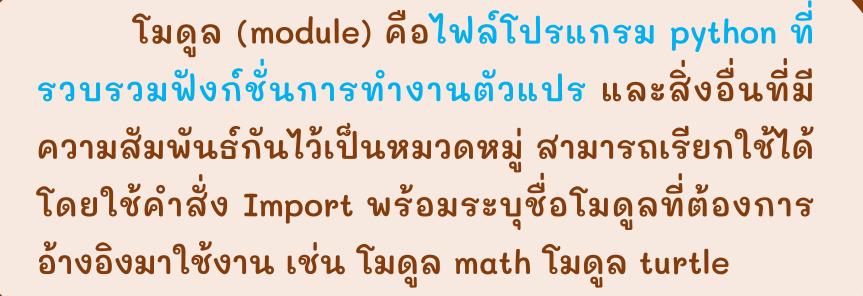






คลิกปุ่ม Execute เพื่อรันโปรแกรม

เกร็ดน่ารู้









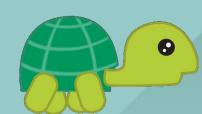
ฝึกเขียนโปรแกรม กับเต่าไพทอน

ไพทอน มีโมดูล turtle สำหรับฝึกเขียน โปรแกรมขั้นเริ่มต้น โดยใช้เต่าในการลากเส้นหรือ วาดรูป หากนักเรียนใช้ PyCharm EDU จะสามารถ เขียนคำสั่งเพื่อวาดรูป ได้ทั้งในโหมดอิมมีเดียทและ โหมดสคริปต์ แต่ในที่นี้จะแนะนำให้ใช้โหมดสคริปต์



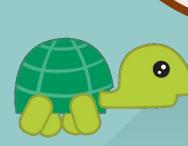








```
from turtle import *
       shape ("turtle")
       left (90)
       forward (100)
       left_(90)
       forward (100)
       left_(90)
       forward (100)
10
```

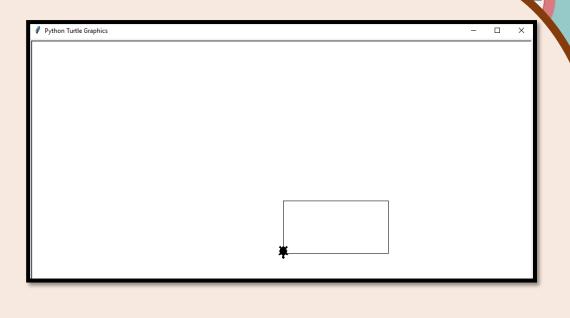


ต่อไปนี้



กิจกรรมที่ 3.4

1. ให้นักเรียนปรับ
โปรแกรมตัวอย่าง
ที่ 3.10 ให้วาดรูป
สี่เหลี่ยมที่มีความ
ยาวเป็น 2 เท่าของ
รูปเดิม



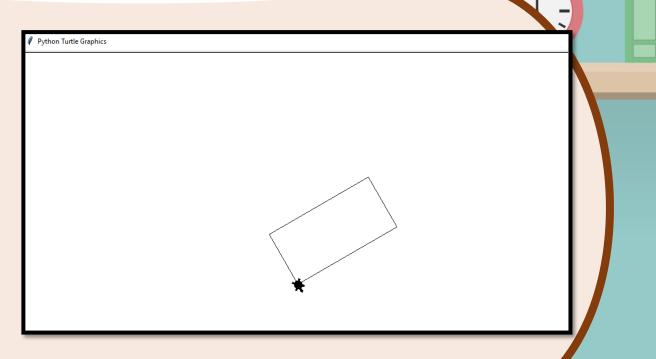




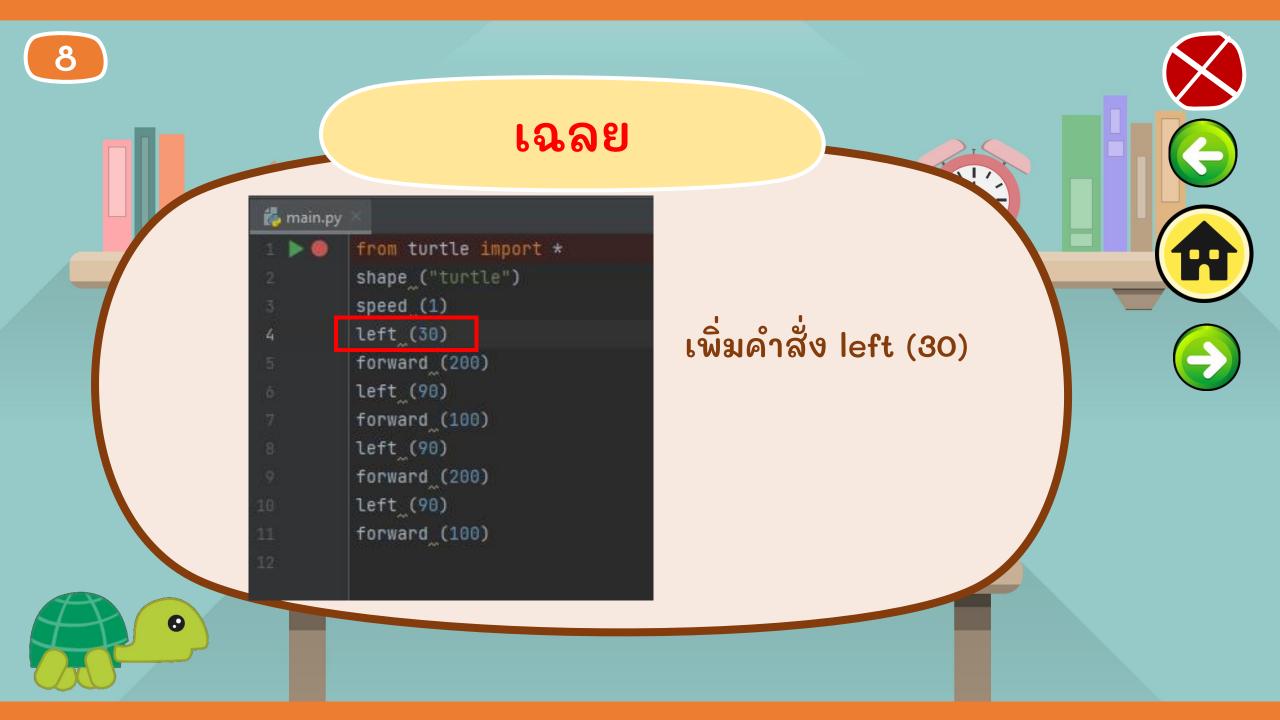


กิจกรรมที่ 3.4

2. ปรับโปรแกรม ในข้อที่ 1 เพื่อให้ สี่เหลี่ยมที่วาดขึ้น เอียงทำมุม 30 องศา กับแนวราบ ดังรูป







เกร็ดน่ารู้

ฟังก์ชันเพิ่มเติมในโมดูล turtle

- 1. การเรียกใช้โมดูล turtle แบบ from turtle import * ทำให้เราสามารถเรียกใช้ตัว แปรฟังก์ชัน และอื่นๆทั้งหมดจาก module turtle ได้
- 2. คำสั่ง shape()สามารถระบุให้เต่าเปลี่ยนรูปร่างไปเป็นอย่างอื่นได้ โดยระบุรูปร่าง เป็นสต่งอยู่ภายในวงเล็บ เช่น shape ('circle') หรือ shape ('square') หากต้องการ เปลี่ยนกลับเป็นรูปลูกศร ให้ใช้สตริง 'classic'
- 3. คำสั่ง color () ใช้เปลี่ยนสีให้กับเส้นที่เต่าลาก โดยระบุชื่อสีเป็นสตริง อยู่ภายใน วงเล็บ เช่น color ('green') หรือ color ('yellow')
- 4. คำสั่ง pensize () ใช้กำหนดขนาดความหนาของเส้น โดยระบุขนาดเป็นจำนวน เต็มอยู่ภายในวงเล็บ เช่น pensize(5)





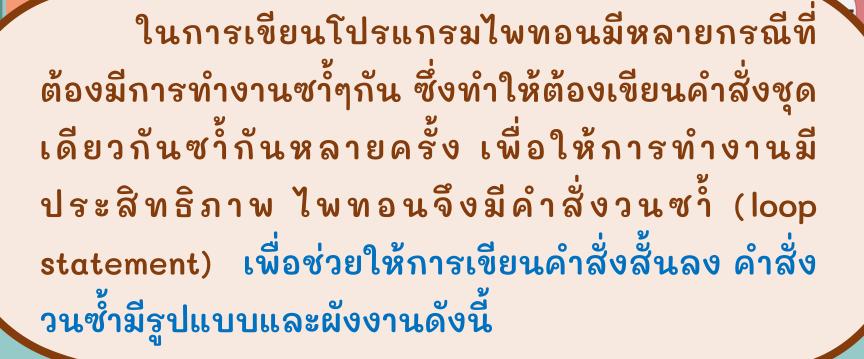








้การทำงานแบบวนซา๊











้การทำงานแบบวนซา



ชุดคำสั่ง # คำสั่งที่ต้องการให้ทำซา้

ค่าใน ตัวแปร จะเปลี่ยนไปเรื่อยๆ

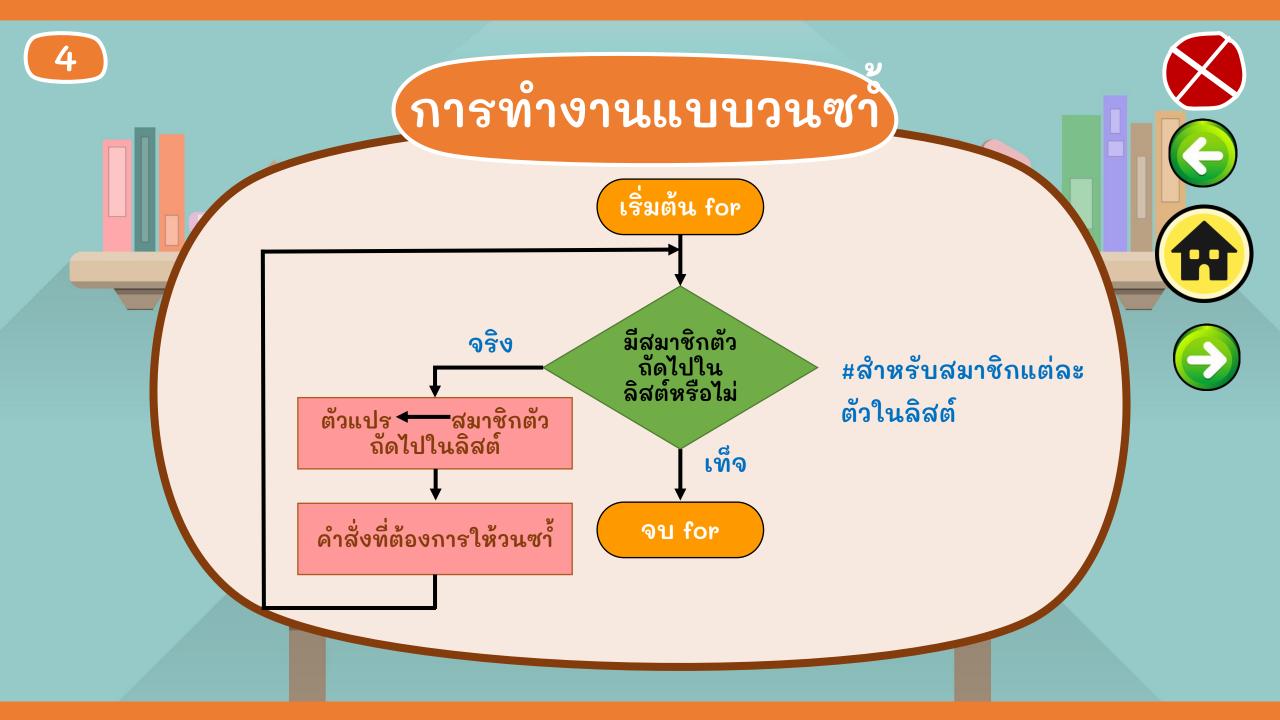
ตามลำดับใน ลิสต์











ตัวอย่างที่ 3.11 echo

โปรแกรมต่อไปนี้ รับชื่อ และนามสกุล แล้วพิมพ์ซา้ออกทาง จอภาพจำนวน 5 ครั้ง

```
name = input('Please enter your firstname')
surname = input('Please enter your lastname')
for x in range(5) :
    print(x, name)
    print(x, surname)
print('bye bye!')
```

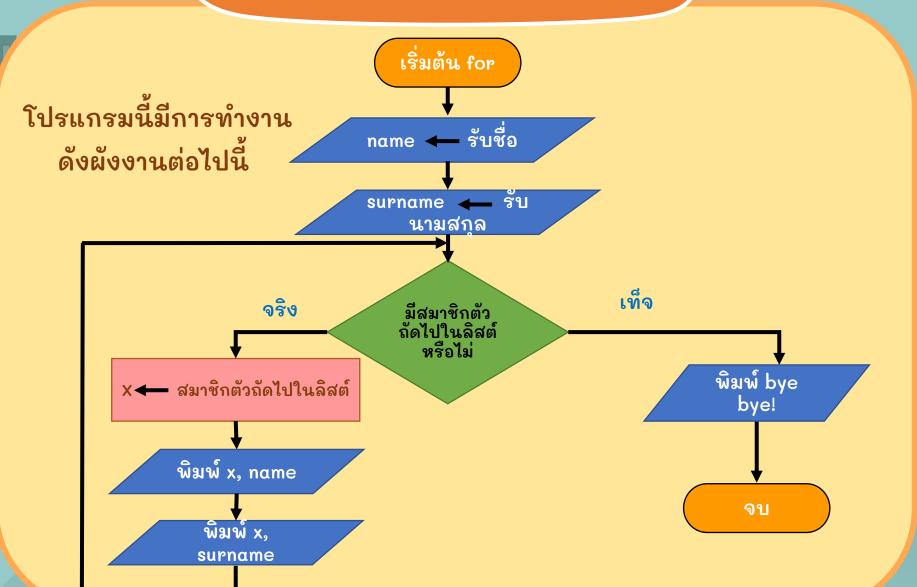






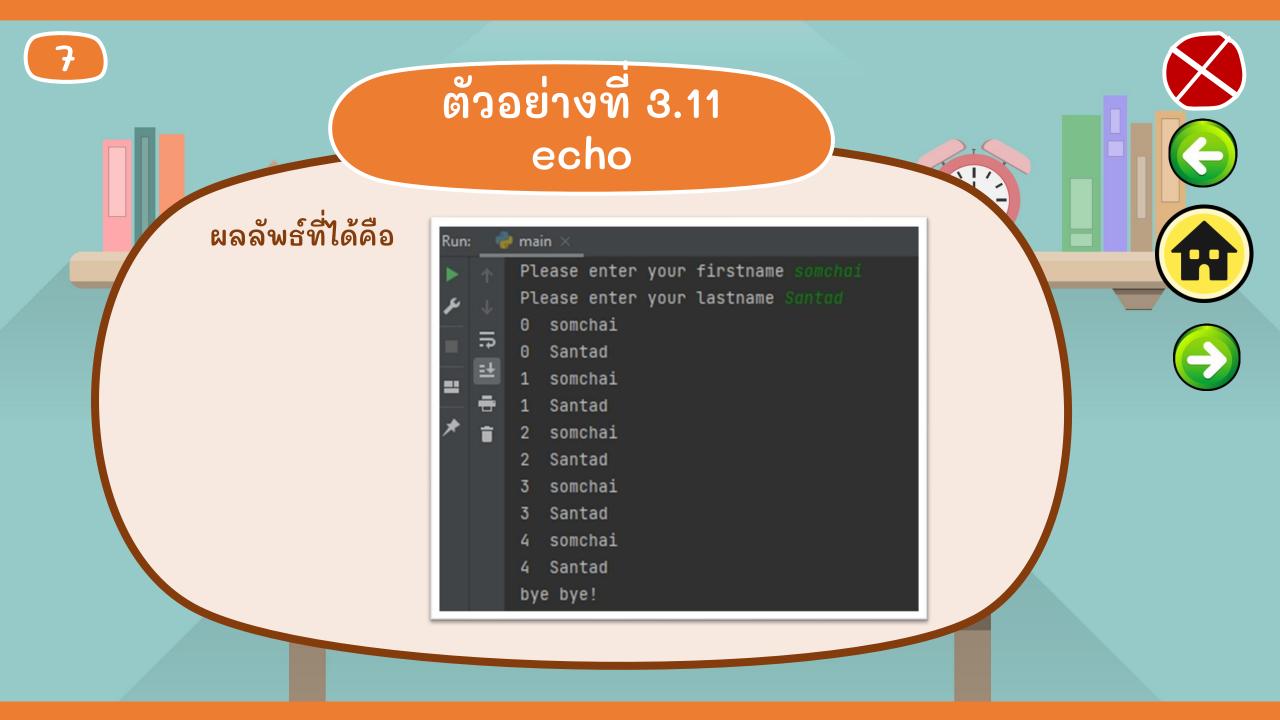


์ ตัวอย่างที่ 3.11 echo









ตัวอย่างที่ 3.11 echo

ตัวอย่างที่ 3.11 อธิบายได้ดังนี้

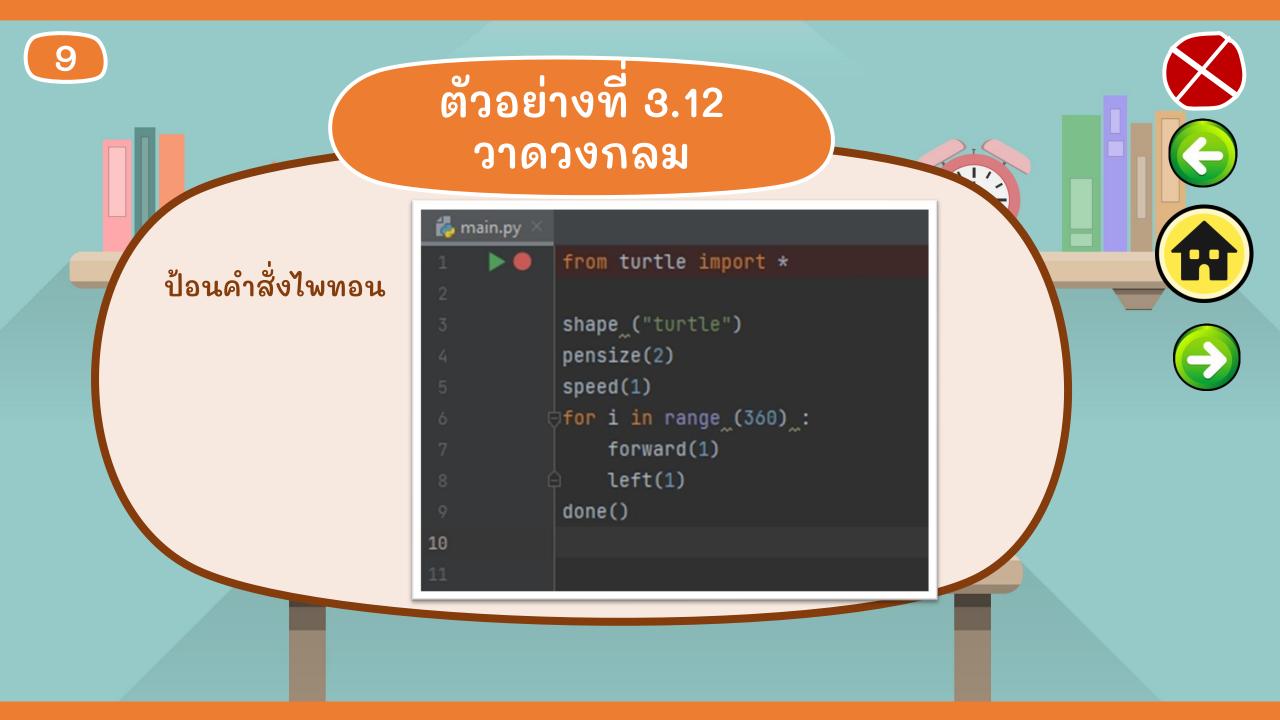
- 1. บรรทัดที่ 3 ถึง 5 เป็นส่วนที่กำหนดให้มีการรวนซา้ำให้สังเกตย่อ หน้าของบรรทัดที่ 4 และ 5 เป็นการกำหนดว่าส่วนนี้จะต้องถูกวน ซา้ภายใต้คำสั่ง for ในบรรทัดที่ 3
- 2. คำสั่ง range(5) ในบรรทัดที่ 3 เป็นคำสั่งสร้างลิสต์จำนวนเต็ม 5 จำนวน ตั้งแต่ 0 , 1 , 2 , 3 , 4 ซึ่งในแต่ละรอบของการวนซา้์ ตัว แปร x จะมีค่าเปลี่ยนไปตามลำดับแต่ละค่าในลิสต์
- 3. ผลลัพธ์ที่ได้ คือ คำสั่งในบรรทัดที่ 4 และ 5 จะทำงานซา้ำกันเป็น จำนวน 5 รอบ และแต่ละรอบ ค่า x ที่ถูกพิมพ์ออกมา จึงมีค่า เปลี่ยนไปตามค่าที่อยู่ในลิสต์

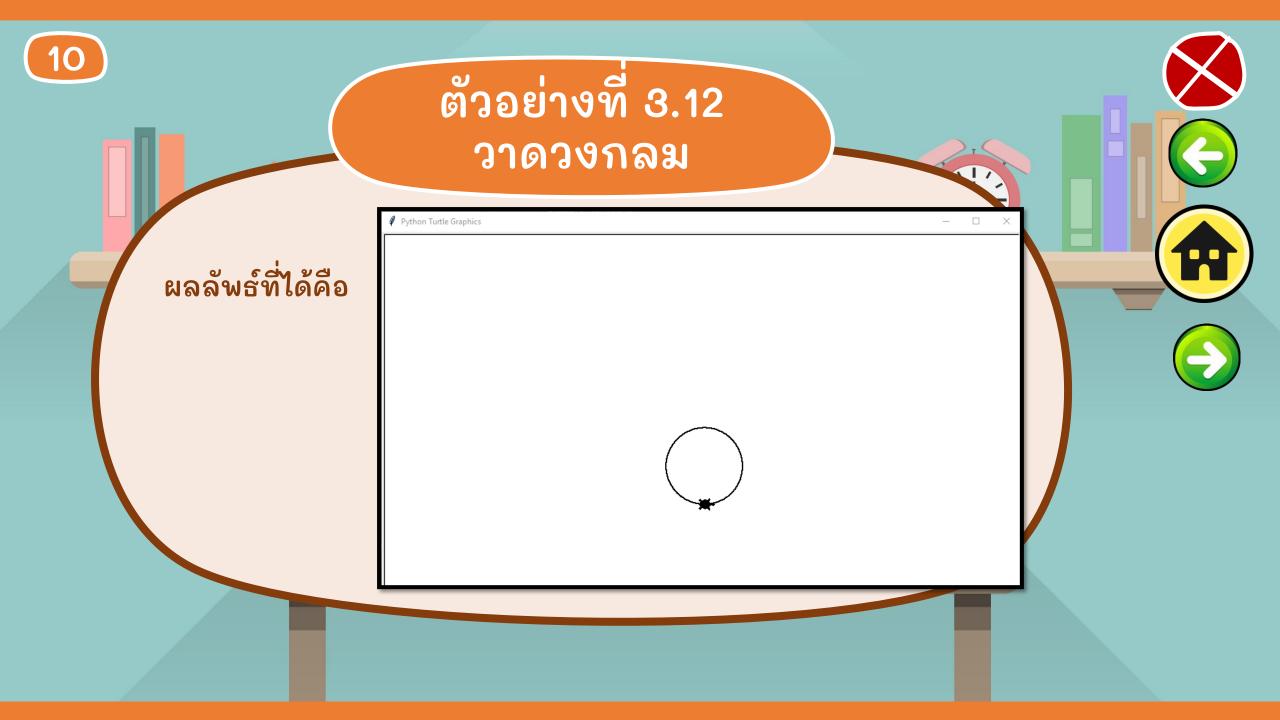












ป้อนคำสั่งไพทอน

```
i = int(input('คุณต้องการคำนาณกีรอบคะ'))

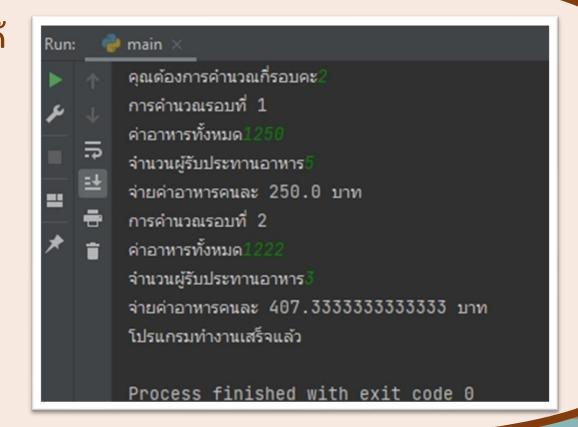
for count in range(i) :
    print('การคำนาณรอบที่', count+1)
    totalPrice = int(input('คำอาหารทั้งหมด'))
    number = int(input('ลำนานผู้รับประหานอาหาร'))
    avg = totalPrice/number
    print('ล่ายคำอาหารคนละ', avg, 'บาท')

print('โปรแกรมทำงานเสร็จแล้ว')
```





ผลลัพธ์ที่ได้









1. ในขณะที่เขียนโปรแกรม นักเรียนจะไม่ทราบจำนวนรอบที่ต้องการ ดังนั้นจึงต้องรับจำนวนรอบจากผู้ใช้ ในบรรทัดที่ 1 ด้วยคำสั่ง input() แล้วจึงใช้คำสั่ง int() แปลงสตริงที่รับเข้าจาก input() ให้เป็นจำนวน เต็ม เพื่อนำไปกำหนดค่าให้กับตัวแปร i

2. บรรทัดที่ 3 ถึง 7 เป็นส่วนที่ต้องการให้ทำงานซา้ำ จึงนำไปใส่ไว้ ภายใต้คำสั่ง for ในบรรทัดที่ 2 ซึ่งคำสั่ง for จะวนซา้ำเท่ากับจำนวน i รอบ โดยใช้ตัวแปร count นับรอบการทำงาน









3. บรรทัดที่ 3 จะแสดงตัวเลขระบุ รอบที่ ของการวนซา้ ซึ่งต้องการให้ แสดงข้อความ "การคำนวณรอบที่ 1" ไปจนถึง "การคำนวณรอบที่ i" แต่เนื่องจาก range() ให้ค่าตั้งแต่ 0 , 1, ..., i-1 ดังนั้น เพื่อให้แสดง จำนวนรอบที่ถูกต้อง ตัวแปร count ที่เริ่มต้นด้วย 0 จึงต้องบวกด้วย 1























ตัวอย่างที่ 3.14 มากไป-น้อยไป



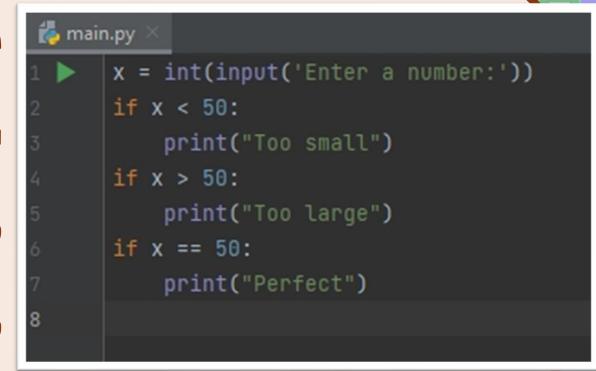






โปรแกรมด้านขวารับค่าจาก ผู้ใช้แล้วเก็บไว้ในตัวแปร x

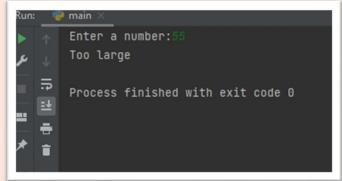
- ถ้า x มีค่าน้อยกว่า 50 โปรแกรม จะแสดงผลลัพธ์ว่า Too small
- ถ้า x มีค่ามากกว่า 50 จะแสดง ผลลัพธ์ว่า Too large
- แต่ถ้า x มีค่าเท่ากับ 50 จะแสดง
 ผลลัพธ์ว่า Perfect



ตัวอย่างที่ 3.14 มากไป-น้อยไป

ตัวอย่างผลลัพธ์ที่ได้









x มีค่าน้อยกว่า 50

x มีค่ามากกว่า 50

x มีค่าเท่ากับ 50

จะสังเกตเห็นว่าในตัวอย่างข้างต้นเงื่อนไข x < 50,x > 50 และ x==50 จะ เป็นตัวกำกับว่าคำสั่งภายใต้ if จะทำงานเมื่อเงื่อนไขเป็นจริงเท่านั้น

ชวนคิด

จากตัวอย่างที่ 3.14 ถ้า x มีค่า ดังต่อไปนี้ ผลลัพธ์ที่ได้คืออะไร

$$x = 80$$

$$x = 30$$

$$x = 50$$

$$x = -5$$



เกร็ดน่ารู้

ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการที่ใช้ในการเปรียบเทียบ ได้แก่ น้อยกว่า (<) มากกว่า (>) เท่ากับ (==) น้อยกว่าหรือ เท่ากับ (<=) มากกว่าหรือเท่ากับ (>=) และ ไม่เท่ากับ (!=)









การทำงานแบบมีทางเลือก

การทำงานแบบมีทางเลือก มีประโยชน์กรณีที่ เราต้องการให้โปรแกรมทำการตัดสินใจบางอย่าง แล้ว เลือกทำงานชุดคำสั่งตามผลของเงื่อนไขที่กำหนดไว้ โดยมีรูปแบบคำสั่งแบบมีทางเลือก (selection statement) ในการใช้งานเบื้องต้นคือ

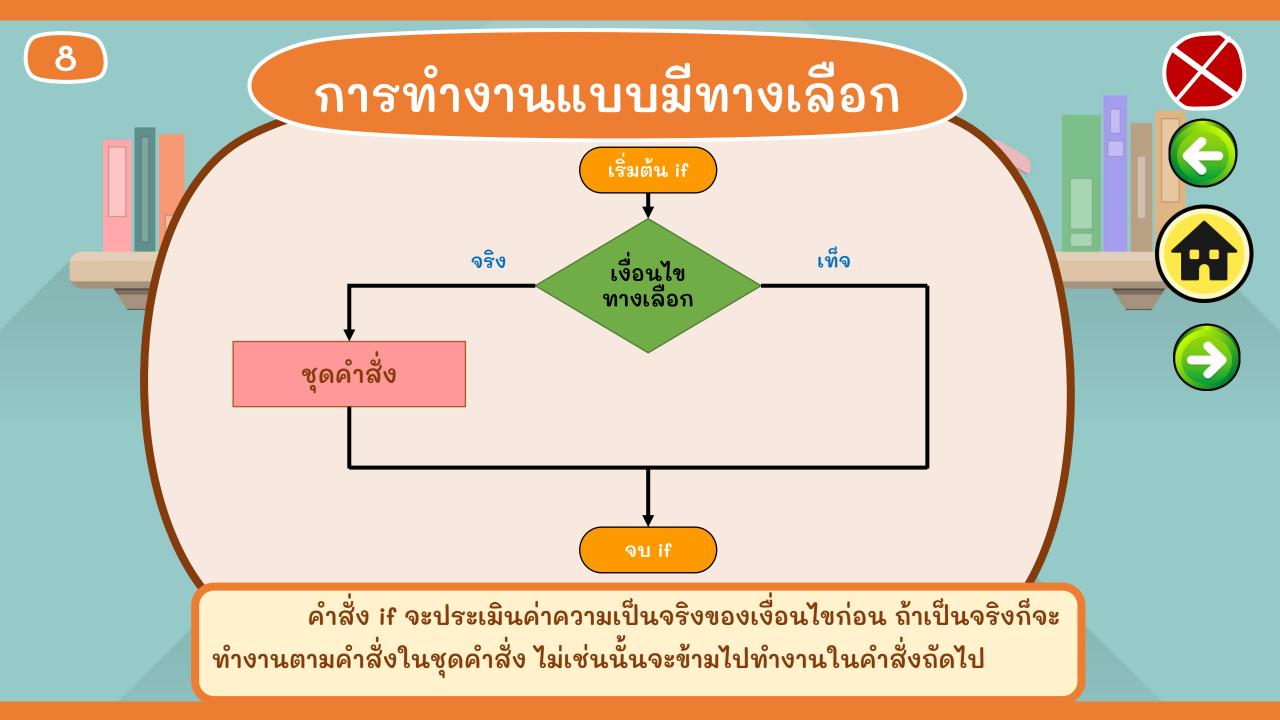
> if เงื่อนไขทางเลือก : ชุดคำสั่ง











ตัวอย่าง 3.15 ทายใจ

เขียนโปรแกรมไพทอนเพื่อให้เพื่อนเล่นเกมทายตัวเลข ดังนี้

```
puzzle = 3
guess = int(input('เลขที่ทาย (1-10) คือ'))
if puzzle == guess:
print('คุณทายถูกแล้วค่ะ')
```

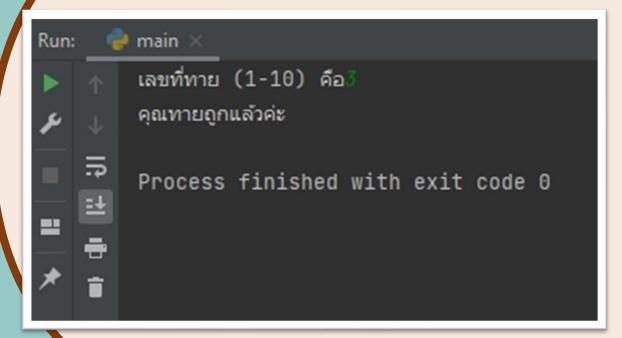






ตัวอย่าง 3.15 ทายใจ





ตัวอย่างที่ 3.15 อธิบายได้ดังนี้

1. บรรทัดที่ 1 กำหนดตัวเลขที่ให้ทาย เก็บไว้ในตัวแปร puzzle

- 2. บรรทัดที่ 2 รับข้อมูลเข้าจากผู้ถ่ายเก็บ ไว้ในตัวแปร guess
- 3. ถ้าเงื่อนไขทางเลือก puzzle == guess ในคำสั่ง if ในบรรทัดที่ 3 เป็นจริง นั่น หมายความว่าทายตัวเลขถูกต้อง โปรแกรมจะแสดงข้อความตามคำสั่ง print









การทำงานแบบมีทางเลือก

ในบางสถานการณ์ของการเขียนโปรแกรมที่ต้องการให้ โปรแกรมทำงานมากกว่าหนึ่งทางเลือก เช่น ถ้าโปรแกรมใน ตัวอย่างที่ 3.15 สามารถบอกผลของการทายตัวเลขเมื่อทายไม่ถูก ได้ด้วย ก็จะน่าสนใจขึ้น ภาษาไพทอน มีคำสั่ง if-else ให้ใช้ โดยมี รปแบบการใช้ดังนี้

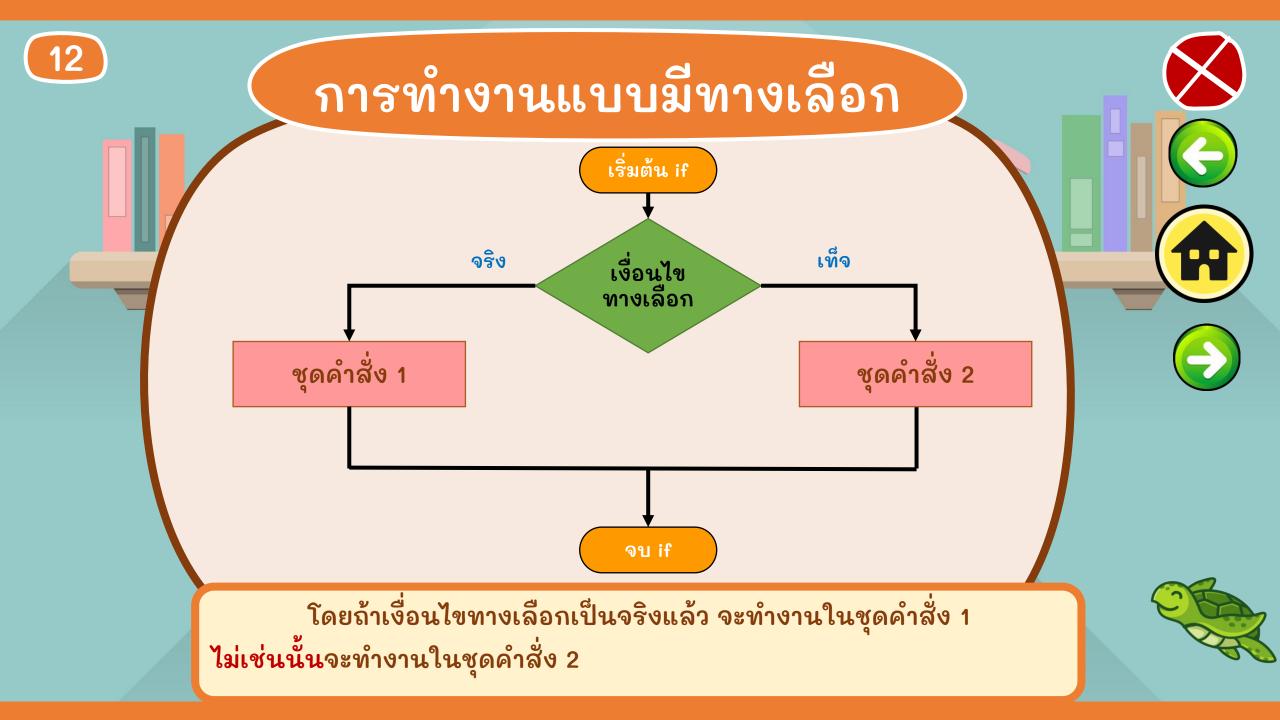




if เงื่อนไขทางเลือก : ชุดคำสั่ง 1

else:

ชุดคำสั่ง 2



ตัวอย่าง 3.16 ทายใจ 2

ปรับโปรแกรมในตัวอย่างที่ 3.15 ให้สามารถแจ้งผลของการทาย ตัวเลขที่ไม่ถูกต้องได้

```
    main.py ×

puzzle = 3

guess = int(input('เลขที่หาย (1-10) คือ'))

if puzzle == guess:
    print('คุณทายถูกแล้วค่ะ')

else:
    print('คุณยังทายไม่ถูกค่ะ')

7

8
```





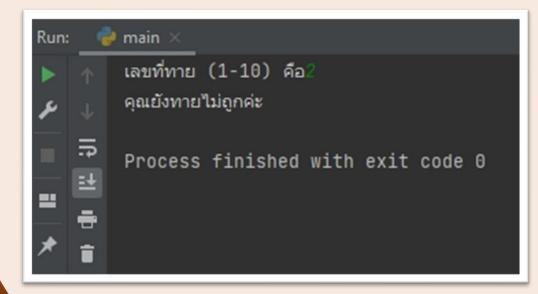






ตัวอย่าง 3.15 ทายใจ





ตัวอย่างที่ 3.16 อธิบายได้ดังนี้

ต่อเนื่องจากตัวอย่างที่ 3.15 หากผู้ใช้ป้อนตัวเลขที่ทายไม่ถูกต้อง จะทำให้ผลของเงื่อนไขทางเลือกใน คำสั่ง if บรรทัดที่ 3 เป็นเท็จ คำสั่ง ในบรรทัดที่ 4 จะถูกข้ามไป และ คำสั่งในบรรทัดที่ 6 ที่อยู่ภายใต้ else: จะถูกทำงานแทน











เกร็ดน่ารู้

์การสุ่มตัวเลข

การสุ่มตัวเลขสามารถนำมาใช้ประโยชน์ในการสร้างเกมได้ เช่น การจำลองการทอดลูกเต๋า โดยให้โปรแกรมสุ่มตัวเลขในช่วง 1-6

ในไพทอนสามารถทำได้โดยการเรียกฟังก์ชัน randint () จากโมดูล random แล้วเรียกคำสั่ง randint (1, 6) ดังนี้

from random import randint

x = randint(1, 6)

print (x)











ชา สรุปท้ายบท

การเขียนโปรแกรมไพทอนจะใช้ไอดีอี เป็นเครื่องมือ ช่วยในการพัฒนาโปรแกรม ซึ่งประกอบด้วยเครื่องมือแก้ไข โปรแกรมต้นฉบับ เครื่องมือแก้ไขจุดบกพร่องของ โปรแกรม และเครื่องมือที่ช่วยรันโปรแกรมไอดีอีไพทอน โดยทั่วไปจะทำงานได้ในโหมดอิมมีเดียทและโหมดสคริปต์ บทนี้ได้อธิบายคำสั่งพื้นฐาน เช่น print () ใช้สำหรับแสดง ค่าข้อมูลออกทางจอภาพ และ input () ทำหน้าที่รับข้อมูล เข้าจากคีย์บอร์ด









สรุปท้ายบท - พ

