



วิทยาการคำนวณ ม.1

บทที่ 3 การโปรแกรมด้วยภาษาไพทอน

คลิกเพื่อเข้าสู่บทเรียน

กรุณากรอกชื่อ

เข้าสู่ระบบ

ยินดีต้อนรับ

2

เข้าสู่บทเรียน!

เมนูหลัก

คำชี้แจง

แบบทดสอบ

ผู้จัดทำ

การเขียนโปรแกรมไพทอน
ในโหมดสคริปต์

ฝึกเขียนโปรแกรม
กับเต่าไพทอน

การทำงานแบบวนซ้ำ

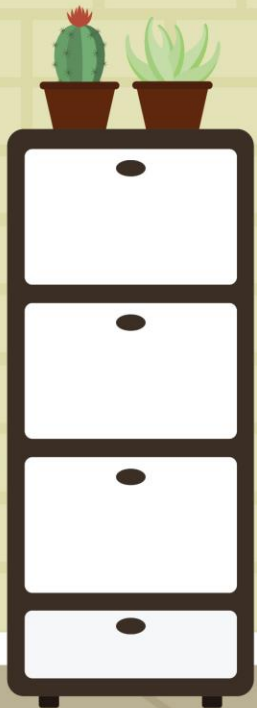
การทำงานแบบมีทางเลือก



คำชี้แจง

1. ให้นักเรียนศึกษาบทเรียน
2. ให้นักเรียนทำแบบทดสอบหลังเรียนทั้งหมด 10 ข้อ 10 คะแนน

OCTOBER
Monday
08



การเขียนโปรแกรมไพทอน ในโหมดสคริปต์



การเขียนโปรแกรมไพทอน ในโหมดสคริปต์

ในโหมดสคริปต์ นักเรียนต้องเขียนชุดคำสั่งไพทอนที่ต้องการทำงานให้ครบถ้วนทั้งหมดก่อน ซึ่งเรียกว่าโปรแกรม โดยจะต้องบันทึกไว้เป็นไฟล์ แล้วจึงสั่งโปรแกรมทำงานตามคำสั่งทั้งหมดตามลำดับ เรียกว่า **การรันโปรแกรม**



ข้อดีของการทำงาน โดยใช้โหนดสคริปต์

ผู้ใช้สามารถบันทึกเก็บไว้เป็นไฟล์ได้เพื่อ
จะได้นำมาแก้ไขเพิ่มเติมได้ในภายหลัง และยัง
สามารถสั่งรันโปรแกรมที่บันทึกไว้ได้หลายครั้ง



การเขียนโปรแกรมไพทอน ใน PyCharm EDU ทำได้ดังนี้

1. สร้างโปรเจกต์

โดยเลือกเมนู File -> New Project... แล้วตั้งชื่อให้กับโปรเจกต์ จากนั้นคลิก Create ซึ่งโปรเจกต์จะเป็นแหล่งรวมโปรแกรมไพทอนที่นักเรียนสร้างขึ้น โดยอาจมีหลายโปรแกรมสำหรับงานขนาดใหญ่ก็ได้



การเขียนโปรแกรมไพทอน ใน PyCharm EDU ทำได้ดังนี้

2. สร้างโปรแกรมไพทอนในโปรเจกต์

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



การเขียนโปรแกรมไพทอน ใน PyCharm EDU ทำได้ดังนี้

3. การบันทึกไฟล์และโปรเจกต์

โดยเลือกเมนู File -> Save all

4. การรันโปรแกรมที่สร้างขึ้น

ให้คลิกปุ่ม  (Execute)



ตัวอย่างที่ 3.9 ร่วมด้วยช่วยแชร์ 2 Re-run

ให้นักเรียนสร้างโปรเจกต์และไฟล์โปรแกรมไพทอน แล้วตั้งชื่อ หลังจากนั้นป้อนคำสั่งไพทอนตามลำดับเหมือนในตัวอย่างที่ 3.8 พร้อมกับทดลองรันด้วยการคลิกปุ่ม Execute แล้วป้อนราคาอาหารและจำนวนผู้รับประทานอาหารที่แตกต่างกัน



ตัวอย่างที่ 3.9 ร่วมด้วยช่วยแชร์ 2 Re-run

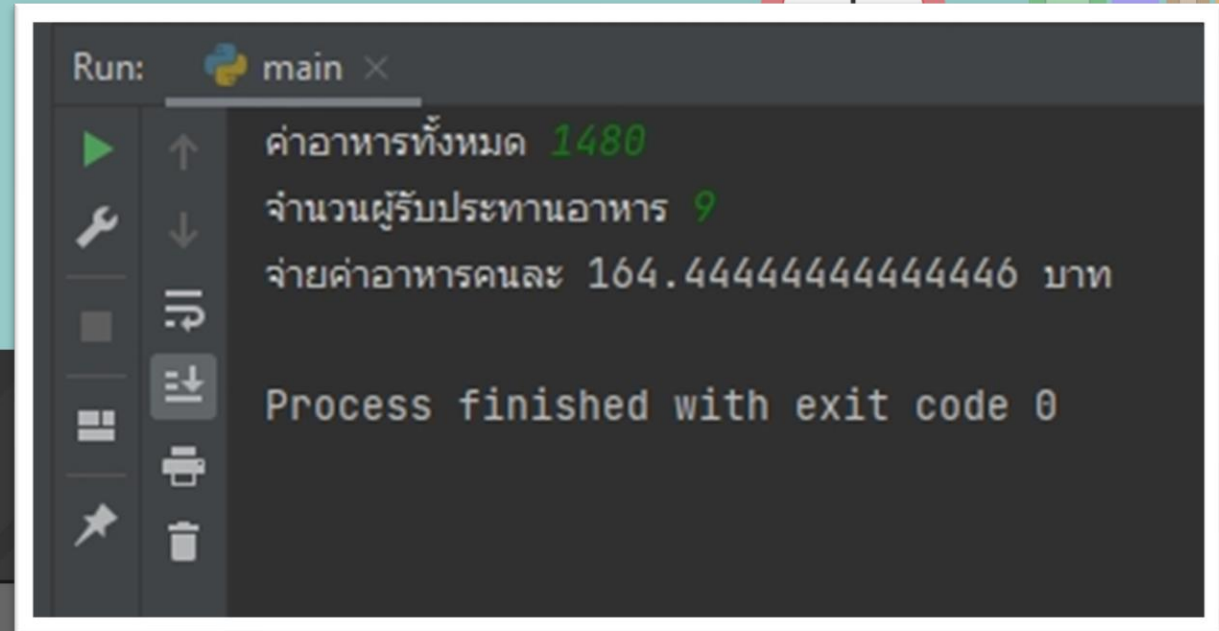
ป้อนคำสั่งไพทอน

```
onProject1 - main.py
main.py x
1 totalPrice = int(input('ค่าอาหารทั้งหมด'))
2 number = int(input('จำนวนผู้รับประทานอาหาร'))
3 avg = totalPrice / number
4 print('จ่ายค่าอาหารคนละ', avg, 'บาท')
5
6
7
```



ตัวอย่างที่ 3.9 ร่วมด้วยช่วยแชร์ 2 Re-run

รูปต่อไปนี้จะแสดงผลที่ได้
จากการรัน หนึ่งครั้ง ที่ป้อน
ราคาอาหารเป็น 1,480 บาท
และจำนวนผู้รับประทานอาหาร
อาหารเป็น 9 คน



```
Run: main ×
▶ ↑
⚙ ↓
□ ↕
☰ ⏮
★ ⏭
   ⏹

ค่าอาหารทั้งหมด 1480
จำนวนผู้รับประทานอาหาร 9
จ่ายค่าอาหารคนละ 164.44444444444446 บาท

Process finished with exit code 0
```

คลิกปุ่ม Execute
เพื่อรันโปรแกรม

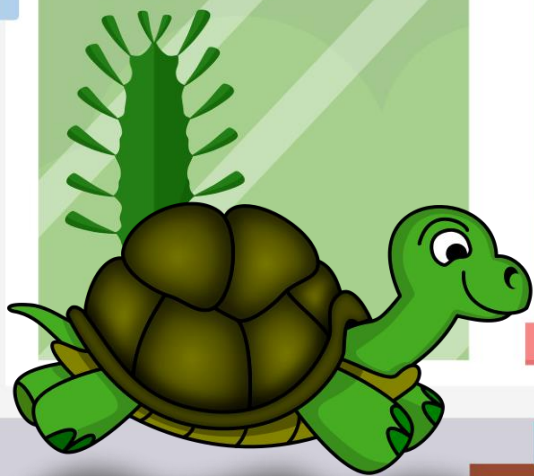


เกร็ดน่ารู้

โมดูล (module) คือไฟล์โปรแกรม python ที่รวบรวมฟังก์ชันการทำงานตัวแปร และสิ่งอื่นที่มีความสัมพันธ์กันไว้เป็นหมวดหมู่ สามารถเรียกใช้ได้โดยใช้คำสั่ง Import พร้อมระบุชื่อโมดูลที่ต้องการอ้างอิงมาใช้งาน เช่น โมดูล math โมดูล turtle

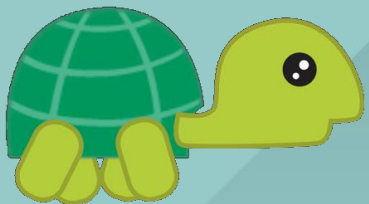


ฝึกเขียนโปรแกรม กับเต่าไพทอน



ฝึกเขียนโปรแกรม กับเต่าไพทอน

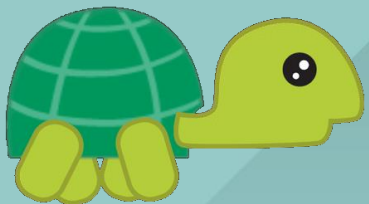
ไพทอน มีโมดูล `turtle` สำหรับฝึกเขียนโปรแกรมขั้นเริ่มต้น โดยใช้เต่าในการลากเส้นหรือวาดรูป หากนักเรียนใช้ PyCharm EDU จะสามารถเขียนคำสั่งเพื่อวาดรูป ได้ทั้งในโหมดอิมมีเดียทและโหมดสคริปต์ แต่ในที่นี้จะแนะนำให้ใช้โหมดสคริปต์



ตัวอย่างที่ 3.10 วาดรูปสี่เหลี่ยมจัตุรัส

เขียน บันทึก
และรันโปรแกรม
ต่อไปนี้

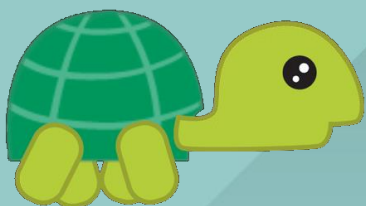
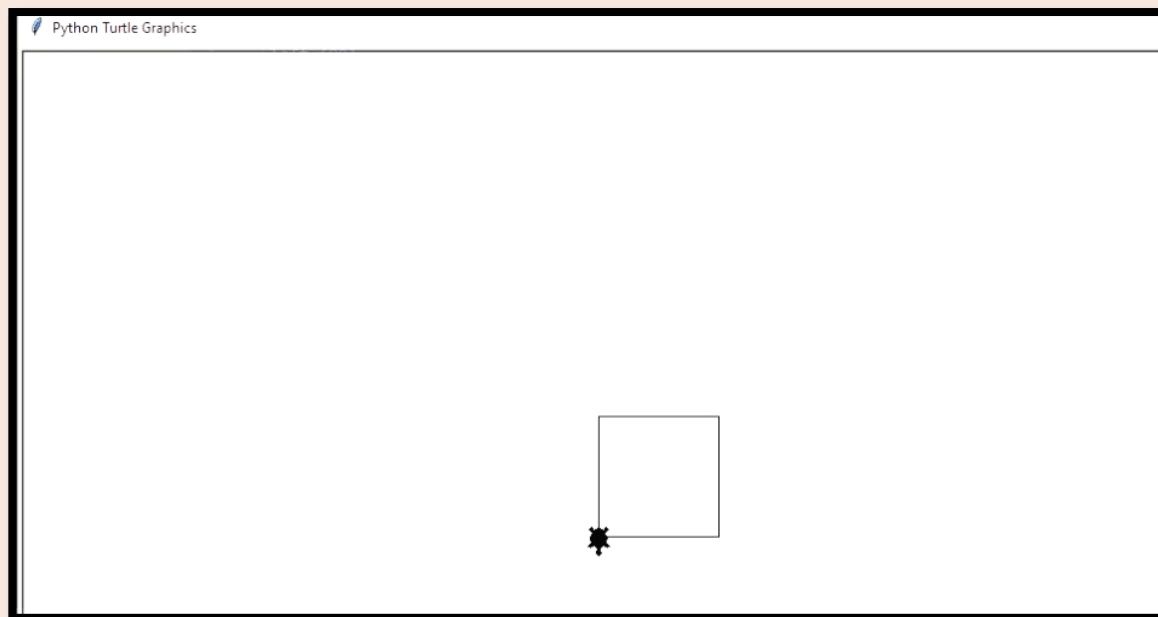
```
main.py x
1  ▶ from turtle import *
2    shape("turtle")
3    speed(1)
4    forward(100)
5    left(90)
6    forward(100)
7    left(90)
8    forward(100)
9    left(90)
10   forward(100)
11
```



4

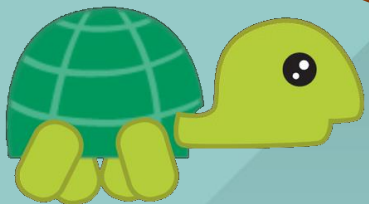
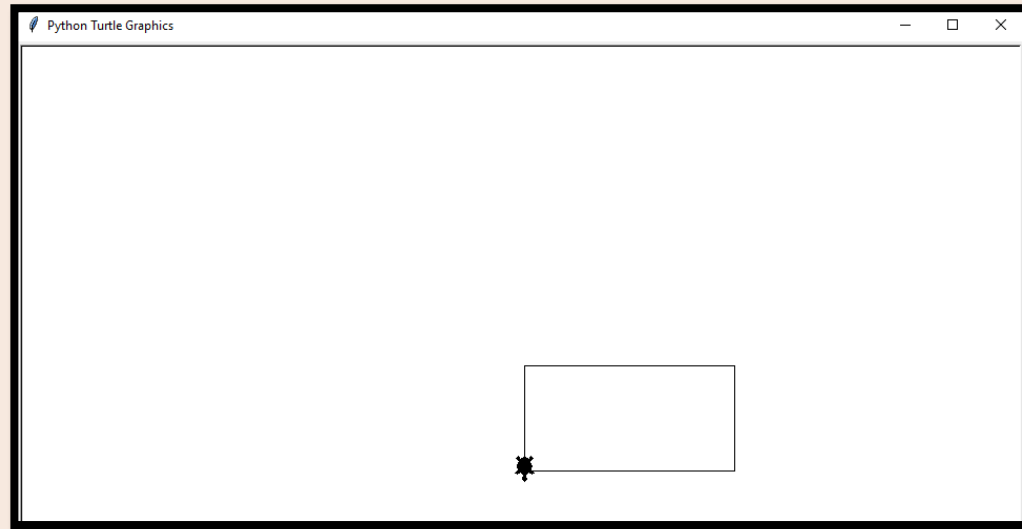
ตัวอย่างที่ 3.10 วาดรูปสี่เหลี่ยมจตุรัส

ผลลัพธ์ที่ได้



กิจกรรมที่ 3.4

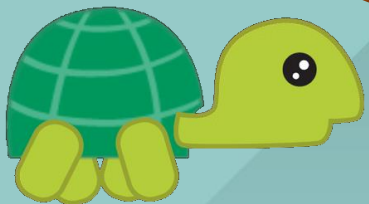
1. ให้นักเรียนปรับ
โปรแกรมตัวอย่าง
ที่ 3.10 ให้วาดรูป
สี่เหลี่ยมที่มีความ
ยาวเป็น 2 เท่าของ
รูปเดิม



เฉลย

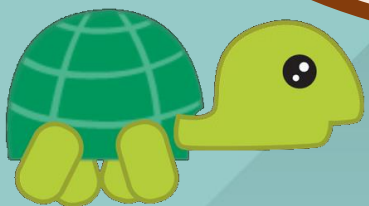
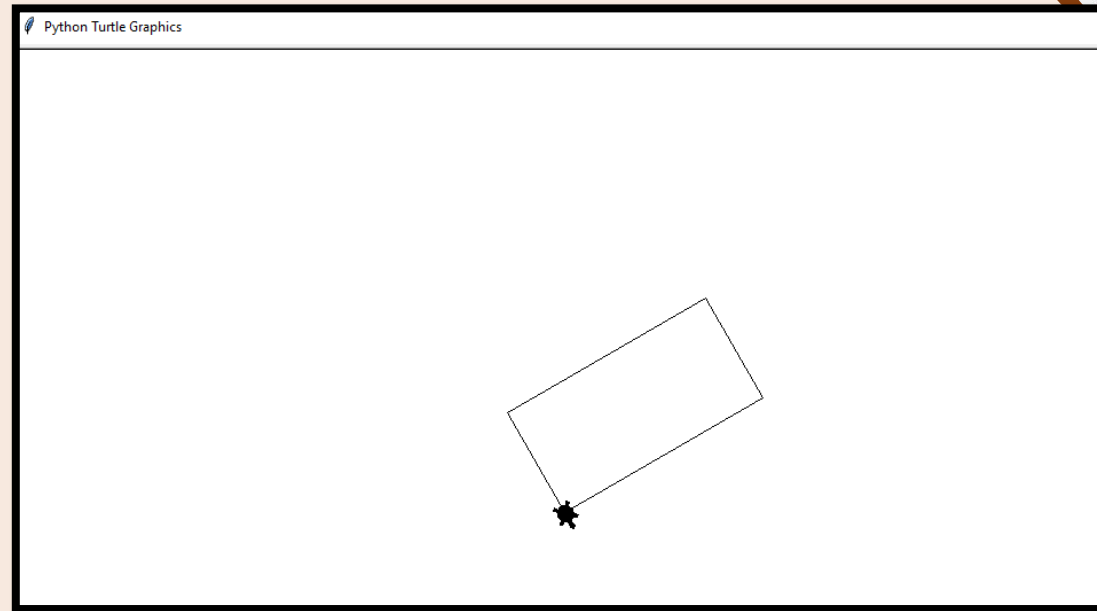
ปรับค่าให้เป็น 200
ในบรรทัดที่ 4 และ 8

```
main.py x
1  ▶ from turtle import *
2    shape("turtle")
3    speed(1)
4    forward(200)
5    left(90)
6    forward(100)
7    left(90)
8    forward(200)
9    left(90)
10   forward(100)
11
```



กิจกรรมที่ 3.4

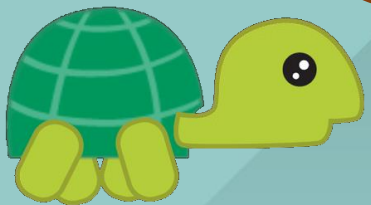
2. ปรับโปรแกรม
ในข้อที่ 1 เพื่อให้
สี่เหลี่ยมที่วาดขึ้น
เอียงทำมุม 30
องศา กับแนวนอน
ดังรูป



เฉลย

```
main.py x
1  from turtle import *
2  shape("turtle")
3  speed(1)
4  left(30)
5  forward(200)
6  left(90)
7  forward(100)
8  left(90)
9  forward(200)
10 left(90)
11 forward(100)
12
```

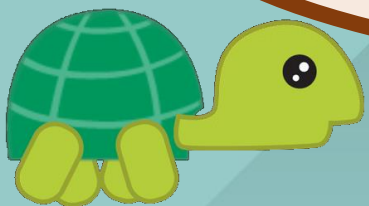
เพิ่มคำสั่ง left (30)



เก๋รดนนำร้

ฟังก์ชันเพิ่มเติมในโมดูล turtle

1. การเรียกใช้โมดูล turtle แบบ `from turtle import *` ทำให้เราสามารถเรียกใช้ตัวแปรฟังก์ชัน และอื่นๆทั้งหมดจาก module turtle ได้
2. คำสั่ง `shape()` สามารถระบุให้เต่าเปลี่ยนรูปร่างไปเป็นอย่างอื่นได้ โดยระบุรูปร่างเป็นสตริงอยู่ในวงเล็บ เช่น `shape('circle')` หรือ `shape('square')` หากต้องการเปลี่ยนกลับเป็นรูปลูกศร ให้ใช้สตริง `'classic'`
3. คำสั่ง `color()` ใช้เปลี่ยนสีให้กับเส้นที่เต่าลาก โดยระบุชื่อสีเป็นสตริง อยู่ในวงเล็บ เช่น `color('green')` หรือ `color('yellow')`
4. คำสั่ง `pensize()` ใช้กำหนดขนาดความหนาของเส้น โดยระบุขนาดเป็นจำนวนเต็มอยู่ในวงเล็บ เช่น `pensize(5)`



1

การทำงานแบบวนซ้ำ



การทำงานแบบวนซ้ำ

ในการเขียนโปรแกรมไพทอนมีหลายกรณีที่ต้องมีการทำงานซ้ำๆกัน ซึ่งทำให้ต้องเขียนคำสั่งชุดเดียวกันซ้ำๆกันหลายครั้ง เพื่อให้การทำงานมีประสิทธิภาพ ไพทอนจึงมีคำสั่งวนซ้ำ (loop statement) เพื่อช่วยให้การเขียนคำสั่งสั้นลง คำสั่งวนซ้ำมีรูปแบบและผังงานดังนี้



การทำงานแบบวนซ้ำ

for ตัวแปร in ลิสต์:

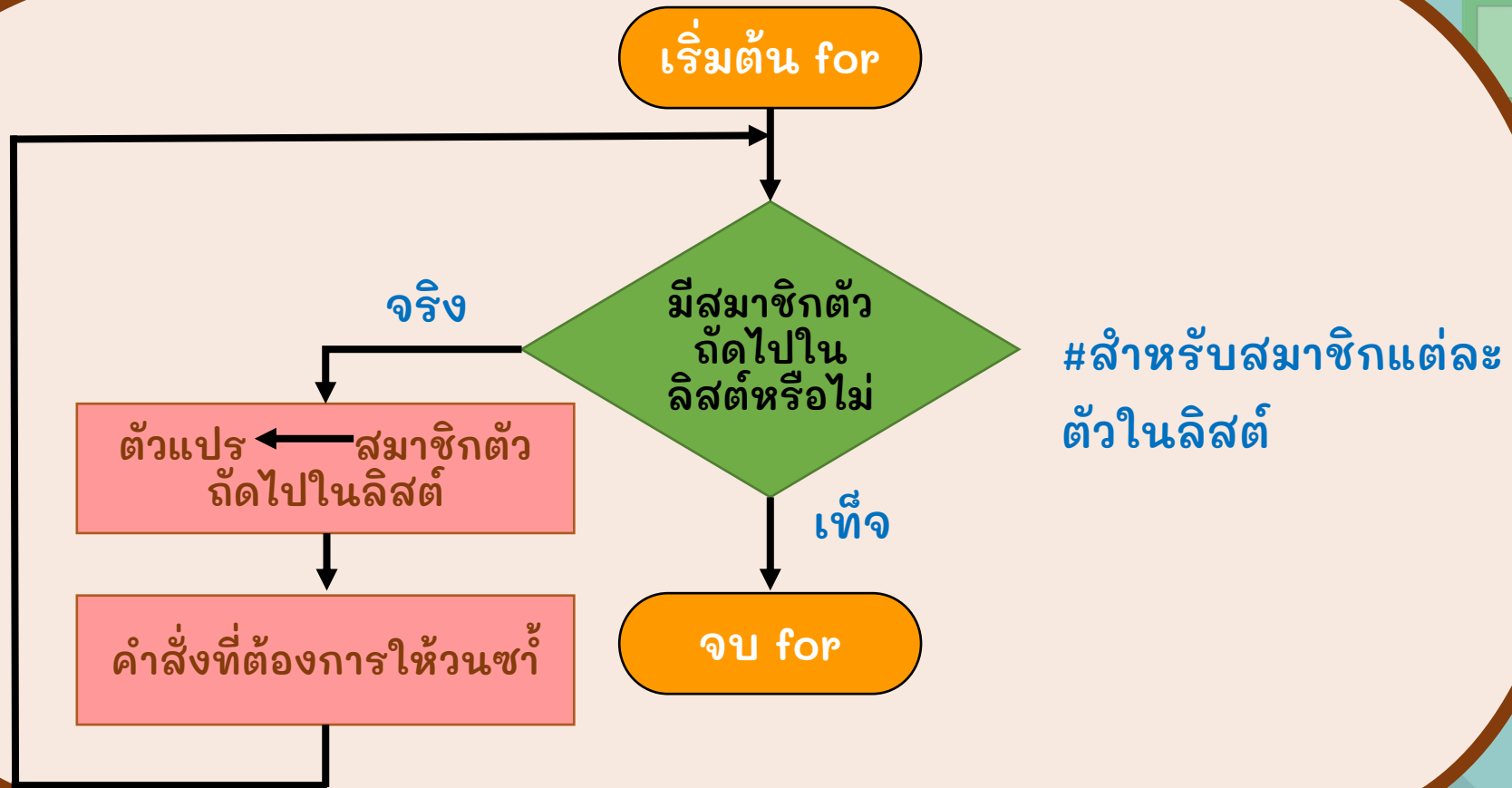
ชุดคำสั่ง # คำสั่งที่ต้องการให้ทำซ้ำ

ค่าใน ตัวแปร จะเปลี่ยนไปเรื่อยๆ

ตามลำดับใน ลิสต์



การทำงานแบบวนซ้ำ



ตัวอย่างที่ 3.11 echo

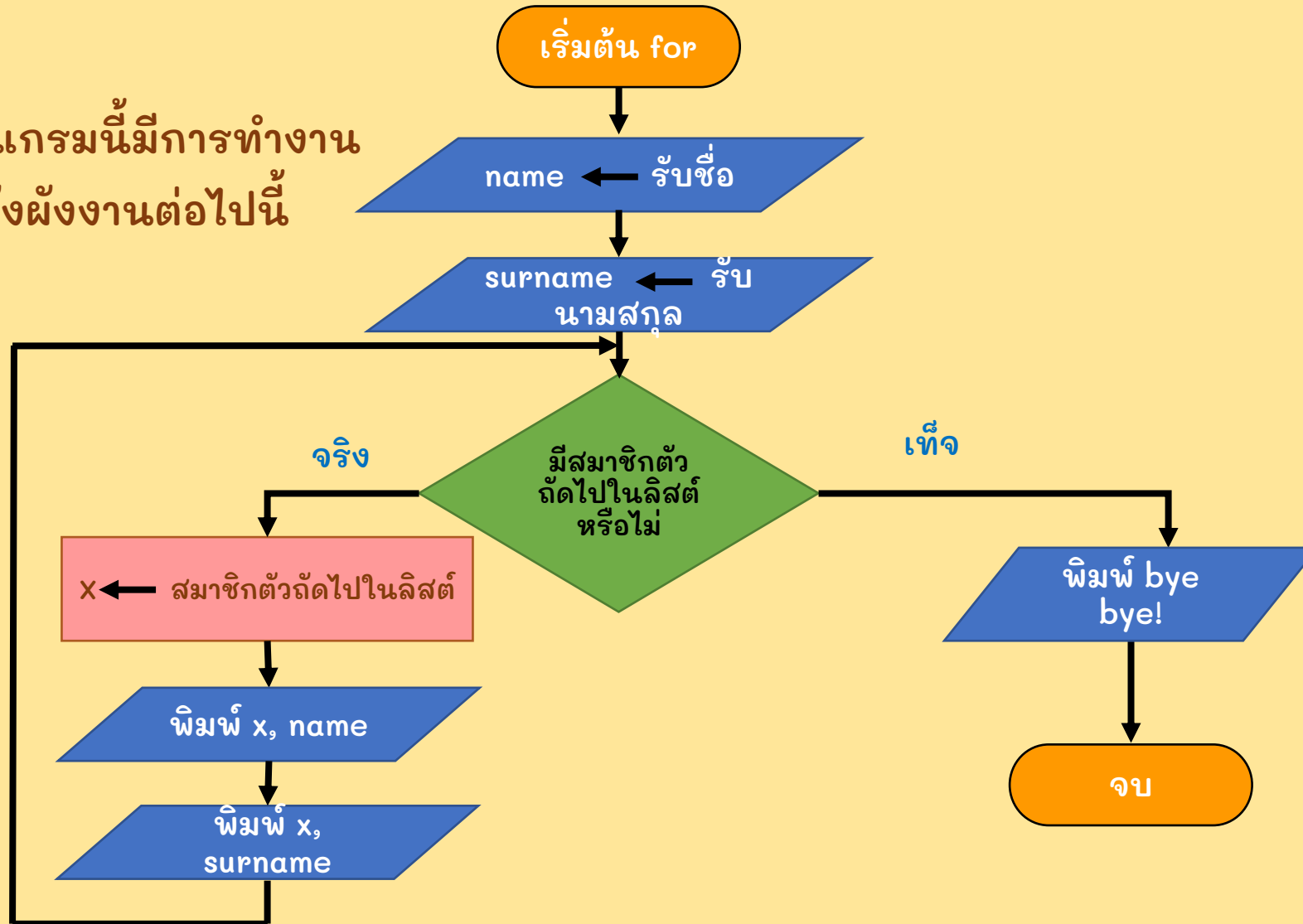
โปรแกรมต่อไปนี้ รับชื่อ และนามสกุล แล้วพิมพ์ซ้ำออกทางจอภาพจำนวน 5 ครั้ง

```
main.py x
1  ▶  name = input('Please enter your firstname')
2     surname = input('Please enter your lastname')
3     for x in range(5):
4         print(x, name)
5         print(x, surname)
6     print('bye bye!')
7
```



ตัวอย่างที่ 3.11 echo

โปรแกรมนี้มีการทำงาน
ดังผังงานต่อไปนี้



ตัวอย่างที่ 3.11 echo

ผลลัพธ์ที่ได้คือ

```
Run: main x
Please enter your firstname somchai
Please enter your lastname Santad
0 somchai
0 Santad
1 somchai
1 Santad
2 somchai
2 Santad
3 somchai
3 Santad
4 somchai
4 Santad
bye bye!
```



ตัวอย่างที่ 3.11 echo

ตัวอย่างที่ 3.11 อธิบายได้ดังนี้

1. บรรทัดที่ 3 ถึง 5 เป็นส่วนที่กำหนดให้มีการวนซ้ำ ให้สังเกตย่อหน้าของบรรทัดที่ 4 และ 5 เป็นการกำหนดว่าส่วนนี้จะต้องถูกวนซ้ำภายใต้คำสั่ง for ในบรรทัดที่ 3
2. คำสั่ง `range(5)` ในบรรทัดที่ 3 เป็นคำสั่งสร้างลิสต์จำนวนเต็ม 5 จำนวน ตั้งแต่ 0 , 1 , 2 , 3 , 4 ซึ่งในแต่ละรอบของการวนซ้ำ ตัวแปร `x` จะมีค่าเปลี่ยนไปตามลำดับแต่ละค่าในลิสต์
3. ผลลัพธ์ที่ได้ คือ คำสั่งในบรรทัดที่ 4 และ 5 จะทำงานซ้ำกันเป็นจำนวน 5 รอบ และแต่ละรอบ ค่า `x` ที่ถูกพิมพ์ออกมา จึงมีค่าเปลี่ยนไปตามค่าที่อยู่ในลิสต์



ตัวอย่างที่ 3.12 วาดวงกลม

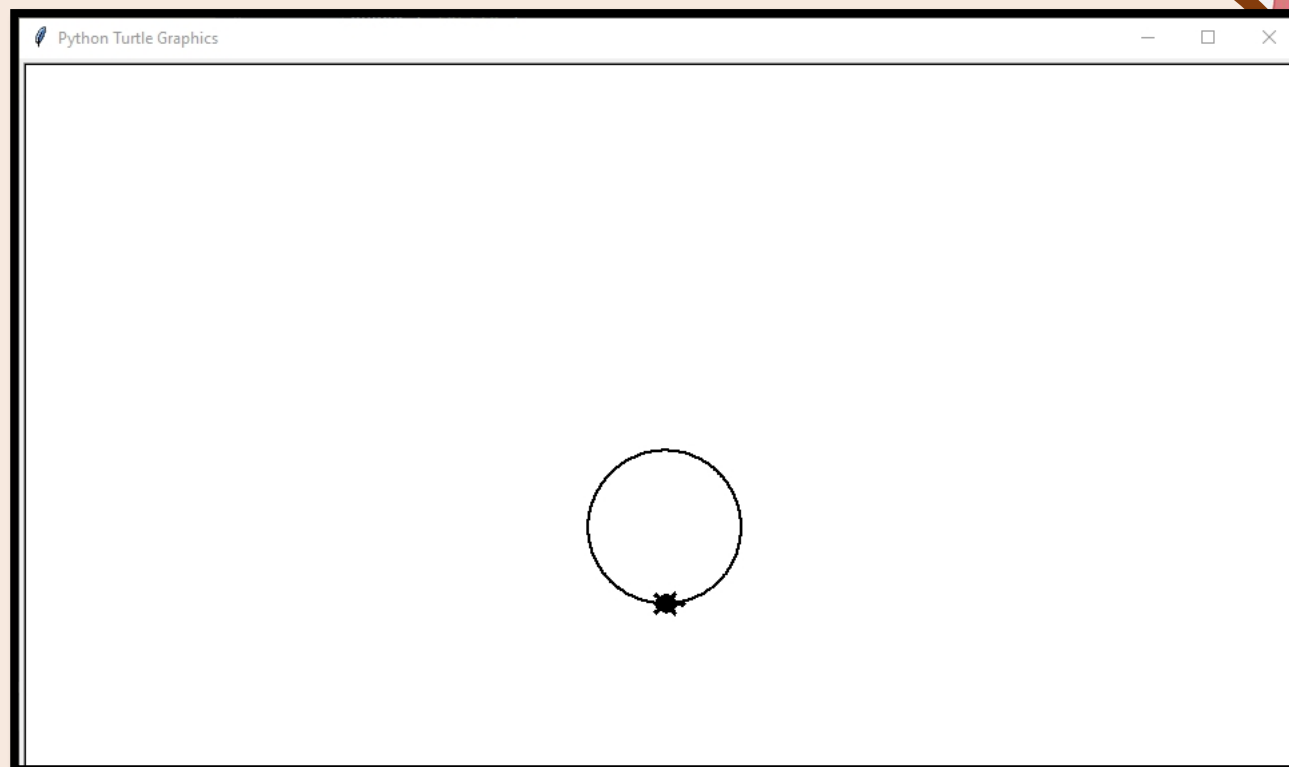
ป้อนคำสั่งไพทอน

```
main.py x
1  ▶ ● from turtle import *
2
3  shape_~("turtle")
4  pensize(2)
5  speed(1)
6  for i in range_~(360)_~:
7      forward(1)
8      left(1)
9  done()
10
11
```



ตัวอย่างที่ 3.12 วาดวงกลม

ผลลัพธ์ที่ได้คือ



ตัวอย่างที่ 3.13 รวมด้วยช่วยแชร์
แบบคำนวณซ้ำตามจำนวนรอบที่ผู้ใช้ระบุ

ป้อนคำสั่งไพทอน

```
main.py x
1  i = int(input('คุณต้องการคำนวณกี่รอบคะ'))
2  for count in range(i):
3      print('การคำนวณรอบที่', count+1)
4      totalPrice = int(input('ค่าอาหารทั้งหมด'))
5      number = int(input('จำนวนผู้รับประทานอาหาร'))
6      avg = totalPrice/number
7      print('จ่ายค่าอาหารคนละ', avg, 'บาท')
8      print('โปรแกรมทำงานเสร็จแล้ว')
9
10
```



ตัวอย่างที่ 3.13 ร่วมด้วยช่วยแชร์ แบบคำนวณซ้ำตามจำนวนรอบที่ผู้ใช้ระบุ

ผลลัพธ์ที่ได้

```
Run: main x
> คุณต้องการคำนวณกี่รอบคะ2
> การคำนวณรอบที่ 1
> ค่าอาหารทั้งหมด1250
> จำนวนผู้รับประทานอาหาร5
> จ่ายค่าอาหารคนละ 250.0 บาท
> การคำนวณรอบที่ 2
> ค่าอาหารทั้งหมด1222
> จำนวนผู้รับประทานอาหาร3
> จ่ายค่าอาหารคนละ 407.3333333333333 บาท
> โปรแกรมทำงานเสร็จแล้ว

Process finished with exit code 0
```



ตัวอย่างที่ 3.13 ร่วมด้วยช่วยแชร์ แบบคำนวณซ้ำตามจำนวนรอบที่ผู้ใช้ระบุ

ตัวอย่างที่ 3.13 อธิบายได้ดังนี้

1. ในขณะที่เขียนโปรแกรม นักเรียนจะไม่ทราบจำนวนรอบที่ต้องการ ดังนั้นจึงต้องรับจำนวนรอบจากผู้ใช้ ในบรรทัดที่ 1 ด้วยคำสั่ง `input()` แล้วจึงใช้คำสั่ง `int()` แปลงสตริงที่รับเข้าจาก `input()` ให้เป็นจำนวนเต็ม เพื่อนำไปกำหนดค่าให้กับตัวแปร `i`
2. บรรทัดที่ 3 ถึง 7 เป็นส่วนที่ต้องการให้ทำงานซ้ำ จึงนำไปใส่ไว้ภายใต้คำสั่ง `for` ในบรรทัดที่ 2 ซึ่งคำสั่ง `for` จะวนซ้ำเท่ากับจำนวน `i` รอบ โดยใช้ตัวแปร `count` นับรอบการทำงาน



ตัวอย่างที่ 3.13 ร่วมด้วยช่วยแชร์
แบบคำนวณซ้ำตามจำนวนรอบที่ผู้ใช้ระบุ

3. บรรทัดที่ 3 จะแสดงตัวเลขระบุ รอบที่ ของการวนซ้ำ ซึ่งต้องการให้แสดงข้อความ “การคำนวณรอบที่ 1” ไปจนถึง “การคำนวณรอบที่ i ” แต่เนื่องจาก `range()` ให้ค่าตั้งแต่ 0 , 1, ..., $i-1$ ดังนั้น เพื่อให้แสดงจำนวนรอบที่ถูกต้อง ตัวแปร `count` ที่เริ่มต้นด้วย 0 จึงต้องบวกด้วย 1



1

การทำงานแบบ มีทางเลือก



การทำงานแบบมีทางเลือก

ที่ผ่านมามีคำสั่งทุกคำสั่งที่ปรากฏในโปรแกรมจะมีการทำงานเสมอแต่ในบางสถานการณ์ต้องการให้คำสั่งบางคำสั่งทำงานเมื่อเงื่อนไขเป็นจริงตามที่ระบุเท่านั้น



ตัวอย่างที่ 3.14 มากไป-น้อยไป

โปรแกรมด้านขวารับค่าจาก
ผู้ใช้แล้วเก็บไว้ในตัวแปร x

- ถ้า x มีค่าน้อยกว่า 50 โปรแกรม
จะแสดงผลลัพท์ว่า Too small
- ถ้า x มีค่ามากกว่า 50 จะแสดง
ผลลัพท์ว่า Too large
- แต่ถ้า x มีค่าเท่ากับ 50 จะแสดง
ผลลัพท์ว่า Perfect

```
main.py x
1  x = int(input('Enter a number:'))
2      if x < 50:
3          print("Too small")
4      if x > 50:
5          print("Too large")
6      if x == 50:
7          print("Perfect")
8
```



ตัวอย่างที่ 3.14 มากไป-น้อยไป

ตัวอย่างผลลัพธ์ที่ได้

```
Run: main x
Enter a number:23
Too small
Process finished with exit code 0
```

x มีค่าน้อยกว่า 50

```
Run: main x
Enter a number:55
Too large
Process finished with exit code 0
```

x มีค่ามากกว่า 50

```
Run: main x
Enter a number:50
Perfect
Process finished with exit code 0
```

x มีค่าเท่ากับ 50

จะสังเกตเห็นว่าในตัวอย่างข้างต้นเงื่อนไข $x < 50$, $x > 50$ และ $x == 50$ จะเป็นตัวกำกับว่าคำสั่งภายใต้ 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
($!=$)



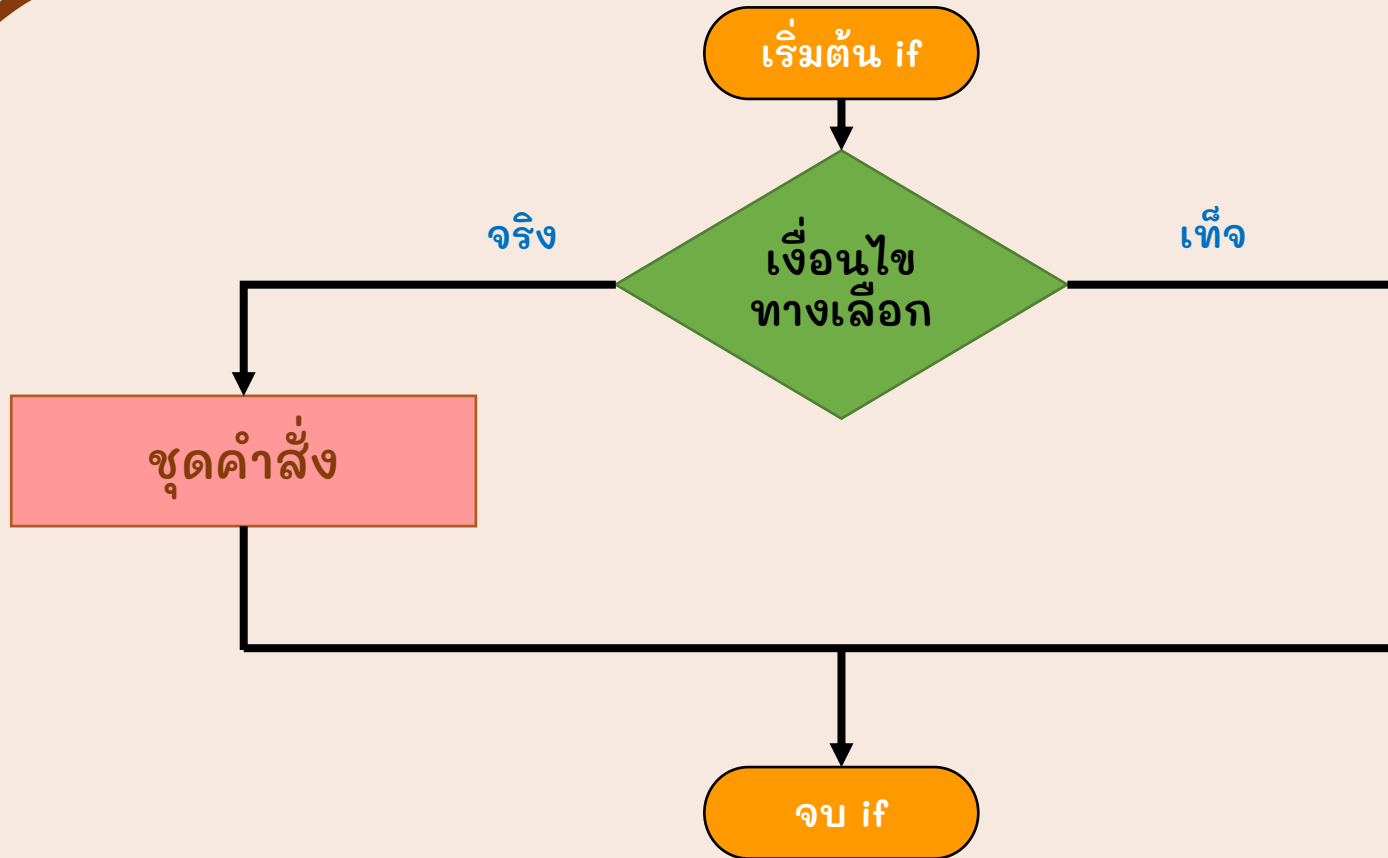
การทำงานแบบมีทางเลือก

การทำงานแบบมีทางเลือก มีประโยชน์กรณีที่ เราต้องการให้โปรแกรมทำการตัดสินใจบางอย่าง แล้ว เลือกทำงานชุดคำสั่งตามผลของเงื่อนไขที่กำหนดไว้ โดยมีรูปแบบคำสั่งแบบมีทางเลือก (selection statement) ในการใช้งานเบื้องต้นคือ

if เงื่อนไขทางเลือก :
ชุดคำสั่ง



การทำงานแบบมีทางเลือก



คำสั่ง if จะประเมินค่าความเป็นจริงของเงื่อนไขก่อน ถ้าเป็นจริงก็จะทำงานตามคำสั่งในชุดคำสั่ง ไม่เช่นนั้นจะข้ามไปทำงานในคำสั่งถัดไป



ตัวอย่าง 3.15 ทายใจ

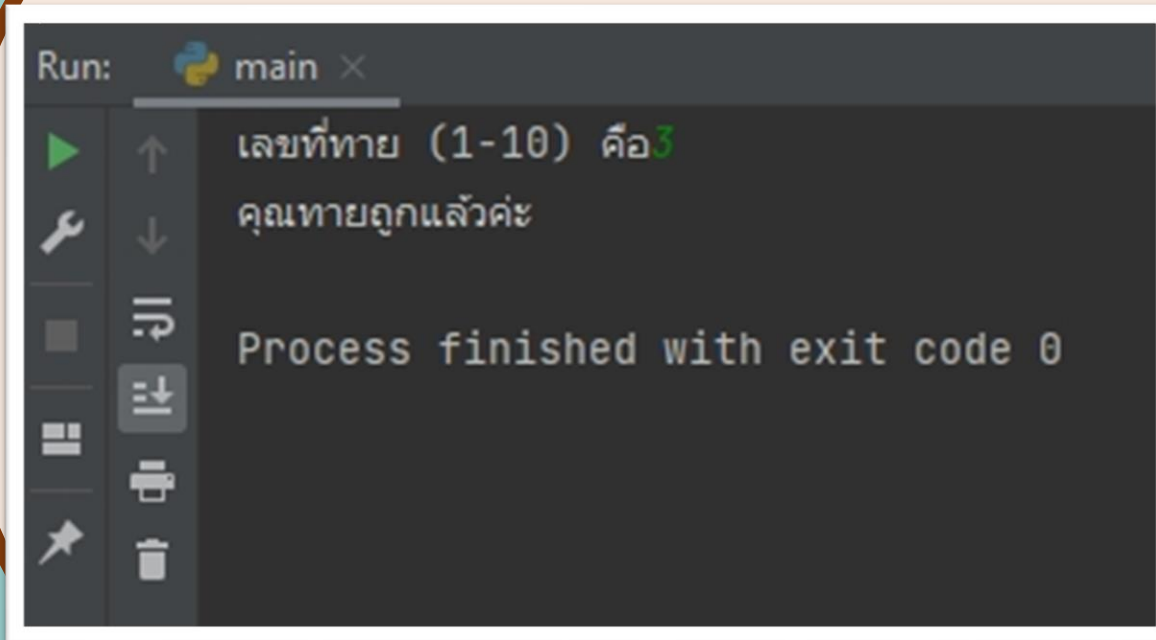
เขียนโปรแกรมไพทอนเพื่อให้เพื่อนเล่นเกมทายตัวเลข ดังนี้

```
main.py x
1  puzzle = 3
2  guess = int(input('เลขที่ทาย (1-10) คือ'))
3  if puzzle == guess:
4      print('คุณทายถูกแล้วค่ะ')
5
6
```



ตัวอย่าง 3.15 ทายใจ

ผลลัพธ์ที่ได้คือ



```
Run: main ×
เลขที่ทาย (1-10) คือ3
คุณทายถูกแล้วค่ะ

Process finished with exit code 0
```

ตัวอย่างที่ 3.15 อธิบายได้ดังนี้

1. บรรทัดที่ 1 กำหนดตัวเลขที่ให้ทายเก็บไว้ในตัวแปร puzzle
2. บรรทัดที่ 2 รับข้อมูลเข้าจากผู้ถ่ายเก็บไว้ในตัวแปร guess
3. ถ้าเงื่อนไขทางเลือก `puzzle == guess` ในคำสั่ง `if` ในบรรทัดที่ 3 เป็นจริง นั่นหมายความว่าทายตัวเลขถูกต้อง โปรแกรมจะแสดงข้อความตามคำสั่ง `print`



การทำงานแบบมีทางเลือก

ในบางสถานการณ์ของการเขียนโปรแกรมที่ต้องการให้โปรแกรมทำงานมากกว่าหนึ่งทางเลือก เช่น ถ้าโปรแกรมในตัวอย่างที่ 3.15 สามารถบอกผลของการทายตัวเลขเมื่อทายไม่ถูกต้องได้ด้วย ก็จะน่าสนใจขึ้น ภาษาไพทอน มีคำสั่ง `if-else` ให้ใช้ โดยมีรูปแบบการใช้ดังนี้

if เงื่อนไขทางเลือก :

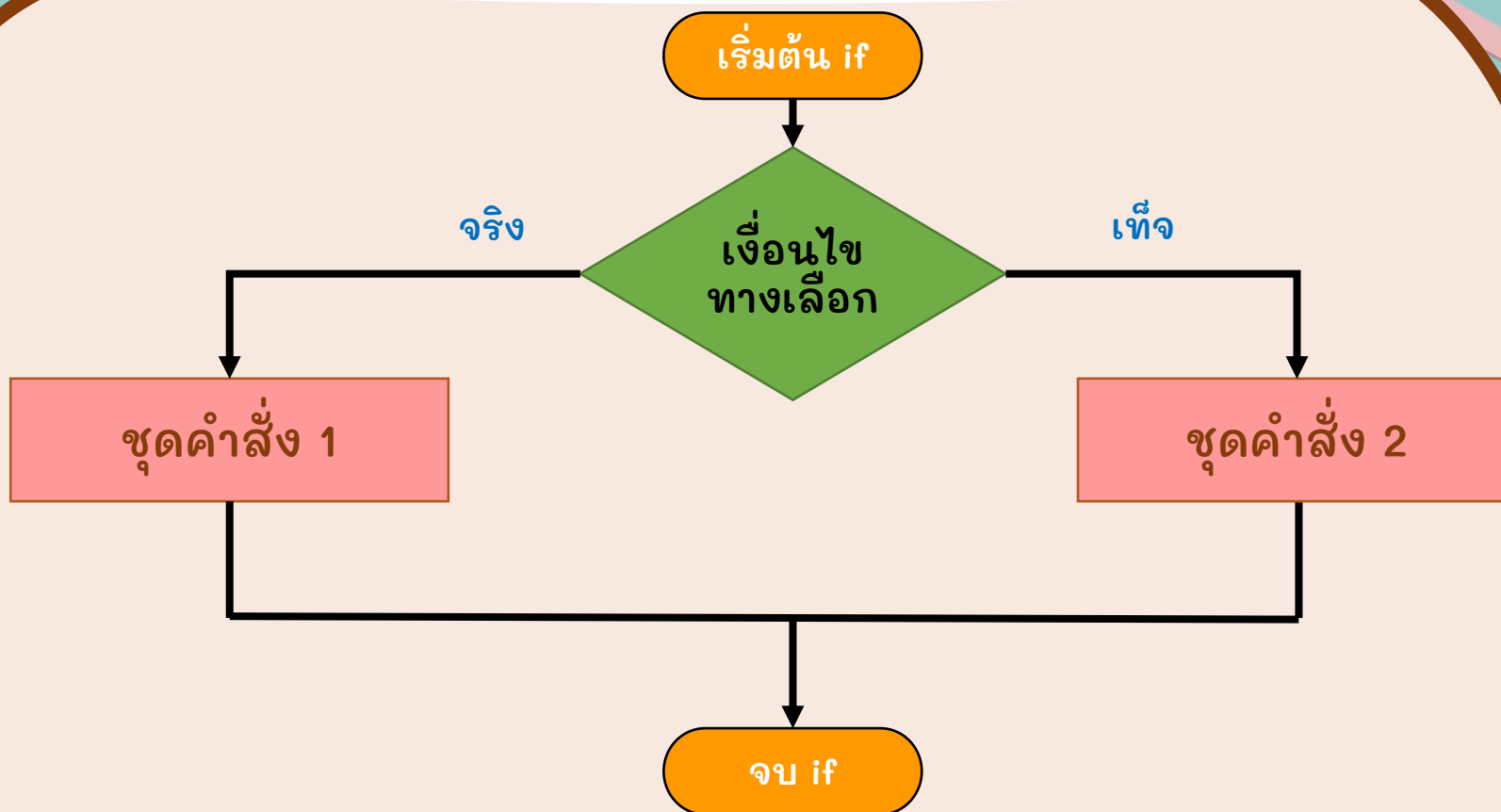
 ชุดคำสั่ง 1

else:

 ชุดคำสั่ง 2



การทำงานแบบมีทางเลือก



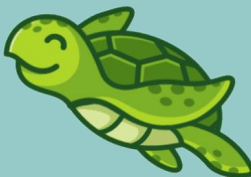
โดยถ้าเงื่อนไขทางเลือกเป็นจริงแล้ว จะทำงานในชุดคำสั่ง 1
ไม่เช่นนั้นจะทำงานในชุดคำสั่ง 2



ตัวอย่าง 3.16 ทายใจ 2

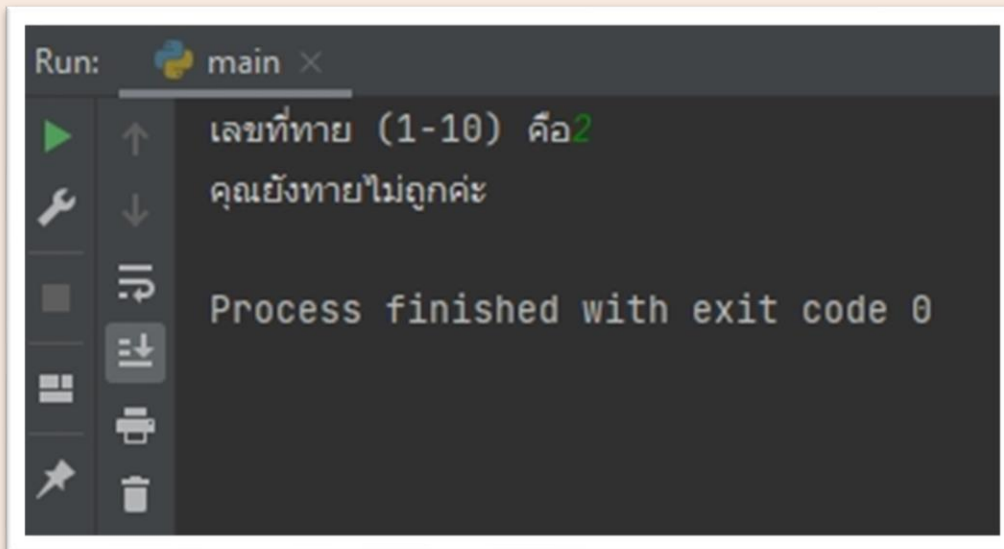
ปรับปรุงโปรแกรมในตัวอย่างที่ 3.15 ให้สามารถแจ้งผลของการทายตัวเลขที่ไม่ถูกต้องได้

```
main.py x
1 puzzle = 3
2 guess = int(input('เลขที่ทาย (1-10) คือ'))
3 if puzzle == guess:
4     print('คุณทายถูกแล้วค่ะ')
5 else:
6     print('คุณยังทายไม่ถูกค่ะ')
7
8
```



ตัวอย่าง 3.15 ทายใจ

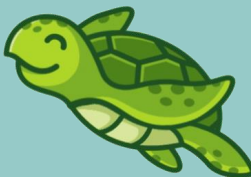
ผลลัพธ์ที่ได้คือ



```
Run: main x
> เลขที่ทาย (1-10) คือ2
> คุณยังทายไม่ถูกค่ะ
> Process finished with exit code 0
```

ตัวอย่างที่ 3.16 อธิบายได้ดังนี้

ต่อเนื่องจากตัวอย่างที่ 3.15 หากผู้ใช้ป้อนตัวเลขที่ทายไม่ถูกต้อง จะทำให้ผลของเงื่อนไขทางเลือกในคำสั่ง if บรรทัดที่ 3 เป็นเท็จ คำสั่งในบรรทัดที่ 4 จะถูกข้ามไป และคำสั่งในบรรทัดที่ 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)
```





สรุปท้ายบท



การเขียนโปรแกรมไพทอนจะใช้ไอดีอี เป็นเครื่องมือช่วยในการพัฒนาโปรแกรม ซึ่งประกอบด้วยเครื่องมือแก้ไขโปรแกรมต้นฉบับ เครื่องมือแก้ไขจุดบกพร่องของโปรแกรม และเครื่องมือที่ช่วยรันโปรแกรมไอดีอีไพทอนโดยทั่วไปจะทำงานได้ในโหมดอิมมีเดียทและโหมดสคริปต์ บทนี้ได้อธิบายคำสั่งพื้นฐาน เช่น `print ()` ใช้สำหรับแสดงค่าข้อมูลออกทางจอภาพ และ `input ()` ทำหน้าที่รับข้อมูลเข้าจากคีย์บอร์ด





สรุปท้ายบท



นอกจากนี้ยังส่งเสริมให้เรียนรู้การใช้งานตัวแปร ซึ่งตัวแปรใช้ในการอ้างถึงค่าข้อมูล โดยตัวแปรจะถูกกำหนดค่าด้วยเครื่องหมายเท่ากับ (=) การกำหนดชนิดข้อมูลสตริงและจำนวน รวมถึงการแปลงชนิดข้อมูล การใช้งานโมดูล turtle เพื่อวาดรูป คำสั่งวนซ้ำและคำสั่งแบบมีทางเลือก การเขียนนิพจน์ทางคณิตศาสตร์ในไพทอน การนำไพทอนมาใช้เป็นเครื่องมือประยุกต์แก้ปัญหาทางคณิตศาสตร์หรือวิทยาศาสตร์ โดยพื้นฐานนี้จะนำไปสู่การเขียนโปรแกรมคอมพิวเตอร์ในระดับสูงต่อไป





แบบทดสอบ 10 ข้อ 10 คะแนน

เริ่มทำแบบทดสอบ

1

การสั่งโปรแกรมทำงาน
ตามคำสั่งทั้งหมดตามลำดับ คือข้อใด

ก

การดำเนินการเปรียบเทียบ

ข

ไฟล์โปรแกรมไพทอน
ที่รวบรวมฟังก์ชันการทำงาน

ค

การรับข้อมูลจากผู้ใช้ผ่านคีย์บอร์ด

ง

การรันโปรแกรม



2

ข้อใดคือความหมายของโมดูล (module)

ก

ไฟล์โปรแกรมไพทอน
ที่รวบรวมฟังก์ชันการทำงาน

ข

การรันโปรแกรม

ค

การดำเนินการเปรียบเทียบ

ง

การรับข้อมูลจากผู้ใช้ผ่านคีย์บอร์ด



3

การเรียกใช้โมดูล turtle
ต้องป้อนคำสั่งใด

ก

`from turtle import *`

ข

`color("green")`

ค

`Shape('circle')`

ง

`pensize(5)`



4

กำหนดขนาดความหนาของเส้น
ในโมดูล turtle ต้องป้อนคำสั่งใด

ก

`color("green")`

ข

`Shape('circle')`

ค

`pensize(5)`

ง

`from turtle import *`



5

ระบุให้เต่าเปลี่ยนรูปร่างไปเป็นอย่างอื่น
ในโมดูล turtle ต้องป้อนคำสั่งใด

ก

`pensize(5)`

ข

`from turtle import *`

ค

`Shape('circle')`

ง

`color("green")`



6

เปลี่ยนสีเส้นที่เต่าลาก
ในโมดูล turtle ต้องป้อนคำสั่งใด

ก

Shape('circle')

ข

color("green")

ค

from turtle import *

ง

pensize(5)



7

ข้อใดคือคำสั่งการทำงานแบบวนซ้ำ

ก

if เงื่อนไขทางเลือก :
ชุดคำสั่ง

ข

for ตัวแปร in ลิสต์ :

ค

randint ()

ง

random



8

ข้อใดคือคำสั่งการทำงานแบบมีทางเลือก

ก

random

ข

randint ()

ค

for ตัวแปร in ลิสต์ :

ง

if เงื่อนไขทางเลือก :
ชุดคำสั่ง



9

ตัวดำเนินการเปรียบเทียบ
(>=) คือข้อใด

ก

มากกว่า

ข

น้อยกว่าหรือเท่ากับ

ค

ไม่เท่ากับ

ง

มากกว่าหรือเท่ากับ



10

ตัวดำเนินการเปรียบเทียบ
(!=) คือข้อใด

ก

ไม่เท่ากับ

ข

มากกว่า

ค

มากกว่าหรือเท่ากับ

ง

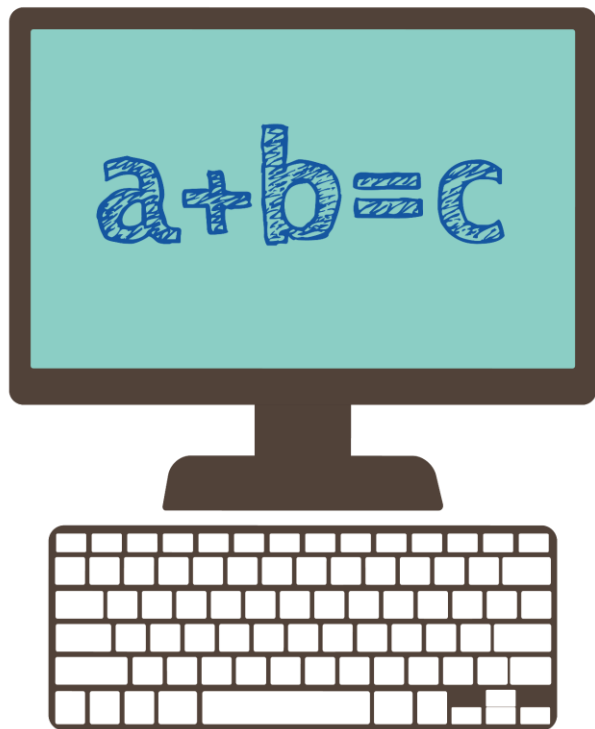
น้อยกว่าหรือเท่ากับ



บทเรียนคอมพิวเตอร์ช่วยสอน วิทยาการคำนวณ



คะแนนที่ได้



10

คะแนน





ผู้จัดทำ

นางสาว.....
ตำแหน่ง
โรงเรียน.....