



บทที่ 10

การแบ่งปัญหาใหญ่ออกเป็น ปัญหาย่อยและการสร้างลำดับ ขั้นตอนย่อย

(Dividing main problems into sub
problems)



จุดประสงค์การเรียนรู้

- เข้าใจความจำเป็นในการแบ่งปัญหาใหญ่เป็นปัญหาย่อย
- เข้าใจการทำงานแบบฟังก์ชัน (การแก้ปัญหาย่อย)
- เข้าใจและสามารถแยกฟังก์ชันออกจากปัญหาใหญ่ได้
- เข้าใจและสามารถเขียนผังงานแบบมีฟังก์ชันได้



หัวข้อเรื่อง

- ฟังก์ชันคืออะไร
- ข้อดีของฟังก์ชัน
- ตัวอย่าง



ฟังก์ชันคืออะไร

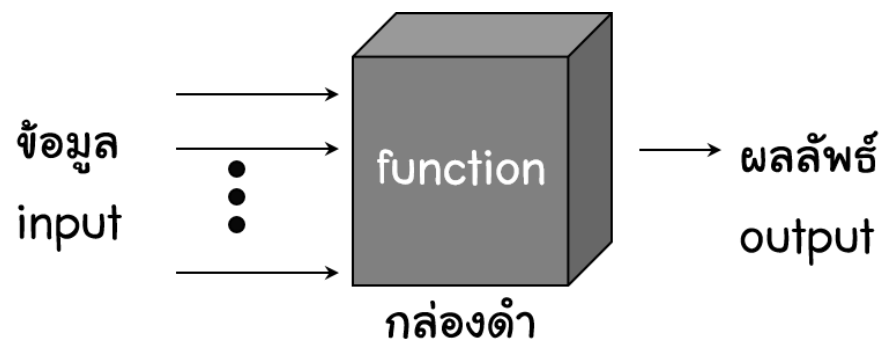
ฟังก์ชัน คือ การทำงานในส่วนย่อย ๆ ของงานหลัก ซึ่งงานหลักมักจะประกอบไปด้วยส่วนย่อย ๆ นี้หลายส่วน

หรืออาจพูดได้ว่า ฟังก์ชัน เกิดจากการแตกปัญหาใหญ่ออกเป็นปัญหาย่อย ๆ และแก้ปัญหาย่อยออกไปทีละส่วนจนกว่าจะหมด เพื่อลดความซับซ้อนของปัญหา และหาข้อผิดพลาดได้ง่าย

ที่มา : <https://th.wikipedia.org>

เข้าใจการทำงานของฟังก์ชัน

การทำงานของฟังก์ชัน เปรียบเสมือนกล่องดำ เมื่อใส่ข้อมูลเข้าไป
จำนวนหนึ่ง จะได้ผลลัพธ์จำนวนหนึ่ง ที่เกิดจากความสัมพันธ์ที่กำหนดไว้ภายใน
ฟังก์ชัน



ข้อดีของการใช้ฟังก์ชันในการแก้ปัญหา

- ช่วยให้แก้ปัญหาได้ง่ายขึ้น เพราะมีการแบ่งการทำงานหลักออกเป็นงานย่อย ๆ ซึ่งช่วยลดความซับซ้อน
- ช่วยให้เกิดโอกาสผิดพลาดน้อยลง เพราะทดสอบข้อผิดพลาดในงานส่วนงานย่อย ๆ มาแล้ว
- ช่วยให้สามารถนำการทำงานย่อย ๆ ไปใช้ในงานหลักอื่นได้ซึ่งทำให้ช่วยลดเวลาในการทำงานหลักอื่น ๆ

ตัวอย่างที่ 1 ตู้ขายสินค้าหยอดเหรียญ

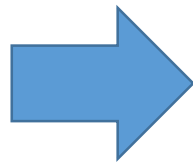
- ถ้ากดปุ่มน้ำอัดลมก็จะได้น้ำอัดลม
- ถ้ากดปุ่มน้ำเปล่าก็จะได้น้ำเปล่า
- แต่ผู้ซื้อจะไม่ทราบโครงสร้างการทำงานภายในของเครื่องจักร
- ฟังก์ชันเปรียบเสมือนโครงสร้างการทำงานภายในของเครื่องจักร



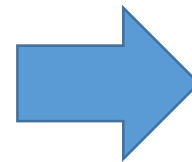
เปรียบเทียบตู้ขายสินค้าหยอดเหรียญกับฟังก์ชัน



หยอดเหรียญ + กดปุ่ม

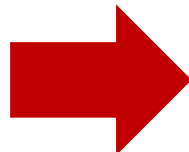


เครื่องจักรทำงาน

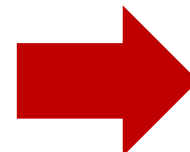


สินค้าจะออกมา

ใส่เงินหรือค่าหนึ่ง

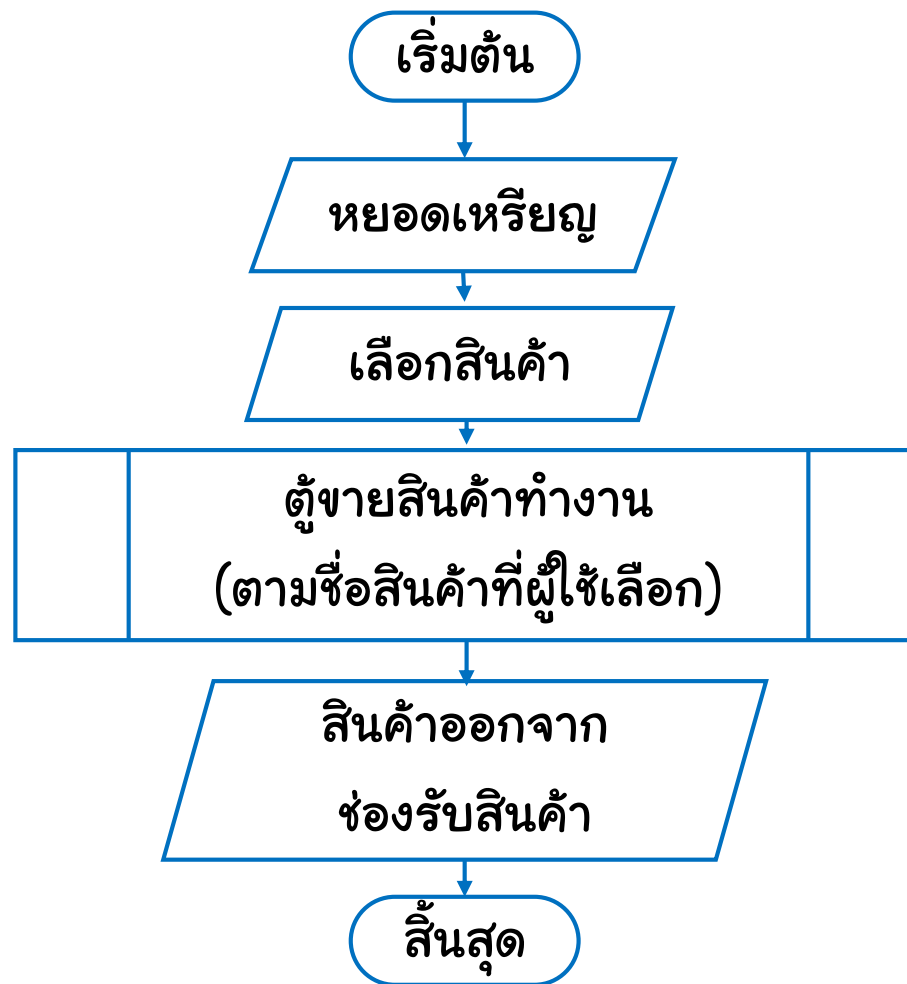


ฟังก์ชันทำงาน



ได้ผลลัพธ์ออกมา

จากตัวอย่างตู้ขายสินค้าอัตโนมัติ
สามารถเขียนผังงานได้ ดังนี้



ตัวอย่างที่ 2 หาค่ามากที่สุด และค่าน้อยสุด

- โปรแกรมสามารถรับตัวเลขได้ 5 จำนวน
- ส่งตัวเลข 5 จำนวนจากโปรแกรมหลักไปยังฟังก์ชัน 2 ฟังก์ชัน
- ฟังก์ชันหนึ่งทำการหาค่ามากที่สุดจากข้อมูลทั้ง 5 จำนวน
- อีกฟังก์ชันหนึ่งทำการหาค่าน้อยที่สุดจากข้อมูล 5 จำนวน
- สุดท้าย ทั้งสองฟังก์ชันส่งค่ามากที่สุด และค่าน้อยที่สุดที่หาได้ กลับมายังโปรแกรมหลัก
- โปรแกรมหลักแสดงผลลัพธ์ที่รับมาจากฟังก์ชัน

รับข้อมูลเข้า 5 จำนวน ดังนี้

2, 5, 90, 8, 12

จากนั้นส่งข้อมูลทั้ง 5 จำนวนไปยังฟังก์ชันสองฟังก์ชัน คือ

- ฟังก์ชันหาค่าที่มากที่สุด ชื่อฟังก์ชัน max
- ฟังก์ชันหาค่าที่น้อยที่สุด ชื่อฟังก์ชัน min



โปรแกรมหลัก

เริ่มต้น

รับค่า $n_1, n_2,$
 n_3, n_4, n_5

$m = \max(n_1, n_2, n_3, n_4, n_5)$

$n = \min(n_1, n_2, n_3, n_4, n_5)$

แสดงค่า m, n

สิ้นสุด

รับค่า 2, 5, 90, 8, 12

$m = \max(2, 5, 90, 8, 12)$

$n = \min(2, 5, 90, 8, 12)$

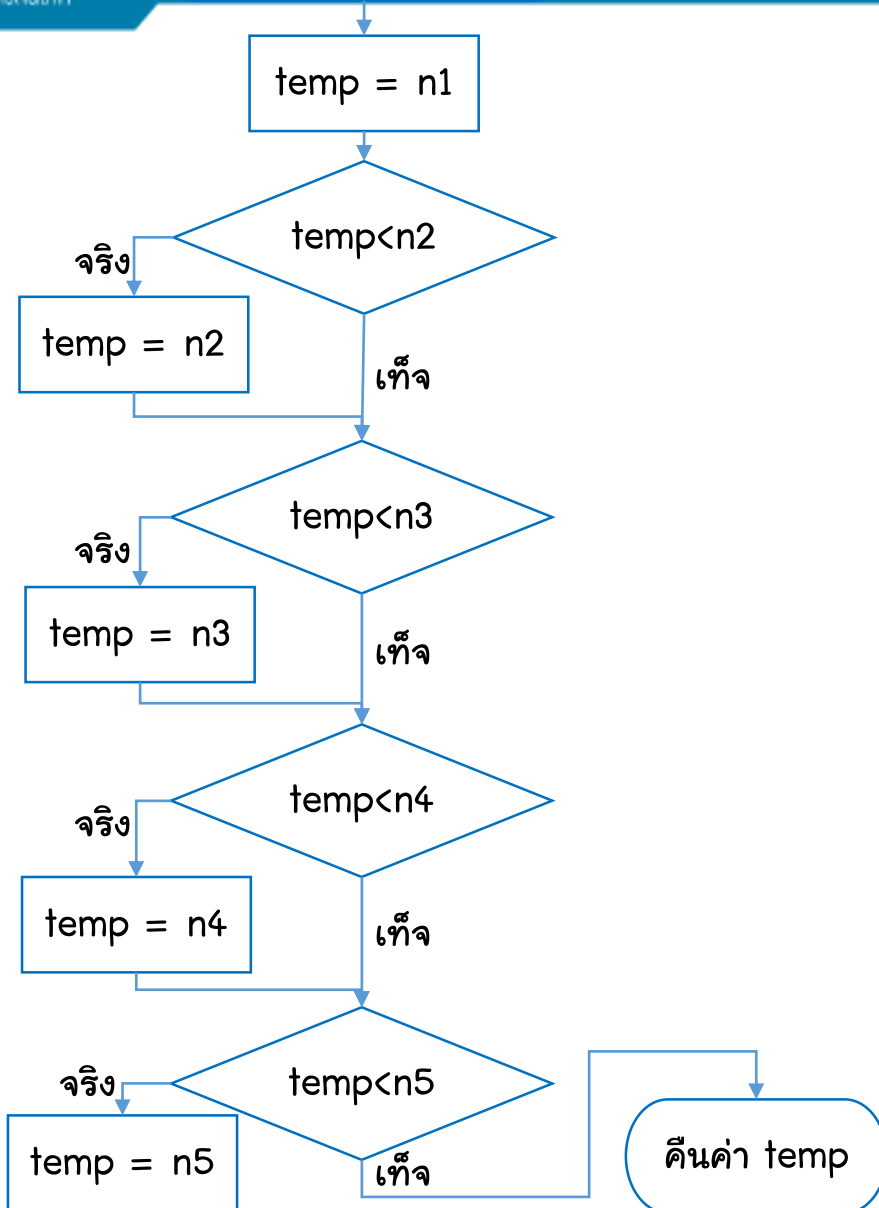
แสดงค่า m, n



$\max(n1, n2, n3, n4, n5)$

$\max(2, 5, 90, 8, 12)$

ฟังก์ชัน max



$\text{temp} = 2$

$2 < 5$

$\text{temp} = 5$

$5 < 90$

$\text{temp} = 90$

$90 < 8$

$90 < 12$

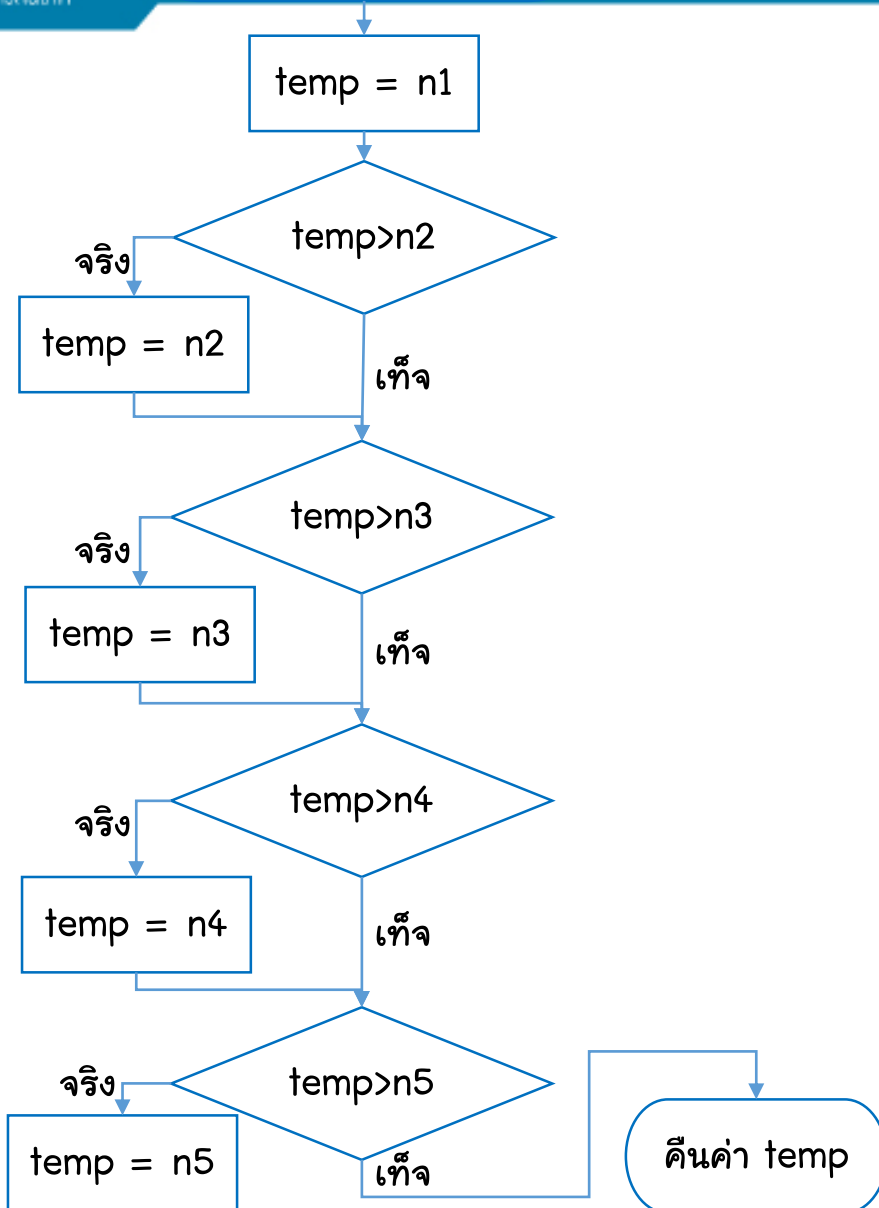
คืนค่า 90



$\min(n1, n2, n3, n4, n5)$

$\min(2, 5, 90, 8, 12)$

ฟังก์ชัน min



temp = 2

2 > 5

2 > 90

2 > 8

2 > 12

คืนค่า 2



โปรแกรมหลัก

เริ่มต้น

รับค่า $n_1, n_2,$
 n_3, n_4, n_5

$m = \max(n_1, n_2, n_3, n_4, n_5)$

$n = \min(n_1, n_2, n_3, n_4, n_5)$

แสดงค่า m, n

สิ้นสุด

รับค่า 2, 5, 90, 8, 12

$m = \max(2, 5, 90, 8, 12)$

$n = \min(2, 5, 90, 8, 12)$

แสดงค่า 90, 2

ตัวอย่างที่ 3 โรงงานตุ๊กตา

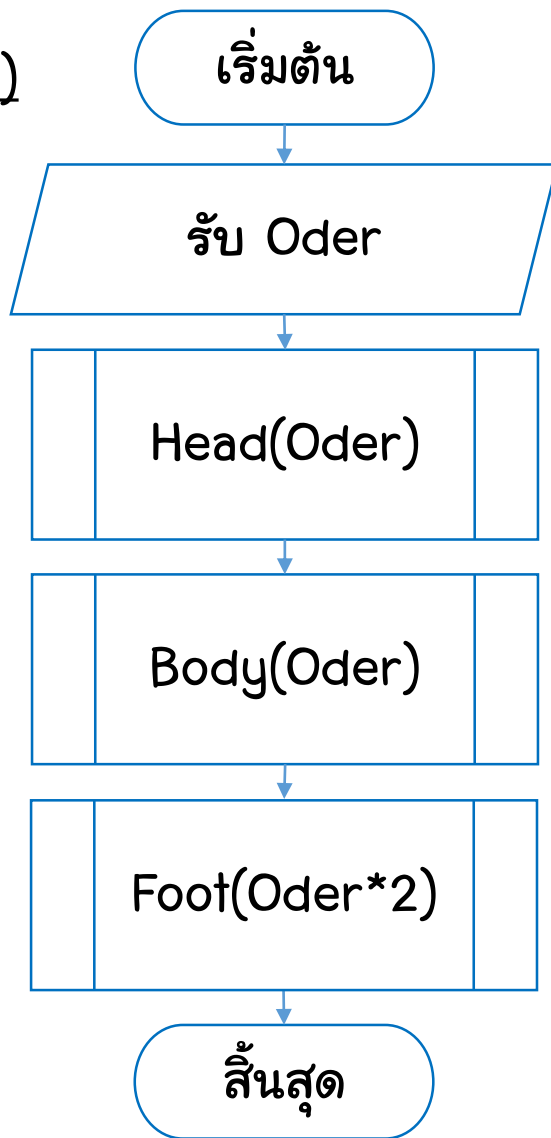
- เปรียบเทียบโรงงานผลิตตุ๊กตาเป็นโปรแกรมหลัก
- เปรียบเทียบแผนกเย็บชิ้นส่วนต่าง ๆ เป็นฟังก์ชัน
- โรงงานประกอบไปด้วยแผนกเย็บชิ้นส่วนต่าง ๆ ของตุ๊กตา คือ ส่วนหัว ส่วนตัว และส่วนเท้า
- แต่ละแผนกจะเย็บตามรูปแบบที่ถูกระบุกำหนดไว้แล้ว

- โรงงานเปิดตัวขึ้นด้วยเงินทุนที่ไม่มากนัก
- โรงงานผลิตตุ๊กตาแบบเดียวเท่านั้น คือ ตุ๊กตาหมี
- โรงงาน (โปรแกรมหลัก) จะต้องรับออเดอร์เป็นจำนวนตัวของตุ๊กตา
- โรงงาน (โปรแกรมหลัก) จะต้องจ่ายงานไปยังแผนก (ฟังก์ชัน) สำหรับเย็บส่วนหัว ส่วนตัว และส่วนเท้า ว่าจะใช้ส่วนต่าง ๆ ในปริมาณเท่าไร
- โดยจะต้องสั่งดังนี้
 - Head(1) หมายความว่า สั่งชิ้นส่วนหัว 1 ชิ้น
 - Body(1) หมายความว่า สั่งชิ้นส่วนตัว 1 ชิ้น
 - Foot(2) หมายความว่า สั่งชิ้นส่วนเท้า 2 ชิ้น หรือ 1 คู่ นั่นเอง

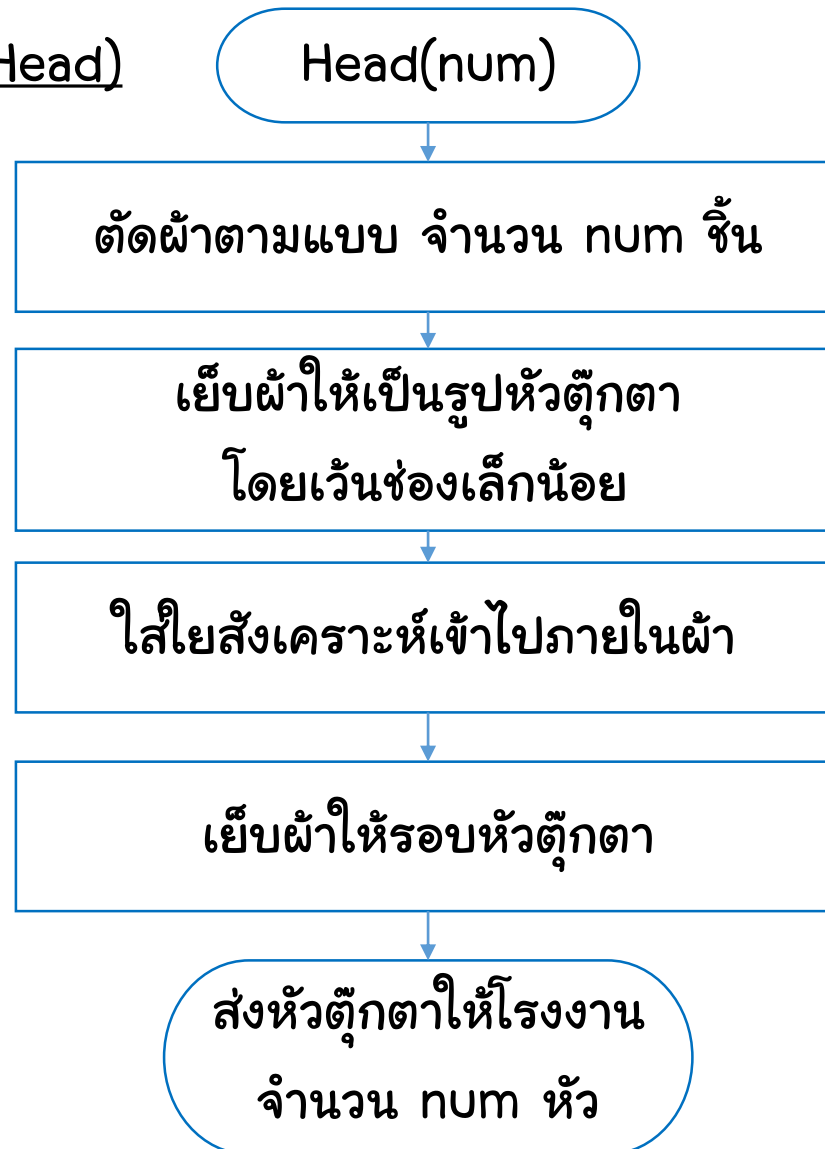




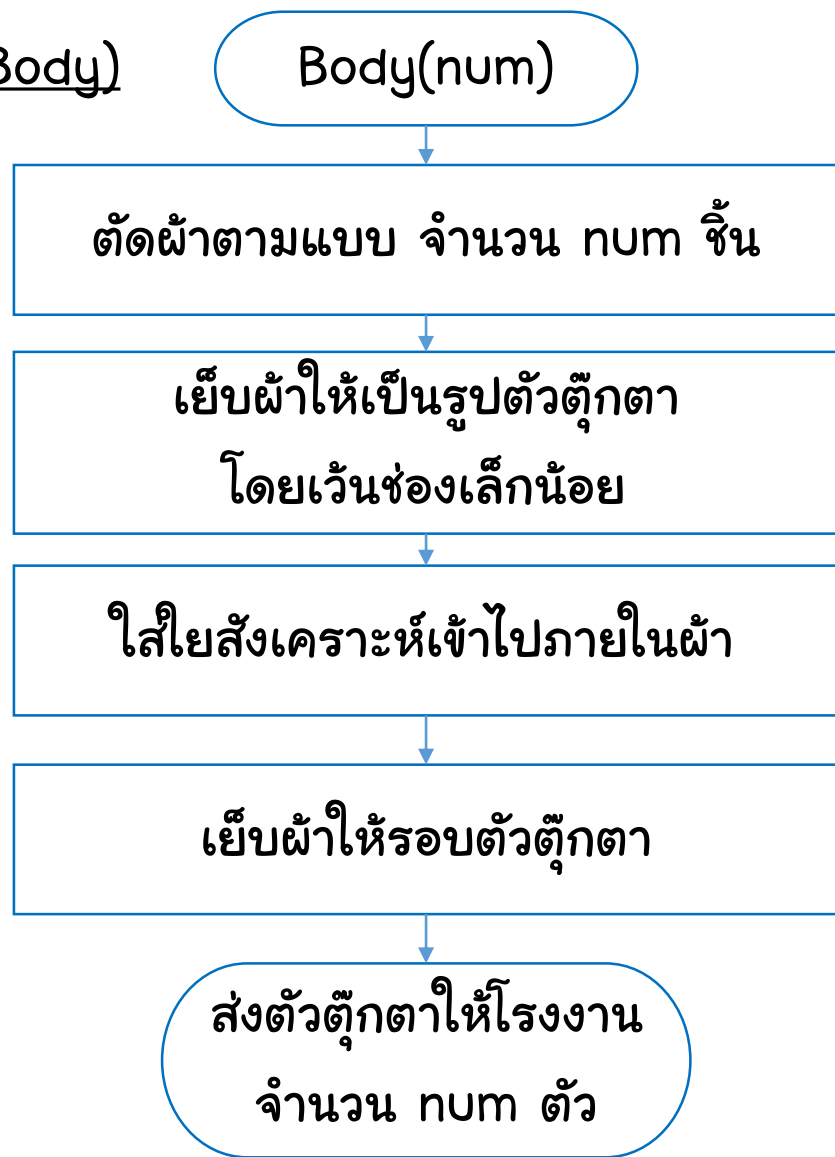
โรงงาน (โปรแกรมหลัก)



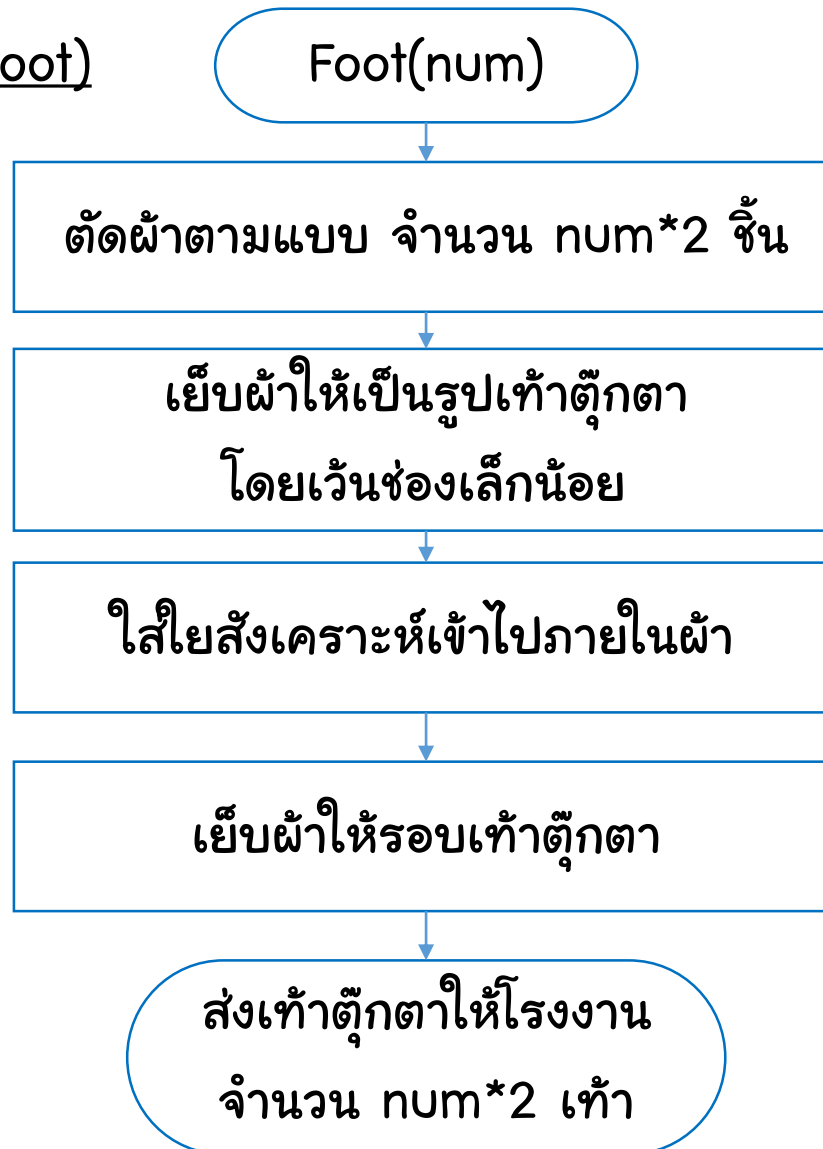
แผนกเย็บส่วนหัว (ฟังก์ชัน Head)



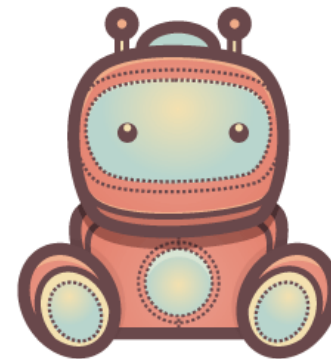
แผนกเย็บส่วนตัว (ฟังก์ชัน Body)



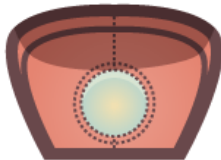
แผนกเย็บส่วนขา (ฟิงก์ชั้น Foot)



- ต่อมาโรงงานสะสมกำไรจากการผลิตตุ๊กตา ทำให้สามารถขยายโรงงานได้
- โรงงานผลิตตุ๊กตาหลายแบบมากขึ้น นอกจากตุ๊กตาหมีแล้ว โรงงานยังมีตุ๊กตาแบบอื่น ๆ อีก 3 แบบให้ลูกค้าเลือกสั่งได้ คือ ตุ๊กตากระต่าย ตุ๊กตาไดโนเสาร์ และตุ๊กตาหุ่นยนต์

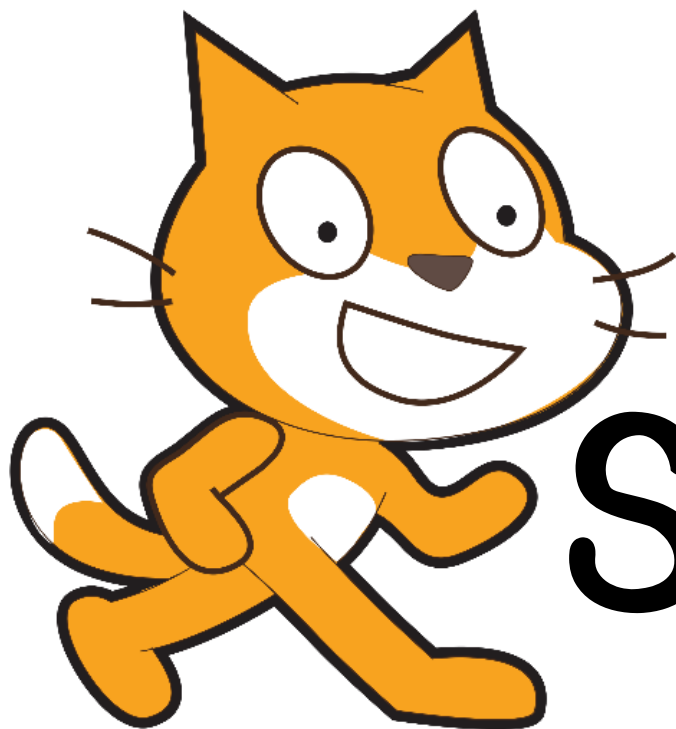


- โรงงาน (โปรแกรมหลัก) จะต้องจ่ายงานไปยังแผนก (ฟังก์ชัน) สำหรับแยกส่วนหัว ส่วนตัว และส่วนเท้า ว่าจะใช้ส่วนต่าง ๆ ในปริมาณเท่าไร
- โดยมีคีย์เวิร์ดในการสั่ง ดังนี้

หัว				
คีย์เวิร์ด	Bear	Rabbit	Dino	Robot
ตัว				
คีย์เวิร์ด	Bear	Rabbit	Dino	Robot
เท้า				
คีย์เวิร์ด	Bear	Rabbit	Dino	Robot

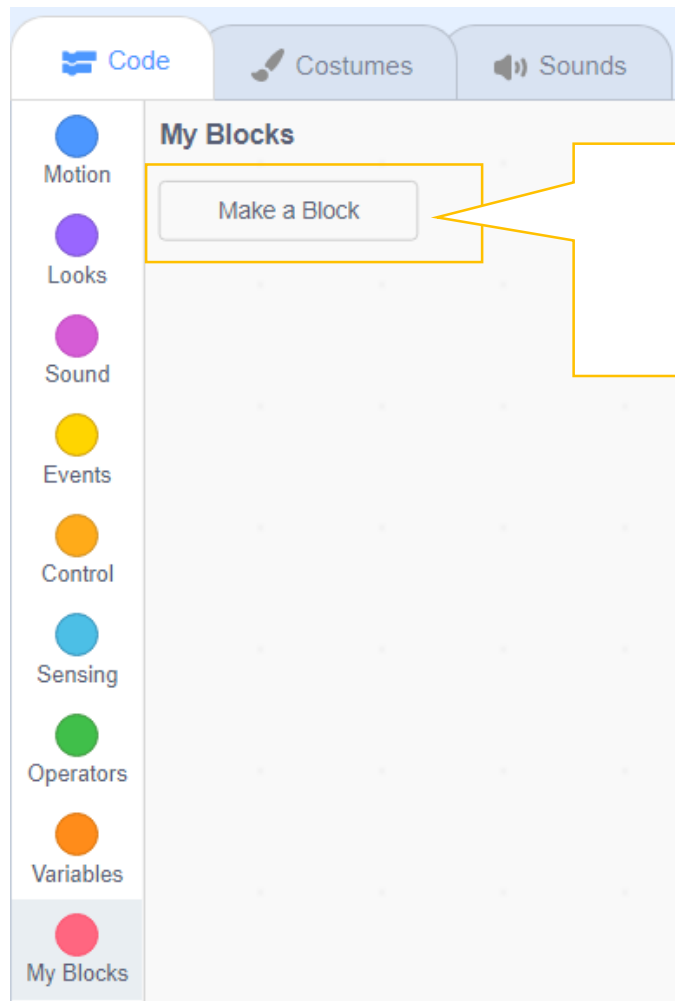
- ตัวอย่างคำสั่งผลิต
 - Head(Rabbit,1)
หมายความว่า สั่งผลิตชิ้นส่วนหัวรูปแบบ Rabbit จำนวน 1 ชิ้น
 - Body(Bear,1)
หมายความว่า สั่งผลิตชิ้นส่วนหัวรูปแบบ Bear จำนวน 1 ชิ้น
 - Foot(Dino,2)
• หมายความว่า สั่งผลิตชิ้นส่วนเท้ารูปแบบ Dino จำนวน 2 ชิ้น หรือ 1 คู่





Scratch

บล็อกคำสั่งสำหรับสร้างฟังก์ชัน



ไปที่ Make a Block



Make a Block

block name

Add an input
number or text

Add an input
boolean

text
Add a label

☐ Run without screen refresh

Cancel OK

ตั้งชื่อฟังก์ชัน

เพิ่มข้อมูลเข้าสำหรับ
ฟังก์ชัน

Make a Block

num1

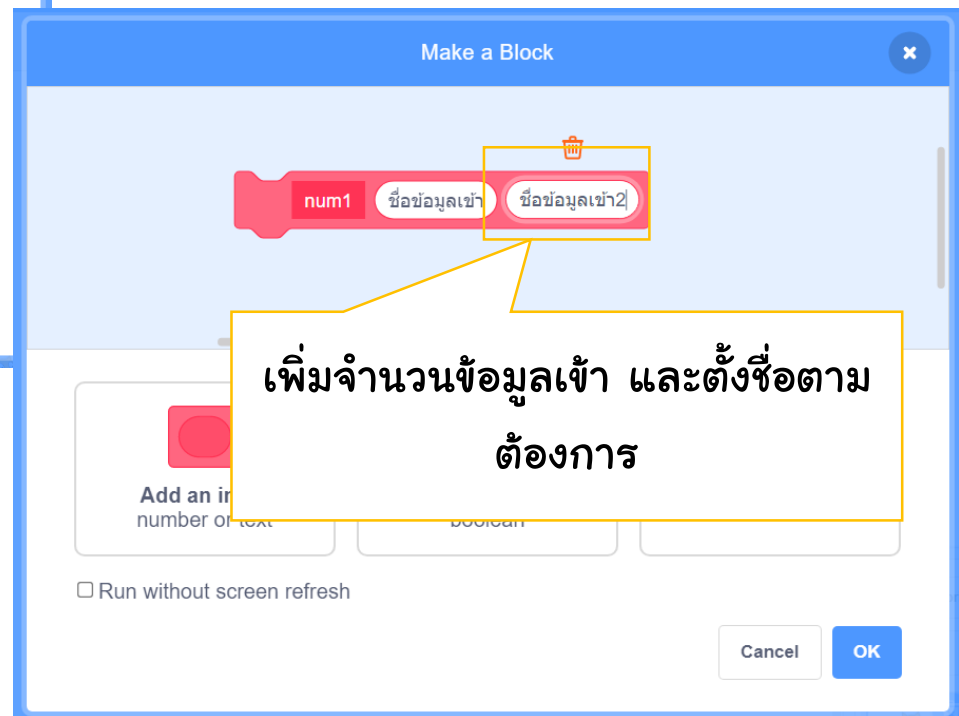
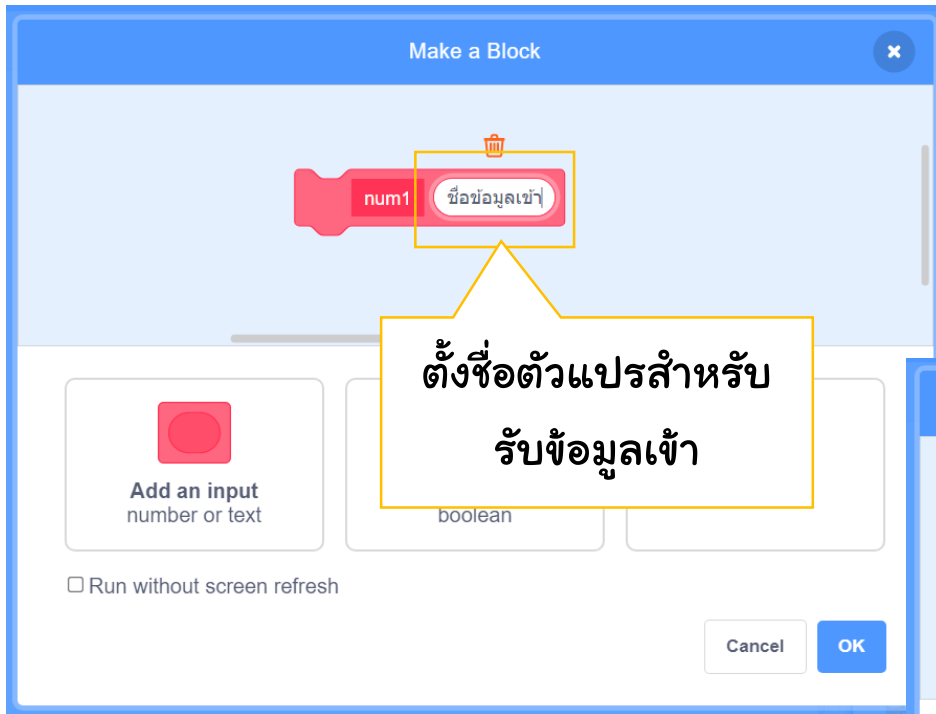
Add an input
number or text

Add an input
boolean

text
Add a label

☐ Run without screen refresh

Cancel OK





My Blocks

Make a Block

add 1 1

define add num1 num2

ผลลัพธ์หลังจากสร้างบล็อก
ฟังก์ชันเรียบร้อยแล้ว

Operators

pick random 1 to 10

set sum to num1 + num2

ใส่รายละเอียดภายในฟังก์ชัน



CodeCostumesSounds

MotionLooksSoundEventsControlSensingOperatorsVariablesMy Blocks

Operators

+−×÷

pick random 1 to 10

> 50< 50= 50

and or not

join apple bananaletter 1 of applelength of appleapple contains a ?

define add num1 num2

set sum to num1 + num2

when clicked

ask กรอกตัวเลข and wait

set n1 to answer

ask กรอกตัวเลขอีก 1 ตัว and wait

set n2 to answer

add n1 n2

say join n1 join + join n2 join = sum for 2 seconds

ตัวอย่างการเรียกใช้ฟังก์ชัน

หน้า | 30



Scratch interface showing a project titled "test_function" by logicalThinking. The project is a Scratch script for a Totoro character.

Code Editor:

```
define add num1 num2
  set sum to num1 + num2

when clicked
  ask กรอกตัวเลข and wait
  set n1 to answer
  ask กรอกตัวเลขอีก 1 ตัว and wait
  set n2 to answer
  add n1 n2
  say join n1 join + join n2 join = sum for 2 seconds
```

Stage: The stage displays a Totoro character (Sprite: totoro.jpg) on a dark blue background with a night sky and a tree. The Totoro is positioned at x: -94, y: -35, with a size of 60 and a direction of 90.

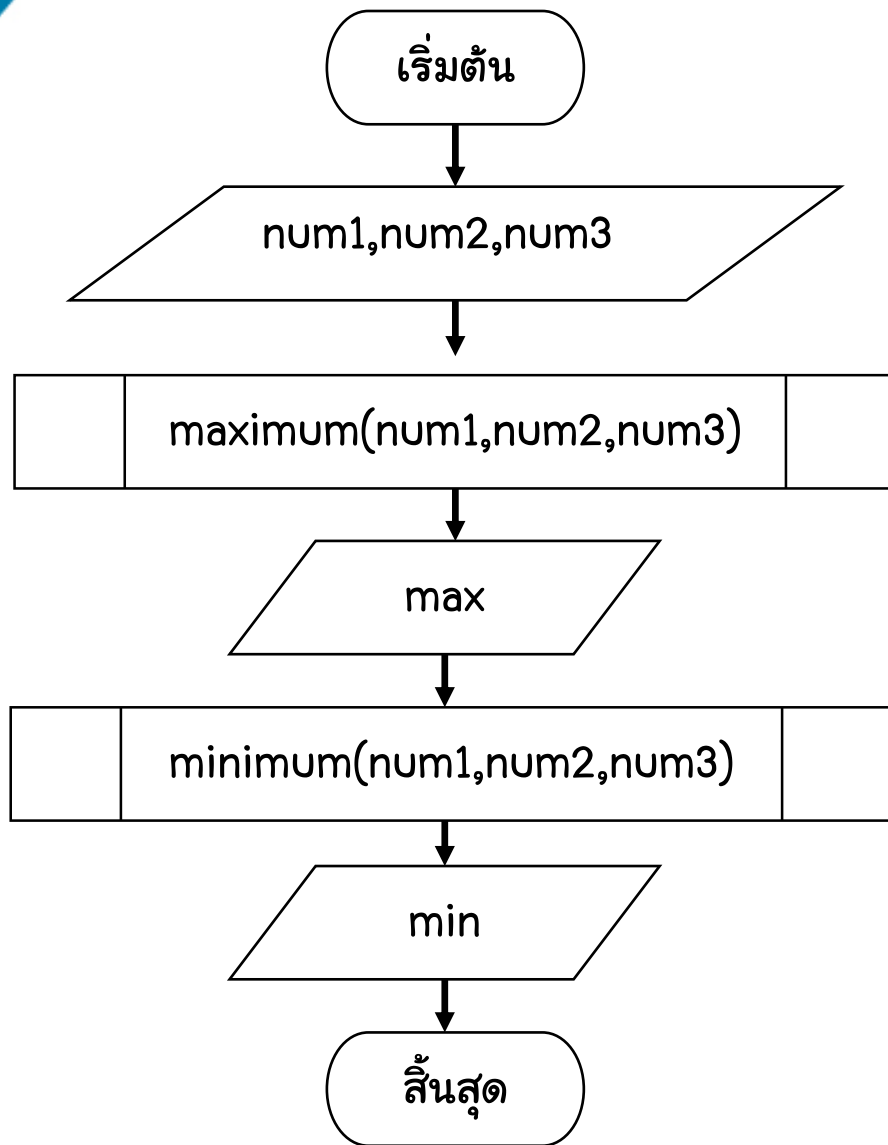
ชมตัวอย่างได้ที่ <https://scratch.mit.edu/projects/111957257/>

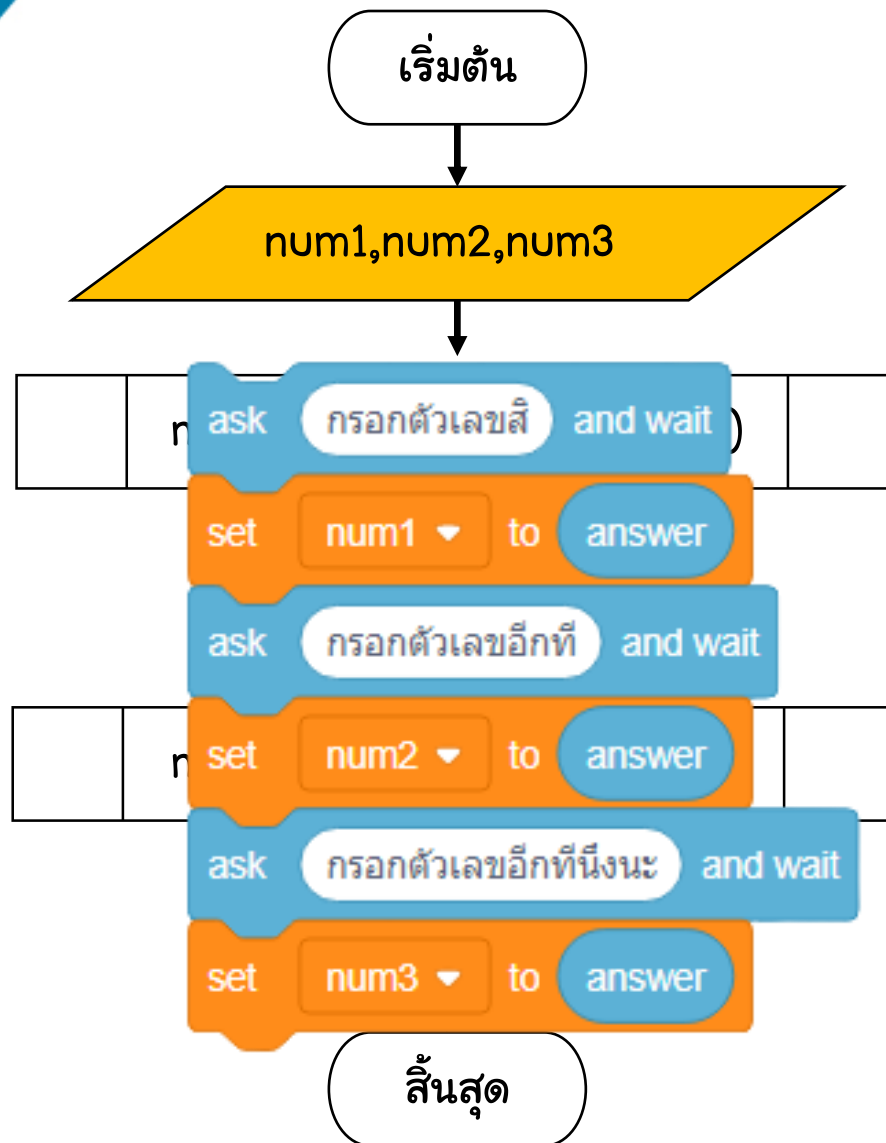
ตัวอย่างชิ้นงาน Scratch ที่มีการใช้ฟังก์ชัน

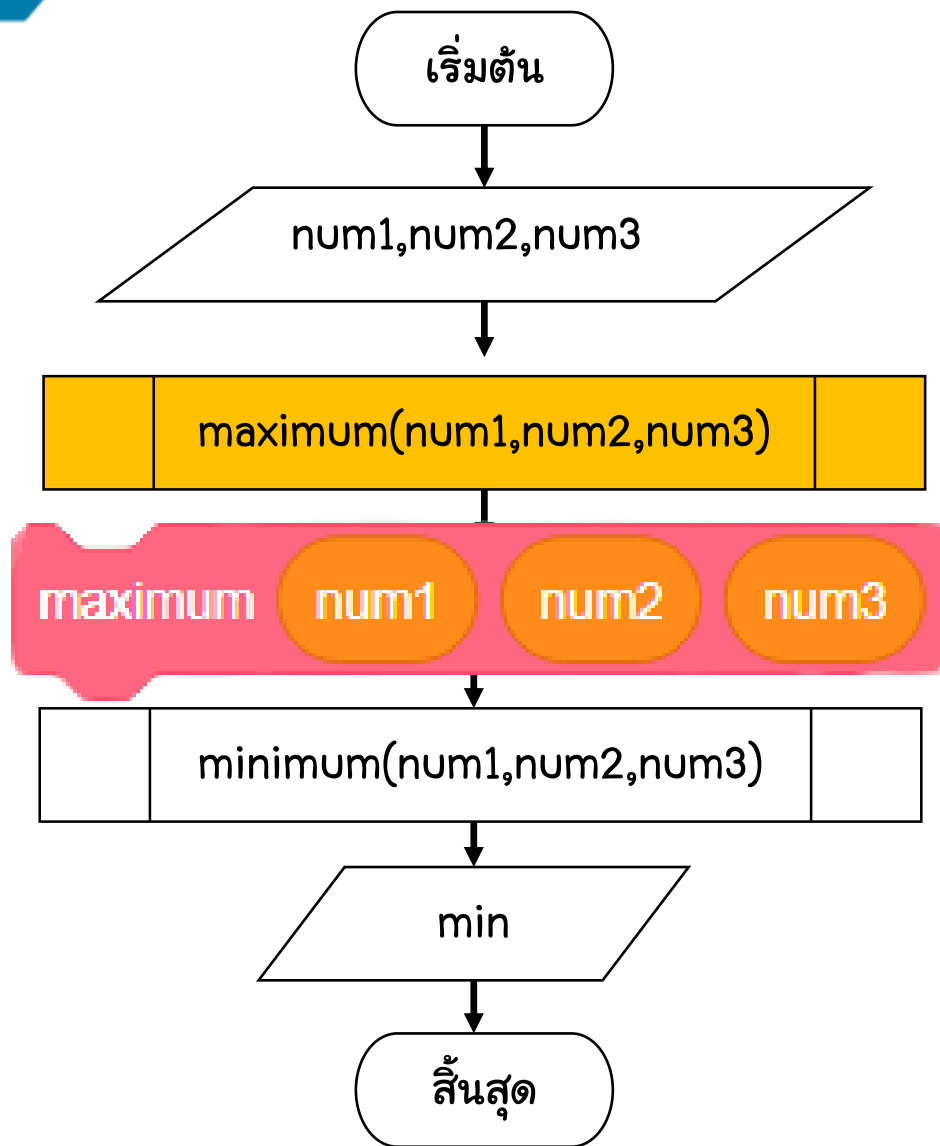
- รับข้อมูลเข้า 3 จำนวน
- ส่งข้อมูลทั้ง 3 จำนวนไปยังฟังก์ชันหาค่าที่มากที่สุด
- ส่งข้อมูลทั้ง 3 จำนวนไปยังฟังก์ชันหาค่าที่น้อยที่สุด
- แสดงผลลัพธ์ค่าที่มากที่สุด และค่าที่น้อยที่สุด

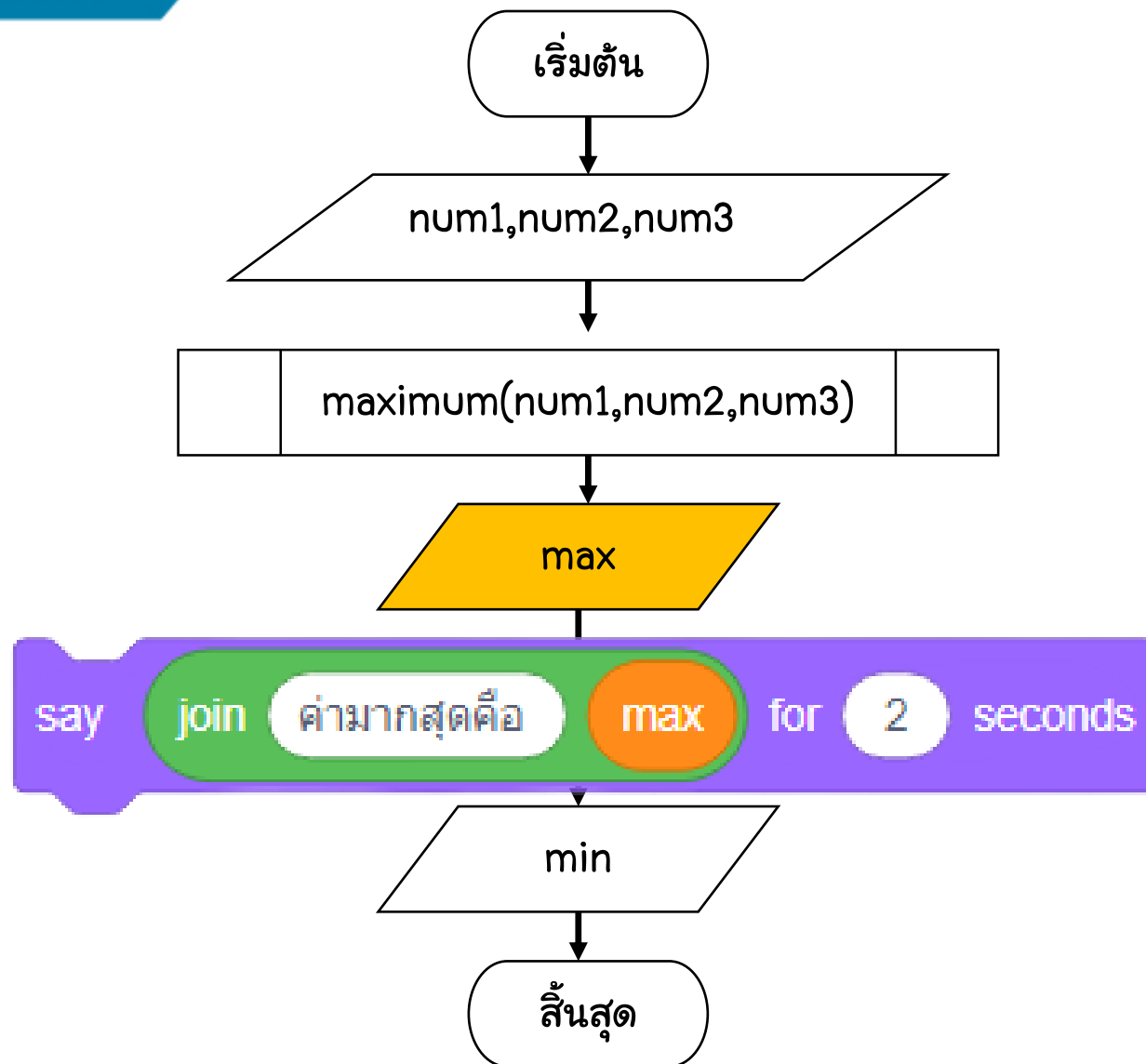


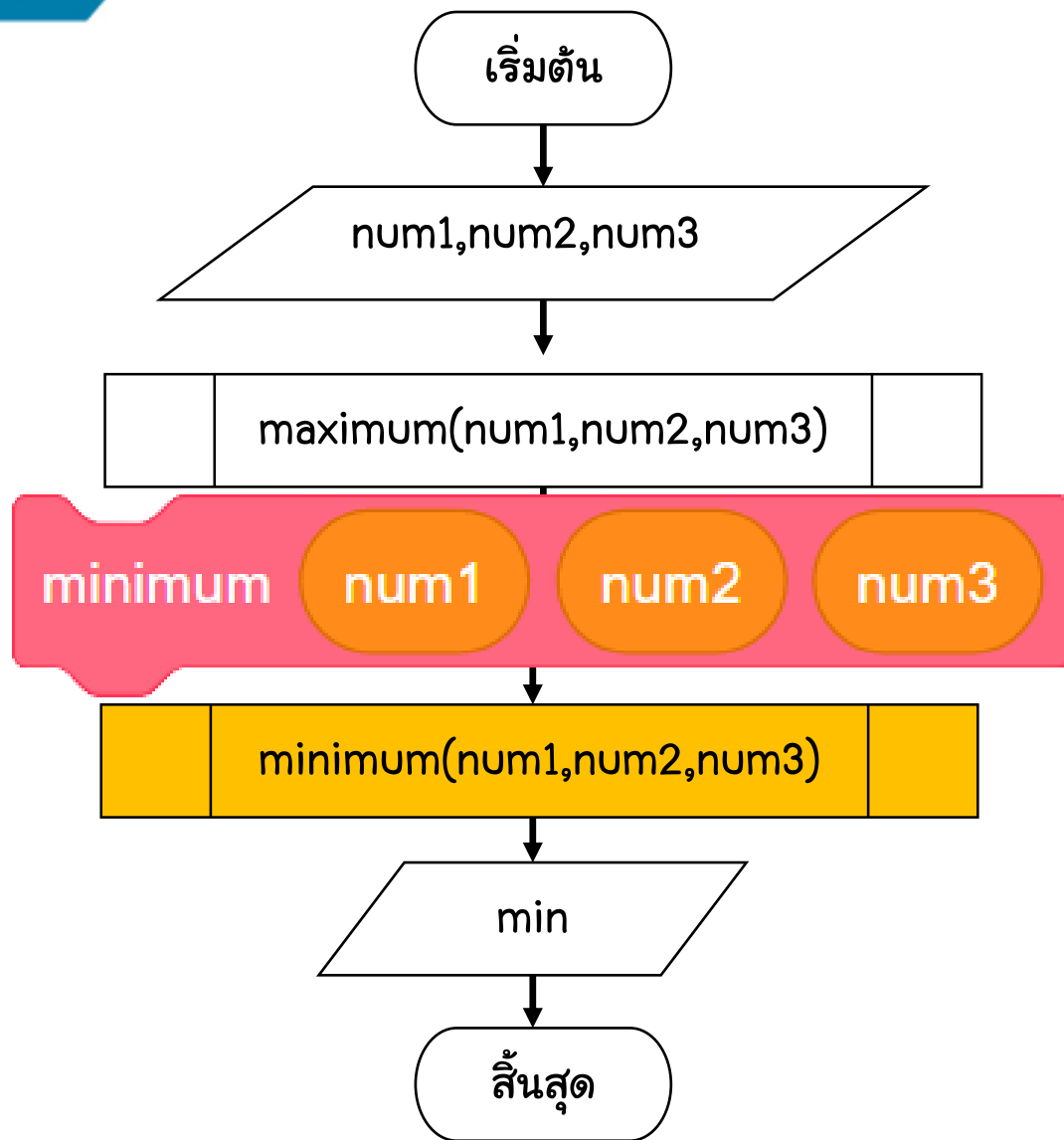
โปรแกรมหลัก

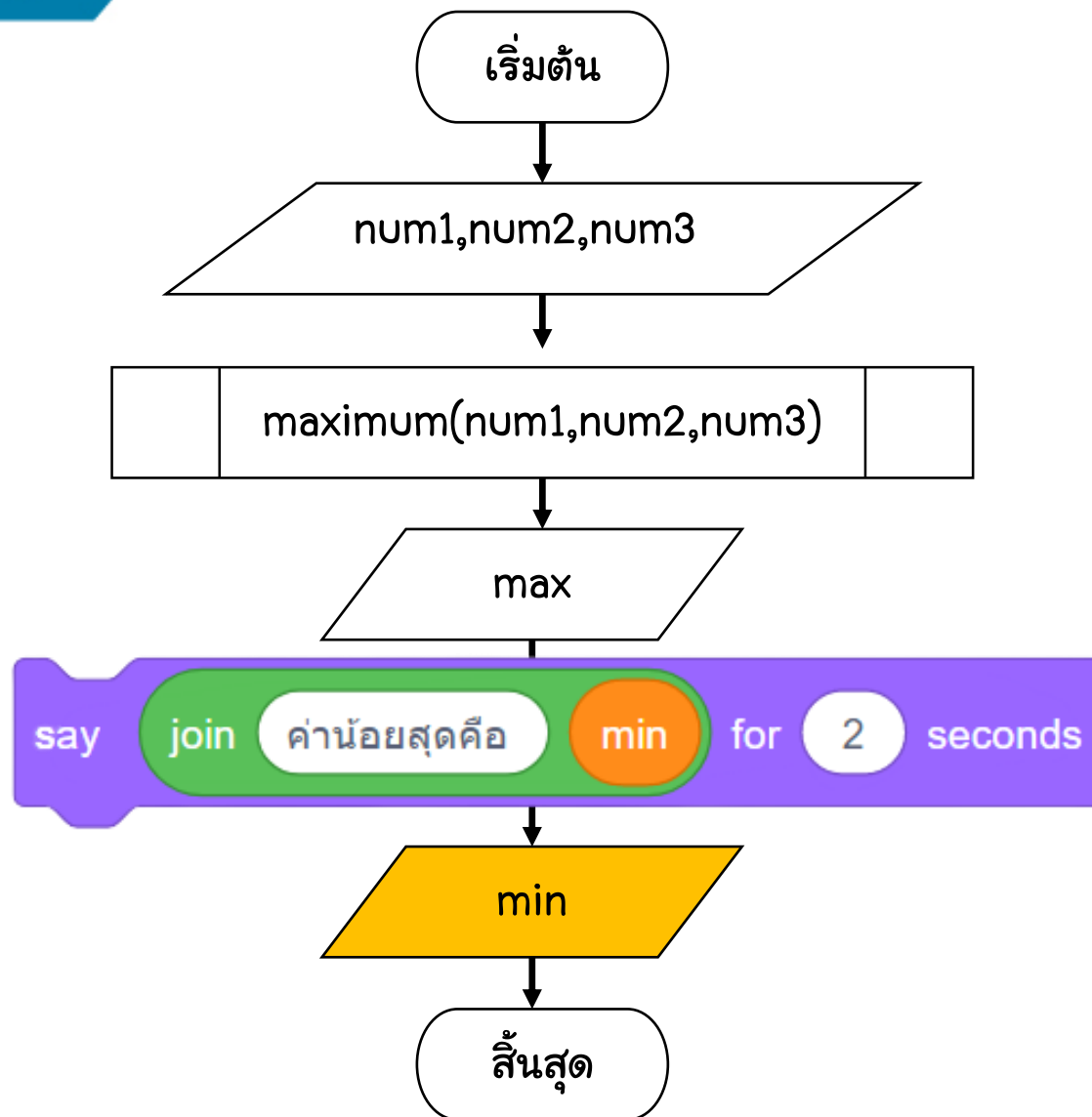








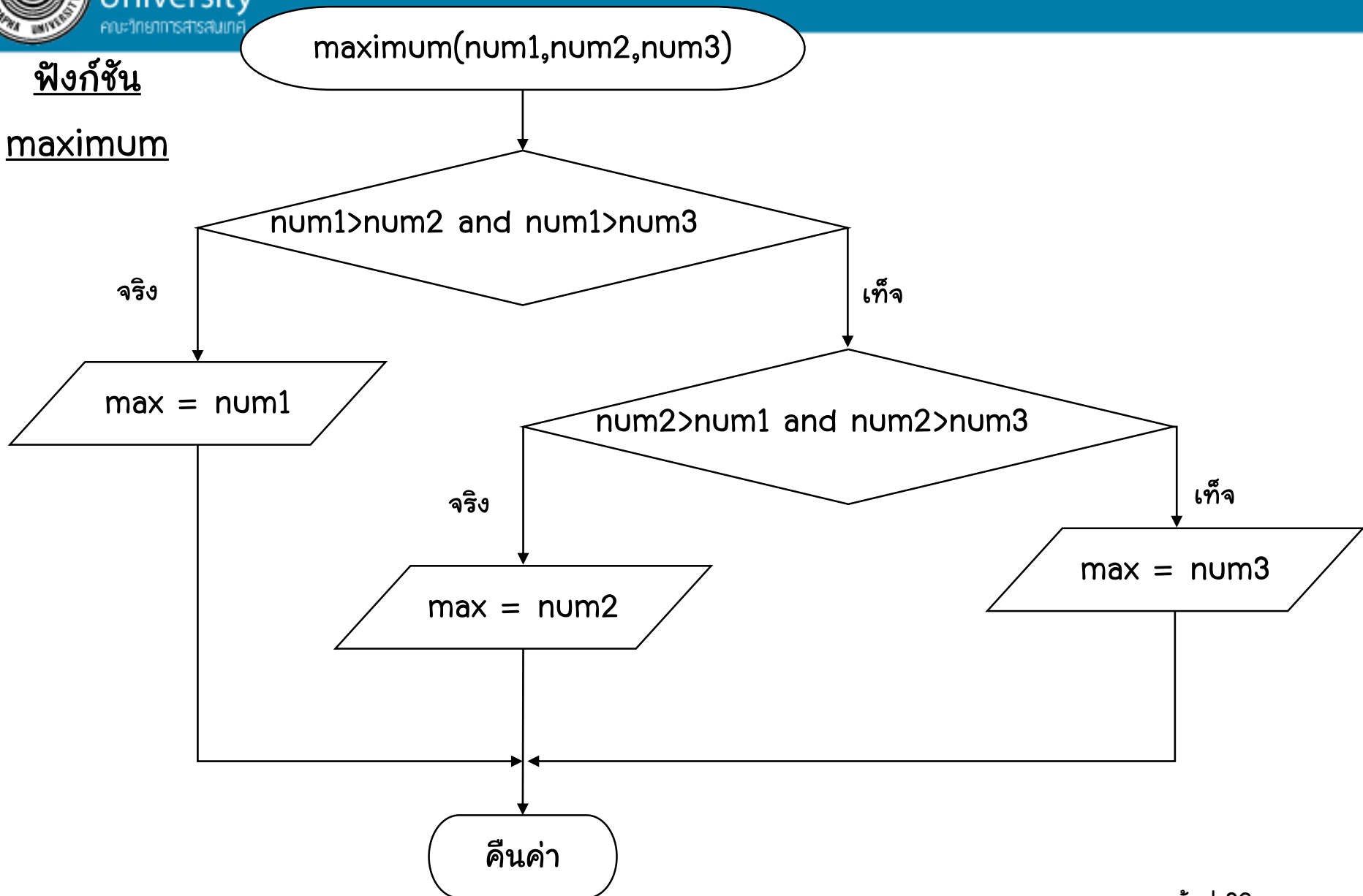






ฟังก์ชัน

maximum



maximum(num1,num2,num3)

define

maximum

number1

number2

number3

จริง

max = num1

เท็จ

num2>num1 and num2>num3

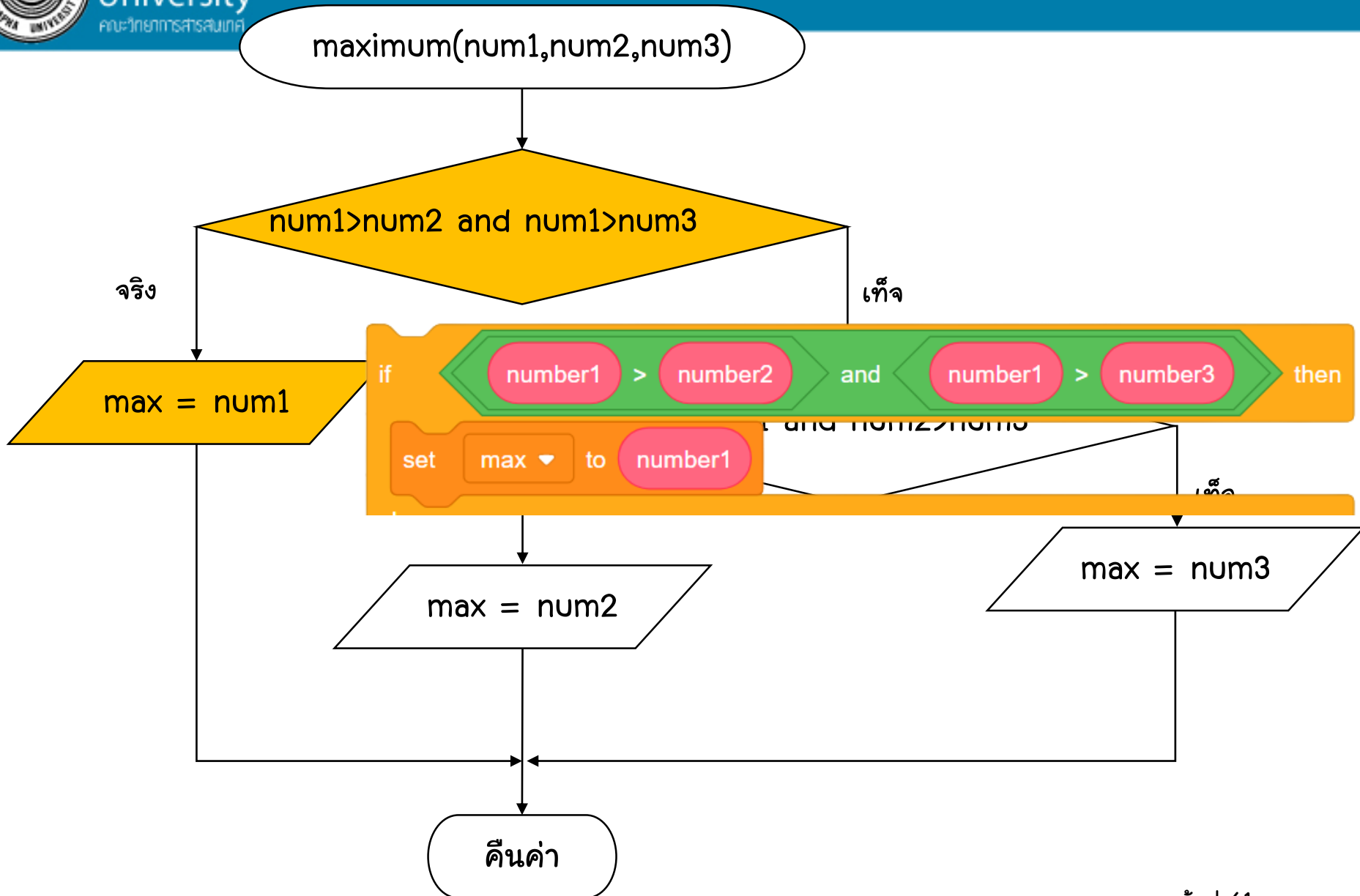
จริง

max = num2

เท็จ

max = num3

คืนค่า





ma

num1

จริง

max = num1

จริง

max = num2

เท็จ

max = num3

คืนค่า

else

if

number2 > number1

and

number2 > number3

then

set

max

▼

to

number2

else

set

max

▼

to

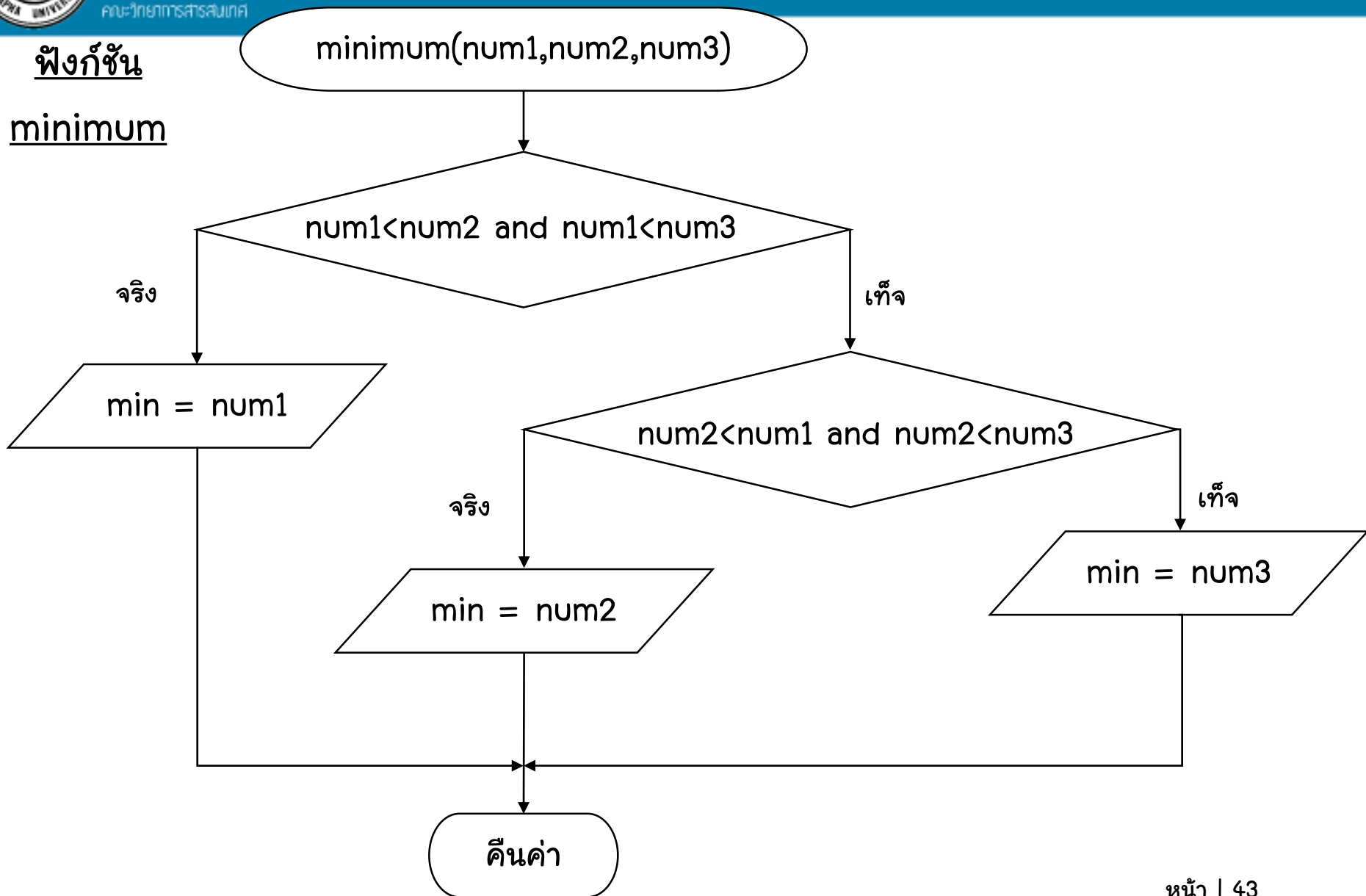
number3

num2>num1 and num2>num3



ฟังก์ชัน

minimum



minimum(num1,num2,num3)

define

minimum

number1

number2

number3

จริง

min = num1

เท็จ

num2<num1 and num2<num3

จริง

min = num2

เท็จ

min = num3

คืนค่า



minimum(num1,num2,num3)

num1<num2 and num1<num3

จริง

เท็จ

min = num1

if

number1 < number2

and

number1 < number3

then

set

min

to

number1

num1 and num2<num3

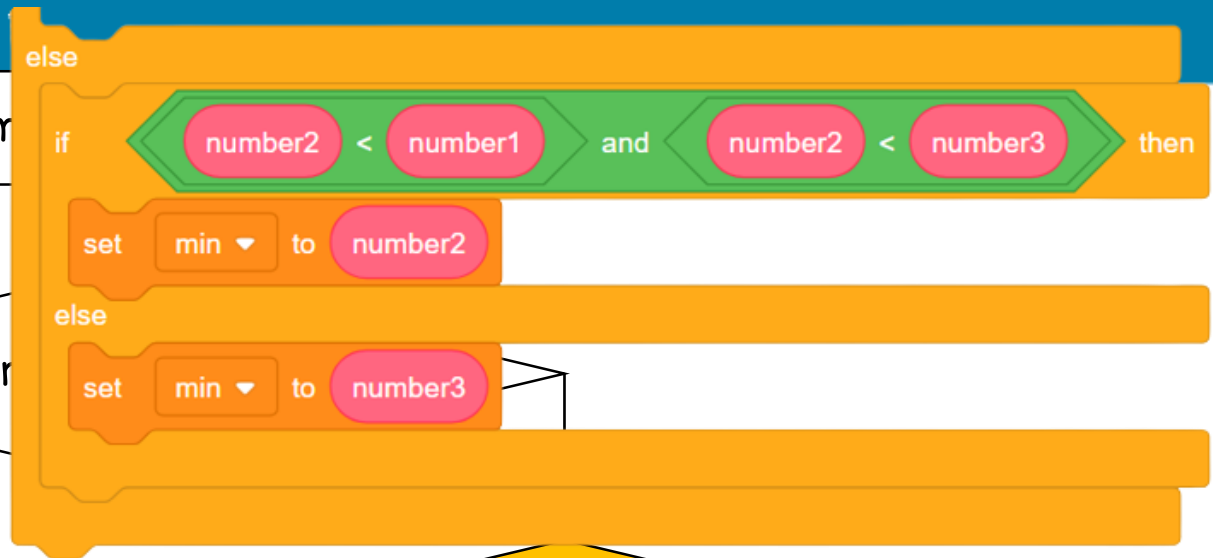
min = num2

min = num3

คืนค่า



minimum



num1 < num2

จริง

min = num1

จริง

min = num2

เท็จ

min = num3

คืนค่า

Scratch! File Edit Tutorials week11Exam4 by logicalThinking See Project Page Join Scratch Sign in

Code Costumes Sounds

Motion

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: -48 y: -109
- glide 1 secs to random position
- glide 1 secs to x: -48 y: -109
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to -48
- change y by 10

Code

```

define average number1 number2 number3
set sum to number1 + number2 + number3
set avg to sum / 3

define maximum number1 number2 number3
if number1 > number2 and number1 > number3 then
  set max to number1
else
  if number2 > number1 and number2 > number3 then
    set max to number2
  else
    set max to number3
  
```

Stage

Sprite Crab x: -48 y: -109

Show Size 100 Direction 90

Backdrops 1

ชมตัวอย่างได้ที่ <https://scratch.mit.edu/projects/111687899/>

แหล่งอ้างอิง

- การเขียนผังงานฟังก์ชัน

จาก : <http://kanokwan.sru.ac.th/e-learning/11.php>

- ความหมายฟังก์ชันนิยามต่าง ๆ

จาก : <https://goo.gl/IJzZ54>

- ตัวอย่างตู้น้ำหยอดเหรียญ กล่องดำ และข้อดีของฟังก์ชัน

จาก : สอนให้เก่งและอยากเรียน “คณิตศาสตร์” แปลโดย ดร.อรรณพ เรืองวิเศษ

- ตัวอย่าง Scratch

จาก : <https://scratch.mit.edu/>



คำถาม ?

