

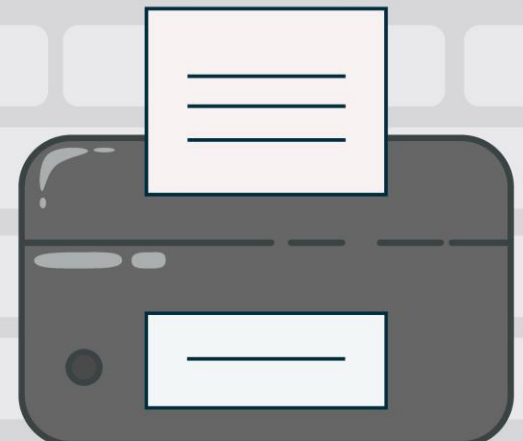
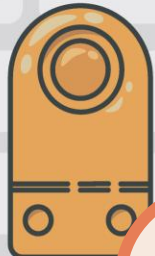


# วิทยาการคำนวณ ม.3

## บทที่ 3 สร้างสรรค์ผลงานด้วยScratch



คลิกเพื่อเข้าสู่บทเรียน



# เมนูหลัก

คำชี้แจง

ผู้จัดทำ

ผลรวมของนั้น  
เป็นจำนวนคู่หรือไม่

รายการข้อมูล

การเขียนโปรแกรมร่วมกับวิดีโอ

การสร้างแอปพลิเคชัน  
ด้วย Scratch

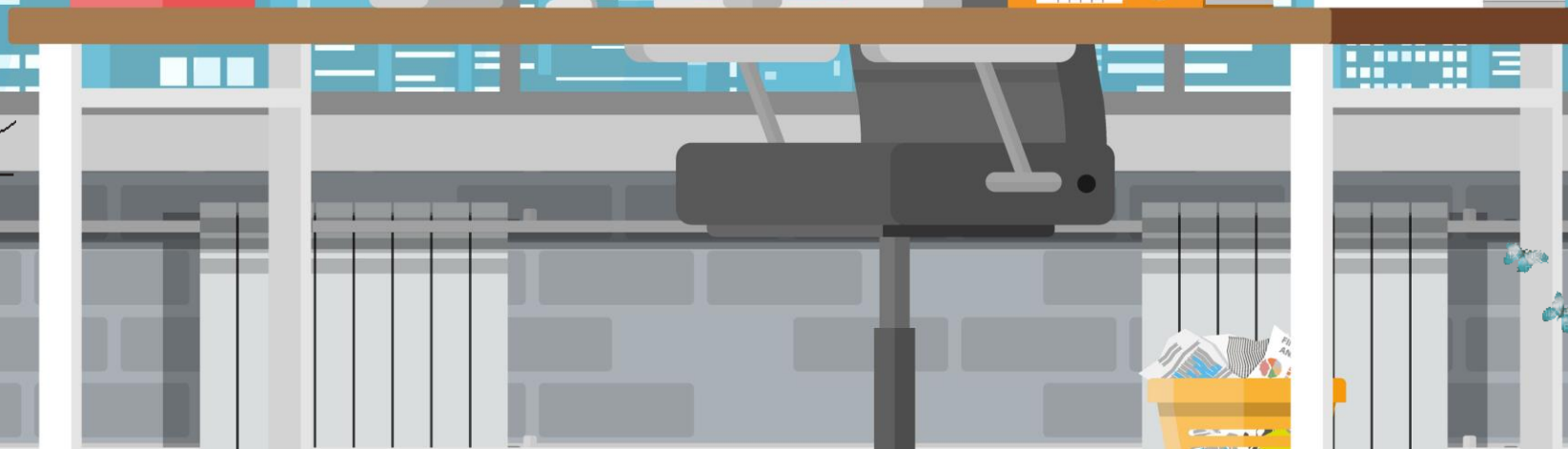


# คำชี้แจง

1. ให้นักเรียนศึกษาบทเรียน
2. ให้นักเรียนทำแบบทดสอบ  
หลังเรียนทั้งหมด 10 ข้อ 10  
คะแนน



ผลรวมของฉันทัน  
เป็นจำนวนคู่หรือไม่





## จุดประสงค์การเรียนรู้



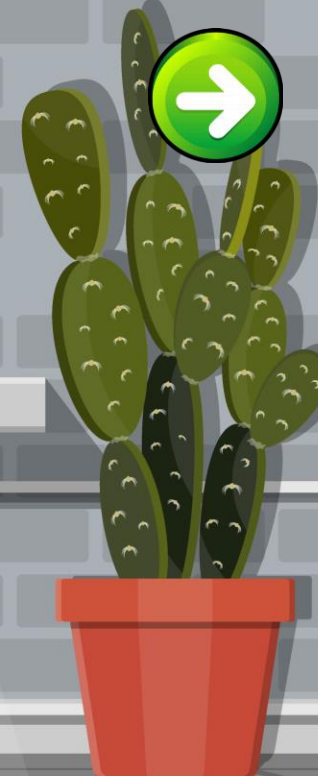
1. นักเรียนสามารถใช้บล็อกคำสั่ง List เพื่อสร้างรายการข้อมูลได้
2. นักเรียนสามารถเขียนโปรแกรมสั่งงานโดยใช้แป้นลูกศรซ้าย/ขวาได้
3. นักเรียนสามารถเขียนโปรแกรมโดยใช้งานวิดีโอได้
4. นักเรียนสามารถสร้างแอปพลิเคชันด้วย Scratch ได้



## ทบทวนความรู้ก่อนเรียน



คำสั่ง `forever` ใน Scratch  
เป็นคำสั่งที่ใช้ในการวนซ้ำแบบ  
ระบุจำนวนรอบ **ถูกต้องหรือไม่?**



## ทบทวนความรู้ก่อนเรียน



หากต้องการเขียนโปรแกรม  
ตรวจสอบที่มี 2 เงื่อนไข จะใช้  
คำสั่ง if-else ถูกต้องหรือไม่?

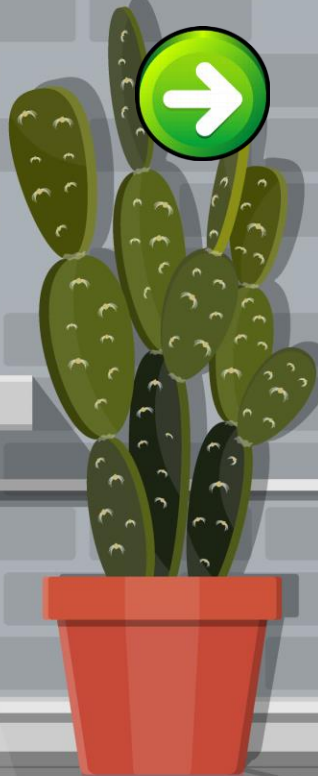




# ทบทวนความรู้ก่อนเรียน



คำสั่ง Create clone of  
myself ใช้ในการโคลนตัวละคร  
ถูกต้องหรือไม่?

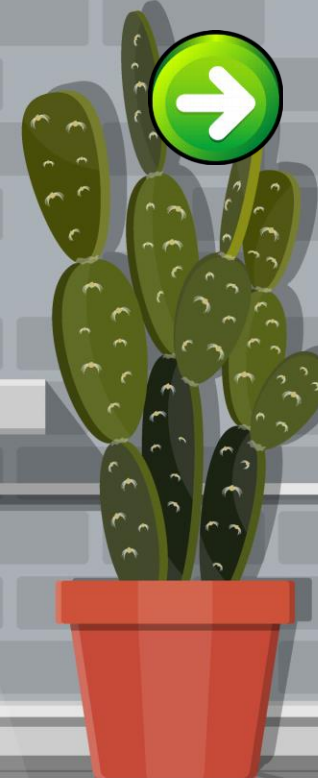




## ทบทวนความรู้ก่อนเรียน



คำสั่ง `set x to 5` ทำให้  
ตัวแปร `x` มีค่าเท่ากับ 5 **ถูกต้อง**  
**หรือไม่?**



# ทบทวนความรู้ก่อนเรียน



คำสั่ง if  $y > 5$  then แสดง  
ค่า  $y$  ถ้า  $y$  มีค่าเท่ากับ 3 โปรแกรม  
จะแสดงค่า 3 ถูกต้องหรือไม่?



## ผลรวมของฉันทัน เป็นจำนวนคู่หรือไม่

**สถานการณ์** โปรแกรมจะสุ่มจำนวน 2 จำนวน  
แล้วให้ผู้เล่นทายว่าผลรวมเป็นจำนวนคู่หรือคี่ เมื่อ  
พิจารณาสถานการณ์แล้วสามารถแบ่งปัญหาใหญ่  
เป็นปัญหาย่อยได้ดังนี้

คำถาม (Question)

คำตอบ (Answer)

## ผลรวมของฉันท เป็นจำนวนคู่หรือไม่

### คำถาม (Question)

1. เลือกจำนวน 2 จำนวนขึ้นมาอย่างไร

2. มีวิธีตรวจสอบอย่างไร ว่าผลรวมของ  
สองจำนวนเป็นจำนวนคู่หรือคี่

### คำตอบ (Answer)

เลือกจากการสุ่มจำนวน 1 ถึง 10

ถ้าผลรวมของสองจำนวนหารด้วย 2  
แล้วมีเศษเท่ากับ 0 แสดงว่าเป็น  
จำนวนคู่ แต่ถ้าเศษเท่ากับ 1 แสดงว่า  
เป็นจำนวนคี่



## ผลรวมของฉันท เป็นจำนวนคู่หรือไม่

### คำถาม (Question)

3. มีการแสดงผลอย่างไร  
เมื่อผู้เล่นตอบถูก



### คำตอบ (Answer)

แสดงเครื่องหมายถูก ถ้าผู้เล่นเลือก  
คำตอบถูก

- เมื่อเศษจากการหารผลรวมสอง  
จำนวนเป็น 0 แล้วผู้เล่นตอบ  
จำนวนคู่
- เมื่อเศษจากการหารผลรวมสอง  
จำนวนเป็น 1 แล้วผู้เล่นตอบ  
จำนวนคี่

## ผลรวมของฉันท เป็นจำนวนคู่หรือไม่

### คำถาม (Question)

4. มีการแสดงผลอย่างไร  
เมื่อผู้เล่นตอบผิด



### คำตอบ (Answer)

แสดงเครื่องหมายผิด ถ้าผู้เล่นเลือก  
คำตอบผิด

- เมื่อเศษจากการหารผลรวมสอง  
จำนวนเป็น 0 แล้วผู้เล่นตอบ  
จำนวนดี
- เมื่อเศษจากการหารผลรวมสอง  
จำนวนเป็น 1 แล้วผู้เล่นตอบ  
จำนวนคู่

## ผลรวมของฉันท เป็นจำนวนคู่หรือไม่

### คำถาม (Question)

5. มีการแสดงผลอย่างไร  
เมื่อผู้เล่นไม่ตอบ

6. ผู้เล่นเลือกคำตอบด้วยวิธีการใด

### คำตอบ (Answer)

ไม่มีการแสดงผลอะไรเพิ่มเติม

เคลื่อนย้ายตัวละครที่ทำหน้าที่เลือก  
คำตอบไปสัมผัสกับตัวละครที่เป็น  
คำตอบจำนวนคู่หรือคี่

# ผลรวมของฉันท เป็นจำนวนคู่หรือไม่

## คำถาม (Question)

## คำตอบ (Answer)

7. คำถามมีกี่ข้อ

ไม่จำกัด

8. มีการเปลี่ยนคำถามเมื่อใด

เปลี่ยนคำถามเมื่อผู้เล่น  
เลือกคำตอบถูกหรือผิด

9. เมื่อผู้เล่นตอบถูกจะได้กี่คะแนน

เมื่อตอบถูกจะได้ 1 คะแนน

10. เมื่อผู้เล่นตอบผิดจะได้กี่คะแนน

เมื่อตอบผิดจะได้ -1 คะแนน



### ตัวอย่างที่ 3.1

## การสร้างโปรแกรมผลรวมของฉันทัน เป็นจำนวนคู่หรือไม่

- การวิเคราะห์และกำหนดรายละเอียดของปัญหา

ข้อมูลเข้า

คือ คำตอบของผู้เล่น

ข้อมูลออก

คือ แสดงผลการตรวจสอบ  
โดยแสดงผลเป็นเครื่องหมาย  
ถูกหรือผิด

### ตัวอย่างที่ 3.1

## การสร้างโปรแกรมผลรวมของฉันทัน เป็นจำนวนคู่หรือไม่

วิธีการตรวจสอบ

ตัวอย่าง ดังนี้

- เมื่อสุ่มจำนวน 2 จำนวน ได้ 2 และ 6 ผู้เล่นตอบว่าเป็นจำนวนคู่ จะแสดงเครื่องหมายถูก
- เมื่อสุ่มจำนวน 2 จำนวน ได้ 3 และ 8 ผู้เล่นตอบว่าเป็นจำนวนคู่ จะแสดงเครื่องหมายผิด
- เมื่อสุ่มจำนวน 2 จำนวน ได้ 2 และ 7 ผู้เล่นไม่ตอบ จะไม่มีการแสดงผลเพิ่มเติม

### ตัวอย่างที่ 3.1 การสร้างโปรแกรมผลรวมของฉันทัน เป็นจำนวนคู่หรือไม่

#### ❑ การวางแผนการแก้ปัญหา

การออกแบบหน้าจอโปรแกรมต่อไปนี้จะนำเสนอเฉพาะคำตอบที่เป็นจำนวนคู่ ให้ผู้เล่นเลือก และถ้าผู้เล่นเลือกถูกต้องจะแสดงเครื่องหมายถูก

### ตัวอย่างที่ 3.1 การสร้างโปรแกรมผลรวมของฉันทัน เป็นจำนวนคู่หรือไม่

จากสถานการณ์และการแบ่งปัญหาย่อย อาจกำหนด  
ส่วนประกอบหน้าจอได้ดังนี้

6

+

6

EVEN





### ตัวอย่างที่ 3.1 การสร้างโปรแกรมผลรวมของฉันท เป็นจำนวนคู่หรือไม่

6

+

6

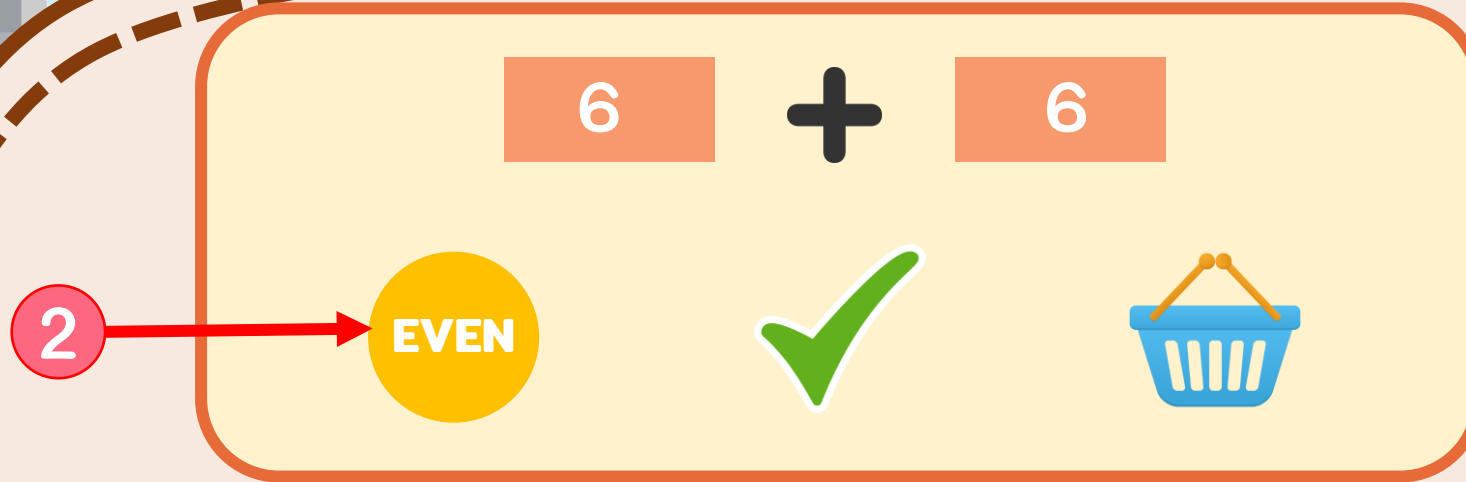
EVEN



1

1. ตัวละคร Answer สำหรับเลือกตอบ ผู้เล่นสามารถเคลื่อนที่ตัวละครนี้ไปเลือกคำตอบ โดยกดแป้นลูกศรซ้าย/ขวา ที่คีย์บอร์ด ซึ่งจะเป็นการเคลื่อนที่ไปทางซ้ายหรือขวา ตามลำดับ นอกจากนี้ยังทำหน้าที่ตรวจสอบความถูกต้องของคำตอบ

### ตัวอย่างที่ 3.1 การสร้างโปรแกรมผลรวมของฉันท เป็นจำนวนคู่หรือไม่



2. ตัวละคร **Even** แทนคำตอบจำนวนคู่ เพื่อให้ผู้เล่นเลือกเมื่อคิดว่าผลรวมเป็นจำนวนคู่อาจกำหนดให้ตัวละครนี้เคลื่อนที่จากขอบด้านบนลงด้านล่างของเวที

ตัวอย่างที่ 3.1  
การสร้างโปรแกรมผลรวมของฉันท  
เป็นจำนวนคู่หรือไม่

6

+

6

EVEN



3

3. ตัวละคร Correct จะปรากฏขึ้นเมื่อผู้เล่นเลือกคำตอบถูกต้อง

ตัวอย่างที่ 3.1  
การสร้างโปรแกรมผลรวมของฉันท  
เป็นจำนวนคู่หรือไม่



4. จำนวนสองจำนวน จะปรากฏขึ้นเพื่อให้ผู้เล่นหาผลรวมและ  
คำนวณว่าเป็นจำนวนคู่หรือไม่ โดยใช้การแสดงค่าตัวแปรบนเวที

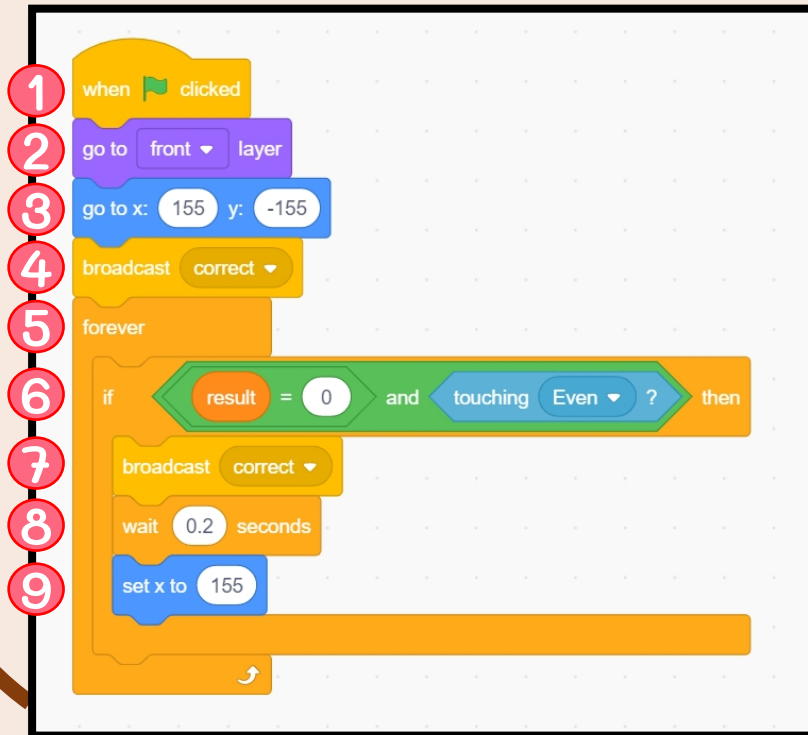


### ตัวอย่างที่ 3.1 การสร้างโปรแกรมผลรวมของฉันทัน เป็นจำนวนคู่หรือไม่

ตัวละคร Answer



สำหรับเลือกคำตอบ



เมื่อผู้เล่นคลิกธงเขียว

1. บรรทัดที่ 2 กำหนดตัวละครให้อยู่เลเยอร์ (layer) บนสุด โดยถ้าหากตัวละครหลายตัวมีการเคลื่อนที่มาทับกัน ตัวละครที่กำหนด go to front จะอยู่ด้านบนเสมอ
2. บรรทัดที่ 3 กำหนดให้ตัวละครมีตำแหน่งเริ่มต้นที่พิกัด (155,-155)

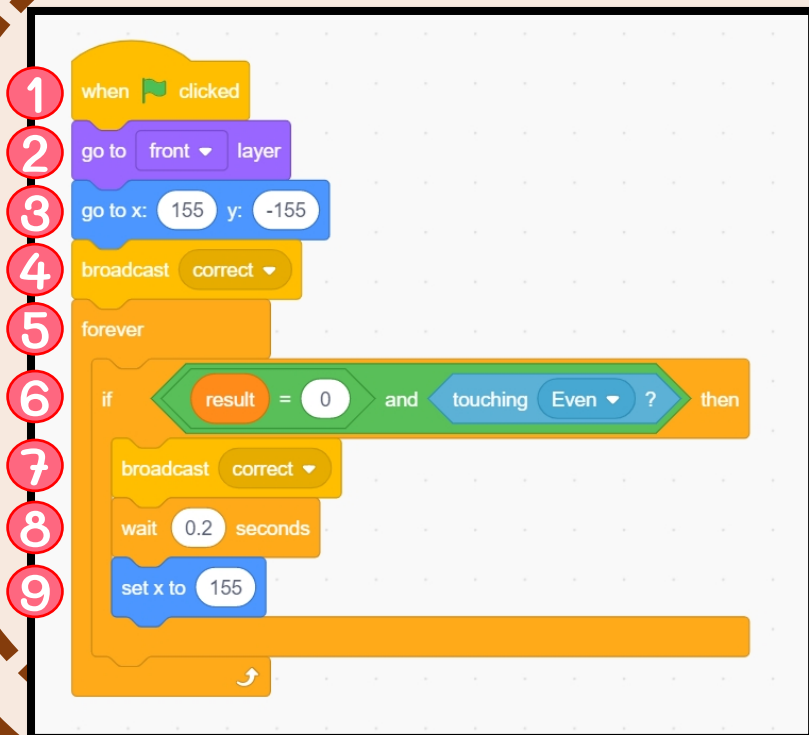
### ตัวอย่างที่ 3.1

#### การสร้างโปรแกรมผลรวมของฉันท เป็นจำนวนคู่หรือไม่

ตัวละคร Answer



สำหรับเลือกคำตอบ



3. บรรทัดที่ 4 กระจายสารไปยัง random\_check ที่  
ทำหน้าที่สุ่มจำนวน 2 จำนวนและหาว่าผลลัพธ์  
เป็นจำนวนคู่หรือไม่ (อยู่ที่ตัวละคร Even)

4. บรรทัดที่ 5 ทำซ้ำคำสั่งในบรรทัดที่ 6-9 ไม่รู้จบ  
โดยตรวจสอบ 2 ประเด็นคือ

- 4.1 ผลลัพธ์ที่ได้จากข้อ 3 ว่าเท่ากับ 0 หรือไม่  
ถ้าใช่แสดงว่าเป็นจำนวนคู่
- 4.2 ตรวจสอบว่าตัวละคร Answer นี้สัมผัสถูก  
ตัวละคร Even หรือไม่

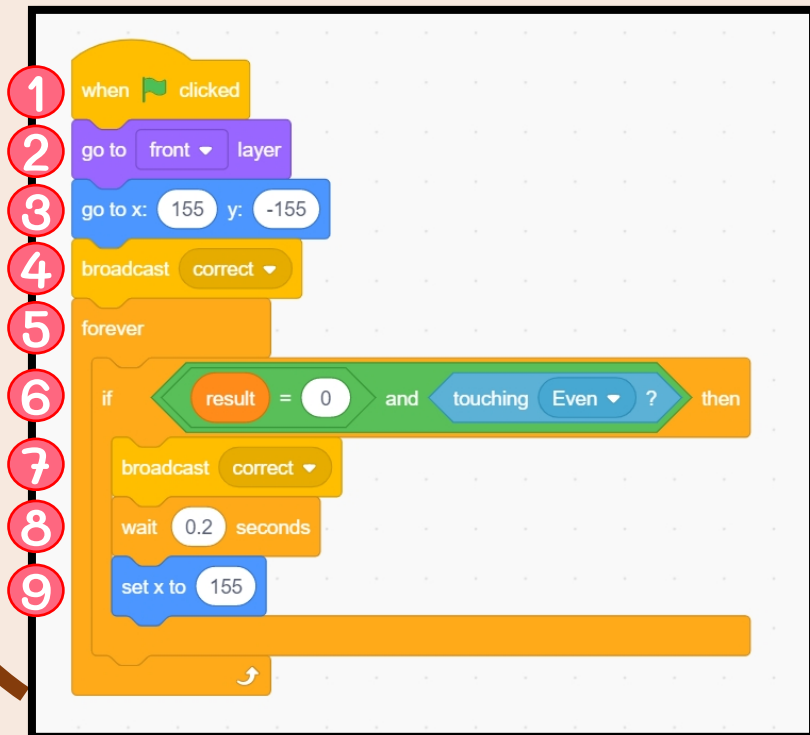
### ตัวอย่างที่ 3.1

#### การสร้างโปรแกรมผลรวมของฉันทัน เป็นจำนวนคู่หรือไม่

ตัวละคร Answer



สำหรับเลือกคำตอบ



ถ้าเป็นจริงทั้งสองประเด็น จะทำ  
คำสั่งในบรรทัดที่ 7 คือ จะกระจายสาร  
correct ซึ่งจะแสดงเครื่องหมายถูก บรรทัด  
ที่ 8 หน่วงเวลา 0.2 วินาที และบรรทัดที่ 9  
เปลี่ยนตำแหน่งของตัวละคร Answer ไปยัง  
ตำแหน่งเริ่มต้น



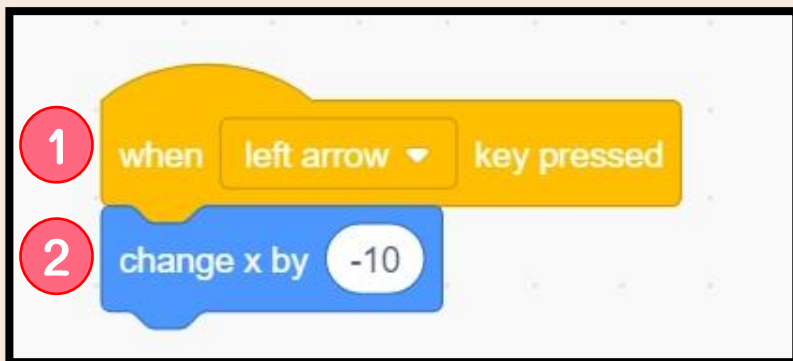
### ตัวอย่างที่ 3.1

#### การสร้างโปรแกรมผลรวมของจำนวนคู่หรือไม่

ตัวละคร Answer



สำหรับเลือกคำตอบ



การตรวจสอบการกดแป้นลูกศรซ้าย

บรรทัดที่ 1 ประกาศส่วนหัว left arrow เพื่อตรวจสอบว่ามีการกดแป้นลูกศรซ้ายบนคีย์บอร์ดหรือไม่ ถ้ามีจะทำงานในบรรทัดที่ 2 คือ เปลี่ยนตำแหน่งโดยลดค่า  $x$  ลงจากเดิม 10 หน่วย มีผลทำให้ตัวละครขยับไปทางซ้ายมือ



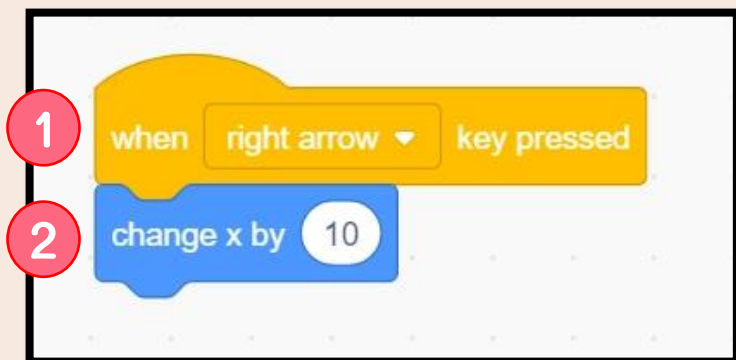
### ตัวอย่างที่ 3.1

#### การสร้างโปรแกรมผลรวมของจำนวนคู่หรือไม่

ตัวละคร Answer



สำหรับเลือกคำตอบ

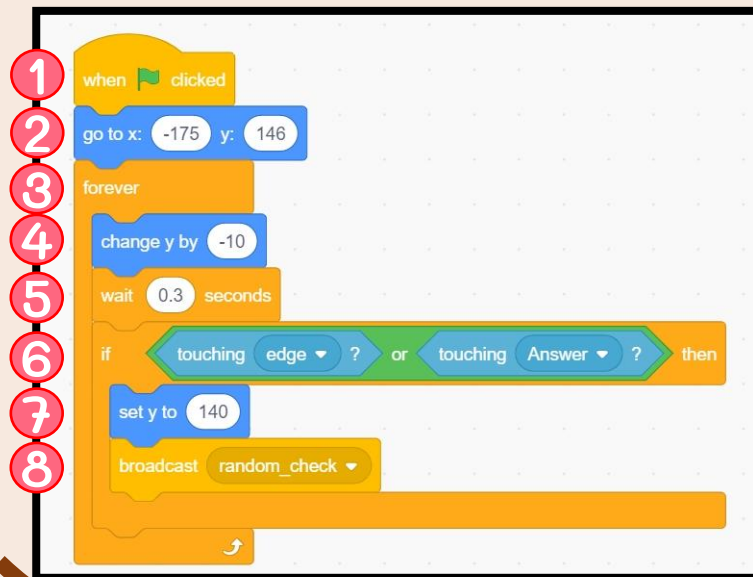


การตรวจสอบการกดแป้นลูกศรซ้าย

บรรทัดที่ 1 ประกาศส่วนหัว right arrow เพื่อตรวจสอบว่ามีการกดแป้นลูกศรขวาบนคีย์บอร์ดหรือไม่ ถ้ามีจะทำงานในบรรทัดที่ 2 คือ เปลี่ยนตำแหน่งโดยเพิ่มค่า x จากค่าเดิมอีก 10 หน่วย มีผลทำให้ตัวละครขยับไปทางขวามือ

## ตัวอย่างที่ 3.1 การสร้างโปรแกรมผลรวมของฉันทัน เป็นจำนวนคู่หรือไม่

ตัวละคร Even **EVEN** แทนคำตอบที่เป็นจำนวนคู่



เมื่อผู้เล่นคลิกธงเขียว มีการทำงานดังนี้

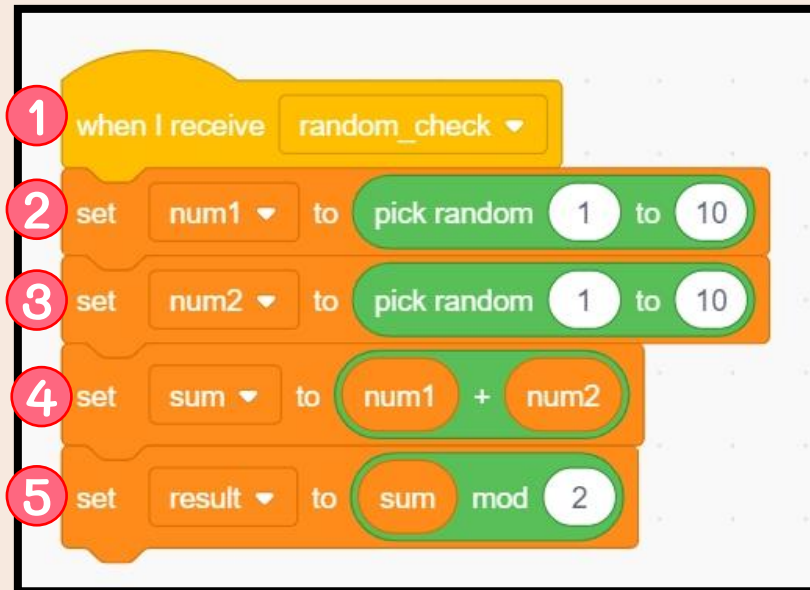
1. บรรทัดที่ 3 กำหนดการทำซ้ำคำสั่งในบรรทัดที่ 4-8 ไม่รู้จบ
2. บรรทัดที่ 4 เปลี่ยนตำแหน่ง โดยลดค่า y จากเดิม 10 หน่วย
3. บรรทัดที่ 5 หน่วงเวลา 0.3 วินาที
4. บรรทัดที่ 6 ตรวจสอบว่า

ถ้า ตัวละครสัมผัสขอบจอหรือสัมผัสตัวละคร Answer ให้ทำ  
บรรทัดที่ 7 คือ กำหนด  $y = 140$  หน่วย และบรรทัดที่ 8 กระจาย  
สาร random\_check (ทำหน้าที่สุ่มจำนวน 2 จำนวน และหา  
ผลลัพธ์ว่าเป็นจำนวนคู่หรือไม่)

### ตัวอย่างที่ 3.1

#### การสร้างโปรแกรมผลรวมของฉันท เป็นจำนวนคู่หรือไม่

ตัวละคร Even **EVEN** แทนคำตอบที่เป็นจำนวนคู่



การรับสาร random\_check เพื่อให้สุ่ม  
จำนวนและหาผลลัพธ์ว่าเป็นจำนวนคู่

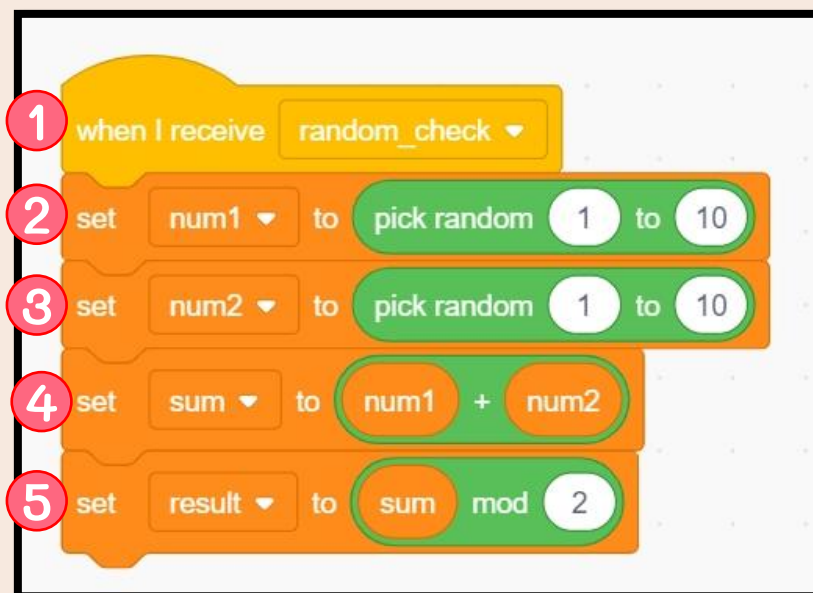
1. บรรทัดที่ 1 ประกาศส่วนหัวของการรับสาร random\_check
2. บรรทัดที่ 2 กำหนดค่าตัวแปร num1 ให้มีค่าเท่ากับค่าที่สุ่มได้ในช่วง [1-10]
3. บรรทัดที่ 3 กำหนดค่าตัวแปร num2 ให้มีค่าเท่ากับค่าที่สุ่มได้ในช่วง [1-10]



### ตัวอย่างที่ 3.1

#### การสร้างโปรแกรมผลรวมของฉันทัน เป็นจำนวนคู่หรือไม่

ตัวละคร Even **EVEN** แทนคำตอบที่เป็นจำนวนคู่



- บรรทัดที่ 4 นำผลบวกของตัวแปร num1 และ num 2 ไปเก็บไว้ในตัวแปร sum
- บรรทัดที่ 5 หาเศษจากการหารตัวแปร sum ด้วย 2 แล้วนำไปเก็บไว้ในตัวแปร result (ถ้าเศษจากการหารเป็น 0 แสดงว่าเป็นจำนวนคู่ แต่ถ้าเป็น 1 แสดงว่าเป็นจำนวนคี่)



### ตัวอย่างที่ 3.1 การสร้างโปรแกรมผลรวมของฉันทัน เป็นจำนวนคู่หรือไม่

ตัวละคร Correct



จะปรากฏเมื่อผู้ใช้ตอบถูก



เมื่อผู้เล่นคลิกธงเขียว  
โปรแกรมจะทำคำสั่งในบรรทัดที่ 2 ซึ่ง  
จะกำหนดให้ซ่อนตัวละคร

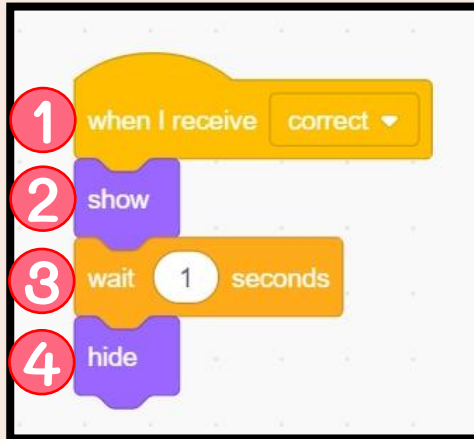
### ตัวอย่างที่ 3.1

#### การสร้างโปรแกรมผลรวมของฉันทัน เป็นจำนวนคู่หรือไม่

ตัวละคร Correct



จะปรากฏเมื่อผู้ใช้ตอบถูก



ประกาศส่วนหัวของการรับสาร correct  
เพื่อแสดงตัวละคร correct

1. บรรทัดที่ 1 กำหนดการรับสาร correct
2. บรรทัดที่ 2 แสดงตัวละคร
3. บรรทัดที่ 3 หน่วงเวลาในการแสดงตัวละคร 1 วินาที
4. บรรทัดที่ 4 ซ่อนตัวละคร



# รายการข้อมูล





## รายการข้อมูล

ใน Scratch มีคำสั่งสำหรับสร้างตัวแปรที่เก็บได้เพียงค่าเดียว แต่หากต้องการสร้างตัวแปรเดียวที่เก็บรายการของค่าหลายค่าก็สามารถทำได้โดยใช้คำสั่ง **Make a List** ซึ่งอยู่ในกลุ่มบล็อก **Variables**

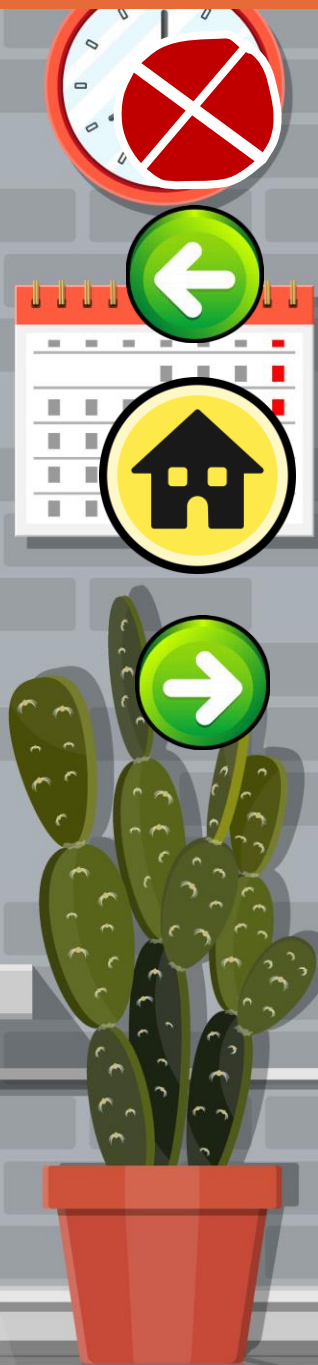




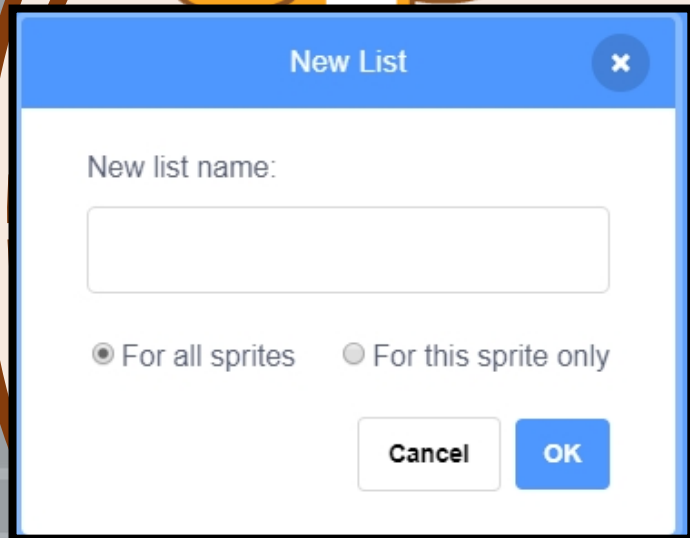
# การสร้างรายการ

การสร้างรายการ (List) มีขั้นตอนดังนี้

1. เลือกกลุ่มบล็อก Variable และคลิกปุ่ม Make a List



## การสร้างรายการ



A screenshot of the 'New List' dialog box in Scratch. The dialog has a blue title bar with the text 'New List' and a close button. Below the title bar, there is a label 'New list name:' followed by a text input field. At the bottom, there are two radio buttons: 'For all sprites' (which is selected) and 'For this sprite only'. Below the radio buttons are two buttons: 'Cancel' and 'OK'.

2. เมื่อปรากฏกรอบโต้ตอบ New List ให้  
เติมชื่อรายการที่ช่อง New List name:  
ในที่นี้ให้เติมชื่อรายการ myList แล้ว  
คลิกปุ่ม OK

**For all sprites** กำหนดให้ตัวละคร  
ทุกตัว สามารถใช้รายการนี้ได้

**For this sprite only** กำหนดให้  
เฉพาะตัวละครนี้ สามารถใช้  
รายการนี้ได้

## บล็อกคำสั่งที่ใช้สำหรับรายการ



บล็อกรายการที่ผู้ใช้สร้างขึ้น  
ในที่นี้คือ myList ซึ่งสามารถ  
นำบล็อกนี้ไปใช้กับคำสั่งอื่น  
เพื่ออ้างถึงข้อมูลในรายการ



เพิ่มข้อมูลต่อท้ายจากข้อมูล  
สุดท้ายในรายการ เช่น เพิ่ม  
“thing” ในรายการ myList



## บล็อกคำสั่งที่ใช้สำหรับรายการ

delete 1 of myList ▼

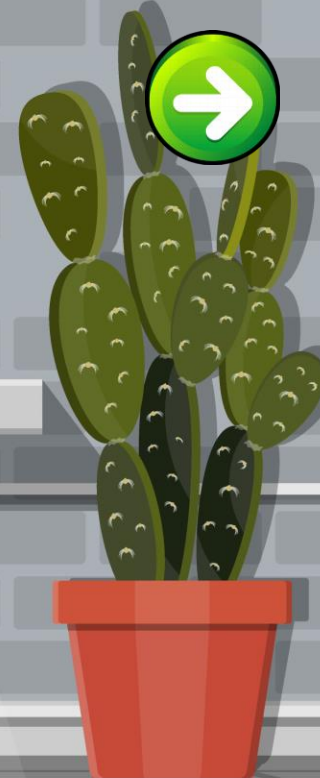


ลบข้อมูลลำดับที่ระบุในรายการ จะส่งผลให้ข้อมูลนั้นถูกลบไปและข้อมูลอื่นต่อจากนั้นในรายการขยับขึ้น 1 ลำดับ เช่น ลบข้อมูลลำดับที่ 1 ในรายการ myList

delete all of myList ▼



ลบข้อมูลทั้งหมดในรายการ myList





## บล็อกคำสั่งที่ใช้สำหรับรายการ

insert 1 at 1 of myList ▼

แทรกข้อมูลในรายการโดยระบุลำดับ  
เช่น แทรก “1” ลงในลำดับที่ 1 ของ  
รายการ myList จะส่งผลให้ข้อมูล  
ตั้งแต่ลำดับที่ 1 เดิม ขยับลง 1 ลำดับ

replace item 1 of myList ▼ with 1

แทนข้อมูลลำดับที่ระบุใน  
รายการ เช่น แทนข้อมูล “1”  
ในรายการ myList ลำดับที่ 1



## บล็อกคำสั่งที่ใช้สำหรับรายการ

item 1 of myList ▼



อ้างถึงข้อมูลลำดับที่ระบุในรายการ  
เช่น อ้างถึงข้อมูลลำดับที่ 1 ใน  
รายการ myList

item # of thing in myList ▼



อ้างถึงลำดับของข้อมูล “thing” ใน  
รายการ myList

length of myList ▼



จำนวนข้อมูลทั้งหมดในรายการ  
myList



## บล็อกคำสั่งที่ใช้สำหรับรายการ

myList ▼ contains thing ?

ตรวจสอบข้อมูลในรายการ myList ว่า  
มีรายการ “thing” หรือไม่

show list ▼ myList ▼

แสดงข้อมูลทั้งหมดในรายการ  
myList

hide list ▼ myList ▼

ซ่อนข้อมูลในรายการ myList



## การเพิ่มและลบข้อมูล ที่กรอบแสดงรายการ

นอกจากการเพิ่มและลบข้อมูลใน  
รายการโดยใช้บล็อกคำสั่งแล้ว **ยังสามารถ**  
**กระทำผ่านกรอบแสดงรายการ** ที่ปรากฏขึ้น  
เมื่อมีการสร้างรายการ หรือมีการใช้บล็อก  
คำสั่ง `show list`





## การเพิ่มและลบข้อมูล ที่กรอบแสดงรายการ

การเพิ่มและลบข้อมูลในรายการทำได้ดังนี้

1. คลิกเครื่องหมาย +  
ที่มุมล่างซ้าย  
ของกรอบแสดงรายการ



## การเพิ่มและลบข้อมูล ที่กรอบแสดงรายการ

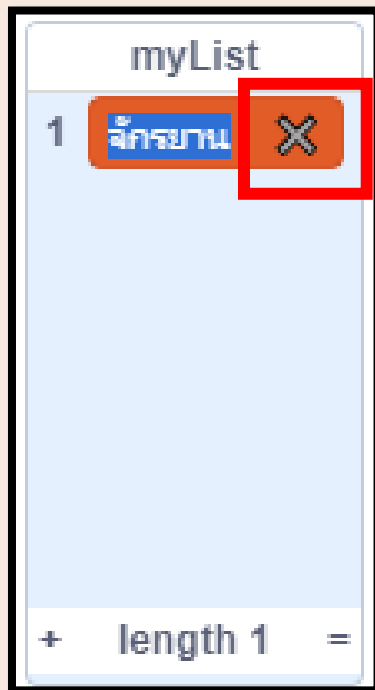
2. จะปรากฏช่องพร้อมตัวเลขกำหนดลำดับข้อมูลในรายการ ให้เติมข้อมูลที่ต้องการ เมื่อเติมเสร็จแล้วให้คลิกบนพื้นที่อื่น ถ้ากดแป้น Enter จะปรากฏช่องให้เติมข้อมูลใหม่



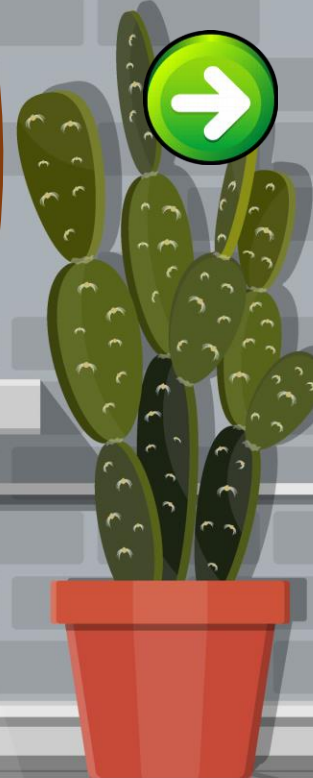
myList	
1	<input type="text"/>

+ length 1 =

## การเพิ่มและลบข้อมูล ที่กรอบแสดงรายการ



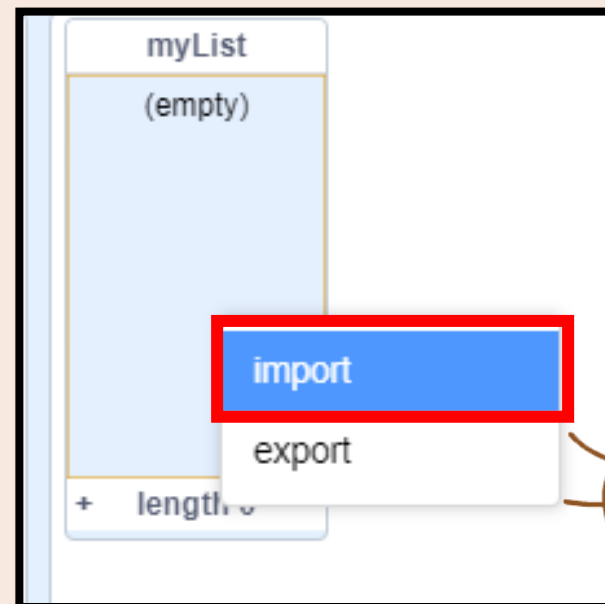
3. หากต้องการลบข้อมูล  
ในรายการ ให้คลิกเครื่องหมาย  
กากบาทหลังข้อมูลนั้น



## การนำเข้าและส่งออก ไฟล์รายการ

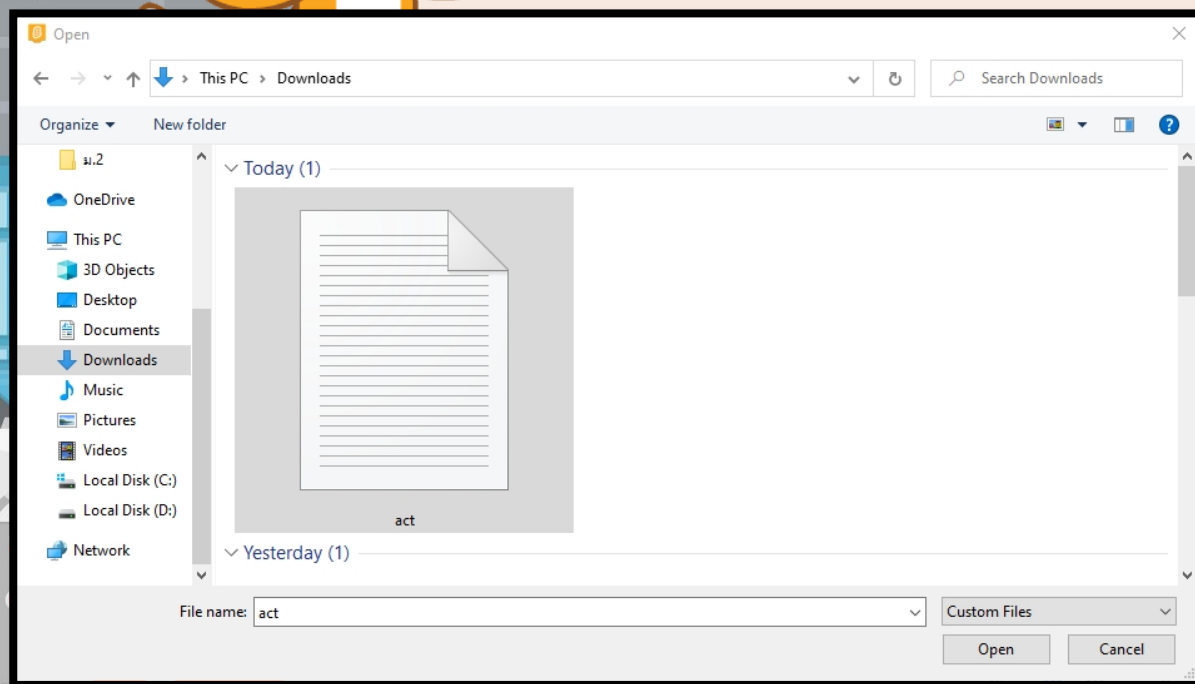
เราสามารถเพิ่มข้อมูลจำนวนมากในรายการโดยใช้วิธีการนำเข้าไฟล์ข้อมูลที่มีอยู่แล้ว สามารถทำได้โดย

1. คลิกขวาที่กรอบแสดงรายการ แล้วเลือก import





## การนำเข้าและส่งออก ไฟล์รายการ

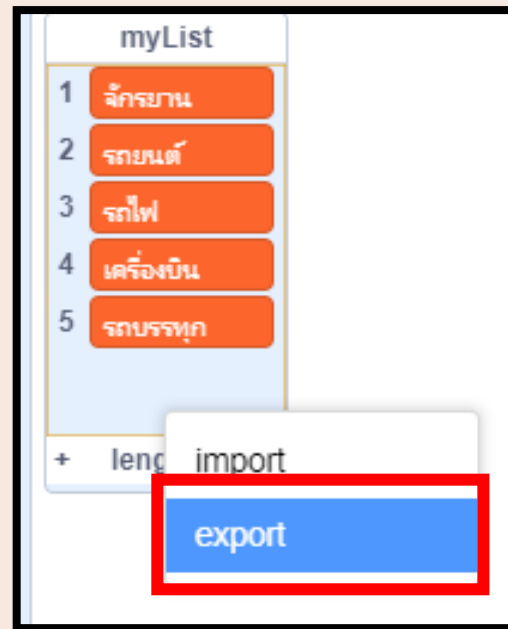


2. จะปรากฏหน้าต่าง Open  
เพื่อให้ผู้ใช้เลือกไฟล์ที่ต้องการ  
โดยไฟล์ที่นำเข้า จะต้องเป็นไฟล์  
ซีเอสวี (.csv) , ทีเอสวี (.tsv) , หรือ  
เท็กซ์ (.txt) หลังจากเลือกไฟล์  
ที่ต้องการแล้วให้คลิกปุ่ม Open

## การส่งออกรายการเป็นไฟล์

ขั้นตอนการการส่งออกรายการเป็นไฟล์ มีดังนี้

คลิกขวาที่รอบรายการ  
แล้วเลือก export จากนั้น  
ตั้งชื่อไฟล์ที่ต้องการในช่อง  
File name: แล้วคลิกปุ่ม Save



## ตัวอย่างที่ 3.2 หานำหนักค่าเฉลี่ย ของเพื่อในห้องด้วยรายการ

สามารถเขียนเป็นรหัสจำลองโดยใช้รายการได้ดังนี้

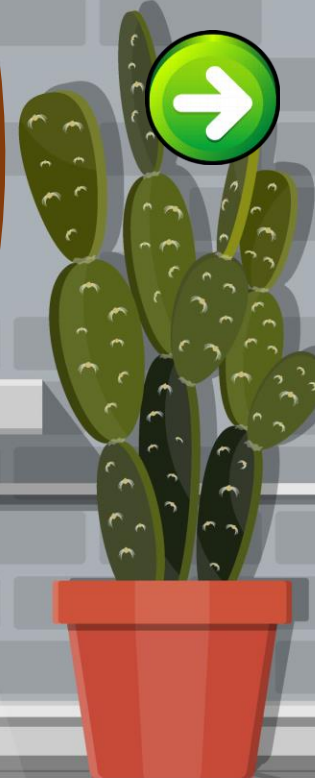
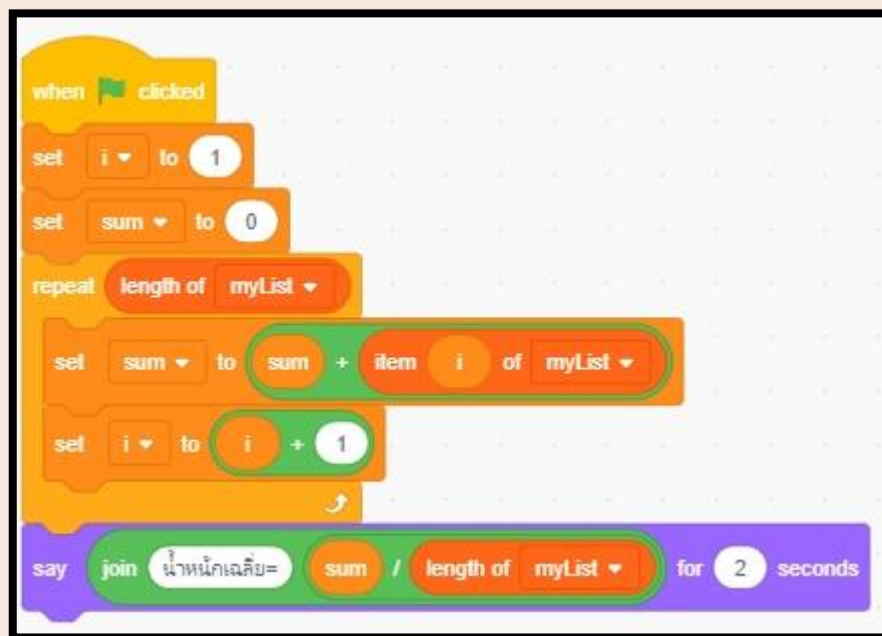
1.  $I \leftarrow 1$
2.  $sum \leftarrow 0$
3. ทำซ้ำคำสั่งต่อไปนี้จนกว่าจะ  
หมดข้อมูลในรายการ  
3.1  $sum \leftarrow sum + \text{ข้อมูลใน}$   
รายการลำดับที่  $I$   
3.2  $i \leftarrow i + 1$
4. แสดงค่า  $sum / \text{จำนวนข้อมูล}$   
ในรายการ





## ตัวอย่างที่ 3.2 หานำหนักค่าเฉลี่ย ของเพื่อในห้องด้วยรายการ

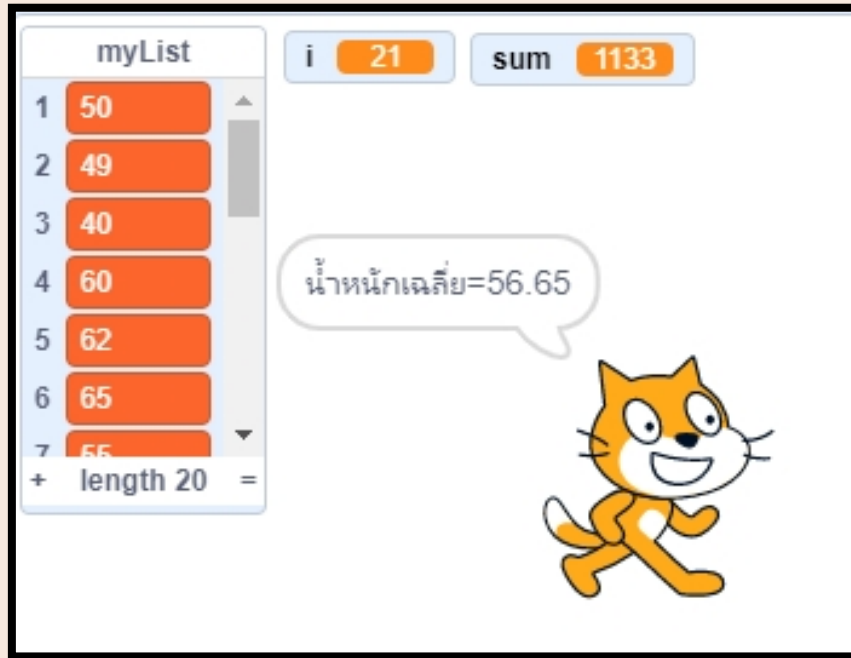
โปรแกรมมีดังนี้





## ตัวอย่างที่ 3.2 หาน้ำหนักค่าเฉลี่ย ของเพื่อนในห้องด้วยรายการ

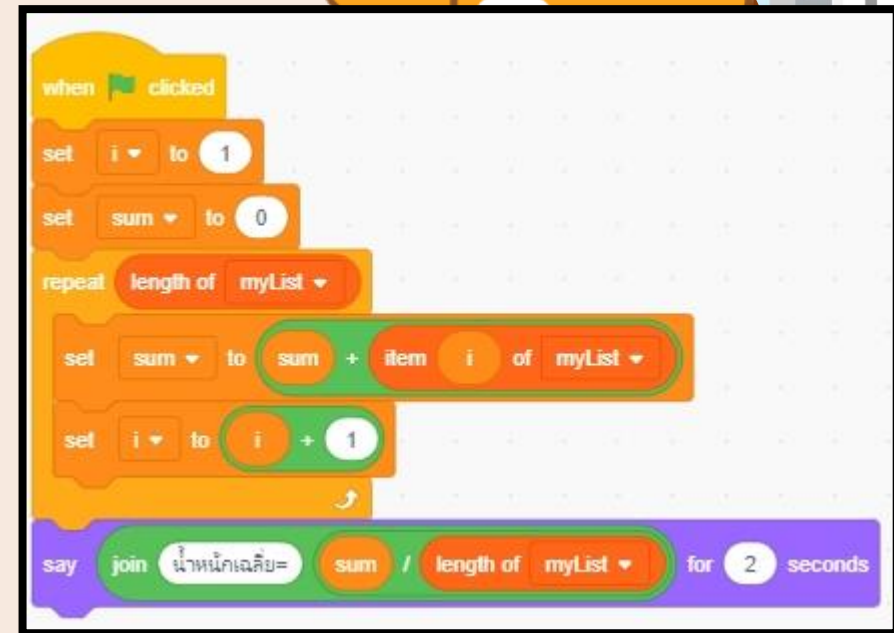
ผลลัพธ์ที่ได้



## ตัวอย่างที่ 3.2 หาน้ำหนักค่าเฉลี่ย ของเพื่อนในห้องด้วยรายการ

อธิบายการทำงานได้ดังนี้

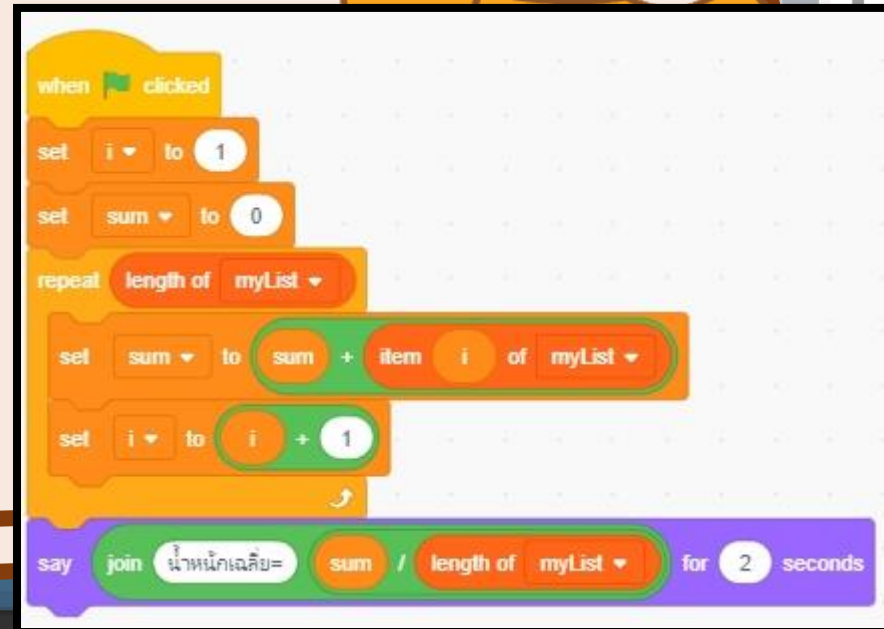
1. บรรทัดที่ 2 กำหนดค่าเริ่มต้นให้ตัวแปร  $i$  เพื่อเก็บลำดับของข้อมูลในรายการ
2. บรรทัดที่ 3 กำหนดค่าเริ่มต้นให้ตัวแปร  $sum$  เพื่อเก็บผลรวมน้ำหนัก



```
when clicked
  set i to 1
  set sum to 0
  repeat (length of myList)
    set sum to sum + item i of myList
    set i to i + 1
  say join น้ำหนักเฉลี่ย= sum / length of myList for 2 seconds
```

## ตัวอย่างที่ 3.2 หานำหนักค่าเฉลี่ย ของเพื่อนในห้องด้วยรายการ

3. บรรทัดที่ 4 ทำซ้ำในคำสั่งบรรทัดที่ 5-6 จนหมดข้อมูลในรายการ myList โดยบรรทัดที่ 5 นำผลรวมของตัวแปร sum กับข้อมูลที่อยู่ใน myList ลำดับที่ i ไปเก็บไว้ในตัวแปร sum

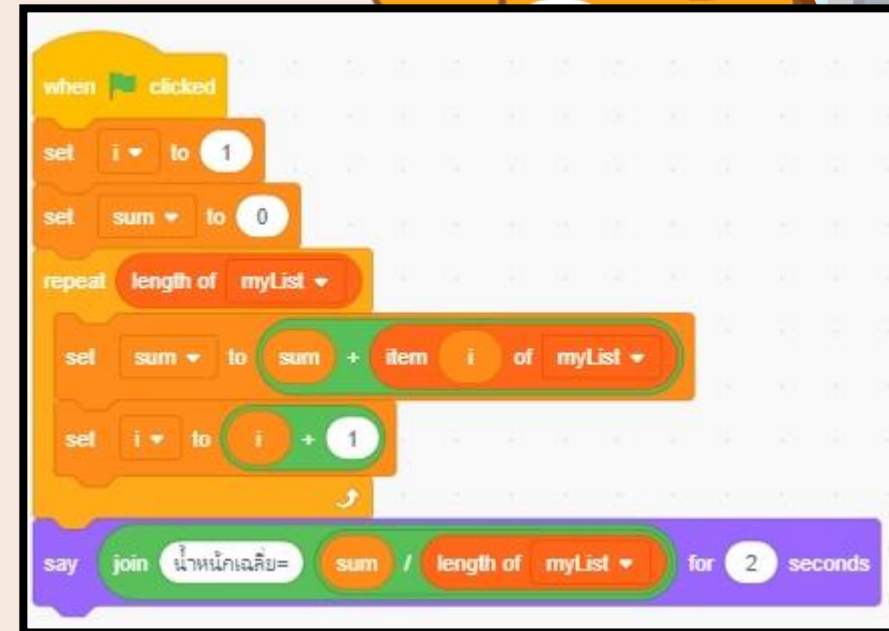


```
when clicked
  set i to 1
  set sum to 0
  repeat (length of myList)
    set sum to (sum + item i of myList)
    set i to (i + 1)
  say join นำหนักเฉลี่ย= sum / length of myList for 2 seconds
```



ตัวอย่างที่ 3.2 หาน้ำหนักค่าเฉลี่ย  
ของเพื่อนในห้องด้วยรายการ

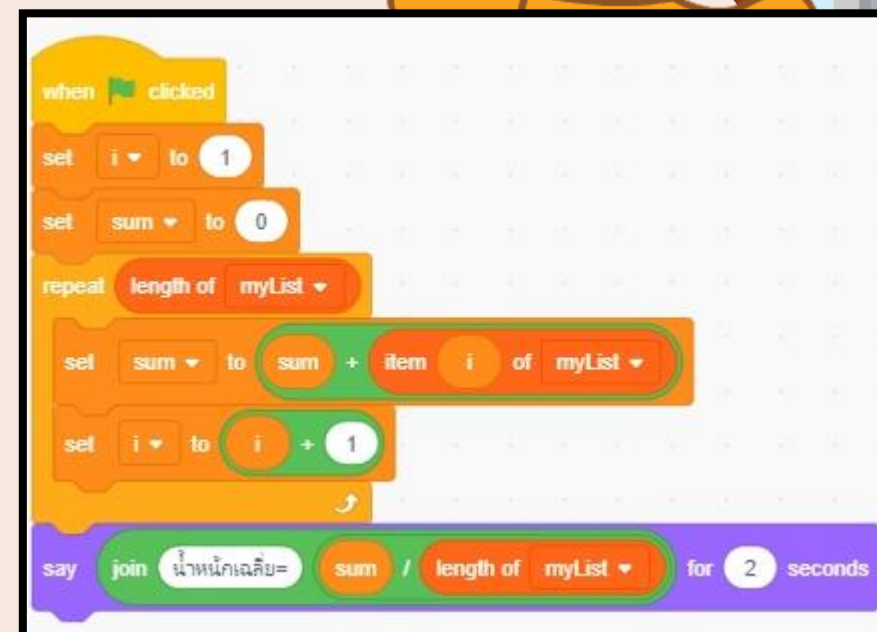
บรรทัดที่ 6 เพิ่มค่า  $i$  อีก 1  
เพื่อให้การทำงานในรอบถัดไปมีการ  
นำข้อมูลในรายการถัดไป ( $i+1$ ) มา  
บวกกับผลรวมน้ำหนัก  $sum$  เมื่อจบ  
การทำงานของการทำซ้ำจะได้  
ผลรวมน้ำหนักของนักเรียนทุกคน





### ตัวอย่างที่ 3.2 หาน้ำหนักค่าเฉลี่ย ของเพื่อในห้องด้วยรายการ

4. บรรทัดที่ 7 แสดงข้อความ  
“น้ำหนักเฉลี่ย =” ตามด้วยน้ำหนัก  
เฉลี่ยของนักเรียนที่เกิดจากผลรวม  
ของน้ำหนักหารด้วยจำนวนนักเรียน  
(จำนวนข้อมูลในรายการ myList)



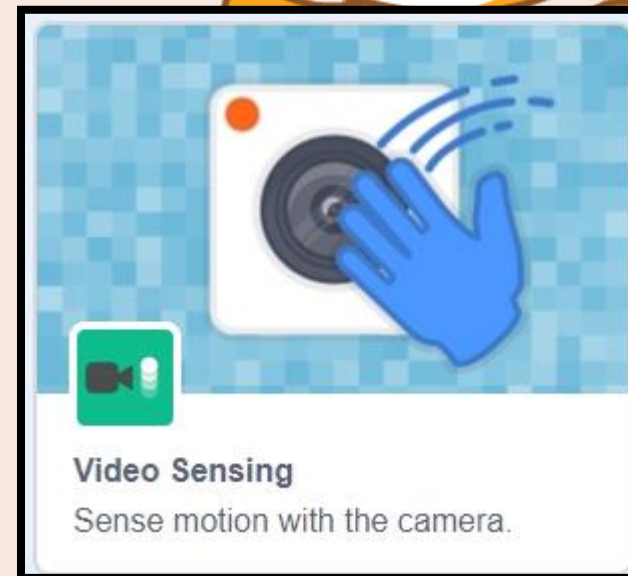


# การเขียนโปรแกรม ร่วมกับวิดีโอ




## การเขียนโปรแกรมร่วมกับวิดีโอ

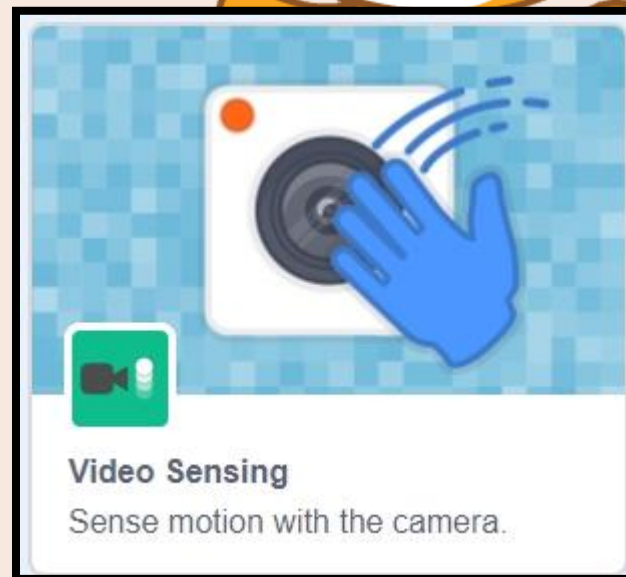
ใน Scratch มีบล็อกคำสั่งที่  
ใช้งานกับกล้องที่ติดอยู่กับเครื่อง  
คอมพิวเตอร์ที่ใช้เขียนโปรแกรม  
โดยอยู่ใน **กลุ่มบล็อก Video  
Sensing**





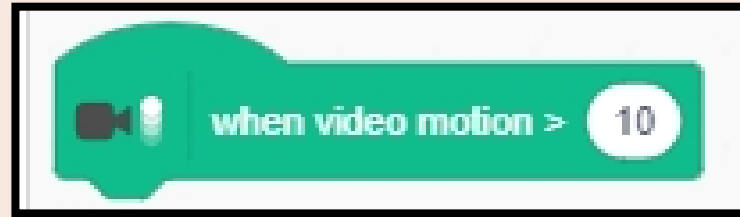
## การเขียนโปรแกรมร่วมกับวิดีโอ

โดยกลุ่มบล็อกคำสั่ง **Video Sensing** นักเรียนจะต้องคลิกในส่วน **Add Extension** หรือปุ่ม  ซึ่งอยู่ด้านล่างซ้ายของหน้าต่างโปรแกรม





## กลุ่มบล็อกคำสั่ง Video Sensing



ใช้สำหรับการตรวจสอบการเปลี่ยนแปลงของภาพ  
วิดีโอบริเวณที่อยู่ภายใต้ตัวละคร ถ้ามีการเปลี่ยนแปลง  
มากกว่าค่าที่กำหนดจะทำคำสั่งที่อยู่ภายใต้บล็อกนี้ โดยผู้ใช้  
สามารถกำหนดค่าได้ระหว่าง 0 (ไม่มีการเคลื่อนไหว) ถึง  
100 (มีการเคลื่อนไหวสูงสุด)

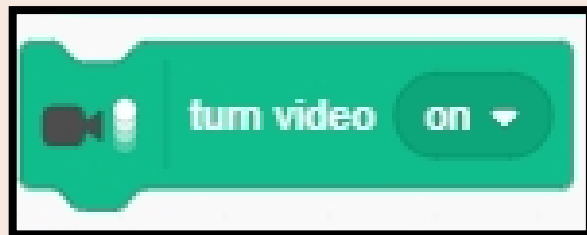
## กลุ่มบล็อกคำสั่ง Video Sensing



### กำหนดลักษณะการแสดงภาพวิดีโอ

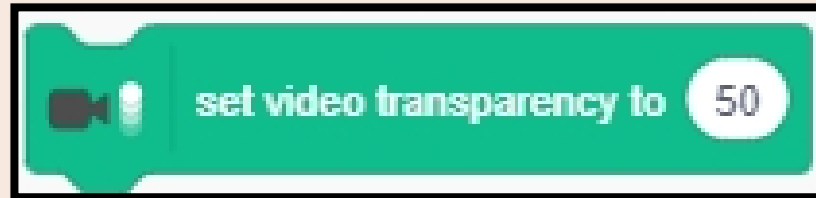
- **motion** ระดับการเคลื่อนไหวของภาพวิดีโอที่สัมผัสตัวละครหรือเวที มีค่าอยู่ระหว่าง 0 ถึง 100
- **direction** ทิศทางที่ภาพวิดีโอเคลื่อนผ่านตัวละครหรือเวทีโดยรวมมีค่าอยู่ระหว่าง -180 ถึง 180 เหมาะกับการนำมาใช้งานร่วมกับบล็อก point to direction เพื่อเปลี่ยนทิศทางตัวละครไปตามทิศทางที่ภาพวิดีโอเคลื่อนที่

## กลุ่มบล็อกคำสั่ง Video Sensing



ใช้เปิดหรือปิดการทำงานของกล้องวิดีโอ โดยเมื่อ  
สั่งเปิด ภาพจากกล้องจะถูกนำมาแสดงเป็นพื้นหลังของฉาก  
นอกจากนั้นยังสามารถกำหนดการแสดงภาพในลักษณะที่  
กลับด้านซ้ายขวา โดยกำหนดเป็น on flipped

## กลุ่มบล็อกคำสั่ง Video Sensing



ปรับความโปร่งใสของวิดีโอในระดับ 0% (มองเห็นภาพวิดีโอชัดเต็มที่) จนถึง 100% (มองไม่เห็นภาพวิดีโอ)



## ตัวอย่างที่ 3.3 เกม Don't more

นักเรียนต้องการเขียนโปรแกรมเกม ที่มีการตรวจสอบการเคลื่อนไหวใบหน้า หากมีการเคลื่อนไหว คะแนนจะลดลงครั้งละ 1 คะแนน โดยมีคะแนนเริ่มต้น 10 คะแนน

- การวิเคราะห์และกำหนดรายละเอียดของปัญหา

ข้อมูลเข้า

คือ การเคลื่อนไหว  
ของใบหน้าผู้เล่น

ข้อมูลออก

คือ คะแนนที่เหลือ

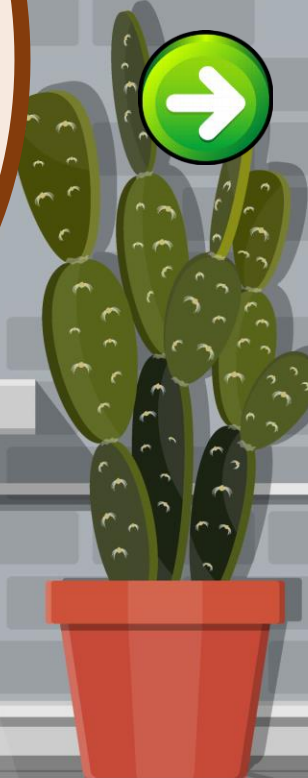
## ตัวอย่างที่ 3.3 เกม Don't more



วิธีการตรวจสอบ

มีดังนี้

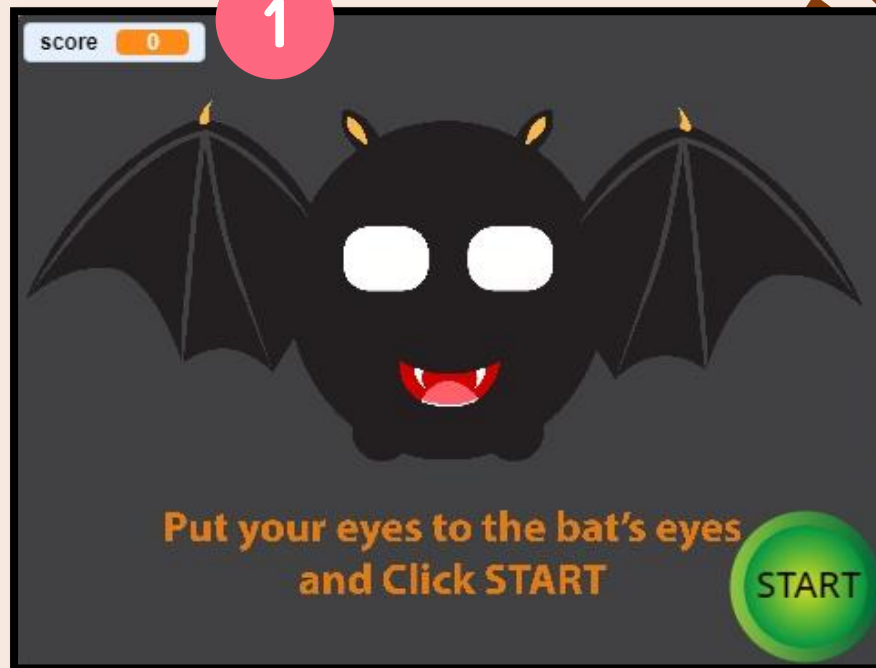
- ครั้งที่ 1 เมื่อเคลื่อนไหวไปหน้า คะแนน = 9
- ครั้งที่ 2 เมื่อเคลื่อนไหวไปหน้า คะแนน = 8
- ครั้งที่ 3 เมื่อเคลื่อนไหวไปหน้า คะแนน = 7



## ตัวอย่างที่ 3.3 เกม Don't more

### การวางแผนการแก้ปัญหา

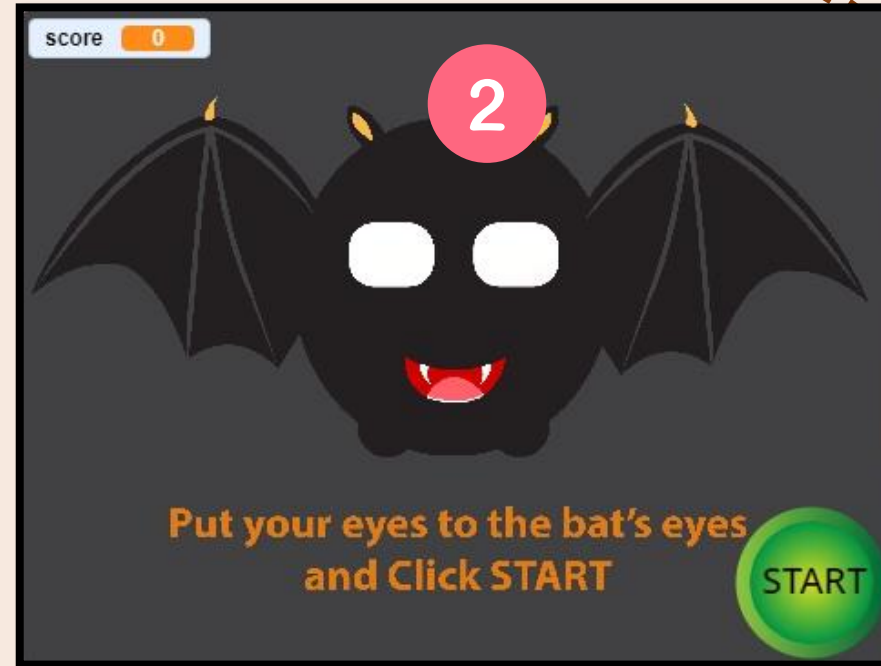
1. ส่วนที่แสดงคะแนน  
โดยใช้การแสดงค่าตัวแปรเมื่อมี  
การเคลื่อนไหวคะแนนจะลดลง  
ครั้งละ 1 คะแนน







## ตัวอย่างที่ 3.3 เกม Don't more

2. ตัวละครหน้ากาก  
โดยอาจออกแบบให้เป็นที่วาง  
ดวงตาของผู้เล่นให้ตรงช่อง  
ดวงตาของค้างคาว





## ตัวอย่างที่ 3.3 เกม Don't more

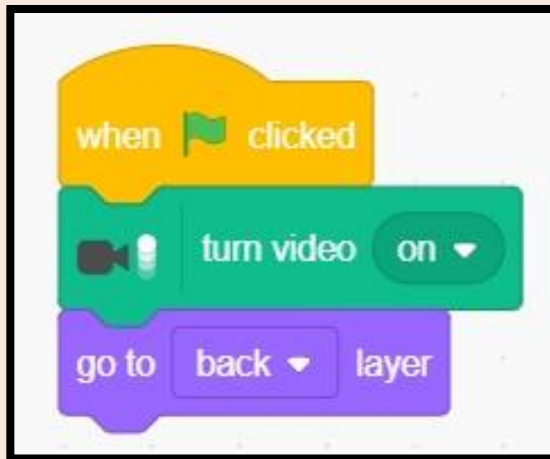
3. ตัวละคร START ใช้  
สำหรับคลิกเมื่อต้องการเริ่มเล่นเกม  
โดยกำหนดให้มีชุดตัวละคร 2 ชุด  
คือ ชุดตัวละครที่แสดงเมื่อยังไม่มี  
การคลิกปุ่ม  กับชุดตัวละครที่  
แสดงหลังจากมีการคลิกปุ่ม 



## ตัวอย่างที่ 3.3 เกม Don't more

สามารถเขียนโปรแกรมได้ดังนี้

ตัวละครหน้ากาก



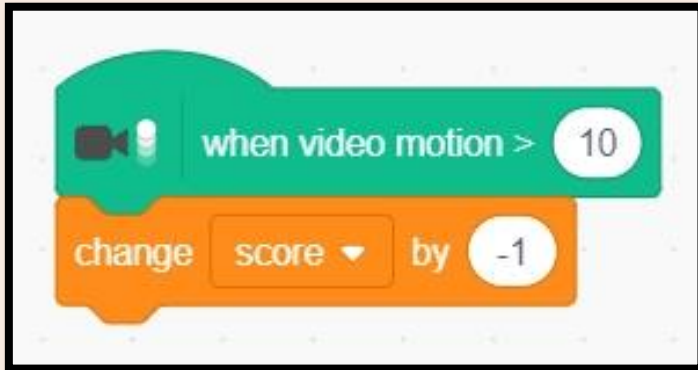
โปรแกรมหลัก อธิบายได้ดังนี้

1. บรรทัดที่ 1 กำหนดจุดเริ่มต้น เมื่อผู้ใช้คลิกปุ่มธงเขียว
2. บรรทัดที่ 2 กำหนดให้วิดีโอทำงาน
3. บรรทัดที่ 3 กำหนดให้ตัวละครหน้ากากอยู่ในเลเยอร์ล่างสุด



## ตัวอย่างที่ 3.3 เกม Don't more

สามารถเขียนโปรแกรมได้ดังนี้



ตัวละครหน้ากาก



เหตุการณ์เมื่อมีการเคลื่อนไหว

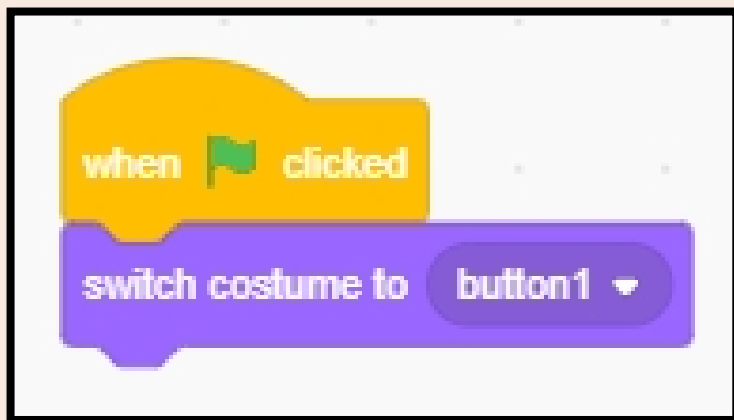
1. บรรทัดที่ 1 ตรวจสอบระดับการเคลื่อนไหวของตัวละคร ถ้ามีค่ามากกว่า 10 ให้ไปทำบรรทัดที่ 2
2. บรรทัดที่ 2 ตัวแปร score จะมีค่าลดลงจากเดิม 1





## ตัวอย่างที่ 3.3 เกม Don't more

สามารถเขียนโปรแกรมได้ดังนี้



ตัวละคร START



โปรแกรมหลักอธิบายได้ดังนี้

1. บรรทัดที่ 1 กำหนดจุดเริ่มต้น เมื่อผู้ใช้คลิกปุ่มธงเขียว
2. กำหนดชุดตัวละคร button 1

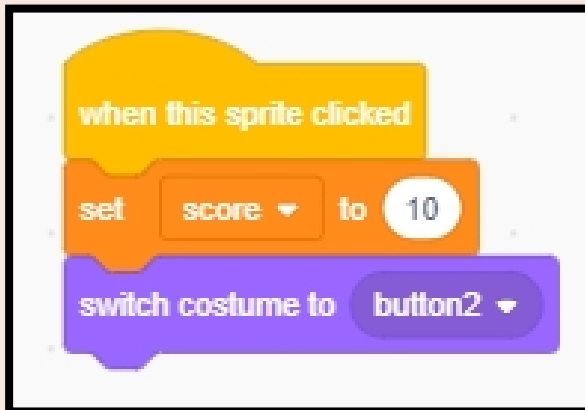




## ตัวอย่างที่ 3.3 เกม Don't more

สามารถเขียนโปรแกรมได้ดังนี้

ตัวละคร START



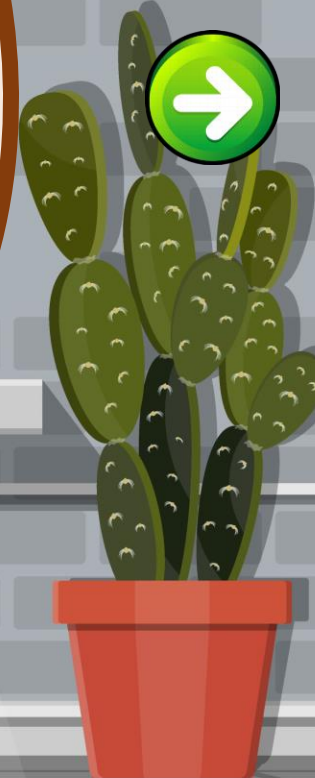
ส่วนตรวจสอบการคลิกปุ่ม

1. บรรทัดที่ 1 ตรวจสอบหากมีการคลิกที่ตัวละคร START จะทำคำสั่งที่บรรทัดที่ 2 ต่อไป
2. บรรทัดที่ 2 กำหนดตัวแปร score เท่ากับ 10
3. บรรทัดที่ 3 เปลี่ยนชุดตัวละครเป็น button2 เพื่อแสดงการเริ่มเล่นเกม



# ชวนคิด

นักเรียนสามารถเขียน  
โปรแกรม Scratch ให้มีเสียงพูด  
ตามข้อความที่ระบุและแปลเป็น  
ภาษาต่างๆ ได้หรือไม่ อย่างไร



# การสร้างแอปพลิเคชัน ด้วย Scratch



## การสร้างแอปพลิเคชัน ด้วย Scratch

หลังจากที่ได้เรียนรู้การใช้  
งานบล็อกคำสั่งต่างๆ นักเรียน  
สามารถนำมาสร้างเป็นแอปพลิเคชัน  
หรือซอฟต์แวร์ประยุกต์อย่างง่ายได้  
ดังตัวอย่างต่อไปนี้





## ตัวอย่างที่ 3.4 เกมสอยดาว

ในงานโรงเรียนมีกิจกรรมต่างๆ  
จำนวนมาก กิจกรรมหนึ่งที่จัดขึ้น คือ  
กิจกรรมสอยดาว มีสลากเป็นรูปดาวแขวนไว้  
บนต้นไม้ และในสลากรูปดาว จะมีของ  
รางวัลต่างๆ ราคาสอยดาว ดวงละ 20 บาท  
เมื่อพิจารณาสถานการณ์แล้วสามารถแบ่ง  
ปัญหาใหญ่เป็นปัญหาย่อยได้ดังนี้



## ตัวอย่างที่ 3.4 เกมสอยดาว

### คำถาม (Question)

1. บัตรสอยดาวราคาเท่าไร

2. มีกี่รางวัล อะไรบ้าง  
และรางวัลละกี่ชิ้น

3. หากลูกค้าสอยดาวแล้วไม่ได้รับ  
รางวัลที่กำหนด จะมีรางวัลปอบใจ  
หรือไม่ ถ้ามีรางวัลจะเป็นอะไร

### คำตอบ (Answer)

บัตรราคา 20 บาท

มี 5 รางวัล คือ ข้าวสาร โทรศัพท์มือถือ  
จักรยาน หม้อหุงข้าว บัตรกำนัลมูลค่า  
500 บาท รางวัลละ 1 ชิ้น

หากลูกค้าสอยดาวแล้วไม่ได้รับรางวัลที่  
กำหนด จะได้ลูกอมเป็นรางวัลปอบใจ

## ตัวอย่างที่ 3.4 เกมสอยดาว

### คำถาม (Question)

4. หน้าจอปรากฏดาวกี่แบบ อะไรบ้าง  
แบบละกี่ดวง

5. ดาวปรากฏที่ใด และปรากฏอย่างไร

6. ค่าที่ใช้ในการสุ่มรางวัล  
ของดาวอยู่ในช่วงใด

### คำตอบ (Answer)

ปรากฏดาว 1 แบบ ที่มีสีแตกต่างกัน  
ทั้งหมดจำนวน 20 ดวง

ดาวปรากฏบนต้นไม้  
โดยการสุ่มพิกัดบนต้นไม้ที่ต้องการ

สุ่มรางวัลอยู่ในช่วง 1 ถึง 20



## ตัวอย่างที่ 3.4 เกมสอยดาว

### คำถาม (Question)

7. การสอยดาวใช้วิธีการใด

8. การทำงานเมื่อมีการสอยดาวมี  
อะไรบ้าง

### คำตอบ (Answer)

การคลิกดาว

- ดาวที่ถูกสอยจะหายไป
- แสดงผลที่ได้จากการลุ่มของรางวัล  
ถ้าผลของการลุ่มได้รับรางวัลที่กำหนด  
ให้ลบรายชื่อรางวัลนั้นออกจาก  
รายการ ถ้าไม่ได้จะแสดง “ลูกอม”
- รายได้จากการสอยดาวเพิ่มขึ้นครั้งละ  
20 บาท

## ตัวอย่างที่ 3.4 เกมสอยดาว

- ❑ การวิเคราะห์และกำหนดรายละเอียดของปัญหา

ข้อมูลเข้า

คือ ดาวที่ถูกคลิก

ข้อมูลออก

คือ รางวัลที่ได้และ  
รายได้จากการขาย



## ตัวอย่างที่ 3.4 เกมสอยดาว

วิธีการตรวจสอบ

มีดังนี้



- ครั้งที่ 1 คลิกดาว แล้วแสดงรางวัลที่ได้จากการลุ่ม เช่น ได้ “ข้าวสาร” แสดงรายได้จากการขาย 20 บาท ลบ รายชื่อข้าวสารออกจากรายการและลบดาวออก 1 ดวง
- ครั้งที่ 2 คลิกดาว แล้วแสดงรางวัลที่ได้จากการลุ่ม เช่น ได้รางวัลพลอบใจ เป็น “ลูกอม” แสดงรายได้จากการขาย เป็น 40 บาท และลบดาวออก 1 ดวง





## ตัวอย่างที่ 3.4 เกมสอยดาว

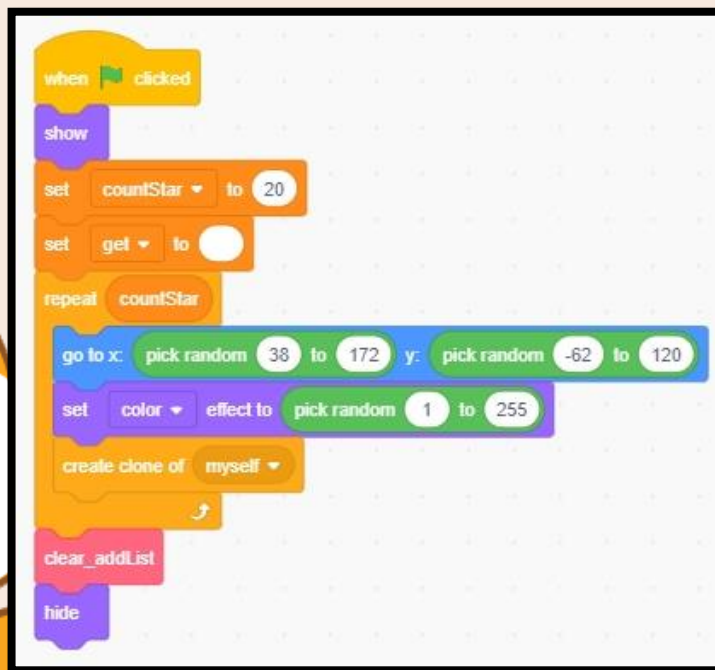
การวางแผนการแก้ปัญหา

การออกแบบหน้าจอโปรแกรมได้ดังนี้



## ตัวละครดาวมีโปรแกรม 3 ส่วนดังนี้

### 1. โปรแกรมหลักเมื่อมีการคลิกธงเขียว



### อธิบายการทำงานได้ดังนี้

1. บรรทัดที่ 2 จะแสดงตัวละคร
2. บรรทัดที่ 3-4 กำหนดค่าเริ่มต้นให้ตัวแปร countStar เป็น 20 เพื่อกำหนดจำนวนดาวทั้งหมด และกำหนดค่าตัวแปร get เพื่อใช้แสดงของรางวัลที่ได้ โดยกำหนดค่าเริ่มต้นเป็นช่องว่าง

## ตัวละครดาวมีโปรแกรม 3 ส่วนดังนี้

3. บรรทัดที่ 5 จะทำซ้ำคำสั่งในบรรทัดที่ 6-8 จำนวน countStar รอบ โดยบรรทัดที่ 6 จะสุ่มพิกัด x ในช่วง [38,172] และสุ่มพิกัด y ในช่วง [-62 , 120] สำหรับบรรทัดที่ 7 จะกำหนดสีตัวละครจากค่าที่ได้จากการสุ่มในช่วง [1,225] และบรรทัดที่ 8 จะโคลนตัวละครดาว
4. บรรทัดที่ 9 เรียกฟังก์ชัน clear\_addList และบรรทัดที่ 10 ซ่อนตัวละครดาวที่ไม่ได้เกิดจากการโคลน

```
when clicked
show
set countStar to 20
set get to 
repeat countStar
  go to x: pick random 38 to 172 y: pick random -62 to 120
  set color effect to pick random 1 to 255
  create clone of myself
clear_addList
hide
```



## ตัวละครดาวมีโปรแกรม 3 ส่วนดังนี้

### 2. ฟังก์ชัน clear\_addList



### อธิบายการทำงานได้ดังนี้

1. บรรทัดที่ 1 ประกาศส่วนหัวฟังก์ชัน clear\_addList
2. บรรทัดที่ 2 ลบข้อมูลทั้งหมดในรายการ award
3. บรรทัดที่ 3-7 เพิ่มข้อมูลในรายการ award เป็นข่าวสาร โทรศัพท์มือถือ จักรยาน หม้อหุงข้าว บัตรกำนัลมูลค่า 500 บาท ตามลำดับ

## ตัวละครดาวมีโปรแกรม 3 ส่วนดังนี้

### 3. เหตุการณ์เมื่อคลิกตัวละครดาว



#### อธิบายการทำงานได้ดังนี้

1. บรรทัดที่ 2 ตรวจสอบว่าหากไม่มีข้อมูลในรายการรางวัล ให้ทำคำสั่งในบรรทัดที่ 3 คือ ซ่อนตัวละครดาว
2. บรรทัดที่ 4 กำหนดตัวแปร ran เพื่อเก็บค่าที่ได้จากการสุ่มรางวัลในช่วง [1,countStar] ในที่นี้ countStar มีค่า 20

## ตัวละครดาวมีโปรแกรม 3 ส่วนดังนี้



```
when this sprite clicked
  if length of award = 0 then
    hide
  set ran to pick random 1 to countStar
  change countStar by -1
  if ran > length of award then
    set get to ลูกอม
  else
    set get to item ran of award
    delete ran of award
  set total to total + 20
  delete this clone
```

3. บรรทัดที่ 5 ลดค่าตัวแปร countStar ลง 1 เนื่องจากดาวถูกคลิก
4. บรรทัดที่ 6 ตรวจสอบว่าค่าที่สุ่มในบรรทัดที่ 4 มากกว่าจำนวนรางวัลที่มีในรายการ award หรือไม่ ถ้าใช่ ลูกค้าจะได้รับลูกอม จากคำสั่งในบรรทัดที่ 7 ไม่เช่นนั้น จะทำคำสั่งในบรรทัดที่ 8-9 คือให้แสดงข้อมูลในรายการที่มีค่าตรงกับลำดับที่สุ่มได้จากคำสั่งในบรรทัดที่ 4 และลบตัวละครโคลนออก
5. บรรทัดที่ 11 รวมเงินรายได้จากการขายดาวดวงละ 20 บาท และบรรทัดที่ 12 ลบตัวละครโคลนรูปดาว



## สรุปท้ายบท

การสร้างแอปพลิเคชันหรือซอฟต์แวร์ประยุกต์ให้ผู้อื่นได้ใช้งานมีหลายรูปแบบ Scratch เป็นภาษาโปรแกรมเชิงกราฟิกที่นำมาพัฒนาแอปพลิเคชันอย่างง่ายได้โดยมีบล็อกคำสั่งต่างๆ ให้เลือกใช้งานจำนวนมากสามารถใช้งานได้สะดวก และทำให้โปรแกรมมีความน่าสนใจ เช่น มีการใช้งานบล็อก List ที่สามารถนำข้อมูลในรายการมาประมวลผลเขียนโปรแกรมร่วมกับวิดีโอ ต่อบันทึกการสัมผัสใช้งานร่วมกับปุ่มบนคีย์บอร์ด ซึ่งจะทำให้การพัฒนาแอปพลิเคชันมีความน่าสนใจและทำได้สะดวก



ผู้จัดทำ

นายกานตพงศ์ สุวรรณทา  
ตำแหน่ง ครู

