

การแก้ปัญหาด้วย Scratch

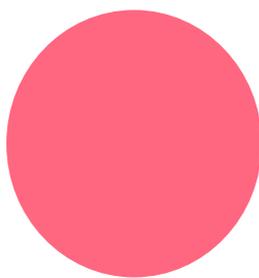
ฟังก์ชัน

โปรแกรมขนาดใหญ่ประกอบด้วยคำสั่งจำนวนมากเพื่อทำงานหลายหน้าที่ หากเขียนคำสั่งเหล่านั้นต่อเนื่องเรียงต่อกันไป จะทำให้การตรวจสอบและแก้ไขโปรแกรมทำได้ยาก ถ้านักเรียนจัดกลุ่มคำสั่งเหล่านี้ให้เป็นโปรแกรมย่อย ที่ทำงานเฉพาะอย่างจะทำให้ง่ายต่อการตรวจสอบและแก้ไข

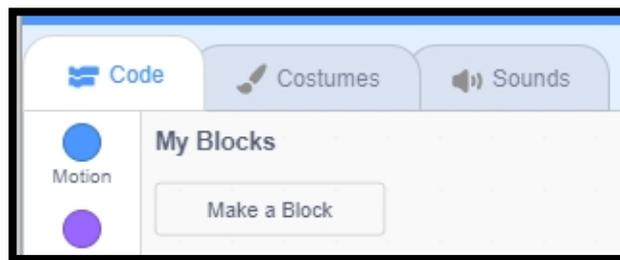
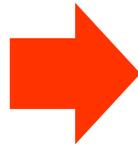
อีกทั้งยังสามารถนำโปรแกรมย่อยที่มีอยู่แล้วไปใช้ในโปรแกรมอื่น ที่มีลักษณะคล้ายคลึงได้ ใน scratch มีคำสั่งสร้างโปรแกรมย่อย (subroutine) หรือฟังก์ชัน (function) ให้ใช้งาน โดยผู้ใช้จะนิยามชื่อฟังก์ชันเพื่อใช้อ้างอิงในการเรียกใช้งาน และกำหนดคำสั่งที่ต้องการภายในฟังก์ชันนั้น ชื่อฟังก์ชันควรกำหนดให้เหมาะสมกับหน้าที่การทำงานเพื่อให้ง่ายต่อการทำความเข้าใจ

การสร้างฟังก์ชัน

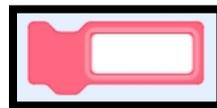
การสร้างฟังก์ชันมีขั้นตอนดังต่อไปนี้



คลิกปุ่มบล็อก My Blocks

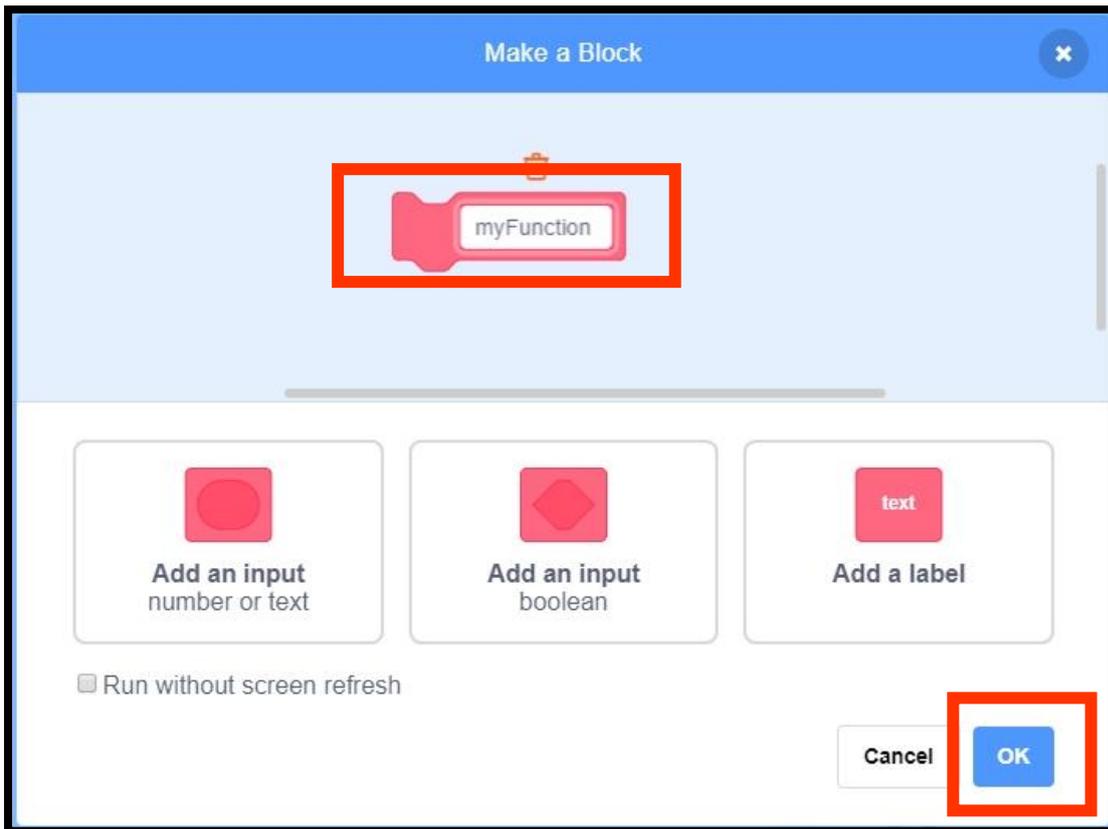


คลิก Make a Block

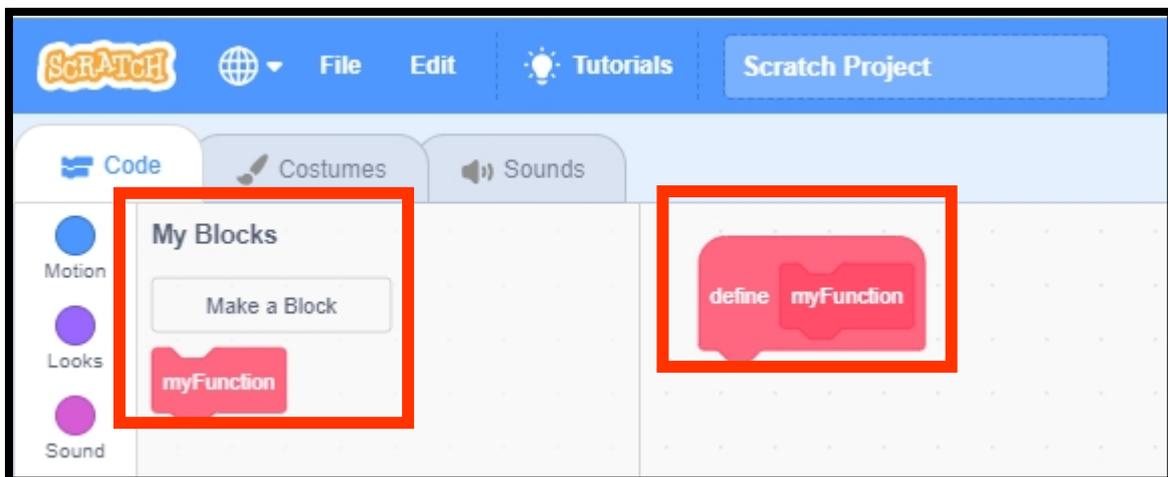


จะปรากฏหน้าต่าง New Block ให้พิมพ์ชื่อฟังก์ชันใน แล้วคลิกปุ่ม OK

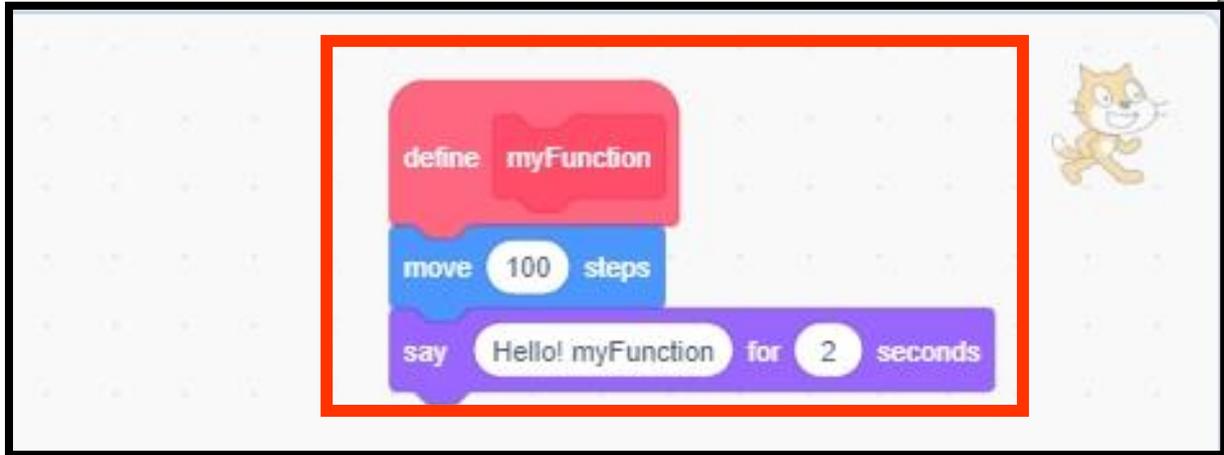
ในที่นี้ให้พิมพ์ myFunction



จะปรากฏบล็อกชื่อฟังก์ชันที่สร้างขึ้น และบล็อกคำสั่งที่นิยามส่วนหัวของฟังก์ชัน (define) ในพื้นที่เขียนโปรแกรม

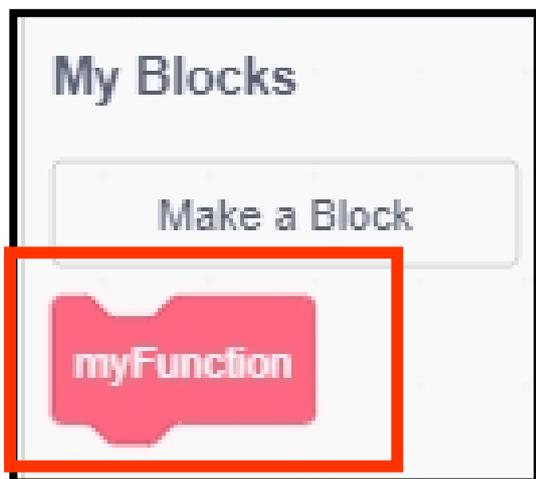


เมื่อสร้างบล็อกฟังก์ชันไว้ที่ตัวละครใด การเรียกใช้งานจะสามารถทำได้เฉพาะในส่วนพื้นที่เขียนโปรแกรมของตัวละครนั้นเท่านั้น ตัวละครอื่นจะไม่สามารถนำไปใช้ได้

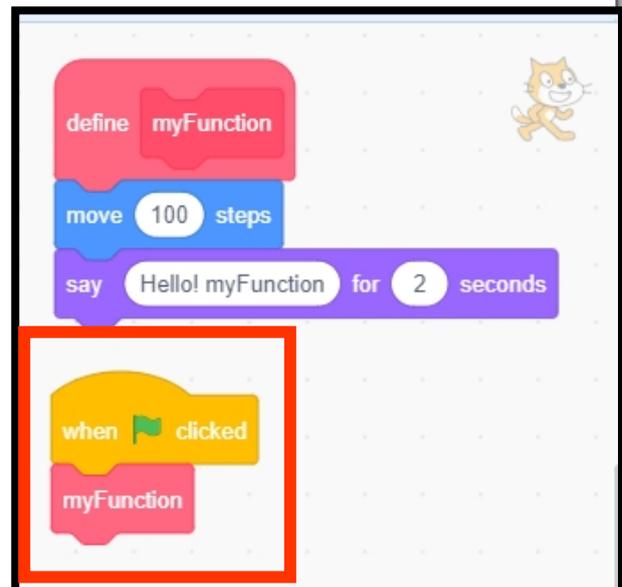


นักเรียนสามารถลากคำสั่งที่ต้องการมาต่อส่วนหัวของฟังก์ชันได้ดังรูปด้านบนนี้

เมื่อสร้างฟังก์ชันแล้วสามารถเรียกใช้งานฟังก์ชันได้ดังขั้นตอนต่อไปนี

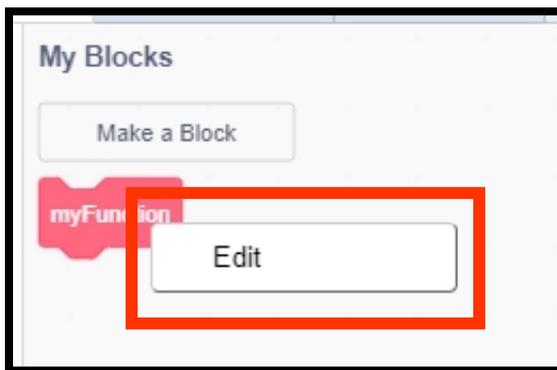


คลิกบล็อกฟังก์ชันที่ต้องการ

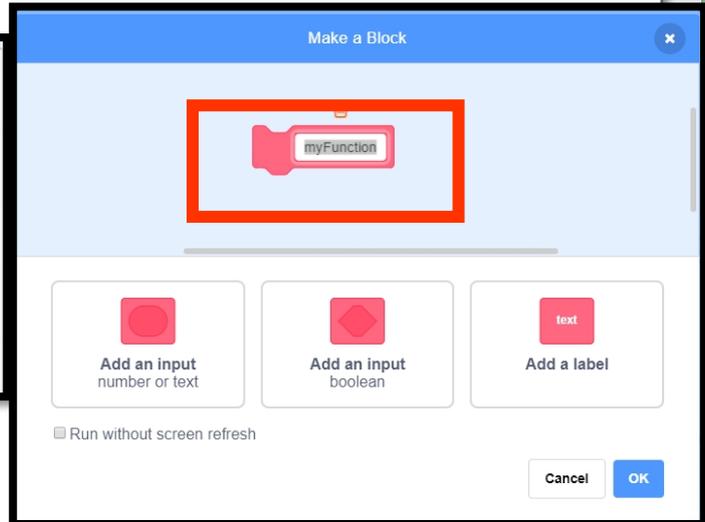


ลากวางในพื้นที่เขียนโปรแกรม

หากต้องการแก้ไขชื่อฟังก์ชันสามารถทำได้ดังขั้นตอนต่อไปนี

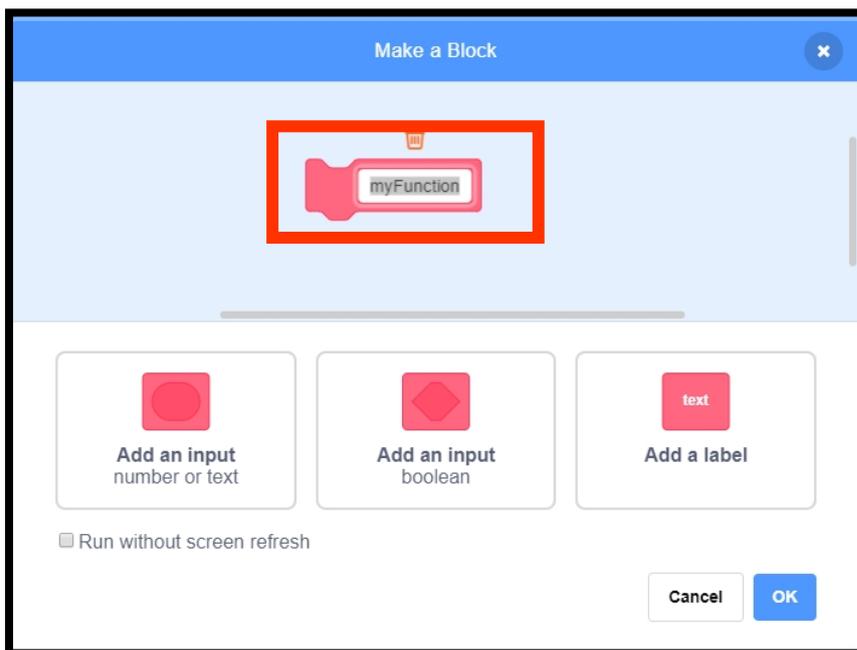


คลิกขวาที่บล็อกฟังก์ชัน
ที่สร้างขึ้น แล้วเลือก edit

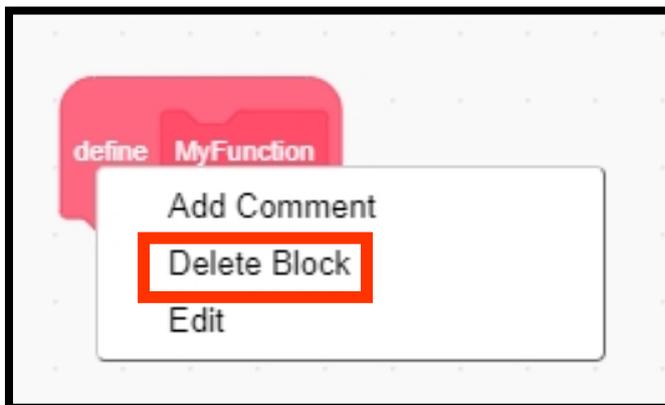


เมื่อปรากฏหน้าต่าง Edit Block
แก้ไขเป็นชื่อฟังก์ชันที่ต้องการแล้วคลิกปุ่ม OK

เมื่อมีการเปลี่ยนชื่อฟังก์ชัน โปรแกรม scratch จะเปลี่ยนชื่อฟังก์ชัน ในส่วนที่มีการอ้างอิงทุกที่ให้โดยอัตโนมัติ

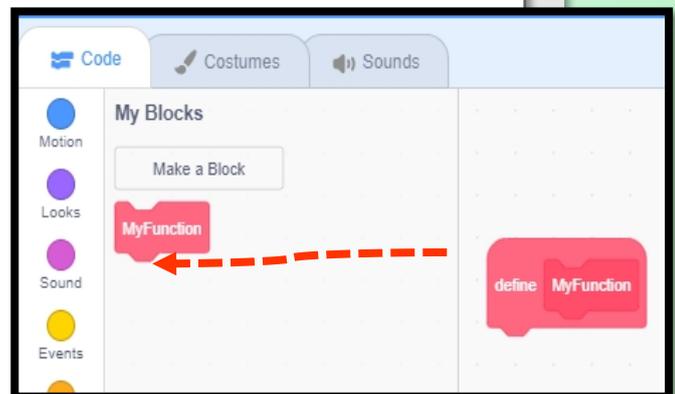


การลบฟังก์ชัน



การลบฟังก์ชันสามารถทำได้โดยคลิกขวา

ที่ฟังก์ชันแล้วเลือก Delete



หรือลากฟังก์ชันออกจากพื้นที่เขียนโปรแกรม

มาวางในพื้นที่แสดงบล็อกคำสั่ง

ตัวอย่างที่ 1 การสร้างเกมตัดแต่งโม

ให้นักเรียนสร้างเกมตัดแต่งโมอย่างง่าย โดยเมื่อเริ่มต้นให้แสดงภาพแต่งโมที่ละลูก ที่มีขนาดและตำแหน่งแตกต่างกัน เมื่อผู้ใช้คลิกเมาส์ที่แต่งโม ให้แสดงภาพแต่งโมลูกนั้นถูกตัด และมีการเพิ่มคะแนน หากผู้เล่นไม่สามารถคลิกแต่งโมได้ทันในช่วงเวลาที่กำหนด แต่งโมจะหายไป

ออกแบบส่วนของโปรแกรมโดยพิจารณาการแก้ปัญหาต่อไปนี้

- ทำอย่างไรให้แต่งโมปรากฏบนหน้าจอด้วยขนาดและตำแหน่งแตกต่างกัน
- ทำอย่างไรให้แต่งโมหายไปจากหน้าจอเมื่อเวลาผ่านไปประยะหนึ่ง
- ทราบได้อย่างไรว่าแต่งโมถูกตัด
- แสดงภาพแต่งโมที่ถูกตัดอย่างไร
- คิดคะแนนอย่างไรเมื่อตัดแต่งโมได้สำเร็จ 1 ลูก

สร้างแต่งโมให้ปรากฏในตำแหน่งสุ่ม

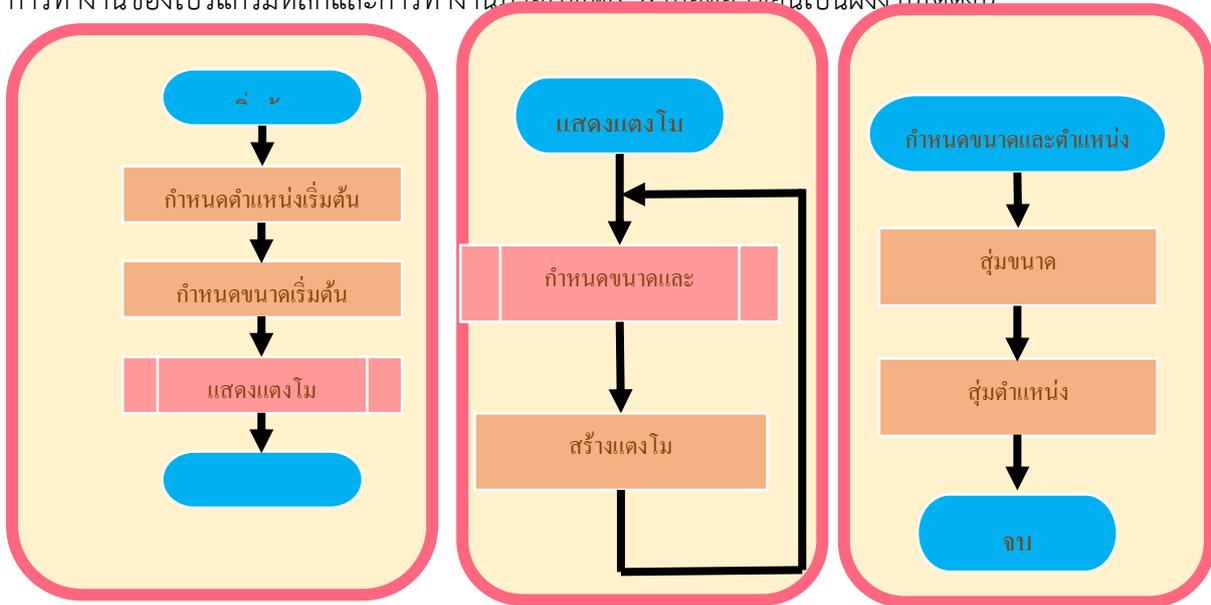
แต่งโมจะถูกสร้างขึ้นทีละลูกซึ่งอยู่ในตำแหน่งที่แตกต่างกันโดยการสุ่มแต่งโมแต่ละลูกจะมีขนาดไม่เท่ากัน

แนวคิด

กำหนดการทำงานเป็นส่วนย่อย ได้ดังนี้

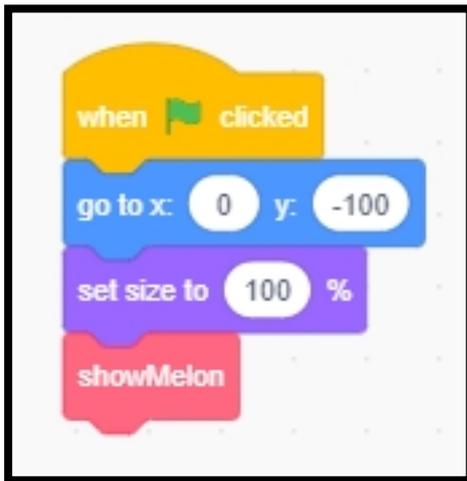
- แสดงแต่งโม
- กำหนดขนาดและตำแหน่งแต่งโม

การทำงานของโปรแกรมหลักและการทำงานภายในแต่ละส่วนย่อย เขียนเป็นผังงานได้ดังนี้



เขียนโปรแกรมได้ดังนี้

โปรแกรมหลักอธิบายได้ดังนี้



1. บรรทัดที่ 1 กำหนดจุดเริ่มต้นเมื่อผู้ใช้คลิกธงเขียว
2. บรรทัดที่ 2 กำหนดตำแหน่งเริ่มต้นเพื่อแสดงแตงโม โดยกำหนดที่ตำแหน่ง $x = 0$ และ $Y = -100$
3. บรรทัดที่ 3 กำหนดขนาดของตัวละครแตงโม ให้มีค่าเท่ากับขนาดจริง (100%) ของตัวละคร
4. บรรทัดที่ 4 เรียกฟังก์ชัน showMelon เพื่อแสดงแตงโม



ฟังก์ชัน showMelon สำหรับแสดงแตงโมอธิบายได้ดังนี้

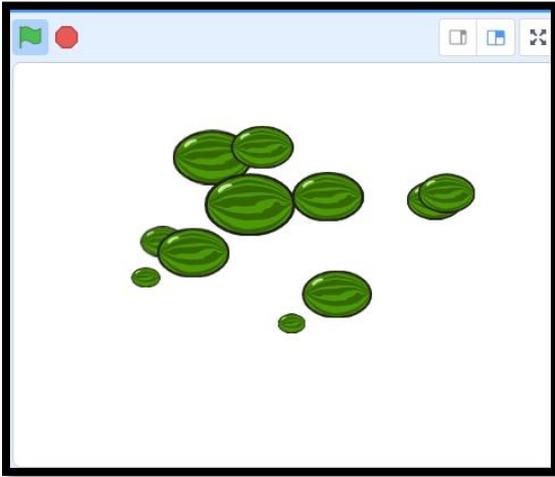
1. บรรทัดที่ 1 ประกาศส่วนหัวฟังก์ชัน showMelon
2. กำหนดให้มีการทำงานวนซ้ำแบบไม่รู้จบในบรรทัดที่ 3 ถึง 5
3. บรรทัดที่ 3 เรียกฟังก์ชัน setSizeAndPos เพื่อกำหนดขนาดและตำแหน่งของตัวละครแตงโม
4. บรรทัดที่ 4 สร้างแตงโมด้วยการโคลน (clone) ซึ่งเป็นการคัดลอกตัวละคร โดยใช้บล็อกคำสั่ง create clone of myself ซึ่งหมายถึง การโคลนตัวละครที่มีการใช้คำสั่งนี้ ในที่นี้คือโคลนตัวละครแตงโม
5. บรรทัดที่ 5 กำหนดให้มีการหน่วงเวลาในการสร้างแตงโม 1 วินาที ด้วยคำสั่ง wait 1 secs



ฟังก์ชัน setSizeAndPos สำหรับสุ่มขนาดและตำแหน่งของแตงโมอธิบายได้ดังนี้

1. บรรทัดที่ 1 ประกาศส่วนหัวฟังก์ชัน setSizeAndPos
2. บรรทัดที่ 2 กำหนดขนาดแตงโมด้วยการ สุ่มค่า 20-100 ซึ่งค่าที่สุ่มได้จะเป็นอัตราส่วนร้อยละของขนาดจริง เช่น ถ้าสุ่ม ได้ 30 จะหมายถึง แสดงตัวละครขนาด 30% ของขนาดจริง (100%)
3. บรรทัดที่ 3 กำหนดตำแหน่งของการแสดงแตงโมแกน x ด้วยการสุ่มค่าในช่วง -180 ถึง 180
4. บรรทัดที่ 4 กำหนดตำแหน่งของการแสดงแตงโมแกน y ด้วยการสุ่มค่าในช่วง - 150 ถึง 150

ผลลัพธ์ที่ได้คือ จะมีแตงโมปรากฏในตำแหน่งต่างๆจำนวนมากดังนี้

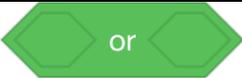


ตัวดำเนินการบูลีน

นักเรียนเคยเขียนโปรแกรมที่มีการคำนวณโดยใช้ตัวดำเนินการทางคณิตศาสตร์ $+$ $-$ $*$ $/$ และโปรแกรมที่มีการตรวจสอบเงื่อนไขที่มีการใช้ตัวดำเนินการเปรียบเทียบ $>$ $<$ $=$ มาแล้ว

ในส่วนนี้นักเรียนจะได้เรียนรู้เกี่ยวกับตัวดำเนินการบูลีน ที่ใช้เขียนโปรแกรมในกรณีที่มีเงื่อนไขมากกว่า 1 เงื่อนไข

ตัวดำเนินการบูลีนแสดงดังตาราง

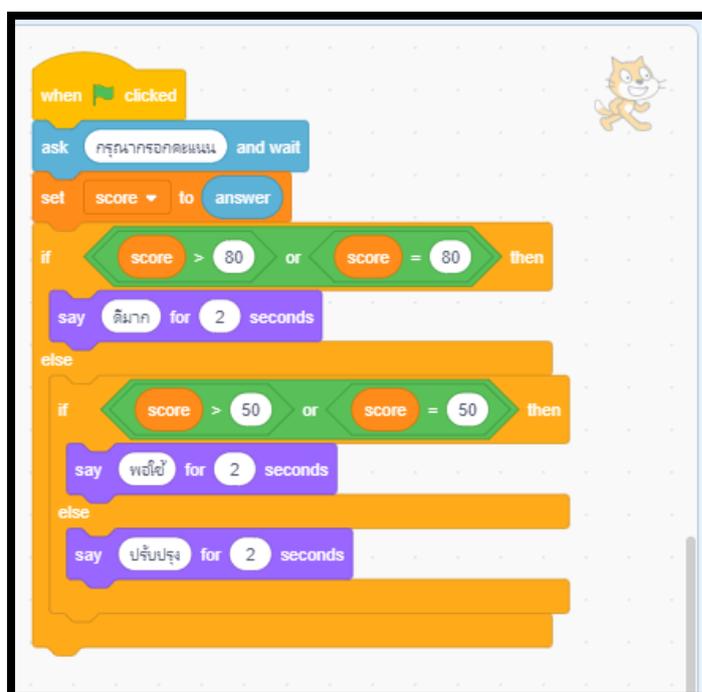
ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	สถานการณ์ที่เกิดขึ้นกับตัว ละคร/เวที		ผลการ ตรวจสอบ
			1	2	
	และ		สัมผัสตัวชี้เมาส์	สัมผัสสีส้ม	จริง
			สัมผัสตัวชี้เมาส์	สัมผัสสีน้ำเงิน	เท็จ
			ไม่ได้สัมผัสตัวชี้เมาส์	สัมผัสสีส้ม	เท็จ
			ไม่ได้สัมผัสตัวชี้เมาส์	สัมผัสสีชมพู	เท็จ
	หรือ		count=2	มีการคลิกเมาส์	จริง
			count=1	ไม่ได้คลิกเมาส์	จริง
			count=5	มีการคลิกเมาส์	จริง
			count=4	ไม่ได้คลิกเมาส์	เท็จ
	นิเสธ ไม่ใช่		สีเขียวสัมผัสสีส้ม		จริง
			สีเขียวสัมผัสสีชมพู		จริง

โปรแกรมตัดเกรด

สถาบันการสอนแห่งหนึ่งมีการสอบวัดระดับการใช้ภาษา โดยมีคะแนน 0 ถึง 100 คะแนน ผู้ที่ได้คะแนน 80 ขึ้นไป อยู่ในระดับ "ดีมาก" คะแนน 50-79 อยู่ในระดับ "พอใช้" และคะแนนต่ำกว่า 50 อยู่ในระดับ "ปรับปรุง" เขียนรหัสจำลองได้ดังนี้

1. รับค่าคะแนนสอบ
2. ถ้า คะแนนมากกว่าหรือเท่ากับ 80 ทำ
 - 2.1 แสดงข้อความ "ดีมาก"
 - ไม่เช่นนั้น
 - 2.2 ถ้า คะแนนมากกว่าหรือเท่ากับ 50 ทำ
 - 2.2.1 แสดงข้อความพอใช้
 - ไม่เช่นนั้น
 - 2.2.2 แสดงข้อความปรับปรุง

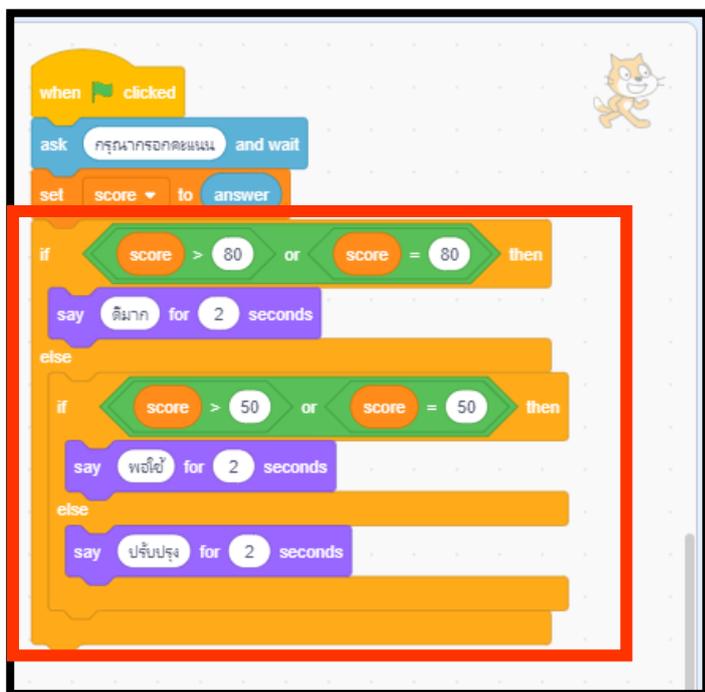
เขียนโปรแกรมได้ดังนี้



จากตัวอย่างที่ อธิบายได้ดังนี้

1. บรรทัดที่ 2 รับข้อมูลคะแนนจากผู้ใช้มาเก็บไว้ในตัวแปร answer
2. บรรทัดที่ 3 นำค่าคะแนนใน answer มาเก็บไว้ในตัวแปร score

3. บรรทัดที่ 4 ตรวจสอบเงื่อนไขว่า ถ้าคะแนน(score)มากกว่าหรือเท่ากับ 80 ให้แสดงข้อความ"ดีมาก"จากคำสั่งในบรรทัดที่ 5
4. บรรทัดที่ 7 ตรวจสอบเงื่อนไขว่าถ้าคะแนน(score)มากกว่าหรือเท่ากับ 50 ให้แสดงข้อความ"พอใช้" จากคำสั่งในบรรทัดที่ 8 แต่ถ้าไม่ใช่ให้แสดงข้อความ"ปรับปรุง"จากคำสั่งในบรรทัดที่ 10



ผลลัพธ์โปรแกรมได้ดังนี้

