

ใบความรู้ที่ 3 ตัวดำเนินการบูลีน

ตัวดำเนินการบูลีนเป็นตัวดำเนินการสำหรับตรวจสอบเงื่อนไข ตัวดำเนินการประเภทนี้จะใช้กระทำกับตัวถูกดำเนินการที่เป็นนิพจน์ทางตรรกศาสตร์หรือข้อมูลที่มีค่าความจริง (จริงหรือเท็จ) ตัวดำเนินการประเภทนี้ได้แก่ AND OR และ NOT โดยตัวดำเนินการ NOT จะกระทำกับตัวถูกตัวดำเนินการตัวเดียว ส่วนตัวดำเนินการตัวอื่นๆ จะกระทำกับตัวถูกตัวดำเนินการสองตัว

3.1 ตัวดำเนินการบูลีน

นักเรียนเคยเขียนโปรแกรมที่มีการคำนวณโดยใช้ตัวดำเนินการทางคณิตศาสตร์ + - * / และโปรแกรมที่มีการตรวจสอบเงื่อนไขที่มีการใช้ตัวดำเนินการเปรียบเทียบ > < = มาแล้ว ในส่วนนี้นักเรียนจะได้เรียนรู้เกี่ยวกับตัวดำเนินการบูลีน ที่ใช้เขียนโปรแกรมในกรณีที่มีเงื่อนไขมากกว่าหนึ่งเงื่อนไข ตัวดำเนินการบูลีนแสดง ดังตาราง 3

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน	สถานการณ์ที่เกิดกับตัวละคร/เวที		ผลการตรวจสอบ
			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)		สัมผัสสีส้ม		จริง
			สัมผัสสีชมพู		เท็จ



คำสั่งควบคุมของ Scratch

คำสั่งควบคุมเป็นคำสั่งกำหนดการทำงานของคำสั่งอื่น ขึ้นกับเงื่อนไขหรือข้อกำหนดประกอบด้วยคำสั่งเกี่ยวกับค่าความจริง เงื่อนไข ทำซ้ำ และการรอ ดังนี้

ค่าความจริง

1. รู้จักค่าความจริง

ค่าความจริงมีสองค่า คือ จริง (true) กับ เท็จ (false) เช่น “นกบินได้” มีค่าความจริงเป็น จริง (true) ส่วน “ควายบินได้” มีค่าความจริงเป็นเท็จ (false) ในทางคณิตศาสตร์มีเครื่องหมายเปรียบเทียบจำนวน = เท่ากับ, > มากกว่า, < น้อยกว่า เมื่อนำมาเขียนเป็นประพจน์ และพิจารณาค่าความจริง เช่น

5 = 9 ประพจน์นี้มีค่าความจริงเป็น เท็จ (false)

7 > 4 ประพจน์นี้มีค่าความจริงเป็น จริง (true)

1 < 2 ประพจน์นี้มีค่าความจริงเป็น จริง (true)

ซึ่งค่าความจริงนี้จะนำไปใช้ในการเขียนโปรแกรมคำสั่งควบคุมด้วย Scratch

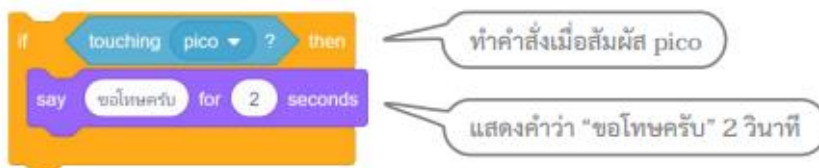
2. คำสั่งเกี่ยวกับค่าความจริงใน Scratch

คำสั่งเกี่ยวกับค่าความจริงใน Scratch อยู่ในหมวดคำสั่ง operators แบ่งออกเป็น 2 กลุ่ม คือการเปรียบเทียบ และตัวดำเนินการทางตรรกศาสตร์

เงื่อนไข

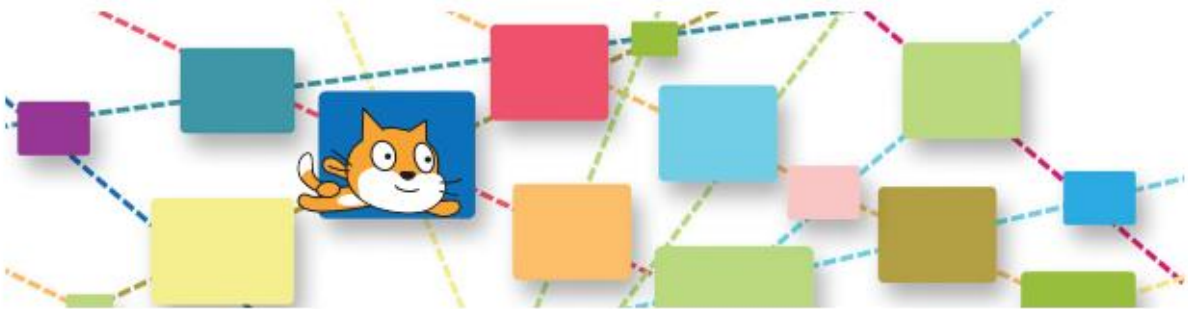
คำสั่งเงื่อนไข เป็นคำสั่งควบคุมการทำงานตามเงื่อนไข มี 2 คำสั่ง คือ if () then () และ if () then () else ()

1. คำสั่ง if () then () เป็นคำสั่งควบคุมให้ทำตามคำสั่งเมื่อเงื่อนไขเป็นจริงรูปแบบ if เงื่อนไข then คำสั่งที่กำหนดเริ่มด้วยการเพิ่มตัว sprite ชื่อ pico ลากบล็อกคำสั่ง if () then () แล้วกำหนดเงื่อนไขโดยลากบล็อกคำสั่ง touching กำหนดค่าให้เป็น pico บล็อกคำสั่ง say () for () secs พิมพ์ข้อความ “ขอโทษครับ” แล้ว run โดยการคลิกที่บล็อกคำสั่งที่กำหนด



ภาพที่ 1 ตัวอย่างการใช้ if () then ()

2. คำสั่ง if () then () else () เป็นคำสั่งควบคุมที่จะทำตามคำสั่งตามที่กำหนดเมื่อเงื่อนไขเป็นจริง หรือเท็จ



6+7 = ?

ถาม 6+7 แล้วรอคำตอบ

if answer = 13 then

say ถูกต้องนะครับ

else

say เป็นคำตอบที่ผิดครับ

ถ้าคำตอบเท่ากับ 13

แสดง "ถูกต้องนะครับ"

แสดง "เป็นคำตอบที่ผิดครับ"

ภาพที่ 2 แสดงตัวอย่างการใช้คำสั่ง if () then () else ()

ทำซ้ำ

คำสั่งทำซ้ำเป็นคำสั่งให้ทำงานวนรอบชุดคำสั่งตามจำนวนรอบ หรือเงื่อนไขที่กำหนด มี 3 คำสั่ง คือ repeat (), repeat until () และ forever

pen down

set pen size to 5

set pen color to 0

repeat 4

move 100 steps

turn 90 degrees

จรดปากกา (เริ่มวาด)

กำหนดขนาดปากกาเป็น 5

กำหนดสีปากกาเป็นสีแดง

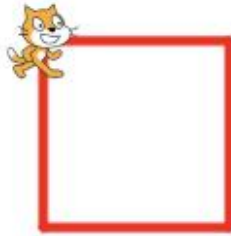
วนซ้ำ 4 รอบ

เคลื่อนที่ไปข้างหน้า 100

หมุนตามเข็มนาฬิกา 90 องศา

ภาพที่ 3 ตัวอย่างการใช้งานคำสั่ง repeat ()





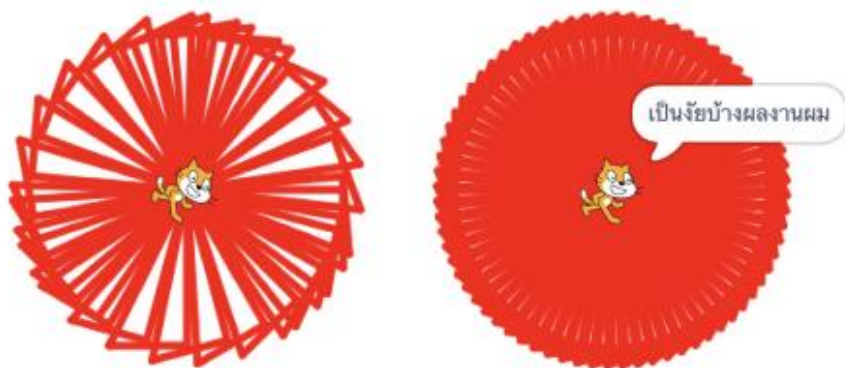
ภาพที่ 4 แสดงผลลัพธ์จากการใช้งานคำสั่ง repeat ()



ภาพที่ 5 ตัวอย่างการใช้งานคำสั่ง repeat () ซ้อนกัน



ภาพที่ 6 แสดงผลลัพธ์จากการใช้งานคำสั่ง repeat () ซ้อนกัน



ภาพที่ 7 ผลลัพธ์จากตัวอย่างการใช้คำสั่ง repeat until ()



ภาพที่ 8 ตัวอย่างการใช้คำสั่ง forever



ภาพที่ 9 ผลลัพธ์จากตัวอย่างการใช้คำสั่ง forever

รอ

คำสั่งรอ เป็นคำสั่งให้หยุดรอตามเวลาหรือเงื่อนไขที่กำหนด มี 2 คำสั่ง คือ wait () secs และ wait until ()

1. คำสั่ง wait () secs เป็นคำสั่งให้หยุดรอตามเวลาที่กำหนด เมื่อนำคำสั่ง say () และ wait () secs รวมกัน จะทำงานเช่นเดียวกับคำสั่ง say () for () secs



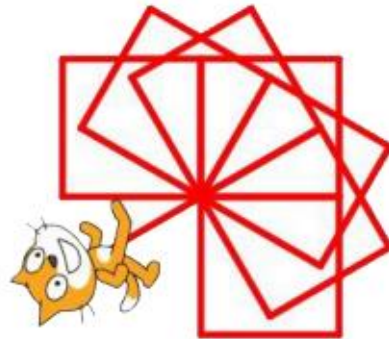


ภาพที่ 10 การใช้คำสั่ง say () และ wait () secs รวมกัน

คำสั่ง wait () secs สามารถนำไปใช้ร่วมกับคำสั่งอื่น เพื่อรอ หน่วงเวลา หรือเร่งเวลาการทำงานแต่ละขั้น สร้างคำสั่งดังภาพ แล้ว run โดยการคลิกที่บล็อกคำสั่ง



ภาพที่ 11 การใช้คำสั่ง wait () secs หน่วงเวลาในการวาด



ภาพที่ 12 ผลลัพธ์จากการใช้คำสั่ง wait () secs หน่วงเวลาในการวาด

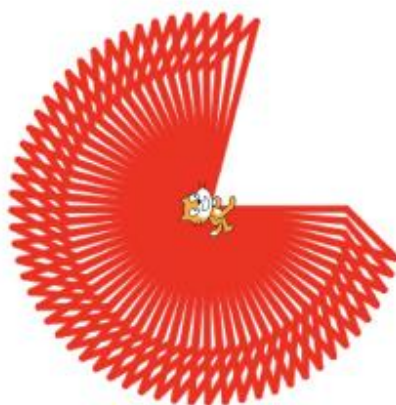
จากการใช้คำสั่ง wait () secs จะหน่วงเวลาในการวาด ทำให้เห็นขั้นตอนการวาดภาพที่เกิดขึ้น



2. คำสั่ง wait until () เป็นคำสั่งให้หยุดรอตามเงื่อนไขที่กำหนด เมื่อเงื่อนไขเป็นจริง จึงจะให้ทำคำสั่งถัดไป สร้างคำสั่งดังภาพ แล้ว run โดยการคลิกที่บล็อกคำสั่ง



ภาพที่ 13 ตัวอย่างการใช้คำสั่ง wait until ()



ภาพที่ 14 ผลลัพธ์จากตัวอย่างการใช้คำสั่ง wait until ()

Scratch จะหยุดรอจนกว่า จะกดปุ่ม space bar ที่คีย์บอร์ดจึงจะทำการวาดภาพต่อ

