

การพัฒนาโปรแกรม LOGO



หน่วยการเรียนรู้ที่ 7 ตัวแปรกับภาษาโลโก้

สาระการเรียนรู้

1. การวาดภาพด้วยคำสั่งซ้ำ
2. การสร้างภาพกราฟิกจากตัวแปร
3. การส่งค่าตัวเลขให้ตัวแปรในกระบวนการทำงาน
4. การเขียนข้อความบนกระดานทำงาน
5. การหยุดการทำงานของกระบวนการความ

ผลการเรียนรู้ที่คาดหวัง

สามารถกำหนดตัวแปรเพื่อการสร้างชิ้นงานด้วยภาษาโลโก้ได้



ใบความรู้ที่ 7 เรื่อง ตัวแปรกับภาษาโลโก้

1. การวาดภาพด้วยคำสั่งซ้ำ

1.1 สร้างด้วยกระบวนงานทำซ้ำ

สามารถนำกระบวนงานต่าง ๆ มาสร้างภาพด้วยคำสั่งทำซ้ำ เพื่อให้เกิดภาพกราฟิก ได้เช่น การใช้คำสั่งสร้างภาพ สี่เหลี่ยม (SQUARE) มาสร้างกระบวนงานใหม่ที่ชื่อว่า PATTERN เป็นคำสั่งที่สั่งให้วาดภาพสี่เหลี่ยมซ้ำ ๆ กันได้ ดังตัวอย่างที่ 1

ตัวอย่างที่

To square

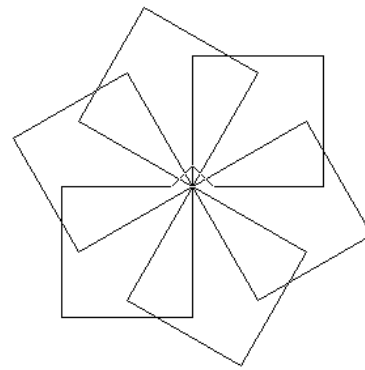
Repeat 4 [FD 100 RT 90]

End

To Pattern

Repeat 6 [Square RT 60]

End

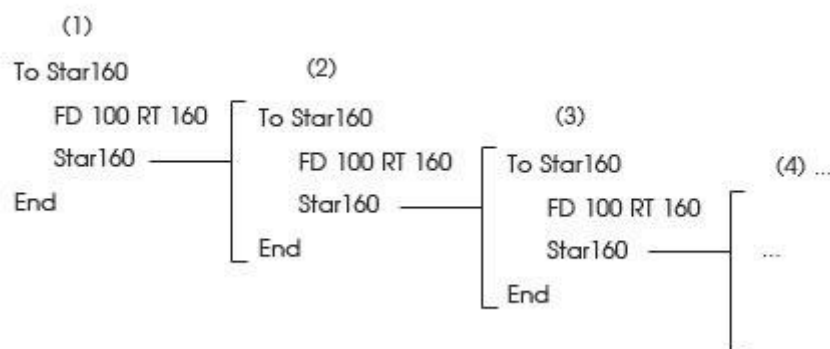


รูปที่ แสดงภาพที่เกิดจากคำสั่งทำซ้ำ

1.2 สร้างกระบวนงานซ้อนกระบวนงาน

การเรียกซ้ำ คือ การเรียกกระบวนความของตัวเองมาทำงาน เพื่อจะเข้าใจง่ายขึ้นลองพิจารณาการทำงานของกระบวนความ Star160 ซึ่งเป็นกระบวนความที่มีการเรียกซ้ำ ตัวอย่างที่ 2

ตัวอย่างที่ 1



ตัวอย่างที่ 2

ขั้นตอนที่ 1. สร้างกระบวนความชื่อ Triangle

To Triangle

FD 30 RT 120 FD 30 RT 120 FD 30

End

ขั้นตอนที่ 2. สร้างกระบวนความชื่อ Flower

To Flower

Triangle

RT 60

Flower

End

จะเห็นว่าเมื่อมีการสั่งให้เต่าทำงานตามกระบวนความนี้ เต่าจะเริ่มโดยทำคำสั่ง FD 100 RT 160 เมื่อมาพบคำสั่ง Star160 ซึ่งก็คือการเรียกใช้กระบวนความ Star160 เต่าจะไปทำกระบวนความใหม่อีกคือ FD 100 RT 160 และ Star160 อีก **เช่นนี้เรื่อยไปโดยไม่มีหยุด** ถ้าต้องการให้หยุดทำได้นี้ให้คลิกปุ่ม Halt ในหน้าต่างบังคับการ กระบวนงาน Flower ก็เช่นเดียวกัน

ตัวอย่างที่ 3

ขั้นตอนที่ 1

To one

Repeat 3 [FD 30 RT 120]

END

ขั้นตอนที่ 2

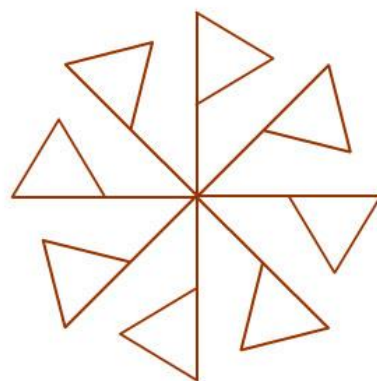
To two

FD 30

Repeat 3 [FD 30 RT 120]

BK 30

END



2. การสร้างภาพกราฟิกจากตัวแปร

คำสั่งภาษาโลโก้ ส่วนใหญ่จะเขียนคำสั่งและตามด้วยจำนวน เช่น FD 20 จำนวน 20 คือค่าที่บอกว่าจะให้เต่าเดินหน้าไปกี่หน่วย ข้อมูลจำนวนเหล่านี้จะเรียกว่า "ตัวแปร" (variable) ซึ่งจะทำให้ง่ายในการเปลี่ยนค่าเหล่านี้ เช่นหากเราทำกระบวนการสร้างสามเหลี่ยม แล้วให้สามเหลี่ยมนี้มีความยาวด้านละ 30 หน่วย แต่หากในภายหลังเราอยากเปลี่ยนเป็น 50 หน่วย เราก็ต้องเสียเวลาไปนั่งแก้ไข หากเรากำหนดเป็นตัวแปรแล้วให้พิมพ์ว่าจะต้องการความยาวเท่าไร เราก็ไม่ต้องไปเสียเวลาไปนั่งแก้

รูปแบบ

การกำหนดกระบวนการที่มีกำหนดตัวแปร

To ชื่อกระบวนการ : ตัวแปร : ตัวแปร : ...

หากเราต้องการจะใช้ตัวแปรให้พิมพ์ :ตัวแปร เช่น FD :Unit หมายถึง ให้เดินหน้าไปเท่ากับ ตัวแปร Unit สิ่งที่สำคัญในการเรียกตัวแปรคือเครื่องหมาย " : " หากไม่มีเครื่องหมายนี้ก็จะถือว่าเป็นชื่อกระบวนการ

ตัวอย่างที่ 1 การประกาศใช้งานตัวแปร

```
To sqr :size
Repeat 4 [ FD :size RT 90 ]
End
```

คำอธิบาย หากเราเรียก sqr 50 เต่าจะสร้างรูปสี่เหลี่ยมจัตุรัสขนาด 50 หน่วย หากเรียก sqr 60 เต่าก็จะสร้างรูปสี่เหลี่ยมขนาด 60 แต่ต่างจาก

```
To sqr
Repeat 4 [FD 50 RT 90]
End
```

ซึ่งไม่ว่าจะเรียกกี่ครั้งก็จะสร้างได้แค่สี่เหลี่ยมที่มีขนาด 50 แล้วถ้าเปลี่ยนมาใช้ตัวแปรใน กระบวนการ ความ sqr และกระบวนการ sqr_5

```
To sqr :size
Repeat 4 [FD :size RT 90]
End
```

```
To sqr_5
sqr 10
sqr 20
sqr 30
sqr 40
sqr 50
End
```

จะเห็นได้ว่าเราสามารถใส่กระบวนความเรียกกระบวนความอื่นได้ แม้กระทั่ง

ตัวอย่างที่ 2 วิธีการสร้างภาพกราฟิกรูปเหลี่ยม ด้วยการใส่คำสั่งกำหนดค่าให้กับตัวแปร

ขั้นตอนที่ 1 กำหนดกระบวนงานชื่อว่า polygon

ขั้นตอนที่ 2 กำหนดตัวแปรที่ชื่อว่า Sides (ค่าจำนวนด้านหรือเหลี่ยมที่ต้องการ) และ Edge (ค่าความยาวด้าน)

ขั้นตอนที่ 3 เขียนกระบวนงานที่ต้องการดังนี้

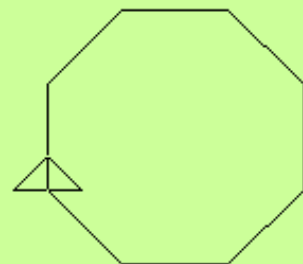
```
TO Polygon :Sides :Edge
Repeat :Sides [FD :Edge RT 360/:Sides]
End
```

หมายเหตุ คำสั่งในบรรทัดที่สอง ค่าของ 360/:Sides หมายถึงการหาค่ามุม 360 องศาด้วยจำนวนด้านของภาพที่ต้องการ แทนค่าการคำนวณมุมที่เท่าไรก็จะต้องเลี้ยวแต่ละด้าน

พิมพ์คำสั่งเรียกใช้กระบวนงาน ในช่องป้อนคำสั่ง (Input Box)

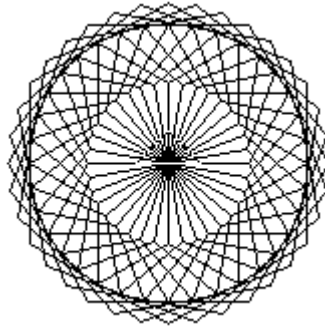
Polygon 8 50

คือ การสร้างภาพรูป 8 เหลี่ยม แต่ละด้านยาว 50 หน่วย



ทดลองใช้คำสั่งทำซ้ำอย่างต่อเนื่องจากด้านบนโดยป้อนคำสั่งที่ช่องป้อนคำสั่ง (Input Box) ด้วยคำสั่งต่อไปนี้

```
Repeat 36 [Polygon 6 40 RT 10]
```



3. การส่งค่าตัวเลขให้ตัวแปรในกระบวนการวน

การสั่งให้เต่าโลโก้วาดภาพ การกำหนดคำสั่งที่ทำให้เกิดความคล่องตัวมากขึ้น สามารถทำได้โดยการส่งค่าตัวเลขให้ไปใช้กับตัวแปร ในกระบวนการวนนั้นโดยการ กำหนดชื่อตัวแปร และเครื่องหมาย : ข้างหลังชื่อกระบวนการวนในบรรทัดแรกของกระบวนการวน เช่น

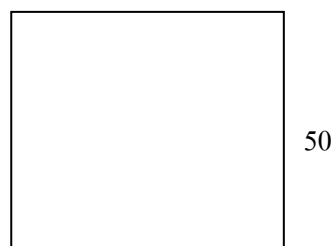
ตัวอย่าง

```
To Square : Sizes
```

```
Repeat 4 [FD : Sizes RT 90]
```

```
End
```

เมื่อต้องการเรียกใช้กระบวนการวนโดยไม่ต้องเข้าไปเปลี่ยนแปลงข้อมูลตัวเลขของความยาวรูปสี่เหลี่ยม สามารถทำได้โดยพิมพ์ชื่อ กระบวนการวน และตามด้วยจำนวนตัวเลขที่ต้องการ เช่น Square 50 ก็จะได้รูปสี่เหลี่ยมที่มีความยาวด้านละ 50 หน่วย



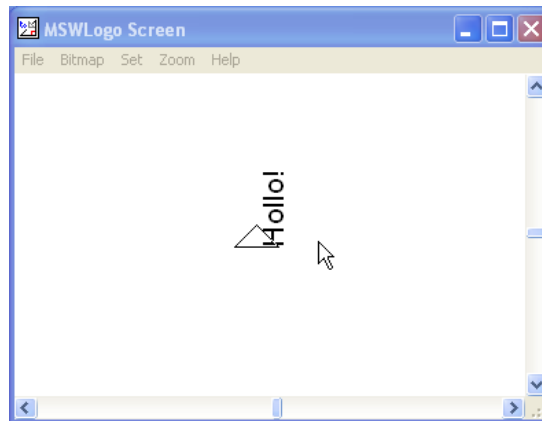
50

4. การเขียนข้อความบนกระดานทำงาน

ในบางครั้งภาพที่วาดอาจมีความจำเป็นในการแสดงภาพและเขียนข้อความบนกระดานทำงานพร้อมกัน การแสดงข้อความสามารถทำได้โดยใช้คำสั่ง Label ตามด้วยเครื่องหมายคำพูด ดังรูปแบบรูปแบบการใช้คำสั่ง

Label "ข้อความ"

ตัวอย่าง ลองพิมพ์คำสั่ง Label "Hollo!" ในช่องป้อนคำสั่ง จะได้ผลลัพธ์ดังรูป



5. การหยุดการทำงานของกระบวนการ

การหยุดการทำงานของเต้านั้น แต่เดิมเราทำโดยการคลิกที่ปุ่ม Halt หมายถึงการสั่งหยุดการทำงานของเต่า ไม่ใช่การทำงานเมื่อสิ้นสุดคำสั่งตามปกติ ซึ่งไม่เหมาะกับการนำไปใช้งานในกระบวนการทั้งหมด การจะทำให้หยุดโดยไม่กด Halt นั่นคือ การใส่เงื่อนไขการหยุดทำงาน คำสั่งที่ใช้คือ IF ซึ่งมีรูปแบบดังนี้

IF เงื่อนไข [Stop]

เช่น ถ้าต้องการให้กระบวนการหยุดทำงานเมื่อตัวแปร size มากกว่า 100 คำสั่งที่ต้องใช้คือ

IF :size>100 [stop]

หมายเหตุ การหยุดด้วยวิธีนี้ส่วนมากจะใช้ตัวแปรมาควบคุมเงื่อนไข กระบวนการนั้นๆ จึงมีตัวแปรรวมอยู่

ตัวอย่าง

ขั้นตอนที่ 1. สร้างกระบวนการความชื่อ s_sqr

```
To s_sqr :size
Repeat 4 [ FD :size RT 90 ]
End
```

ขั้นตอนที่ 2. สร้างกระบวนการความ grow1

```
To grow1 :size
s_sqr :size
grow1 :size+10
IF :size > 100 [Stop]
End
```

ขั้นตอนที่ 3. สร้างกระบวนการความ grow2

```
To grow2 :size
s_sqr :size
IF :size>100 [stop]
grow2 : size+10
End
```

ขั้นตอนที่ 4. สร้างกระบวนการความ grow3

```
To grow3 :size
IF :size>100 [stop]
s_sqr :size
grow3 :size+10
End
```

เมื่อพิมพ์เสร็จแล้วให้เต่าวาด grow1 10 แล้วสั่งให้เต่าวาด grow2 10 แล้วก็สั่ง grow3 10 เปรียบเทียบผลลัพธ์ทั้งสามคำสั่งแล้วบันทึกผล เปรียบเทียบคำสั่งที่แตกต่างกันในกระบวนการทั้งสาม

ทั้งสามกระบวนการนั้นแตกต่างกันที่ตำแหน่งของเงื่อนไข ซึ่งเป็นสิ่งสำคัญในการควบคุมการทำงานของกระบวนการ การใส่เงื่อนไขใน grow1 ไม่สามารถหยุดได้เนื่องจากเต่าไม่มีโอกาสที่จะไปทำคำสั่งเงื่อนไขหยุดการทำงาน (แบบนี้ใส่กับไม่ใส่ก็มีค่าเท่ากัน) มาดู grow2 จะหยุดเมื่อมีการวาดรูปที่ใหญ่กว่า 100 แล้ว เพราะจะวาดรูปก่อนที่จะไปตรวจสอบว่ามีขนาดใหญ่กว่า 100 หรือไม่ ส่วน grow3 จะตรวจก่อนเลยว่าใหญ่กว่า100 หรือไม่ แล้วจึงวาดรูป ดังนั้นจะหยุดรูปที่วาดจึงไม่มีรูปไหนที่ขนาดใหญ่กว่า 100 เลยหลัก

สำคัญที่สุดของกระบวนการเรียกซ้ำนี้คือ การตรวจสอบว่าจะหยุดทำงานเมื่อไร วิธีที่ดีที่สุดในการไล่กระบวนการความนี้คือ การไล่ดูคำสั่งที่ลบบรรทัดโดยสมมติว่าเราคือเต่า ซึ่งทำหน้าที่ทำตามคำสั่งที่ลบบรรทัด แล้วจะได้ผลลัพธ์อย่างไร

ตัวอย่าง วิธีทำรูปบ้าน

ขั้นตอนที่ 1 รูปบ้านจะประกอบไปด้วย สามเหลี่ยมและสี่เหลี่ยม ส่วนของสามเหลี่ยมจะใช้คำสั่ง

RT 90 Repeat 3 [FD 30 LT 120]

ขั้นตอนที่ 2 ส่วนของสี่เหลี่ยมจะใช้คำสั่ง

Repeat 4 [FD 30 RT 90]

เราจะวาดสี่เหลี่ยมข้างล่างก่อน จากนั้นให้เต่าขึ้นไปอยู่ด้านบนของสี่เหลี่ยม แล้ววาดสามเหลี่ยม แล้วให้เต่ากลับมาที่จุดเริ่มต้น กระบวนการความจะเป็น

To House

Repeat 4 [FD 30 RT 90]

FD 30

RT 90 Repeat 3 [FD 30 LT 120]

LT 90 BK 30

End



จะเห็นว่าหากคิดแบบง่ายๆ กระบวนการความก็จะไม่ซับซ้อนมาก และไม่เสียเวลาในการเขียน หากต้องการใช้ตัวแปรในการเขียนอาจจะเปลี่ยนเป็น ให้ตัวแปรนั้นแทนขนาดของแต่ละด้าน อาจจะเขียนได้ว่า

To House :size

Repeat 4 [FD :size RT 90]

FD :size

RT 90 Repeat 3 [FD :size LT 120]

LT 90 BK :size

End

ก็จะได้บ้านที่สามารถกำหนดขนาดของบ้านได้ หากเราต้องการสร้างบ้านที่มีขนาดใหญ่ขึ้นเรื่อยๆ แบบภาพนี้เราก็สามารถทำได้โดยอาจจะสร้างกระบวนการความขึ้นมาเพิ่ม คือ กระบวนการความเคลื่อนที่เต่าให้ไปอยู่ตำแหน่งต่างๆ และกระบวนการความสั่งรวมกระบวนการความให้เต่าเคลื่อนที่

```
To Move :size
```

```
PU RT 90
```

```
FD :size +10
```

```
LT 90 PD
```

```
End
```

กระบวนการรวม (หากเราต้องการบ้าน 4 หลัง)

```
To Village
```

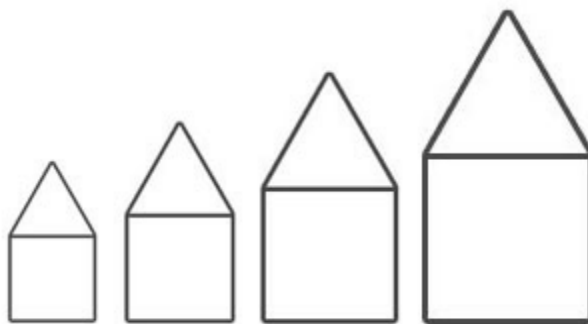
```
House 10 Move 10
```

```
House 20 Move 20
```

```
House 30 Move 30
```

```
House 40 Move 40
```

```
End
```



เราสามารถนำกระบวนการอื่นมาเรียกกระบวนการที่เราเคยเขียนไว้แล้ว จะทำให้สามารถเขียนกระบวนการที่ซับซ้อนได้ และเป็นการทำงานที่ซ้ำกันด้วยการใช้ตัวแปรมากกว่า 1 ตัวแปรในกระบวนการ เราสามารถใช้ตัวแปรได้มากกว่า 1 ตัวแปรในกระบวนการเช่น การสร้างรูปสี่เหลี่ยมผืนผ้า

```
To Rectangle :Length :Width
```

```
Repeat 2 [ FD :Length RT 90 FD :Width RT 90]
```

```
End
```

อธิบายกระบวนการคือ กำหนดให้ตัวแปร Length รับค่าความยาวของสี่เหลี่ยมและ Width รับความกว้างของสี่เหลี่ยม การเรียกใช้เช่น Rectangle 10 50 เค้าก็จะสร้างรูปสี่เหลี่ยมผืนผ้าที่มีขนาดความยาว 10 ความกว้าง 50 (ความยาวในที่นี้คือความยาวตามแนวดิ่ง ความกว้างคือ ความยาวตามแนวระดับ) เป็นต้น

6. การสร้างภาพเคลื่อนไหว

ภาษาโลโก้มีคุณสมบัติเด่นอย่างหนึ่ง ก็คือสามารถเขียนโปรแกรมเรียกตัวเองที่เรียกว่า Recursion ได้ ด้วยคุณสมบัตินี้ ทำให้เราสามารถนำมาเขียนโปรแกรมสร้างภาพเคลื่อนไหว ได้อย่างง่าย ๆ ตัวอย่างดังโปรแกรมหมายเลข 1

โปรแกรมหมายเลข 1

```

Editor
File Edit Search Set Test! Help
TO TRIANGLE
  RT 30 REPEAT 3[FD 50 RT 120] LT 30
END

TO MOVE :X
  PU FD :X PD
END

TO TRI_GO
  HT PD SETPC 4
  TRIANGLE
  SETPC 7 WAIT 20
  TRIANGLE
  MOVE 5
  TRI_GO
END

```

จะสังเกตเห็นว่าในโปรแกรมน้อยชื่อ TRI_GO จะมีคำสั่งในบรรทัดสุดท้ายเรียกใช้โปรแกรมน้อยชื่อ TRI_GO ซึ่งก็คือเรียกใช้ตัวเอง (วิธีการนี้มีชื่อเรียกว่า Recursion) ลองดูอีกสักโปรแกรมหนึ่ง คราวนี้จะป็นรูปปลา 1 ตัว สีแดง เคลื่อนที่ไปทางซ้ายมือ ชุดคำสั่งทั้งหมดดังโปรแกรมหมายเลข 2

โปรแกรมหมายเลข 2

```

Editor
File Edit Search Set Test! Help
TO SQUARE :SIZE
  REPEAT 4[FD :SIZE RT 90]
END

TO MOVE :X
  PU LT 90 FD :X RT 90 PD
END

TO FISH :SIZE
  RT 45 SQUARE :SIZE LT 45
  PU RT 90 FD (:SIZE*SQRT 2) LT 90
  PD RT 45 SQUARE :SIZE/3 LT 45
  PU RT 90 BK (:SIZE*SQRT 2) LT 90 PD
END

TO FISH_GO :SIZE :X :T
  HT PD SETPC 4
  FISH :SIZE
  SETPC 7 WAIT :T
  FISH :SIZE
  MOVE :X
  FISH_GO :SIZE :X :T
END

```