

# ภาษาโลโก้และการสร้างชิ้นงาน

MSW Logo

# ความเป็นมาของภาษาโลโก

เริ่มขึ้นในปี พ.ศ.2513 เมื่อกลุ่มนักวิจัยของสถาบันเทคโนโลยีแมสซาชูเซตส์ นำโดย เซย์มัว พาเพิร์ต ได้ทำการออกแบบและสร้างหุ่นยนต์ คล้ายกับของ เกย์ วอลเทอร์ เพื่อให้เด็ก ๆ สามารถเขียนโปรแกรมคำสั่งที่ใช้ภาษาที่เข้าใจง่าย สั่งให้โปรแกรมทำงานตามที่ต้องการ พวกเขาจึงทำการพัฒนาภาษาคอมพิวเตอร์รุ่นใหม่ เรียกว่า ภาษา “โลโก”

เป็นภาษาที่ง่ายสำหรับเด็ก ช่วยให้เด็กสามารถ  
เขียนคำสั่งให้หุ่นยนต์เต่า (Logo) เคลื่อนที่ไป  
มาและเปลี่ยนทิศทางตามที่ต้องการ ภาษาโล  
โกจึงเป็นทางเลือกใหม่ สำหรับเด็กในการฝึก  
ทักษะทางภาษาคอมพิวเตอร์ และสามารถสร้าง  
งานจากจินตนาการ โดยอาศัยความเข้าใจ  
พื้นฐานของวิชาคณิตศาสตร์ที่ใช้ใน  
ชีวิตประจำวันได้



ในปัจจุบันเครื่องคอมพิวเตอร์และอุปกรณ์ต่าง ๆ มี  
ราคาถูกลง จึงได้มีการพัฒนาโปรแกรมสำเร็จรูปให้  
สามารถจำลองหุ่นยนต์เต่าอิเล็กทรอนิกส์ เป็น  
ภาพกราฟิกเต่า เคลื่อนที่ไปมาบนจอภาพ  
คอมพิวเตอร์ ต่อมา มีการเปลี่ยนแปลงรูปแบบของ  
ภาพกราฟิกเต่า มาเป็นภาพสัญลักษณ์สามเหลี่ยม

คำว่า “โลโก” หมายถึงหุ่นยนต์เต่า สามารถเคลื่อนที่ไปในทิศทางต่าง ๆ ได้โดยการควบคุมจากเครื่องคอมพิวเตอร์ ในขณะที่เคลื่อนที่จะเขียนรอยทางเดินไปตามทิศทางนั้น ๆ ด้วยผู้เขียนโปรแกรมสามารถใช้คำสั่งภาษาอังกฤษที่เข้าใจได้ง่าย ๆ

ภาษาโลโกมีจุดเริ่มต้นการพัฒนาโปรแกรมจาก ไบรอัน ฮาร์เวย์ และคณะทำงานที่มหาวิทยาลัยแคลิฟอร์เนีย—เบิร์กลีย์ ซึ่งเรียกโปรแกรมนี้ว่า "โลโกเบิร์กลีย์" หลังจากนั้นได้มีการพัฒนาอย่างต่อเนื่อง ให้สามารถใช้งานภายใต้ระบบปฏิบัติการที่แตกต่างกัน จากระบบยูนิกซ์ (**UNIX**) เป็นระบบดอส (**DOS**) จนถึงปัจจุบัน ยอร์จ มิลส์ ได้พัฒนาให้สามารถทำงานภายใต้ระบบปฏิบัติการวินโดวส์ได้ เราเรียกโปรแกรมนี้ว่า "เอ็มเอสดับบลิวโลโก" (**MSWLogo**)



**MSWLogo** เป็นโปรแกรมรุ่นที่มีประสิทธิภาพสูงสุดจากโปรแกรมโลโก้ที่มีใช้ทั้งหมดในขณะนี้ เหมาะสำหรับใช้ในการเรียนระดับประถมศึกษาและมัธยมศึกษาตอนต้น และได้มีการเรียนการสอนเกี่ยวกับโปรแกรมนี้อย่างแพร่หลายในประเทศสหรัฐอเมริกา ญี่ปุ่น ออสเตรเลีย และประเทศอื่น ๆ ทั่วโลก การใช้งานสะดวก ไม่ยุ่งยาก เหมือนโปรแกรมอื่นๆ

**MSWLogo** โปรแกรมนี้จะกระตุ้นให้นักเรียนมีความกล้าใน  
การสร้างสรรคงานกราฟิกรูปแบบต่าง ๆ จากจินตนาการของ  
ตนเองได้ง่าย ช่วยให้มึพื้นฐานและความถนัดเบื้องต้นในการ  
เขียนโปรแกรม นอกจากนี้ความแพร่หลายของโปรแกรม

**MSWLogo** ที่นำไปสอนในโรงเรียนต่าง ๆ ทำให้มีการสนับสนุน  
ทางด้านวิชาการและตัวอย่างโปรแกรมที่น่าสนใจเป็นจำนวน  
มาก โดยสามารถสืบค้นจากอินเทอร์เน็ตได้



เมื่อเรารู้จักกับประวัติความเป็นมาของโลโก้กันแล้ว คราวนี้ มาดูหน้าตาของโปรแกรมกันดีกว่า



# คำสั่งพื้นฐานในการ เขียนโปรแกรมโลโก้

เราสามารถเขียนคำสั่งย่อ (Shortcut Commands) ได้ดังนี้

Home	ไม่มีคำย่อ	กลับไปยังจุดเริ่มต้น
CS	หมายถึง	ClearScreen (เคลียร์หน้าจอ)
HT	หมายถึง	Hide Turtle (การซ่อนเต่า)
ST	หมายถึง	Show Turtle (การแสดงตัวเต่า)
Pu	หมายถึง	Penup (ยกเต่า)
Pd	หมายถึง	Pendown (วางเต่า)
Fill	หมายถึง	เทสีภายใน
Pe	=	เปลี่ยนจากปากกาเป็นยางลบ
PPt	=	PenPaint คือ เปลี่ยนจากยางลบเป็นปากกา



# คำสั่งพื้นฐานในการ เขียนโปรแกรมโลโก้

เราสามารถเขียนคำสั่งย่อ (Shortcut Commands) ได้ดังนี้

FD = Forward (เดินหน้า)

BK = Back (ถอยหลัง)

RT = Right (เลี้ยวขวา)

Lt = Left (เลี้ยวซ้าย)

Ellipse = วงรี เช่น ellipse 100 200

Circle = วงกลม เช่น circle 60

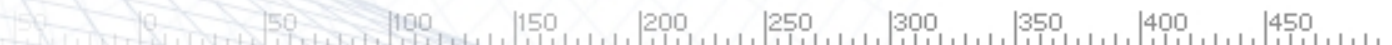
Arc = รัศมี เช่น arc 180 100

Setsc = setscreencolor (ใช้กำหนดสีฉากหลังของจอภาพ)

Setpc = setpencolor (ใช้ในการเปลี่ยนสีของปากกา)

Setfc = setfillcolor (กำหนดสีของพื้นที่)

Repeat 4 [ fd 100 rt 90] คือ สร้างรูปสี่เหลี่ยม



# เกี่ยวกับสี

การผสมสี 3 สี คือ แดง เขียว ฟ้า แทนค่าตั้งแต่ 0 ถึง 255

เช่น `setsc [R G B] = setsc [50 200 255]`

หรือ สีที่ผสมแล้ว ใช้ `setsc ...` เช่น `setsc 3`

0 = สีดำ

1 = สีน้ำเงิน

2 = สีเขียว

4 = แดง

5 = ม่วง

6 = เหลือง

7 = ขาว

8 = น้ำตาล

9 = น้ำตาลอ่อน

เป็นต้นๆ





# การลากเส้นสี

Cs

SetPc [0 0 255] ได้สี.....

Fd 100

SetPc [255 0 0] ได้สี.....

Fd 100

SetPc [0 255 0] ได้สี.....

Fd 100

SetPc [0 0 0] ได้สี.....

Fd 100

## กระบวนการความ

คือ คำสั่งต่าง ๆ ที่ใช้สั่งเต่าให้ทำงาน สามารถนำมารวมกันสร้างเป็นชุดคำสั่งเพื่อทำงานตามต้องการ และตั้งชื่อคำสั่งไว้เพื่อความสะดวกในการเรียกใช้

รูปแบบของกระบวนการความ

To ชื่อกระบวนการความ

คำสั่ง 1

คำสั่ง 2

.

.

END

ตัวอย่าง เช่น

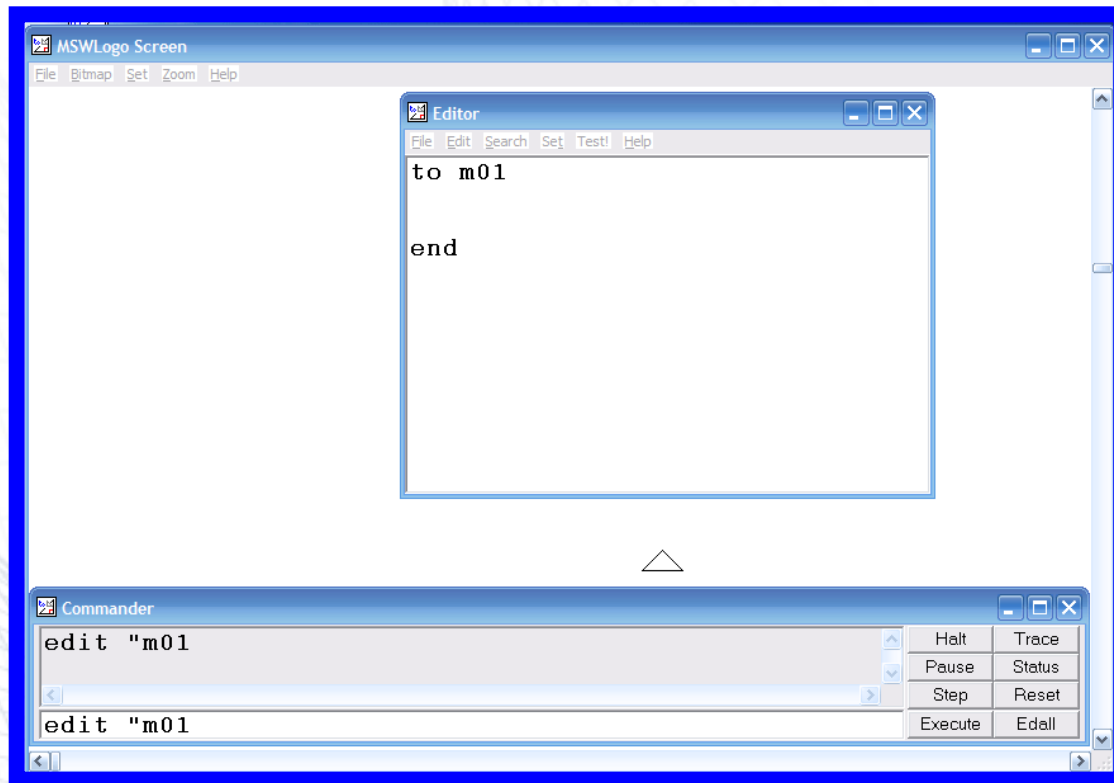
To m01

Repeat 4 [ fd 100 rt 90]

End

# การเรียกใช้กระบวนการความ

โดยพิมพ์ **Edit** “.....” ในส่วนที่ไว้ป้อนคำสั่ง

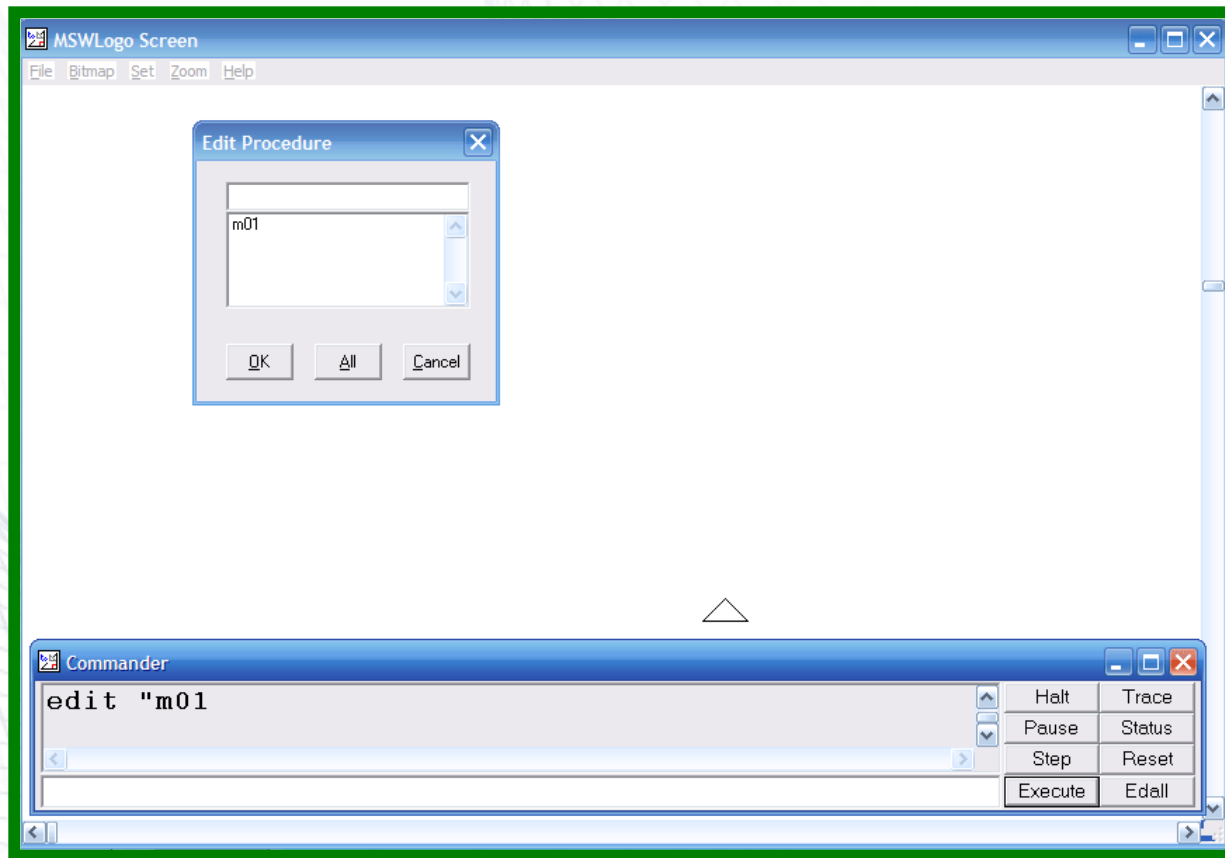


50 100 150 200 250 300 350 400 450



# การแก้ไขกระบวนการความ

เลือกเมนู file แล้วเลือก edit



# 1. การบันทึกกระบวนการความที่หน้าต่าง Editor

- เลือกเมนู file
- เลือก Save and Exit

# 2. การบันทึกกระบวนการความที่หน้าต่างโปรแกรม

msw logo

- เลือกเมนู file
- เลือก Save as ระบุที่อยู่



# กระบวนการความสร้างรูปสี่เหลี่ยม จำนวน 4 รูป

To m02

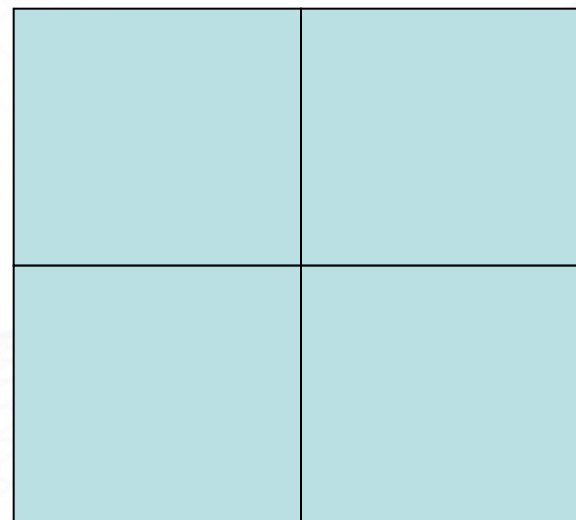
m01 Lt 90

M01 Lt 90

M01 Lt 90

M01

End





# กระบวนการความสร้างรูปสี่เหลี่ยม ซ้อนกัน 2 รูป

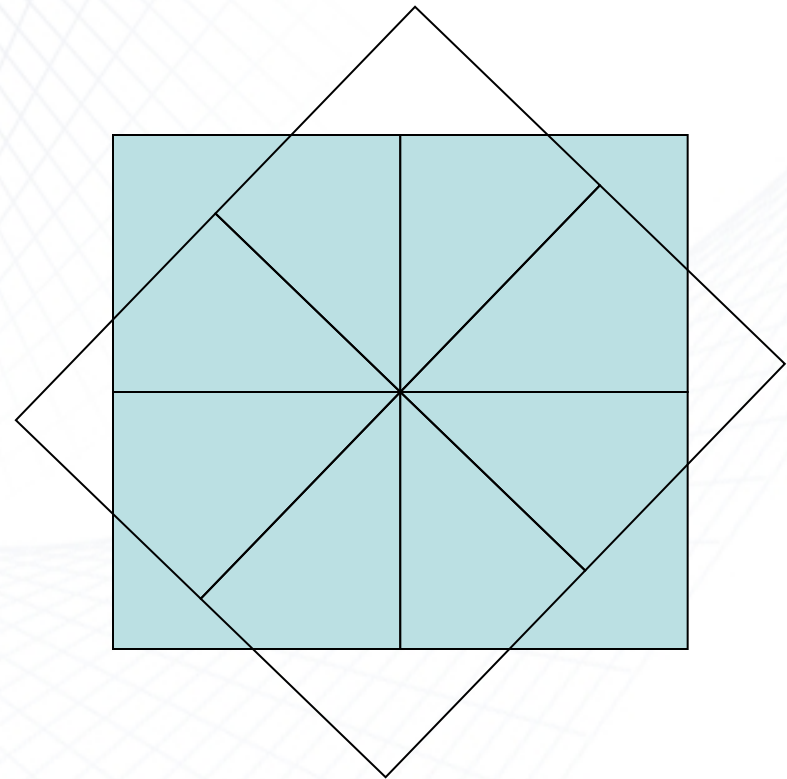
To m03

m02

rt 45

m02

End



# Star

```
Repeat 5 [fd 100 rt 145]
```

สร้างกระบวนความ Star

To star

```
Repeat 5 [fd 100 rt 144]
```

End



# 1. การบันทึกงานกราฟิกที่สร้างเสร็จแล้ว

- เลือกเมนู Bitmap
- เลือก Save as แล้วระบุที่อยู่

# 2. การบันทึกกระบวนการความเก็บไว้ใช้

- เลือกเมนู file
- เลือก Save as ระบุที่อยู่

## 1. การเปิดภาพกราฟิกขึ้นมาแก้ไข

- เลือกเมนู Bitmap
- เลือก Load แล้วเลือกไฟล์รูป (GiF , bmp)

## 2. การเปิดกระบวนการความที่เก็บไว้มาใช้งาน

- เลือกเมนู file
- เลือก Load แล้วเลือกกระบวนการที่จะใช้(.Lgo)



## คำสั่งพิมพ์ข้อความ

Label “Navamin

LABEL [THIS IS TURTLE]

## คำสั่งกำหนดของเต่าบนจอภาพ

Setpos [x y]

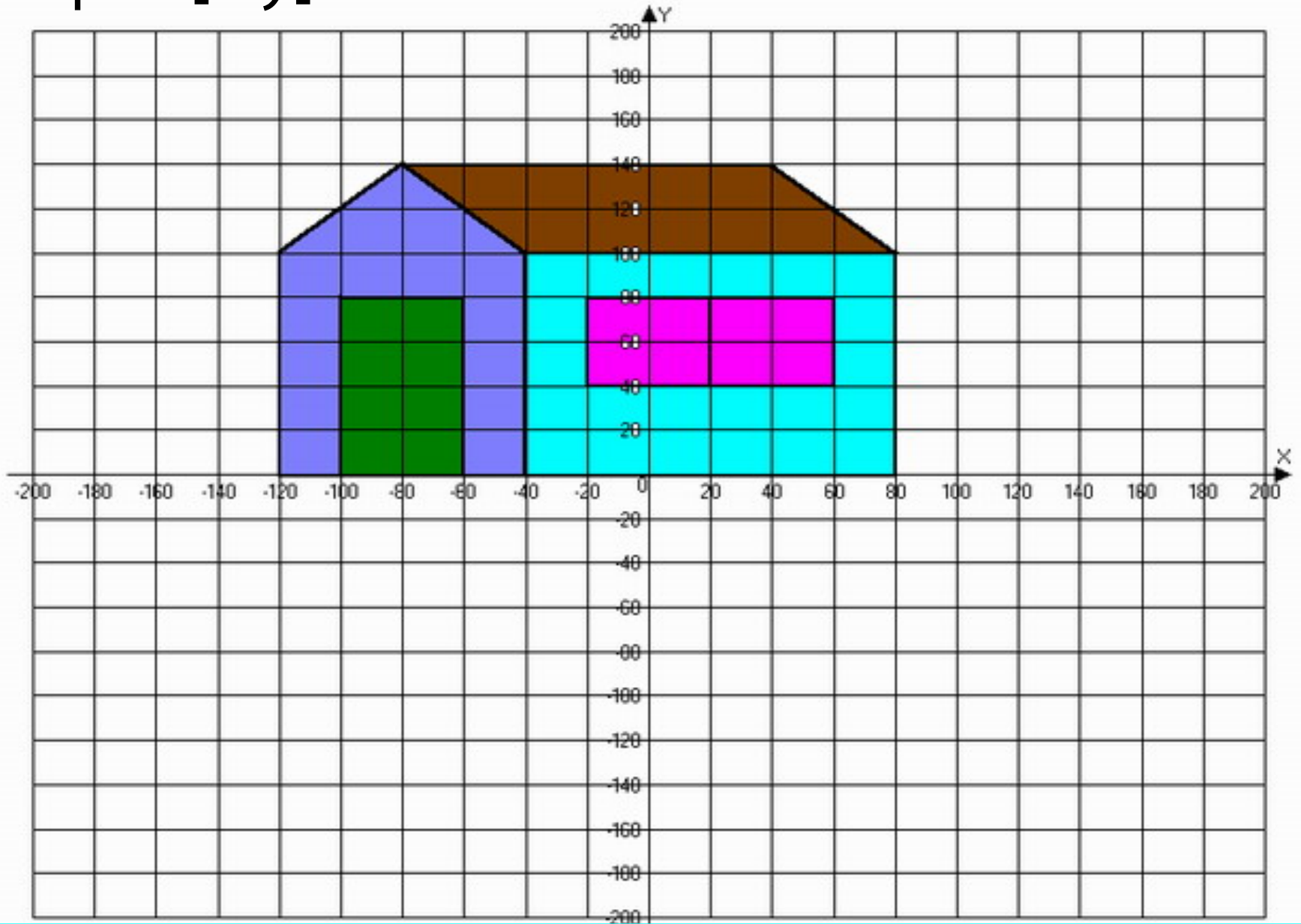
โดย ค่า X เป็นตำแหน่งในแนวนอน

ค่า Y เป็นตำแหน่งในแนวตั้ง

ตำแหน่งเริ่มต้นของเต่าคือ HOME ค่า  $X = 0$  และค่า  $Y = 0$

เช่น Setpos [-100 50]

# Setpos [x y]



# รู้จักกับตัวแปรในภาษาโลโก

ตัวแปร เป็นตัวอักษรหรือตัวอักษรปนตัวเลข ที่กำหนดขึ้นเพื่อแทนค่าใดค่าหนึ่ง อาจเป็นค่าตัวเลข หรือข้อความ หรือรายการ

ตัวแปร หมายถึง ข้อมูลที่มีการกำหนดค่าและเปลี่ยนแปลงค่าด้วยคำสั่งภายในโปรแกรม ประกอบด้วย ชื่อตัวแปร ค่าตัวแปร และตำแหน่งที่เก็บค่าตัวแปรในหน่วยความจำ

ตัวแปรมีสองชนิด คือ ชนิดตัวเลข (numeric) คือ ตัวแปรที่มีค่าเป็นตัวเลขใช้คำนวณได้ และชนิดสายอักขระ (string) คือ ตัวแปรที่เป็นข้อความ

## การกำหนดตัวแปรของภาษาโลโกมี 2 วิธี

- วิธีแรกเป็นการใช้คำสั่ง Make หรือ ตัวแปรสาธารณะ (global variable)
- วิธีใช้กระบวนการความที่มีการกำหนดให้ใส่ค่าตัวแปร หรือ ตัวแปรเฉพาะงาน (local variable)



## รูปแบบกระบวนการความถี่มีการกำหนดค่าตัวแปร

คือ

To ชื่อกระบวนการ :ตัวแปร1 :ตัวแปร2 :...

เช่น To m01 :times :size

วิธีป้อน Edit “m01:times:size

# ข้อควรระวัง

1. ต้องเว้นวรรคระหว่างชื่อกระบวนการความกับตัวแปร และตัวแปรกับตัวแปร
2. ตัวแปรทุกตัวต้องนำหน้าด้วยเครื่องหมาย : และ ไม่มีช่องว่างระหว่าง : กับชื่อตัวแปร
3. การอ้างถึงตัวแปรในกระบวนการความต้องมีเครื่องหมาย : กำกับหน้าชื่อตัวแปรทุกครั้ง

# เปรียบเทียบกระบวนการความที่กำหนดค่าคงที่ กับ กำหนดตัวแปร

กระบวนการความ SQUARE	กระบวนการความ S_SQUARE
<pre>TO SQUARE   FORWARD 50   RIGHT 90   FORWARD 50   RIGHT 90   FORWARD 50   RIGHT 90   FORWARD 50   RIGHT 90 END</pre>	<pre>TO S_SQUARE :SIZE   FORWARD :SIZE   RIGHT 90   FORWARD :SIZE   RIGHT 90   FORWARD :SIZE   RIGHT 90   FORWARD :SIZE   RIGHT 90 END</pre>
<pre>หรือ TO SQUARE   REPEAT 4 [FORWARD 50 RIGHT 90] END</pre>	<pre>หรือ TO S_SQUARE :SIZE   REPEAT 4 [FORWARD :SIZE RIGHT 90] END</pre>

# การสร้างกระบวนความวงกลม

Edit “N\_circle:times:size

```
To N_circle :times :size
```

```
  repeat :times[pu fd 20 pd circle :size]
```

```
End
```

```
to cir_cir :x
```

```
  repeat 10[pu fd :x pd circle :x rt 36]
```

```
end
```





# การสร้างกระบวนการความ วงรี

```
to wongree :x :y
```

```
repeat 10[ellipse :x :y rt 18]
```

```
end
```

```
to petal :size
  pu fd :size pd
  arc -90 :size
  pu bk :size rt 90
  fd :size rt 90 pd
  arc -90 :size
  pu rt 90 fd :size rt 90 pd
end
```

```
to flower :size  
repeat 6[petal :size rt 60]  
end
```

```
to stem :size  
fd :size petal :size  
fd :size * 2  
end
```

```
to tree :size
```

```
  stem :size
```

```
  flower :size
```

```
  bk :size * 3
```

```
end
```



**to myhome**

Pu fd 100 rt 90 pd

Repeat 3 [ fd 100 lt 120 ]

Rt 90

Repeat 5 [ fd 100 lt 90 ]

Fd 30 lt 90

Repeat 2 [ fd 60 rt 90 fd 40 rt 90 ]

Pu rt 90 fd 10 lt 90 fd 10

Setfc [ 220 210 60 ] fill

Fd 60

Setfc [ 70 160 210 ] fill

Fd 70

Setfc [ 30 250 220 ] fill

Home

Lt 90 fd 150 rt 90 fd 100 pd

**End**

**to mytree**

Lt 120

Repeat 7 [ repeat 120 [ fd 1 rt 1 ] lt 70]

Rt 120

Pu fd 60 pd

Lt 180 fd 160 rt 100 fd 55 rt 100 fd 160

Pu fd 20 setfc [ 20 200 60 ] fill

Rt 170

Pu fd 60

Setfc [ 200 140 140 ] fill

**End**

## คำสั่งวาดรูปปลา

```
to square :size  
Repeat 4 [ fd :size rt 90 ]  
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 move :x  
Pu lt 90 fd :x rt 90 pd  
End
```

```
to fishgo :size :x :y  
Ht pd setpc 4  
Fish :size  
Setpc 7 wait :y  
Fish :size  
Move :x  
Fishgo :size :x :y  
End
```

# โครงสร้างข้อมูลและคำสั่งจัดการข้อมูล

## 1. ตัวเลข คำ และรายการ

บทบาทของคอมพิวเตอร์ คือ การประมวลผลข้อมูลหรือการจัดการข้อมูล (data processing) ดังนั้นสิ่งสำคัญของการสั่งให้คอมพิวเตอร์ทำงานนอกจากคำสั่งต่างๆ คือ ข้อมูล

ภาษาโลโก้มีการจัดการข้อมูล 3 ชนิด คือ

- ตัวเลข (numbers)
- คำ (words)
- รายการ (List)



## ตัวเลข (numbers)

ตัวเลข คือ จำนวน ซึ่งแต่ละจะสามารถนำมา บวก ลบ คูณ หาร  
เปรียบเทียบ เช่น ใช้คำสั่ง PRINT ซึ่งมีรูปแบบดังนี้

PRINT จำนวน หรือ PRINT นิพจน์

ผลลัพธ์ที่ได้จะแสดงในส่วนแสดงผลข้อความ

## กิจกรรมเรื่อง ตัวเลข

2) พิมพ์คำสั่งสำหรับเปรียบเทียบจำนวนในช่องป้อนคำสั่ง  
ดังนี้

PRINT 5=7

PRINT 5=(7-2)

PRINT 5>7

PRINT 5<7

# คำ (words)

คำในภาษาโลโกมีความหมายเหมือนกับคำใน  
ภาษาอังกฤษ โดยมีเครื่องหมายอัฒภาค(“)  
นำหน้าเพื่อแสดงว่า สิ่งที่มาเป็นคำ  
โดยไม่ต้องมีเครื่องหมายอัฒภาคปิดท้าย

# กิจกรรมเรื่อง คำ

1) พิมพ์คำสั่งต่อไปนี้ในช่องป้อนคำสั่ง

PRINT “

PRINT “ HELLO

PRINT “ ABCXYZ

PRINT “ R2D2

PRINT “AB.\$-)\*\*

PRINT “ 25

PRINT “3+4

PRINT “25+ “25



# รายการ (List)

รายการ เป็นการที่จะเชื่อมคำให้เป็นกลุ่มโดยอยู่  
ภายใต้วงเล็บก้ามปู รายการอาจคลุมทั้ง ตัวเลข  
คำ และตัวรายการเอง

# กิจกรรมเรื่อง รายการ

1) พิมพ์คำสั่งต่อไปนี้ในช่องป้อนคำสั่ง

```
PRINT []
```

```
PRINT [HELLO THERE]
```

```
PRINT [1 2 3 4 5 6]
```

```
PRINT [MY NAME IS DANG]
```

```
PRINT [THIS IS LIST:[THIS IS A LIST:]]
```

```
PRINT [[AB][CD][EF]]
```

# คำสั่งจัดการคำและรายการ

ในการประมวลผลข้อมูล จำเป็นต้องมีคำสั่งจัดการคำและรายการ เพื่อช่วยให้การเขียนโปรแกรมสามารถใช้ข้อมูลต่างชนิดกันได้อย่างมีประสิทธิภาพ คำสั่ง WORD เป็นคำสั่งในการรวมคำหลายคำให้เป็นคำเดียวกัน และคำสั่ง SENTENCE เป็นคำสั่งที่ใช้ในการรวมคำกับรายการ และ รายการกับรายการ

# คำสั่ง WORD

คำสั่ง word ใช้สำหรับรวมคำ มีรูปแบบดังนี้

WORD “คำสั่ง 1 “ คำสั่ง 2

ถ้าต้องการเขียนคำมากกว่า 2 คำ ให้ใส่ไว้ในวงเล็บพร้อมกับคำ WORD

# กิจกรรมเรื่อง การใช้คำสั่ง WORD

1) พิมพ์คำสั่งต่อไปนี้ในช่องป้อนคำสั่ง

```
PRINT WORD "BIG "WORD
```

```
PRINT WORD "WO "RD
```

```
PRINT (WORD "WO "RD)
```

```
PRINT WORD "BIG "GER "WORD
```

```
PRINT (WORD "BIG "GER "WORD)
```

```
PRINT WORD ("BIG "GER "WORD)
```

# คำสั่ง SENTENCE

คำสั่ง sentence ใช้ในการเชื่อมคำกับคำ  
คำกับรายการ และรายการกับรายการ โดยมีรูปแบบ  
ดังนี้

SENTENCE “คำที่ 1 “คำที่ 2 เมื่อต้องการเชื่อมคำกับคำ

SENTENCE “คำ [รายการ] เมื่อต้องการเชื่อมคำกับรายการ

SENTENCE [รายการ][รายการ] เมื่อต้องการเชื่อมรายการกับรายการ



# หมายเหตุ

1. ถ้ามีคำหรือรายการมากกว่า 2 ตัวที่จะมาเชื่อม ต้องใส่วงเล็บโดยให้นำคำสั่ง SENTENCE ไว้ในวงเล็บพร้อมกับคำหรือรายการที่ต้องการเชื่อม
2. SENTENCE อาจใช้ตัวย่อเป็น SE

# กิจกรรมเรื่อง การใช้คำสั่ง SENTENCE

## 1) พิมพ์คำสั่งต่อไปนี้ในช่องป้อนคำสั่ง

PRINT SENTENCE "A [Word plus a list]

PRINT SENTENCE [A List plus A] "Word

PRINT SENTENCE "TWO "WORDS

PRINT SE [TWO Lists] [MAKE A List,too]

PRINT (SE "THIS [will become][one List] "Too)

# คำสั่งตัดคำ

โปรแกรมโลโกมีคำสั่งที่ใช้ในการตัดคำ ซึ่งใช้ในการจัดการข้อมูลวิธีหนึ่ง คำสั่งเหล่านี้คือ FIRST LAST BUTFIRST(BF) และ BUTLAST(BL) ใช้เมื่อต้องการส่วนหนึ่งส่วนใดของคำและรายการ

## กิจกรรมเรื่อง การใช้คำสั่งตัดคำ

1) พิมพ์คำสั่งต่อไปนี้ในช่องป้อนคำสั่ง

```
PRINT FIRST "HELLO
```

```
PRINT LAST "HELLO
```

```
PRINT BUTFIRST "HELLO
```

```
PRINT BUTLAST "HELLO
```

# กิจกรรมเรื่อง การใช้คำสั่งตัดคำ

## 1) พิมพ์คำสั่งต่อไปนี้ในช่องป้อนคำสั่ง

PRINT FIRST [HELLO MY FRIEND]

PRINT LAST [HELLO MY FRIEND]

PRINT BUTFIRST [HELLO MY FRIEND]

PRINT BUTLAST [HELLO MY FRIEND]

PRINT FIRST BUTFIRST [HELLO MY FRIEND]

PRINT BUTFIRST FIRST [HELLO MY FRIEND]

# คำสั่ง READWORD

การรับข้อมูลในการประมวลผล นอกจากจะใส่ข้อมูลลงไป  
ในคำสั่งของกระบวนการหรือการเรียกกระบวนการ  
แล้วอาจจะรับข้อมูลโดยการป้อนข้อมูลขณะทำงานได้โดย  
ใช้คำสั่ง READWORD รอผู้ใช้พิมพ์ข้อมูลเข้าไปกรอบ  
โต้ตอบ Input Mode ก่อนทำงานตามคำสั่งต่อไป



# การใช้คำสั่ง READWORD

To TALK

Print [Please Type Something for me To Say]

Print SE [you just made me say] READWORD

END

To BACKTALK

Print [Please Type Something for me To Say]

Print SE [But I hate to say] READWORD

BACKTALK

END

To AGREE

Print [Tell me Something you like]

Print (SE [I like] READWORD [Too])

Print [Tell me Something you hate]

Print (SE [I Hate] Readword [Even more Than you Do!])

AGREE

END

## การใช้ค่าและรายการเป็นตัวแปร

กระบวนการความที่มีตัวแปร โดยค่าตัวแปรที่กำหนดให้เป็นข้อมูลตัวเลขใช้สำหรับเปลี่ยนค่าของขนาดหรือรูปร่างของรูปภาพ ส่วนในกิจกรรมนี้กล่าวถึงกระบวนการความที่มีตัวแปร โดยค่าของตัวแปรจะเป็นค่าหรือรายการเพื่อสามารถเปลี่ยนแปลงการประมวลผลค่าได้หลายรูปแบบโดยใช้กระบวนการเดียว

# การกำหนดค่าตัวแปรเป็นคำ รายการ และ ตัวเลข

To SPEAK :Message

Print [The Message I am going to print is]

Print :Message

END

ป้อนคำสั่งให้เต่าทำงานด้วยคำสั่งต่อไปนี้

SPEAK “Hello ผลที่ได้คือ .....

ค่าที่กำหนดให้กับตัวแปร Message คือ.....เป็นข้อมูลชนิด .....

SPEAK [Now Is The Time] ผลที่ได้คือ.....

ค่าที่กำหนดให้กับตัวแปร Message คือ.....เป็นข้อมูลชนิด .....

SPEAK 103 ผลที่ได้คือ.....

ค่าที่กำหนดให้กับตัวแปร Message คือ.....เป็นข้อมูลชนิด .....

# การใช้คำสั่ง MAKE กำหนดค่าตัวแปรเป็น คำ รายการ และตัวเลข

## 1. สั่งเต่าให้ทำงานตามคำสั่งต่อไปนี้ในช่องป้อนคำสั่ง

```
MAKE "message [this is getting silly]
```

```
PRINT :message
```

```
MAKE "message "Hello
```

```
PRINT :message
```

```
MAKE "message 1000
```

```
PRINT :message 1000
```

```
PRINT "message
```

```
PRINT :message
```

# การใช้คำสั่ง MAKE เพื่อเปลี่ยนค่าของตัวแปรชนิดตัวเลข

## 1. สั่งเต่าให้ทำงานตามคำสั่งต่อไปนี้ในช่องป้อนคำสั่ง

```
MAKE "Number 5
```

```
PRINT :Number
```

```
MAKE "Number :Number + 5
```

```
PRINT :Number
```



# การใช้คำสั่ง MAKE เพื่อเปลี่ยนค่าของตัวแปรชนิดรายการ หรือคำ

## 1. สั่งเต่าให้ทำงานตามคำสั่งต่อไปนี้ในช่องป้อนคำสั่ง

MAKE "Message [Hello There]

PRINT :Message

MAKE "Message Sentence :Message "Friend

PRINT :Message

# การกำหนดค่าตัวแปรเป็น คำ รายการ และตัวเลข

## 1. สร้างกระบวนการความ grow :message

To grow :message

PRINT [Type Something new]

MAKE “NewPart Readword

MAKE “