**LOGO Programming**

*- an introduction to Logo by*[*Mike Koss*](http://mckoss.com/)

**Logo Lessons**

I prepared a number of hand-out materials and work sheets when we started the class.

**Logo Facts**

* [**Turtle Basics**](http://mckoss.com/logo/turtle.htm)
* [**Drawing Polygons**](http://mckoss.com/logo/polygons.htm)
* [**The Basic Logo Commands**](http://mckoss.com/logo/basiclogo.htm)
* [**Worksheet**](http://mckoss.com/logo/logowork.htm)

**Using the Repeat Command**

* [**Fun with REPEAT**](http://mckoss.com/logo/repeat.htm)

**Making your own Words**

* [**Making new words**](http://mckoss.com/logo/newwords.htm)
* [**Words that use Numbers**](http://mckoss.com/logo/wordnumbers.htm)
* [**Worksheet**](http://mckoss.com/logo/logows2.htm)

**Music and Random Numbers**

* [**Making Music with LOGO**](http://mckoss.com/logo/logomusic.htm)
* [**How to "Roll the Dice"**](http://mckoss.com/logo/rolldice.htm)
* [**Worksheet**](http://mckoss.com/logo/logows3.htm)

[**Fun Programs**](http://mckoss.com/logo/funprog.htm)

***Turtle Basics***



The turtle looks like a little triangle in the middle of the screen. The head shows you which direction he is facing. When the turtle moves he draws a line behind him.



You can turn the turtle by telling him to turn RIGHT or LEFT. This pictures shows how to make the turtle turn in different directions. Instead of typing out RIGHT and LEFT I use the smaller words RT and LT, which mean the same thing to the turtle.

FD 100

To get the turtle to draw a line, just tell him to move FORWARD a number of steps (turtles take small steps, so you need to tell him to move FORWARD many steps, like 100, to get him to move enough to see). I like to use the smaller word FD instead of typing FORWARD. Both work the same way.



To get the turtle to draw a shape, like a square, you can give him the instructions he needs to "walk" around the shape of a square.

***Polygons***

One of the easiest things to do in logo is use the turtle to draw various polygons. You can notice that in order to draw a closed polygon, the turtle has to *walk* around the edges, eventually turning through 360 degrees before coming back *home*. Since the turtle makes N turns for an N-sided polygon, the size of each turn in 360/N degrees.

****

**Triangle (3 sides)
Turtle Turns 120º**

REPEAT 3 [FD 100 RT 120]

****

**Square (4 sides)
Turtle Turns 90º**

REPEAT 4 [FD 100 RT 90]

****

**Pentagon (5 sides)
Turtle Turns 72º**

REPEAT 5 [FD 100 RT 72]

****

**Hexagon (6 sides)
Turtle Turns 60º**

REPEAT 6 [FD 100 RT 60]

****

**Septagon (7 sides)
Turtle Turns 51º**

REPEAT 7 [FD 100 RT 51]

****

**Octagon (8 sides)
Turtle Turns 45º**

REPEAT 8 [FD 100 RT 45]

***Basic Commands***

|  |  |
| --- | --- |
| **Command** | **What it does** |
| FD 100 | Move the turtle forward 100 steps. |
| RT 90 | Turn the turtle to the right 900. |
| LT 90 | Turn the turtle to the left 900. |
| BK 100 | Move the turtle backwards 100 steps. |
| PU | Pick the turtle's pen up off the paper. |
| PD | Put the turtles pen back down on the paper. |
| CS | Clear the screen and start over. |
| HT | Hide the turtle (triangle). |
| ST | Show the turtle (triangle). |
| REPEAT 3 [...] | Repeat the commands 3 times. |

***WORKSHEET – BAI TAP THUC HANH***

**Draw the shape in the right box that goes with the turtle instructions in the left box.**

|  |  |
| --- | --- |
| **FD 100RT 90FD 100RT 90FD 100RT 90FD 100RT 90** |  |
| **FD 100RT 120FD 100RT 120FD 100RT 120** |  |
| **REPEAT 8 [FD 100 RT 135]** |  |
| **FD 100BK 50RT 90FD 50LT 90FD 50BK 100PURT 90FD 25LT 90PDFD 50PUFD 10PDFD 5HT** |  |

***Fun with Repeat***

You can have fun with the REPEAT command just by changing three numbers!

**

Which three numbers make your favorite drawing?

**CS REPEAT \_\_ [FD \_\_\_ RT \_\_\_]**

Try this! What does it draw?

**CS REPEAT 100 [FD 10 RT REPCOUNT]**

**Make Your Own Words**

**(TẠO MỘT THỦ TỤC/HÀM CỦA BẠN)**

**One of the coolest things about Logo is that you can make your own words that do whatever you want them to do. When you make a word, you can use it in other commands and programs just as if it was part of Logo to begin with.**

**What if we want to draw a lot of squares. Each time we have to type in: (muốn vẽ nhiều hình vuông)**

**REPEAT 4 [FD 100 RT 90]**

**That's not a whole lot of typing, but wouldn't it be nice if we could just type: (ta định nghĩa một thủ tục/hàm)**

**SQUARE**

**You can teach LOGO what a "SQUARE" is like this:**

**(cách định nghĩa, sử dụng nút Edall – và lưu vào workspace)**

**TO SQUARE
    REPEAT 4 [FD 100 RT 90]
END**

**Now we can use our new word to make cool new programs. Can you figure out what this does?**

**(có thể gọi lệnh từ thủ tục đã tạo SQUARE)**

**REPEAT 36 [SQUARE RT 10]**

**Words that Use Numbers**

**(TRUYỀN THAM SỐ)**

The words we know in LOGO can do something different if you type different numbers after them. FD 100 draws a line with 100 steps in it, and FD 50 draws a line with only 50 steps in it.

We can make our new words use numbers too. Let's change our SQUARE word to draw squares of different sizes:

**TO SQUARE :SIZE
    REPEAT 4 [FD :SIZE RT 90]
END**

**Now you can tell logo to:**

**SQUARE 50**

**or**

**SQUARE 100**

**Here's another cool program. What does it do?**

**REPEAT 100 [SQUARE REPCOUNT RT 10]**

**WORKSHEET – BÀI TẬP**

Draw the shape in the right box that goes with the turtle instructions in the left box.

|  |  |
| --- | --- |
| **TO TRI :SIZEREPEAT 3 [FD :SIZE RT 120]END** |  |
| **TO HEXTRI :SIZEREPEAT 6 [TRI :SIZE RT 60]END** |  |
| **TO POLY :SIDESREPEAT :SIDES [FD 50 RT 360/:SIDES]END** |  |
| **to star :sidesmake "ang 360/:sidesrepeat :sides [FD 50 RT 2\*:ANG FD 50 LT :ANG]end** |  |

**Making Music with Logo**

You can play music in the same way that you make drawings using Logo programs. In order to make it easier to use, I've written a helper program called "[player.lgo](http://mckoss.com/logo/player.htm)". You need to load this file before using the commands in this lesson. Your computer needs to have a sound card or MIDI instrument attached in order to use this program.

You can type in notes, and logo will play them in the same order you type them in:

**do re mi fa sol fa mi re do**

This makes a sound like a piano, but you can also make sounds like different instruments:

**banjo high do re mi fa sol fa mi re do**

You can make notes that play fast or slow:

**eighth do re mi
quarter do re mi
half do re mi
whole do re mi**

You can play really low notes or really high notes:

**dinosaur do re mi
low do re mi
middle do re mi
high do re mi**

**How to "Roll the Dice"**

It's really fun to see if you can get the Turtle to do what you want it to do, but it's also fun to have the turtle "make up his own mind". There is a word in logo (RANDOM) that will pick a number for you. You can use RANDOM anyplace you need to use a number in your program. This program prints random numbers between 0 and 11.

**REPEAT 20 [PRINT RANDOM 12]**

Let's use RANDOM to draw lines all over the screen:

**TO LINES
    REPEAT 100 [SETXY RANDOM 200 RANDOM 200]
END**

We can also make "random" music:

**TO SONG
    REPEAT 24 [NOTE RANDOM 12]
END**

We can even make a "picture" of the music:

**TO SONG2
    REPEAT 24 [PLAYDRAW RANDOM 12 REPCOUNT]
END

TO PLAYDRAW :NOTE :WHERE
    SETXY :WHERE\*12 :NOTE\*12
    NOTE :NOTE
END**

WORKSHEET – BAI TAP THUC HANH

**Here are some cool new words. What do they do?**

|  |  |
| --- | --- |
| to sheep    quarter mi re do re mi mi half mi    quarter re re half re    quarter mi sol half sol    quarter mi re do re mi mi mi mi    re re mi re whole doend |  |
| to sample    reset    repeat 128 [instrument repcount    quarter do re mi fa whole sol]end |  |
| piano sheephigh banjo sheephigh flute sheep |  |

|  |  |
| --- | --- |
| **Pumpkin** | http://mckoss.com/logo/images/pumpkin.jpg |

*The text of this program can be downloaded from here:*[pumpkin.lgo](http://mckoss.com/logo/pumpkin.lgo).  *Right click on the link and select Save Target As... to save it to your machine.  You can then use the File/Load menu command in MSWLogo to load the file.  Once you do that, type "****pumpkin 200****" in the command window to see it draw on screen.*

[pumpkin.lgo](http://mckoss.com/logo/pumpkin.lgo)

; Fillpoly

;

; Creates a polygon filled in with a color. The :size is the approximate

; diameter. :color is an RGB list. The polygon is created about the turtle

; at its center.

**to fillpoly :sides :size :color**

**poly :sides :size**

**setfloodcolor :color**

**fill**

**end**

; Draws the head of the Pumpkin (i.e., the orange circle)

to head :size

fillpoly 50 :size [255 127 0]

end

; Draws the mouth of the pumpkin - turtle starts in the middle

to mouth :size

local "dzMouth

local "ptSave

make "dzMouth :size/8

make "ptSave pos

pd

lt 45 fd :dzMouth

lt 90 fd :dzMouth

rt 90 fd :dzMouth

rt 45

pu setpos :ptSave pd

rt 45 fd :dzMouth

rt 90 fd :dzMouth

lt 90 fd :dzMouth

lt 45

pu setpos :ptSave

end

; A generic move function - always picks up the pen.

; x and y are relative to the current heading of the turtle.

to move :dx :dy

pu fd :dy rt 90 fd :dx lt 90

end

; A Generic polygon drawing functions. The turtle is placed in the middle

; of the polygon. The :size is the approximate diameter of the polygon.

to poly :sides :size

local "step

make "step 3.14\*:size/:sides

pu

bk :size/2 lt 90 bk :step/2

pd

repeat :sides [fd :step rt 360/:sides]

pu

fd :step/2 rt 90 fd :size/2

end

; Draws a pumpkin of a given diameter.

to pumpkin :size

local "dxEye

local "dyEye

local "dyMouth

local "dzMouth

local "ptCenter

make "dxEye :size/5

make "dyEye :size/8

make "dyMouth :size/3

make "dzMouth :size/8

make "ptCenter pos

setpensize [5 5]

head :size

tri :size

move -:dxEye :dyEye rt 180

tri :size

move -2\*:dxEye 0

tri :size

pu setpos :ptCenter

fd :dyMouth rt 180

mouth :size

pu setpos :ptCenter

end

; A cool array of 7 pumpkins.

to pumpkins

pumpkin 200

repeat 6 [fd 210 pumpkin 200 bk 210 rt 60]

ht

end

; Draws triangular eyes and nose dimensioned for the :size'd pumpkin head.

to tri :size

fillpoly 3 :size/6 [0 0 0]

end

**Recursive Trees**

You can make some nifty pictures of tree by using recursion.

to tree :size
  if :size < 5 [stop]
  fd :size
  lt 30 tree :size\*.7
  rt 60 tree :size\*.7
  lt 30 bk :size
end

to rtree :size
  if :size < 5 [stop]
  fd :size
  lt 30 rtree :size\*(((random 5)+5)/10)
  rt 60 rtree :size\*(((random 5)+5)/10)
  lt 30 bk :size
end

**Towers of Hanoi**

The Towers of Hanoi puzzle may be familiar to you. It is a game of 3 pegs and a number of disks. Each disk has a different size. The puzzle starts with all the disks on one peg, with the largest disks on the bottom. The problem is to move the disks from the first peg to the last peg **without** ever placing a larger disk on top of a smaller one.

A simple program to solve this problem is this one. It merely prints the "moves" to be made.

to move :from :to
(print "Move "disk "on "peg :from "to "peg :to ".)
end

to tower :disks :from :to :using
if :disks = 1 [move :from :to stop]
tower :disks-1 :from :using :to
move :from :to
tower :disks-1 :using :to :from
end

A more complex version (with some satisfying graphics displaying an animation of all the moves - type "towers 10" to start the demo...you can turn Trace on to slow down the program to see the individual moves if the program is running too swiftly).



to drawdisk :disk :peg :height :fDraw
pu
setxy :peg\*150-300 :height\*10
ifelse :fDraw [penpaint] [penerase]
bk :disk\*5
fd :disk\*10
end

to move :from :to
make "moves :moves + 1
; (print :moves ". "Move "disk "on "peg :from "to "peg :to ".)

make "height (item :from :stacks)
make "disk (item :height (item :from :pegs))
setitem :height (item :from :pegs) ".
setitem :from :stacks (item :from :stacks)-1
drawdisk :disk :from :height "false

make "height (item :to :stacks)+1
setitem :to :stacks :height
setitem :height (item :to :pegs) :disk
drawdisk :disk :to :height "true

; show :stacks
; show :pegs
end

to start :disks
cs
ht
make "moves 0
rt 90
setpensize [5 5]
make "stacks (array 3)
setitem 1 :stacks :disks
setitem 2 :stacks 0
setitem 3 :stacks 0
make "pegs (array 3)
repeat 3 [setitem repcount :pegs (array :disks)]
make "firstPeg (item 1 :pegs)
repeat :disks [setitem repcount :firstPeg :disks-repcount+1]
repeat :disks [drawdisk :disks-repcount+1 1 repcount "true]
end

to tower :disks :from :to :using
if :disks = 1 [move :from :to stop]
tower :disks-1 :from :using :to
move :from :to
tower :disks-1 :using :to :from
end
to towers :disks
start :disks
tower :disks 1 3 2
end