Calico Graphics Reference

Window:
Window(title, width, height) – Constructor
getMouse() - waits until user clicks and returns (x,y) of location
WINDOW.setBackground(makeColor(r,g,b))
WINDOW.clear – clears all objects but keeps background color

Graphics Objects Inherited Methods:
Methods inherited by all graphics objects:
Generating Colors: makeColor(red, green, blue) parameter value range 0...255
OBJ.fill = makeColor('color')
OBJ.outline = makeColor('color')
OBJ.draw(Window) – Draws the object onto a graph window. Updates are automatic.
OBJ.undraw() – Removes the object from a graph window.
OBJ.move(dx,dy) – Relative to current location.
OBJ.border = # - Changes the thickness of the border

Point
Point(x,y) – Only for reference.

Dot
Dot(x,y) – Constructor
DOT.getX()  
DOT.getY()

Line
Line(point1, point2) – Constructor
LINE.getCenter() – Returns point at line center
LINE.getP1()  
LINE.getP2()

Circle
Circle(centerPoint, radius) – Constructor
CIRCLE.getCenter()

Polygon
Polygon(point1,point2,...) - Constructor
POLY.rotate(degrees ccw)

Curve
Curve(point1,point2,point3,point4)-Constructor

Oval
Oval(center, width, height) - Constructor
OVAL.getCenter()

Rectangle
Rectangle(point1, point2)
RECT.getCenter()  
RECT.getP1()  
RECT.getP2()

Text
Text(anchorPoint, string) – Constructor
TEXT.fill = Color('color') or TEXT.setFill(makeColor(r,g,b))
TEXT.rotate(degrees ccw)
TEXT.width – gives you width of text in pixels
TEXT.height – gives you the height of text in pixels
TEXT.fontSize = #

Image
Picture(url) or Picture(filename with path)
savePicture(picture,filename)
savePicture([picture, ...], filename) - GIF
Picture(width, height, color)
**Examples:**

#Bouncing Ball
from Myro import *
from Graphics import *
win = Window("Pong", 500, 500)
xPos = 50
yPos = 50
xDelta = -2
yDelta = 5
aBall = Circle(Point(xPos, yPos), 10)
aBall.draw(win)
aBall.setFill(makeColor(255, 0, 0))

for t in timer(60):
    if (0 > xPos) or (500 < xPos):
        xDelta = -xDelta
    if (0 > yPos) or (500 < yPos):
        yDelta = -yDelta

    xPos = xPos + xDelta
    yPos = yPos + yDelta

    aBall.move(xDelta, yDelta)
wait(.01)

#DrawATriangle
from Graphics import *
def main():
    win = Window('Draw a Triangle', 350, 350)

    win.setBackground(makeColor(0, 0, 200))
    message = Text(Point(win.getWidth()/2, 30),
                'Click on three points')
    message.fill = Color('red')
    message.fontSize = 20
    message.draw(win)

    # Get and draw three vertices of triangle
    p1, p2 = win.getMouse()
    point1 = Dot(p1, p2)
    point1.draw(win)

    p3, p4 = win.getMouse()
    point2 = Dot(p3, p4)
    point2.draw(win)

    p5, p6 = win.getMouse()
    point3 = Dot(p5, p6)
    point3.draw(win)

    # Use Polygon object to draw the triangle
    triangle = Polygon(Point1, Point2, Point3)
    triangle.setFill(makeColor('lightgray'))
    triangle.setOutline(makeColor('cyan'))
    triangle.setOutlineWidth(4)  # width of boundary line
    triangle.draw(win)

    message.undraw()
    message = Text(Point(win.getWidth()/2, 30),
                'Click anywhere to quit')
    message.setFill(Color('red'))
    message.setFontSize(20)
    message.draw(win)
    win.getMouse()
    win.close()

main()