

painting in java (and the ArtLab)

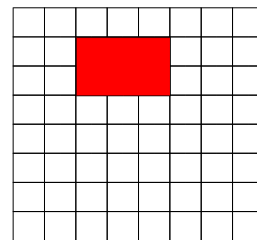
goals

- try out creating images/art with programs
- gain experience with programming concepts
- learn a little about the Java programming language

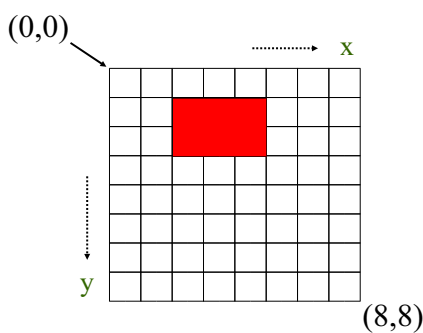
ArtLab drawing components

- line
- rectangle
- oval
- arc

how to specify a rectangle?

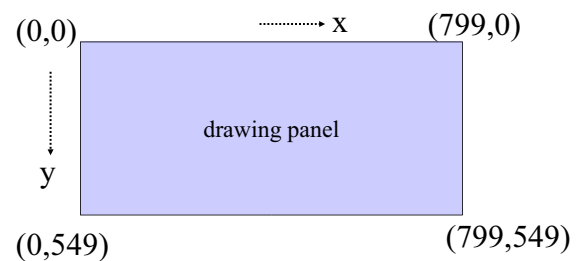


- use *top left* and *bottom right* corners
- or: use *top left corner*, *width*, *height*



top left corner (x,y) = (2,1)
width = 3
height = 2

the ArtLab drawing panel



specifying a rectangle in the ArtLab java code

```
//red rectangle
newShape ();
x = 550;
y = 400;
width = 200;
height = 100;
g2.setPaint (red);
g2.fillRect (x, y, width, height);
```

notice some old friends!

variables

```
// red rectangle    comment
newShape ();
x = 550;           assignment
y = 400;           statements
width = 200;
height = 100;
g2.setPaint (red);
g2.fillRect (x, y, width, height);
```

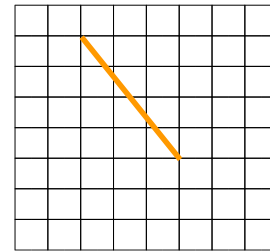
last two lines of rectangle code

```
// red rectangle
newShape ();
x = 550;
y = 400;
width = 200;
height = 100;
g2.setPaint(red);
g2.fillRect(x, y, width, height);
```

methods of g2 include
setPaint and fillRect

the things in parentheses are
parameters to the methods

how to specify a line?

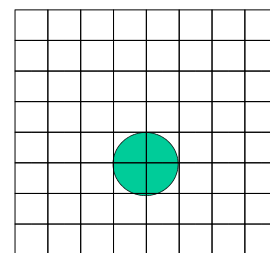


- end point 1: (2,1)
- end point 2: (5,5)

specifying a line in ArtLab

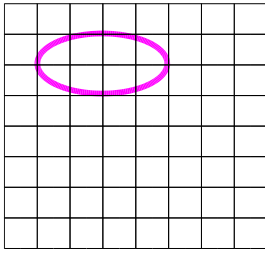
```
// orange line
newShape ();
x1 = 500;
y1 = 200;
x2 = 700;
y2 = 10;
g2.setPaint (orange);
g2.drawLine (x1, y1, x2, y2);
```

how to specify a circle?



- center: (4,5)
- width: 2
- top left corner of bounding square: (3,4)
- width: 2

specifying an oval



- top left corner of bounding square: (1,1)
- width: 4, height: 2

specifying an oval in ArtLab

```
//magenta oval
newShape ( );
x = 100;
y = 100;
width = 120;
height = 75;
g2.setPaint (magenta);
g2.drawOval (x, y, width, height);
```

preview: what you will do in the ArtLab

- you will download java code that will produce a blank picture
- you will add a few lines of code for each shape you want in your picture
- by choosing values for the x and y variables, you can control where the shape is placed
- by choosing the colour in g2.setPaint, you can control the colour of your shape

compiling and running your program

after you have added one or more shapes to your code, to view your picture you need to:

- compile your code (translates your code into machine language)
- run (execute) your code (this tells the system to run your program)
- the lab web page explains how to do this

more on compiling

- if there are *syntax errors* in your code, the compiler will report them
- the compiler cannot find *semantic errors*
- if there are no syntax errors, the compiler will translate your code to machine language (0's and 1's)

java terms seen today

- variables
- assignment statements
- methods (roughly like functions in JavaScript)
- parameters