

# Processing

---

- Introduction
- Basics

# Processing

---

## **Introduction**

To **use** a tool on a computer,  
you need do little more than point and click;  
to **create** a tool you must understand  
the arcane art of computer programming

John Maeda, *Creative Code*

# Why Designers Need to Learn Programming?

- quick to learn, easy to output graphic elements
- learn fundamentals of computer programming in a visual context
- visual approach, simple code = good for designers

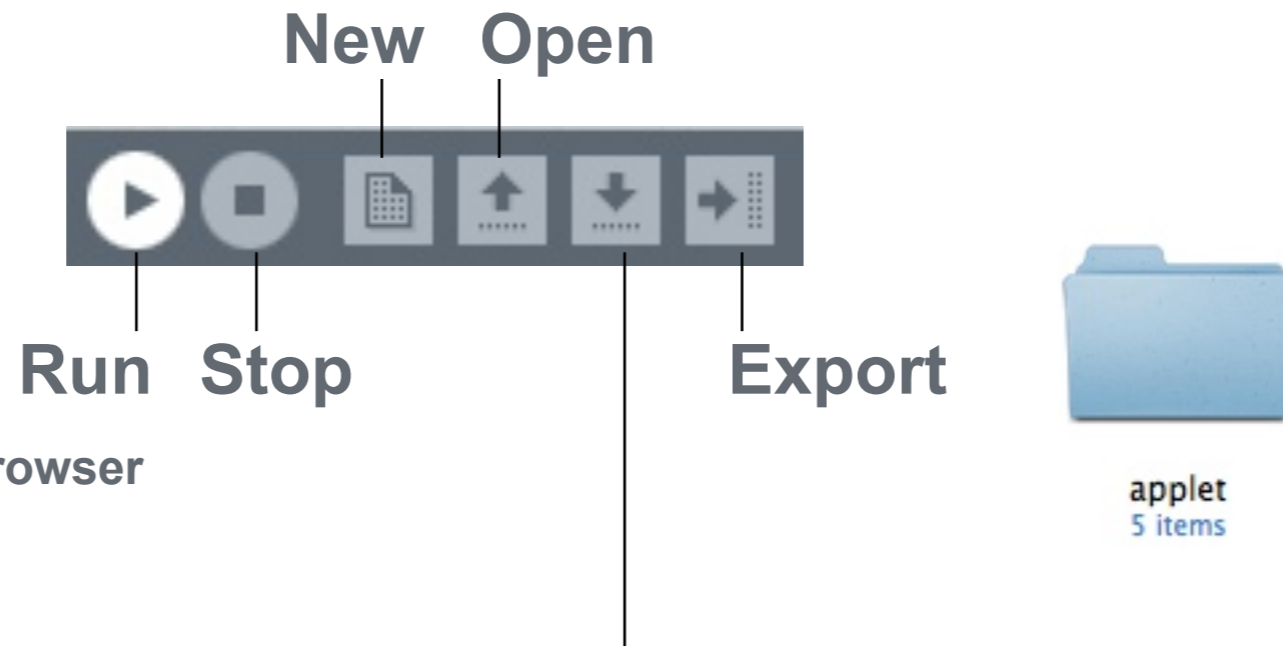
# Why Designers Need to Learn Programming?

- like design, programming is problem solving
- work with programmers, speak their language
- get an idea of what is possible (what isn't)
- designing/programming merging more and more

## Context

- core language and libraries make use of Java, with elements same as C programming
- easily integrated with other media (audio and video) and devices (PC, mobile phones)
- can be as short as one line of code or as long as thousands

# Interface



similar to preview in browser

## Save

use letters, numbers, underscore for filenames  
(but cannot start with numbers)

**Save file before you begin**  
Needed for exporting as image or for web



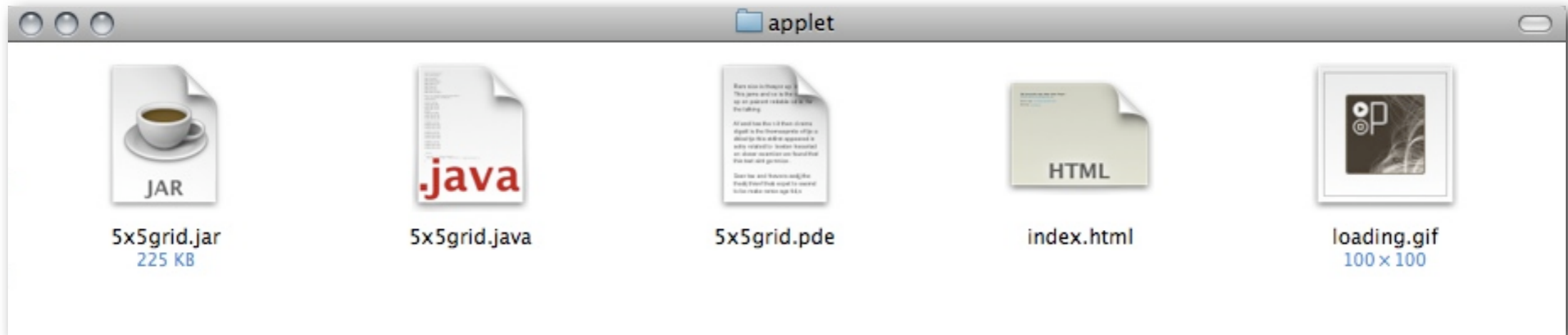
sketch\_jan04b.pde

# Export

applet folder



applet  
5 items





# Errors

what error is

unexpected token: 130

```
at java.awt.Component.processMouseEvent(Component.java:6345)
at javax.swing.JComponent.processMouseEvent(JComponent.java:3255)
at java.awt.Component.processEvent(Component.java:6113)
```

Syntax error, maybe a missing right parenthesis?

```
4714)
2143)
)
ntainer.java:4618)
tainer.java:4279)
er.java:4212)
2129)
)
5)
read.java:296)
d.java:211)
read.java:201)
hThread.java:196)
hThread.java:188)
.java:122)
```

```
expecting RPAREN, found 'B2B'
processing.app.debug.RunnerException: Syntax error, maybe a missing right parenthesis?
    at processing.app.Sketch.preprocess(Sketch.java:1350)
    at processing.app.Sketch.build(Sketch.java:1471)
    at processing.app.Sketch.compile(Sketch.java:1172)
    at processing.app.Editor.handleRun(Editor.java:1564)
    at processing.app.EditorToolBar.mousePressed(EditorToolBar.java:318)
    at java.awt.Component.processMouseEvent(Component.java:6345)
    at javax.swing.JComponent.processMouseEvent(JComponent.java:3255)
    at java.awt.Component.processEvent(Component.java:6113)
    at java.awt.Container.processEvent(Container.java:2085)
    at java.awt.Component.dispatchEventImpl(Component.java:4714)
    at java.awt.Container.dispatchEventImpl(Container.java:2143)
    at java.awt.Component.dispatchEvent(Component.java:4544)
    at java.awt.LightweightDispatcher.retargetMouseEvent(Container.java:4618)
    at java.awt.LightweightDispatcher.processMouseEvent(Container.java:4279)
    at java.awt.LightweightDispatcher.dispatchEvent(Container.java:4212)
    at java.awt.Container.dispatchEventImpl(Container.java:2129)
    at java.awt.Window.dispatchEventImpl(Window.java:2475)
    at java.awt.Component.dispatchEvent(Component.java:4544)
    at java.awt.EventQueue.dispatchEvent(EventQueue.java:635)
    at
```

ignore the scary red text

which line likely has error

# Processing

---

## **Basics**

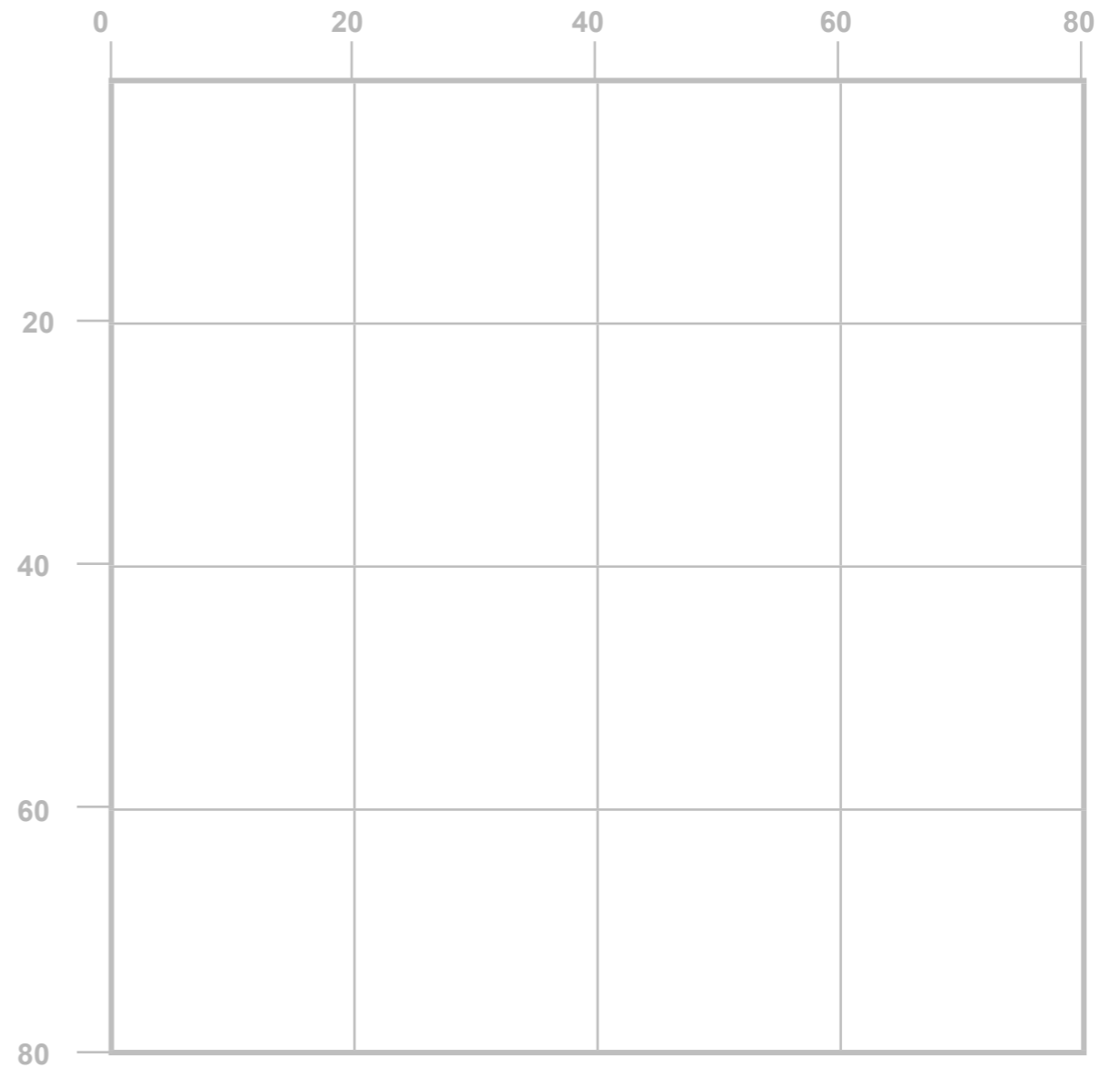
# size

**y axis**  
vertical

```
size(80, 80);
```

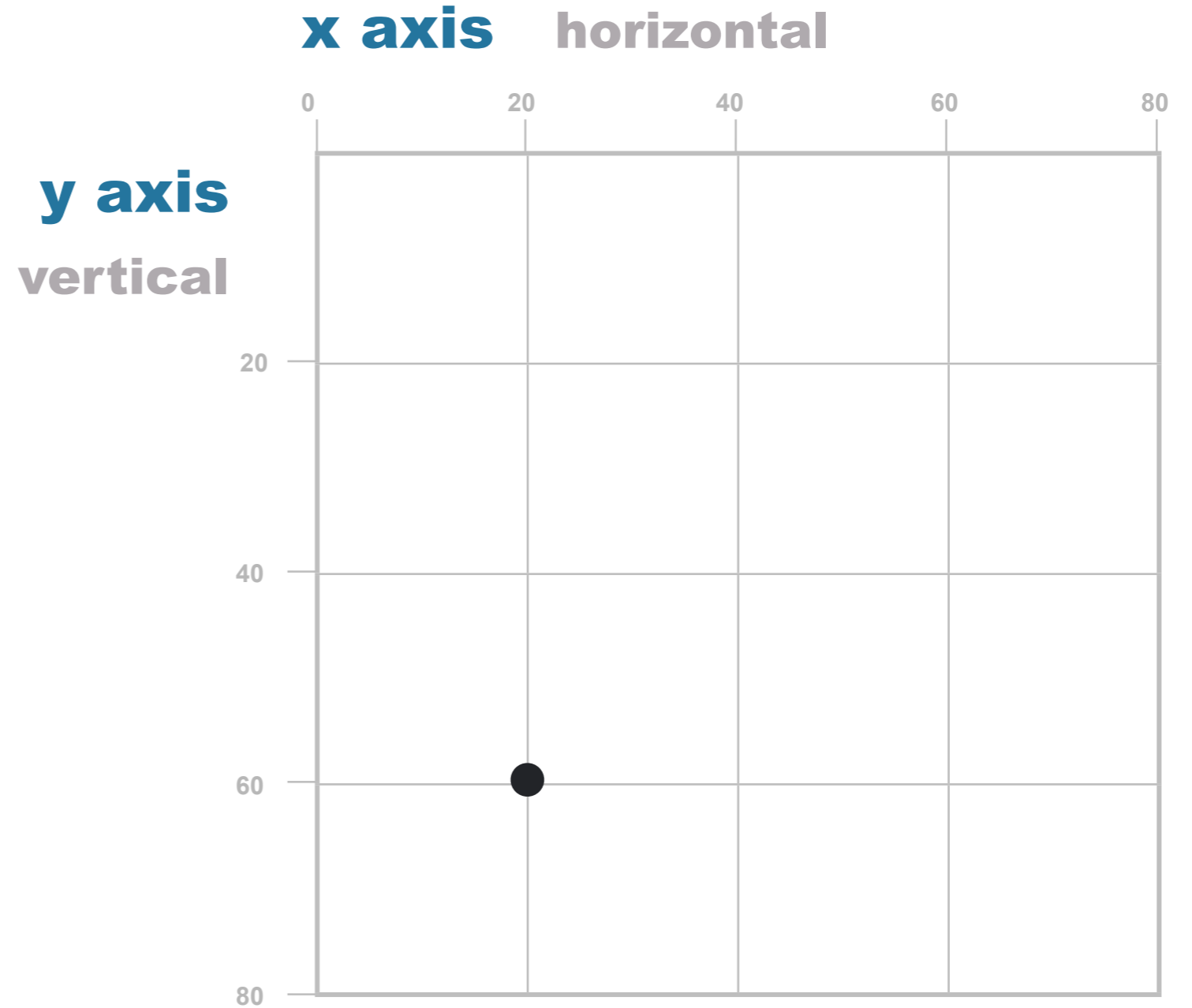
note: the default size of the window is 100x100 pixels.

**x axis** horizontal



## point

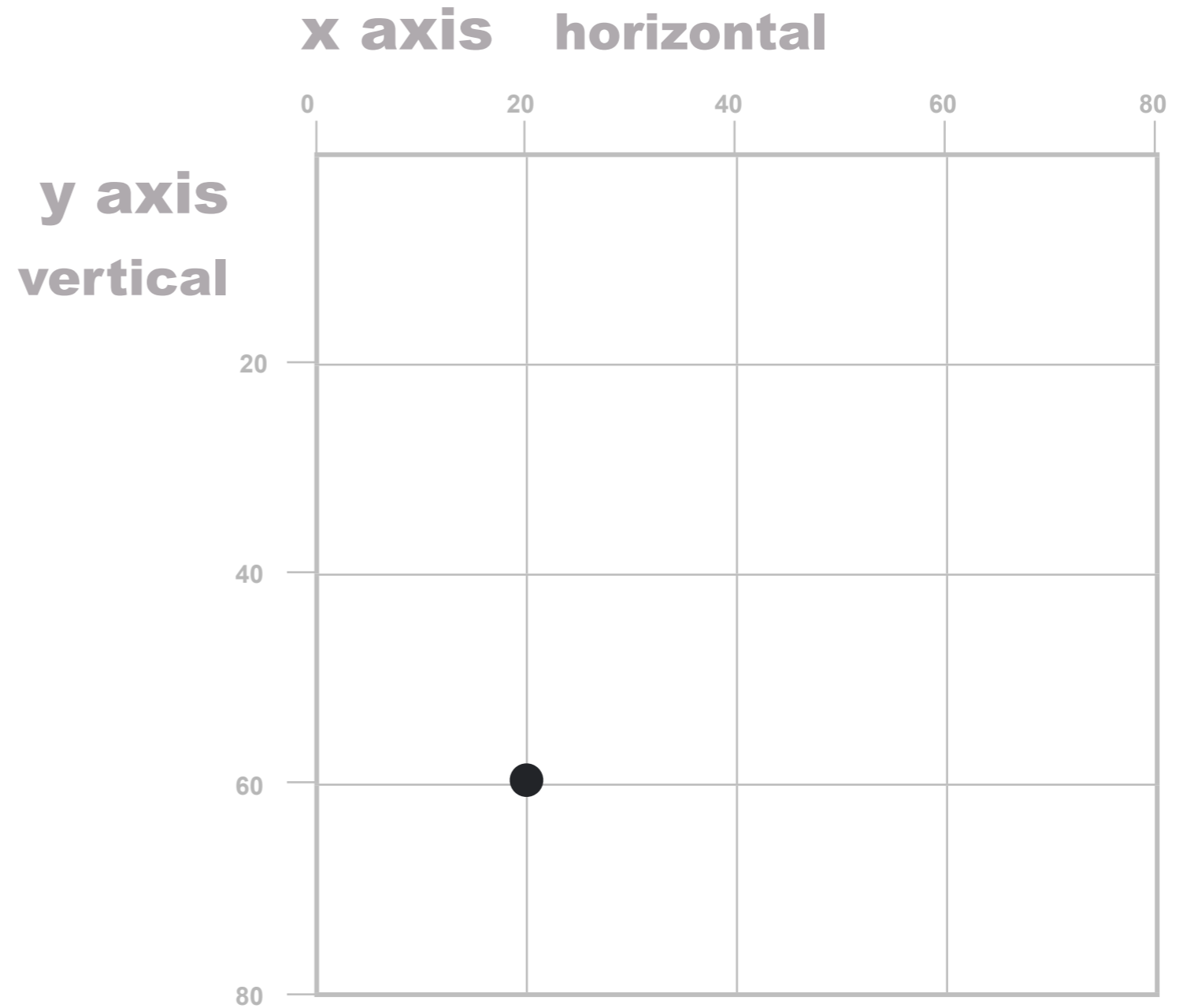
```
      x    y  
point(20, 60);
```



## point

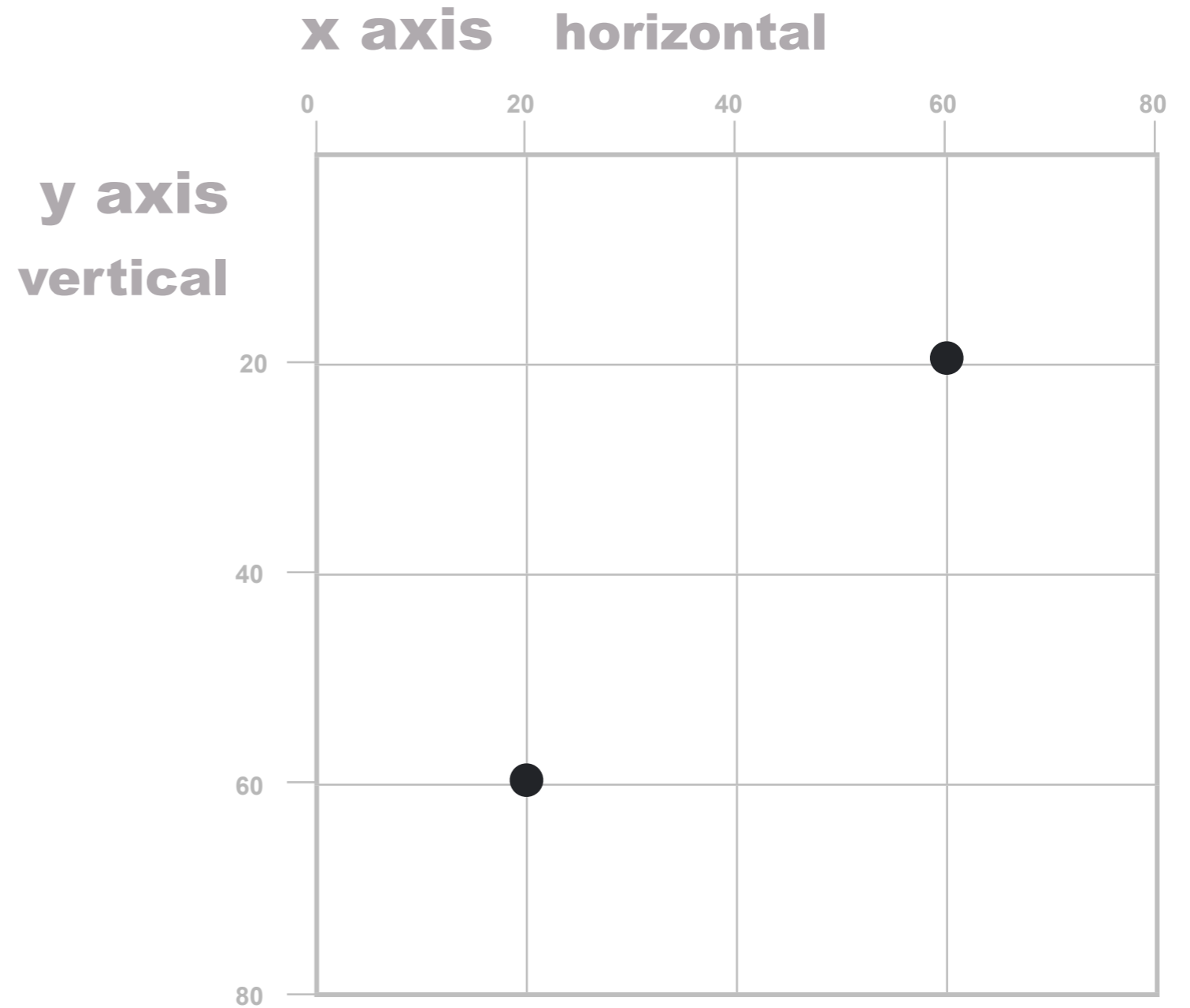
```
point(x, y);
```

Put a point | over here | .



# point

```
      x    y  
point(20, 60);  
point(60, 20);
```

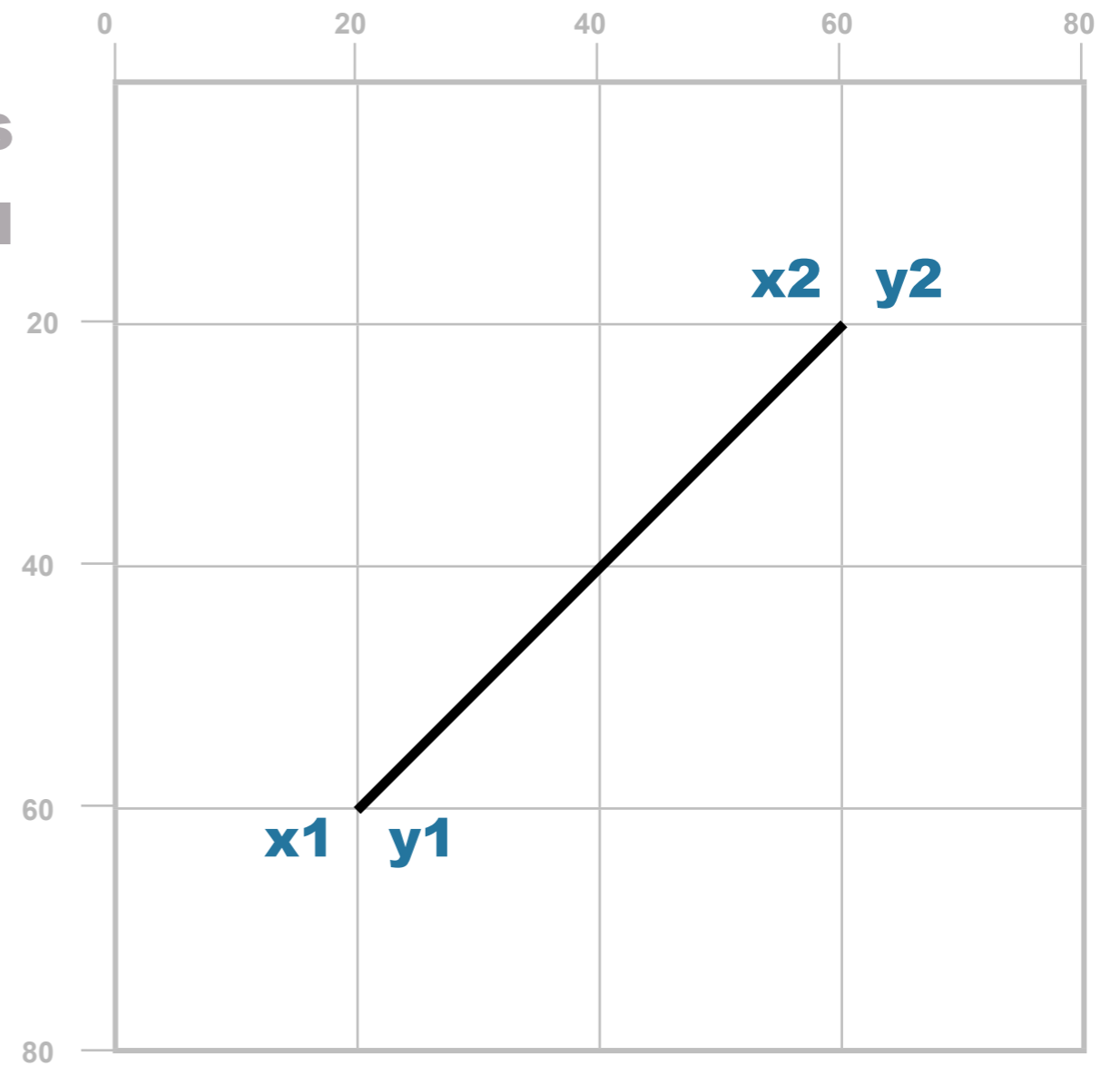


# line

```
      x1   y1   x2   y2  
line(20, 60, 60, 20);
```

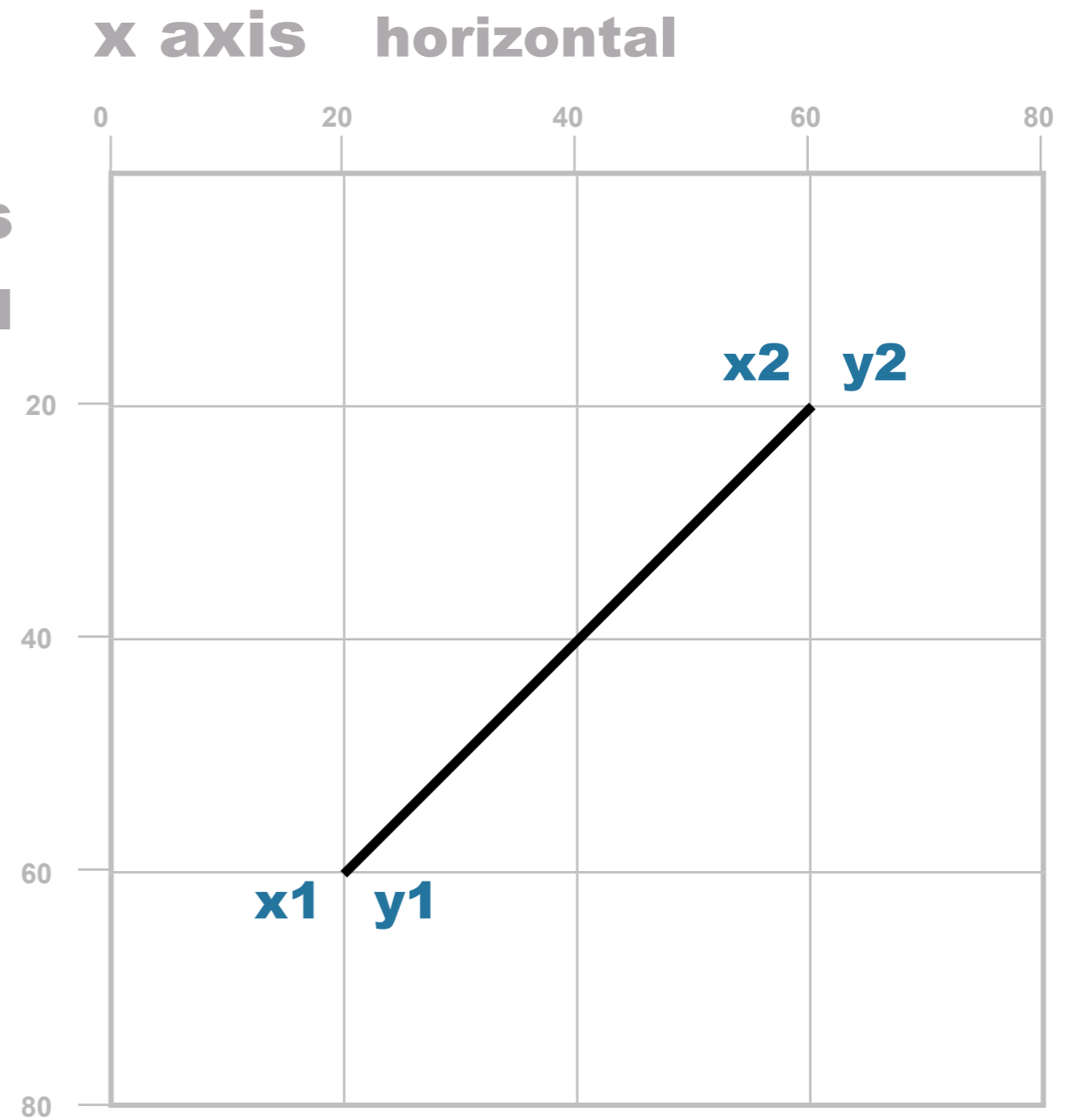
**y axis**  
**vertical**

**x axis** **horizontal**



# line

```
          x1   y1   x2   y2  
line(20, 60, 60, 20);  
Draw a line | from here | to there |.
```



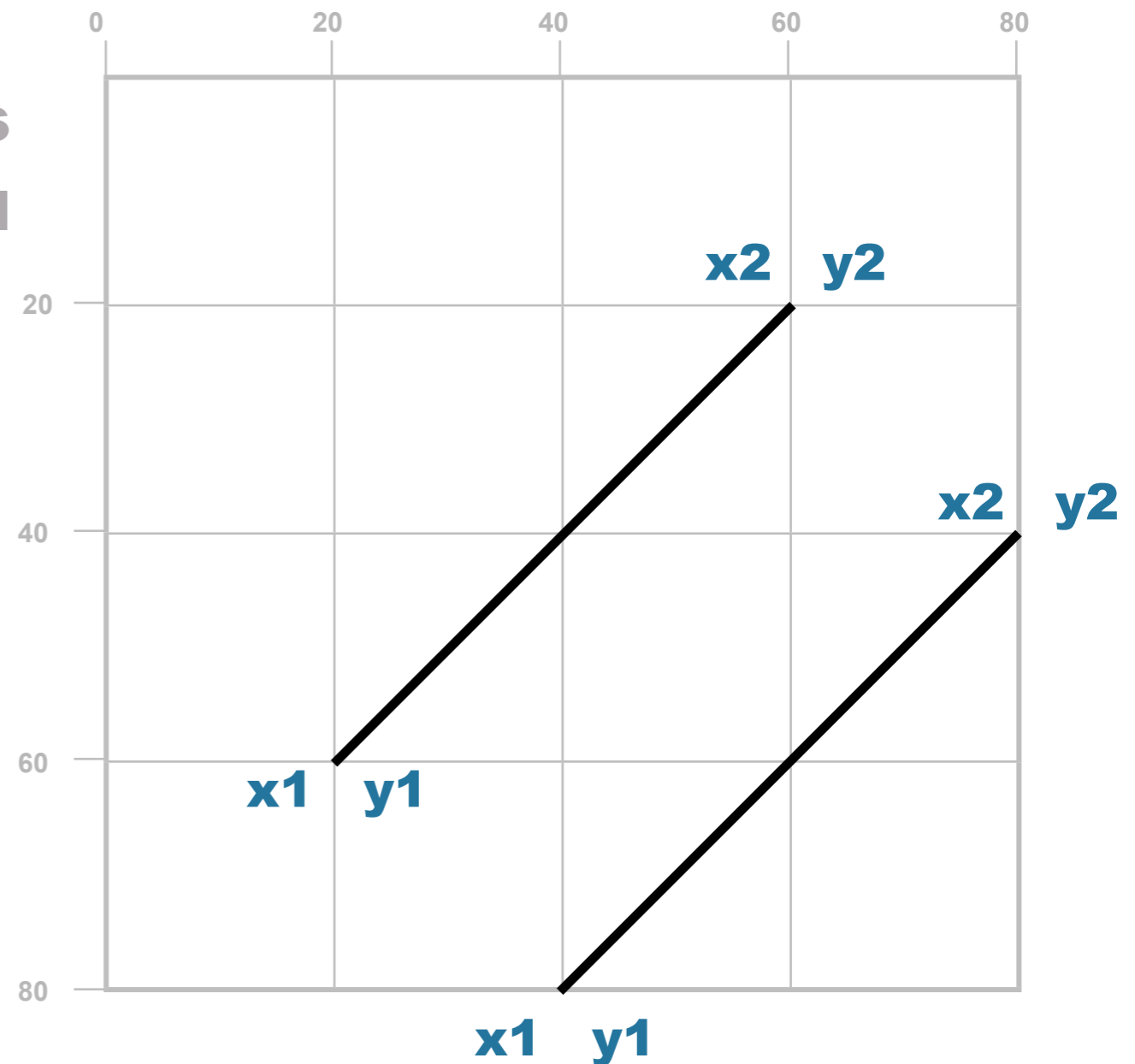


# line

```
      x1   y1   x2   y2  
line(20, 60, 60, 20);  
  
line(40, 80, 80, 40);
```

**y axis**  
**vertical**

**x axis** **horizontal**



# rectangle

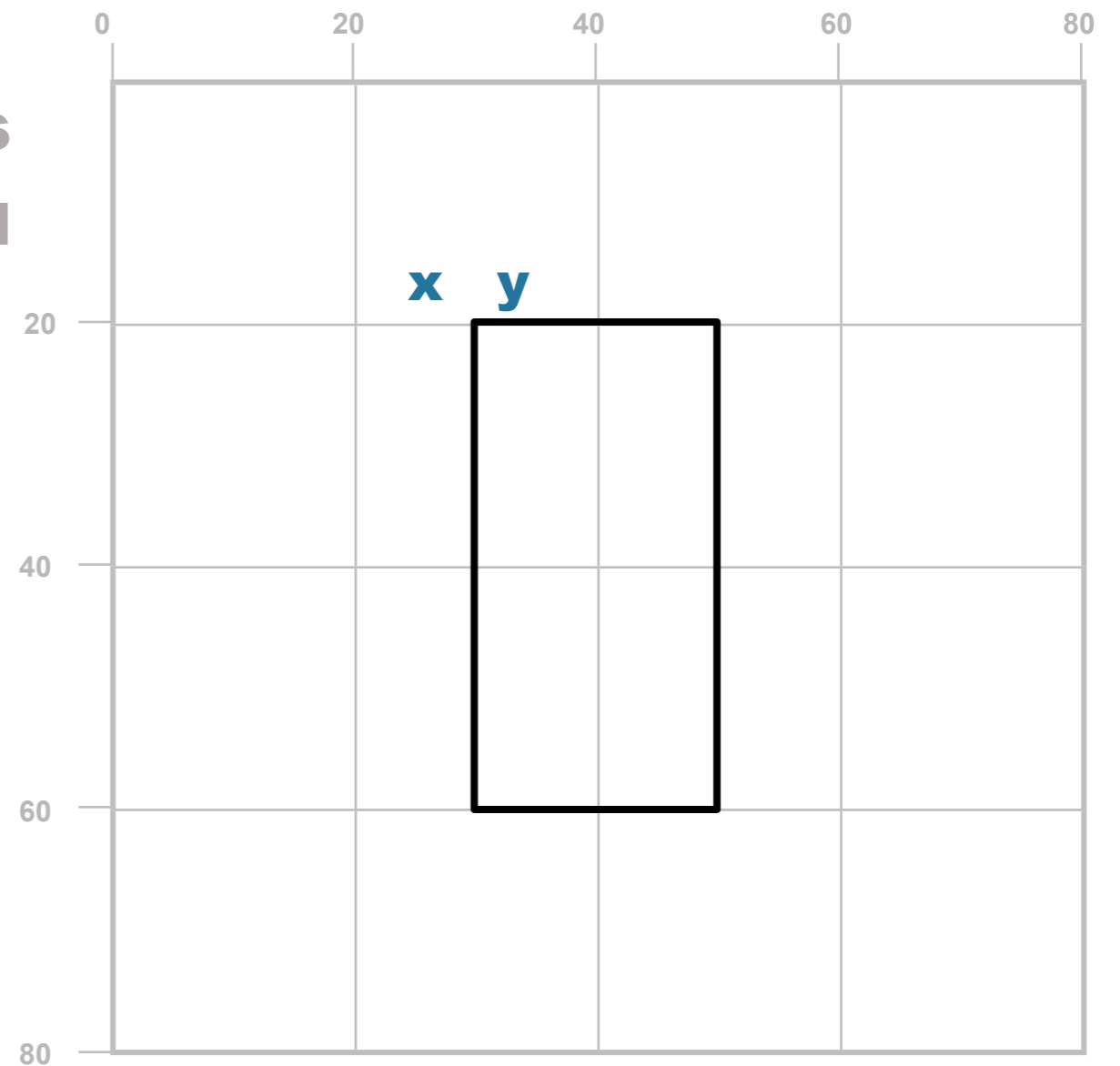
default, drawn from top left

```
rect(30, 20, 20, 40);
```

**x**    **y**    **width**    **height**  
**top left**

**y axis**  
**vertical**

**x axis**    **horizontal**



# rectangle

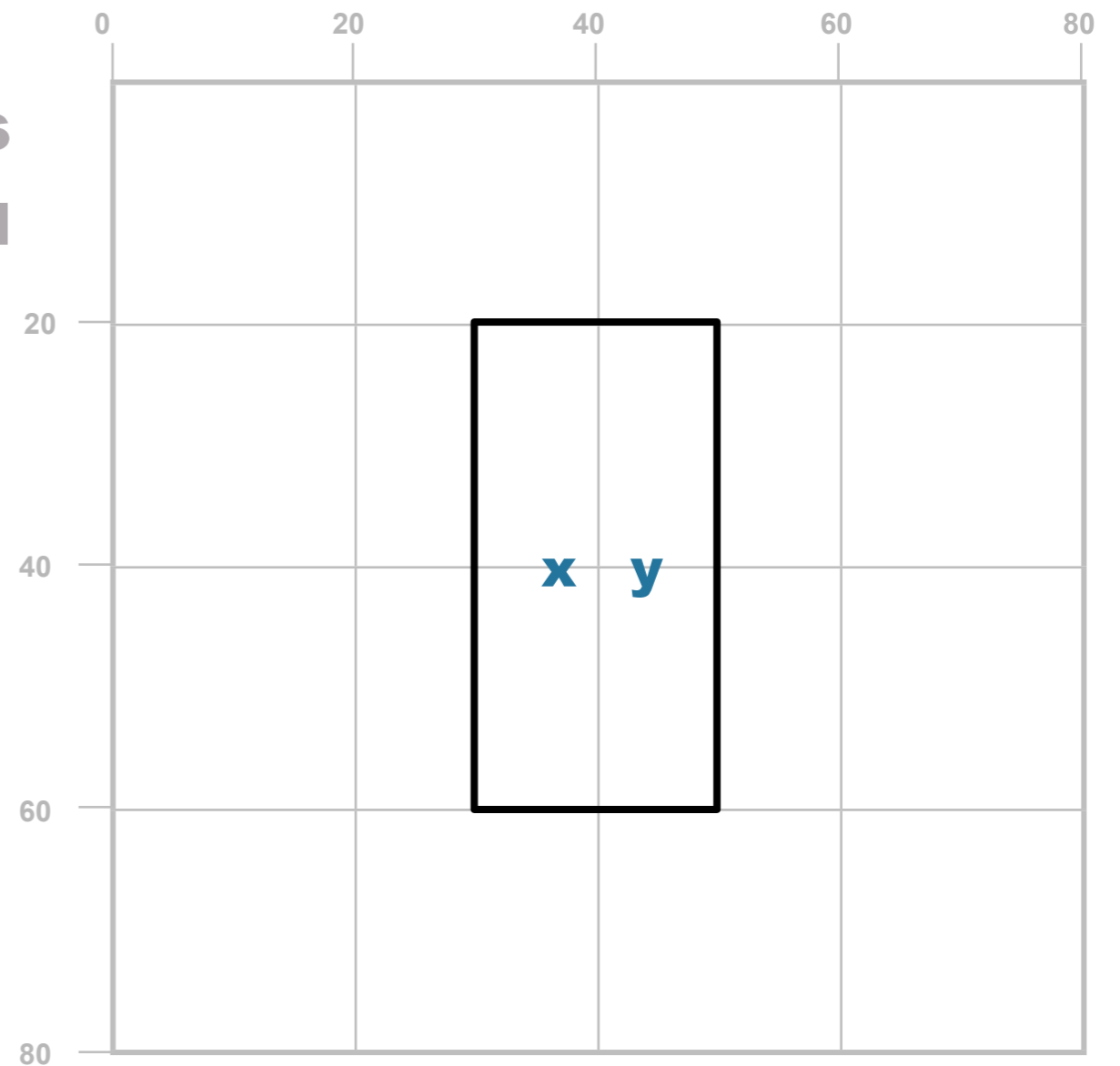
drawn from centre point

```
rectMode(CENTER);  
rect(40, 40, 20, 40);
```

**x**   **y**   **width**   **height**  
**centre**

**y axis**  
**vertical**

**x axis**   **horizontal**



# rectangle

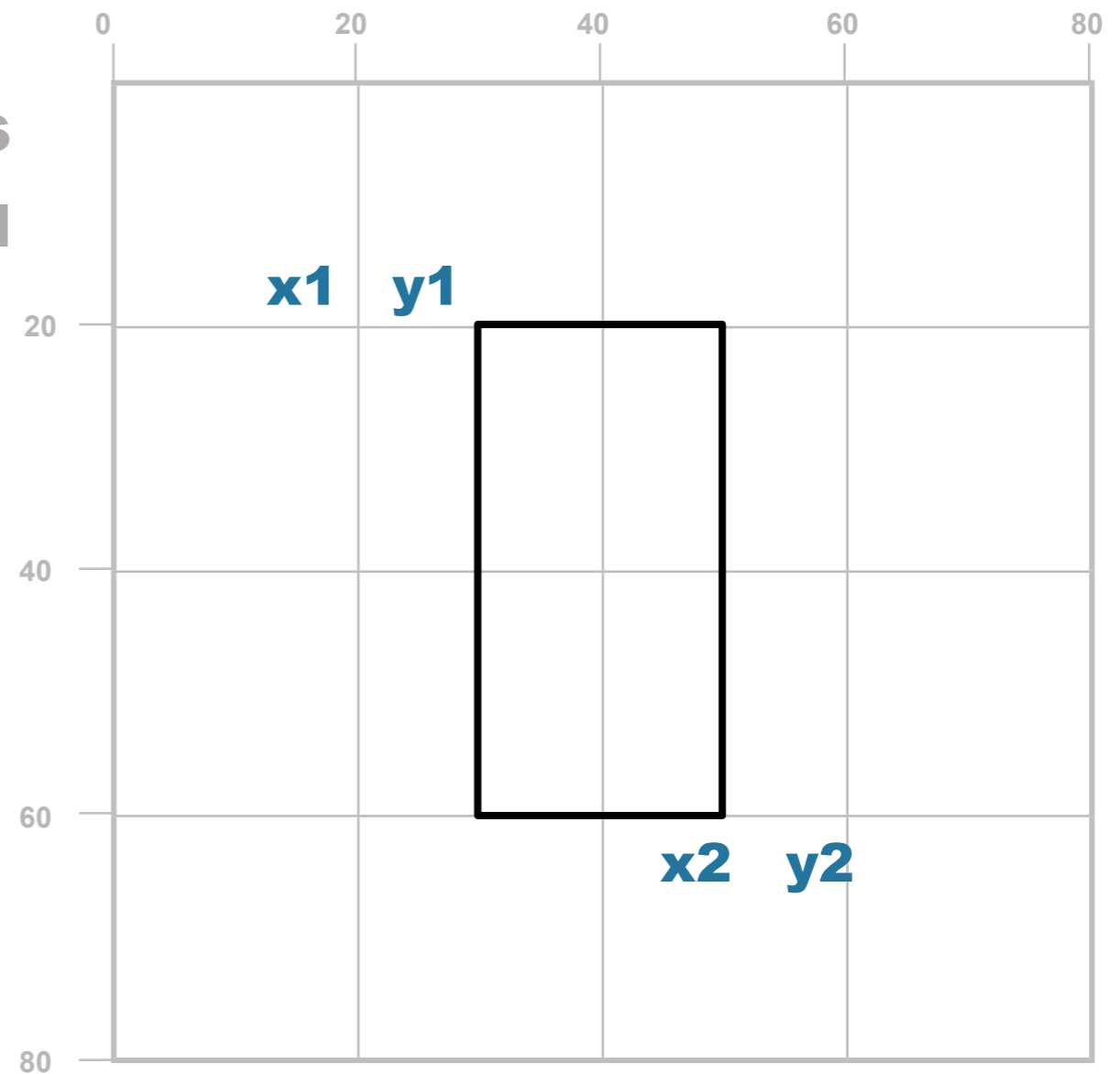
drawn from corners  
top left, bottom right

```
rectMode(CORNERS);  
rect(30, 20, 50, 60);
```

**x1**   **y1**   **x2**   **y2**  
**top left**   **bottom right**

**y axis**  
**vertical**

**x axis**   **horizontal**



# ellipse

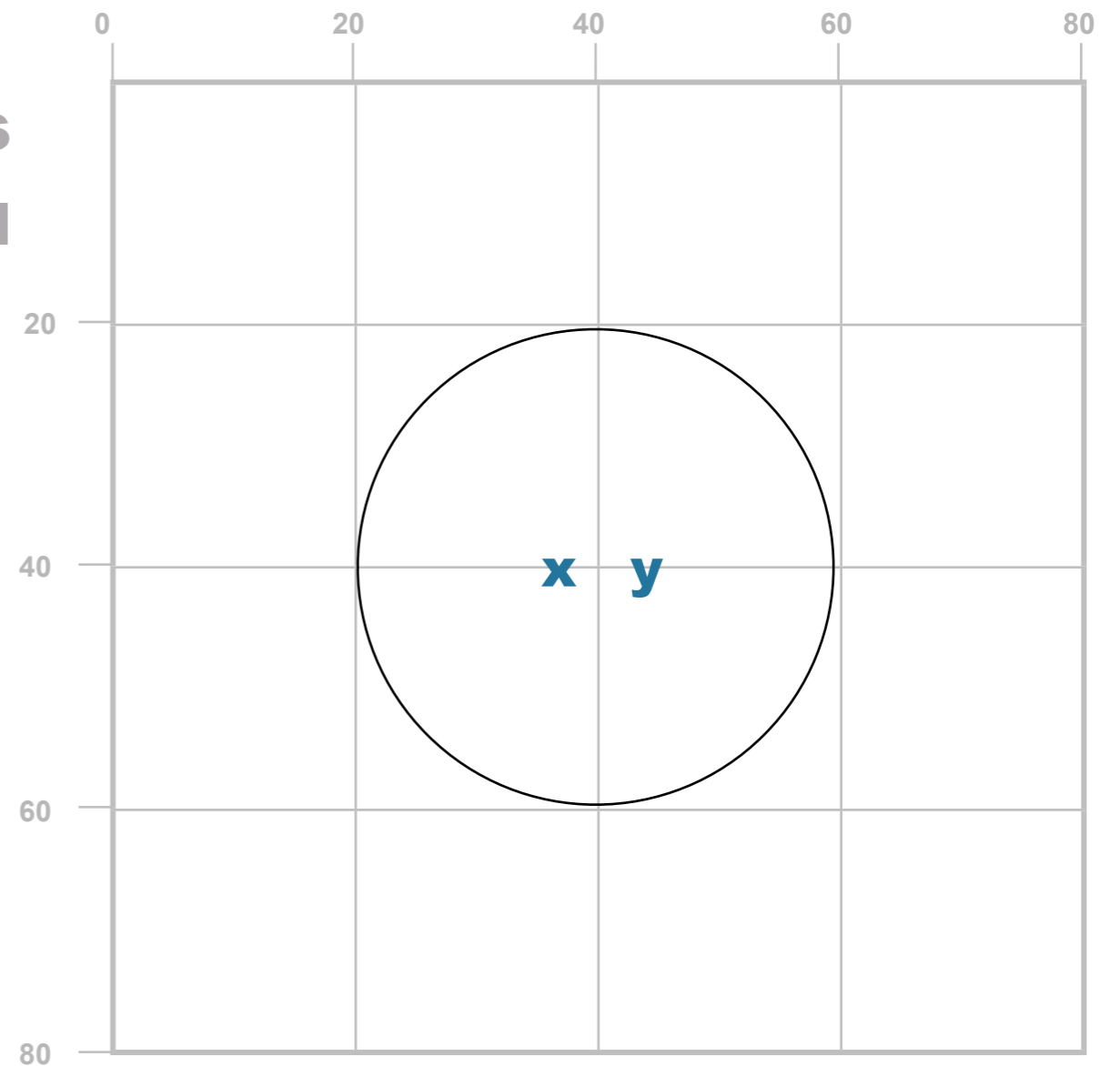
default drawn centre

```
ellipse(40, 40, 40, 40);
```

**x**   **y**   **width**   **height**  
**centre**

**y axis**  
**vertical**

**x axis**   **horizontal**



# ellipse

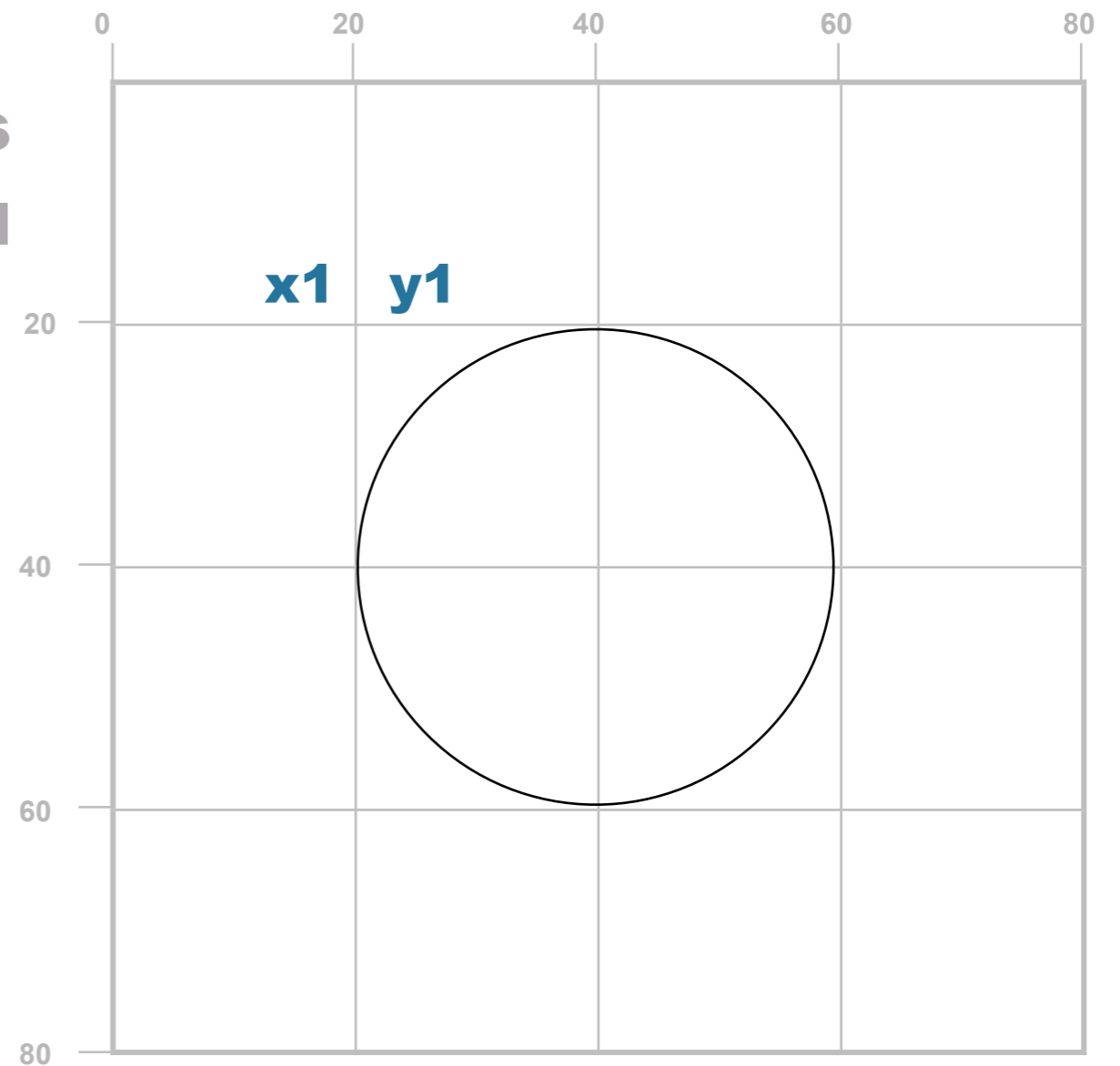
drawn top left

```
ellipseMode(CORNER);  
ellipse(20, 20, 40, 40);
```

**x**   **y**   **width**   **height**  
**top left**

**y axis**  
**vertical**

**x axis**   **horizontal**



# ellipse

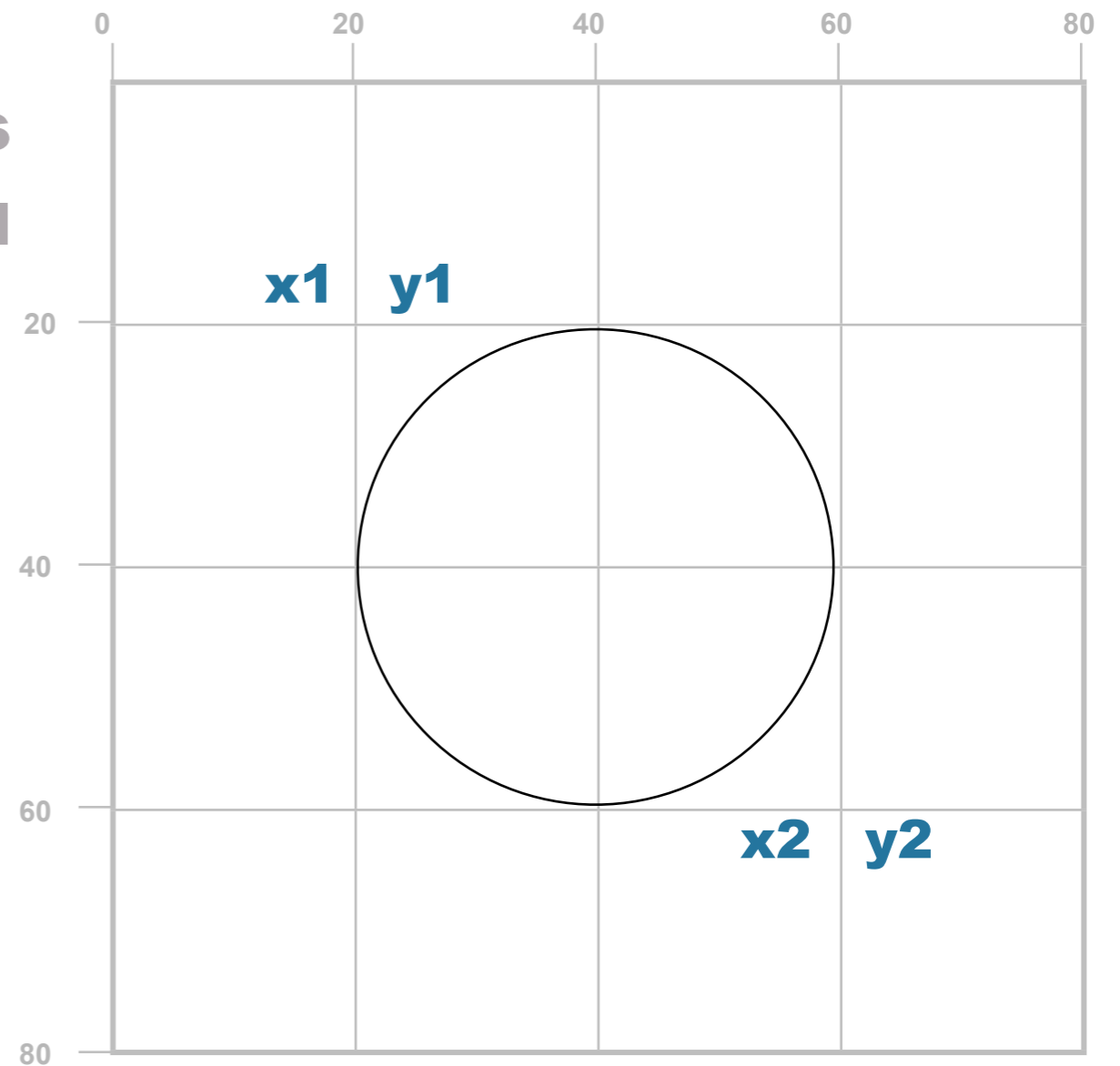
drawn top left

```
ellipseMode(CORNERS);  
ellipse(20, 20, 60, 60);
```

**x**    **y**    **x**    **y**  
**top left**    **bottom right**

**y axis**  
**vertical**

**x axis**    **horizontal**



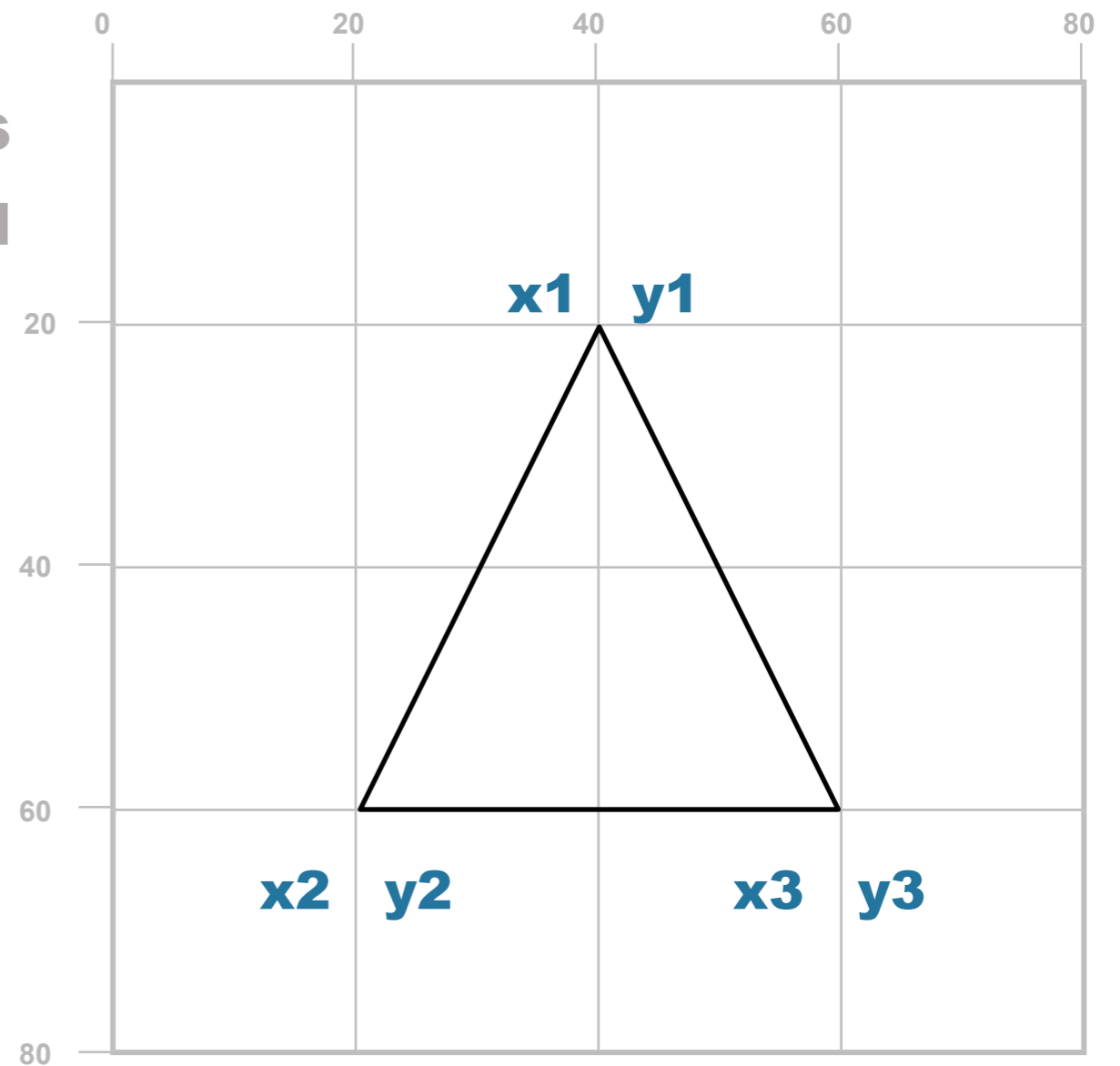
# triangle

drawn top, bottom left,  
bottom right

```
triangle(40, 20, 20, 60, 60, 60);  
      x1 y1, x2 y2, x3 y3,
```

**y axis**  
vertical

**x axis** horizontal





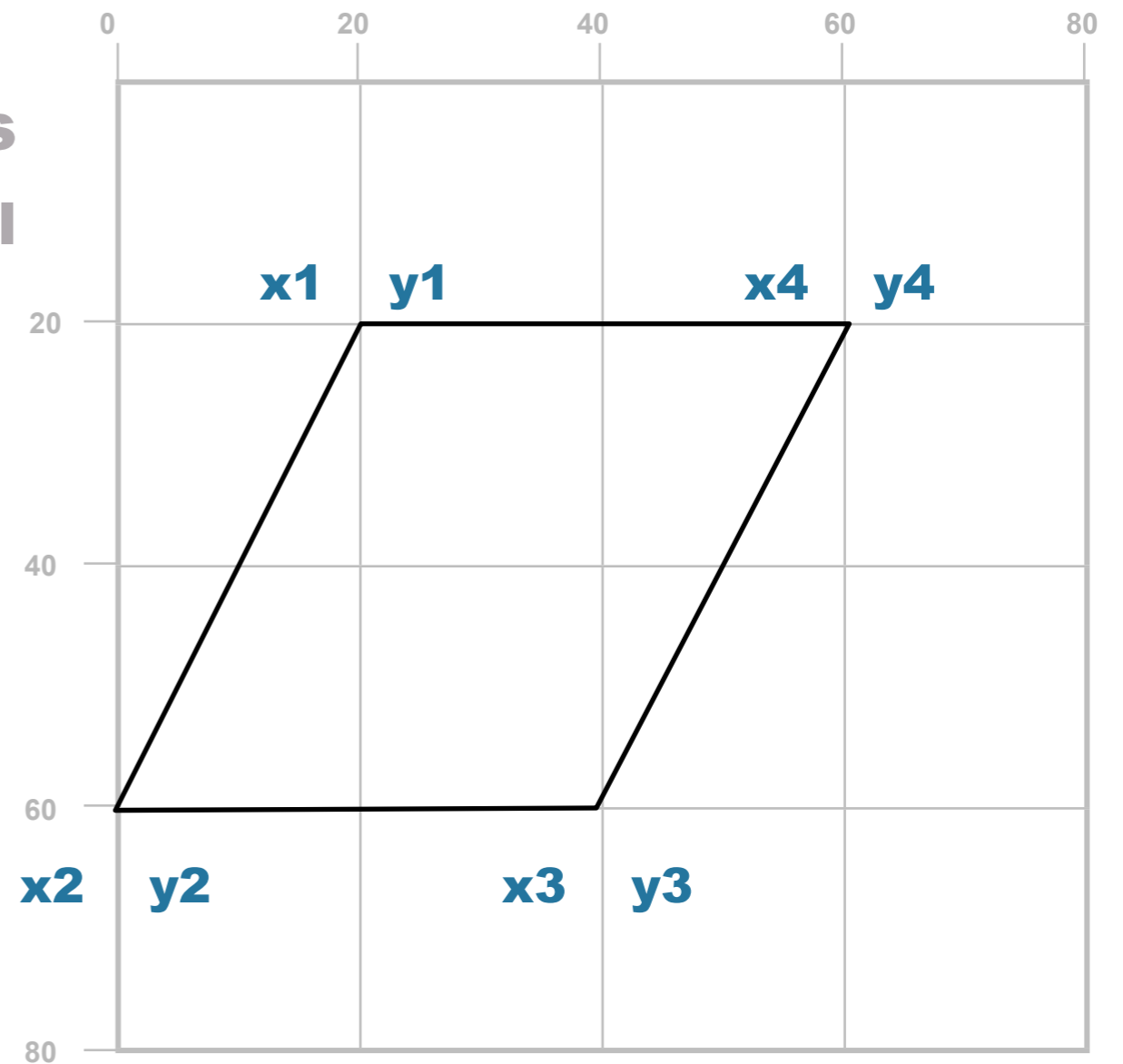
# quad

drawn top left, bottom left,  
bottom right, top right

```
quad(20, 20, 0, 60, 40, 60, 60, 20);  
  x1  y1, x2  y2, x3  y3, x4  y4,
```

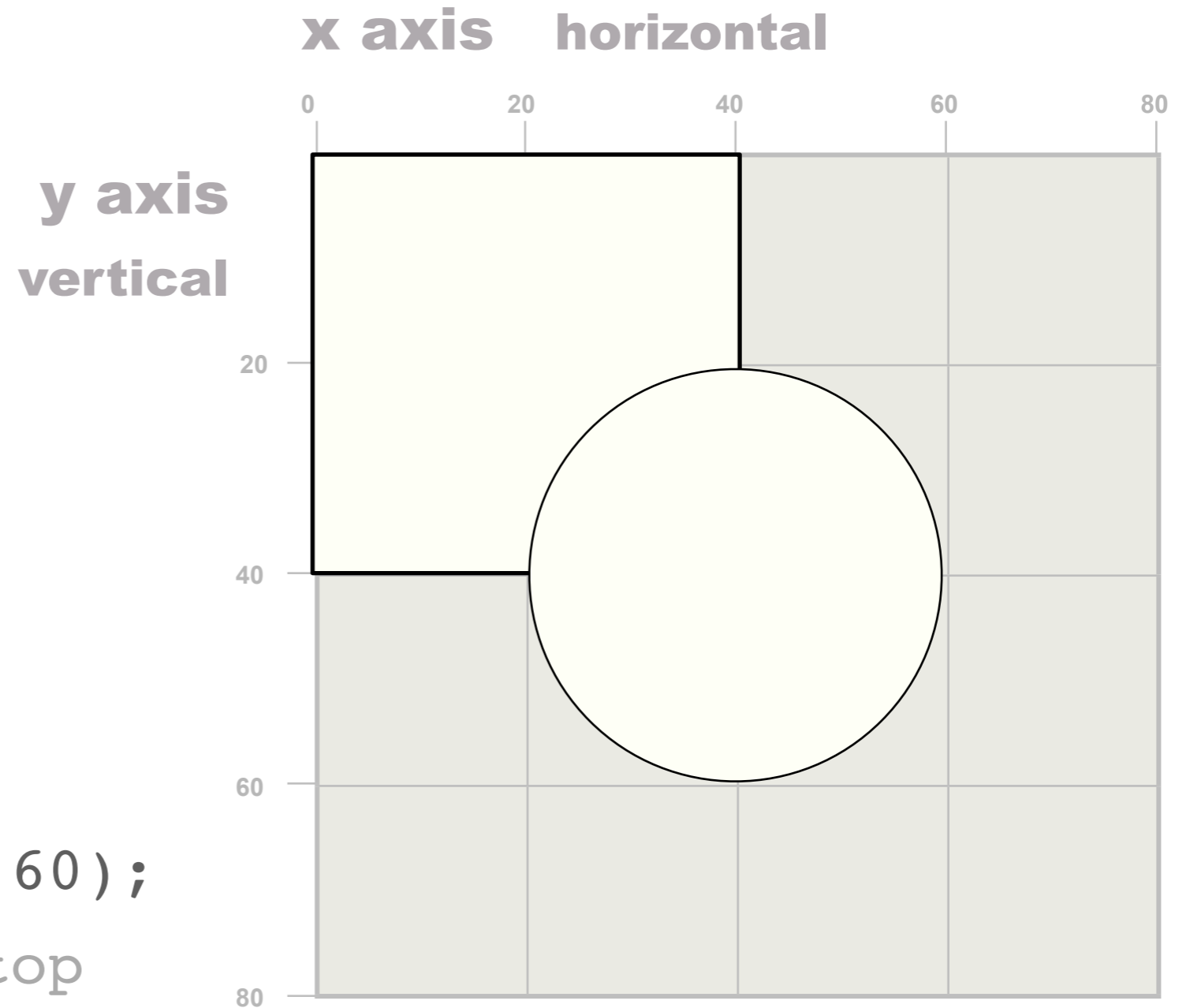
y axis  
vertical

x axis horizontal



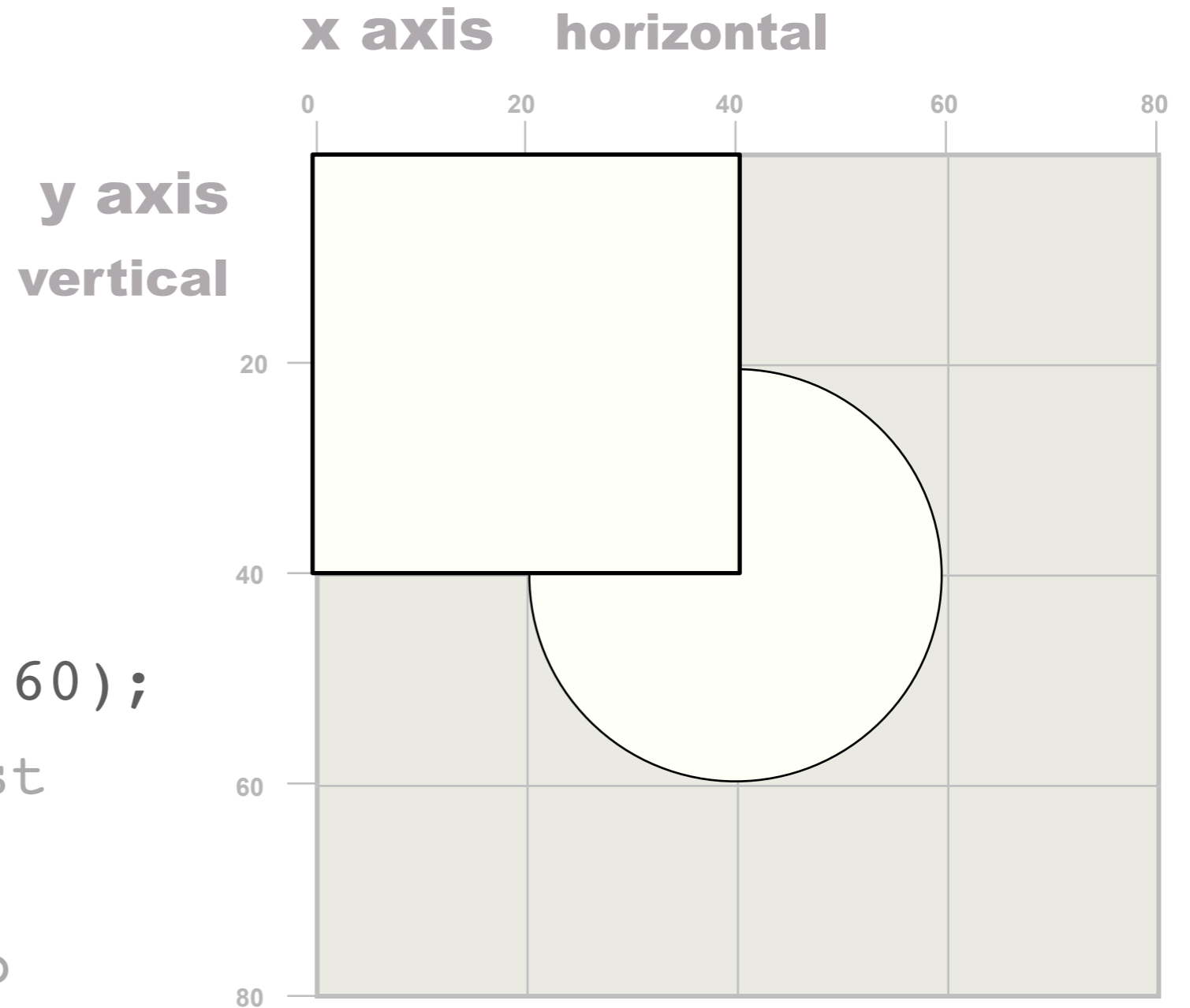
# drawing order

```
size(80,80);  
background(230);  
rect(0, 0, 40, 40);  
// rect drawn first  
ellipse(40, 40, 60, 60);  
// circle drawn on top
```



# drawing order

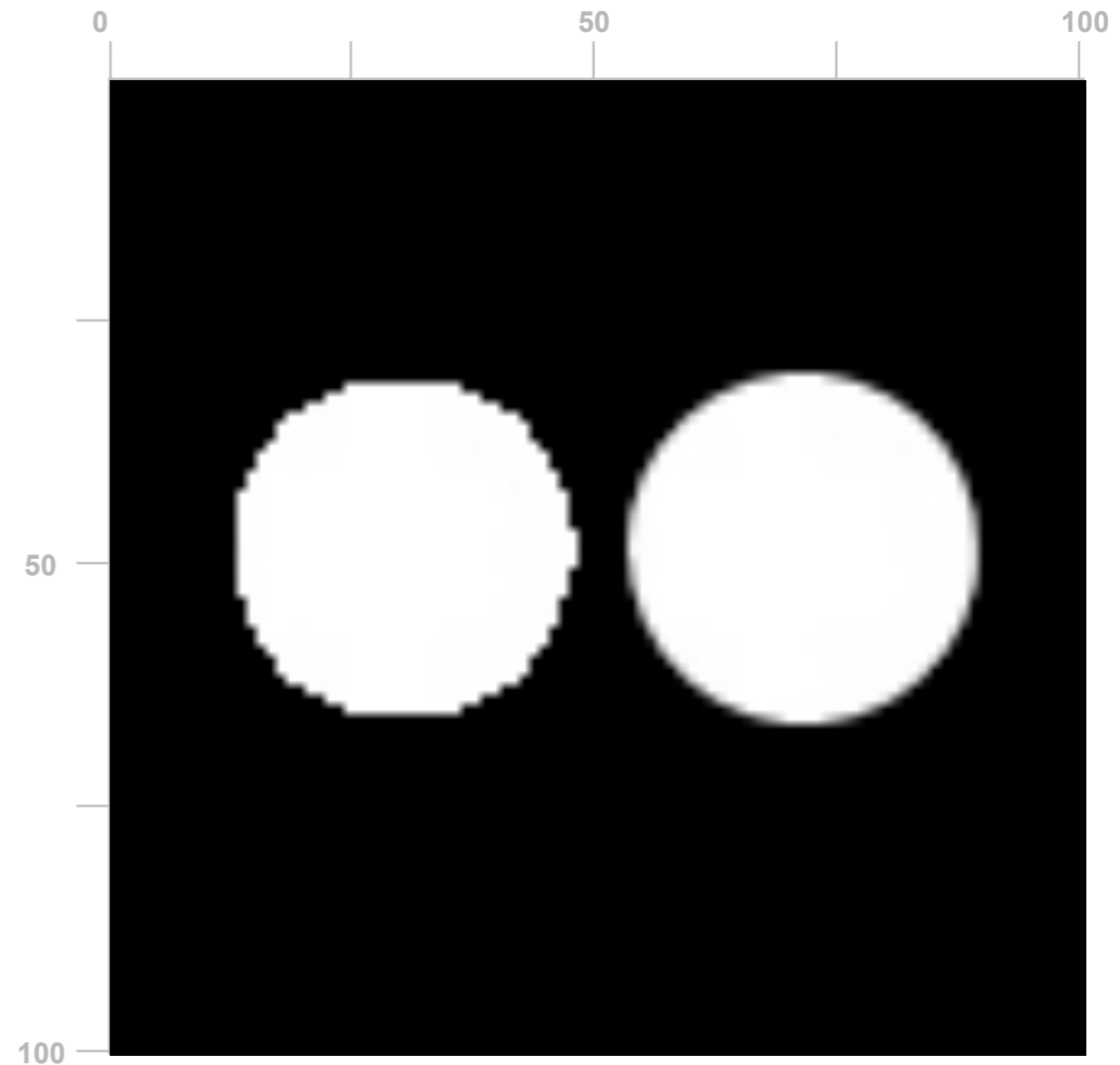
```
size(80,80);  
background(230);  
ellipse(40, 40, 60, 60);  
// circle drawn first  
rect(0, 0, 40, 40);  
// rect drawn on top
```



## smooth

draws all geometry with smooth  
(anti-aliased) edges

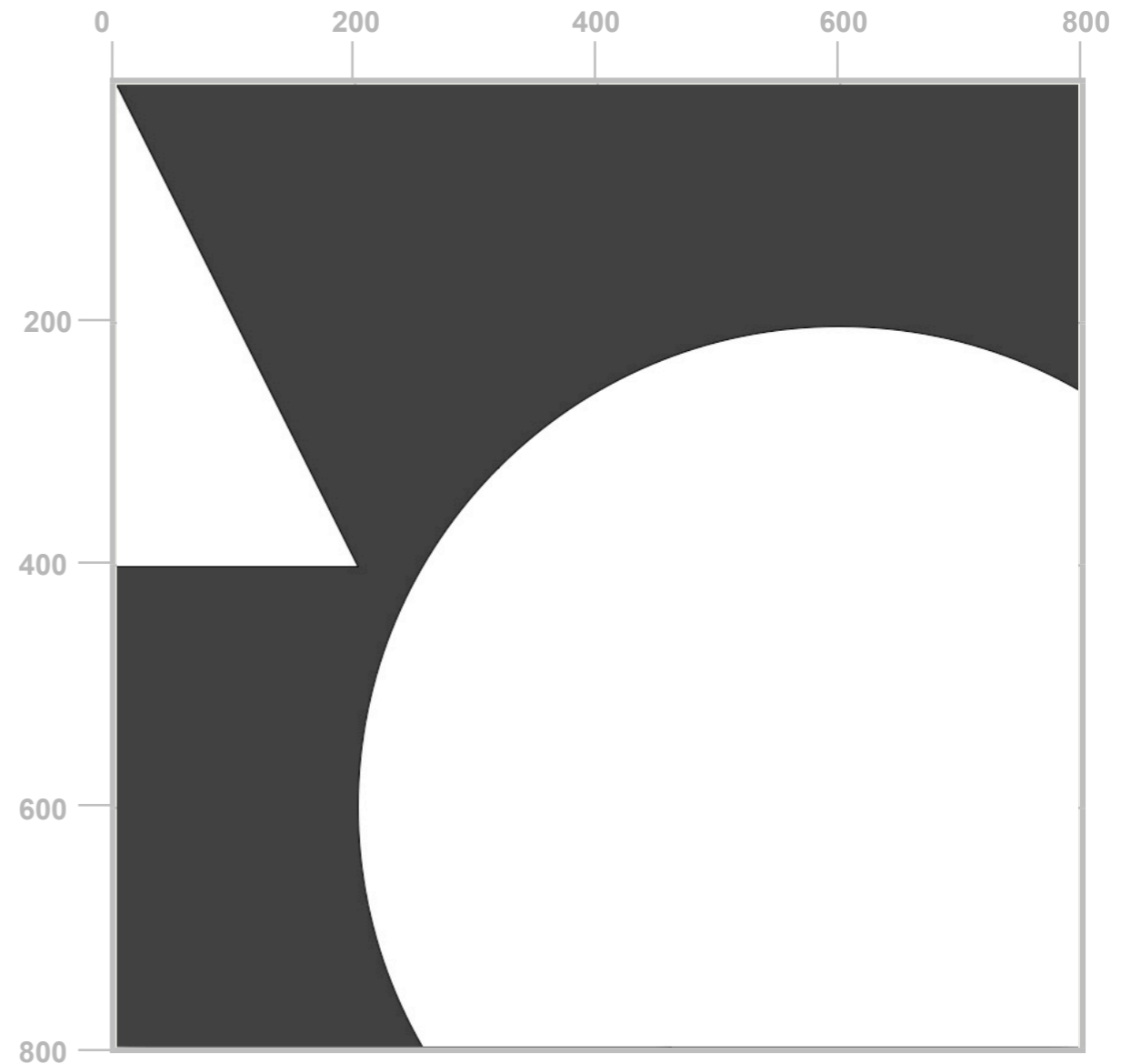
```
background(0);  
ellipse(30, 48, 36, 36);  
smooth();  
ellipse(70, 48, 36, 36);
```



## outside the box

use negative values, centre point to position items outside drawing area

```
size(800,800);  
background(50);  
smooth();  
triangle(0,0,-200,400,200,400);  
ellipse(600, 600, 800, 800);
```



## Background and Fill

using greyscale

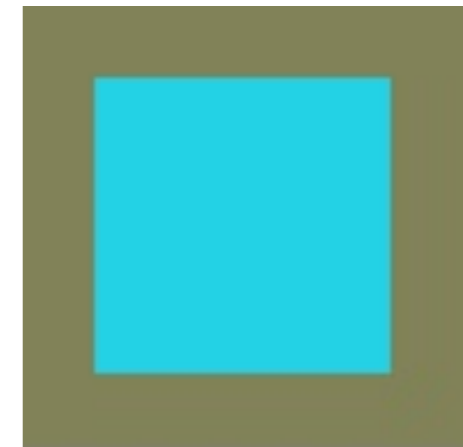
```
size(80,80);  
background(0);  
// 0 black, 255 white  
smooth();  
fill(255, 220);  
// white fill, with transparency  
// 0 totally transparent, 255 all opaque  
rect(0, 0, 40, 40);  
ellipse(40, 40, 60, 60);
```



## Background and Fill

using colour / RGB and HEX values

```
background(129, 130, 87);  
noStroke();  
fill(#23D3E5);  
rect(17, 17, 66, 66);
```



# Colour Selector

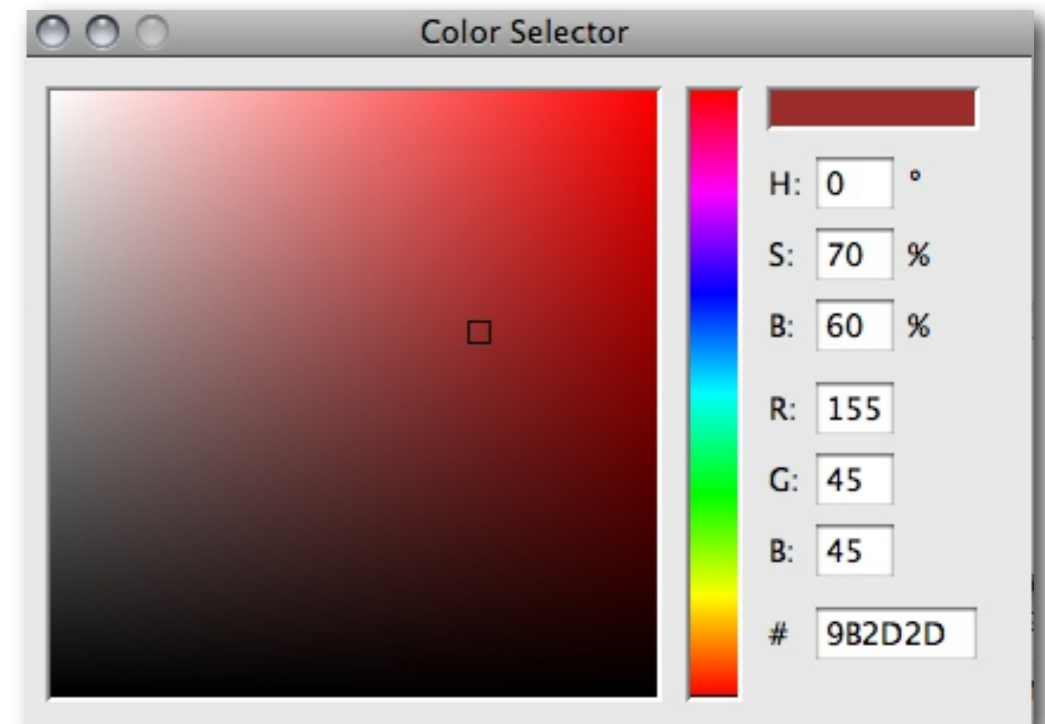
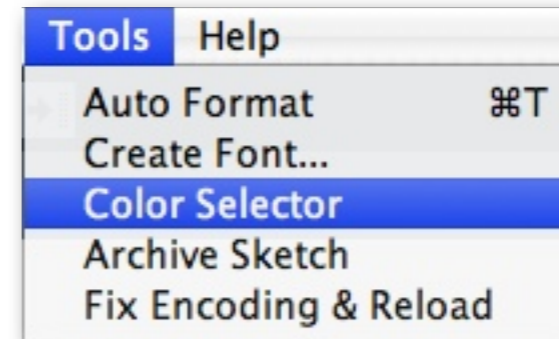
menu: Tools/Color Selector

RGB (red, green, blue,)

```
background(155, 45, 45);
```

hexidecimal (6 characters A-F 0-9)

```
background(#9B2D2D);
```





## Stroke Weight

```
smooth();  
line(20, 20, 80, 20);  
// default line weight of 1  
  
strokeWeight(6);  
line(20, 40, 80, 40);  
// line weight of 6, round ends default  
  
strokeWeight(16);  
strokeCap(SQUARE);  
line(20, 80, 80, 80);  
// line weight of 16, square ends
```



## Stroke Colour

To color a line, use the **stroke()**

```
smooth();  
stroke(126);  
line(20, 20, 80, 20);  
// greyscale stroke  
  
strokeWeight(6);  
stroke(129, 130, 87);  
line(20, 40, 80, 40);  
// RGB value stroke  
  
strokeWeight(16);  
stroke(#4E1A04);  
strokeCap(SQUARE);  
line(20, 80, 80, 80);  
// HEX value stroke
```

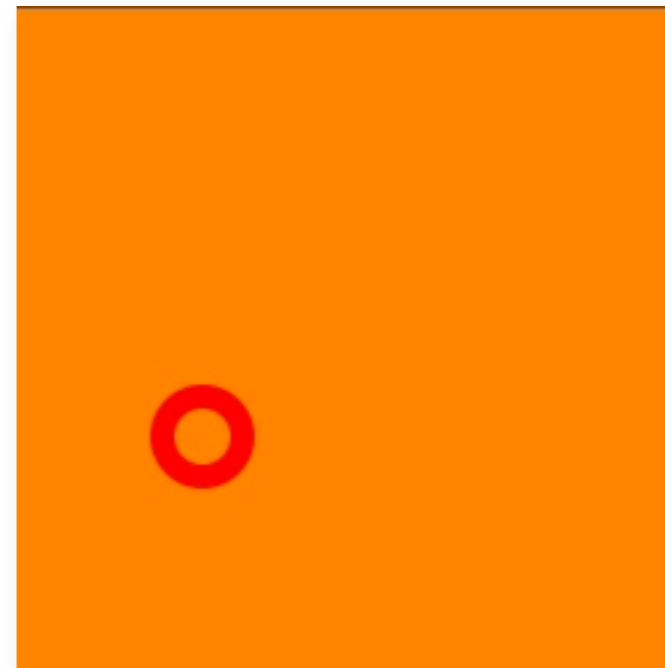


## Save as an image

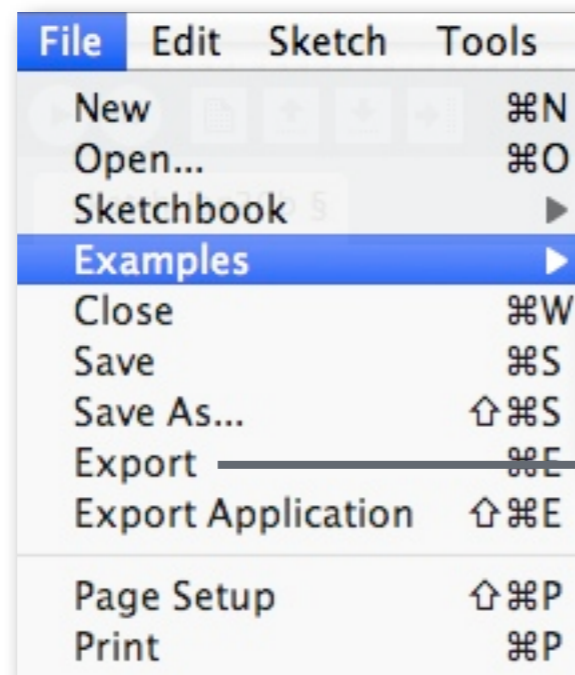
To save your sketch as an image, use `save()`  
Save your PDE file first.

```
//document size
size (250,250);
background (255,132,15);

//circle
noFill ();
stroke (255,0,0);
strokeWeight (9);
smooth ();
ellipse (70,160,30,30);
save ("image.png");
```



# File



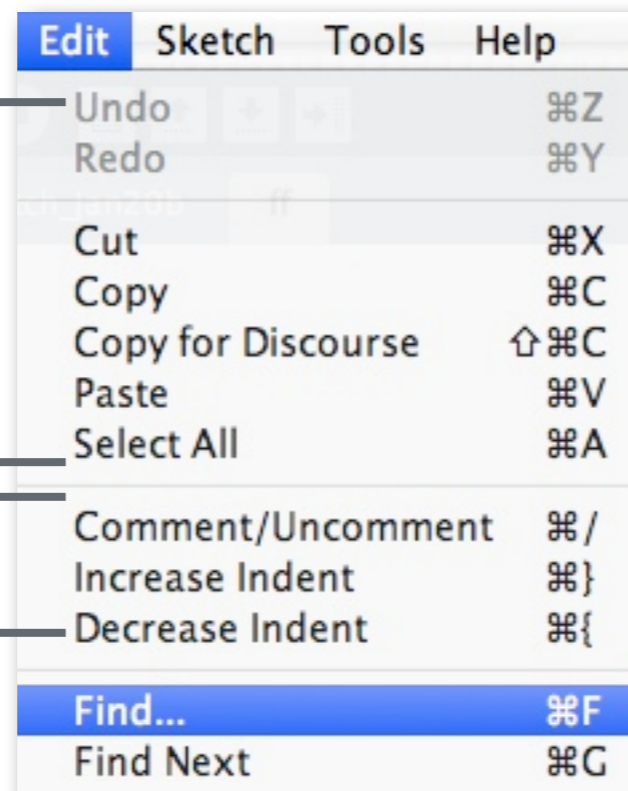
— **Various sample files**

— **Export for Web**

# Edit

Standard text editor controls

Add comments, format code by indenting



Find and replace multiple instances of text in code