

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?

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

Context

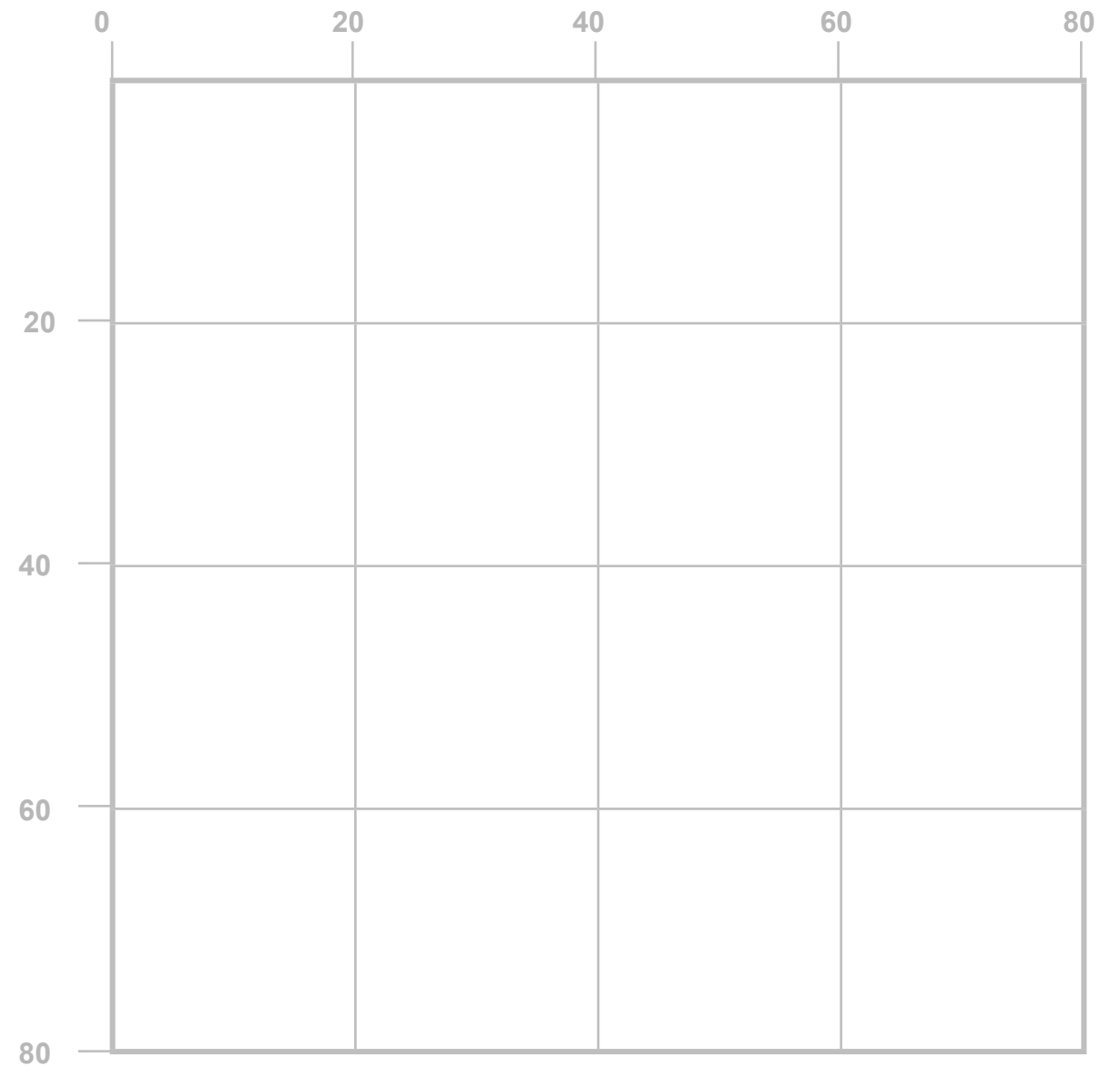
- core language and libraries make use of Java, with elements same as C programming
- easily integrated with other media (audio and video) and devices (PC, mobile phones)
- can be as short as one line of code or as long as thousands

size

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

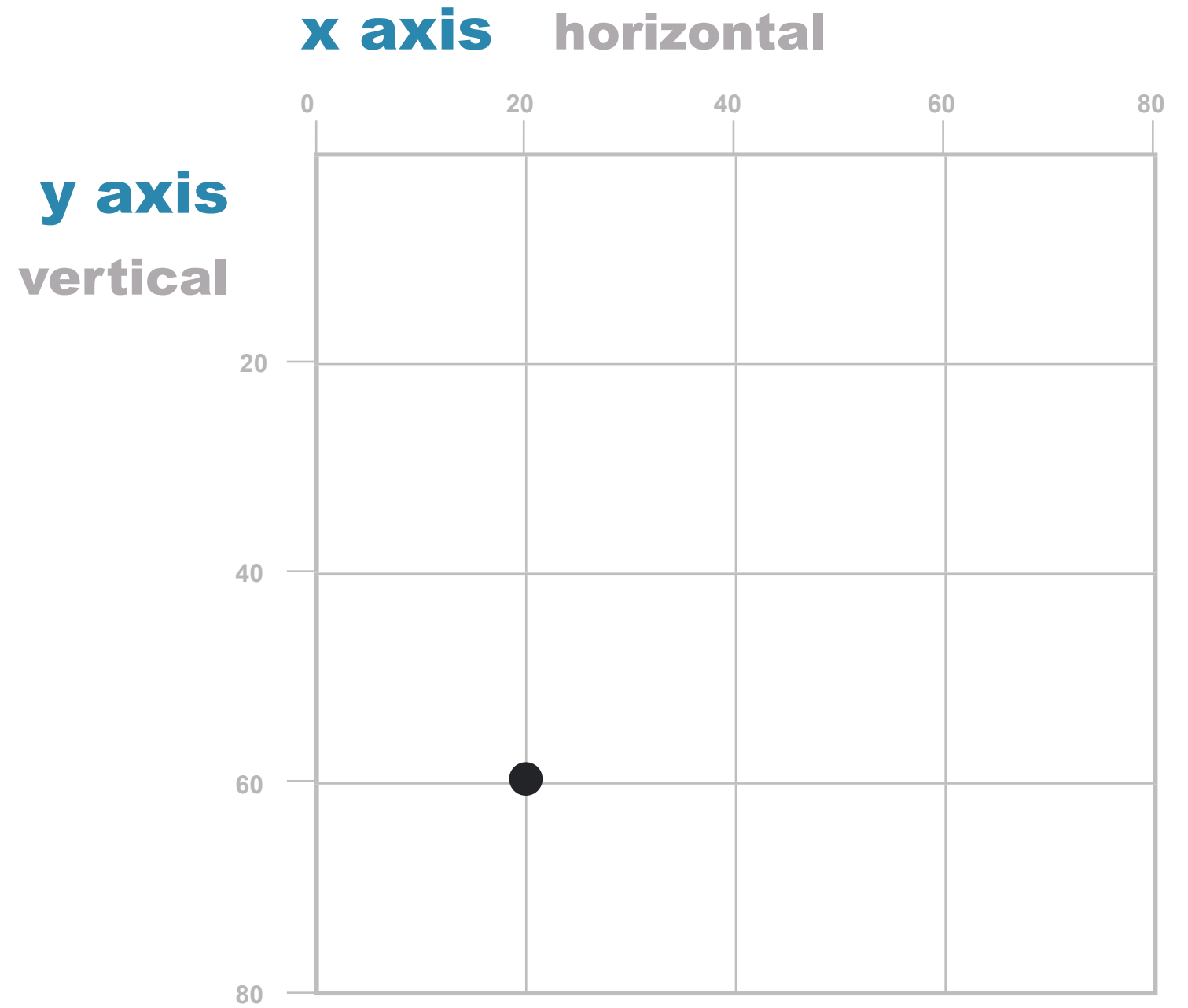
y axis
vertical

x axis horizontal



point

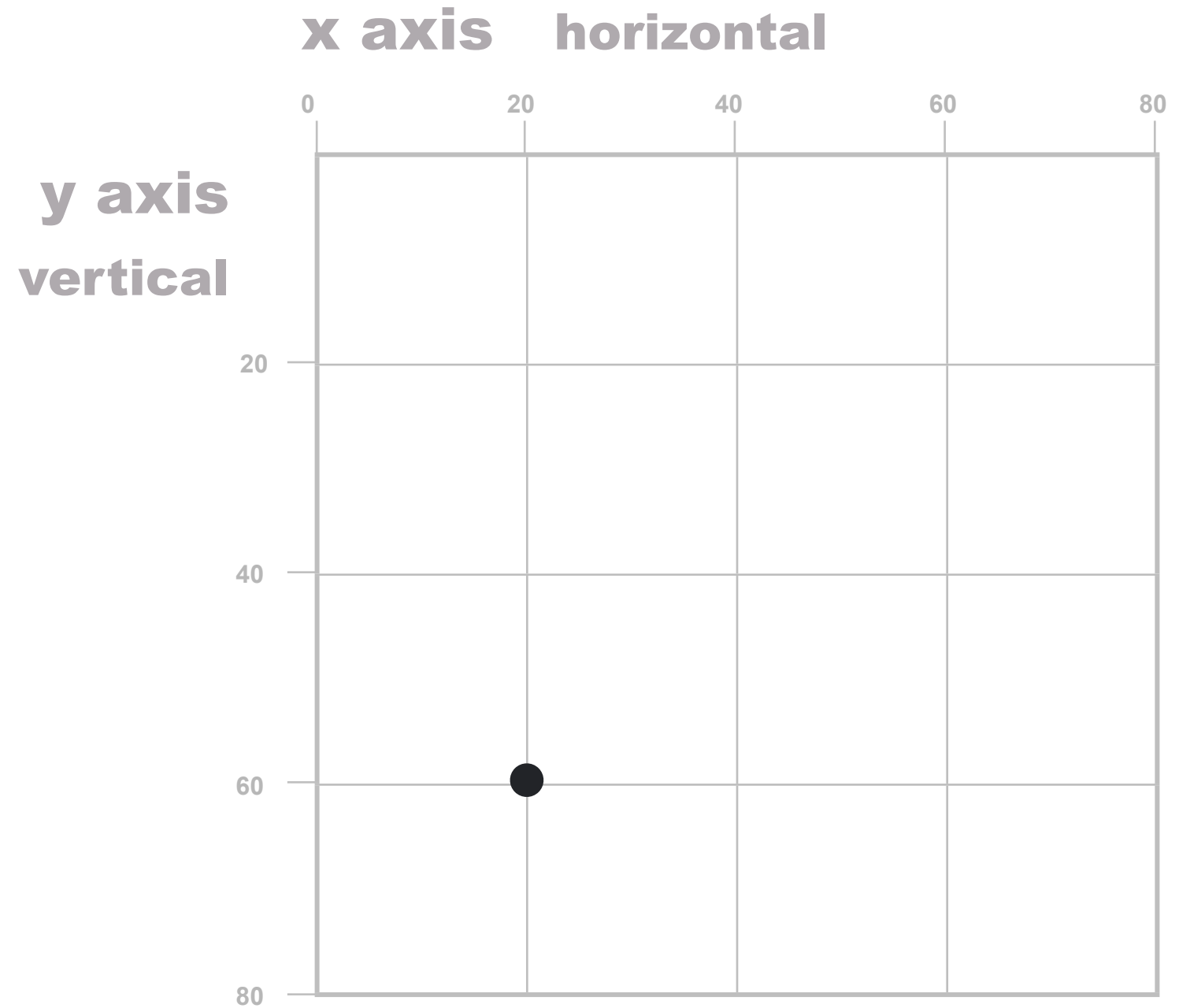
```
      x    y  
point(20, 60);
```



point

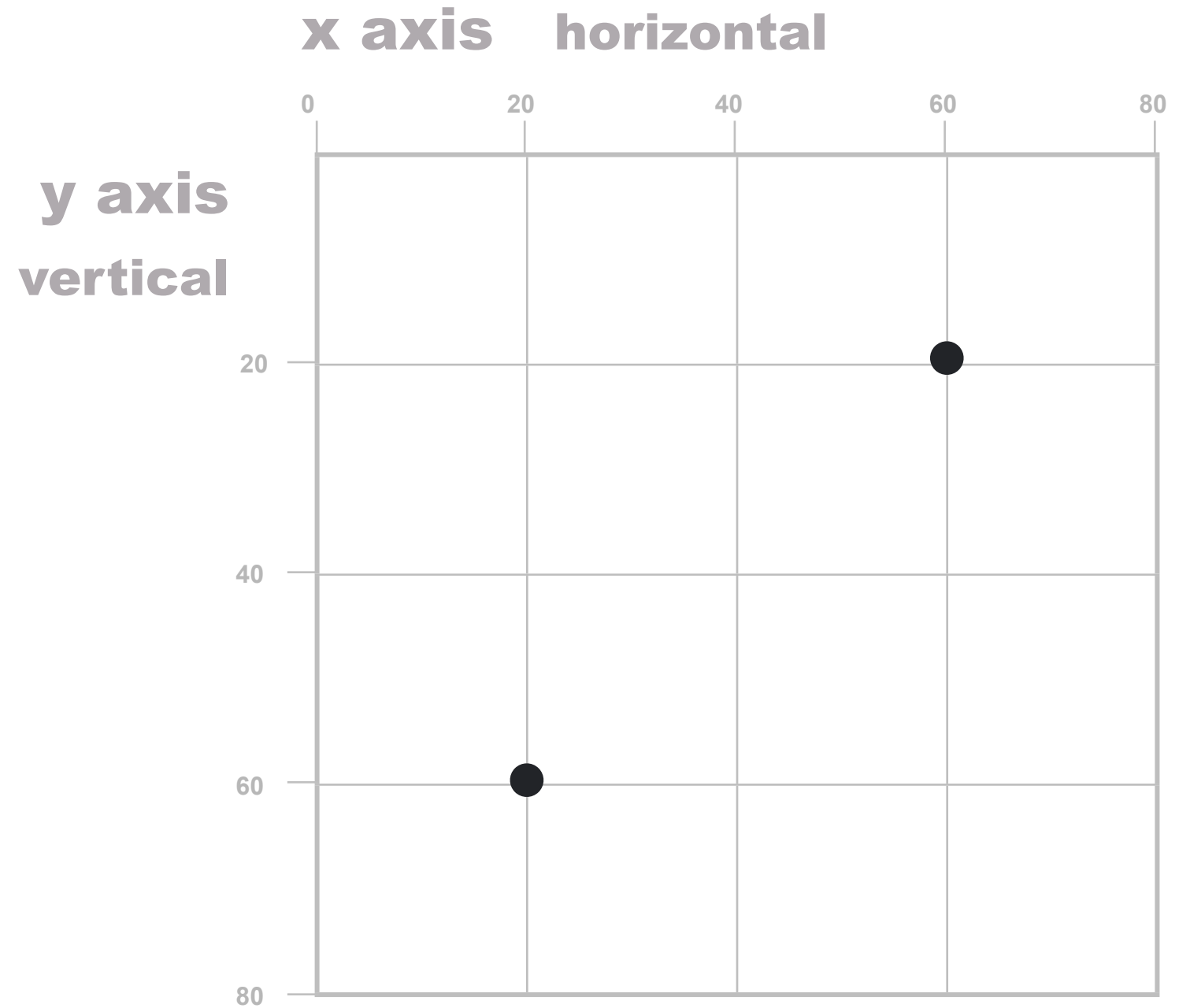
```
point(x, y);
```

Put a point | over here | .



point

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

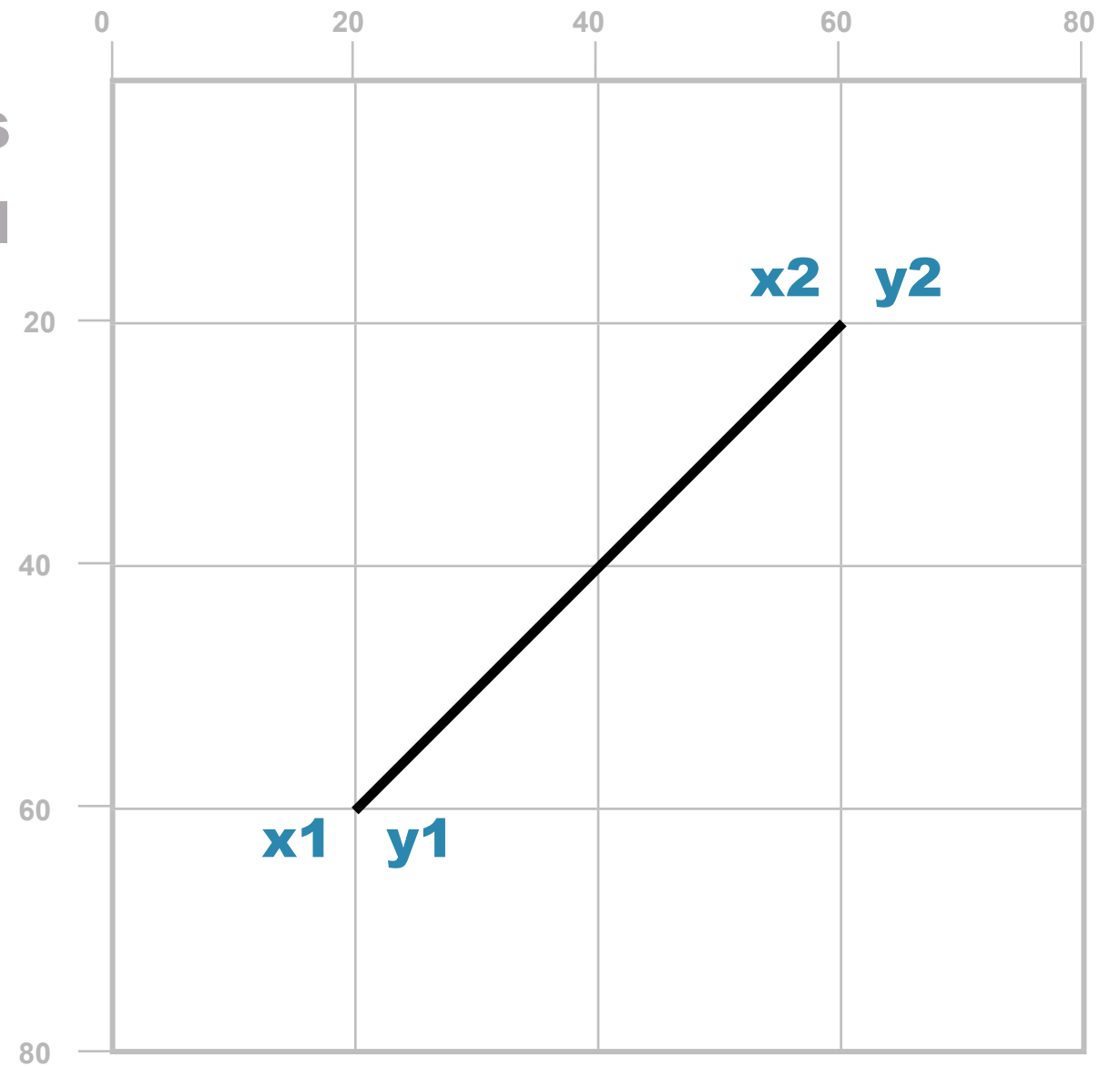


line

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

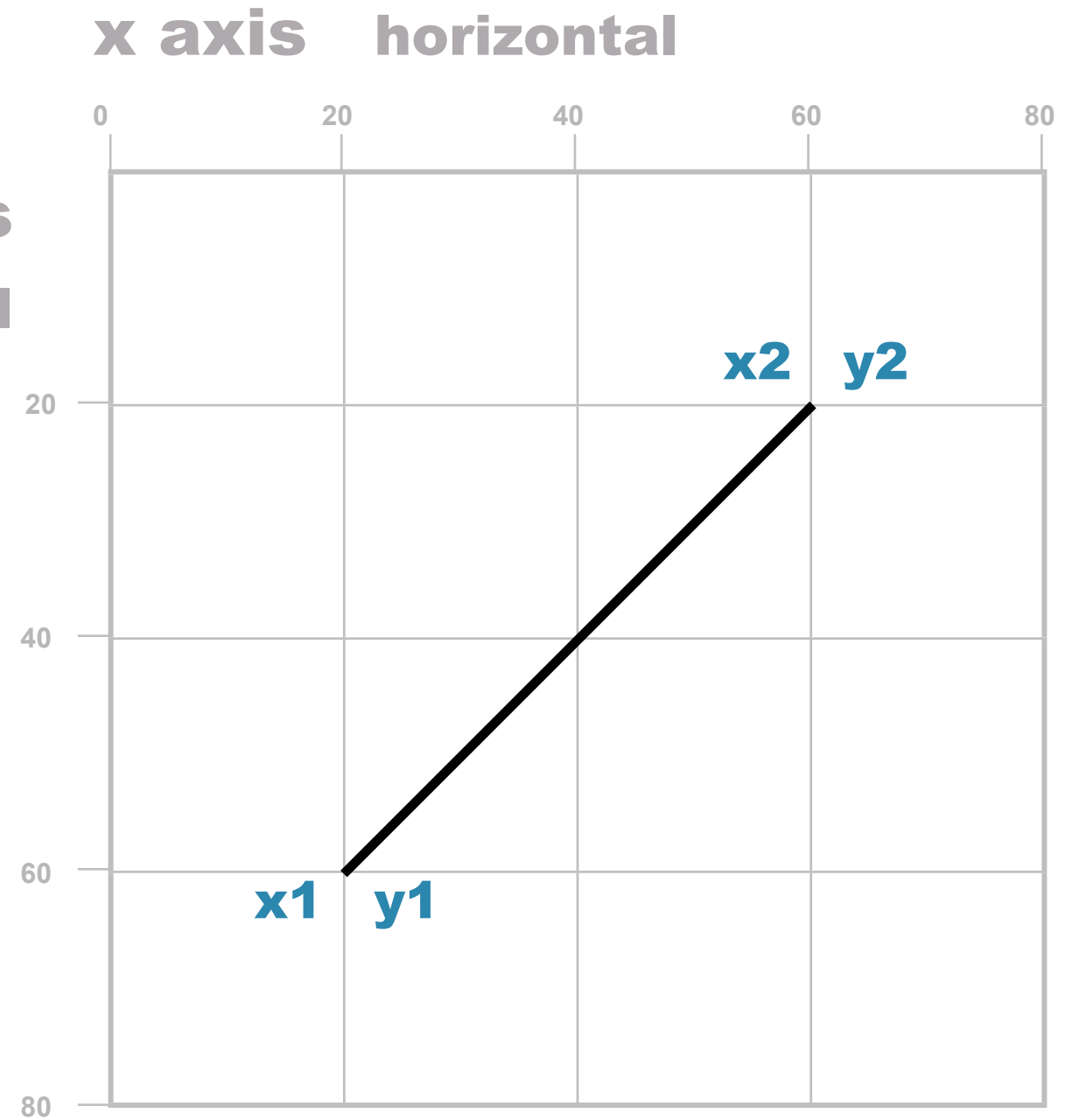
y axis
vertical

x axis **horizontal**



line

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

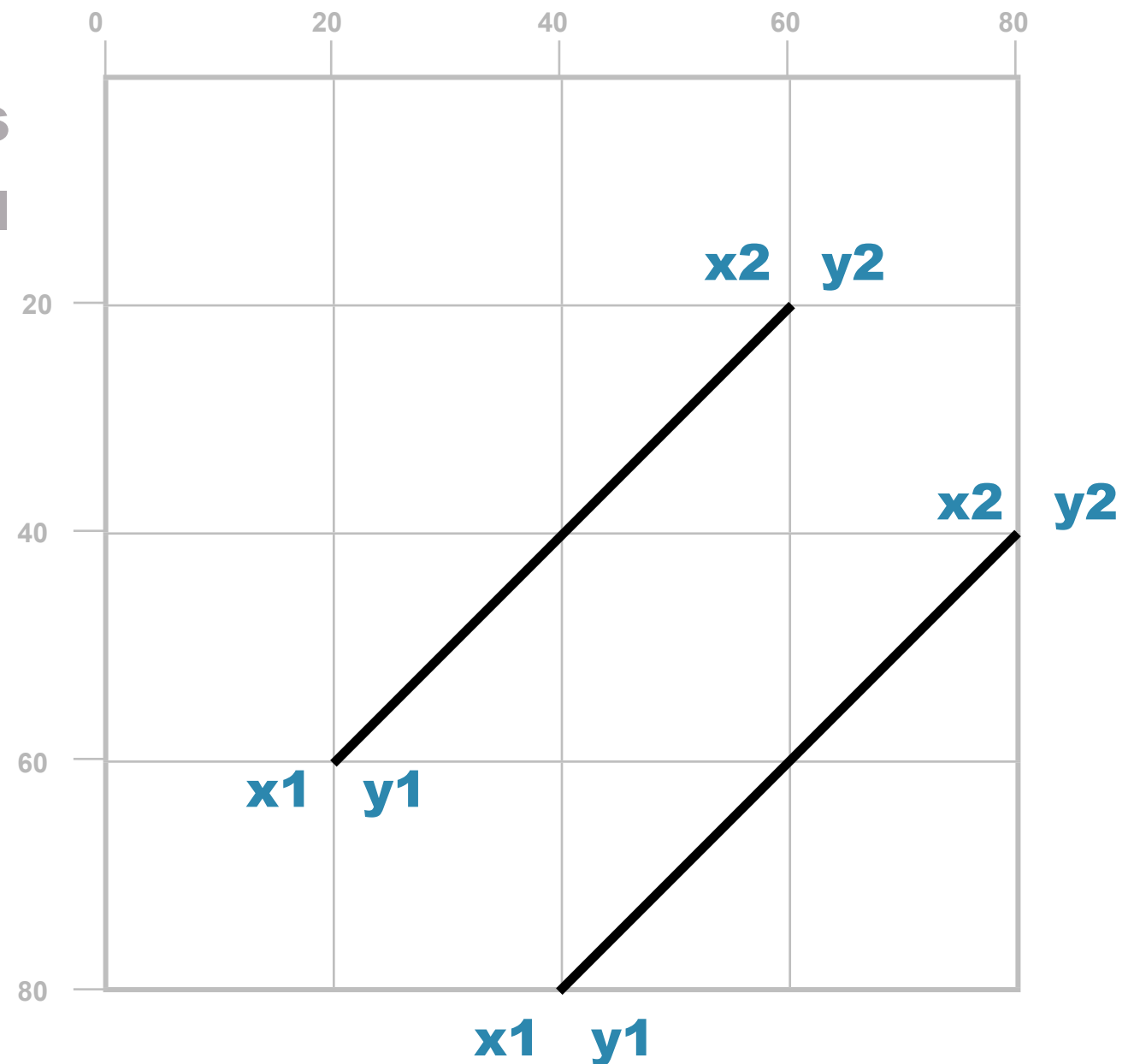


line

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

y axis
vertical

x axis horizontal



rectangle

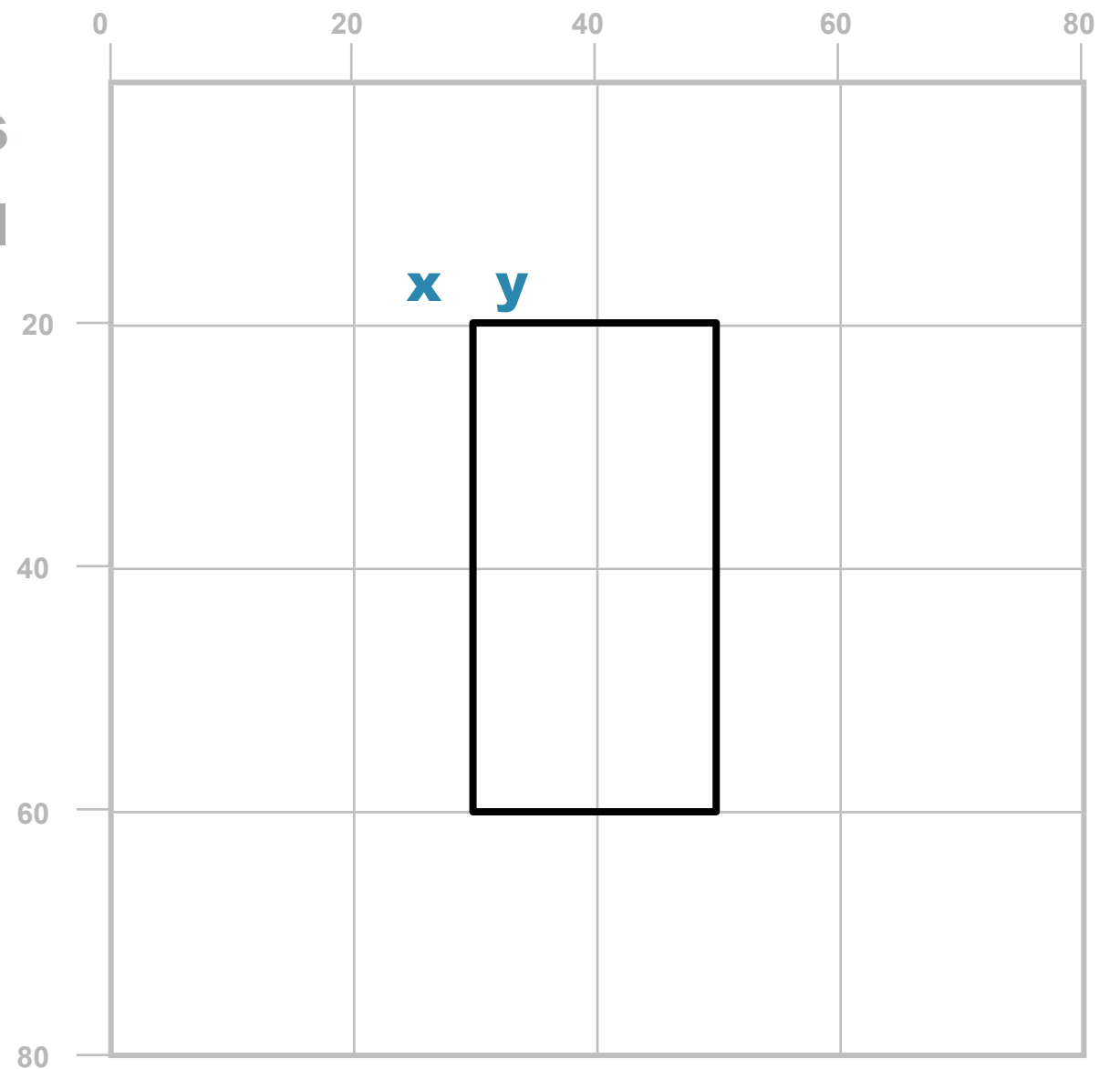
default, drawn from top left

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

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

y axis
vertical

x axis **horizontal**



rectangle

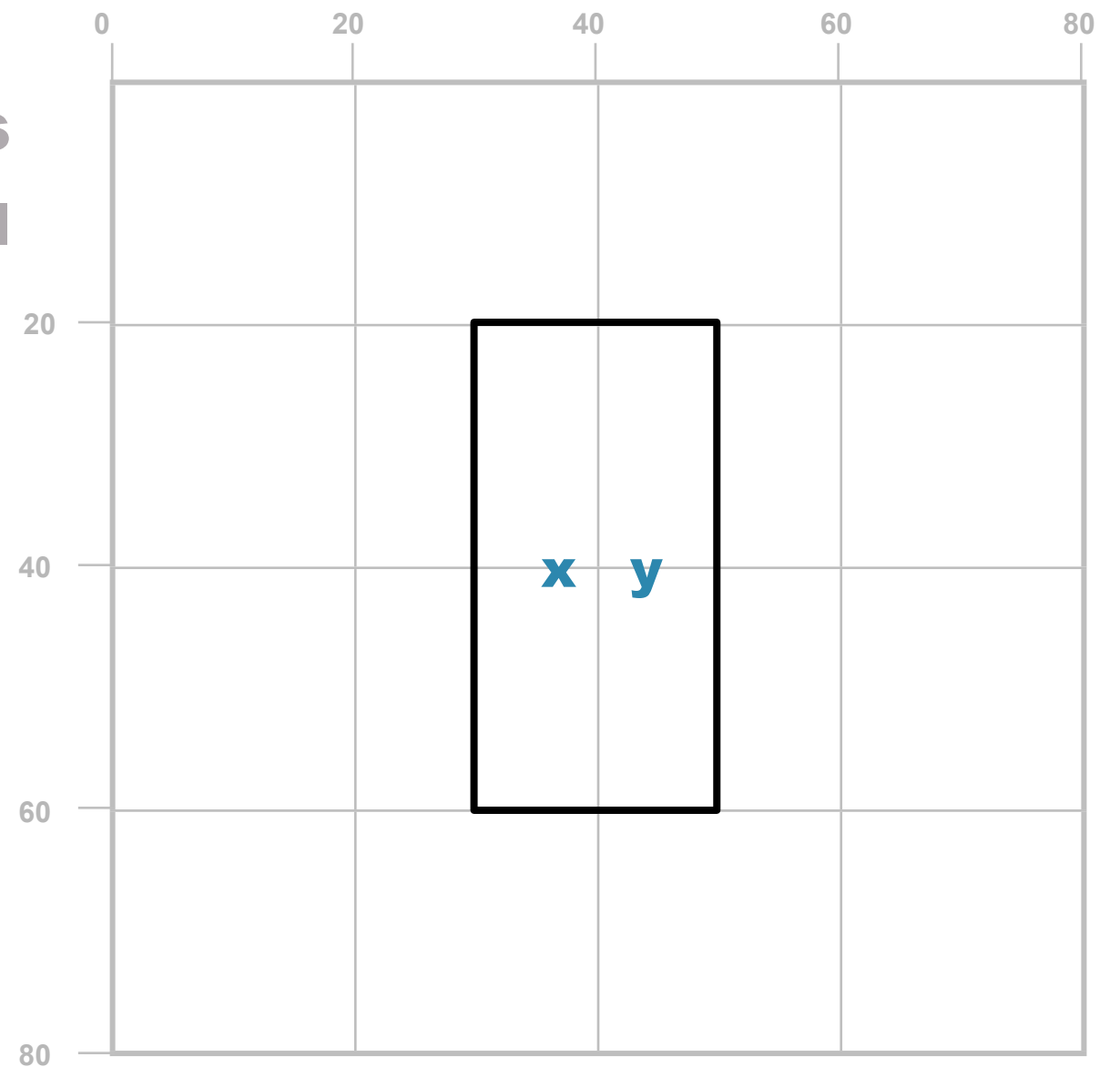
drawn from centre point

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

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

y axis
vertical

x axis **horizontal**



rectangle

drawn from corners

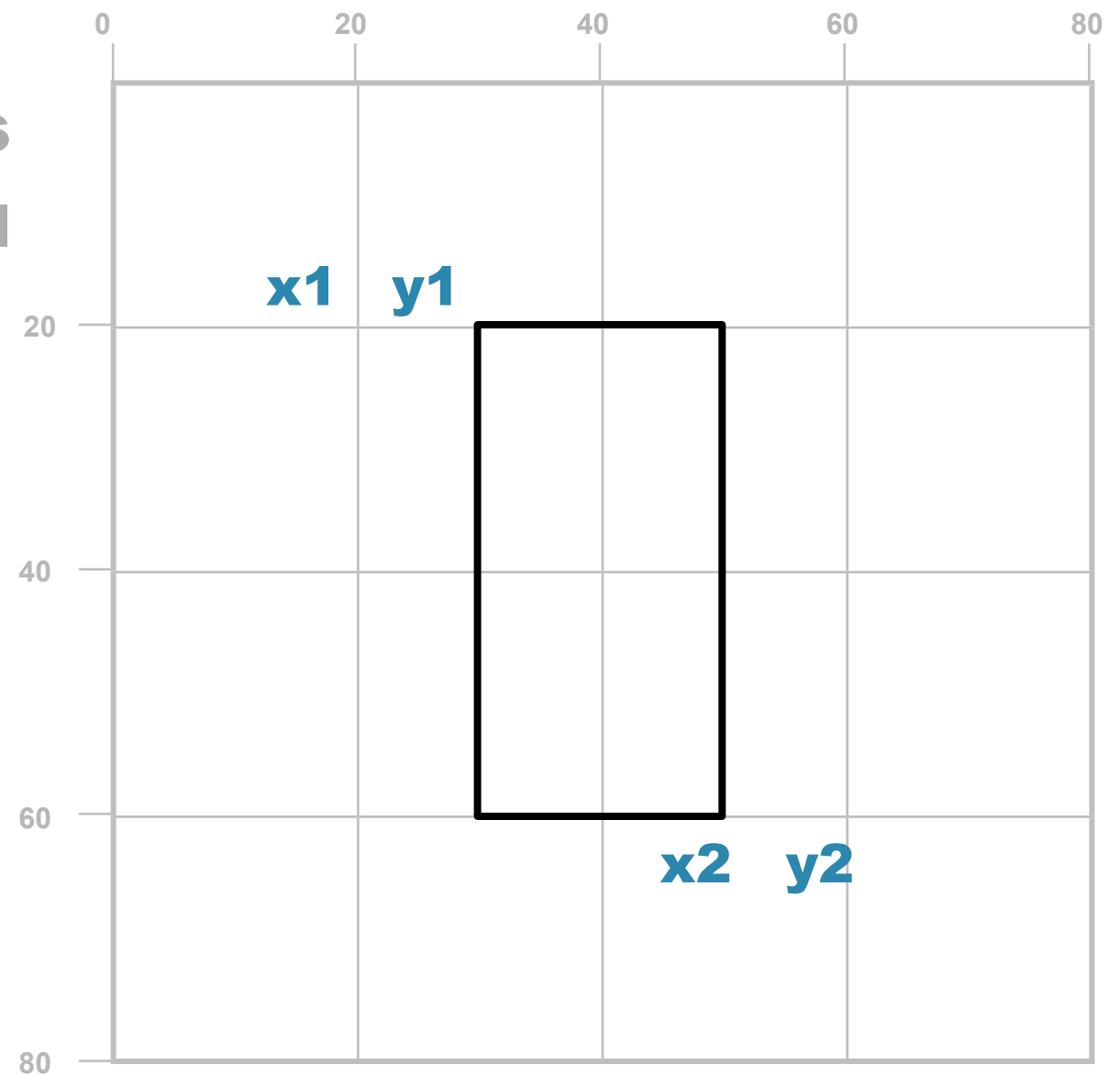
top left, bottom right

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

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

y axis
vertical

x axis **horizontal**



ellipse

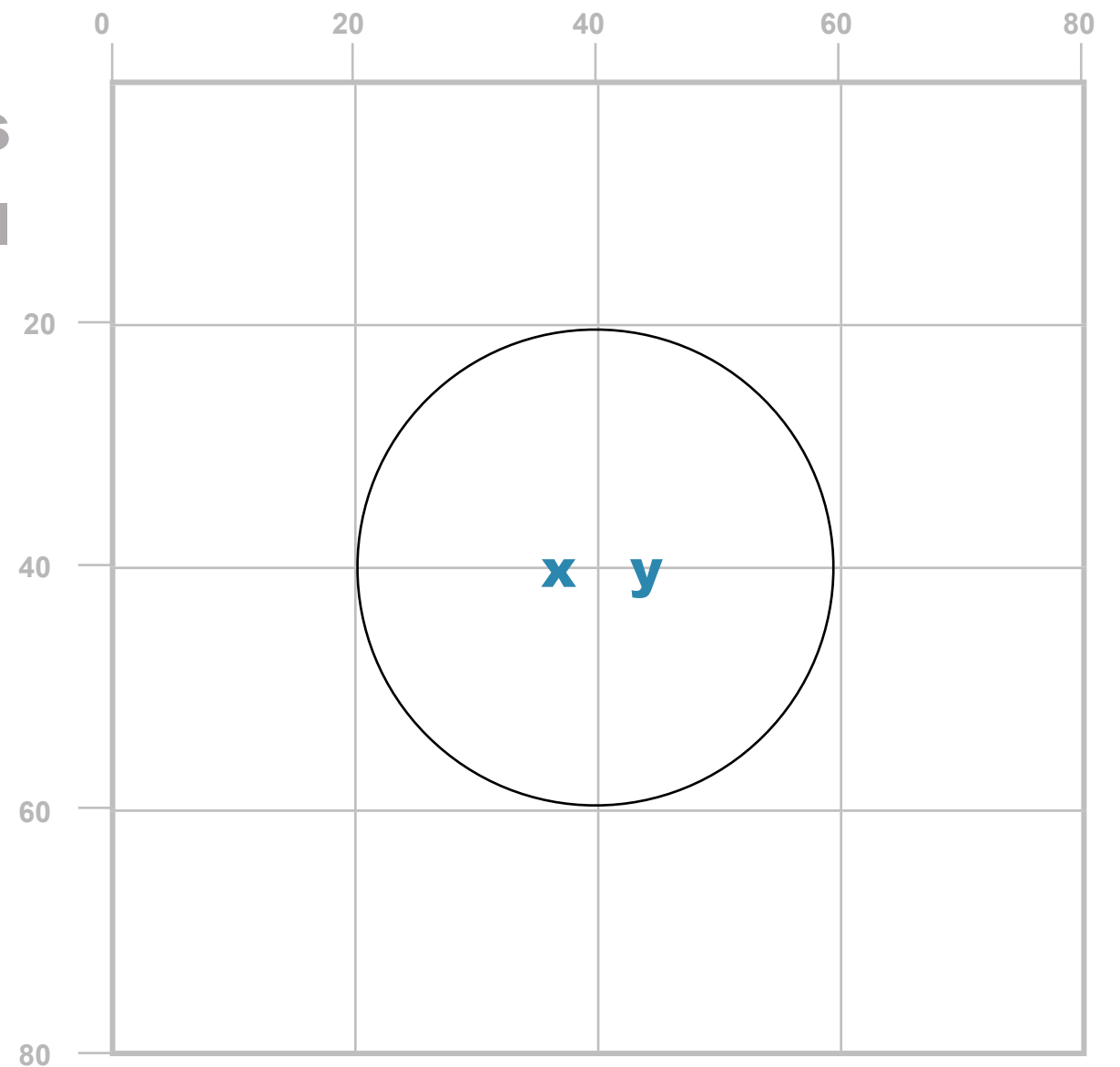
default drawn centre

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

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

y axis
vertical

x axis **horizontal**



ellipse

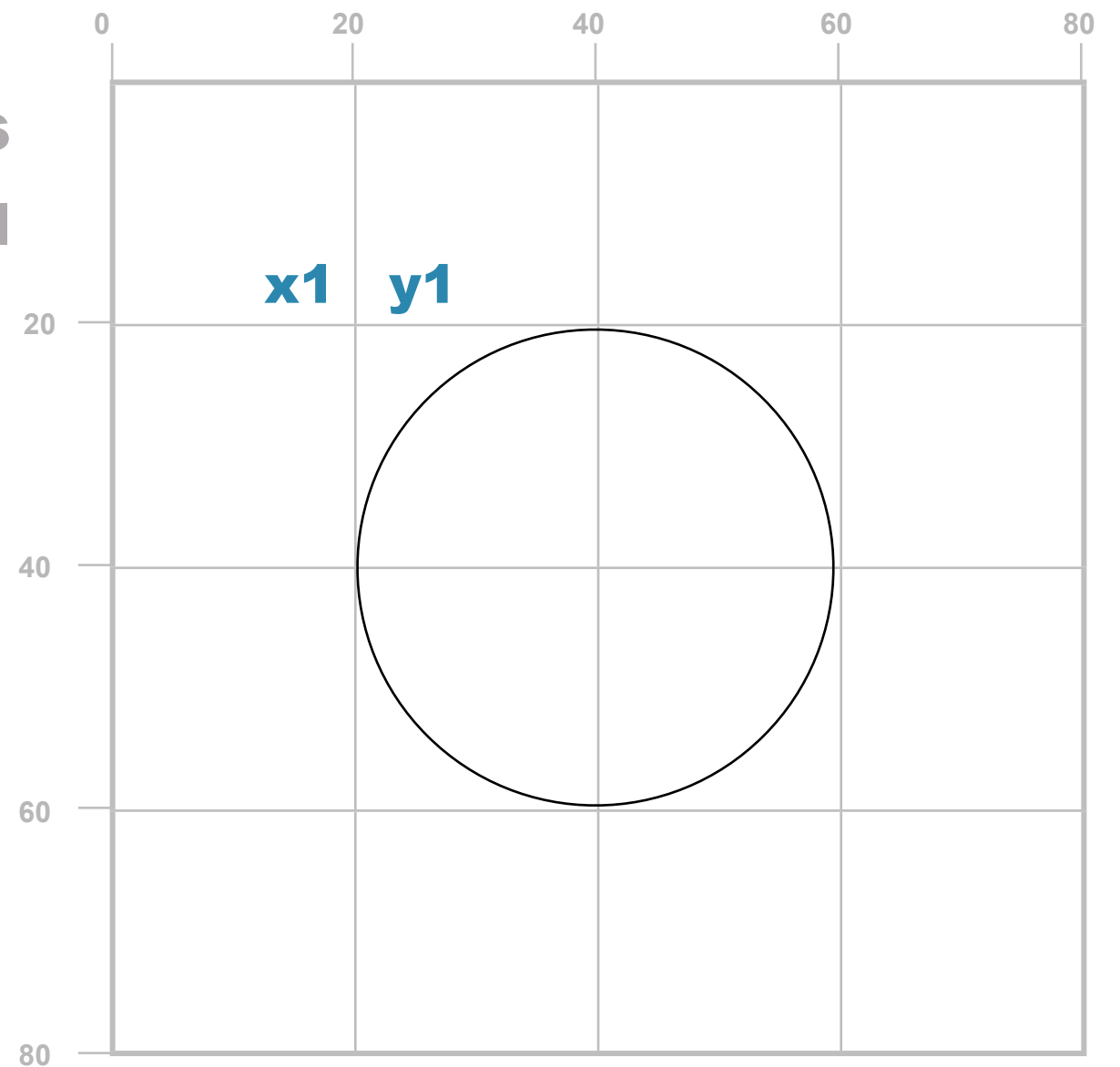
drawn top left

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

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

y axis
vertical

x axis **horizontal**



ellipse

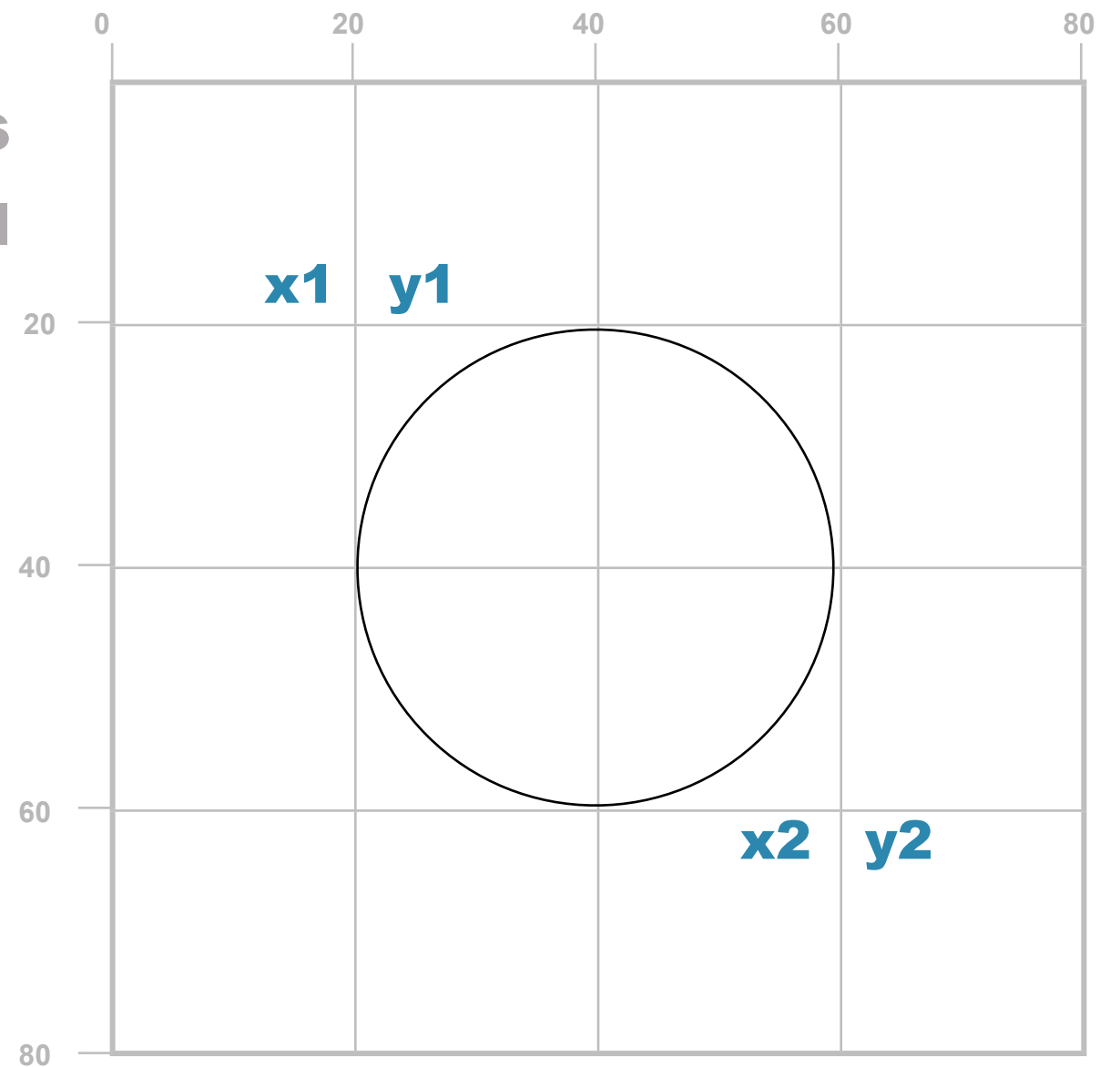
drawn top left

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

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

y axis
vertical

x axis **horizontal**



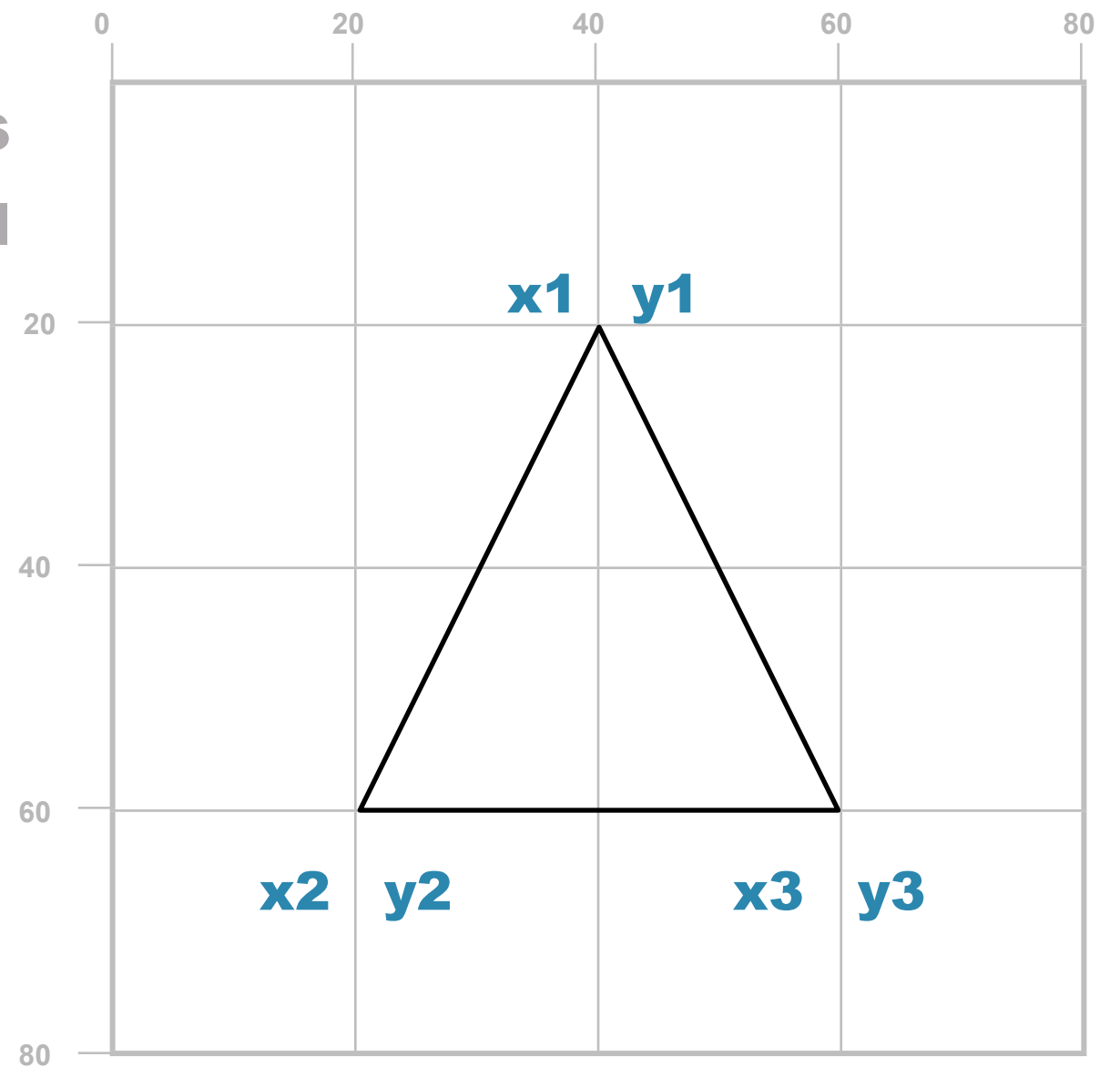
triangle

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



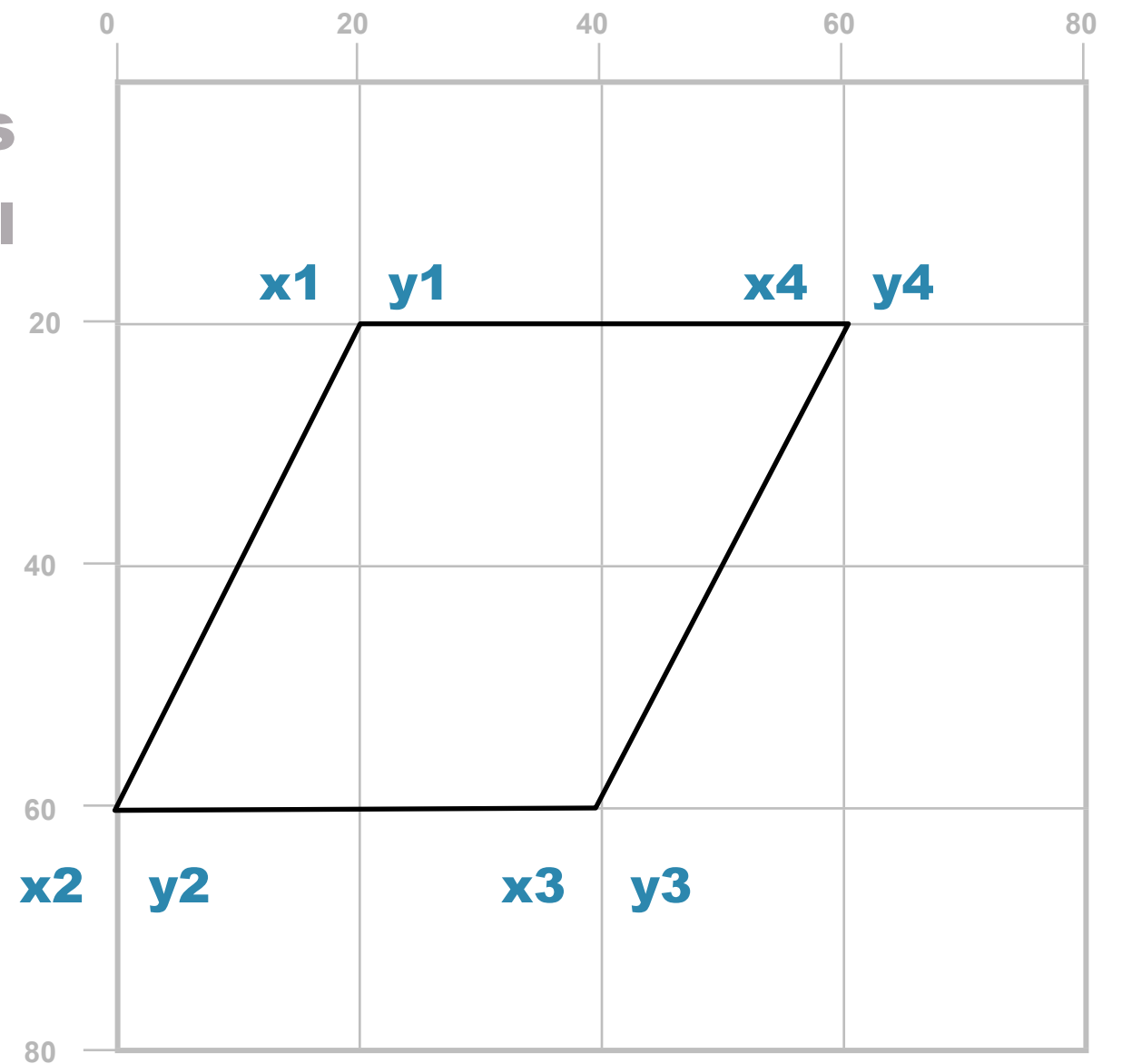
quad

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,
```

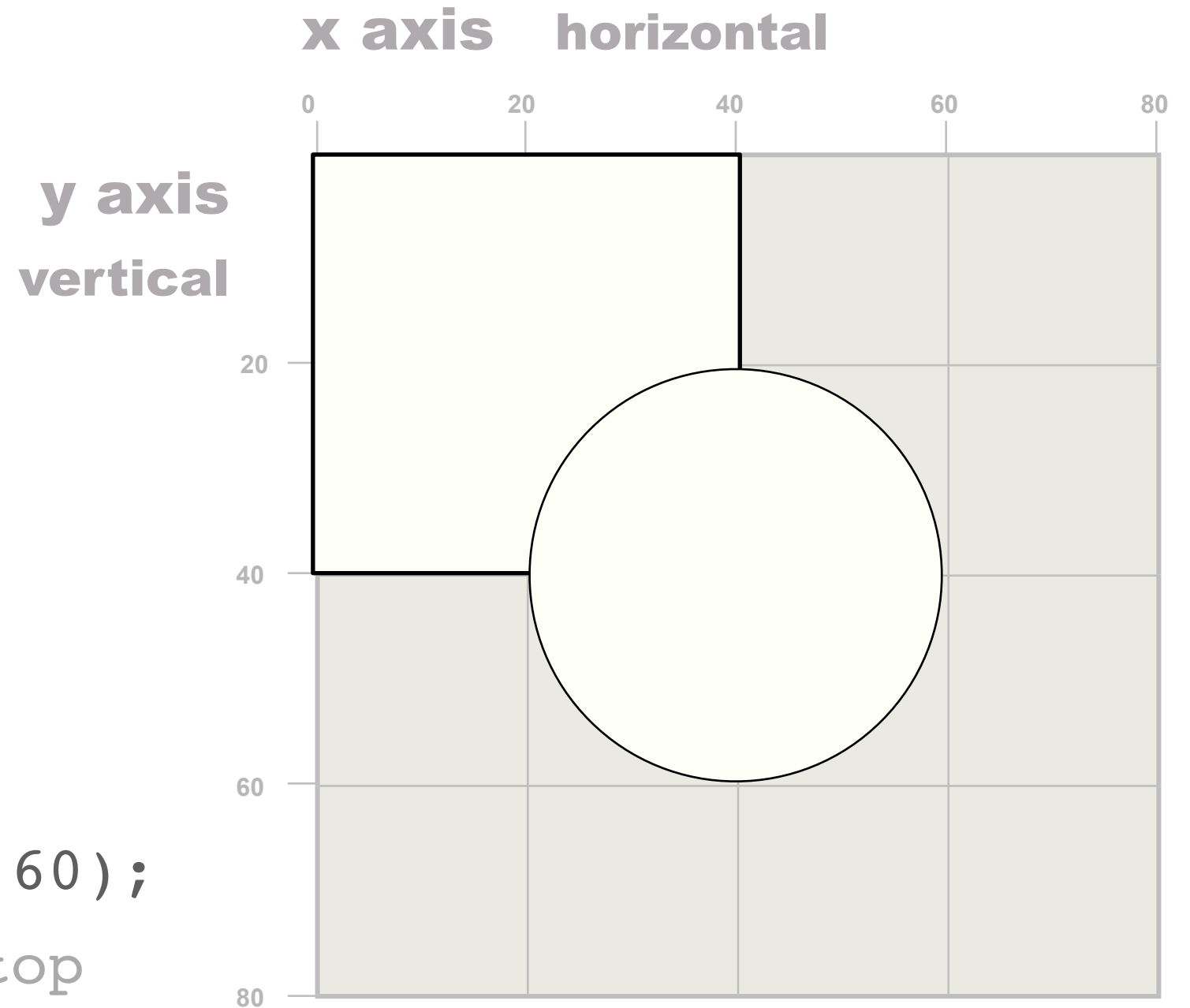
y axis
vertical

x axis horizontal



drawing order

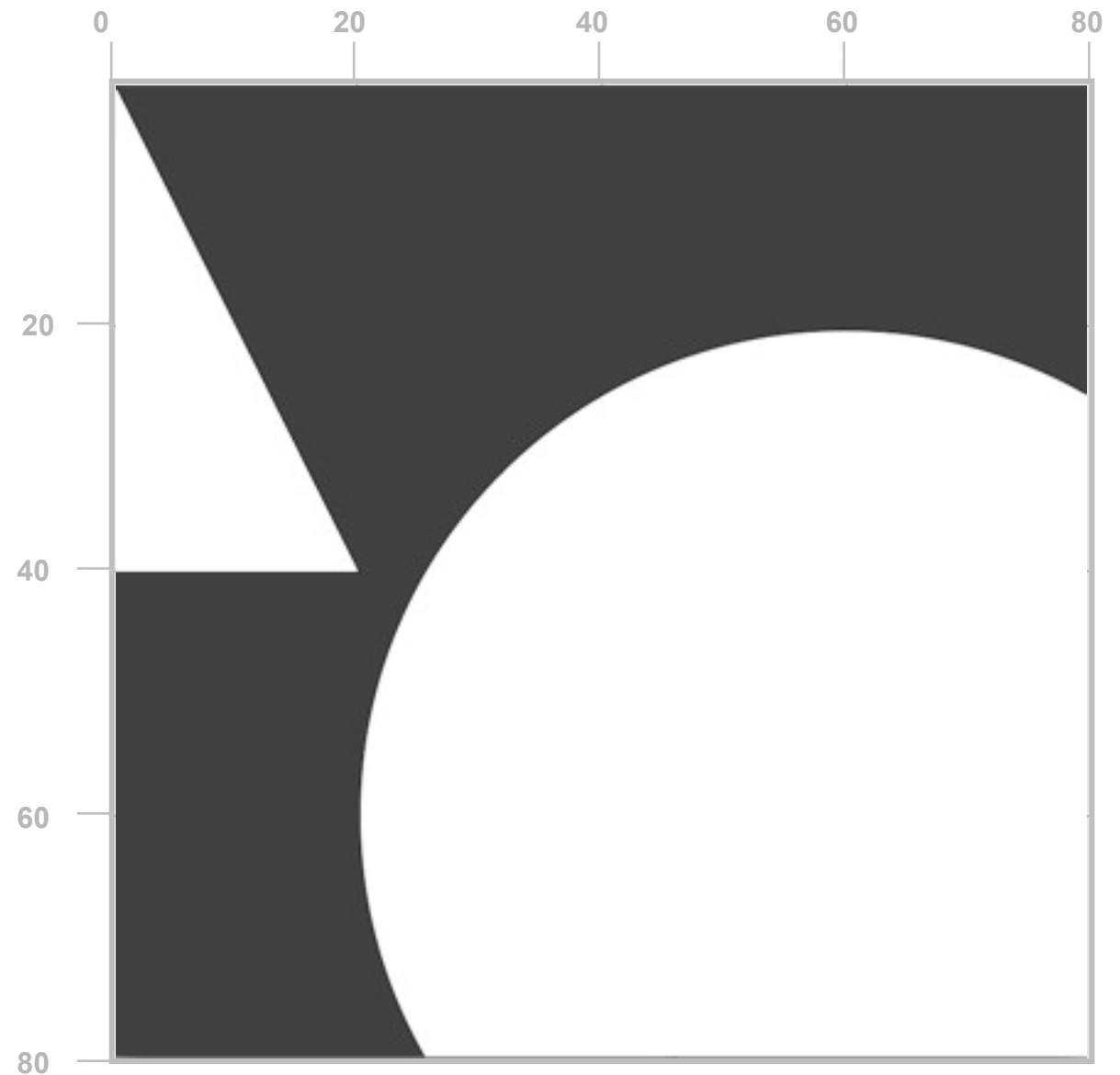
```
size(80,80);  
background(230);  
rect(0, 0, 40, 40);  
// rect drawn first  
ellipse(40, 40, 60, 60);  
// circle drawn on top
```



outside the box

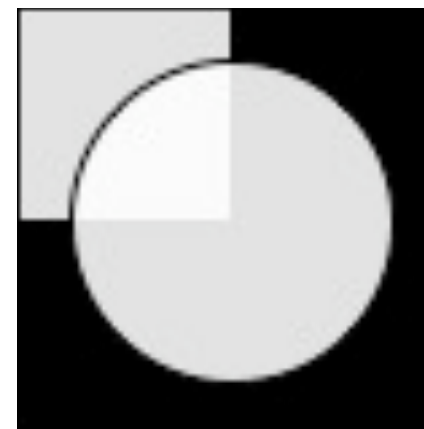
use negative values, centre point to position items outside drawing area

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



Background and Fill

```
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);
```

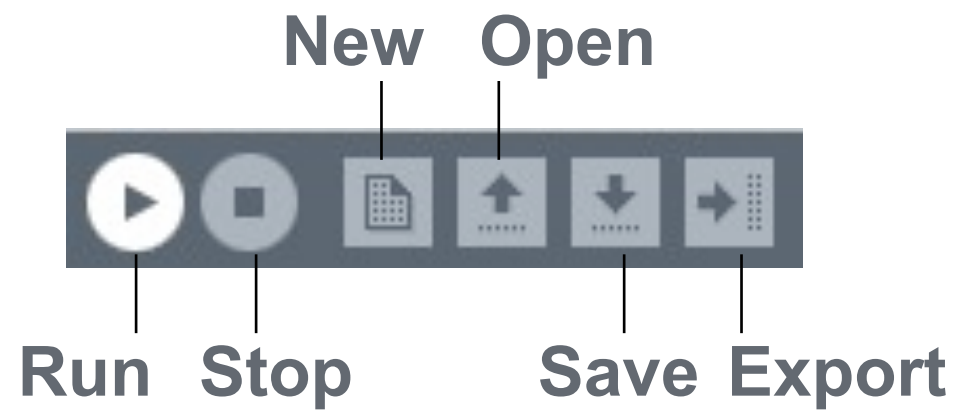


Stroke Weight

```
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
```



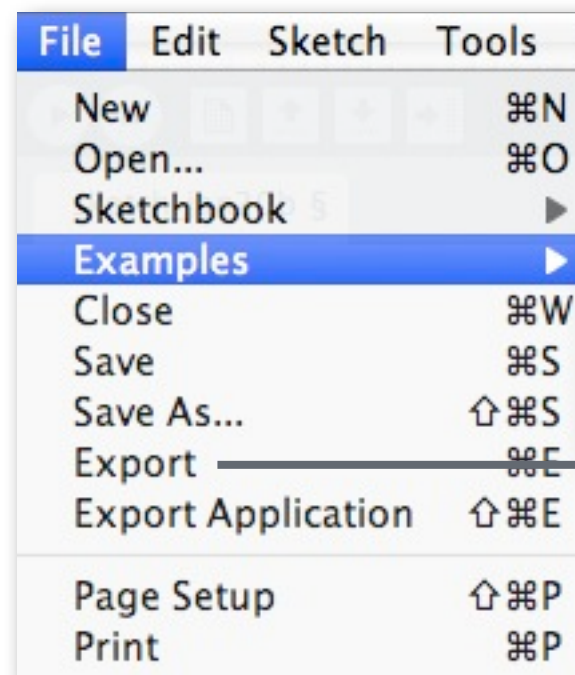
Interface



similar to preview in browser

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

File

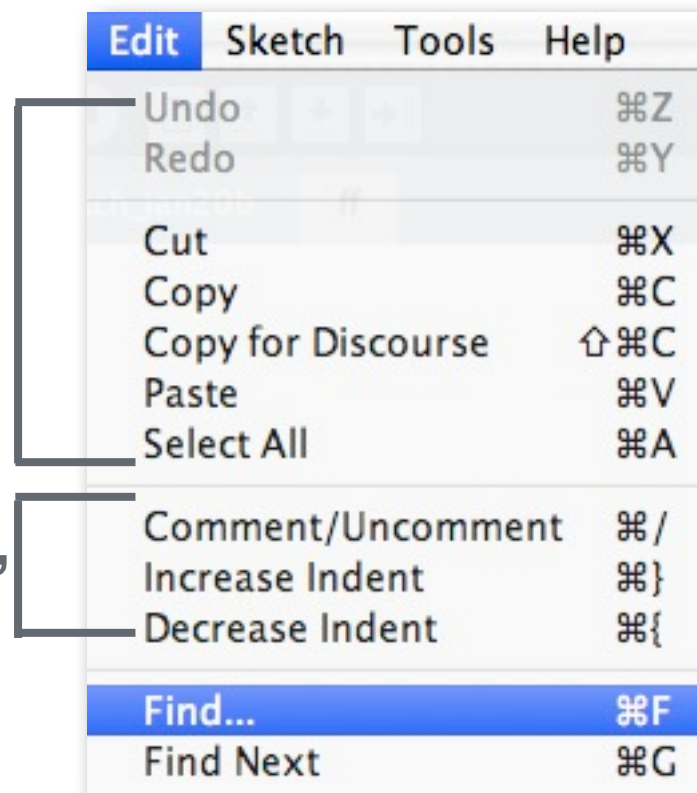


— Various sample files

— Export for Web

Edit

Standard text editor controls



Add comments, format code by indenting

Find and replace multiple instances of text in code

Export

applet folder



applet
5 items

