



# Programming Fundamentals 1

---

Produced by Mr. Dave Drohan ([david.drohan@setu.ie](mailto:david.drohan@setu.ie))  
Dr. Siobhán Drohan  
Ms. Mairead Meagher

Department of Computing & Mathematics  
South East Technological University  
Waterford, Ireland

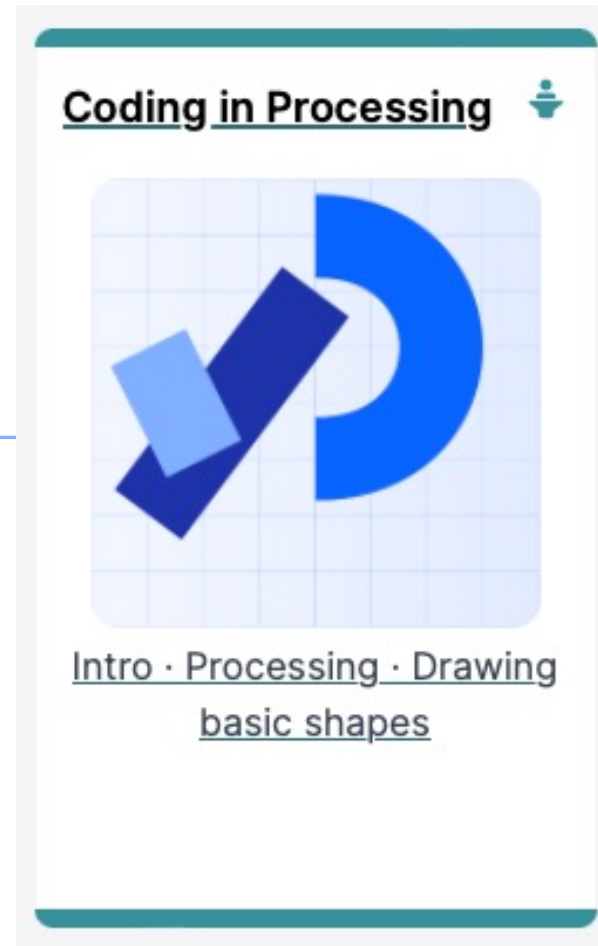
setu.ie





# Introduction to Processing

Starting to Code in Processing





# Agenda

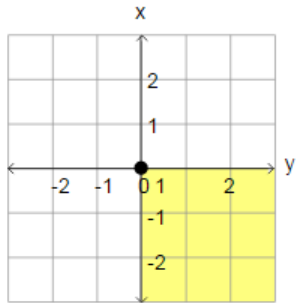
---

- Coordinate System in Computing
- Drawing Shapes
- Formatting the Display Window
- Flow of Control



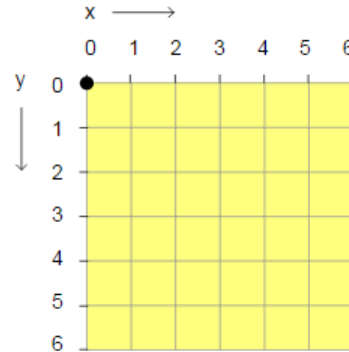
# Coordinate System in Computing

In Geometry,  
we use this type of  
coordinate system:



point (0,0) is in the  
centre.

In Computing, we use this type of  
coordinate system to represent the  
screen:



point (0,0) is in the top left hand  
corner. Each number is a pixel.



# Aside - What is Processing?

---

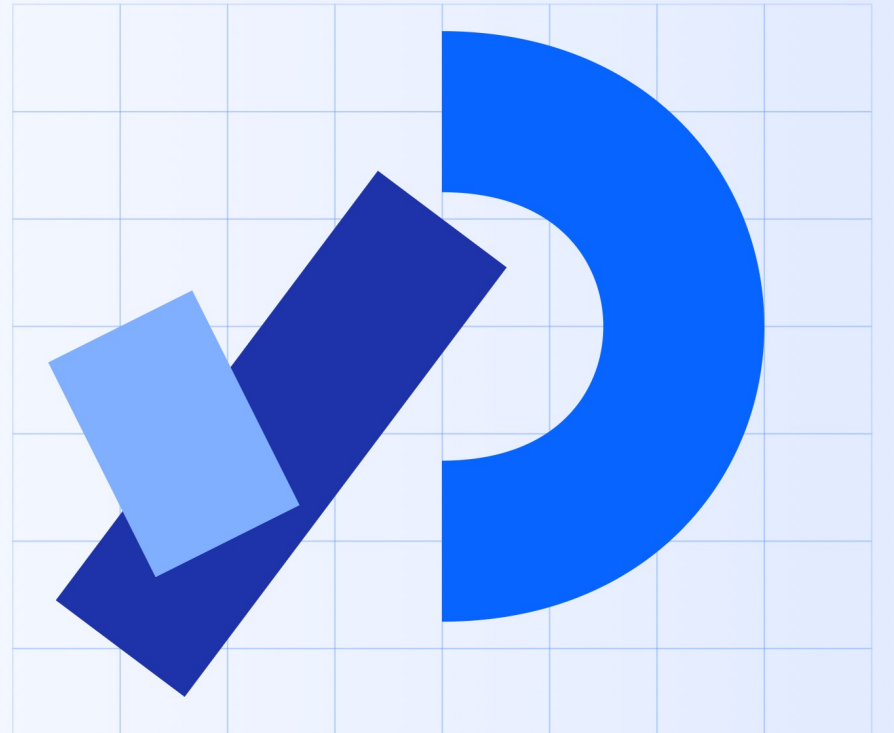
## Welcome to Processing!

Processing is a flexible software sketchbook and a language for learning how to code. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. There are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning and prototyping.

Download

Reference

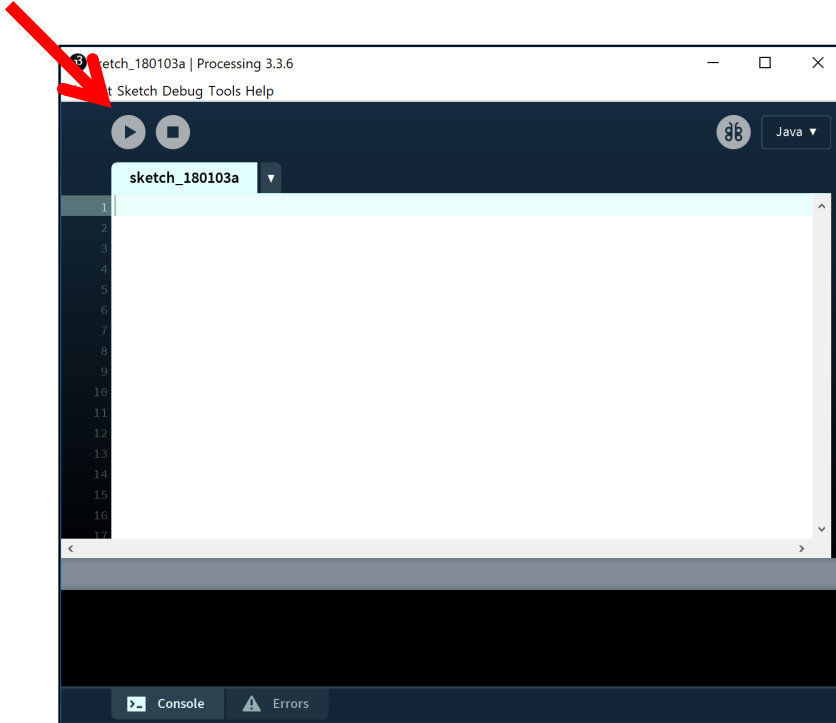
Donate



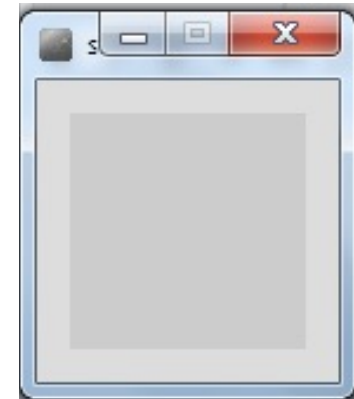
# Coordinate System in Computing



Run  
button



- So how does this relate to Processing?
- When you open Processing and click on the run button, a display window pops up.



Display window



# Coordinate System in Computing

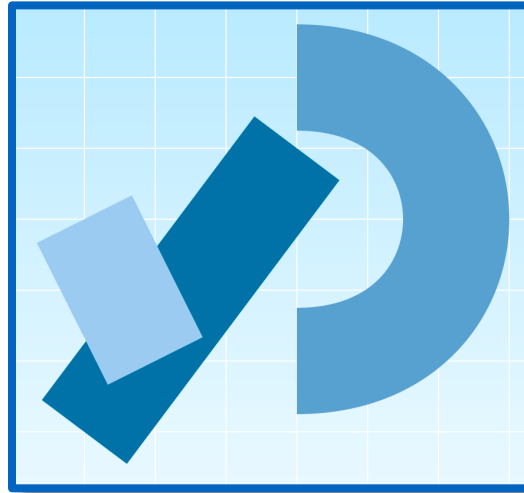
- ❑ The display window is where your code is run/ displayed.
- ❑ It follows the rules of the Computing coordinate system i.e. the top left hand corner is (0,0).
- ❑ A point (10,20) is 10 pixels to the right of (0,0) and 20 pixels below (0,0).



Display window



# Drawing Shapes





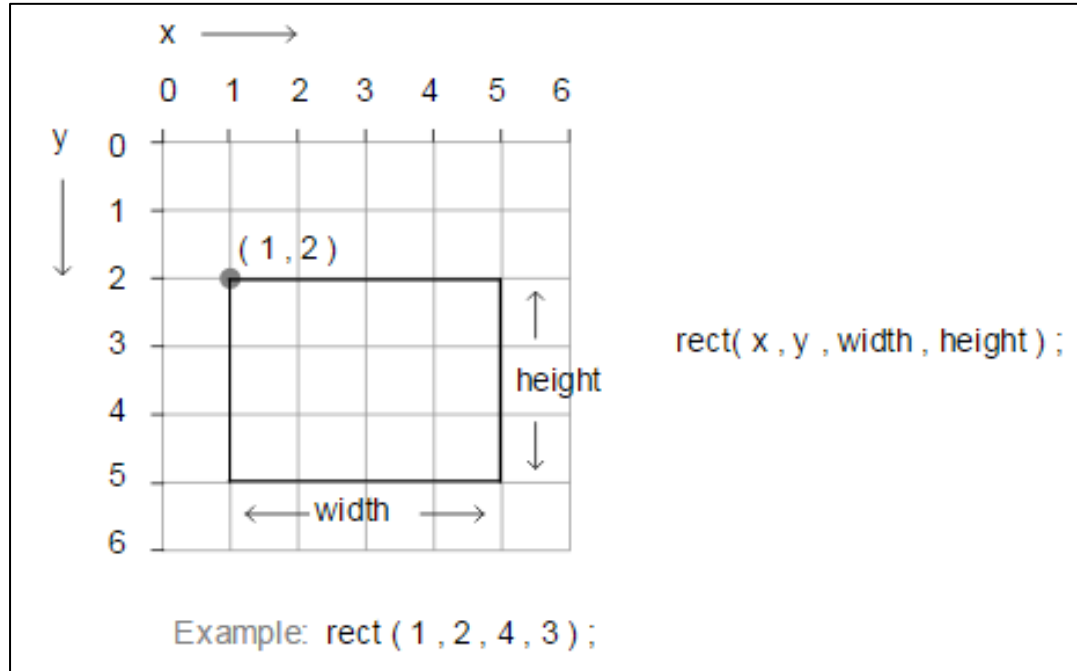


# Functions in Processing

---

- ❑ Processing comes with several pre-written functions that we can use.
- ❑ A function comprises a set of instructions that performs some task.
- ❑ When you call the function, it performs the task.
- ❑ We will now look at functions that draw the following shapes:
  - Rectangle, square, line, oval and circle.

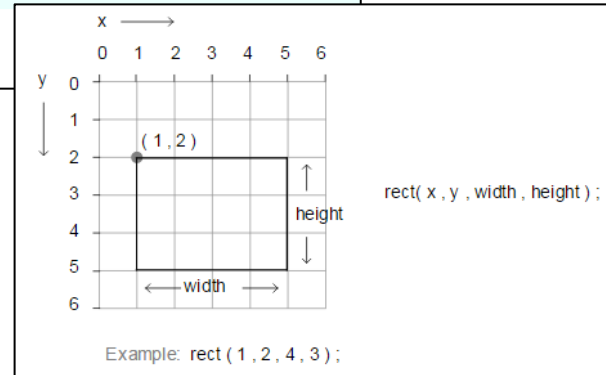
# rect()





# rect() – drawing a rectangle

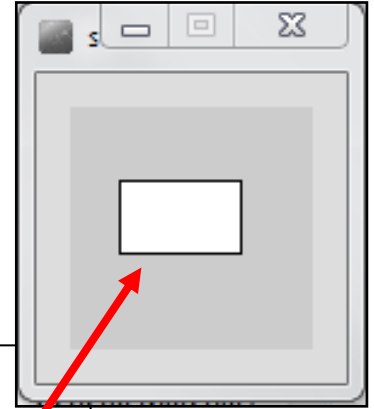
```
sketch_180103a | Processing 3.3.6  
File Edit Sketch Debug Tools Help  
  
[Play] [Stop]  
  
sketch_180103a ▾  
1 rect(20, 30, 50, 30);  
2  
3  
4
```



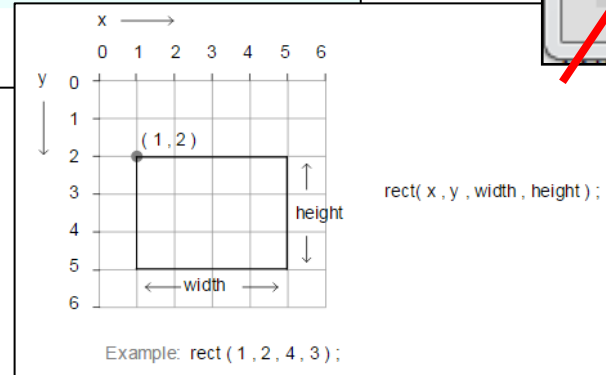


# rect() – drawing a rectangle

```
sketch_180103a | Processing 3.3.6  
File Edit Sketch Debug Tools Help  
sketch_180103a  
1 rect(20, 30, 50, 30);  
2  
3  
4
```



Click to Run

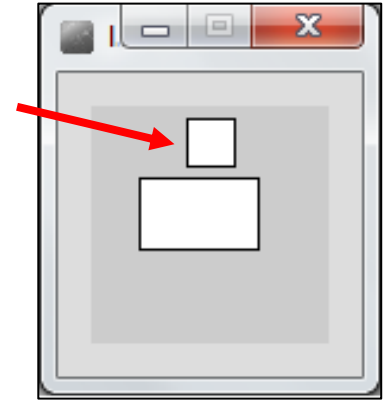




# rect() – drawing a square

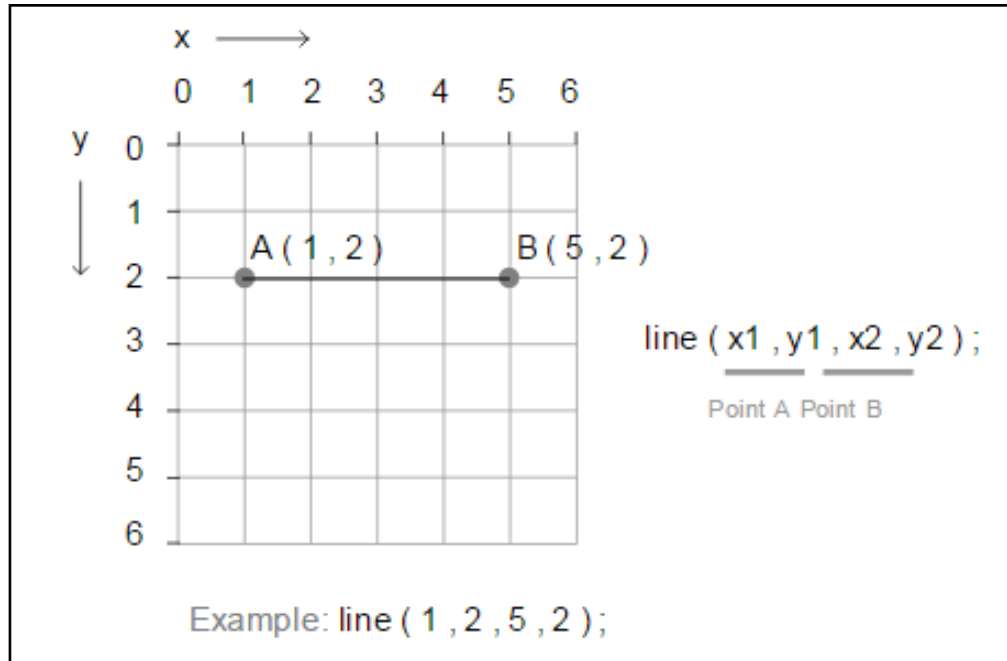
```
sketch_180103a | Processing 3.3.6
File Edit Sketch Debug Tools Help

sketch_180103a
1 rect(20, 30, 50, 30);
2 rect(40, 5, 20, 20);
3
4
5
```



Note how each line of code has a semi-colon (;) at the end of it. This is called a **statement terminator** and must be included.

# line()

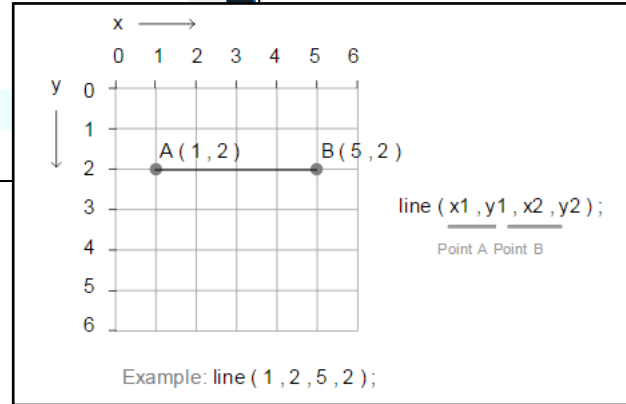




# line() – drawing a line

```
sketch_180103a | Processing 3.3.6
File Edit Sketch Debug Tools Help

sketch_180103a
1 rect(20,30,50,30);
2 rect(40,5,20,20);
3 line(5,30,20,90);
4
5
```

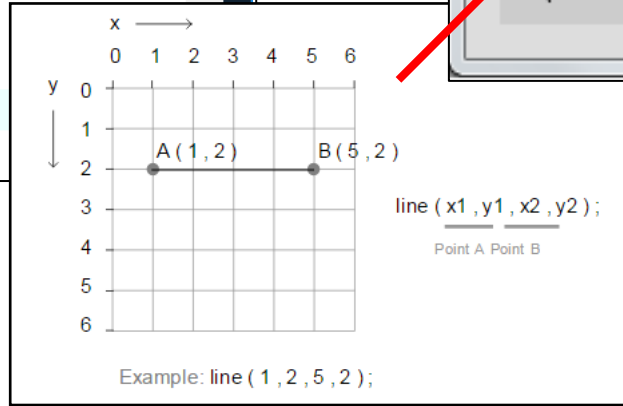
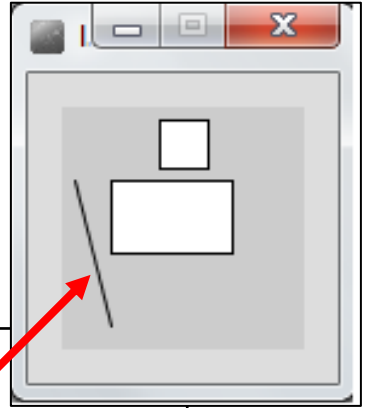




# line() – drawing a line

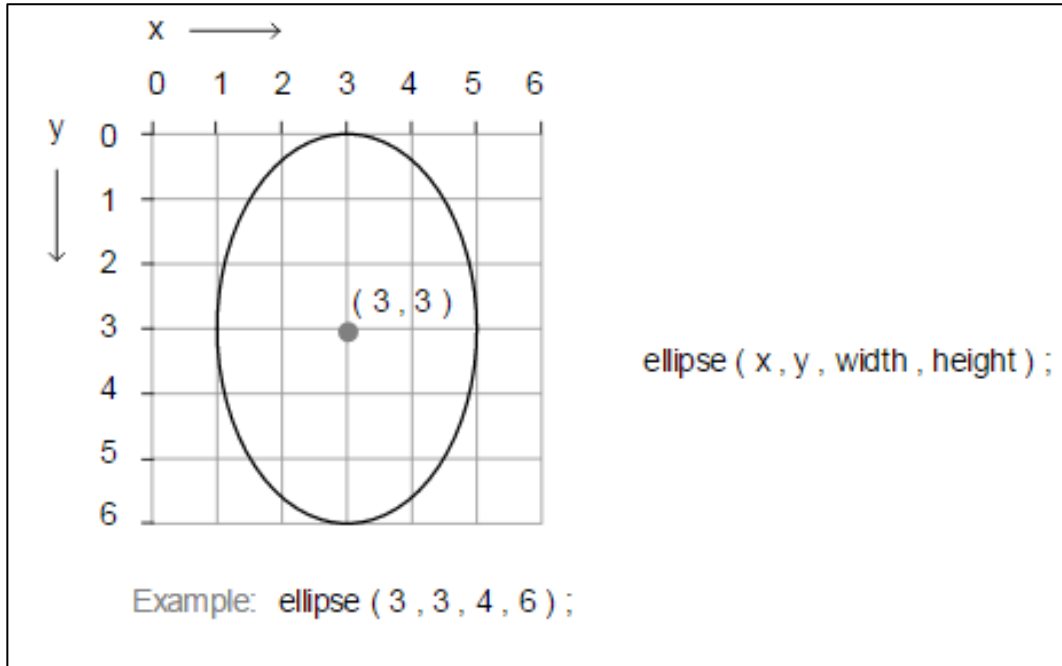
```
sketch_180103a | Processing 3.3.6  
File Edit Sketch Debug Tools Help  
Java ▾  
sketch_180103a ▾  
1 rect(20,30,50,30);  
2 rect(40,5,20,20);  
3 line(5,30,20,90);  
4  
5
```

Click to Run





# ellipse()



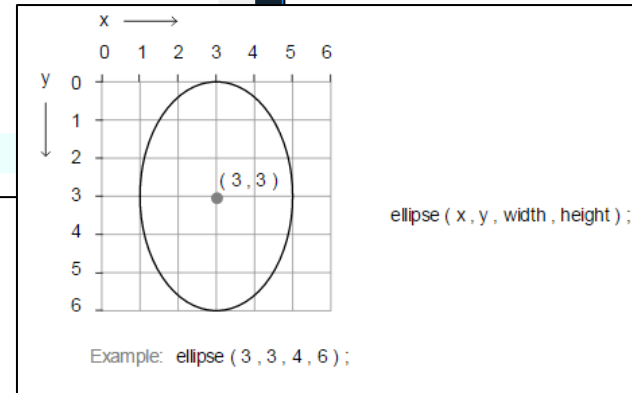


# ellipse() – drawing an oval

```
sketch_180103a | Processing 3.3.6
File Edit Sketch Debug Tools Help

[Run] [Stop] [Java]

sketch_180103a
1 rect(20,30,50,30);
2 rect(40,5,20,20);
3 line(5,30,20,90);
4 ellipse(85,50,20,60);
5
```

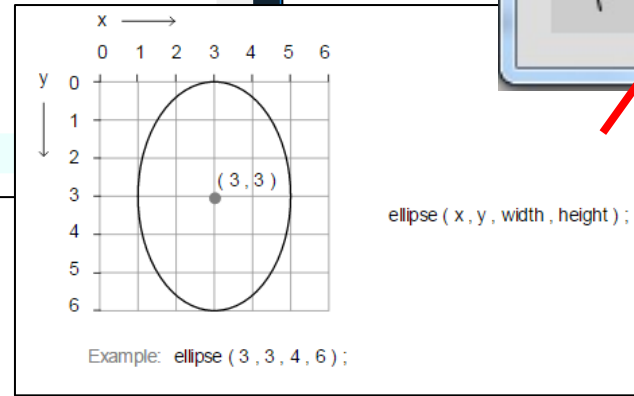
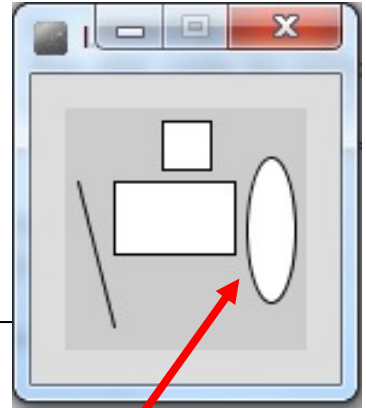




# ellipse() – drawing an oval

```
sketch_180103a | Processing 3.3.6  
File Edit Sketch Debug Tools Help  
Java ▾  
sketch_180103a ▾  
1 rect(20,30,50,30);  
2 rect(40,5,20,20);  
3 line(5,30,20,90);  
4 ellipse(85,50,20,60);  
5
```

Click to Run

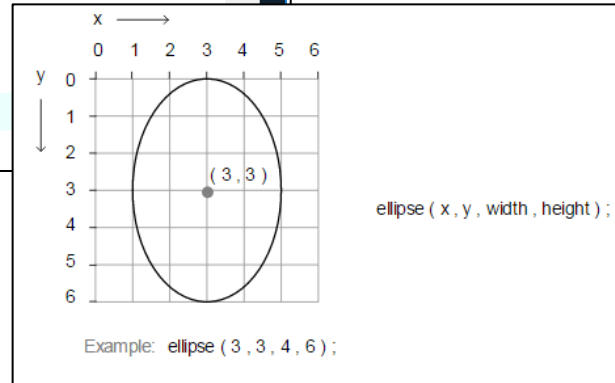




# ellipse() – drawing a circle

```
sketch_180103a | Processing 3.3.6
File Edit Sketch Debug Tools Help

sketch_180103a
1 rect(20,30,50,30);
2 rect(40,5,20,20);
3 line(5,30,20,90);
4 ellipse(85,50,20,60);
5 ellipse(50,80,15,15);
6
7
```

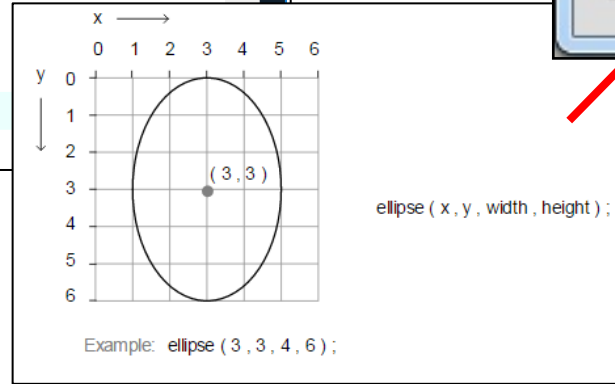
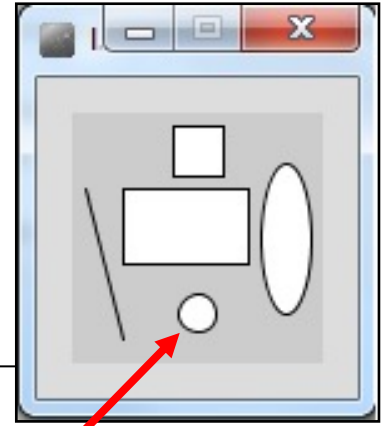




# ellipse() – drawing a circle

```
sketch_180103a | Processing 3.3.6  
File Edit Sketch Debug Tools Help  
Java ▾  
sketch_180103a  
1 rect(20,30,50,30);  
2 rect(40,5,20,20);  
3 line(5,30,20,90);  
4 ellipse(85,50,20,60);  
5 ellipse(50,80,15,15);  
6  
7
```

Click to Run





---

# Formatting the Display Window

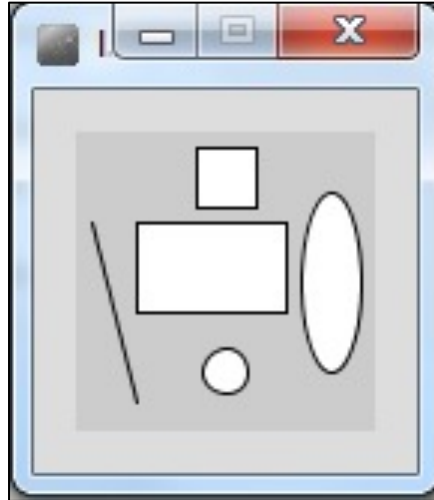




# Formatting the Display Window

---

- ❑ Our display window is looking fairly cramped.
- ❑ The default size of your display window is 100 x 100 pixels, which is quite small.





# Formatting the Display Window

---

- ❑ We can change the size of the display window by calling the `size` function.
- ❑ When you use the `size` function in static drawings, it has to be the first line of code in your sketchbook.

```
size(w, h)
```

w = width of the display window

h = height of the display window



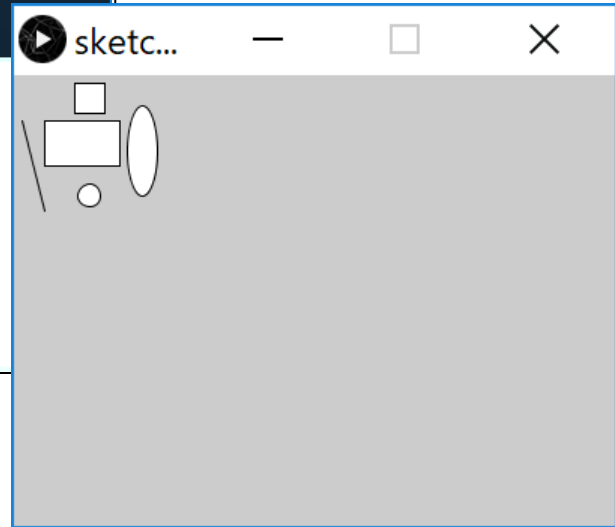
# size()



```
sketch_180103a | Processing 3.3.6
File Edit Sketch Debug Tools Help

[Run] [Stop] [Java]

sketch_180103a
1 size(400,300);
2
3 rect(20,30,50,30);
4 rect(40,5,20,20);
5 line(5,30,20,90);
6 ellipse(85,50,20,60);
7 ellipse(50,80,15,15);
8
```

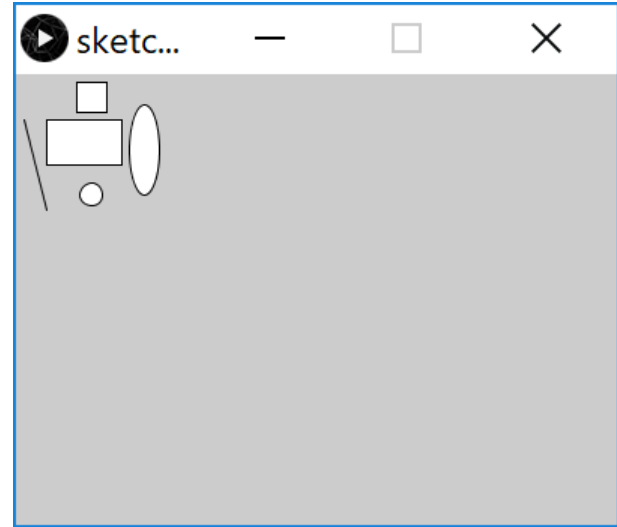




# Formatting the Display Window

---

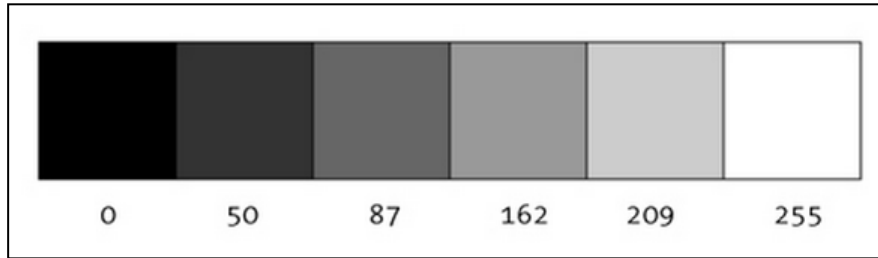
- ❑ Our display window looks less cramped now.
- ❑ But maybe we want to change the default gray colour?
- ❑ We could use the **background** function to set the colour to something else.





# A note on colour first... Grayscale

---



“0 means black, 255 means white. In between, every other number - 50, 87, 162, 209, and so on - is a shade of gray ranging from black to white.”



# background() - syntax

---

background(**grayscale**)

**grayscale** = grayscale colour (a number between  
0 [black] and 255 [white] inclusive)

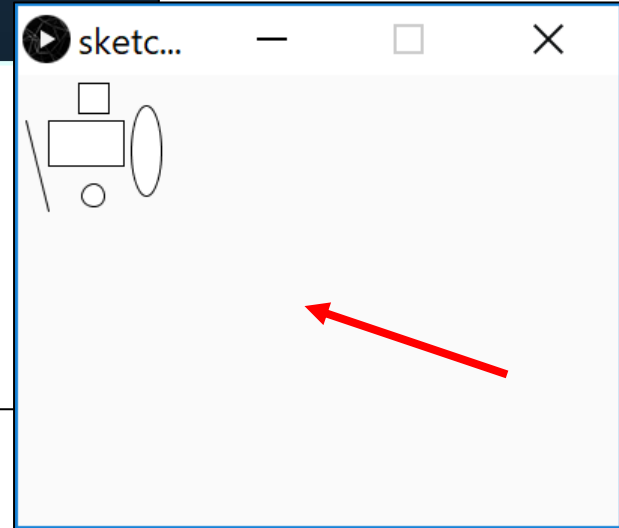


# background()

```
sketch_180103a | Processing 3.3.6
File Edit Sketch Debug Tools Help

[Run] [Stop] [Java]

sketch_180103a
1 size(400,300);
2 background(250);
3
4 rect(20,30,50,30);
5 rect(40,5,20,20);
6 line(5,30,20,90);
7 ellipse(85,50,20,60);
8 ellipse(50,80,15,15);
```





# Flow of Control



# Problem Solving

---



Programming **IS**  
problem solving.





# Flow of Control in a Program

---

□ Each program you write will typically have:

Sequence	Things that will be done in a particular order
Selection	Things that will be done conditionally
Iteration	Things that will be done repetitively





# Flow of Control in a Program

---

- ❑ Each program you write will typically have:

Sequence	Things that will be done in a particular order
Selection	Things that will be done conditionally
Iteration	Things that will be done repetitively

- ❑ The following example demonstrates *Sequence*
- ❑ We will cover *Selection* and *Iteration* in future weeks



# Sequence of Instructions – Example

The image shows a screenshot of the Processing IDE. The main window is titled "sketch\_180103a | Processing 3.3.6" and has a menu bar with "File", "Edit", "Sketch", "Debug", "Tools", and "Help". Below the menu bar are several icons: a play button, a square button, a gear icon, and a "Java" button. A dropdown menu is open, showing "sketch\_180103a". The code editor contains the following code:

```
1 size(400,300);  
2 background(250);  
3  
4 rect(20,30,50,30);  
5 rect(40,5,20,20);  
6 line(5,30,20,90);  
7 ellipse(85,50,20,60);  
8 ellipse(50,80,15,15);
```

Overlaid on the right side of the IDE is a smaller window titled "sketc...". This window displays a preview of the sketch, showing a simple line drawing of a mechanical part or tool with various geometric shapes like rectangles, lines, and ellipses.



# Sequence of Instructions – Matters!!

The screenshot shows the Processing IDE interface. The main window is titled "sketch\_180103a | Processing 3.3.6" and contains a menu bar with "File", "Edit", "Sketch", "Debug", "Tools", and "Help". Below the menu bar are control buttons for play, stop, and a Java icon. The code editor shows the following code:

```
1 size(400,300);  
2  
3 rect(20,30,50,30);  
4 rect(40,5,20,20);  
5 background(250);  
6 line(5,30,20,90);  
7 ellipse(85,50,20,60);  
8 ellipse(50,80,15,15);  
9
```

Overlaid on the right is a smaller window titled "sketc..." which displays a hand-drawn sketch of a line and two ellipses, corresponding to the code in the editor.

**background(250)** moved and is now fourth statement.  
What happened to the rectangle and square?

# Questions?

---





Thanks.

