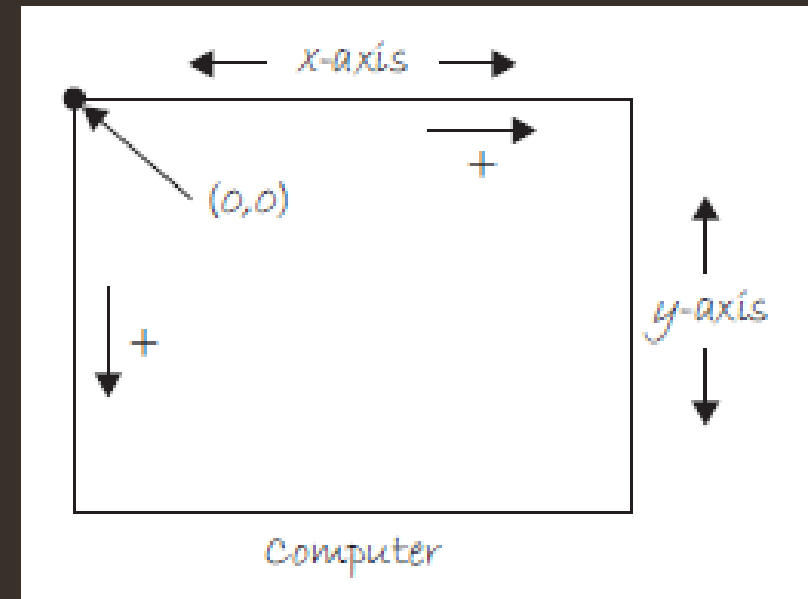


# Algoritmos Computacionales

Centro de Educación y Formación Académica (CEDUK)

# Introducción a Processing

- En Processing, se tiene un sistema de coordenadas un poco distinto al que estamos acostumbrados a utilizar. En este, la esquina superior izquierda de la ventana representa al origen (0,0).
- Cuando se avanza hacia la derecha, se dice que nos movemos en la dirección positiva del eje "x".
- Cuando se avanza hacia abajo, se dice que nos movemos en la dirección positiva del eje "y".



# Tipos de declaraciones en Processing

- **Llamadas a funciones**
- Operaciones de asignación
- Estructuras de control

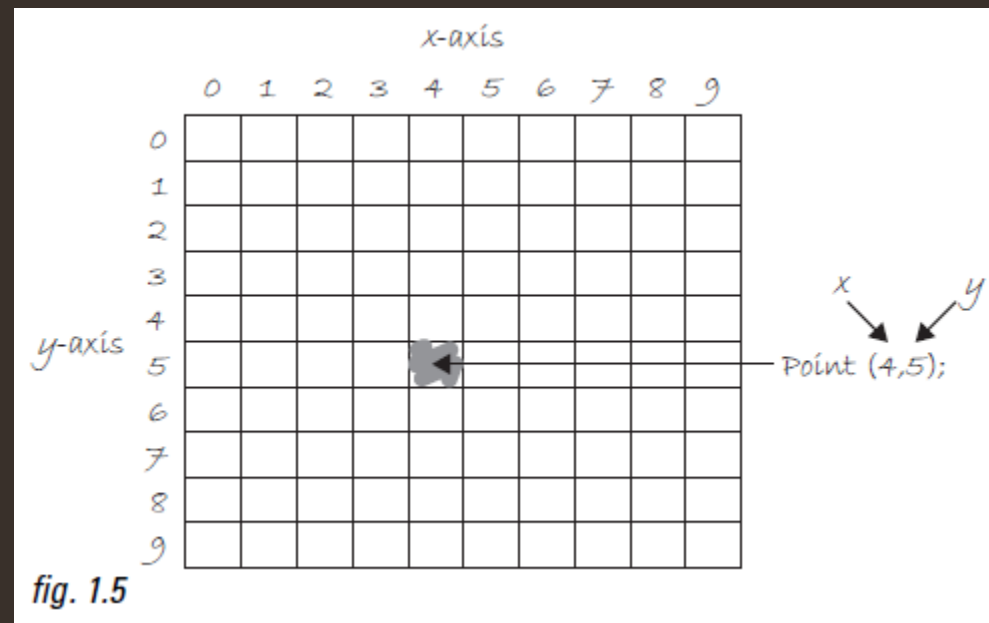
# Punto

- Suponiendo que tenemos una pantalla de 10 píxeles de ancho por 10 píxeles de alto, podemos dibujar un punto en las coordenadas (4,5) por medio de la siguiente función:

Nombre de la función

```
sketch_180201a  
1 point(4,5);  
2
```

Argumentos de la función  
(posición\_x, posición\_y)



# Línea

- O si queremos dibujar una línea que vaya del punto A (1,3) al punto B (8,3) bastaría con escribir:

Nombre de la función

```
sketch_180201a  
1 line(1,3,8,3);  
2
```

Argumentos de la función  
(posición\_xA, posición\_yA,  
posición\_xB, posición\_yB)

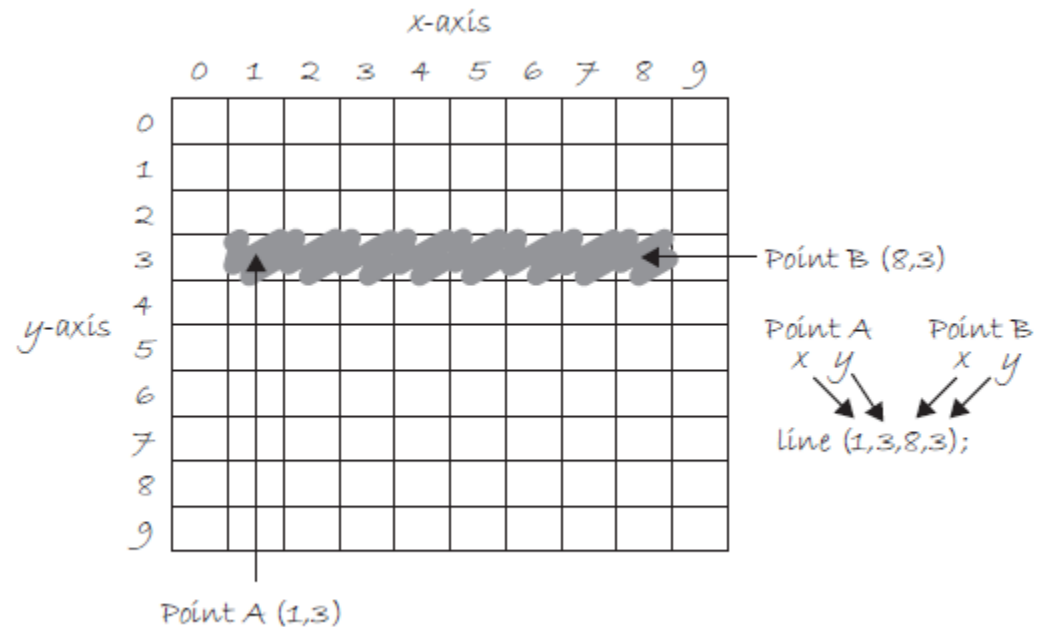
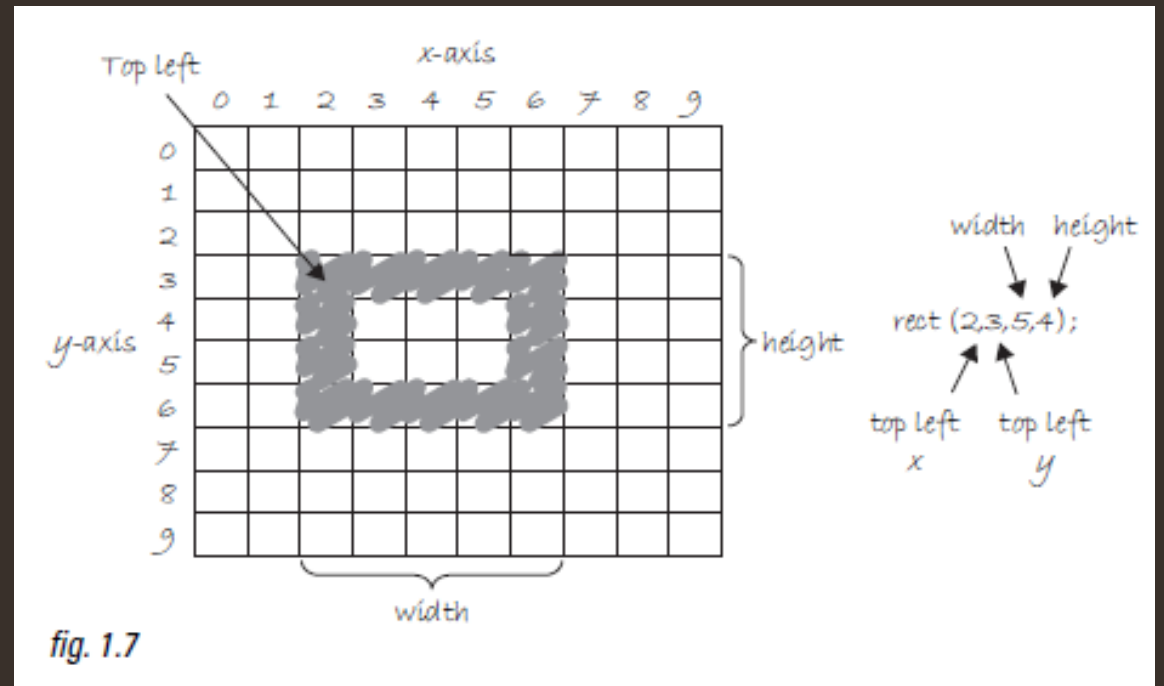


fig. 1.6

# Rectángulo

- Para dibujar un rectángulo habrá que definir en que coordenadas (x,y) estará su esquina superior izquierda, así como su ancho y su alto.

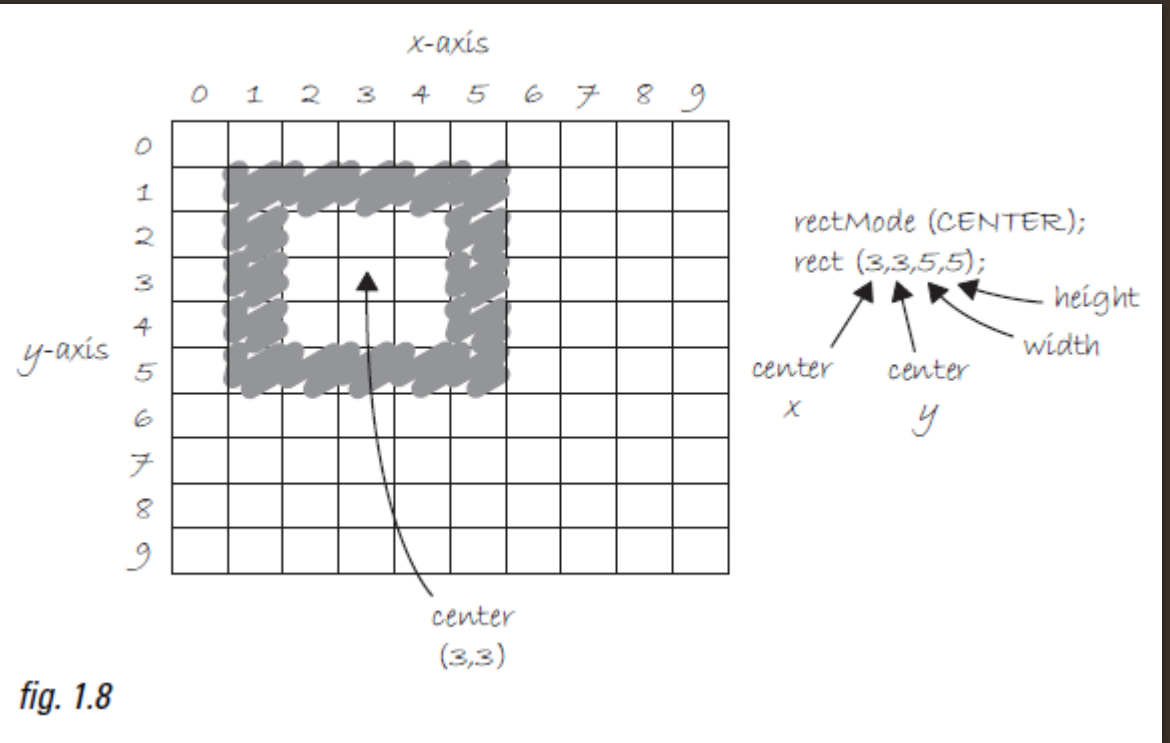
```
sketch_180201a  
1 rect(2,3,5,4);  
2
```



# Rectángulo

- A veces puede ser más simple definir las coordenadas del centro del rectángulo (en vez de la esquina). Para lograrlo solo hay que cambiar el modo de dibujar el rectángulo por medio de la función `rectMode`:

```
sketch_180201a  
1 rectMode(CENTER);  
2 rect(3,3,5,5);  
3
```



# Rectángulo

- También es posible definir un rectángulo dando a conocer las coordenadas de la esquina superior izquierda y las coordenadas de la esquina inferior derecha utilizando el modo CORNERS de la función rectMode:

```
sketch_180201a  
1 rectMode(CORNERS);  
2 rect(5,5,8,7);  
3
```

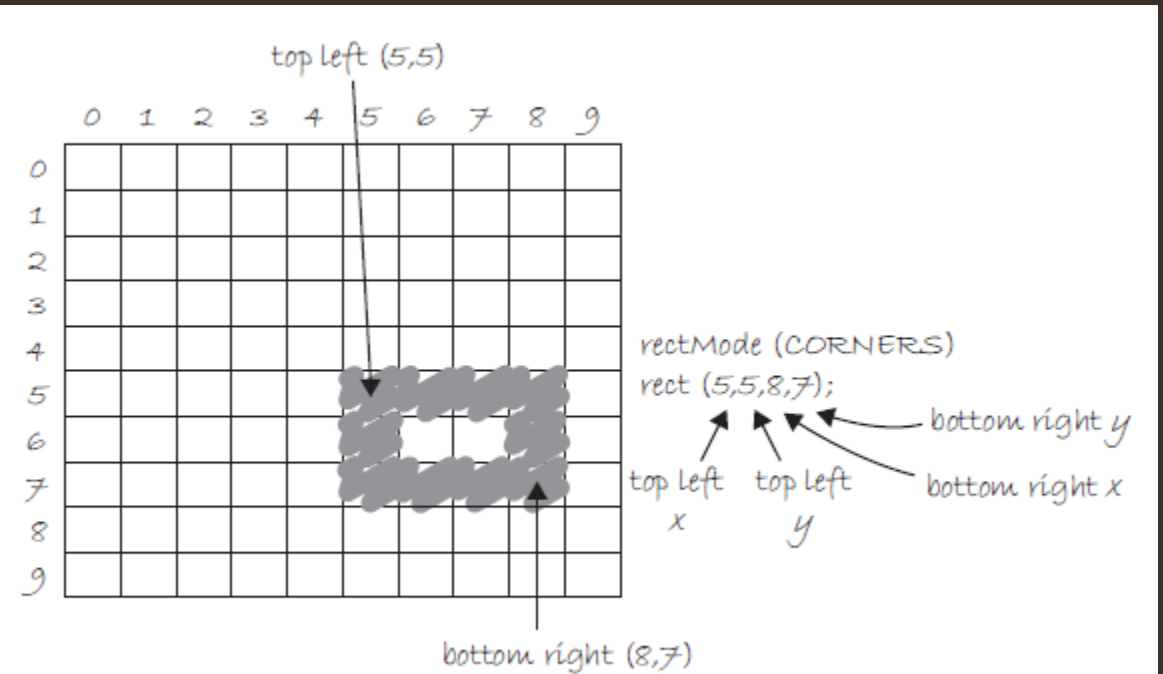
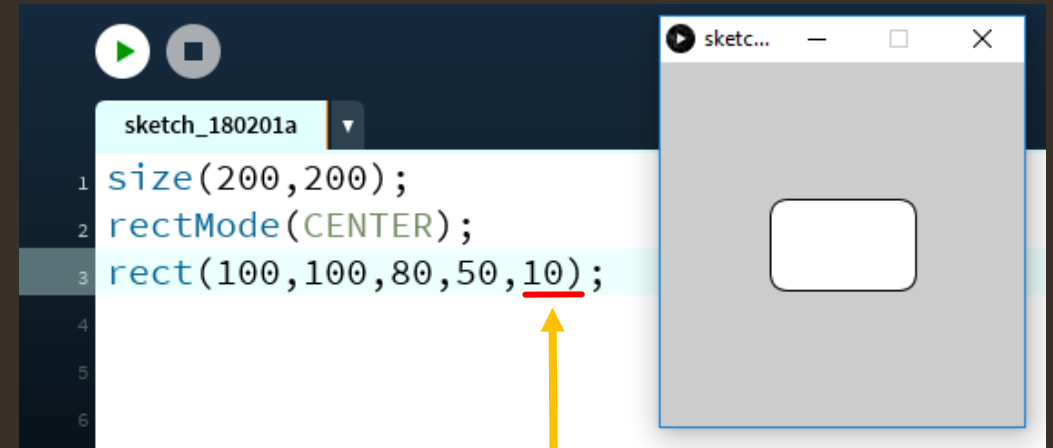


fig. 1.9



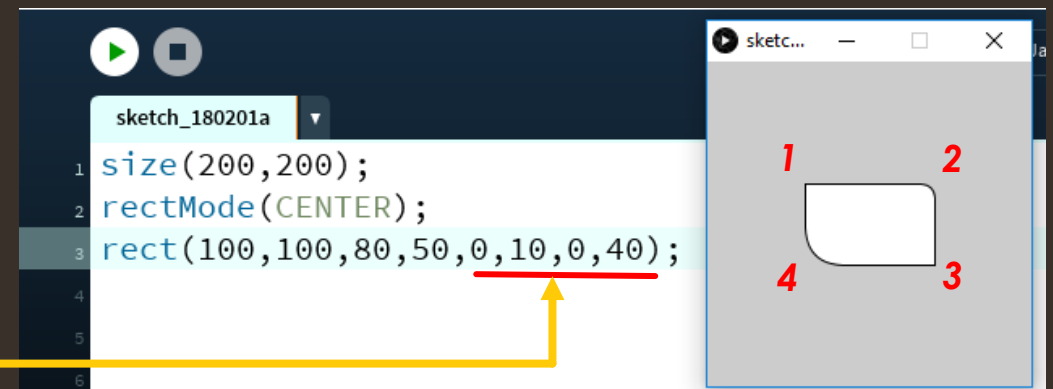
# Rectángulo

- Si a la función “rect” se le dan más argumentos, es posible definir el radio de redondeo de cada una de las cuatro esquinas.



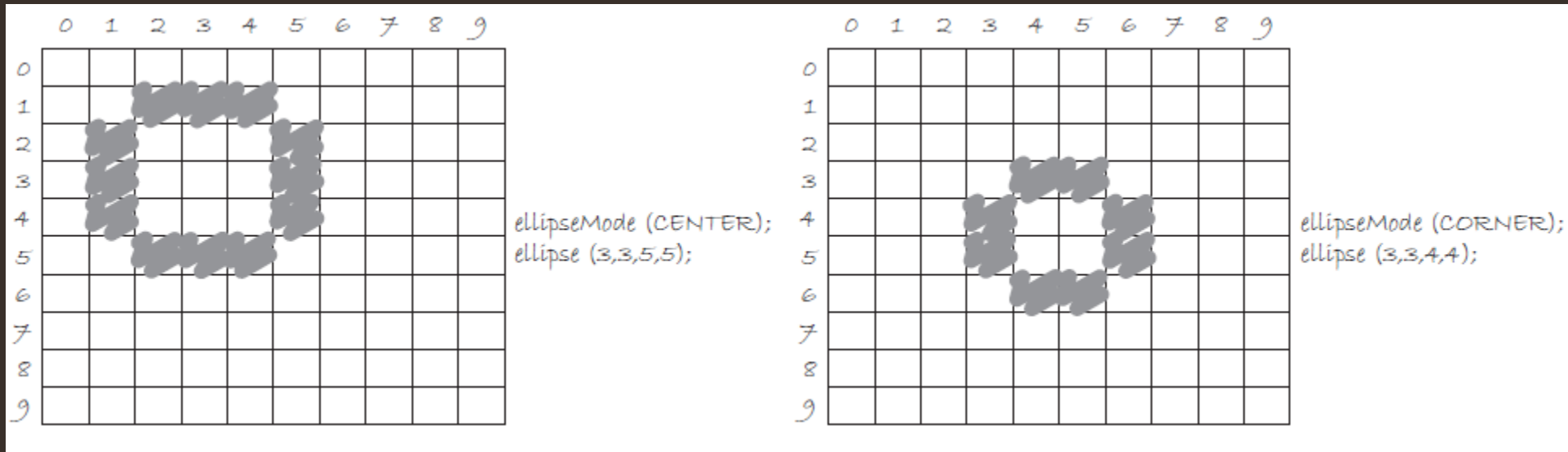
Radio de las cuatro esquinas

Radio de la esquina 1, 2, 3 y 4



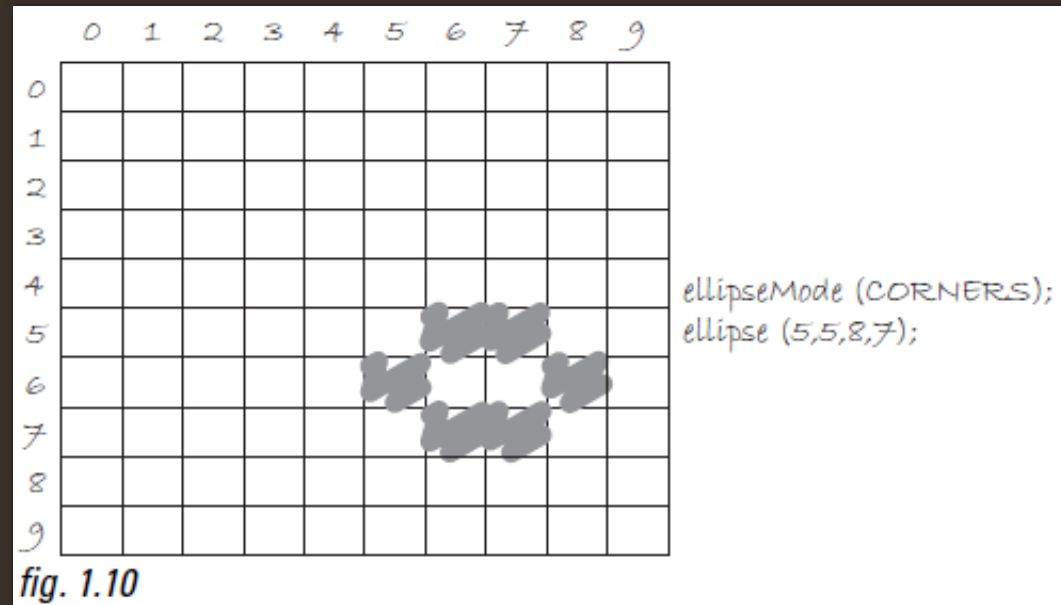
# Elipse

- Dibujar elipses se hace de forma similar a un rectángulo. También existen varios modos.



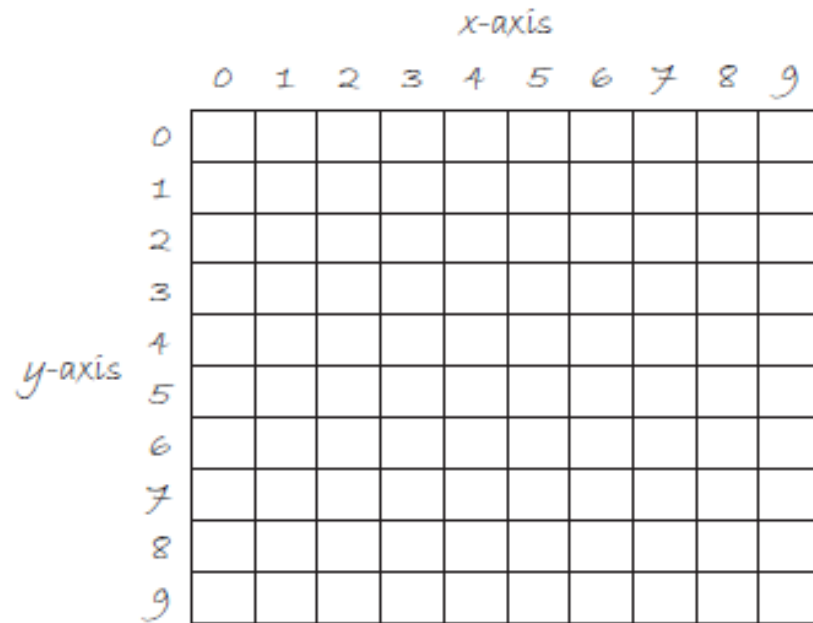
\*El modo por default al dibujar una elipse es CENTER, mientras que al dibujar un rectángulo es CORNER.

# Ellipse



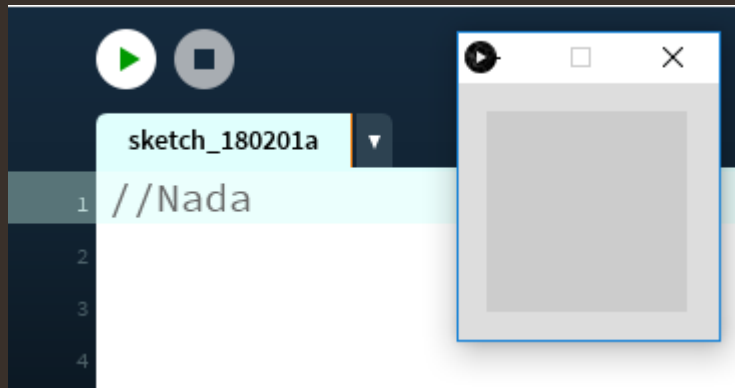
# Ejercicio

```
line(0,0,9,6);  
point(0,2);  
point(0,4);  
rectMode(CORNER);  
rect(5,0,4,3);  
ellipseMode(CENTER);  
ellipse(3,7,4,4);
```



# Ajustando la ventana

- La ventana de Processing tiene un tamaño de 100 píxeles de ancho y 100 píxeles de alto por default. Sin embargo es posible cambiar su tamaño mediante la función `size`.



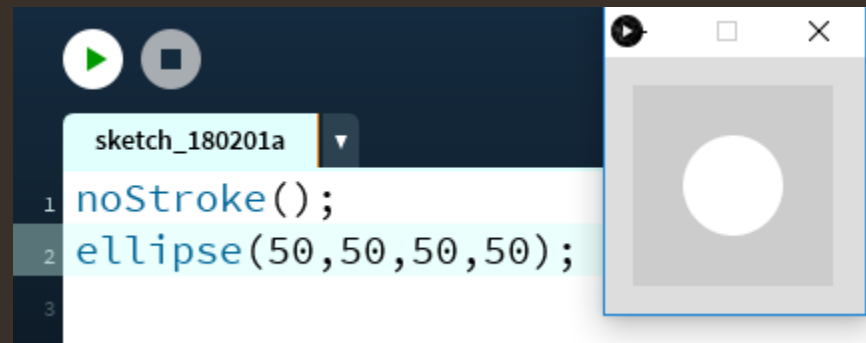
Tamaño default  
(100 píxeles de ancho x 100 píxeles de alto)



Tamaño definido por el programador  
(150 píxeles de ancho x 200 píxeles de alto)

# Sin contorno y sin relleno

- Para dibujar una figura sin contorno se utiliza la función `noStroke()`:

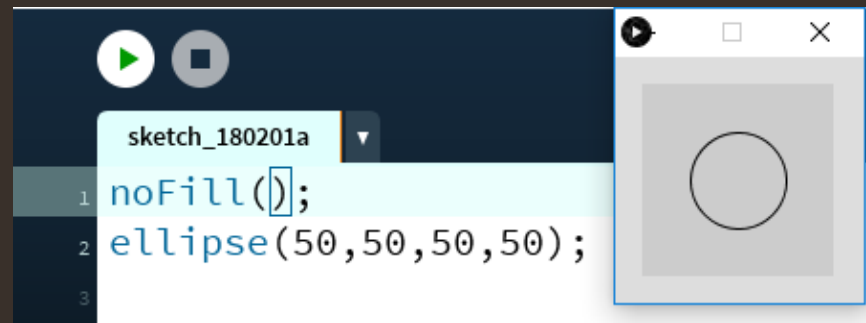


The screenshot shows an IDE window titled "sketch\_180201a" with a play button and a stop button. The code editor contains the following code:

```
1 noStroke();  
2 ellipse(50,50,50,50);  
3
```

To the right of the code editor is a preview window showing a white circle on a gray background, which is the result of the code.

- Para dibujar una figura sin relleno se utiliza la función `noFill()`:



The screenshot shows an IDE window titled "sketch\_180201a" with a play button and a stop button. The code editor contains the following code:

```
1 noFill();  
2 ellipse(50,50,50,50);  
3
```

To the right of the code editor is a preview window showing a gray circle on a gray background, which is the result of the code.

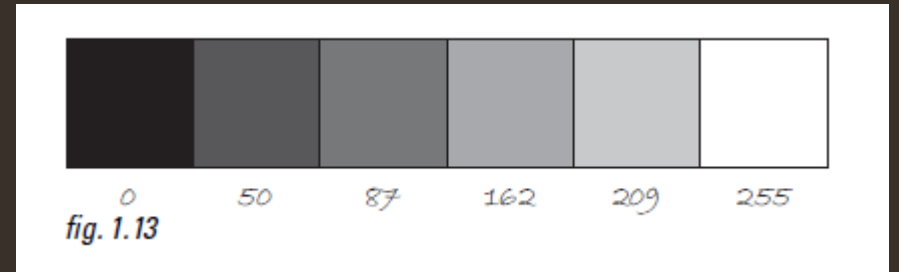
# Escala de grises (se usa un argumento)

- Para pintar contornos usaremos la función:

```
sketch_180201a  
1 stroke(0); //contorno negro  
2
```

- Para pintar rellenos usaremos la función:

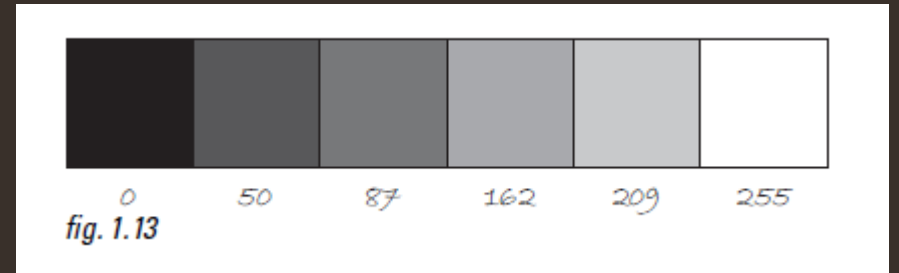
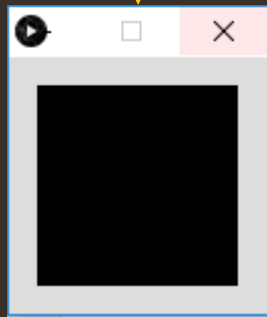
```
sketch_180201a  
1 fill(255); //relleno blanco  
2  
3
```



# Escala de grises (se usa un argumento)

- Para pintar el fondo o toda la ventana se utiliza la función `background`:

```
sketch_180201a  
1 background(0); //fondo negro  
2
```





# Colores (Se usan tres argumentos)

- Para pintar de cualquier color se usan tres argumentos en la función fill/stroke según sea relleno/contorno.
- ¿Qué valores escogería para obtener el color deseado?

```
fill(_____,_____,_____) ;
```

```
ellipse(20,40,16,16) ;
```

Bright blue

```
fill(_____,_____,_____) ;
```

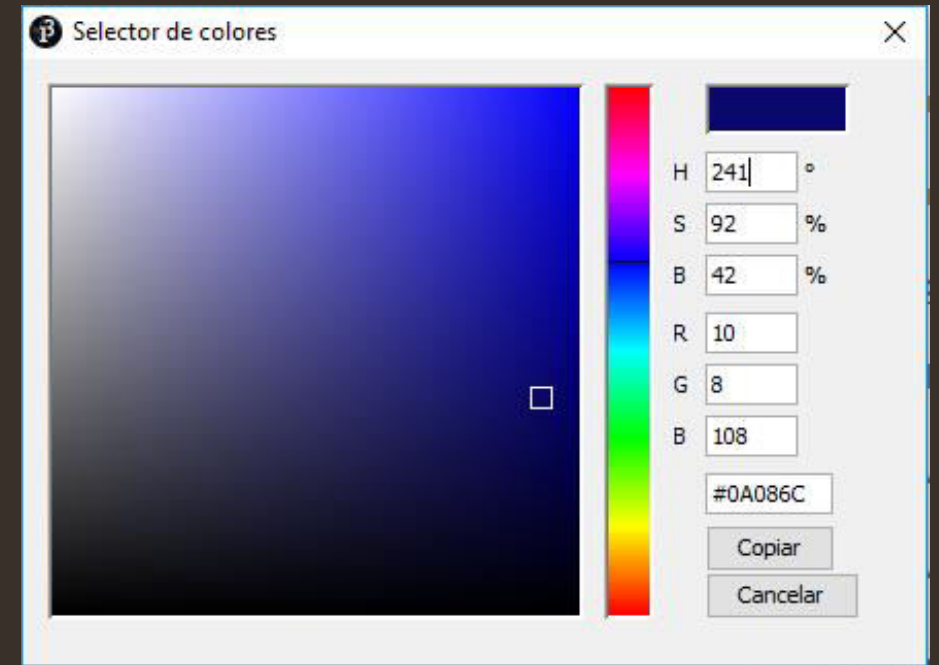
```
ellipse(40,40,16,16) ;
```

Dark purple

```
fill(_____,_____,_____) ;
```

```
ellipse(60,40,16,16) ;
```

Yellow

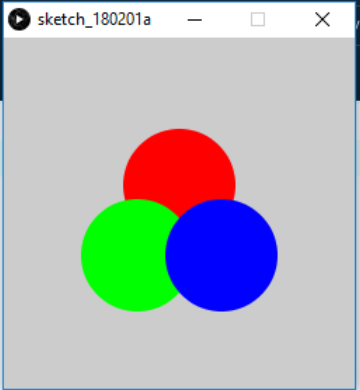


\*Herramientas... > Selector de colores

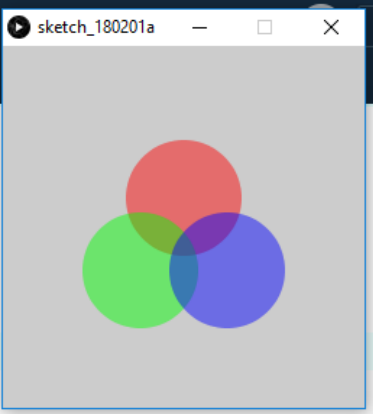
# Transparencia

- Adicionalmente a los componentes rojo, verde y azul, hay un cuarto componente opcional conocido como “alfa”. Alfa se refiere a la transparencia y es particularmente útil cuando se desea ver una figura a través de otra. Los valores de alfa van de 0 (0% opacidad) hasta 255 (100% opacidad).

```
sketch_180201a
1 size(250,250);
2 noStroke();
3 fill(255,0,0);
4 ellipse(125,105,80,80);
5 fill(0,255,0);
6 ellipse(95,155,80,80);
7 fill(0,0,255);
8 ellipse(155,155,80,80);
```

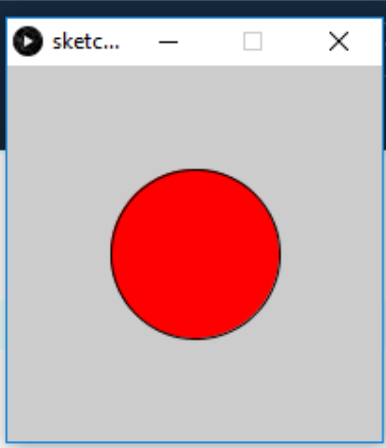
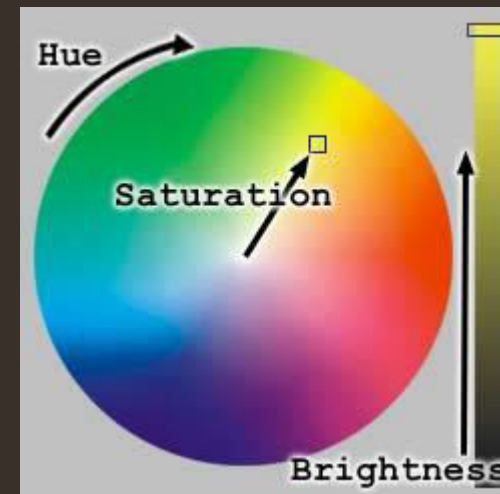
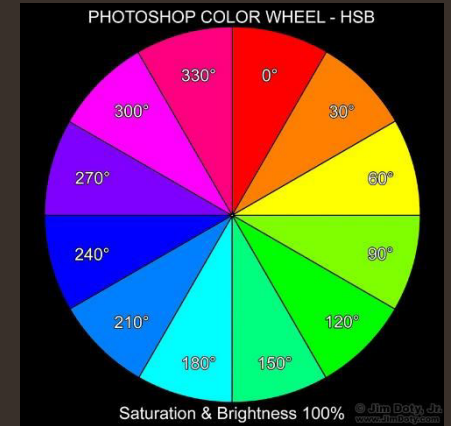
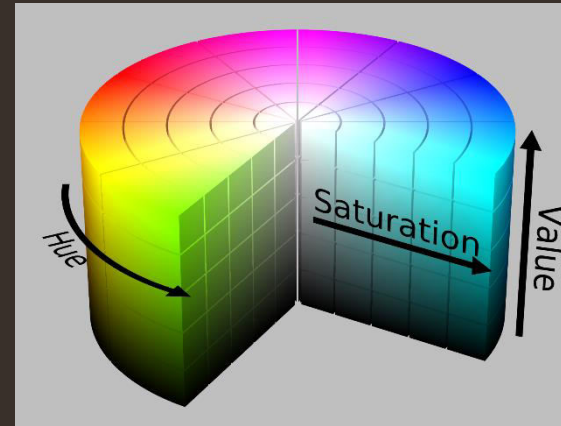
A screenshot of a code editor window titled 'sketch\_180201a' showing a sketch of three overlapping circles. The top circle is red, the bottom-left is green, and the bottom-right is blue. They are drawn with no stroke and are completely opaque, so they do not overlap.

```
sketch_180201a
1 size(250,250);
2 noStroke();
3 fill(255,0,0,120);
4 ellipse(125,105,80,80);
5 fill(0,255,0,120);
6 ellipse(95,155,80,80);
7 fill(0,0,255,120);
8 ellipse(155,155,80,80);
```

A screenshot of a code editor window titled 'sketch\_180201a' showing a sketch of three overlapping circles. The top circle is red, the bottom-left is green, and the bottom-right is blue. They are drawn with no stroke and have a transparency (alpha) of 120. The overlapping areas are visible, showing the colors of the circles behind them.

# Modos de color

```
sketch_180201a
1 size(200,200);
2 colorMode(HSB,360,100,100);
3 fill(0,100,100);
4 ellipse(100,100,90,90);
```

A screenshot of a sketch window titled "sketc...". The window displays a red circle centered on a light gray background. The sketch is contained within a square frame.

# Tipos de errores

- De sintaxis (ejemplos: no poner punto y coma; escribir elipse en vez de elipse)
- De operación (ejemplos: dividir entre cero; hacer referencia a un archivo inexistente)
- De lógica (ejemplo: querer un rectángulo rojo y que salga verde)