

# REPASO VARIABLES

1º DECLARAR LA VARIABLE,  
Indicamos el tipo de variable seguido por el nombre.



A diagram illustrating the components of a variable declaration. At the top, the text 'tipo de dato' (data type) and 'nombre' (name) are written in a light color. Below each, a white arrow points downwards to the corresponding parts of the code 'int edad;'. The word 'int' is in red and 'edad;' is in yellow.

2º INICIALIZAMOS VARIABLE: le asignamos un valor.

```
int count;  
count = 50;
```

← Declarar e inicializar una variable en dos líneas de código.

```
int count = 50;
```

← Declarar e inicializar una variable en una sola línea de código.

# MODO ESTÁTICO Y MODO DINÁMICO:

INICIALIZACIÓN `setup()`



REPETICIÓN `draw()`



```
File Edit Sketch Tools Help
[Icons: Play, Stop, Refresh, Up, Down, Right]
sketch_cuadrado_void_draw
void setup(){
  size(500,500);
}
void draw(){
  background(100);
  stroke(255);
  fill(0);
  rect(mouseX, mouseY, 90, 90);
}
```

## Variables del mouse

<b>mouseX</b>	→	Posición del ratón en el eje X (frame actual)
<b>mouseY</b>	→	Posición del ratón en el eje Y (frame actual)
<b>pmouseX</b>	→	Posición del ratón en el eje X (frame previo)
<b>pmouseY</b>	→	Posición del ratón en el eje Y (frame previo)
<b>mousePressed</b>	→	¿Verdadero o falso? ¿Está presionando el ratón?

**mouseReleased** → cuando soltamos el ratón

**mouseDragged** → cuando el ratón es arrastrado.

## Variables del teclado

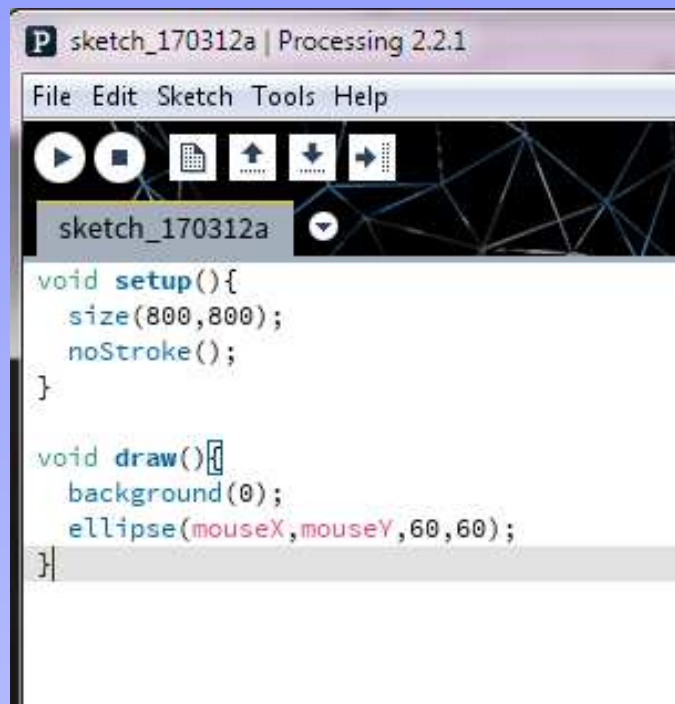
<b>key</b>	→	La última tecla presionada
<b>keyCode</b>	→	Código numérico de la tecla presionada
<b>keyPressed</b>	→	¿Verdadero o falso? ¿Está siendo pulsada una tecla?

**mousePressed:** bloque de código que ejecuta todo lo que está dentro cada vez que el botón del mouse es apretado.

**mouseReleased:** el código dentro de este bloque se ejecuta cuando el botón del ratón es liberado.

**mouseDragged:** es ejecutado cuando el botón del mouse es apretado y arrastrado.

**pmouseX, pmouseY:** toman la posición previa del mouseX, mouseY, es decir, las coordenadas del mouse en el ciclo anterior para ejecutar una continuidad en el trazo.



```
void setup(){
  size(800,800);
  noStroke();
}

void draw(){
  background(0);
  ellipse(mouseX,mouseY,60,60);
}
```

mouseX,mouseY, estas dos variables están siendo leídas constantemente por processing, por este motivo no es necesario declararlas.  
**VARIABLES DE SISTEMA**

## VARIABLES del sistema;

width y height:

### Ancho y Alto

El lenguaje Processing ha incorporado variables de uso cotidiano. Por ejemplo, la altura y la anchura de la ventana de visualización se almacenan en variables denominadas "width" y "height".



<b>width</b>	→	Ancho (en pixeles) de la ventana de visualización
<b>height</b>	→	Alto (en pixeles) de la ventana de visualización

sketch\_160325b\_cuadrado\_mouse | Processing 2.2.1

File Edit Sketch Tools Help



sketch\_160325b\_cuadrado\_mouse

```
void setup(){
  size(500,500);
}
void draw(){
  background(100);
  stroke(255);
  fill(0);
  rectMode(CENTER);
  rect(width/2,height/2,mouseX,mouseY);
}
```

# CREAMOS UNA ESTELA CON EL MOVIMIENTO DE LA FIGURA DIBUJADA



```
void setup(){
  size(800,800);
  noStroke();
}

void draw(){

  fill(255);
  ellipse(mouseX,mouseY,60,60);
}
```

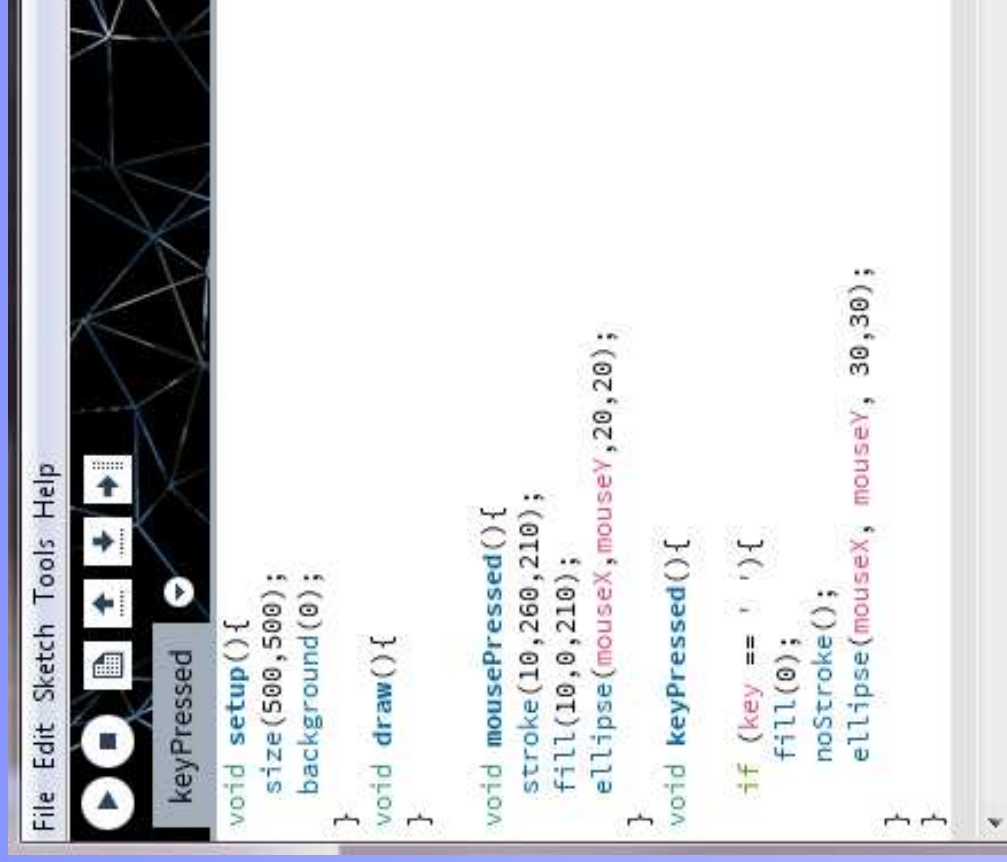
Figura = tamaño de mi lienzo

## Variables del teclado

**key** —————> La última tecla presionada

**keyCode** —————> Código numérico de la tecla presionada

**keyPressed** —————> ¿Verdadero o falso? ¿Está siendo pulsada una tecla?



```
File Edit Sketch Tools Help
keyPressed
void setup(){
  size(500,500);
  background(0);
}
void draw(){
}

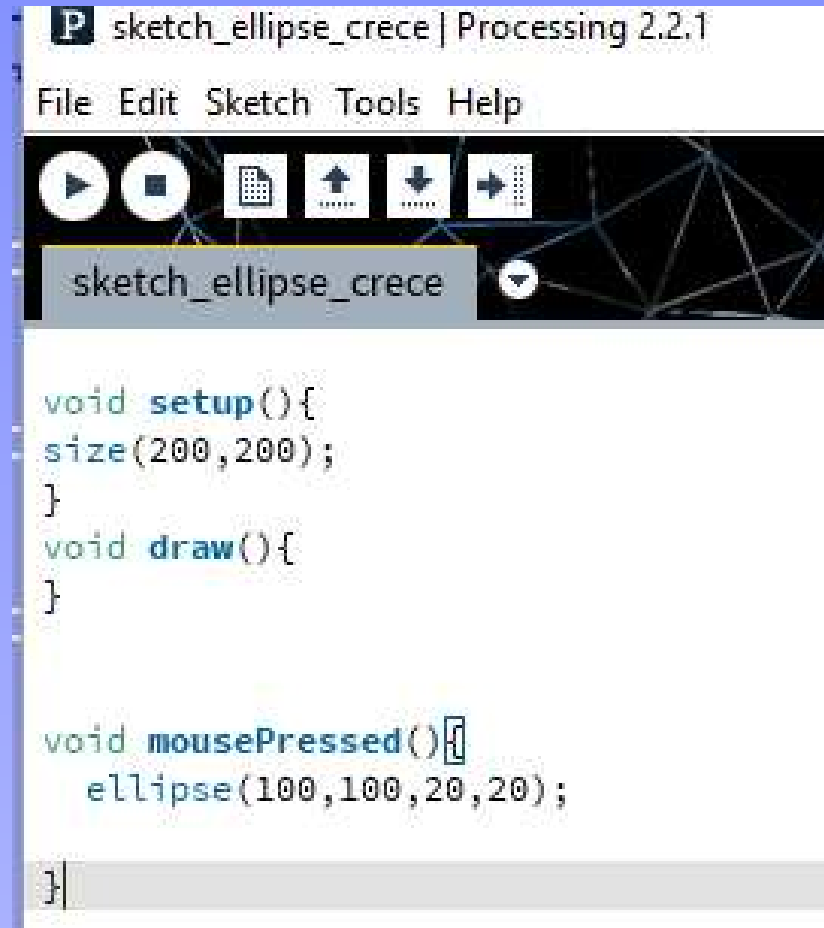
void mousePressed(){
  stroke(10,260,210);
  fill(10,0,210);
  ellipse(mouseX,mouseY,20,20);
}
void keyPressed(){
  if (key == ' '){
    fill(0);
    noStroke();
    ellipse(mouseX, mouseY, 30,30);
  }
}
```



## EJERCICIOS:

\*\* de lo que partimos:

Pelota crece con mousePressed



```
sketch_ellipse_crece | Processing 2.2.1
File Edit Sketch Tools Help

sketch_ellipse_crece

void setup(){
  size(200,200);
}
void draw(){
}

void mousePressed(){
  ellipse(100,100,20,20);
}
```

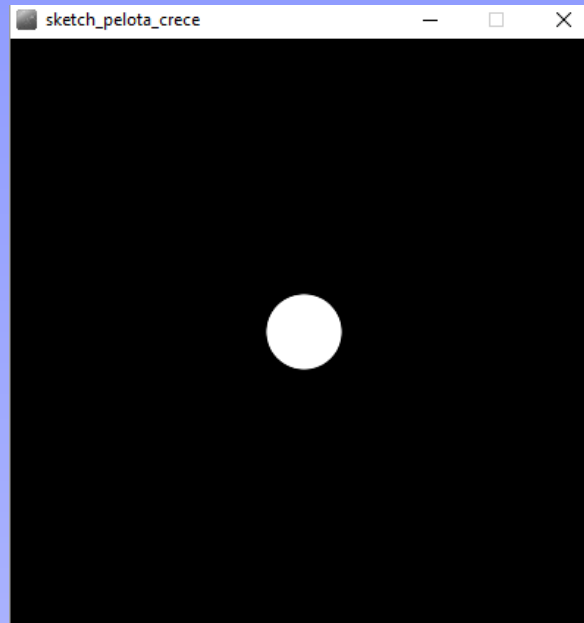
Clave: variable que afecte al diámetro

## ACTIVIDAD:

Pelota crece gradual.

Atención al:

- bloque estático y bloque dinámico
- utiliza variables decimales
- variable que afecte al radio o diámetro



-Muestra de práctica ejecutada en processing

## ACTIVIDAD:

Color gris cambia gradual

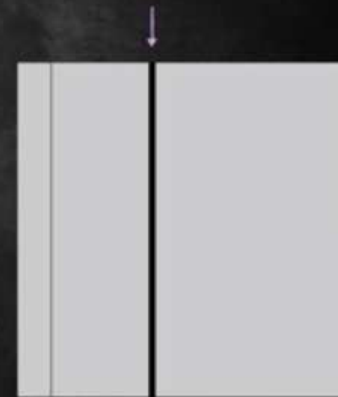
- Plantear una figura simple
- Un fondo blanco
- Contorno de la figura negro
- Relleno de la figura afectada por una variable
- Animación del color de relleno de la figura que inicia en gris oscuro y deriva a blanco.

# OPERADORES NUMÉRICO

Con ellos se realizan operaciones entre dos expresiones del código, devolviendo otro valor. Se utilizan para controlar elementos de la pantalla en cuanto a posición, tamaño...

El símbolo **+** se utiliza para sumar  
El símbolo **-** se utiliza para restar  
El símbolo **\*** se utiliza para multiplicar  
El símbolo **/** se utiliza para dividir

```
size(500, 500);  
int a = 50;  
line(a, 0, a, height);  
a = a + 150;  
strokeWeight(10);  
→ line(a, 0, a, height);
```



a = 200

## ORDEN de las OPERACIONES

El orden de las operaciones determina qué operaciones realizan sus cálculos antes que otras. Por ejemplo, la multiplicación siempre se evalúa antes que la suma con independencia de la secuencia de los elementos.

$$3 + 4 * 5 = 23$$

El orden de las operaciones para evaluar expresiones se puede cambiar añadiendo un paréntesis.

$$(3 + 4) * 5 = 35$$

El operador incremento `++` suma el valor 1 a una variable y el operador de decremento `--` resta el valor de 1.

```
int x = 1;
```

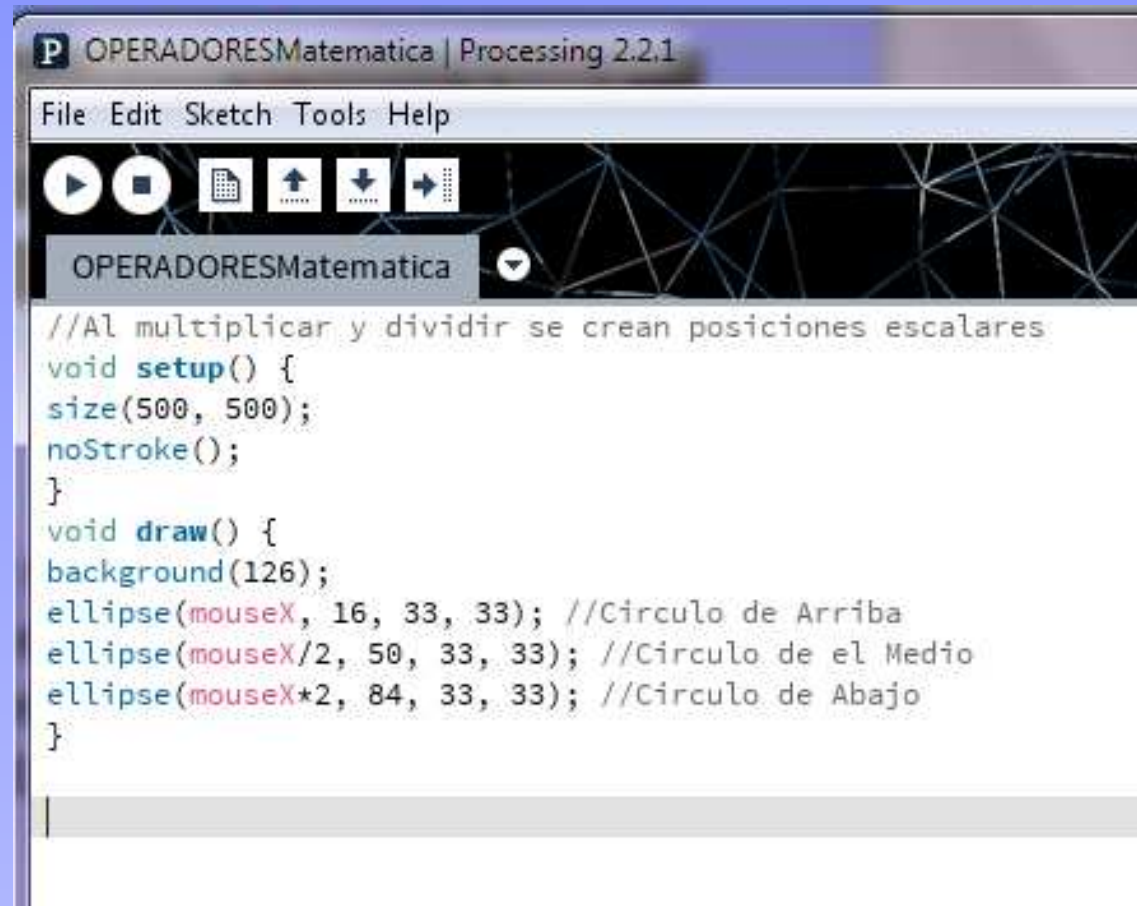
```
x++; // Equivale a x = x + 1
```

```
y--; // Equivale a y = y - 1
```

El operador suma asignación `+=` combina la suma y la asignación. El operador de asignación de resta `-=` combina la asignación con la resta.

```
x+=5; // Equivale a x = x + 5
```

## PRÁCTICA:

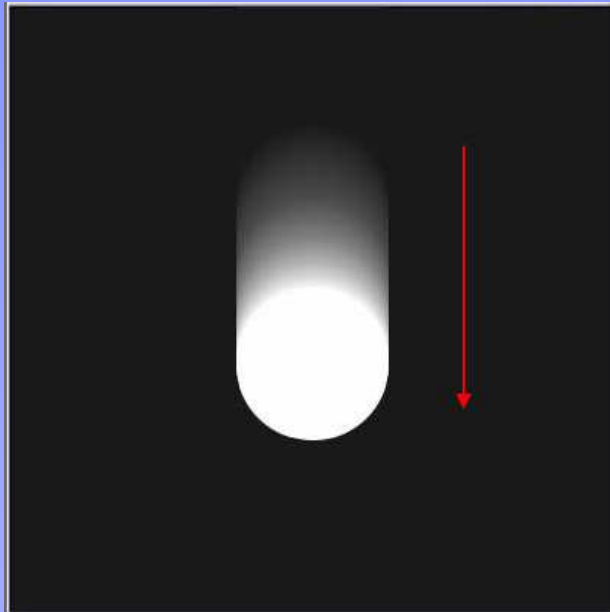


The screenshot shows the Processing IDE interface. The title bar reads 'OPERADORESMatematica | Processing 2.2.1'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for play, stop, new, open, save, and zoom. The sketch name 'OPERADORESMatematica' is displayed in a dropdown menu. The main area contains the following code:

```
//Al multiplicar y dividir se crean posiciones escalares
void setup() {
  size(500, 500);
  noStroke();
}
void draw() {
  background(126);
  ellipse(mouseX, 16, 33, 33); //Circulo de Arriba
  ellipse(mouseX/2, 50, 33, 33); //Circulo de el Medio
  ellipse(mouseX*2, 84, 33, 33); //Circulo de Abajo
}
```

# MOVIMIENTO

Poner una figura en movimiento significa que hay un cambio de posición de la misma. Para generar esto, se debe introducir una variable que varíe la posición de la figura en cualquiera de sus ejes.



-declaración de variable  
que controla la velocidad

-El cambio de posición se va  
a dar en el eje Y

Utilizamos los operadores numéricos

- 1- Pelota cae
- 2- Incremento gradual del diámetro de un círculo.

## ACTIVIDAD:

- Plantear una elipse u otra figura
  - Lienzo de (800,800) por ejemplo
  - Fondo blanco
  - Relleno de la elipse gris oscuro
  - Borde de la elipse negro
- 
- Desplazamiento de la elipse en el eje X



## ACTIVIDAD

Basándonos en el ejemplo anterior:

- Ejecutar una elipse que se desplace en el eje X y a la vez crezca su diámetro gradualmente conforme avanza.

Además que cambie de color en la escala del negro al blanco a la vez que avanza y crece.

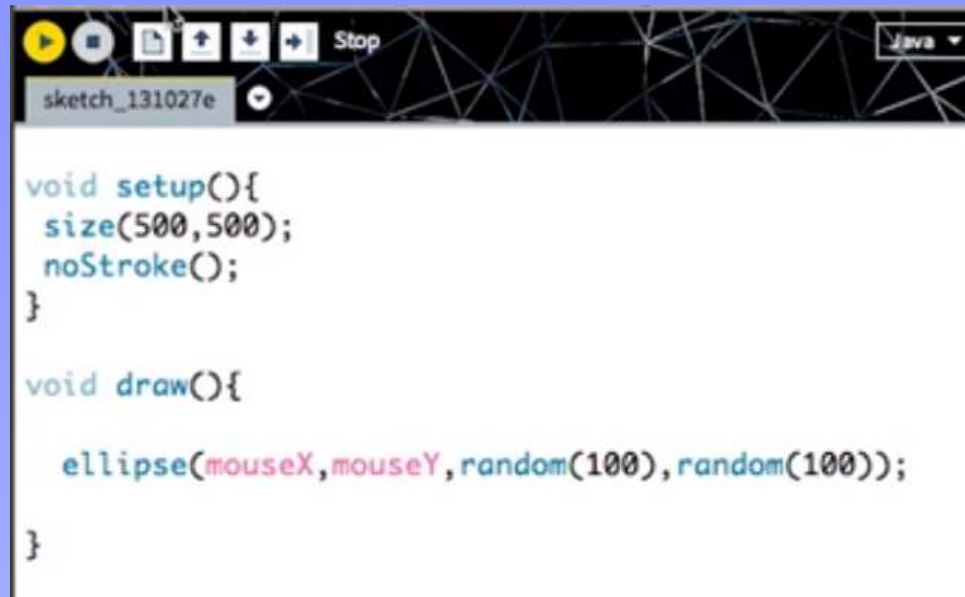
# FUNCIÓN RANDOM

\*aleatorio

\*variedad inesperada a la ejecución en processing

random(max);

random(min,max);



```
void setup(){
  size(500,500);
  noStroke();
}

void draw(){

  ellipse(mouseX,mouseY,random(100),random(100));
}
```

\*\* ACTIVIDAD; random color

Con el ejemplo de arriba, haz que la función random afecte al relleno de la elipse.

## RANDOM aplicado al movimiento:

- Plantea una elipse que cae verticalmente de arriba abajo.
- Aplicamos función random: queremos que mi elipse simule el movimiento de una hoja cayendo.
- Detectamos que la elipse se mueve de derecha a izquierda, pero ahora pretendo que mi hoja simule ese mismo movimiento desplazándose en el eje Y a la vez que cae, para que el movimiento sea más natural.

# ESTRUCTURAS DE CONTROL- CONDICIONALES

\*Dan al programa las pautas para tomar decisiones sobre que líneas de código serán ejecutadas y cuáles no:  
si pasa esto.....haz esto.....

Se lleva a cabo una acción cuando se cumple una condición específica.

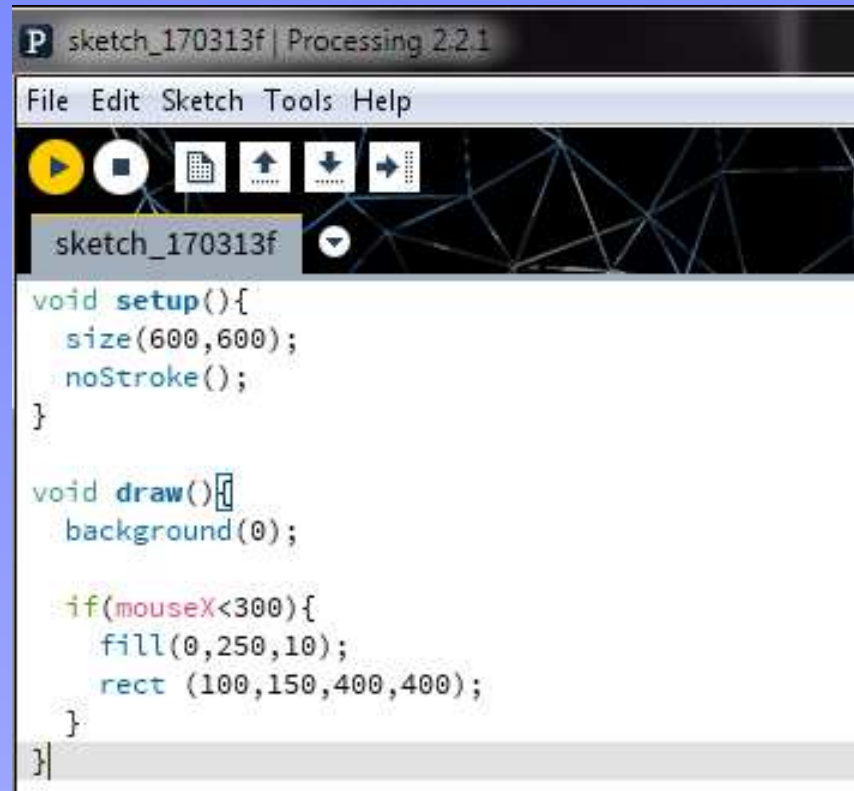
```
if (condicion) {  
    // Código a ejecutar si la condición SI se cumple  
} else {  
    // Código a ejecutar se la condición NO se cumple  
}
```

## OPERADORES DE COMPARACIÓN

```
= es igual a  
!= no es igual a  
< es menor a  
<= es menor o igual a  
> es mayor a  
>= es mayor o igual a
```

```
&& y (a > b) && (c >= a)  
|| o (a > b) || (c >= a)  
! no !(a > b)
```

## Caso de iniciación:



```
void setup(){
  size(600,600);
  noStroke();
}

void draw(){
  background(0);

  if(mouseX<300){
    fill(0,250,10);
    rect (100,150,400,400);
  }
}
```

Va a dibujar un cuadrado con las características descritas si mi posición del ratón (mouseX) es menor que 300. Si no se da esta condición, no dibujará nada.

- **PODEMOS SEGUIR ENCADENANDO CONDICIONALES**
- ¿y si queremos que nuestro cuadrado sea de otro color cuando sobrepase la mitad del lienzo?

## Estructuras condicionales if / else

Cuando la expresión de la estructura, **if**, es falsa, pasa a ejecutar la estructura del bloque, **else**.

```
if(condicion){  
  hazTalCosa();  
}
```

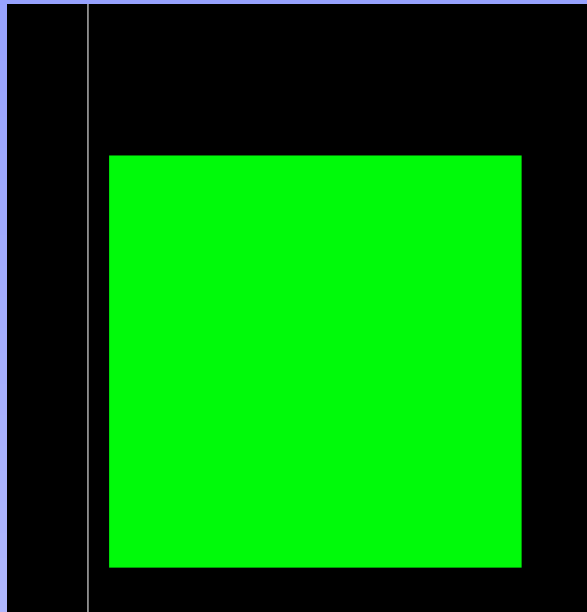


```
if(condicion){  
  hazTalCosa();  
} else {  
  hazTalOtra();  
}
```

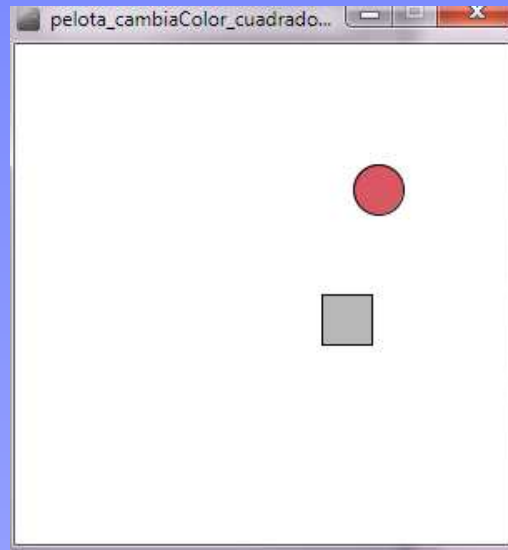
## ACTIVIDAD:

Con el uso de condicional:

- Fondo negro
- Dibuja una línea blanca que siga las coordenadas del ratón en eje X
- Dibuja una elipse (azul) y cuando sobrepase la mitad de mi lienzo con la línea, la elipse se convierte en un rectángulo (rojo).



# ACTIVIDAD DE REPASO



El cuadrado se desplaza empezando en el lado izquierdo de la pantalla y desaparece por el lado derecho.

A la vez del desplazamiento, el cuadrado cambia de color del negro al blanco, pasando por esa escala de grises.

La elipse de forma paralela puede moverse siguiendo los movimiento del mouse en eje X, Y, y cambia de color de forma aleatoria.

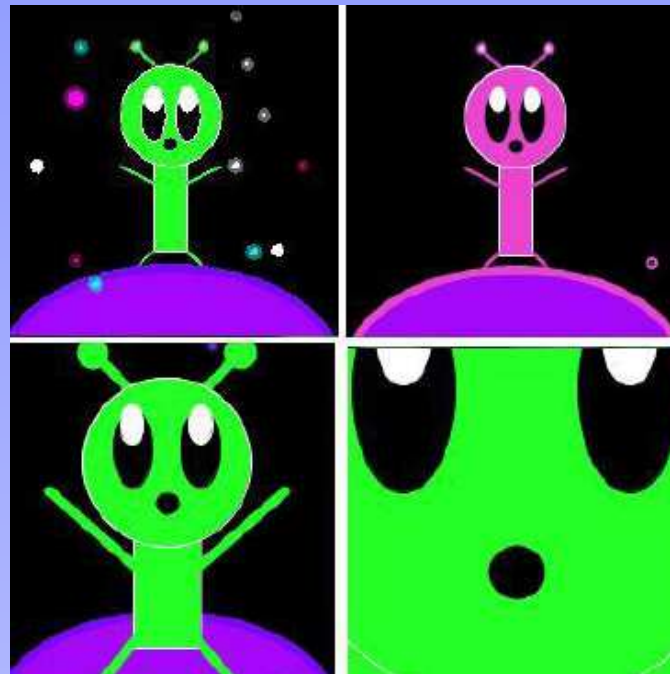
\*\*condicional: reaparece por el lado izquierdo de la pantalla.



## ACTIVIDAD:

### Creación de un Salvapantallas. (libre)

- dibujo de formas
- movimiento
- cambios aleatorios
- color



ejemplo