

# ESTRUCTURAS DE CONTROL- CONDICIONALES

Dan al programa las pautas para tomar decisiones y obrar en consecuencia:

si pasa esto.....haz esto....

- Comprueba si una u otra condición se cumple o no.
- La condición ha de ir entre paréntesis. A continuación, las instrucciones que ponemos desde aquí hasta que cerramos la llave son las que se ejecutarán si la condición se cumple.

```
if( condición ){  
    cuerpo de la estructura  
}
```

## if, then

La estructura if-then permite encerrar una porción de código entre sus llaves. Esa porción de código se ejecuta sólo si se cumple la condición que se encuentra entre los paréntesis:

```
if (condición){  
    hazEsto( )  
}
```

si se cumple A, entonces B.

**hazEsto**, se ejecutará si se cumple la **condición**, sino, no hace nada.

sketch\_condicional1 | Processing 2.2.1

File Edit Sketch Tools Help



sketch\_condicional1

```
void setup(){  
  size(200,200);  
}  
void draw(){  
  background( 0 );  
  fill(255);  
  if( mouseX > 100 ){  
    fill(255,0,127);  
    rect( 50 , 50 , 100 , 100 );  
  }  
}
```

## ACTIVIDAD:

Teniendo de base el ejercicio anterior,

Crear una condicional, a partir de la cual cuando presiono mi ratón, se cumple la condición y se ejecuta un rectángulo que se desplaza.

**Claves:**

Variable int

Variable mousePressed

Condicional; if

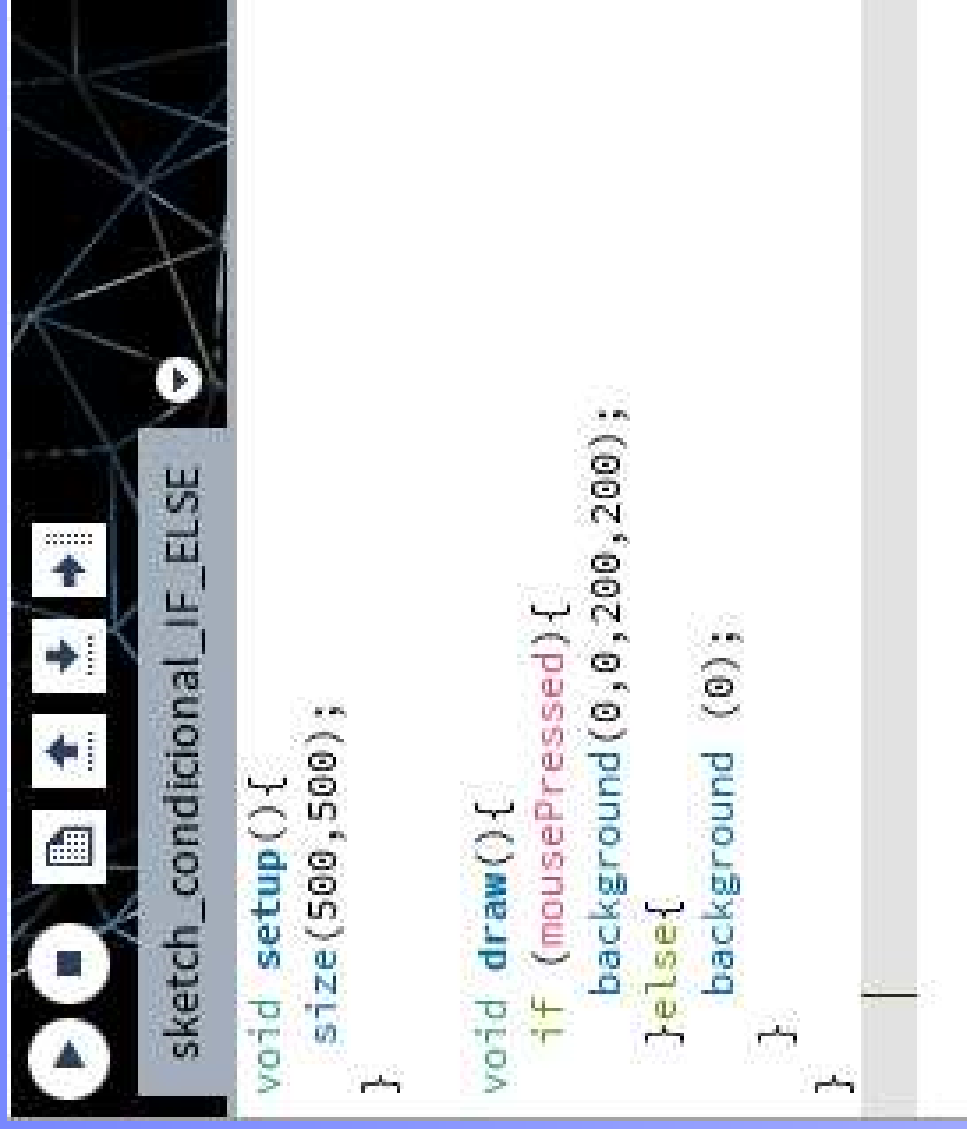
## if, then, else

Si lo que queremos es que según si se cumple o no una condición, se ejecute o no, una cosa u otra, usamos esta estructura:

si-entonces-sino, quiere decir

"si tal cosa, entonces hacer tal cosa, sino hacer tal otra":

```
if( condición ){  
    se ejecuta si la condición es verdadera  
}  
else{  
    se ejecuta si la condición es falsa  
}
```



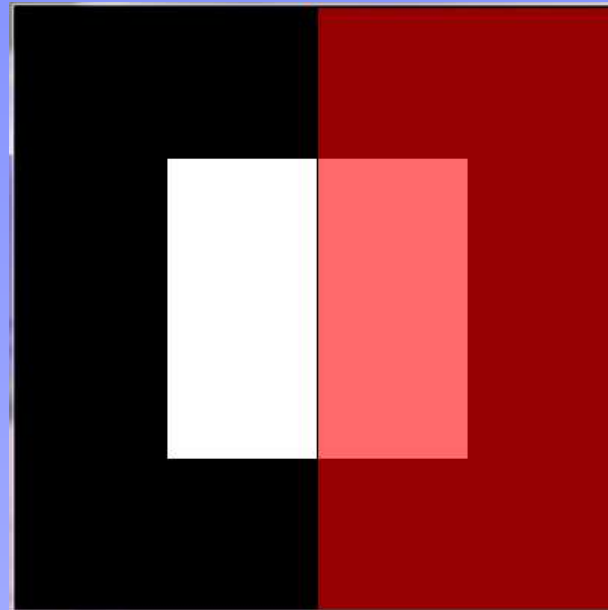
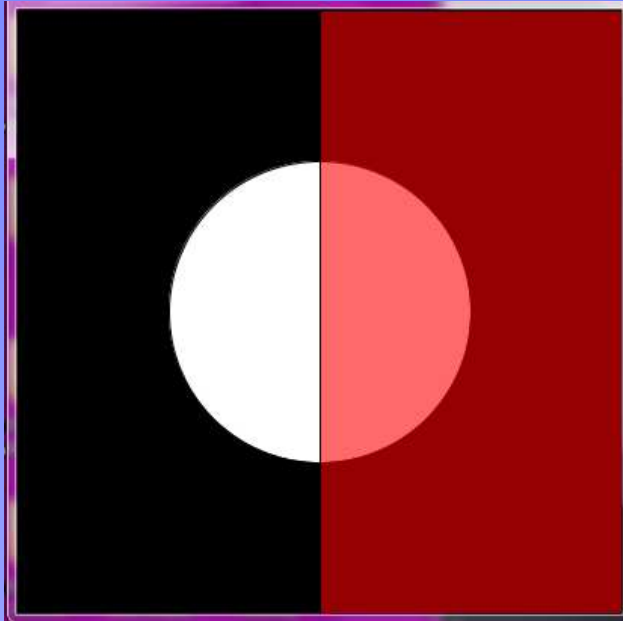
```
sketch_condicional_IF_ELSE

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

void draw() {
  if (mousePressed) {
    background(0, 0, 200, 200);
  } else {
    background (0);
  }
}
```

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

## Actividad de repaso:



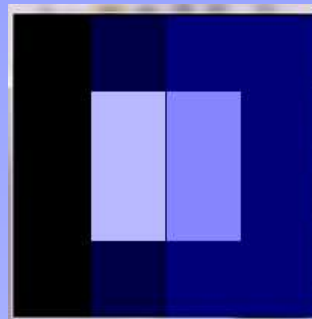
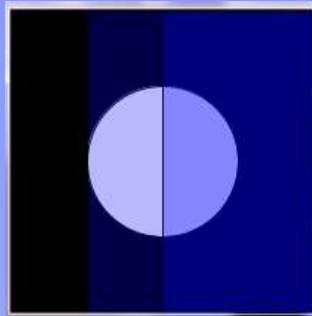
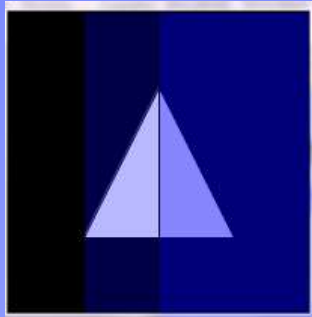
\*\* Condicional con else.

\*\* Cambiamos el fill que afecta a las condicionales de posición.

\*\* Quitamos el último rectángulo. (bloques de códigos)

# Estructura

## if-then-else if-else

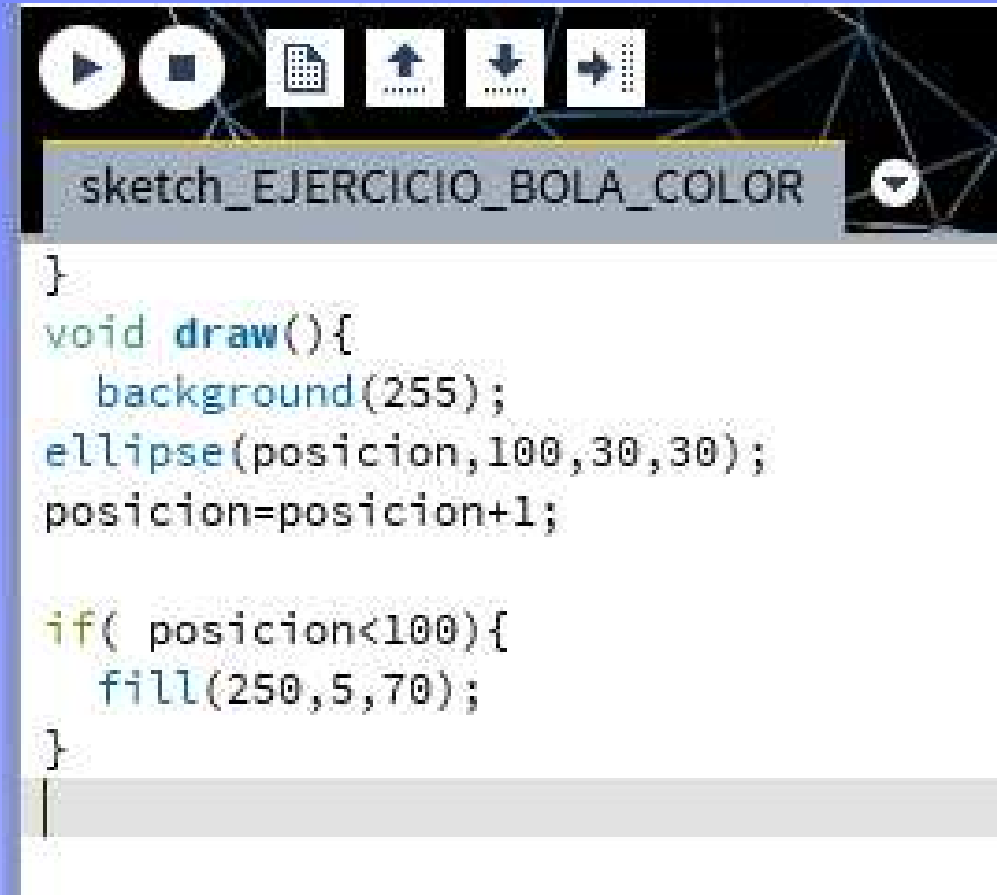


- \*\* fill último
- \*\* orden figuras
- \*\* opacidad

```
sketch_condicional_VARIAS_else
void setup(){
  size(200,200);
}
void draw(){
  background( 0 );
  fill(255);
  if(  ){
    
  }else if(  ){
    
  }else{
    
  }
}
```



## Actividad de repaso:



```
}  
void draw(){  
  background(255);  
  ellipse(posicion,100,30,30);  
  posicion=posicion+1;  
  
  if( posicion<100){  
    fill(250,5,70);  
  }  
}
```

Que una pelota en movimiento sobre el eje X, cambien de color; rojo, verde y azul, sucesivamente, a medida que avanza a lo largo del eje x.

**LA CONDICIÓN:** es cualquier operación que devuelve un valor lógico, y los valores lógicos son los que se representan como verdadero o falso.

\*\* lo que el **if** se pregunta es si el valor de la condición es verdadero o falso **true-false**

Las condicionales están compuestas por valores.

## **OPERADORES RELACIONALES:**

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

## sketch\_movimiento\_condicional\_simple

```
int posicionX=0;
void setup(){
  size(400,200);
  background(255);
}
void draw(){
  ellipse(posicionX,100,30,30);
  posicionX=posicionX+1;
  if (posicionX>width/2){
    background(0);
  }
}
```

# OPERADORES LÓGICOS:

Estos comparan condiciones y operan sobre valores de verdadero o falso.

datos booleanos

<b>&amp;&amp;</b>	<b>y</b>	<b>(a &gt; b) &amp;&amp; (c &gt;= a)</b>
<b>  </b>	<b>o</b>	<b>(a &gt; b)    (c &gt;= a)</b>
<b>!</b>	<b>no</b>	<b>!(a &gt; b)</b>

**&&** ("y")- and

El "y" lógico compara dos valores booleanos y devuelve verdadero solamente si los dos valores comparados son verdaderos.

**||** ("o")- or

El "o" lógico compara dos valores booleanos y devuelve verdadero si uno de los dos valores comparados son verdaderos.

**!** ("no") not

El no lógico no compara valores sino que los invierte.

```
sketch_131101a
void setup() {
  size(500, 500);
}

void draw() {

  if (mousePressed && mouseX > width/2) {
    background(255, 0, 0);
  }
  else {
    background(0);
  }
}
```

Las dos condiciones verdaderas

```
sketch_131101a  
  
void setup() {  
  size(500, 500);  
}  
  
void draw() {  
  
  if (mousePressed || mouseX > width/2) {  
    background(255, 0, 0);  
  }  
  else {  
    background(0);  
  }  
}
```

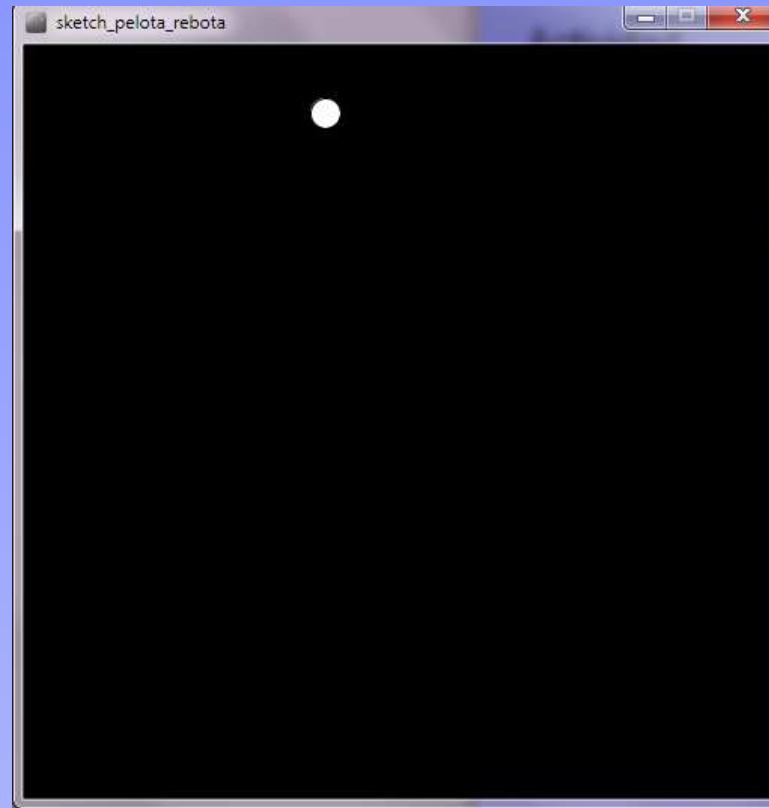
Cualquiera de las condiciones, a la derecha o a la izquierda del “| |” es verdadera, y se ejecutará la condición expresada en el código.

```
sketch_131101a  
  
void setup() {  
  size(500, 500);  
}  
  
void draw() {  
  
  if (!mousePressed) {  
    background(255, 0, 0);  
  }  
  else {  
    background(0);  
  }  
}
```

Se invierten los valores.

Actividad:

Haciendo uso de los operadores lógicos, plantea el código que permita que una esfera rebote en la parte superior e inferior de la pantalla.





## ACTIVIDAD repaso condicionales

Una elipse crece a modo de globo desde del centro de mi lienzo. En el centro de mi ejeY tengo un rectángulo. Cuando la elipse crece hasta tocar el rectángulo, cambia de color.

