



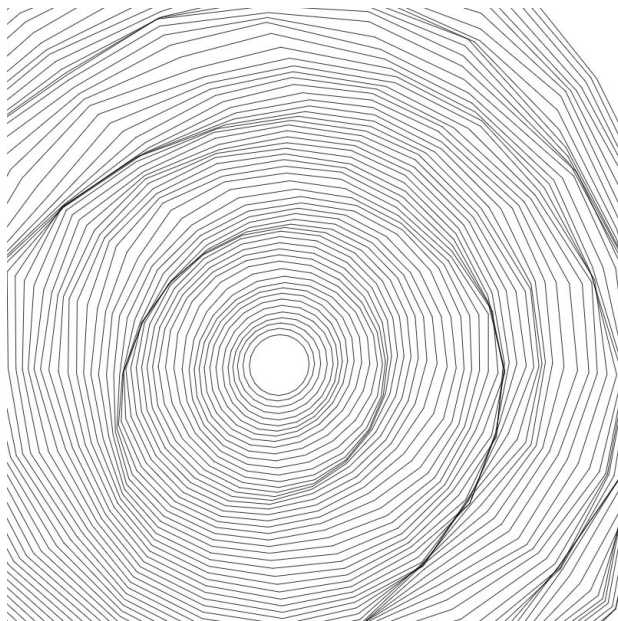
PPEM 2022

Introducción a Processing



Círculos bailando

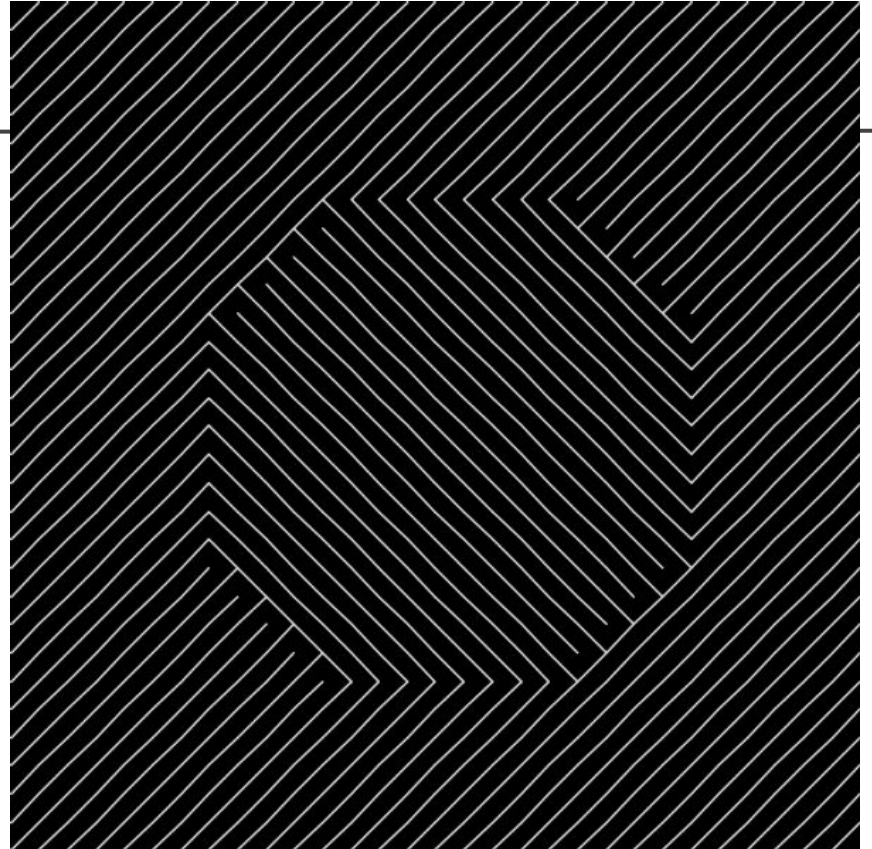
<https://www.generativehut.com/post/recreating-the-noise-orbit>





Ameba rayada

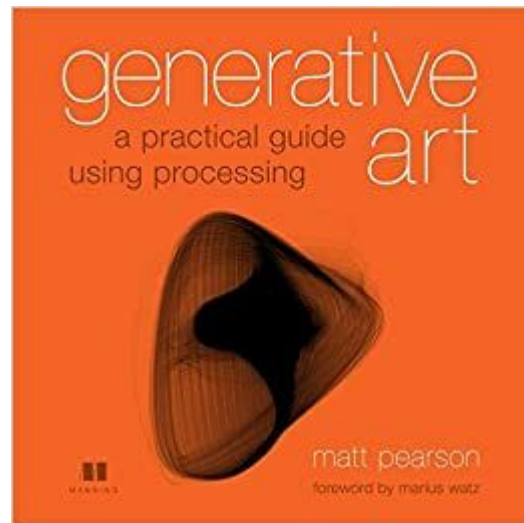
<https://www.generativehut.com/post/using-noise-to-create-looping-gifs-on-processing>





Libro de arte generativo

<https://www.mat.ucsb.edu/~g.legrady/academic/courses/20f594/txt/generativeArt2.pdf>





Arte generativo

<https://aiartists.org/generative-art-design#:~:text=Generative%20Art%20is%20a%20process,new%20works%20on%20your%20behalf.>



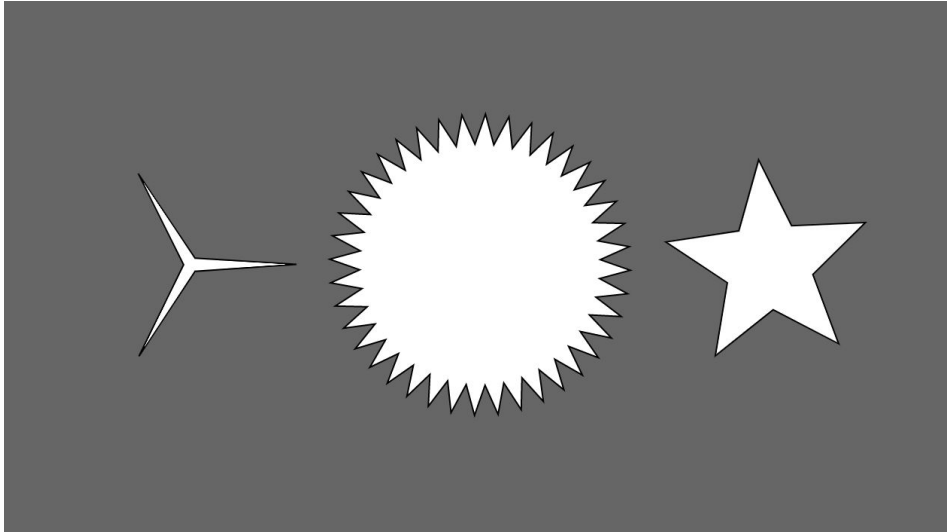
Temario

- Settings, setup y draw
- Tweak Mode
- Sistema de coordenadas (translate, rotate, scale, pushMatrix/popMatrix)
- Formas geométricas: Line, Point, Rect, Quad, Ellipse, Arc, Triangle, Vertex
- **Espacios de colores (RGB, HSV, notación hexadecimal)**
- **Eventos de mouse y teclado**
- **Debug**
- **Librerías externas**



Un ejemplo con lo que vimos hasta ahora

<https://processing.org/examples/star.html>

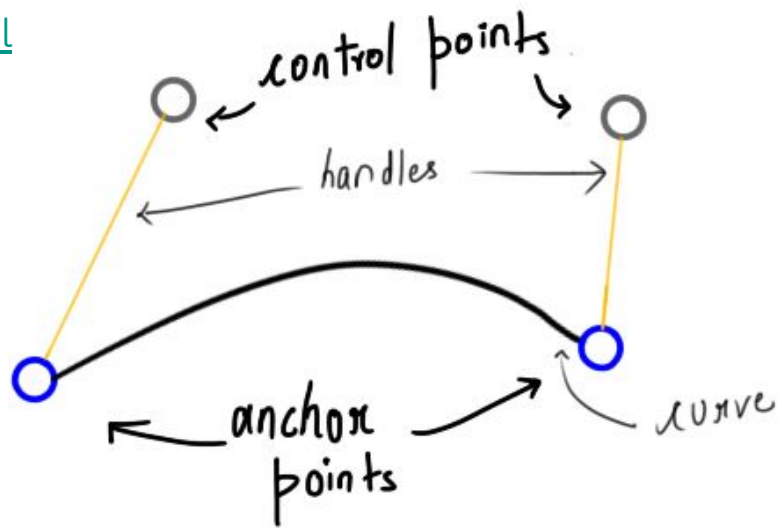
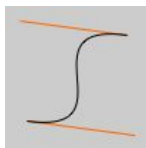


curveVertex, quadraticVertex, bezier

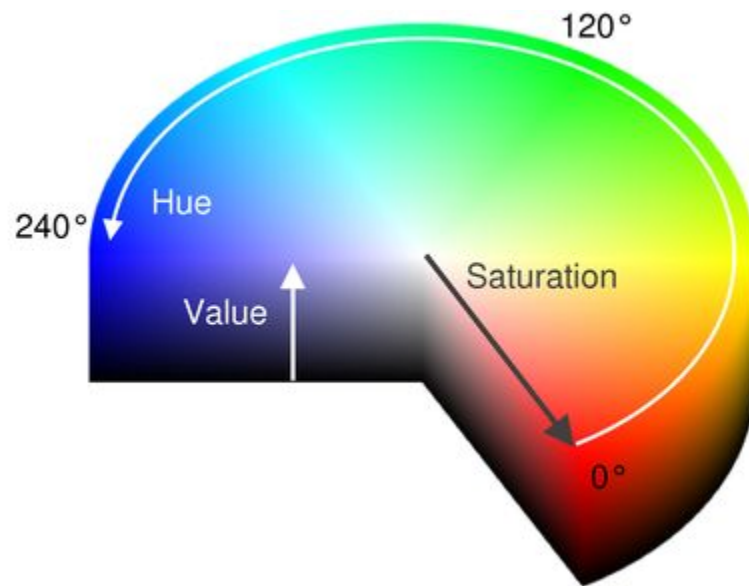
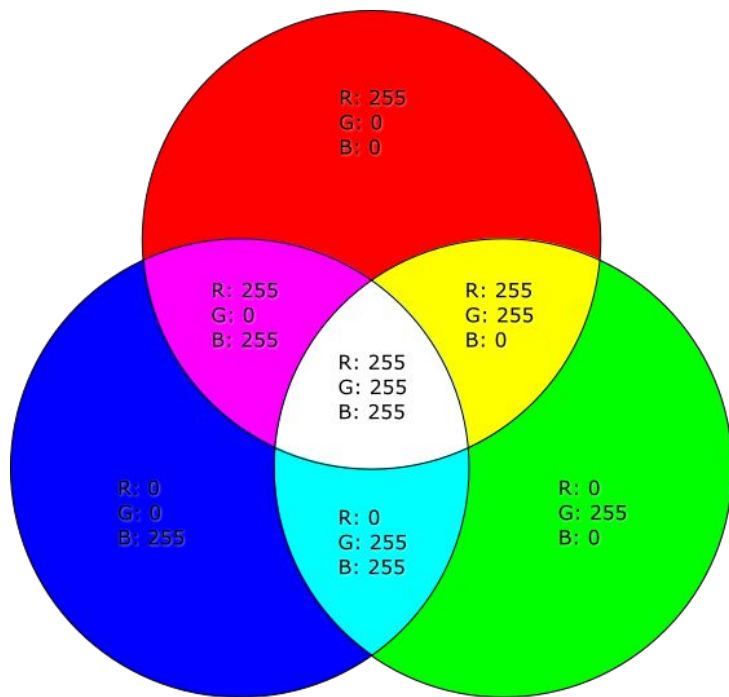
https://processing.org/reference/curveVertex_.html

https://processing.org/reference/quadraticVertex_.html

https://processing.org/reference/bezier_.html



Espacio de colores RGB y HSB (o HSV)



Espacio de colores RGB

`colorMode(`**RGB**`)` // por defecto todos los valores (RGBA) con max 255

`colorMode(`**RGB**`, max)` // para setear valor máximo para todos los canales

`colorMode(`**RGB**`, maxR, maxG, maxB)` // para setear por separado RGB

`colorMode(`**RGB**`, maxR, maxG, maxB, maxA)` // para setear RGBA

`color(gray)` // por defecto 0 para negro y 255 para blanco

`color(gray, alpha)` // alpha en 0 para transparente, 255 para sólido

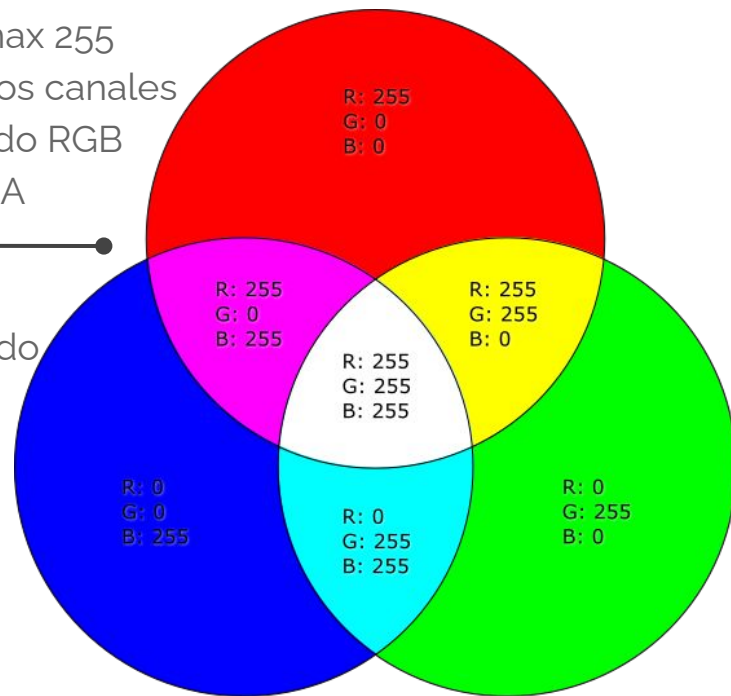
`color(r, g, b)`

`color(r, g, b, alpha)`

ejemplos:

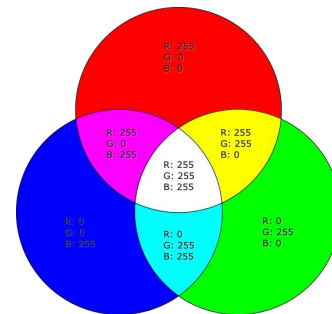
`color(240)` // gris clarito

`color(255,0,0)` // rojo

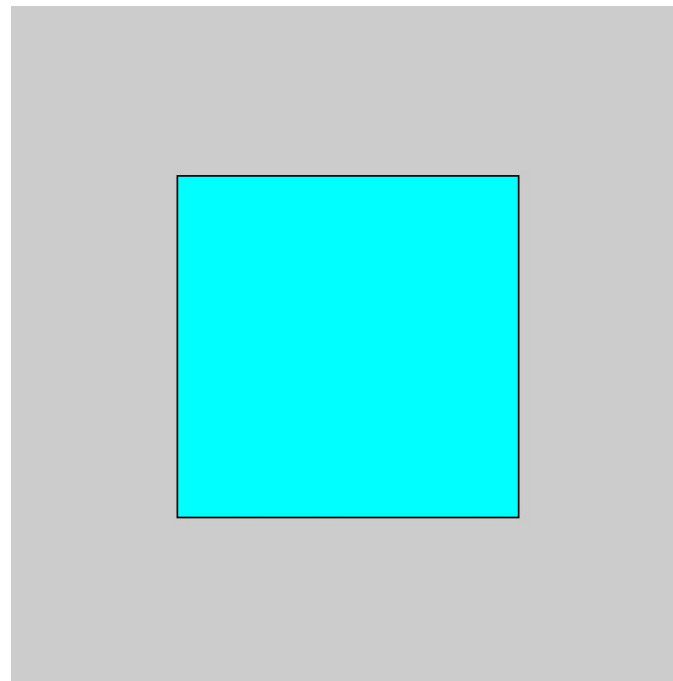


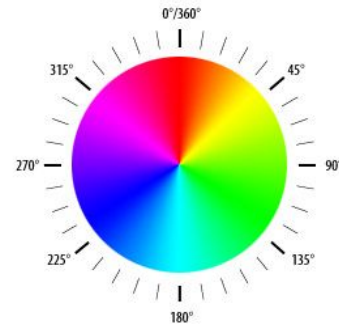


Ejemplo RGB



```
void setup(){  
  size(400,400);  
}  
void draw(){  
  color fillColor = color(0, 255, 255);  
  fill(fillColor);  
  rect(100,100,200,200);  
}
```





Espacio de colores HSB

`colorMode(HSB)` // por defecto todos los valores (HSBA) con **max 255**

`colorMode(HSB, max)` // para setear valor máximo para todos los canales

`colorMode(HSB, maxH, maxS, maxB)` // para setear por separado HSB

`colorMode(HSB, maxH, maxS, maxB, maxA)` // para setear HSBA

`color(gray)` // por defecto 0 para negro y 255 para blanco

`color(gray, alpha)` // alpha 0 para transparente, 255 para sólido

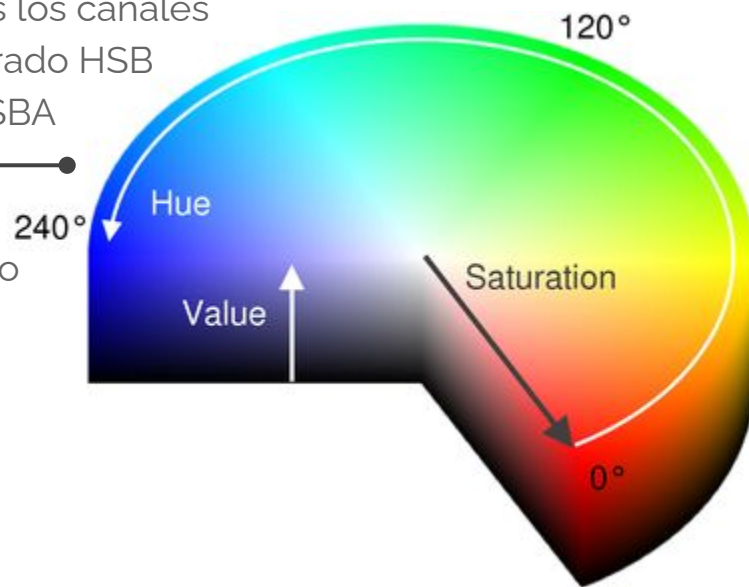
`color(h, s, b)`

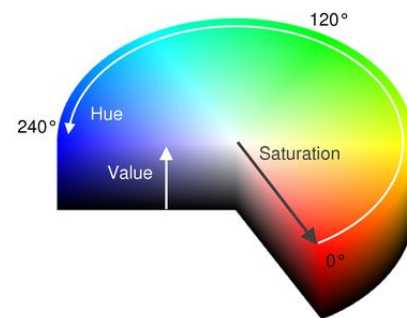
`color(h, s, b, alpha)`

ejemplos:

`color(240)` // gris clarito

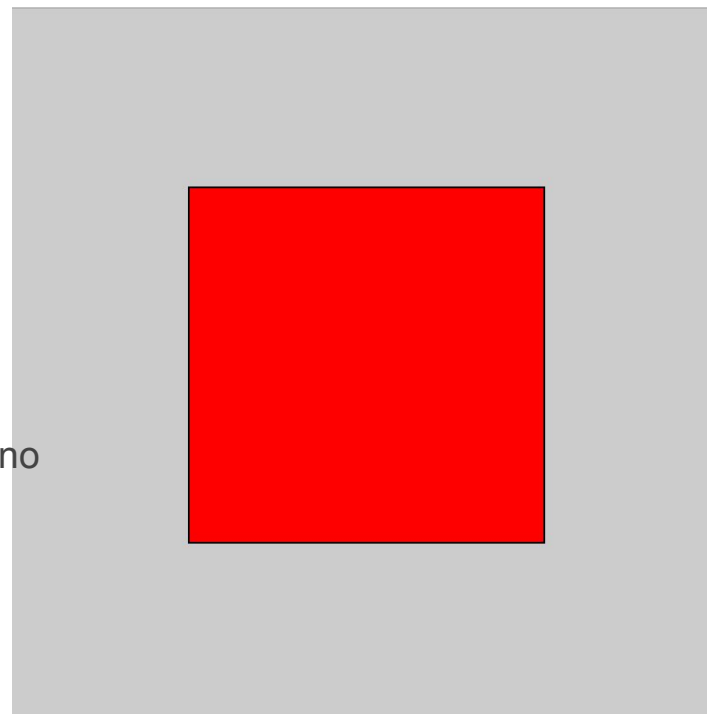
`color(255,0,0)` // negro

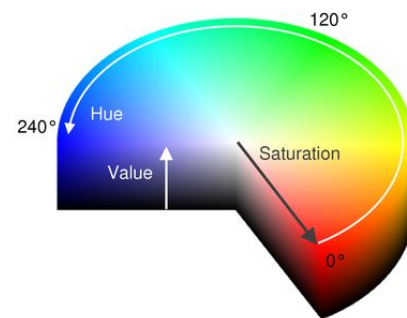




Ejemplo HSB

```
void setup(){  
  colorMode(HSB);  
  size(400,400);  
}  
void draw(){  
  fill(0,255,255); // otra manera de definir el color del relleno  
  rect(100,100,200,200);  
}
```





Ejemplo HSB

```
void setup(){  
  colorMode(HSB);  
  size(400,400);  
}  
void draw(){  
  fill(mouseX,255,255);  
  rect(mouseX,100,200,200);  
}
```

¿Qué hace este código?



Ejemplo más divertido

<https://processing.org/examples/brightness.html>

¿Qué está pasando?



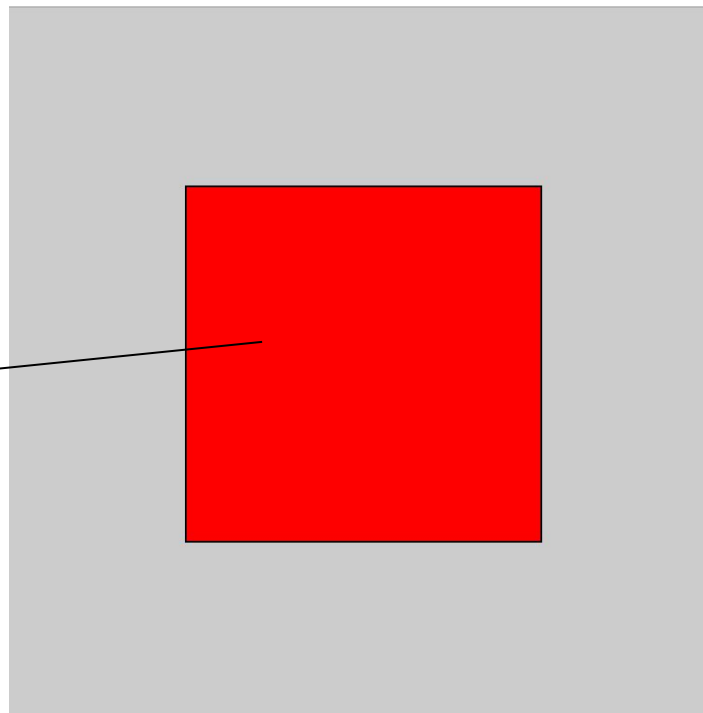
Notación hexadecimal

No cambia aunque cambie el colorMode!

black #000000	gray #808080	silver #c0c0c0	white #ffffff
navy #000080	blue #0000ff	teal #008080	aqua #00ffff
green #008000	lime #00ff00	olive #808000	yellow #ffff00
maroon #800000	red #ff0000	purple #800080	fuchsia #ff00ff

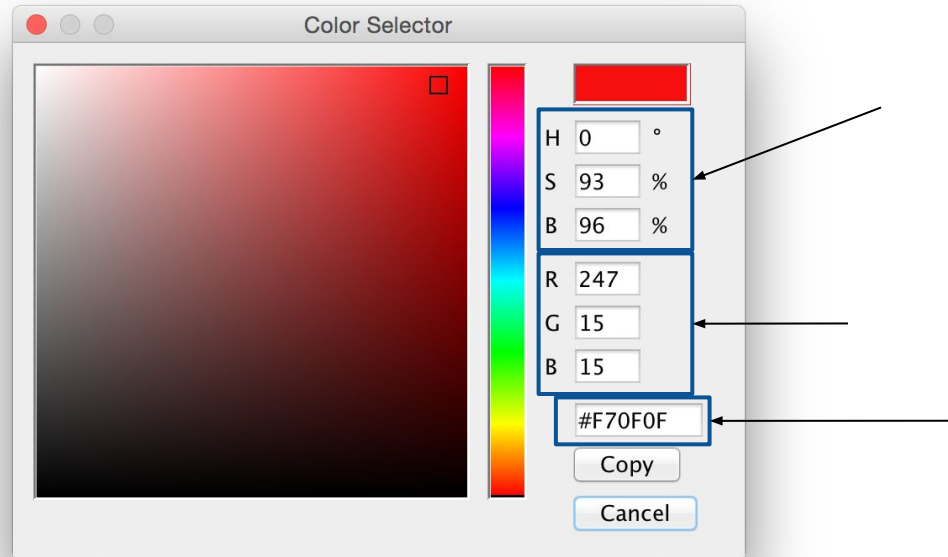
Ejemplo hexadecimal

```
color c = #FF0000;
void setup(){
  colorMode(HSB);
  size(400,400);
}
void draw(){
  fill(c);
  rect(100,100,200,200);
}
```



Color selector

Tools -> Color Selector...





Funciones que reciben colores

`fill()` // relleno de las figuras, opuesto `noFill()`

`stroke()` // contorno, opuesto `noStroke()`

`background()` // fondo

```
fill(colOrHex); // color r = color(255,0,0); fill(r); == fill(#ff0000);  
fill(colOrHex, alpha);  
fill(gray); //  
fill(gray, alpha);  
fill(v1, v2, v3);  
fill(v1, v2, v3, alpha);
```



Consultar los valores RGBA y HSBA

```
int r = red(colorOrHex);
```

```
int g = green(colorOrHex);
```

```
int b = blue(colorOrHex);
```

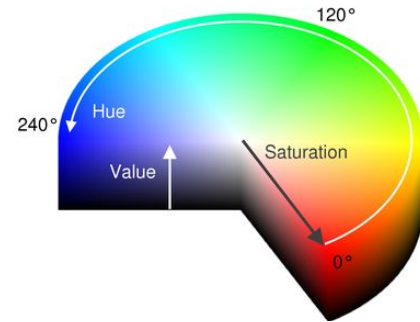
```
int h = hue(colorOrHex);
```

```
int s = saturation(colorOrHex);
```

```
int b = brightness(colorOrHex);
```

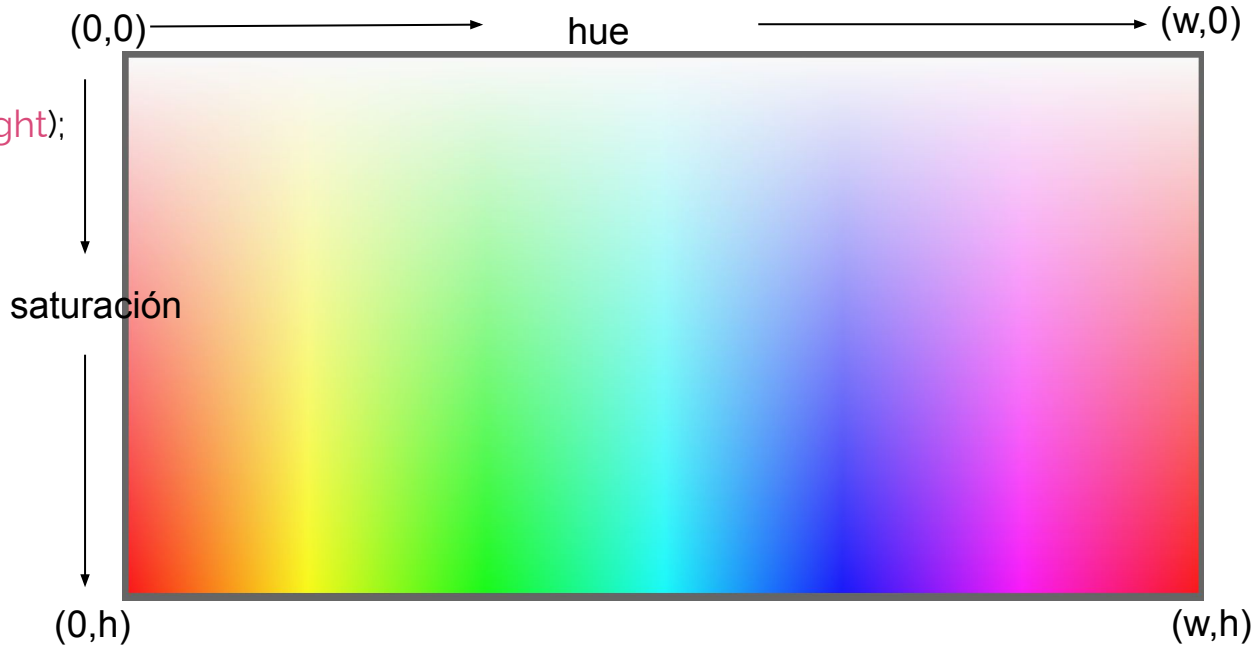
```
int a = alpha(color);
```

```
String hexString = hex(color);
```



Ejemplo HSB

```
size(800,400);  
noStroke();  
colorMode(HSB, width,height,height);  
for (int i = 0; i < width; i++) {  
  for (int j = 0; j < height; j++) {  
    stroke(i, j, height);  
    point(i, j);  
  }  
}
```

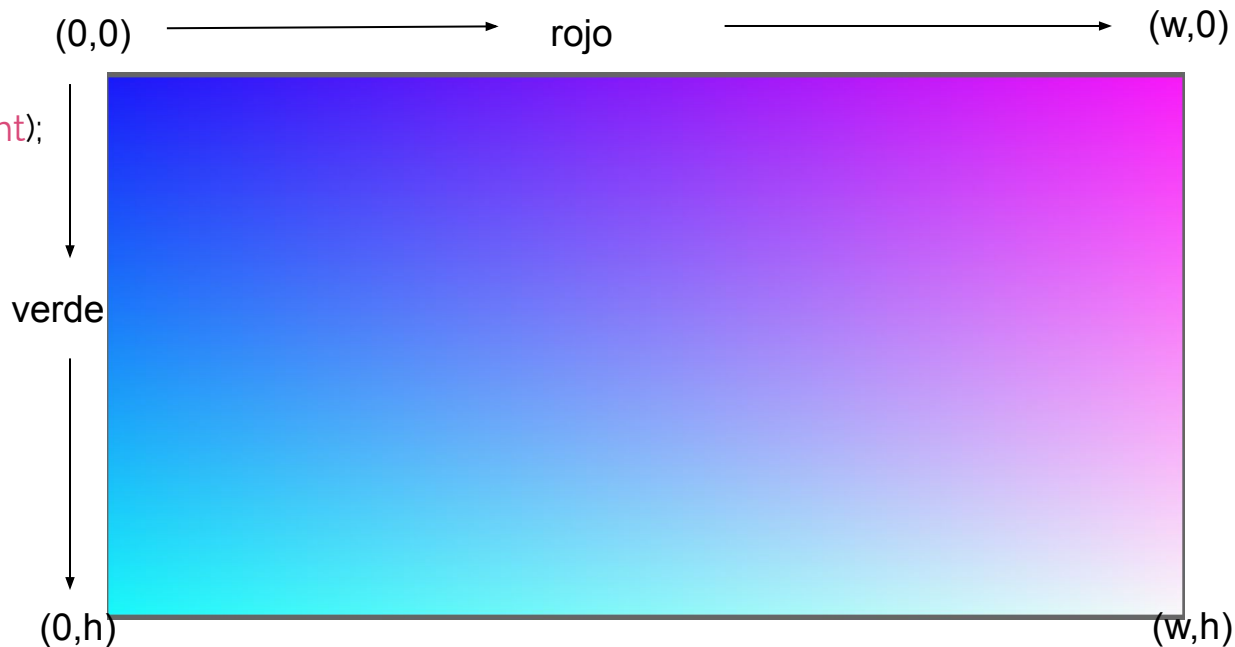


//brillo está en su máximo valor!

Ejemplo RGB (sólo cambió el colorMode)

```
size(800,400);  
noStroke();  
colorMode(RGB, width,height,height);  
for (int i = 0; i < width; i++) {  
  for (int j = 0; j < height; j++) {  
    stroke(i, j, height);  
    point(i, j);  
  }  
}
```

//azul está en su máximo valor!





lerpColor para transiciones entre colores

<https://processing.org/examples/lineargradient.html>

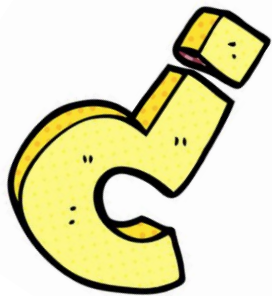


lerpColor en acción

```
color from;  
color to;  
color fillColor;  
float move;
```

```
void setup() {  
  size(600, 600);  
  colorMode(HSB);  
  background(0);  
  from = color(random(255),255,255);  
  to = color(random(255),255,255);  
  move = 0.01;  
}
```

```
void draw() {  
  if(move > 1){  
    move = 0;  
    from = to;  
    to = color(random(255),255,255);  
  }else{  
    move+=0.01;  
  }  
  fillColor = lerpColor(from,to,move);  
  fill(fillColor);  
  ellipse(mouseX,mouseY,60,60);  
}
```

¿Dudas?





Eventos de mouse

mouseX

mouseY

pmouseX

pmouseY

mousePressed

mousePressed()

mouseReleased()

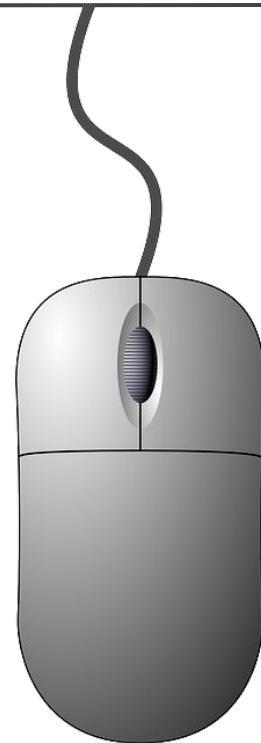
mouseClicked()

mouseMoved()

mouseDragged()

mouseButton

mouseWheel()



Posición del mouse

Posición en el frame anterior

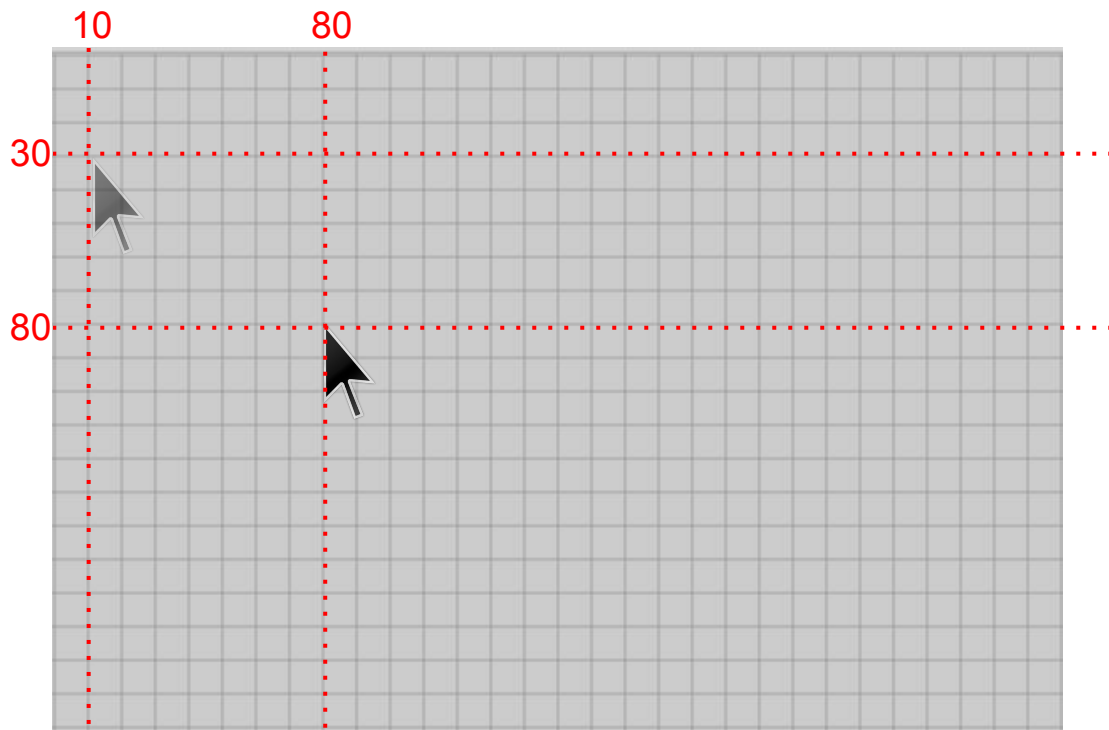
$pmouseX = 10$

$pmouseY = 30$

Posición en el frame actual

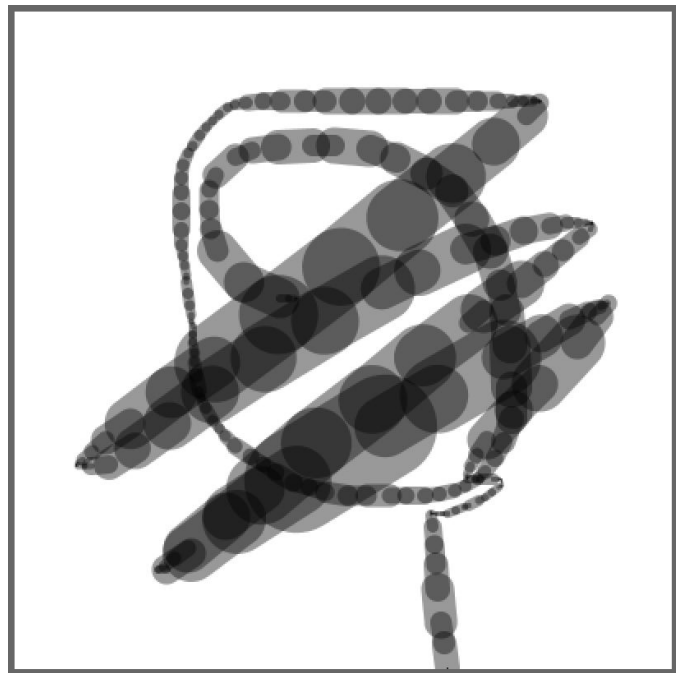
$mouseX = 80$

$mouseY = 80$



Líneas con el mouse

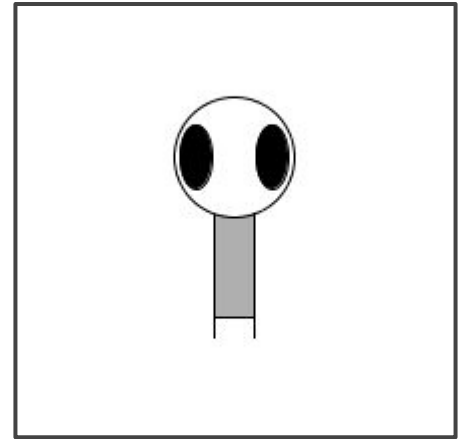
```
float grosor;  
  
void setup(){  
  size(400,400); //fullScreen();  
  stroke(0,102);  
  background(255);  
  grosor = 1;  
}  
  
void draw() {  
  grosor = dist(mouseX,mouseY,pmouseX,pmouseY);  
  strokeWeight(grosor);  
  line(mouseX,mouseY,pmouseX,pmouseY);  
}  
  
void mouseClicked(){  
  background(255);  
}
```





Zoog

<http://learningprocessing.com/examples/chp03/example-03-06-interactive-zoog>





Eventos del mouse

`mousePressed` // booleano, ejemplo de uso -> `if (mousePressed && (mouseButton == LEFT)) fill(o);`

`mousePressed()` // al apretar el botón

`mouseReleased()` // al soltar el botón

`mouseClicked()` // se llama despues de apretar y soltar

`mouseMoved()` // al mover sin tener el botón apretado

`mouseDragged()` // al mover teniendo el botón apretado

`mouseButton` // LEFT, RIGHT o CENTER, para distinguir cual de los botones fue apretado

`mouseWheel()` // al mover la ruedita del mouse

Podemos implementar las funciones: `mousePressed`, `mouseReleased`, `mouseClicked`, `mouseMoved`, `mouseDragged` y `mouseWheel`

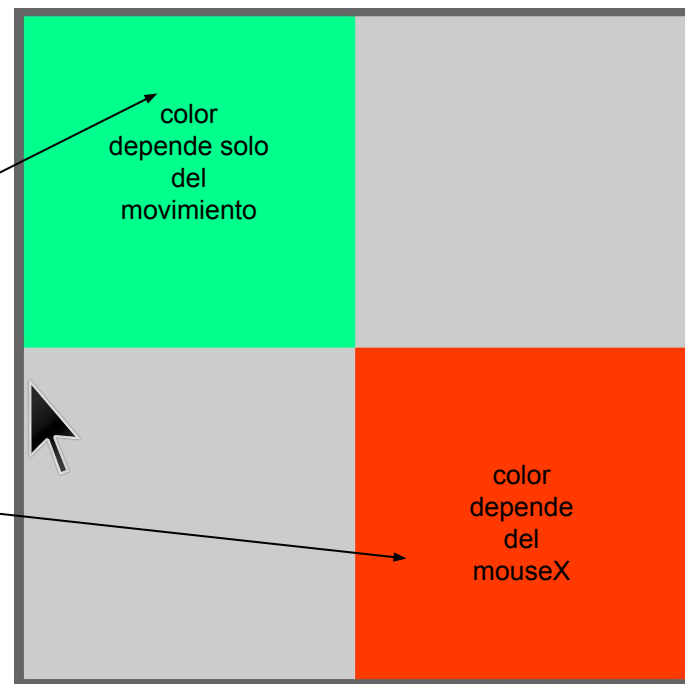
Fondo de colores

```
int hueValue;
void setup(){
  size(400,400);
  colorMode(HSB,width,255,255);
  hueValue = 0;
  noStroke();
}
```

```
void draw() {
  fill(hueValue,255,255);
  rect(0,0,200,200);
```

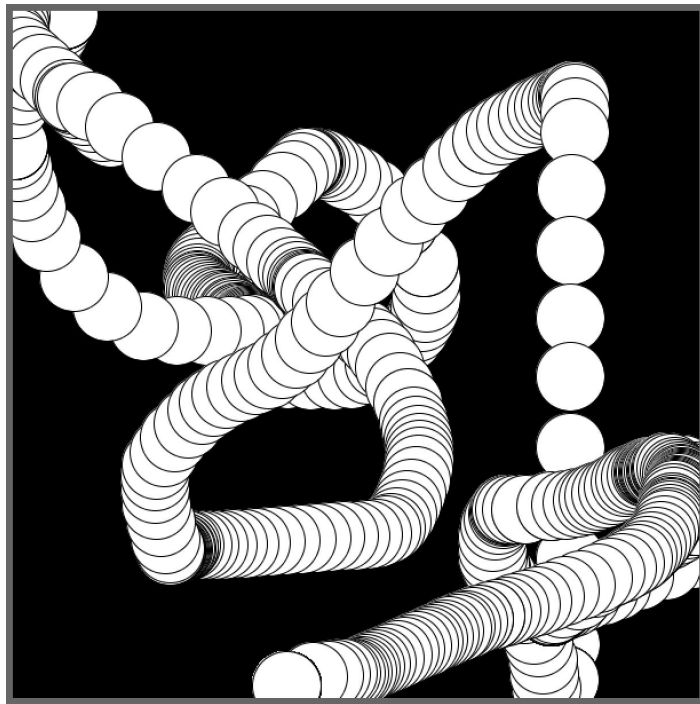
```
  translate(200,200);
  fill(mouseX,255,255);
  rect(0,0,200,200);
}
```

```
void mouseMoved() {
  hueValue = hueValue + 5;
  if (hueValue > width) {
    hueValue = 0;
  }
}
```



Figuras que siguen el mouse

```
void setup() {  
  size(600, 600);  
  background(0);  
  fill(255);  
}  
  
void draw() {  
  ellipse(mouseX,mouseY,60,60);  
}
```

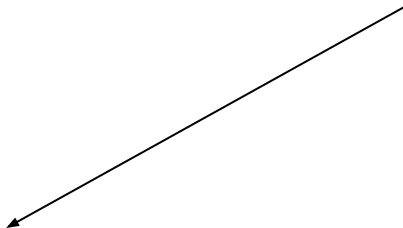




Figuras que siguen el mouse 2

```
void setup() {  
  size(600, 600);  
  background(0);  
  fill(255);  
}  
  
void draw() {  
  background(0);  
  ellipse(mouseX,mouseY,60,60);  
}
```

cambio!



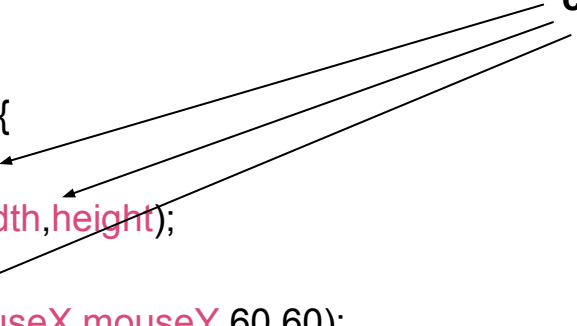
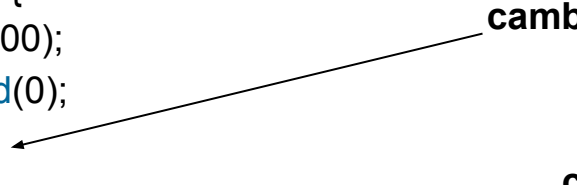


Figuras que siguen el mouse 3

```
void setup() {  
  size(600, 600);  
  background(0);  
  noStroke();  
}  
  
void draw() {  
  fill(0,10);  
  rect(0,0,width,height);  
  fill(255);  
  ellipse(mouseX,mouseY,60,60);  
}
```

cambio!

cambio!





Easing

<https://processing.org/examples/easing.html>

Eventos del teclado

key

keyCode

keyPressed

keyPressed()

keyReleased()



key == 'a', 'A', 'b', 'B', ... + BACKSPACE, TAB, ENTER, RETURN, ESC, and DELETE

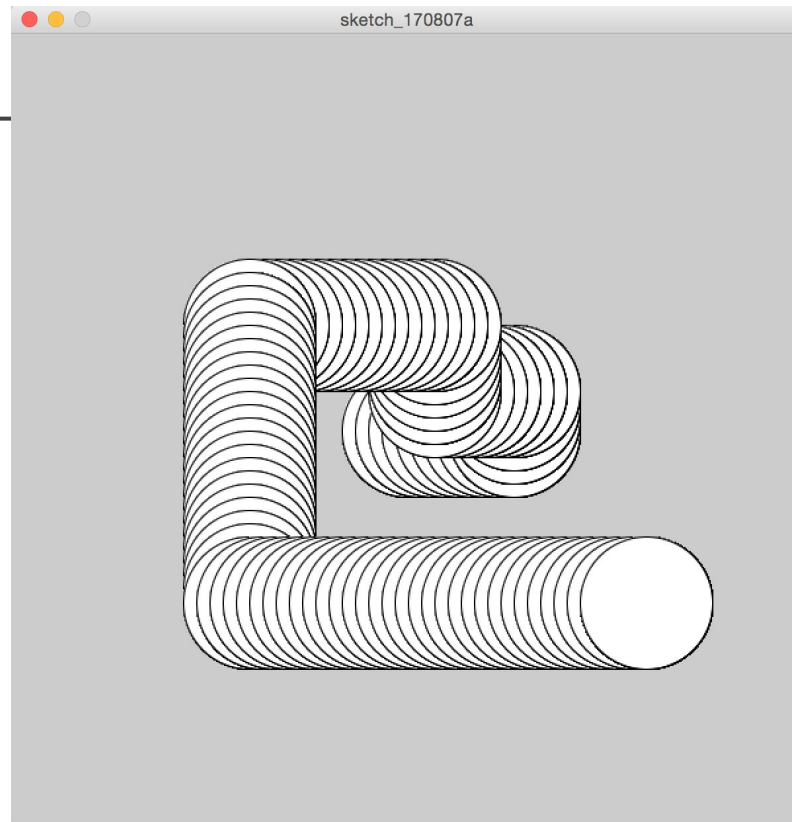
keyCode == UP, DOWN, LEFT, RIGHT, ALT, CONTROL, SHIFT

Mover con las flechas

```
int x,y;
void setup(){
  size(600,600);
  x = width/2;
  y = height/2;
}

void draw(){
  ellipse(x,y,100,100);
}

void keyPressed(){
  if(keyCode == UP){
    y -=10;
  }else if(keyCode == DOWN){
    y +=10;
  }else if(keyCode == RIGHT){
    x +=10;
  }else if(keyCode == LEFT){
    x -=10;
  }
}
```



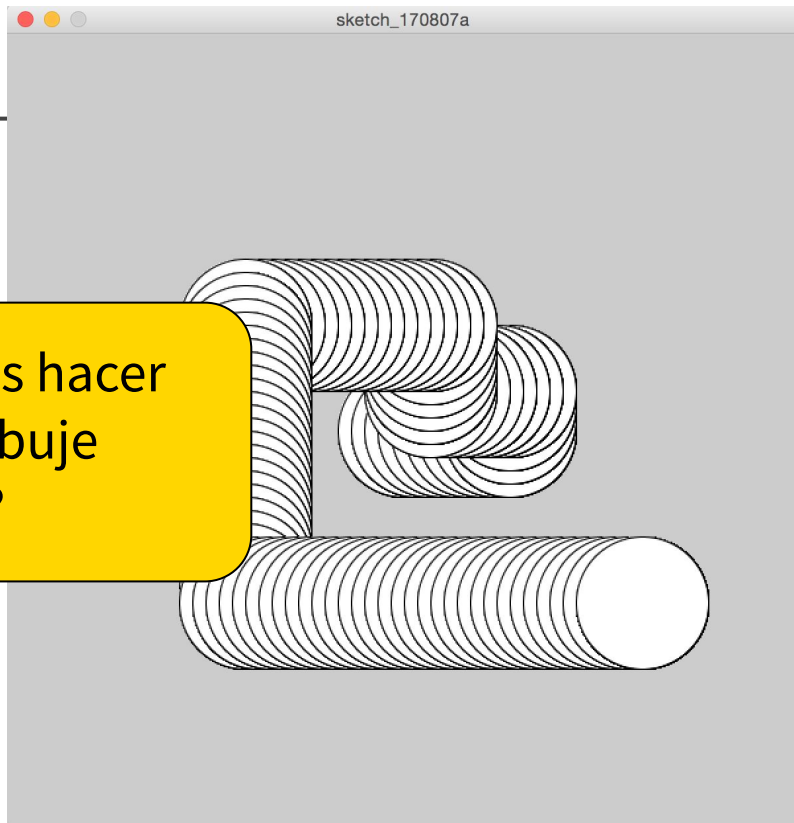
Mover con las flechas

```
int x,y;
void setup(){
  size(600,600);
  x = width/2;
  y = height/2;
}

void draw(){
  ellipse(x,y,100,100);
}

void keyPressed(){
  if(keyCode == UP){
    y -=10;
  }else if(keyCode == DOWN){
    y +=10;
  }else if(keyCode == RIGHT){
    x +=10;
  }else if(keyCode == LEFT){
    x -=10;
  }
}
```

¿Cómo podemos hacer
para no se dibuje
siempre?





Teclas de colores

<https://processing.org/examples/keyboardfunctions.html>

Debug

The screenshot shows the Processing IDE interface. The code editor displays the following code:

```
25 }
26 }
27
28 void dragSegment(int i, float xin, float yin) {
29     float dx = xin - x[i];
30     float dy = yin - y[i];
31     float angle = atan2(dy, dx);
32     x[i] = xin - cos(angle) * segLength;
33     y[i] = yin - sin(angle) * segLength;
34     segment(x[i], y[i], angle);
35 }
36
37 void segment(float x, float y, float a) {
38     pushMatrix();
39     translate(x, y);
40     rotate(a);
41     line(0, 0, segLength, 0);
42     popMatrix();
43 }
```

The line `float dy = yin - y[i];` is highlighted in blue, indicating a breakpoint. The status bar at the bottom of the code editor shows "Depurador detenido." (Debugger stopped).

The Variables window on the right shows the following variables and their values:

Nombre	Valor
i	0
xin	0.0
yin	0.0
dx	0.0
x	instance of float[20]
y	instance of float[20]
segLength	18.0
Processing	
width	640
height	360
mouseX	0
mouseY	0
pmouseX	0
pmouseY	0
key	
keyCode	0
keyPressed	false
focused	true
frameRate	60.12769
frameCount	1



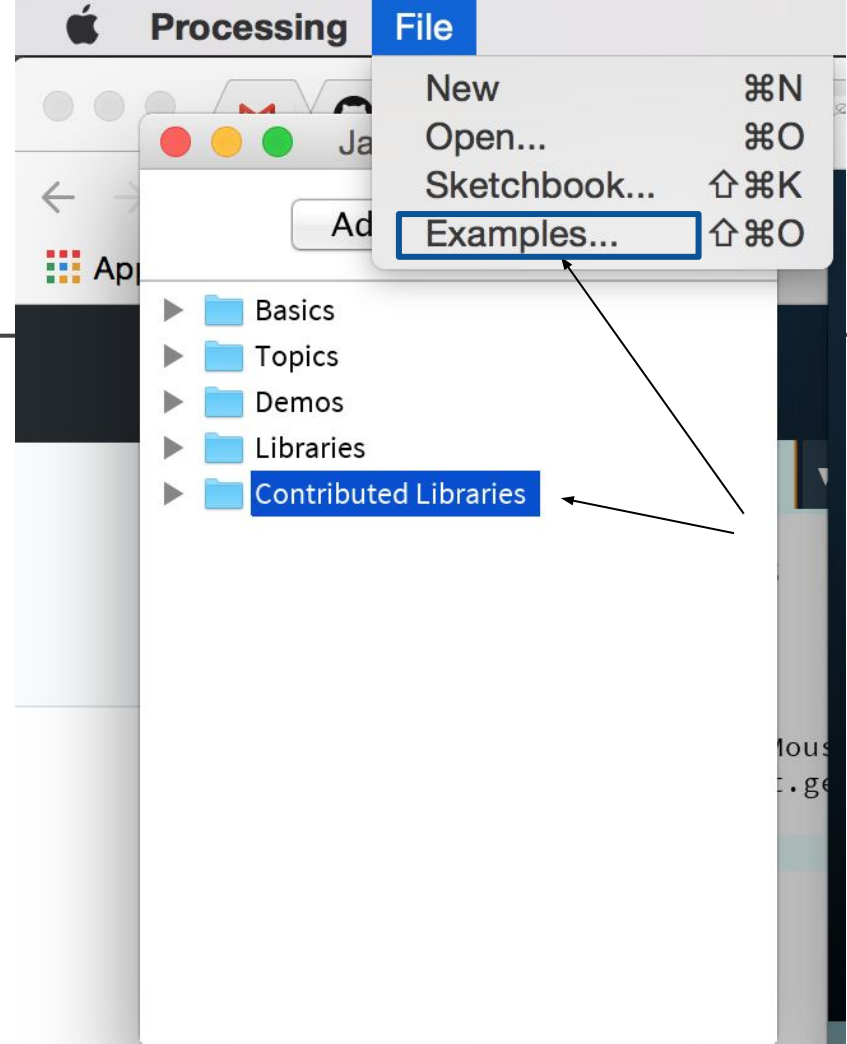
Librerías externas

Importar una librería

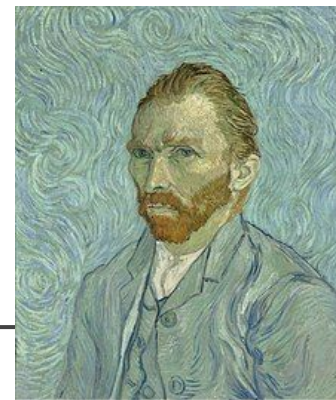
- Opción 1: Desde IDE: Sketch-> Import Library -> Add library
- Opción 2: Descargar la librería y ponerla en el directorio “libraries” que se genera donde está ubicada la carpeta “sketchbook”

Ejemplos de código

Los que ya vienen con el Processing y los de las librerías instaladas



Entrega 1



1. Dibujar un autorretrato usando
 - a. por lo menos 16 formas, dentro de las **16 formas** debería haber por lo menos un **line, arc, rect, ellipse, triangle** y una forma compuesta por **vértices**
 - b. por lo menos **6** colores **sólidos** y **6** con **transparencia**

Debería haber por lo menos un elemento (¿los ojos?) que cambia cuando cambia la posición del **mouse** y el otro que se mueva con las flechas del **teclado**.

2. Mandar **link** al mejor trabajo artístico “programado” que encontraron en la web.

Fecha de la entrega: 30.03.22, 23:59 hora Uruguay, por Eva.