



Introducción a Processing

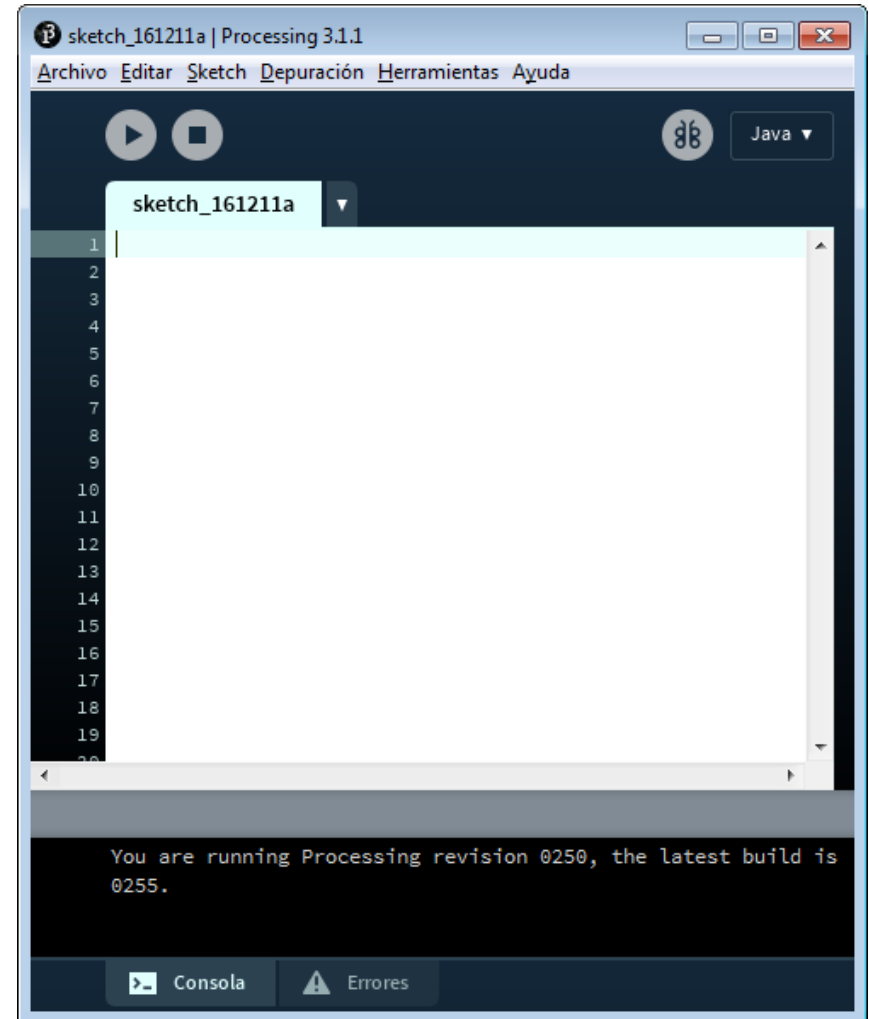
¿Qué es Processing?



- Es un lenguaje basada en Java
`{¿y eso que es?}`
- Es un entorno de programación simplificado
- Inició en 2001

Acerca del entorno

- Parece un bloc de notas
- Funciona en Windows, mac y Linux.
- Requiere de la maquina virtual de java
- Actualmente se encuentra en la versión 3.2.3



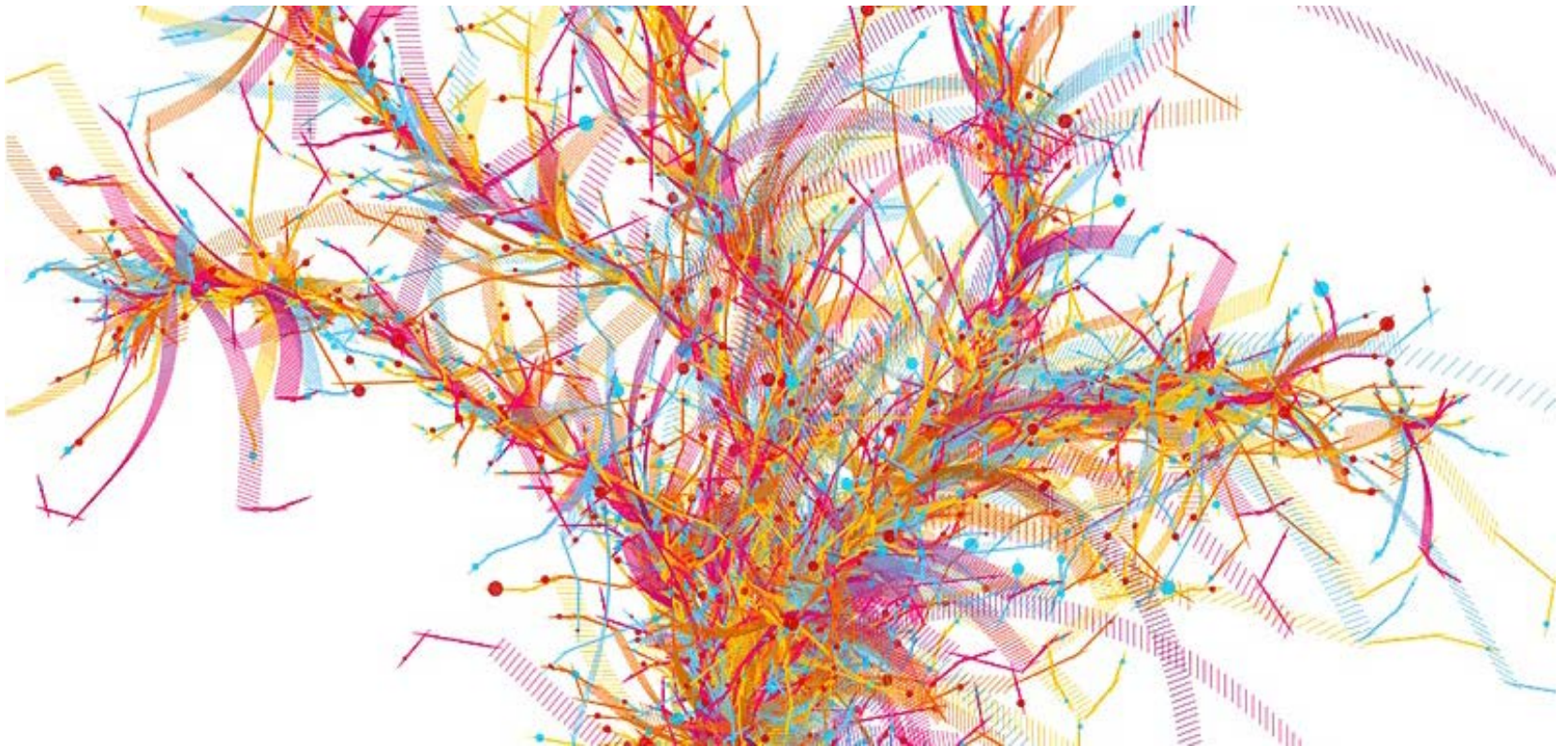
Como lenguaje

- Usa sintaxis de Java.
- Es de tipo de dato, estático y fuerte.
- Sigue el paradigma de programación Orientado a Objetos

```
/*  
  Programa de ejemplo introductorio a Processing  
  Pinta un texto de saludo y una línea acorde a la posición del mouse  
*/  
  
void setup(){  
  //se ajusta el tamaño de la ventana  
  size(500,500);  
}  
  
void draw (){  
  //pon el fondo de color gris  
  background (128);  
  //ajusta el tamaño del texto  
  textSize(25);  
  //define el texto y sus coordenadas  
  text("Hola al Mundo de processing",50,200);  
  //dibuja una línea entre el origen de coordenadas y la posición del mouse  
  line(0,0,mouseX,mouseY);  
}
```

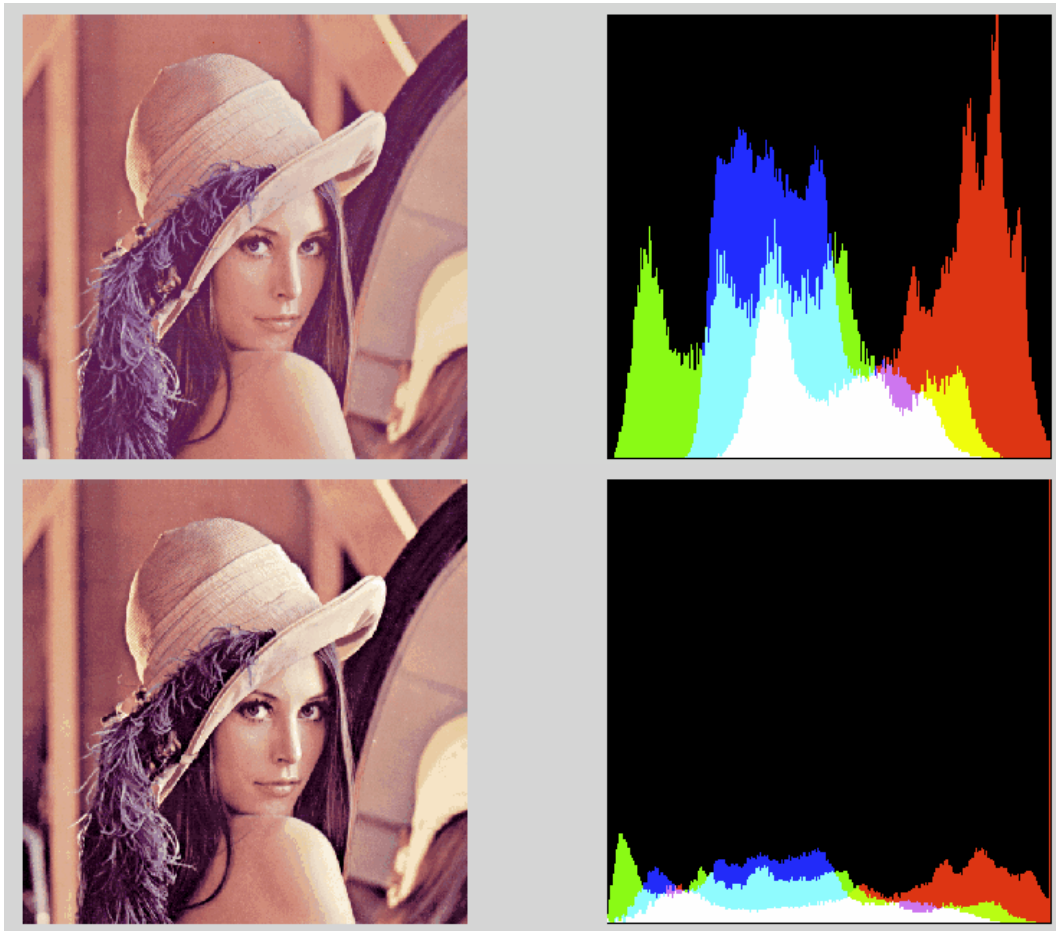
¿Para qué sirve processing?

- Visualización de datos



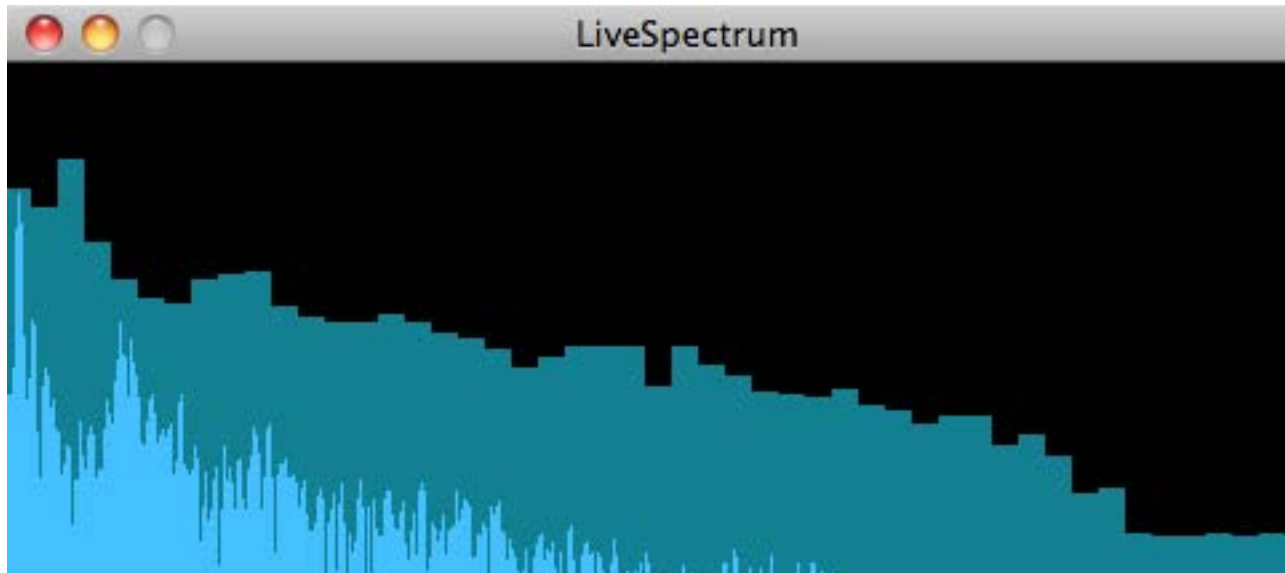
¿Para qué sirve processing?

- Procesamiento y análisis de imágenes



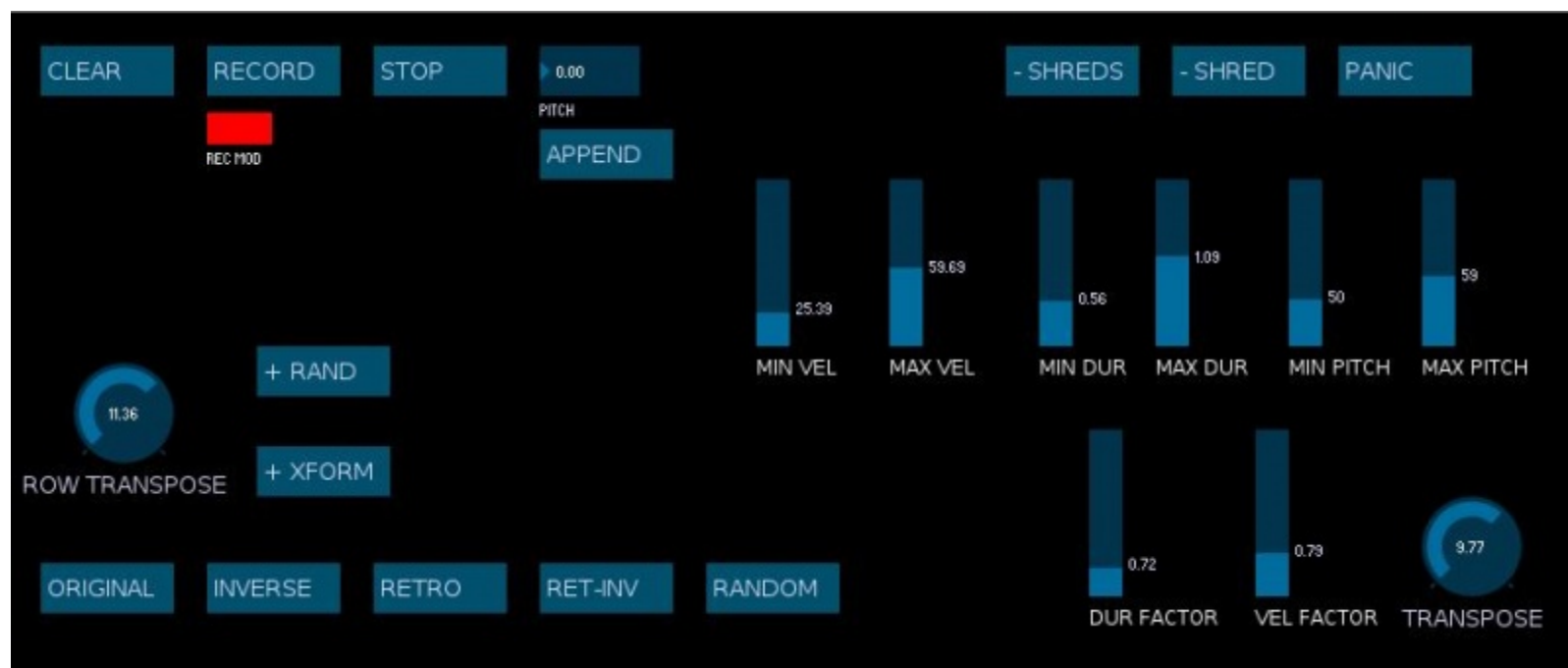
¿Para qué sirve processing?

- Procesamiento y análisis de audio



¿Para qué sirve processing?

- Construcción de interfaces de simulación y control

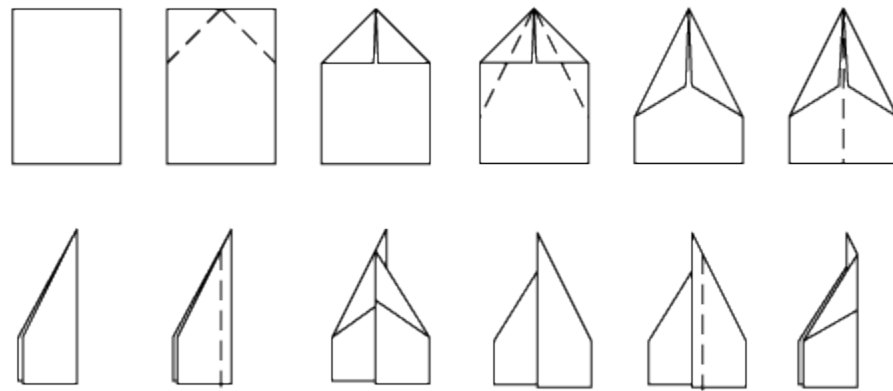


¿Por qué es potente como herramienta?

- Es un punto de partida de baja complejidad para iniciarse en “programación”
- Brinda acceso a las herramientas de java (gran comunidad y cantidad de recursos).
- Es una herramienta de prototipado funcional de baja complejidad.

¿Qué es programar?

Definir una serie de instrucciones para resolver una situación particular a través de la consulta y modificación de datos (algoritmo).



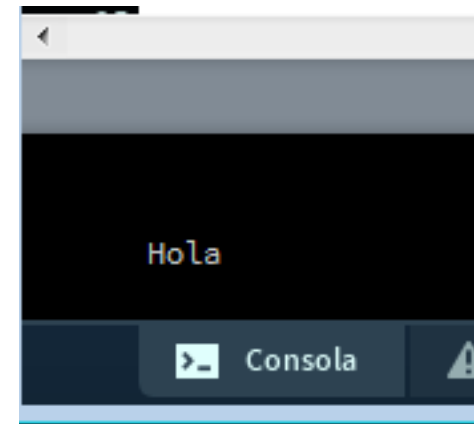
¿Qué necesitamos?

- Compresión del problema
 - Entradas
 - Proceso
 - Salidas
- Procesamiento
 - Instrucciones
- Almacenamiento
 - Información (Datos)

Sobre el procesamiento... 1

- Son instrucciones unitarias.
- El resultado no debe variar si la estructura no varía.
- Requiere una estructura (sintaxis) precisa determinada por el lenguaje.

```
1 println("Hola");
```



Sobre el procesamiento... 2

- Las instrucciones se ordenan en 3 grupos
- Estructuras **Secuenciales**
- Estructuras de **Selección**
- Estructuras de **Repetición**

Sobre el almacenamiento

- Existen 2 tipos de almacenamiento:
 - Temporal o de acceso rápido (RAM)
 - Persistente (Disco duro)

Depende de la categoría a la que pertenecen los datos...

- ¿Son letras? “hola”
- ¿Son números? 10
- ¿tienen decimales? ... 3,14
- ¿son de tipo booleano? ... verdadero o falso

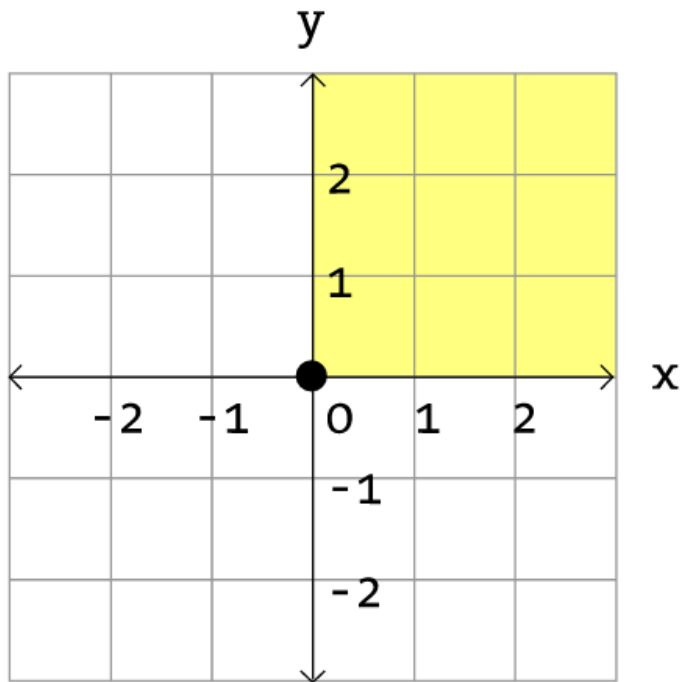
¿Cuáles son los tipos de dato en Java (processing)? (versión simplificada)

- Numéricos
 - Enteros – `int`
 - Flotantes (con decimales) – `float`
- Booleanos – `boolean`
- Alfanuméricos
 - Caracteres – `char`
 - Cadenas de texto (pueden ser letras y números) – `String`

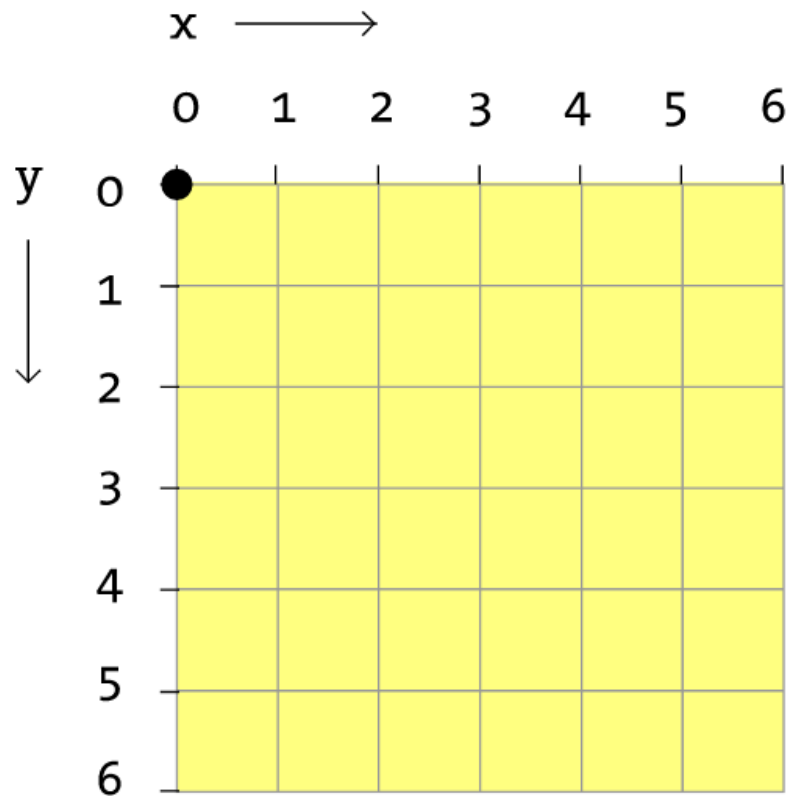
...El truco...

Llevar situaciones “cotidianas” a
escenarios que pueden ser tratados
como instrucciones e información
(combinados).

Sistema de coordenadas



En el colegio



En los computadores

Geometrías básicas en processing

2D Primitives

`arc()`

`ellipse()`

`line()`

`point()`

`quad()`

`rect()`

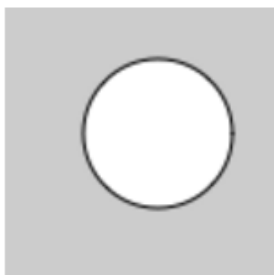
`triangle()`

- Instrucciones básicas de dibujo en processing.

www.processing.org/reference

Name ellipse()

Examples



`ellipse(56, 46, 55, 55);`

Description

Draws an ellipse (oval) to the screen. An ellipse with equal width and height is a circle. By default, the first two parameters set the location, and the third and fourth parameters set the shape's width and height. The origin may be changed with the `ellipseMode()` function.

Syntax

`ellipse(a, b, c, d)`

Parameters

a float: x-coordinate of the ellipse

b float: y-coordinate of the ellipse

c float: width of the ellipse by default

d float: height of the ellipse by default

Ejercicio 1 (Secuencia)



- En un lienzo de 500 por 500 pixeles, crear una cara feliz usando geometrías básicas.

Pista 1: la instrucción que determina el tamaño del lienzo es:

`size(ancho, alto);`

Pista 2: la instrucción que determina el color de la siguiente figura es:

`fill(r, g, b);`

Pista 3: la instrucción que determina el grosor de la línea es:

`strokeWeight(grosor);`

Variables en Java (declaración)

```
1 int posicionX = 250;  
2 int posicionY = 250;  
3 float radio = 58.5;  
4 boolean coloreado = false;
```

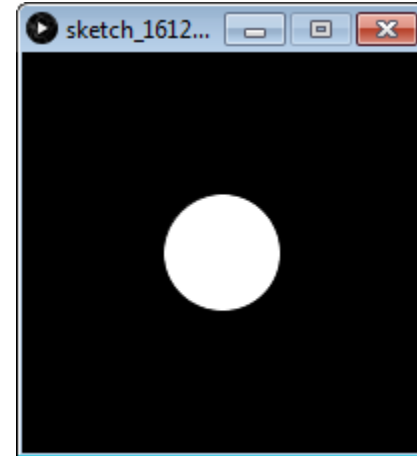
- Los datos se almacenan de acuerdo al tipo de dato al que pertenece.

Depende de la categoría a la que pertenecen los datos...

- ¿Son letras? “hola”
- ¿Son números? 10
- ¿tienen decimales? ... 3,14
- ¿son de tipo booleano? ... verdadero o falso

Variables en Java (utilización)

```
1 int posicionX = 250;  
2 int posicionY = 250;  
3 float radio = 58.5;  
4 boolean coloreado = false;
```

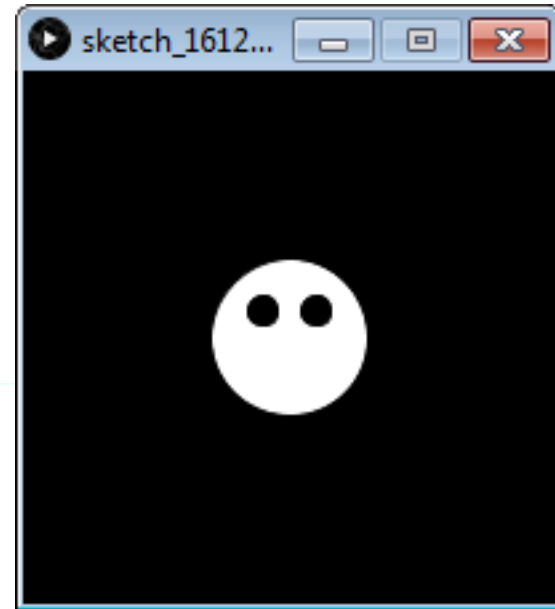


```
6 size(200,200);  
7 background(0,0,0);  
8 fill(255,255,255);  
9 ellipse(posicionX, posicionY, diametro, diametro);
```


Posiciones relativas

- Los ojos están ubicados de manera relativa.
- Y la escala de los mismos depende de la escala de la cabeza

```
int posicionX = 100;
int posicionY = 100;
int modulo = 10;
float diametro = 58.5;
size(200,200);
background(0,0,0);
fill(255,255,255);
ellipse(posicionX, posicionY, diametro, diametro);
fill(0,0,0);
ellipse(posicionX-modulo, posicionY-modulo, diametro/5, diametro/5);
ellipse(posicionX+modulo, posicionY-modulo, diametro/5, diametro/5);
```



Operadores en Java

Lógicos

- Y (&&)
- O (||)
- Negación (!)

Tablas de Verdad (&&)

true && true → true
true && false → false
false && true → false
false && false → false

Aritméticos

- Sumar (+)
- Restar (-)
- Multiplicar (*)
- Dividir (/)
- Módulo (%)

Tablas de Verdad (||)

true || true → true
true || false → true
false || true → true
false || false → false

Comparación

- Mayor (>)
- Menor (<)
- Mayor o igual (>=)
- Menor o igual (<=)
- Igual (==)
- Diferente (!=)

Recuerden...

- Las instrucciones se ordenan en 3 grupos
- ~~Estructuras~~ Secuenciales
- Estructuras de Selección
- Estructuras de Repetición

Utilización de imágenes en Processing

Clase PImage

Clase PImage

- Para la carga y visualización de imágenes en Processing, tipo de dato perteneciente a la clase Pimage
- Los formatos aceptados son: .jpg .png .bmp
- No hay restricciones de tamaño (ancho*alto) pero se recomienda exportar en 72 dpi (rgb pantallas) en el tamaño en el que se va a mostrar en el lienzo.
- <https://processing.org/reference/PImage.html>

¿Cómo se usa la Clase PImage?

```
PImage cara;  
  
void setup(){  
  size(400,400);  
  cara = loadImage("face.jpg");  
}  
void draw(){  
  background(255);  
  image(cara, 0, 0);  
}
```

