Programando en Processing

```
#include <Servo.h>

int a=3;
int b=7;


void setup( ) {

size(480, 120);

}



void draw( ) {

ellipse(50,50,100,100);
line(20,20,100,100);
suma(a,b);

}









int suma(int a, int b);
{
return(a+b);
}
```

Inicio de programa:

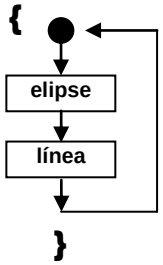
- Se indican las **librerías** necesarias para manejo de objetos y funciones → ***#include <nombre de librería.h>***
- Se crean los objetos de las clases requeridas
- Se definen e inicializan las variables, ejemplo: nombre = sergio;

setup ( )

Esta función contiene los parámetros de configuración, **se leen una única vez** al inicio del programa

- **size(300,300)**: indica el tamaño de la ventana en la que se ejecuta nuestro programa.

draw( ) {



}

Esta función repite los comandos del programa en un bucle infinito, con una frecuencia aprox de 50 veces por segundo.

- **ellipse**(50,50,100,100): dibuja una elipse de ancho y alto indicados en la posición X e Y indicadas.

Funciones definidas por el usuario

1.- REGLAS DE PROGRAMACIÓN

1. Existe una única función **setup** y una única función **draw**
2. Todas las sentencias, comandos y llamadas a función terminan en " **;** "
3. Este lenguaje diferencia entre mayúsculas y minúsculas, ¡ Atención a la sintaxis !

2.- FUNCIÓN SETUP

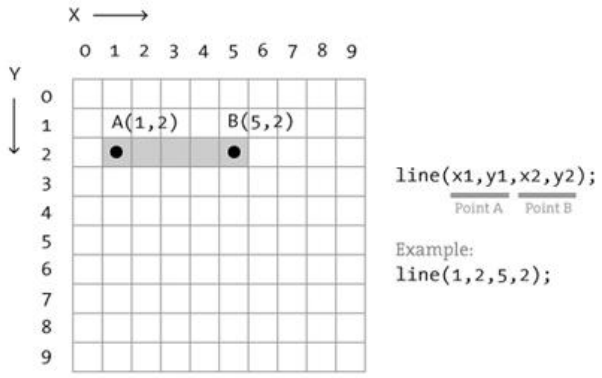
En ella se definen los parámetros iniciales de la ventana del programa:

- size(400,300) → define el ancho y alto de la ventana
- background(255, 204, 0) → define el color de refresco de la ventana

3.- LLAMADAS A FUNCIONES EN PROCESSING

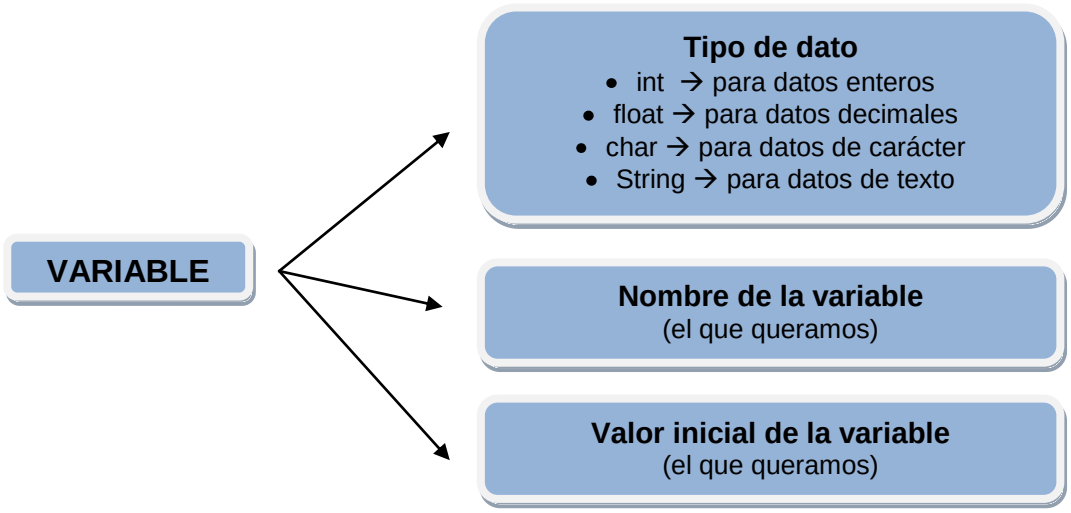
e l l i p s e (5 0 , 5 0 , 1 0 0 , 1 0 0) ;

- noStroke();
- fill(255,255,255);
- line(20,20,100,100);
- point(240, 60);
- rect(180, 60, 220, 40);
- ellipse(20,20,100,100);

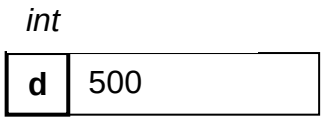


4.- DEFINICIÓN DE TIPOS DE VARIABLES

Una variable queda definida con 3 parámetros:



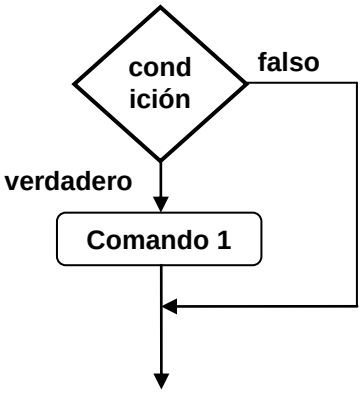
int a = 500; → *define la variable a de tipo entero y le da un valor inicial de 500*
char *c = 's';* → *define la variable c de tipo carácter y le da un* valor inicial de s
String d="hola"; → *define la variable d de tipo texto y le da un* valor inicial de "hola"



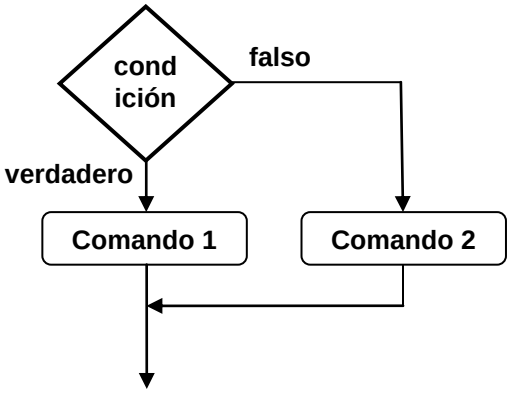
5.- PROPIEDADES DE OBJETOS

- mouseX --> *devuelve el valor de la coordenada X del puntero*
- mouseY --> *devuelve el valor de la coordenada Y del puntero*

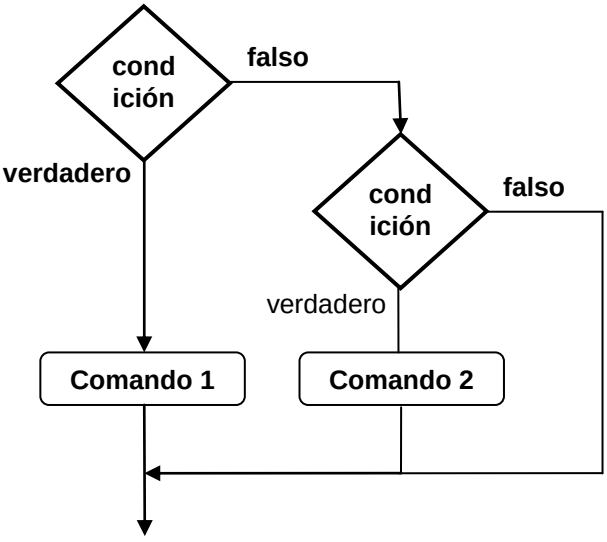
6.- SENTECIAS DE DECISIÓN




```
if (condición)  {
    comando 1
                }
```




```
if (condición)  {
 comando1
 } else {
  comando2
                }
```




```
if (condición 1)  {
      comando1
} else if (condición 2){
      comando2
                }
```

Dentro de las condiciones podemos usar los siguientes operadores lógicos:

- Operador AND: if(condición1 && condición2 && condición3)
- Operador OR: if(condición1 || condición2 || condición3)
- Operador IGUALDAD: if(a==10)
- Operador MAYOR/MENOR: if(a>10)

7.- EVENTOS DE CLASE

Se definen fuera de la función setup() o loop(). Se ejecuta lo que tienen dentro cuando ocurre el evento que representan

```
void keyPressed()  {

        }
```

```
void mousePressed()  {

        }
```

8.- LA CLASE PShape

Esta clase crea objetos de tipo imagen, de forma que luego podamos variar las propiedades de la imagen de forma fácil. Estas propiedades son la posición y la escala en X y en Y

```
PShape miimagen;

void setup() {
  miimagen = loadShape("bot.svg");
}

void draw() {
  shape( miimagen, 10, 10, 80, 80);
}
```

Definimos una variable de tipo PShape

Asignamos una imagen a dicha variable

Usamos la imagen especificando posición y escala:
shape(miimagen, X, Y, XSCALE, YSCALE)

Anexo I: ejercicios

Dificultad baja

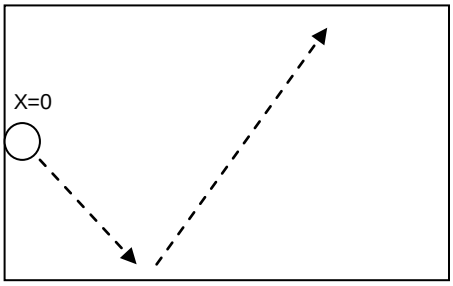
1.- Diseña un sketch de 400x400 píxeles con:
   a) un círculo rojo de radio 50 y centrado en las coordenadas (100,100),
   b) un rectángulo verde de 20x100 píxeles centrado en el (200,200) y
   c) una línea que va de las coordenadas (20,20) a (350,350)

2.- Diseña un sketch de 400x400 píxeles con:
   a) Una variable d, de tipo entero y de valor inicial 50;
   b) Un círculo rojo de radio d, otro de radio d/2 y otro de radio d/4 todos centrados en (100,100)

3.- En el mismo ejercicio anterior el radio vale una unidad menos que su fotograma anterior
4.- En el mismo ejercicio anterior, en cada fotograma, borramos el círculo creado en el fotograma anterior: **background(1,0,255);**
5.- Dibuja una línea horizontal de 15 píxeles de longitud, que se desplace de izquierda a derecha como si fuera una flecha. Sale de la posición y=100px y avanza a razón de 1px por fotograma.
6.- Dibuja un haz de 3 flechas que salgan de la posición y = 100, una de ellas sigue horizontalmente de izquierda a derecha como en el ejercicio anterior. La segunda flecha sale inclinada 3 píxeles hacia arriba. La tercera sale inclinada hacia abajo 3 píxeles. Estas dos últimas avanzan en sentido vertical a razón de 0.1 píxeles por fotogrma.
7.- Dibuja una explosión de bolitas de 15px de diámetro.
8.- Aprende a **resetear** las condiciones iniciales con if. Resetea la posición de las bolitas, de forma que cuando x sea mayor de 300px, vuelvan al centro del dibujo
9.- Resetea los colores de las bolitas para x>300ppx.

Dificultad media

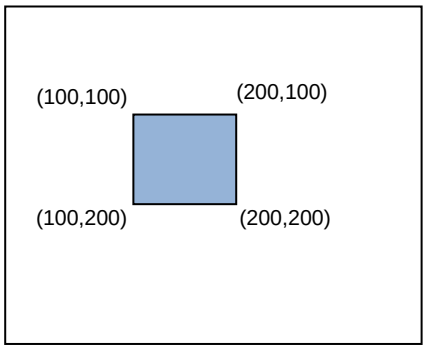
10.- En un sketch de 400x400, una bola de 15px de diámetro rebota cuando llega al fondo, es decir para y>400.
11. Como en el ejercicio anterior pero también rebota en la parte superior del sketch, es decir para y<0.
12.Consigue ahora, que avance en sentido de las X, para que haga una línea quebrada. Haz que salga de X=0;



13.- La pelota debe rebotar en todas las paredes. Cambia las condiciones iniciales para ver los diferentes trazados que hace. Es importante que veas cómo puedes cambiar la velocidad de la bola.
13bis.- En el ejercicio anterior siempre hace el mismo trazado, esto es debido a que la bola siempre bota a 45º(avanza lo mismo en X que en Y) y el sketch es cuadrado. Cambia las ecuaciones para que rebote a un ángulo diferente de 45º, por ejemplo, que avance 3 unidades en X por cada 2 unidades en Y.

Dificultad media-alta

14.- En el sketch debe aparecer una bola negra de radio 60 que se desplace con el puntero del ratón
15.- La bola cambia a color rojo cuando esté en la mitad inferior del sketch, es decir, para y>200
16.- La bola cambia a color rojo cuando está en una franja horizontal comprendida entre Y=150px e Y=250px. Nota: debes utilizar el operador AND. If(condición1 && condición2)
17.- La bola cambia de color para el rectángulo de la figura
18.- La bola cambia de color para tres o cuatro rectángulos que te inventes. El operador OR es ||



Ejercicio 16

Dificultada alta

19.- Un rectángulo de 80px de ancho y 10 de alto se desplaza en el inferior del sketch a Y=380px. Su movimiento en X coincide con el del ratón.
20.- Con el mismo ejercicio haz que caigan piedras de 5px de diámetro. Las piedras empiezan a Y=0 y salen de X en una posición aleatoria. Utiliza **random(400)** para definir la posición de la X. Cuando la piedra llega al final se debe volver a salir por arriba pero con otra X
21.- Ahora salen 5 bolas. Cada bola lleva un velocidad diferente dada por **random(2, 5)**, y vuelve a empezar cuando llega al fondo del sketch, simulando una lluvia de meteoritos.
22.- Cuando una de las 5 bolas toca el rectángulo este cambia a un color más frío.
23.- Cambia el rectángulo por la imagen de una nave espacial y las bolas por meteoritos. Usa **Inkscape.**