

# Textos y eventos

Félix J. Marcelo Wirnitzer  
<fjmarcelo@gmail.com>



## 1. Textos en Processing

Para escribir un texto en Processing usaremos la función

```
text(string Texto,float X,float Y[,float Z])
```

que dibuja en pantalla. Muestra la información especificada en el primer parámetro de sobre el área de trabajo en la posición indicada por los parámetros adicionales. A menos que se indique con la función `textFont()`, se usará una fuente con su tamaño predefinidos. El cambio de color del texto se puede hacer con `fill()`. De igual forma se mostrará el texto de acuerdo con la función `textAlign()`, permitiéndonos dibujar a la izquierda, a la derecha o al centro de las coordenadas.

\_\_\_\_\_ Ejemplo 1: Diferentes formatos \_\_\_\_\_

```
textSize(32);  
text("word", 10, 30);  
fill(0, 102, 153);  
text("word", 10, 60);  
fill(0, 102, 153, 51);  
text("word", 10, 90);
```



\_\_\_\_\_ Ejemplo 2: Alejamiento mediante el parámetro Z \_\_\_\_\_

```
size(100, 100, P3D);  
textSize(32);  
fill(0, 102, 153, 204);  
text("word", 12, 45, -30); // Specify a z-axis value  
text("word", 12, 60); // Default depth, no z-value specified
```



\_\_\_\_\_ Ejemplo 3: Encerrando un texto en una caja Z \_\_\_\_\_

```
String s = "The quick brown fox jumped over the lazy dog.";  
fill(50);  
  
// Text wraps within text box); // Default depth, no z-value specified  
//text(s, x0,y0,x1,y1);  
text(s, 10, 10, 70, 80);
```



## 2. Eventos de teclado y ratón

Hasta el momento hemos codificado programas solamente secuenciales, es decir, ejecutando línea a línea de principio a fin, si bien nos ha servido para los ejemplos es hora de dar un paso mas allá. En esta ocasión veremos lo que son los eventos del sistema y como ejecutarlos.

Un evento es una interrupción en el programa principal, ocurre cuando se tiene un cambio externo como por ejemplo cuando se presiona una tecla o un clic del mouse. Cuando uno de estos eventos sucede se llama a una función, dentro de esta función colocaremos el código a ejecutar.

Las funciones disponibles son las siguientes:

`mouseClicked()` Ocurre cuando se presiona y se libera un botón del ratón.

`mousePressed()` Cuando se presiona un botón del ratón.

`mouseDragged()` Al arrastrar el ratón con un botón pulsado mantenidamente.

`mouseReleased()` En el momento de soltar un botón del ratón que estaba presionado.

`keyPressed()` Cuando se pulsa una tecla.

`keyReleased()` En el momento de soltar una tecla presionada.

Para aprovechar al máximo estas funciones se debe hacer uso de las variables del sistema vistas anteriormente, de este modo sabremos exactamente qué tecla o botón generó la interrupción.

Las funciones anteriores son del tipo void y no necesitan parámetros para su funcionamiento, veamos un ejemplo para comprenderlo mejor:

```
_____ Ejemplo 4: Pintando círculos y limpiando _____  
void setup(){  
  size( 600, 400 );  
  frameRate(60);  
  smooth();  
  fill(255);  
}  
  
void draw(){  
  
}  
  
void mousePressed(){  
  if (mouseButton == LEFT) // boton izquierdo  
    ellipse(mouseX,mouseY,20,20); // dibujamos puntos alla donde pulsamos  
  else // boton derecho  
    background(128); // Limpiamos la pantalla con otro color  
}
```

## 3. Un ejemplo final

A continuación un ejemplo que combina las dos partes de este artículo...

\_\_\_\_\_ Ejemplo5: Persiguiendo el texto \_\_\_\_\_  
//Valores iniciales de la ventana

```
void setup() {  
  size( 600, 400 );  
  frameRate(60);  
  smooth();  
}  
  
void draw(){  
  
}  
  
void mousePressed(){  
  background(128);  
  text("Raton quieto",mouseX,mouseY);  
}  
  
void mouseDragged(){  
  background(128);  
  text("Raton escapando",mouseX,mouseY);  
}
```

\_\_\_\_\_