

# Taller de introducción a la programación con Processing

European Code Week 2015



# Processing

---

Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Fue iniciado por Ben Fry y Casey Reas a partir de reflexiones en el Aesthetics and Computation Group del MIT Media Lab dirigido por John Maeda.

Se distribuye bajo la licencia GNU GPL.

(<https://es.wikipedia.org/wiki/Processing>)



# Processing

---

La mejor ayuda:

<https://processing.org/reference/>

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

<https://processing.org/books/> (€€€)

# Ejemplillos

JaviCepa.com



Games: LudumDare 2015

Monstrosity, Get Rich! and Very Dungeon, my LudumDare games for 2015. Each one made under the allowed 48 hours (except Get Rich, under 72). Monstrosity My LudumDare entry for LudumDare Compo #33, themed "You are the monster" is a game where... you are a monster. I left creativity aside on ...

@JaviCepa



A common guy with an uncommon obsession about design.

Tweets

Seguir



Carlos Fernández  
@sirikon

11 oct

Aprender a programar y programar para aprender [malagacode.org](http://malagacode.org)  
#mlgcode

Retwitteado por Javi Cepa

Abrir



Javi Cepa  
@JaviCepa

11 oct

This... [pic.twitter.com/t6G9Apq1Lw](https://pic.twitter.com/t6G9Apq1Lw)



Retweet



Keiji Takahashi  
@\_kzt

11 oct

Twitter:

@javicepa

Web:

[javicepa.com](http://javicepa.com)

# Processing

---

- “Hello World”
- Introducción a la programación
- Processing gráfico
- Processing interacción
- Processing en red
- Mini Aplicación VUESTRA

# Processing “Hello World”

---

```
void setup () {  
    print("Hello World!");  
}
```

# Introducción a la programación

---

- Tipos de datos y variables
  - Boolean, byte, char, color, double, float, int, long
  - Arrays, clases (y objetos)
- Funciones
- Estructuras de control
  - Condicionales
  - Bucles
- Aplicaciones Processing

# Tipos de datos y variables

---

- Boolean (os)
  - true o false
- Byte
  - -128 a 127
- Int - enteros
  - -2,147,483,647 a 2,147,483,647
- Char
  - Caracteres ASCII
- String
  - <https://processing.org/reference/String.html>



# Tipos de datos y variables

---

- Color
  - `color c1 = color(204, 153, 0);`
  - `color c2 = #FFCC00;`
  - `r=red(c1); g=green(c1); b=blue(c1);`
  - `h=hue(); s=saturation() b=brightness();`
- Float (32 bits)
  - `-3.40282347E+38 a 3.40282347E+38`
- Double (64 bits)
- Long
  - `-9,223,372,036,854,775,808 a 9,223,372,036,854,775,808`

# Arrays, clases (objetos)

---

```
int grados = 360;
float[] cos_vals = new float[grados ];
for (int i=0; i < grados; i++) {
    cos_vals[i] = cos(TWO_PI/grados * i);
}
```

# Arrays, clases (objetos)

---

```
class HLine {  
    float ypos, speed; // propiedades  
    HLine (float y, float s) { // constructor  
        ypos = y;  
        speed = s;  
    }  
    void update() { // Funciones  
        ypos += speed;  
        if (ypos > height) {  
            ypos = 0;  
        }  
        line(0, ypos, width, ypos);  
    }  
}
```

# Arrays, clases (objetos)

---

```
// Declare and construct two objects (h1, h2) from the class HLine
HLine h1 = new HLine(20, 2.0);
HLine h2 = new HLine(50, 2.5);

void setup()
{
    size(200, 200);
    frameRate(30);
}

void draw() {
    background(204);
    h1.update();
    h2.update();
}
```

# Estructuras de control - Condicionales

---

- Operadores
  - != (no igual)
  - < (menor que)
  - <= (menor o igual)
  - == (igual)
  - > (mayor que)
  - >= (mayor o igual que)

# Estructuras de control - Condicionales

---

- IF

```
if (condición) {  
    // Instrucciones si verdadera  
} else {  
    // Instrucciones si falsa  
}
```

# Estructuras de control - Condicionales

---

- SWITCH

```
switch (valor) {  
    case 0:  
        // Si valor == 0  
        break;  
    case 'A':  
        // Si valor == 'A' ('A' = 65)  
        break;  
    default:  
        // No se cumple ningún caso.  
        break;  
}
```

# Estructuras de control - Bucles

---

- WHILE

```
while (condicion) {  
    // Instrucciones mientras la condición sea cierta  
}
```



# Estructuras de control - Bucles

---

- FOR

```
for (inicialización ; condición ; operación) {  
}
```

```
for (int i=0 ; i<10 ; i++) {  
    // Instrucciones  
}
```

```
for (int i=10 ; i>0 ; i--) {  
    // Instrucciones  
}
```

# Estructuras de control - Bucles

---

- FOR (Iterador)

```
for (tipo elemento : array) {  
    // Instrucciones por cada elemento  
}
```

# Processing “Hello World” II

---

```
void setup () {  
    frameRate(2);  
}  
  
int i=0;  
void draw() {  
    i++;  
    print("Hello World! ");  
    println(i);  
}
```

# Processing “Hello World” III

---

```
PFont f;

void setup () {
  frameRate(2);
  size(640,480);
  f = createFont("Arial",16,true);
  textFont(f,16);
  fill(255); // Color (Blanco)
}

void draw() {
  text("Hello World!",random(640),random(480));
}
```

# Primitivas gráficas

---

- `stroke(r,g,b [,a ])`;
- `strokeWeight()`;
- `line()`
- `arc()`
- `ellipse()`
- `point()`
- `quad()`
- `rect()`
- `triangle()`

# Interacción

---

- Teclado
  - keyPressed
  - key
  - keyCode
  - keyPressed()
  - keyReleased()
  - keyTyped()

# Interacción

---

- Ratón
  - `mousePressed`
  - `mouseButton`
  - `mouseClicked()` - Interrupción
  - `mouseDragged()` - Interrupción
  - `mouseMoved()` - Interrupción
  - `mousePressed()` - Interrupción
  - `mouseReleased()` - Interrupción
  - `mouseWheel()` - Interrupción
  - `mouseX`
  - `mouseY`
  - `pmouseX`
  - `pmouseY`

# Red

---

- TCP (librería .net)
  - Conexiones directas
  - HTTP
  - FTP
  - Es fiable, estado de la conexión
- UDP (no disponible de serie)
  - DHCP
  - Streaming de vídeo
  - DNS
  - No es fiable, no hay orden, no hay estado de conexión, poca carga en la red