

p5.js

Hoja resumen para principiantes

Estructura General

```
// Se ejecuta antes de setup
function preload() {
  img = loadImage('img.jpg');
}

// Se ejecuta una sola vez
function setup() {
  // Crea un lienzo de 400 x 400 píxeles
  createCanvas(400, 400);
}

// Se ejecuta 60 veces por segundo
function draw() {
  //noLoop(); Ejecuta draw una sola vez
}
```

Matemáticas

```
+ - / * // Operadores básicos
+= // x+=5 es lo mismo que x = x + 5
++ // x++ es lo mismo que x = x + 1
% // Modulo

random(min, max); // Nros aleatorios
noise(x, y, z); // Ruido Perlin (0, 1)
map(valor, in1, in2, out1, out2);
abs(-7); // 7 (Valor absoluto)
round(2.1); // 2
ceil(4.01); // 5
floor(5.99); // 5
sqrt(9); // 3 (Square root)
pow(5, 3); // 5 * 5 * 5 = 125 (Power)
// Funciones trigonométricas
sin(); cos(); tan();
```

Imprimir datos

```
print("Hola!"); // Imprime Hola!
console.log("Hola Mundo!");
```

Variables y funciones de p5.js

```
// Ancho y alto del lienzo
width / height

// Ancho y alto de la ventana del navegador
windowWidth / windowHeight

// Número de cuadros por segundo
frameCount

// La posición horizontal y vertical del ratón
mouseX / mouseY

// Cuando un tecla es presionada
function keyPressed() {
}

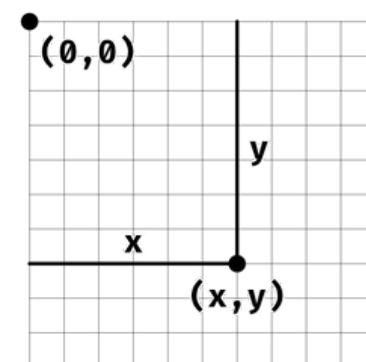
// Cuando el ratón se mueve
function mouseMoved() {
}

// Cuando presionas el ratón
function mousePressed() {
}

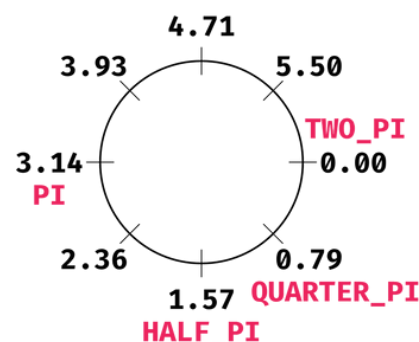
// Cuando presionas y sueltas el ratón
function mouseClicked() {
}

// Cuando cambia el tamaño de la ventana
function windowResized() {
}
```

Sistema de coordenadas



Ángulos en radianes



Color

```
background(color); // Color del fondo
fill(color); // Color de relleno
noFill(); // Sin color de relleno
stroke(color); // Color del borde
noStroke(); // Sin color de borde
strokeWeight(3); // Ancho del borde en px

// Variable que almacena un color
let c = color(23, 50, 30);
// Valores entre 0 y 255 (8 bits)
fill(127);
// Formato R-G-B (24 bits)
fill(100, 75, 50);
// Formato R-G-B + Transparencia
fill(255, 0, 100, 50)
// Nombre del color en inglés
fill('red');
// Formato hexadecimal
fill('#f1f1f1');
```

Operadores Lógicos

```
== // Es igual a
!= // No es igual a
> // Es mayor que
< // Es menor que
>= // Es mayor o igual que
<= // Es menor o igual que
&& // Operador AND "y"
|| // Operador OR "o"
! // Operador NOT (no)
```

Condicional if else

```
if (x < 5) {
  // ejecuta este código
} else if (x < 10) {
  // ejecuta este código
} else {
  // ejecuta este código
}
```

Primivitos 2D

```
point(x, y);
line(x1, y1, x2, y2);
square(x, y, ancho);
rect(x, y, ancho, alto);
circle(x, y, diametro);
ellipse(x, y, ancho, alto);
triangle(x1, y1, x2, y2, x3, y3);
arc(x, y, ancho, alto, comienzo, final);
ellipseMode(CORNER); // Defecto en CENTER
rectMode(CENTER); // Defecto en CORNER
beginShape();
  vertex(x1, y1);
  vertex(x2, y2);
  vertex(x3, y3)
endShape(CLOSE);
```

Transformaciones

```
push(); // Guardar las transformaciones
translate(x, y); // Trasladar
scale(xF, yF); // Escalar
rotate(angulo); // Rotar
pop(); // Restaurar las transformaciones
```

```
angleMode(DEGREES);
// Por defecto en RADIANS
```

Bucles (Loops)

```
for (let i = 0; i < 7; i++) {
  print(i); // Imprime 0 1 2 3 4 5 6
}
```

```
let frutas = ["Uva", "Pera", "Manzana"];
for (let i = 0; i < frutas.length; i++) {
  print(i); // Imprime 0 1 2
}
```

Compilado y traducido por Patricio Isaías
Basado en la [cheat sheet](#) de Ben Moren.
CC-BY-SA-NC-4.0