PROCESSING.PY: ‘0la mundo’ - funcións principais para debuxar

En Processing calquera programa terá cando menos 2 funcións, **setup() e draw()**.
setup() establece os valores iniciais e/ou constantes. É estática, e **só se executa unha única vez**.
draw() se repite constantemente, borrando o anterior, enviando novos valores, modificando os valores das variables...É un bucle infinito.
A orde print envia información á consola, non a debuxa.

```
sketch 250305a
def setup():
    print('valores iniciais')

def draw(): #bucle infinito
    print('60 fps')
    print(frameCount)
```

Variable globais e locais

As **variables globais** se definen ao principio do programa. As **locais** dentro da función. Unha variable local non pode utilizarse noutra función (por exemplo, se a definimos en setup() non podemos utilizala en draw()).

```
x = 280 #variables globais
y = -100
diametro = 380

def setup():
    size(480,120)
    fill(102)
#unha variable dentro de setup()
#non pode utilizarse en draw()

def draw(): #bucle infinito
    background(204)
    for i in range(1000):
        ellipse(x,y,diametro,diametro)
```

Se queremos usar e modificar o valor dunha variable dentro dunha función (normalmente, draw()) temos que usar a verba **global** para invocala. Se non, dará erro.

```
x = 0 #variables globais
y = 0

def setup():
    size(480,420)
    fill(102)

def draw(): #bucle infinito
    global x,y
    x = x+1
    y = y+1
    ellipse(x,y,10,10)
```

Movemento co rato e variables asociadas

mouseX e mouseY son as variables que almacenan a posición do rato.

Exercicio: Anticipa o programa antes de executalo:
Exercicio: Modifica o programa anterior borrando o # de background(204). Que acontece agora e por que?

```
sketch_03seguiraorato_mouseX
def setup():
    size(480,120)
    fill(0,102)
    noStroke()
def draw():
    #background(204) #borra o escritorio
    ellipse(mouseX,mouseY,9,9)
```

pmouseX e pmouseY almacenan o valor da posición no *frame* anterior. Podemos utilizalas para debuxar liñas:

```
sketch_04almacenarposicionrato_pmouseX
def setup():
    size(480,120)
    strokeWeight(4)
    stroke(0,102)

def draw():
    line(mouseX,mouseY,pmouseX,pmouseY)
```

Exercicios de debuxo interactivo co rato

Exercicio: sabendo que a función dist() devolve o valor da distancia entre 2 puntos, analizar o funcionamento do seguinte programa (que fai e como?)

sketch_05velocidadedorato_dist

```
def setup():
    size(480,120)
    stroke(0,102)

def draw():
    weight=dist(mouseX,mouseY,pmouseX,pmouseY) #función dist
    strokeWeight(weight) #grosor en función da velocidade
    line(mouseX,mouseY,pmouseX,pmouseY)

```

Movemento fluído

Algunhas técnicas matemáticas poden axudar a mellorar os programas. Nos dous seguintes o importante é analizar como varía o funcionamento (a "fluidez") en función do valor do parámetro *easing*.

```
x = 0.0
easing = 0.02
#parámetro que modifica a velocidade de movemento

def setup():
    size(220,330)

def draw():
    global x
    targetX= mouseX
    x += (targetX-x)*easing
    ellipse(x,40,12,12)
    print(x,targetX)
```

sketch_07debuxarcoratofluido_tecnicaeasing

```
x = 0.0
y = 0.0
px = 0.0
py = 0.0
easing = 0.05

def setup():
    size(480,120)
    stroke(0,102)

def draw():
    global x,y,px,py
    targetX = mouseX
    x += (targetX-x)*easing
    targetY = mouseY
    y += (targetY-y)*easing
    weight = dist(x,y,px,py)
    strokeWeight(weight)
    line(x,y,px,py)
    py=y
    px=x
```