

PROCESSING.PY: INTERACCIÓN BÁSICA E CONDICIONAL

En programación chamamos **estruturas de control** aos bucles e aos condicionais. Un condicional permite saber se algo é certo (estamos pulsando o rato?) usando o comando `if`.

```
if variable lóxica, comparación... == True:  
    executar isto  
else: #se non é True ten que ser False, non escribimos a condición  
    executar isto
```

Ollo: non é obrigatorio escribir `else`, tan só se queremos executar comandos alternativos

Como saber se o Rato está presionado

mousePressed: variable booleana que almacena `True` se o rato está presionado.

Alternativa ao código anterior con `if...else` (non é necesario escribir o valor `True`):

```
if mousePressed:  
    stroke(0)  
else:  
    stroke(255)
```

Con este código poderíamos “saber” se o rato non está presionado, ou o contrario. No programa modifica a cor.

sketch_09condicional_botonrato_

```
1 def setup():  
2     size(480,120)  
3     strokeWeight(30)  
4  
5 def draw():  
6     background(204)  
7     stroke(102)  
8     line(40,0,70,height)  
9     if mousePressed == True:  
10        stroke(0)  
11        line(0,70,width,50)
```

Ollo! no if temos que escribir `==`; un `=` a secas é unha asignación, non permite comparar. Se queremos escribir “distinto de” escribimos `!=` (por exemplo: `if 5 != 3` é unha condición verdadeira, valor `True`).

Podemos inserir condicionais dentro de condicionais, tendo que cumprirse dúas condicións ao tempo para executar os comandos.

Diferenciar botóns do rato

mouseButton: variable que pode valer `LEFT` ou `RIGHT` (ou outros).

Neste caso se pulsamos o rato diferencia de se pulsamos o botón esquerdo (cor branca) ou outro (cor negra). Se non pulsamos nada a cor é gris.

sketch_10condicional_variosbotonsdorato

```
1 def setup():  
2     size(480,120)  
3     strokeWeight(30)  
4  
5 def draw():  
6     background(204)  
7     stroke(102)  
8     line(40,0,70,height)  
9     if mousePressed:  
10        if mouseButton == LEFT:  
11           stroke(255)  
12        else:  
13           stroke(0)  
14        line(0,70,width,50)
```

Fai os programas da aula virtual que permiten debuxar distinto dependendo de onde está o rato.

Localización: diferenciar onde está o rato na pantalla

Usaremos as variables mouseX e mouseY. E un par de parámetros que modifican o debuxado en pantalla segundo movamos o rato.

Analiza o seguinte código (non é tan sinxelo):

Exercicio: fai que a liña do medio sexa un rectángulo que persegue ao rato.

```
condicional movemento rato
1 x = 0.0
2
3 def setup():
4     global x
5     size(240,120)
6     x = width/2
7
8 def draw():
9     global x
10    print(x,mouseX)
11    background(204)
12    offset = 0 #variable local, para debuxar a frecha
13    if mouseX > x: #detecta que o rato está á dereita
14        x += 0.5
15        offset = -10
16    if mouseX < x: #detecta que o rato está á esquerda
17        x -= 0.5
18        offset = 10
19    line(x,0,x,height) #liña do medio, vai perseguir ao rato
20    line(mouseX,mouseY,mouseX+offset,mouseY-10)
21    line(mouseX,mouseY,mouseX+offset,mouseY+10)
22    line(mouseX, mouseY,mouseX+(offset*3),mouseY)
23
24
25
```

Exercicio: modifica o seguinte código para que o círculo volva a ter o mesmo tamaño ao apartar o rato.

Lembra: a función dist() devolve a distancia entre dous puntos (x1,y1,x2,y2).

```
condicional movemento rato
1 x = 120
2 y = 60
3 radio = 12
4
5
6 def setup():
7     size(480,120)
8     ellipseMode(RADIUS)
9
10 def draw():
11    global radio
12    background(204)
13    d = dist(mouseX,mouseY,x,y)
14    if d < radio: #detecta que o rato está dentro do círculo
15        radio += 1 #aumenta o tamaño
16        fill(0) # cor negra cando está dentro
17    else:
18        fill(255) #se o rato está fora, a cor é branca
19    ellipse(x,y,radio,radio)
20
```