

Gráficos: el módulo turtle (3). Eventos

En esta lección se trata la creación de gráficos usando el módulo turtle. Se recomienda leer previamente la lección [turtle \(1\)](#).

Repetición: mainloop()

Para crear programas de tortuga interactivos necesitamos que los programas no terminen, sino que permanezcan a la espera y reaccionen a los eventos (de ratón o de teclado) generados por el usuario.

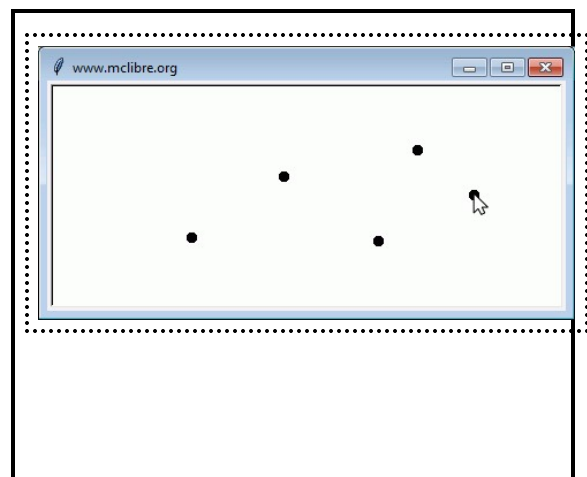
La función **mainloop()**, que se escribe al final del programa, hace que el programa no termine, sino que permanezca a la espera de los eventos definidos en el programa.

Evento hacer clic: onclick()

La función **onclick(función, botón=1, añade=False)** relaciona un clic de ratón con una función, de manera que cuando se haga clic se ejecutará la función. Los tres argumentos de la función son:

- **función**: el nombre de la función. Esta función recibirá siempre dos argumentos, las coordenadas x e y del punto en el que se ha hecho clic.
- **botón**: el botón pulsado (1 = botón izquierdo, 2= botón central, 3 = botón derecho). Si se omite, el valor predeterminado es 1.
- **añade**: si el valor es **True** la relación entre el clic y la función se añade a las anteriores, si el valor es **False** la relación sustituye a las anteriores. Si se omite, el valor predeterminado es **False**.
- El siguiente programa ejecuta la función punto() cada vez que se hace clic en la ventana de tortuga. La función simplemente dibuja un punto en el lugar en el que se ha hecho clic.

```
from turtle import *  
  
setup(450, 200, 0, 0)  
screensize(300, 150)  
title("www.mclibre.org")  
hideturtle()  
penup()  
  
def punto(x, y):  
    goto(x, y)  
    dot(10, "black")  
  
onclick(punto)  
mainloop()
```



- El siguiente programa ejecuta además la función `cuadro()` cada vez que se hace clic con el botón derecho. La función `cuadro()` simplemente dibuja un cuadro en el lugar en el que se ha hecho clic derecho.

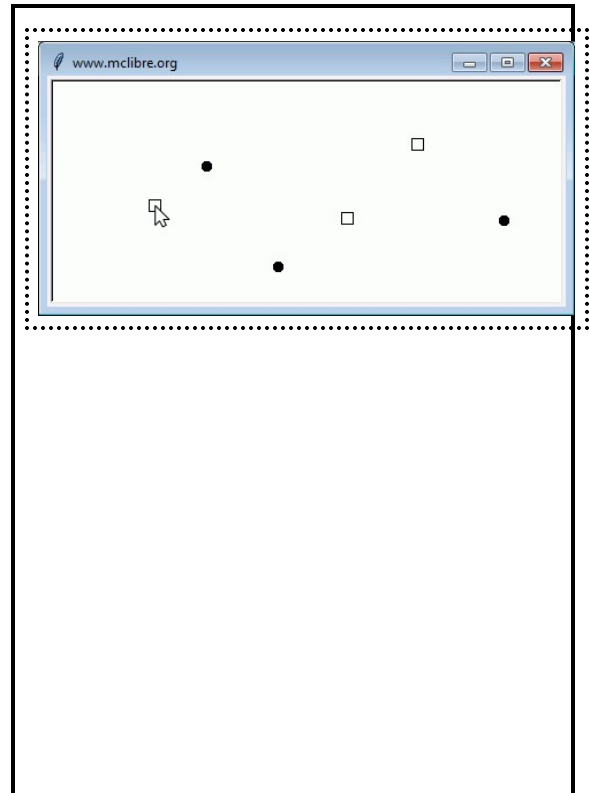
```
from turtle import *

setup(450, 200, 0, 0)
screensize(300, 150)
title("www.mclibre.org")
hideturtle()
penup()

def punto(x, y):
    goto(x, y)
    dot(10, "black")

def cuadro(x, y):
    goto(x - 5, y - 5)
    pendown()
    goto(x + 5, y - 5)
    goto(x + 5, y + 5)
    goto(x - 5, y + 5)
    goto(x - 5, y - 5)
    penup()

onscreenclick(punto)
onscreenclick(cuadro, 3)
mainloop()
```



- El siguiente programa ejecuta además las funciones `punto()` y `cuadro()` cada vez que se hace clic con el botón central del ratón. Para que se asocien las dos funciones al clic central, la segunda llamada incluye el tercer argumento `True`.

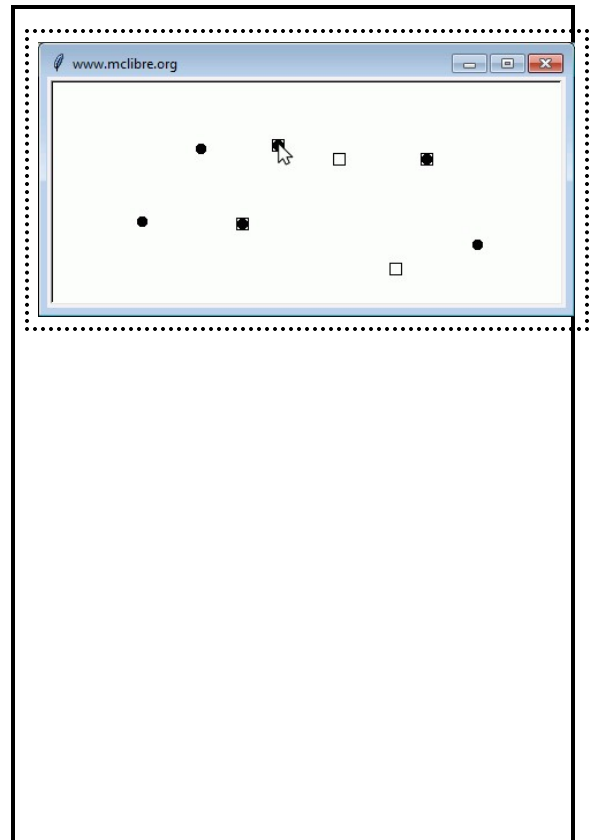
```
from turtle import *

setup(450, 200, 0, 0)
screensize(300, 150)
title("www.mclibre.org")
hideturtle()
penup()

def punto(x, y):
    goto(x, y)
    dot(10, "black")

def cuadro(x, y):
    goto(x - 5, y - 5)
    pendown()
    goto(x + 5, y - 5)
    goto(x + 5, y + 5)
    goto(x - 5, y + 5)
    goto(x - 5, y - 5)
    penup()

onscreenclick(punto)
onscreenclick(cuadro, 3)
onscreenclick(punto, 2)
onscreenclick(cuadro, 2, True)
mainloop()
```



Temporizador: ontimer()

La función **ontimer(función, tiempo)** pone en marcha un temporizador que ejecuta una función al cabo de un tiempo. Los dos argumentos de la función son:

- **función**: la función que se ejecuta cuando el temporizador termina de contar el tiempo.
- **tiempo**: el tiempo en milisegundos que cuenta el temporizador

El temporizador sólo cuenta una vez el tiempo. Si se quiere llamar a la función de forma repetida, se debe incluir un nuevo **ontimer()** en la función a la que llama **ontimer()**.

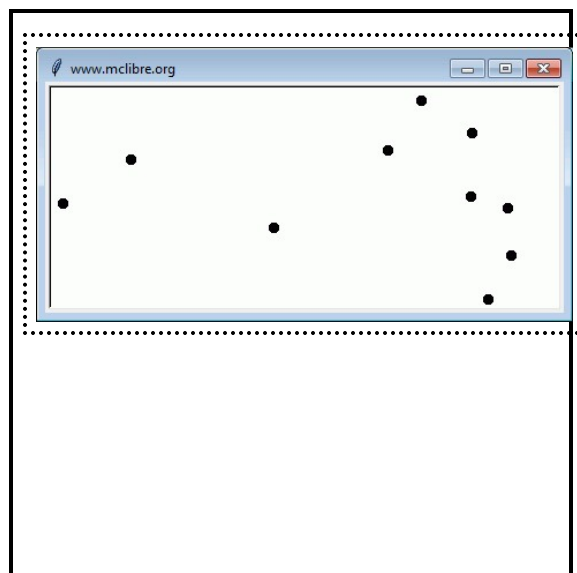
- El siguiente programa pone en marcha un temporizador que al cabo de un segundo llama a la función punto(). La función punto() dibuja un punto al azar en la ventana y vuelve a poner en marcha un temporizador, por lo que el programa no acaba nunca. Este programa no necesita llamar a la función mainloop(), puesto que el programa no responde a eventos de ratón o teclado.

```
from turtle import *
from random import randint

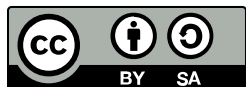
setup(450, 200, 0, 0)
screensize(300, 150)
title("www.mclibre.org")
hideturtle()
penup()

def punto():
    global randint
    goto(randint(-225, 225), randint(-100, 100))
    dot(10, "black")
    ontimer(punto, 1000)

ontimer(punto, 1000)
```



Última modificación de esta página: 22 de marzo de 2015



Esta página forma parte del curso [Introducción a la programación con Python](#) por [Bartolomé Sintes Marco](#) que se distribuye bajo una [Licencia Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional \(CC BY-SA 4.0\)](#).