Desconocido

### Crear la clase Pelota

Ahora crearemos la clase de la pelota. Empezaremos con el código que necesitamos para que la pelota se dibuje a sí misma en el lienzo. Tenemos que hacer lo siguiente

1. Crear un class llamado Ball que tome como parámetros el lienzo y el color de la pelota que vamos a dibujar.
2. Guarda el lienzo como una variable objeto porque dibujaremos nuestra bola en él.
3. Dibuja un círculo relleno en el lienzo utilizando el valor del parámetro color como color de relleno.
4. Guarda el identificador que devuelve tkinter cuando dibuja el círculo (óvalo) porque vamos a utilizarlo para mover la bola por la pantalla.
5. Mueve el óvalo al centro del lienzo.

Este código debe añadirse justo después de las tres primeras líneas del archivo (después de import time ):

from tkinter import \*  
import random  
import time  
  
class Ball:  
 def \_\_init\_\_(self, canvas, color):  
 self.canvas = canvas  
 self.id = canvas.create\_oval(10, 10, 25, 25, fill=color)  
 self.canvas.move(self.id, 245, 100)  
  
 def draw(self):  
 pass

En primer lugar, nombramos a nuestra clase Ball . Luego creamos una función de inicialización (como se describe en "Inicializar un objeto" en [la página 109](ch08.xhtml#ch08lev1sec11) ) que toma los parámetros canvas y color . Establecemos la variable de objeto canvas con el valor del parámetro del mismo nombre. A continuación, llamamos a la función create\_oval con cinco parámetros: coordenadas *x e* *y* de la esquina superior izquierda (10 y 10), *coordenadas* *x e* *y* de la esquina inferior derecha (25 y 25) y el color de relleno del óvalo.

La función create\_oval devuelve un identificador para la forma que ha dibujado, que almacenamos en la variable de objeto id . Movemos el óvalo al centro del lienzo (coordenadas 245, 100). El lienzo sabe qué mover porque utilizamos el identificador de forma almacenado ( id ) para identificarlo.

En las dos últimas líneas de la clase Ball , creamos la función draw con def draw(self) , y el cuerpo de la función es simplemente la palabra clave pass . De momento, no hace nada, pero en breve añadiremos más cosas a esta función.



Ahora que hemos creado nuestra clase Ball , necesitamos crear un objeto de esta clase (recuerda: una clase describe lo que puede hacer, pero el objeto es lo que realmente lo hace). Añade este código al final del programa para crear un objeto bola roja:

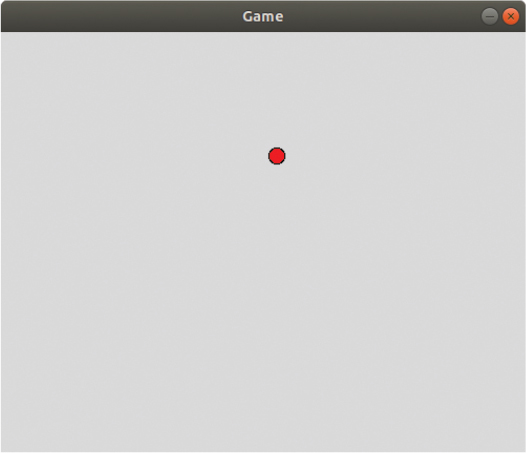
ball = Ball(canvas, 'red')

Puedes ejecutar este programa utilizando **Ejecutar** **▸ Ejecutar Módulo .** Si haces esto fuera de IDLE, el lienzo aparecerá durante una fracción de segundo y luego desaparecerá. Para evitar que la ventana se cierre inmediatamente, tenemos que añadir un bucle de animación, que se llama *bucle* principal de nuestro juego. (IDLE ya tiene un bucle principal, por eso la ventana no desaparece cuando lo ejecutas allí).

Un bucle principal es la parte central de un programa que generalmente controla la mayor parte de lo que hace. Nuestro bucle principal, de momento, sólo le dice a tkinter que redibuje la pantalla. El bucle, también llamado *bucle infinito* , sigue ejecutándose para siempre (o al menos hasta que cerremos la ventana), diciéndole constantemente a tkinter que redibuje la pantalla y durmiendo después durante una centésima de segundo mediante time.sleep . Añadiremos este código al final de nuestro programa:

ball = Ball(canvas, 'red')  
  
while True:  
 tk.update\_idletasks()  
 tk.update()  
 time.sleep(0.01)

Ahora, si ejecutas el código, la bola debería aparecer en el centro del lienzo, como se muestra en la Figura 11-2 [.](ch11.xhtml#ch11fig02)



Figura*11-2:* Pelota en el centro del lienzo

[anterior](ch11_3.html)[Subtema 4 de 7: (Ver todo)](ch11.html)[siguiente](ch11_5.html)