Desconocido

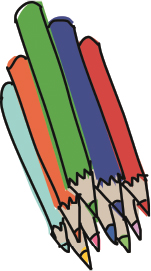
### Configurar el color

Vamos a añadir interés a nuestros gráficos con el color. Cambiaremos la función random\_rectangle para pasar un color para el rectángulo como parámetro adicional ( fill\_color ). Introduce este código en una nueva ventana, y cuando guardes, llama al archivo *colorrect.py :*

from tkinter import \*  
import random  
tk = Tk()  
canvas = Canvas(tk, width=400, height=400)  
canvas.pack()  
  
def random\_rectangle(width, height, fill\_color):  
 x1 = random.randrange(width)  
 y1 = random.randrange(height)  
 x2 = random.randrange(x1 + random.randrange(width))  
 y2 = random.randrange(y1 + random.randrange(height))  
 canvas.create\_rectangle(x1, y1, x2, y2, fill=fill\_color)

La función create\_rectangle toma ahora un parámetro, fill\_color , que especifica el color a utilizar al dibujar el rectángulo.

Podemos pasar colores con nombre a la función de esta forma para crear un montón de rectángulos de colores únicos. Si pruebas este ejemplo, considera la posibilidad de copiar y pegar, después de introducir la primera línea, para ahorrarte el tecleo. Para ello, selecciona el texto a copiar, pulsa CTRL-C para copiarlo, haz clic en una línea en blanco y pulsa CTRL-V para pegar. Añade este código a *colorrect.py* , justo debajo de la función



random\_rectangle(400, 400, 'green')  
random\_rectangle(400, 400, 'red')  
random\_rectangle(400, 400, 'blue')  
random\_rectangle(400, 400, 'orange')  
random\_rectangle(400, 400, 'yellow')  
random\_rectangle(400, 400, 'pink')  
random\_rectangle(400, 400, 'purple')  
random\_rectangle(400, 400, 'violet')  
random\_rectangle(400, 400, 'magenta')  
random\_rectangle(400, 400, 'cyan')

Muchos de estos colores con nombre mostrarán el color que esperas ver, pero otros pueden producir un mensaje de error (dependiendo de si utilizas Windows, macOS o Linux). Pero, ¿qué ocurre con un color personalizado que no es exactamente igual a un color con nombre? Recuerda que en el [Capítulo 9](ch09.xhtml#ch09) establecimos el color de la pluma de la tortuga utilizando porcentajes de los colores rojo, verde y azul. Establecer la cantidad de cada color primario (rojo, verde y azul) que se utilizará en una combinación de colores con tkinter es un poco más complicado, pero lo resolveremos.

Cuando trabajamos con el módulo turtle , creamos el dorado utilizando un 90% de rojo, un 75% de verde y nada de azul. En tkinter , podemos crear el mismo color dorado utilizando esta línea:

random\_rectangle(400, 400, '#e5d800')

La almohadilla ( # ) antes del valor ffd800 indica a Python que estamos proporcionando un número *hexadecimal* . El hexadecimal es una forma de representar números habitual en programación informática. Utiliza una base de 16 (del 0 al 9, y luego de la A a la F) en lugar del decimal, que tiene una base de 10 (del 0 al 9). Si no has aprendido nada sobre bases en matemáticas, debes saber que puedes convertir un número decimal normal a hexadecimal utilizando un *marcador de formato* en una cadena: {:x} (consulta "Incrustar valores en cadenas" en [la página 29](ch03.xhtml#ch03lev1sec4) ). Por ejemplo, para convertir el número decimal 15 a hexadecimal, podrías hacer lo siguiente:

>>> print(f'{15:x}')  
f

Esto es una cadena f con un modificador de formato especial (que es :x ) que indica a Python que convierta el número a hexadecimal.

Para asegurarnos de que nuestro número tiene al menos dos dígitos, podemos cambiar ligeramente el marcador de posición de formato, a esto:

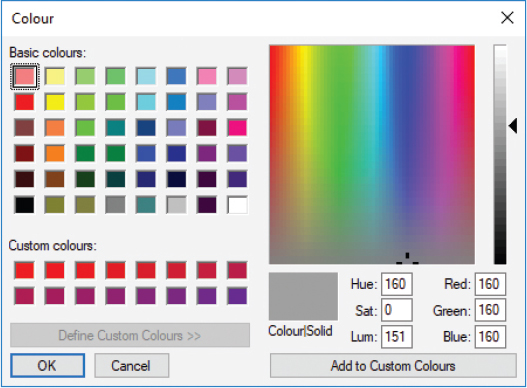
>>> print(f'{15:02x}')  
0f

Esta vez tenemos un modificador de formato ligeramente diferente ( 02x ) que dice que queremos la conversión hexadecimal, pero con dos dígitos (utilizando 0 para cualquier dígito que falte).

El módulo tkinter proporciona una forma sencilla de obtener valores de color hexadecimales. Prueba a ejecutar el siguiente código en IDLE:

from tkinter import \*  
from tkinter import colorchooser  
tk = Tk()  
tk.update()  
print(colorchooser.askcolor())

Este código muestra un selector de colores, que se muestra en la [Figura 10-8 .](ch10.xhtml#ch10fig08) Ten en cuenta que tienes que importar explícitamente el módulo colorchooser porque no está disponible automáticamente en Python cuando utilizas from tkinter import \* .



*Figura 10-8: El selector de color tkinter (que puede tener un aspecto diferente en tu SO)*

Cuando selecciones un color y pulses **OK** , se mostrará una tupla. Esta tupla contiene otra tupla con tres números y una cadena:

>>> print(colorchooser.askcolor())  
((157, 163, 164), '#9da3a4')

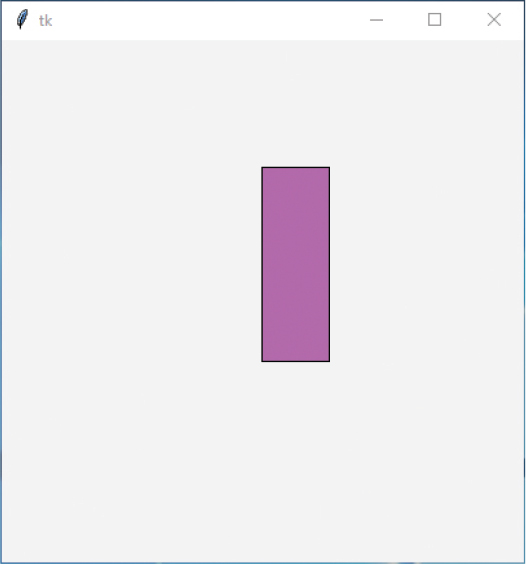
Los tres números representan las cantidades de rojo, verde y azul. En tkinter , la cantidad de cada color primario a utilizar en una combinación de colores se representa mediante un número entre 0 y 255 (que es diferente de utilizar un porcentaje para cada color primario con el módulo turtle ). La cadena de la tupla contiene la versión hexadecimal de esos tres números.

Puedes copiar y pegar el valor de la cadena para utilizarlo, o almacenar la tupla como una variable y utilizar la posición del índice del valor hexadecimal.

Utilicemos la función random\_rectangle para ver cómo funciona, sustituyendo todas las llamadas a random\_rectangle en la parte inferior de *colorrect.* py por el siguiente código:

from tkinter import colorchooser  
c = colorchooser.askcolor()  
random\_rectangle(400, 400, c[1])

Puedes ver el resultado en la [Figura 10-9 .](ch10.xhtml#ch10fig09)



Figura*10-9: Dibujar un rectángulo morado*

[anterior](ch10_7.html)[Subtema 8 de 17: (Ver todo)](ch10.html)[siguiente](ch10_9.html)