

¿Cómo mostrar un mensaje cuando se posiciona el cursor sobre un widget?

Formulada hace 2 años y 6 meses

Modificada hace 2 años y 6 meses

Vista 1k veces

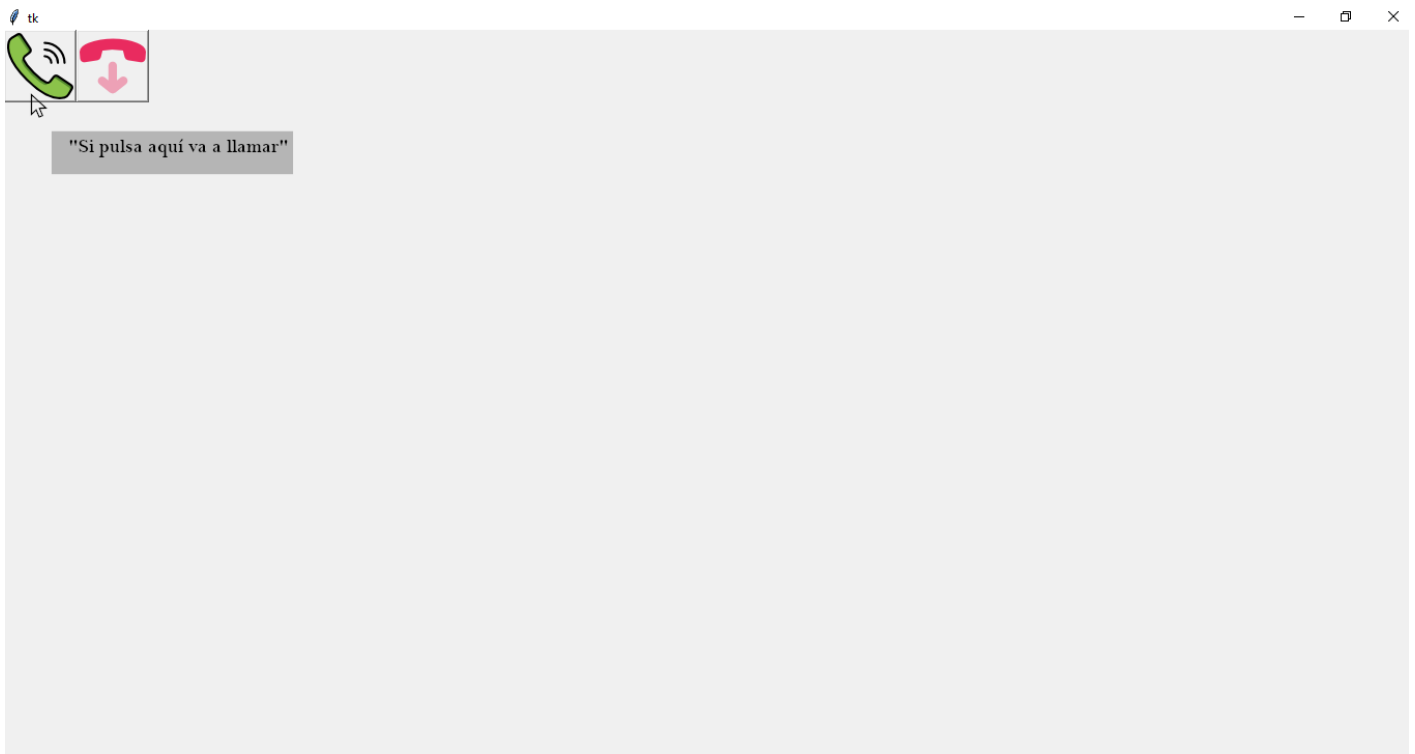
¡Hola Comunidad!

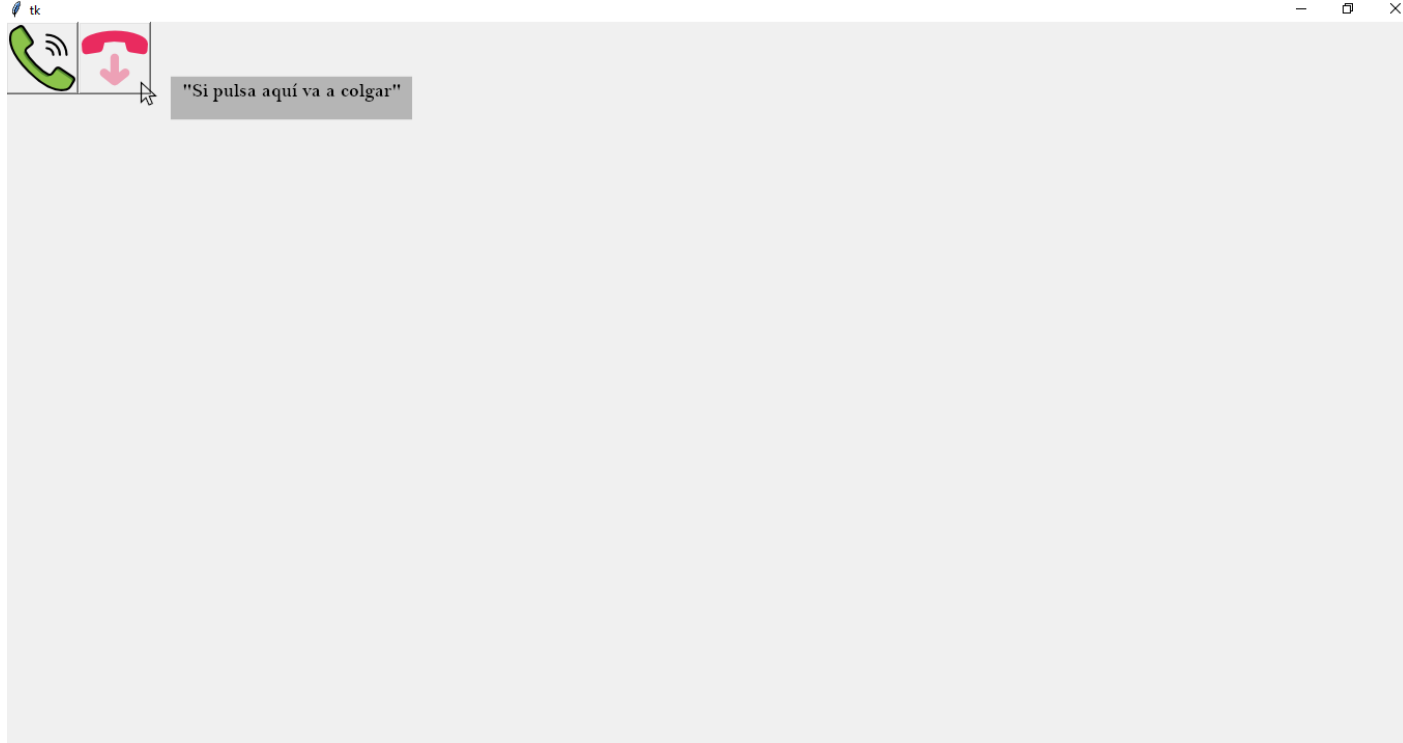
1

En la mayoría de aplicaciones, se muestra un determinado texto cuando el usuario posiciona el cursor sobre determinado elemento de la interfaz. Para esta publicación, se tiene una GUI con `tkinter` en la que se tienen dos (2) botones con imágenes en su interior (obviemos lo que puede llegar a suceder cuando se pulsa cualquiera de ellos). Lo que deseo mostrar es, que antes que ser pulsados, se le muestre al usuario un pequeño texto sobre lo que sucederá si presiona allí.

Para el ejemplo, una imagen representa el típico icono de realizar llamadas telefónicas (teléfono color verde) y, la otra, el de colgar o finalizar llamadas (teléfono color rojo).

El mensaje a mostrar es diferente para cada uno. En el recuadro gris se especifica un tipo de contenido que se podría mostrar:





El código de la GUI quedó definido como:

```
import tkinter as tk
from PIL import Image, ImageTk

class App:

    def __init__(self):
        self.root = tk.Tk()

        llamar_load = Image.open('llamar.png') # Imagen del teléfono color
verde        llamar_image = ImageTk.PhotoImage(llamar_load)

        colgar_load = Image.open('colgar.png') # Imagen del teléfono color
rojo        colgar_image = ImageTk.PhotoImage(colgar_load)

        fm = tk.Frame(self.root) # Frame contenedor de las imágenes
        fm.pack(fill='both', expand=True)

        llamar_btn = tk.Button(fm, image=llamar_image)
        llamar_btn.img_ref = llamar_image
        llamar_btn.grid(row=0, column=0)

        colgar_btn = tk.Button(fm, image=colgar_image)
        colgar_btn.img_ref = colgar_image
        colgar_btn.grid(row=0, column=1)

    def mainloop(self):
        self.root.mainloop()

if __name__ == '__main__':
    ejemplo = App()
    ejemplo.mainloop()
```

Implementar este tipo de prácticas podría ser bastante útil si se llegase a desarrollar una GUI con barras de herramientas, que muestren diversos iconos de comandos, ya que el usuario podría conocer sus funcionalidades tan solo con posicionar el cursor por encima de ellos.

¡Gracias por sus comentarios y/o sugerencias!

[Compartir](#) [Mejora esta pregunta](#) [Seguir](#)

formulada el 27 may. 2020 a las 9:18



gv-pro

371 2 12

Ordenado por:

Mayor puntuación (por defecto)



1 respuesta



2



Lo que buscas es lo que se conoce como "tooltip", "información sobre herramientas" o mejor "descripción emergente" en español.

Tkinter no tiene un widget dedicado a ello, pero no es muy difícil de implementar. Básicamente se suele implementar usando una ventana secundaria (`TopLevel`) `frameless` (sin bordes ni barra de título) con un `Label` o widget `Text` que se que se posiciona encima de la ventana principal.

El propio [IDLE](#), que se implementa en Tkinter mediante el paquete `idlelib`, hace uso de ellos, definidos en el módulo `tooltip`.

De hecho, podemos reutilizarlo:

```
import tkinter as tk
from idlelib.tooltip import Hovertip

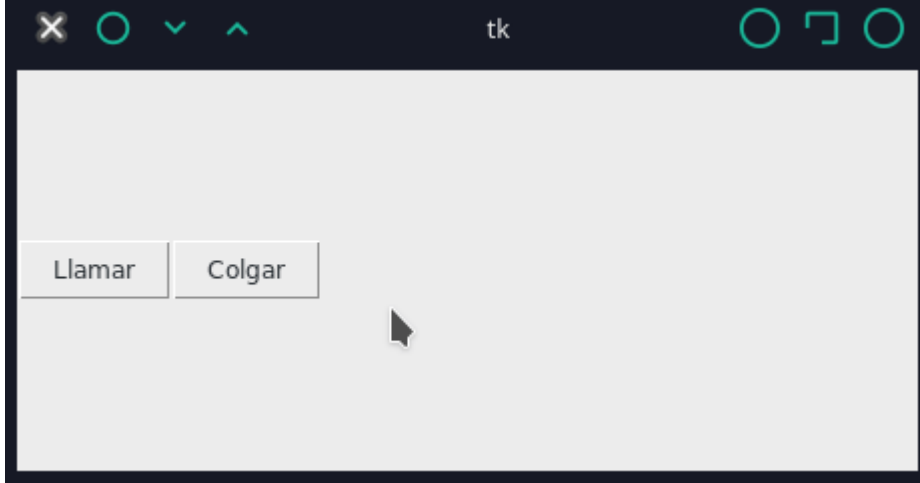
class CustomHovertip(Hovertip):
    def showcontents(self):
        label = tk.Label(
            self.tipwindow, text=f' "{self.text}" ', justify=tk.LEFT,
            bg="#151515", fg="#ffffff", relief=tk.SOLID, borderwidth=1,
            font=("Times New Roman", 12)
        )
        label.pack()

root = tk.Tk()
root.geometry("450x200")

llamar_btn = tk.Button(root, text="Llamar")
llamar_btn.pack(side="left")
Hovertip(llamar_btn, text="Si pulsa aquí va a llamar", hover_delay=500)

colgar_btn = tk.Button(root, text="Colgar")
colgar_btn.pack(side="left")
CustomHovertip(colgar_btn, text="Si pulsa aquí va a colgar", hover_delay=500)

root.mainloop()
```



Éste es el enlace al código fuente del módulo `idlelib.tooltip` en el repositorio de CPython en GitHub:

- <https://github.com/python/cpython/blob/3.8/Lib/idlelib/tooltip.py>.

Como comento, se puede reutilizar, pero dado que es un paquete destinado a implementar el IDLE exclusivamente, debes tener cuidado, dado que en un futuro la librería es susceptible de ser modificada sin previo aviso. No obstante, también puedes usar el código fuente directamente o usarlo como base para tu implementación personalizada.

Compartir Mejora esta respuesta Seguir

editada el 27 may. 2020 a las 11:32

respondida el 27 may. 2020 a las 9:58



FJSevilla

56k

7

38

60