

2.6. LAS CLASES Frame y LabelFrame



Estas clases sirven para crear frames, sin o con etiquetas. Un frame es un widget diseñado para agrupar visualmente otros widgets: etiquetas, botones, frames...

Cuando una GUI tiene cierta complejidad visual es mejor usar tantos frames como sea necesario. Incluso los interfaces más sencillos se gestionan mejor con frames, sobre todo para controlar su aspecto cuando el usuario puede redimensionar la aplicación.

Al derivar de la clase Widget, un frame comparte opciones y métodos con todos los widgets de Tkinter. Podemos asignar valor a sus opciones en su constructor, mediante su método `configure` o sucediendo a su diccionario.

En muchas ocasiones creamos widgets sin asignarle un identificador:

- Etiquetas con mensaje estético
- Botones con comando asociado

Cuando no necesitamos modificar opciones del widget no es necesario asignarle un nombre. En el caso de los frames, dado que los usamos para contener otros widgets, siempre es necesario que tengan nombre, para usarlo cuando creamos los widgets que van a contener necesitamos indicar el nombre de su padre (el frame que lo contiene).

El orden en que se colocan los widgets al usar la clase Pack es el que ha decidido el desarrollador al escribir el código, o no ser que se modifique ese orden usando las opciones `before` o `after` en el constructor del widget o en su método `pack_configure`. El siguiente ejemplo muestra este comportamiento: `l1` y `l3` se colocan en este orden, cada uno bajo el anterior por haber usado el método `pack()` sin argumentos. Antes de hacer la aplicación se ha decidido reordenar `l3` usando su método `pack_configure()`, indicando que se coloque antes de `l1`, por lo que la aplicación muestra inicialmente el frame `l3` sobre `l1`.

Los primeros widgets, a los que se les ha dado el nombre `l1` para poder utilizarlos en el código, muestran el aspecto visual de un enlace: texto azul subrayado y cursor "mano apuntando". Se captura el evento "el usuario hace clic sobre la etiqueta `l1`" que ejecutará la función `f()`, encargada de volver a colocar el frame `l3` después (debajo) del frame `l1`.



```
from tkinter import Tk, Label, Frame, LabelFrame
def P(e=None):
    lf3.pack_configure(after=lf1)
    v = Tk()
    lf1 = LabelFrame(v, text='Grupo 1')
    lf1.pack()
    l1 = Label(lf1, text='1', fg='blue', font=(None, 12, 'underline'),
    lf1.pack(cursor='pointinghand')
    Label(lf1, text='2').pack()
    lf2 = LabelFrame(lf1, text='Grupo 2')
    lf2.pack()
    Label(lf2, text='4').pack(side='left')
    Label(lf2, text='5').pack(side='left')
    lf3 = LabelFrame(v, text='Grupo 3')
    lf3.pack()
    f1 = Frame(lf3)
    f1.pack(side='left')
    Label(f1, text='6').pack(side='right')
    Label(f1, text='7').pack(side='right')
    f2 = Frame(lf3)
    f2.pack(side='left')
    Label(f2, text='8').pack()
    Label(f2, text='9').pack()
    lf3.pack_configure(before=lf1)
    l1.bind('<Button-1>', P)
    v.mainloop()
```