

2.4 LA CLASE Label



La clase `tkinter.Label` está diseñada para mostrar una imagen y/o un mensaje de texto multilinea⁽¹⁾, generalmente estético.

Dos objetos de esta clase no suelen interactuar con el usuario por lo que, por defecto, no reciben el foco del teclado.

Como todos los widgets de `tkinter`, derivos de la clase `tkinter.Widget`, por lo que puede usar los métodos de las clases `Misc`, `Pack`, `Place` y `Grid`. Su constructor es el mismo que el de casi todos los widgets:

```
__init__(master=None, cnt={}, **kwargs)
```

El parámetro posicional `master` puede ser una ventana, `TK` o `TopLevel`, o otro widget. En la jerarquía de objetos de la aplicación representa al padre del objeto:

- Será quien lo destruya cuando no sea necesario.
- " " gestione cómo se mostrará en la aplicación.

El segundo parámetro posicional, `cnt`, es un diccionario formado por claves válidas para este widget, sus valores. Es útil cuando queremos usar la misma configuración en diferentes widgets de la aplicación. El parámetro `**kwargs` permite que añadamos pares `clave = valor` para configurar el widget, prevaleciendo este último valor si se ha modificado la misma clave mediante el parámetro `cnt`.

La mayoría de opciones configurables de `Label` son comunes al resto de widgets de `tkinter`:

```
activebackground  activeforeground  anchor  background
bitmap  borderwidth  cursor  disabledforeground
font  foreground  highlightbackground  highlightcolor
highlightthickness  image  justify  padx  pady
relief  takefocus  text  textvariable  underline
wraplength
```

Las opciones `height`, `state` y `width` son específicas.

```
from tkinter import TK, Label
v = TK()
Label(v, text='Etiqueta').pack()
v.mainloop()
```

⁽¹⁾ La clase `tkinter.Message` quedó obsoleta al admitir múltiples líneas la clase `tkinter.Label`.



El método `pack()` se encarga de colocar el widget en el espacio visual de su padre. Si no nos diéramos las dimensiones del widget, `tkinter` los calcula para que se pueda mostrar su contenido:

- 1.- Determinamos el tamaño de la etiqueta, que los de mostrar el texto 'Etiquetas'.
- 2.- Determinamos el tamaño de la ventana, que los de mostrar la etiqueta.

Estudiamos algunos de sus opciones.

`text` es el mensaje de texto que mostramos, si no se le añade una imagen, se indica cómo mostrarlo respecto al texto.

`image` es la imagen que mostramos.

`compound` es la posición de la imagen respecto al texto. Puede tomar los valores `bottom`, `center`, `left`, `none`, `right` o `top`.

```
from tkinter import TK, Label, PhotoImage
v = TK()
img = PhotoImage(master=v, file='icono/Kind-text.gif')
Label(v, text='Bisquedo', image=img).pack()
Label(v, text='Bisquedo', image=img, compound='left').pack()
v.mainloop()
```

`anchor` indica dónde se coloca el texto respecto al espacio que tiene el widget para mostrarlo. Puede tomar los valores `n`, `ne`, `e`, `se`, `s`, `sw`, `w`, `nw` o `center`.

`bg` y `fg` determinan el color de la etiqueta y del texto, respectivamente.

`font` permite cambiar las características tipográficas del texto. Podemos usar uno tipo indicando `Arial`, `Times` y características de la fuente o usar.

```
from tkinter import TK, Label
import webbrowser as wb
def h(e=None):
    wb.open(['text'], 2)
v = TK()
l = Label(v, text='http://www.umh.es', font=('Courier New', 12,
    'underline'), bg='gray85', fg='blue', cursor='pointinghand')
l.pack()
l.bind('<Button-1>', h)
v.mainloop()
```