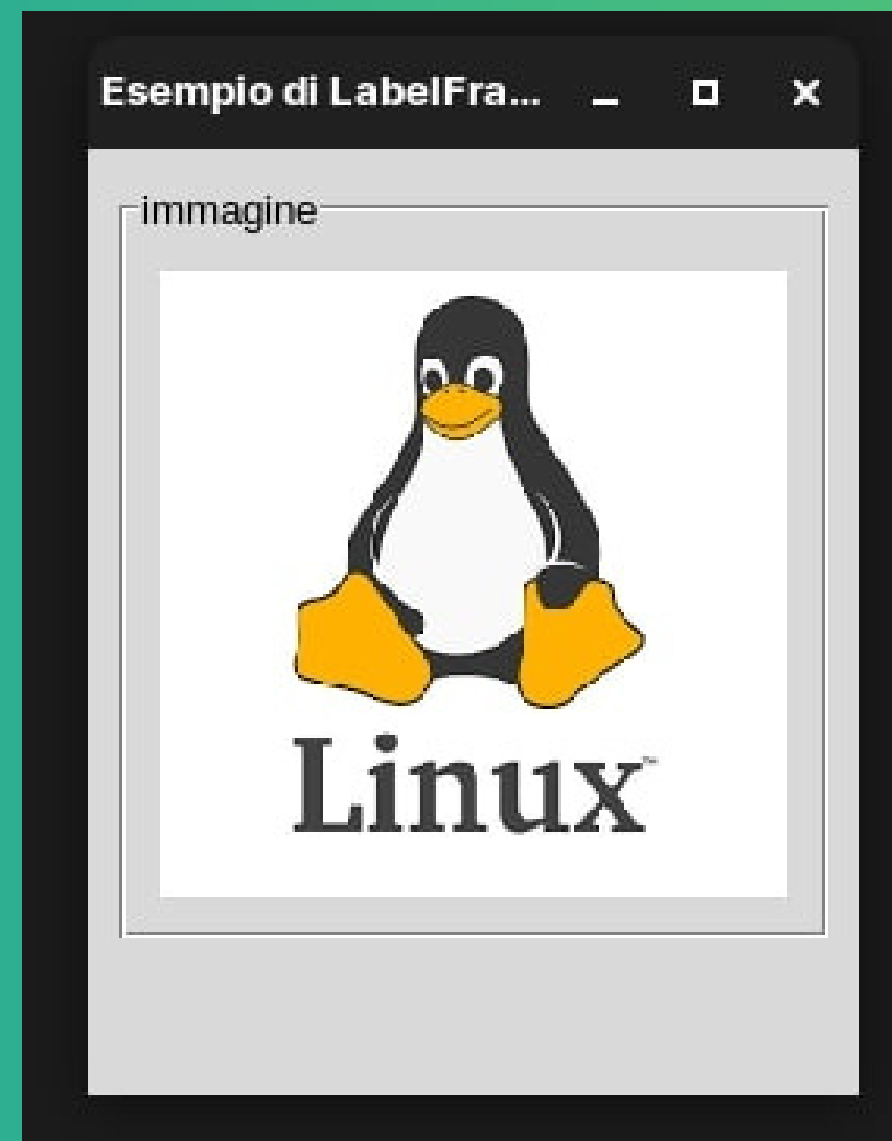


Cos'è

il label frame e un widget della Ttk (temed tkinter) per migliorare la grafica degli script di python



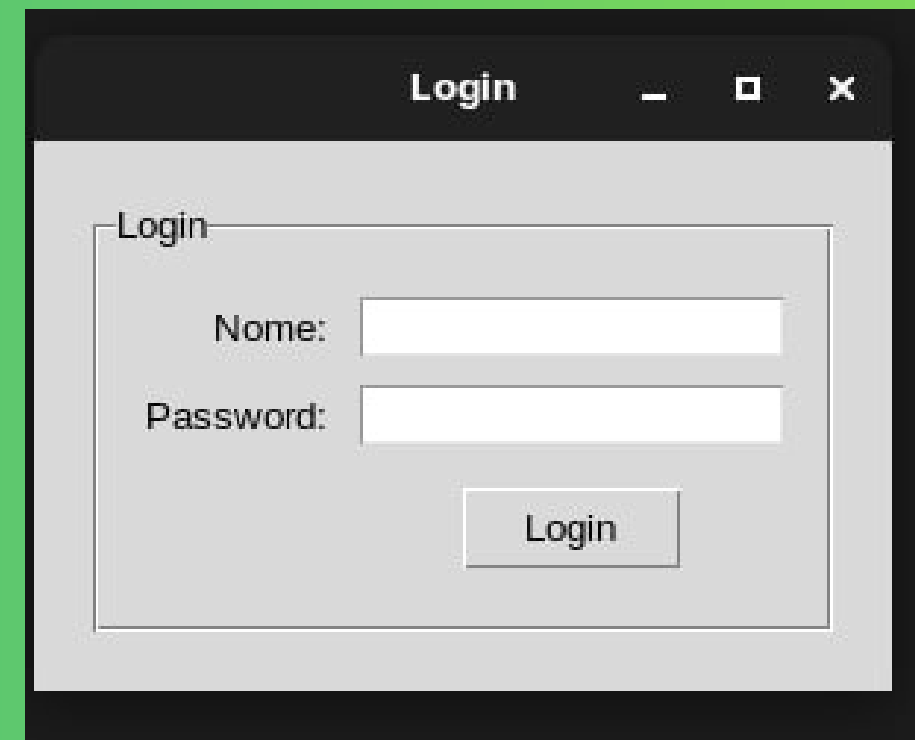
**serve ad ordinare logicamente nella finestra grafica
più widget come tasti, text, scritte**

**dividere per argomento i widget
presenti nello schermo**

A cosa serve

**puo gestire le azioni dell'utente
come la pressione di button o
caselle di testo e quando viene
ridimensionato adatta i
componenti contenuti**

**puo contenere le immagini sia come sfondo che
come oggetto e puo modificare lo sfondo**



```
8 # label frame
9 lf = ttk.LabelFrame(root, text='ancora')
10 lf.grid(column=0, row=0, padx=20, pady=20, sticky=tk.NSEW)
```

alla creazione del labelframe si passa come parametro aggiuntivo la variabile text che contiene la stringa con il titolo del frame che poi con il comando grid() si può tramite i parametri quali column e row decidere il posizionamento su righe e colonne e con padx e pady si possono impostare gli spessori dello spazio vuoto dalle scritte

```

32
33 # creazione radio button e inserimento
34 for key, value in anchors.items():
35     # creazione radio button
36     radio = ttk.Radiobutton(
37         lf,
38         text=key.upper(),
39         value=key,
40         command=change_label_anchor,
41         variable=anchor_var
42     ).grid(**value, padx=10, pady=10, sticky=tk.NSEW)
43

```

```

11
12 anchor_var = tk.StringVar()
13 anchors = {
14     'nw': {'row': 0, 'column': 1},
15     'n': {'row': 0, 'column': 2},
16     'ne': {'row': 0, 'column': 3},
17     'en': {'row': 1, 'column': 4},
18     'e': {'row': 2, 'column': 4},
19     'es': {'row': 3, 'column': 4},
20     'se': {'row': 4, 'column': 3},
21     's': {'row': 4, 'column': 2},
22     'sw': {'row': 4, 'column': 1},
23     'ws': {'row': 3, 'column': 0},
24     'w': {'row': 2, 'column': 0},
25     'wn': {'row': 1, 'column': 0}
26 }
27

```

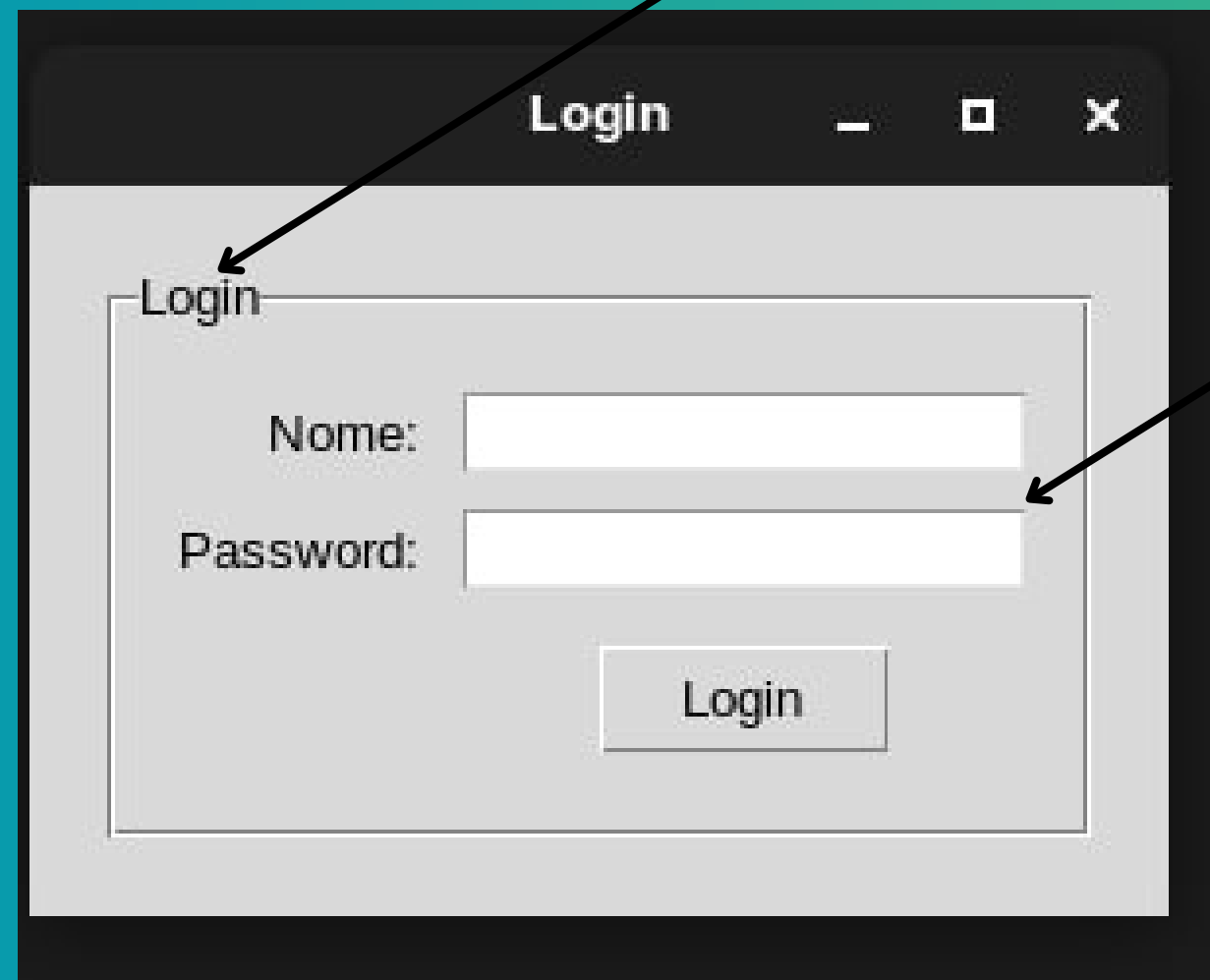
il label frame usando le funzioni della ttk può partendo da una lista di elementi o un dizionario con un ciclo for alla creazione datogi uno spostamento predefinito allinearli gia graficamente e mantenere con il parametri padx e pady una distanza verticale e orizzontale predefinita tra ogni tasto

```

22 for alignment in alignments:
23     # creazione radio button
24     radio = ttk.Radiobutton(lf, text=alignment, value=alignment, variable=alignment_var)
25     radio.grid(column=grid_column, row=0, padx=10, pady=10)
26     # grid column
27     grid_column += 1
28

```

Etichetta personalizzabile: Il LabelFrame può includere un'etichetta personalizzabile nella parte superiore del frame. L'etichetta può contenere testo o immagini e può essere posizionata ovunque attorno al frame



Allineamento dei widget interni: I widget inclusi all'interno del LabelFrame possono essere allineati in base a una griglia, utilizzando il metodo grid() o allineati in verticale o orizzontale utilizzando il metodo pack(). Ciò permette di organizzare i widget all'interno del frame in modo ordinato e coerente.

Gestione degli eventi: Il LabelFrame può gestire eventi come il clic del mouse o la pressione dei tasti e può essere collegato a funzioni di callback che vengono eseguite quando si verificano determinati eventi

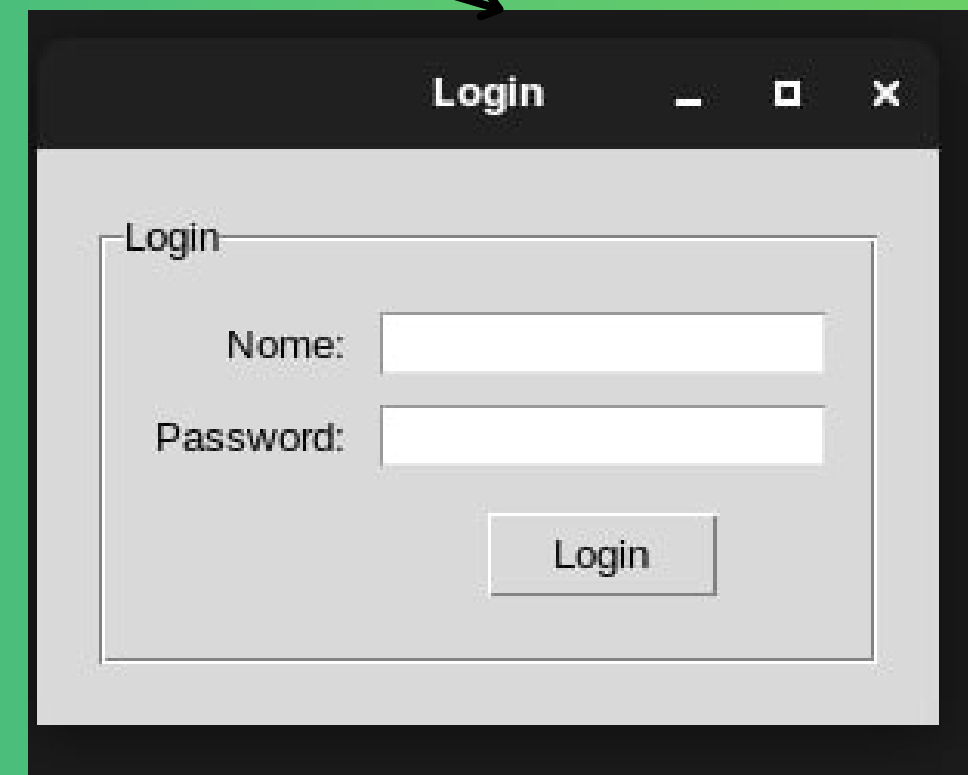
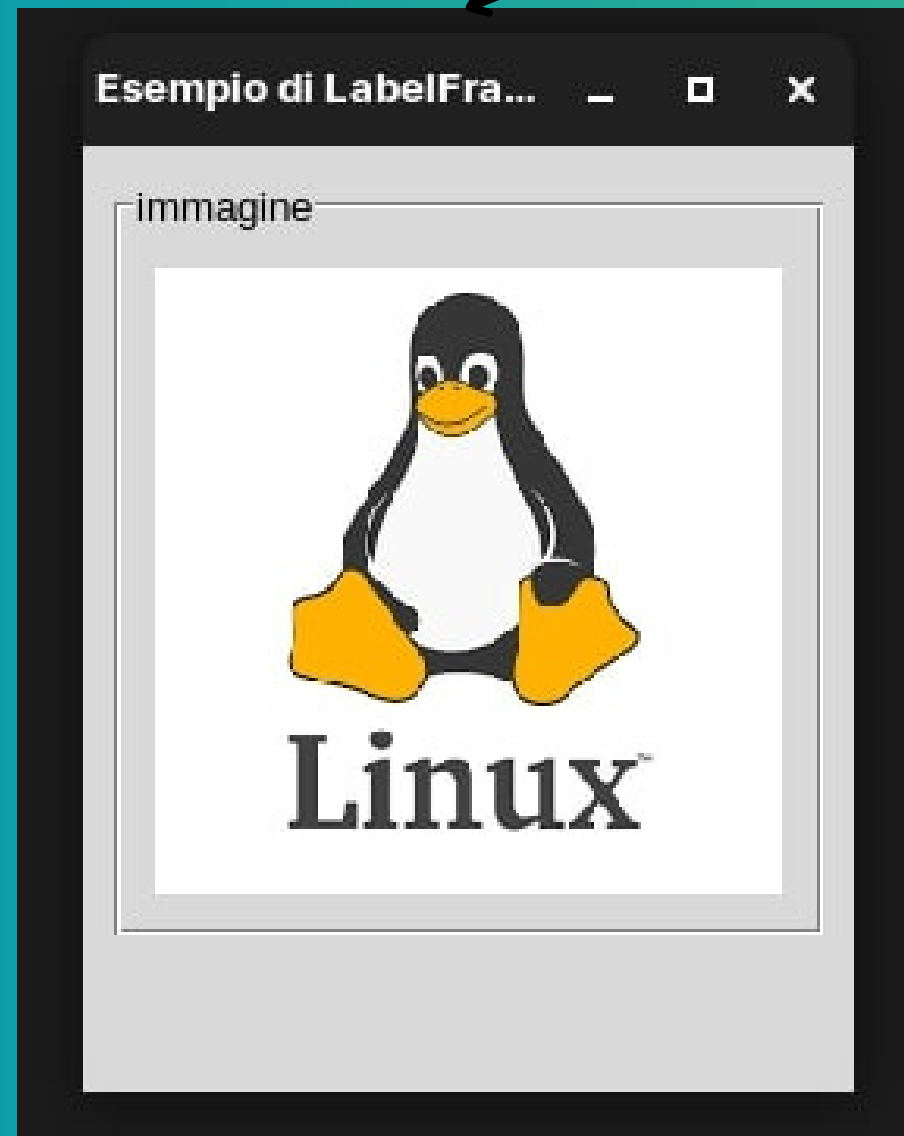
```
11  
12 # Caricamento dell'immagine  
13 immagine = Image.open("linux.png")  
14 immagine = immagine.resize((200, 200), Image.ANTIALIAS)  
15 immagine_tk = ImageTk.PhotoImage(immagine)
```

Opzioni di ridimensionamento: Il LabelFrame supporta diverse opzioni per il ridimensionamento dei widget interni quando il frame stesso viene ridimensionato. Ad esempio, è possibile specificare se i widget devono ridimensionarsi in base alla dimensione del frame o se devono mantenere la loro dimensione originale.

il widget label frame ha i seguenti parametri aggiuntivi che permettono di avere personalizzazioni aggiuntive

- **bg: il background del widget (standard : grigio)**
- **bd : il bordo del label frame (standard :2px)**
- **cursor : permette di cambiare la forma del cursore**
- **font : permette di settare il font delle scritte**
- **height: permette di settare l'altezza nel widget**
- **width: permette di settare la larghezza**
- **labelAnchor: permette di ancorare tramite la posizione in righe e colonne sull schermo**
- **highlightbackground : colore sfondo quando il cursore non è sopra al labelframe**
- **highlightcolor: colore per le immagini per il focus con passaggio del cursore del mouse**
- **highlightthickness: parametro per colore relativo al focus per il bordo**
- **relief : serve per impostare il rilievo del widget dalla finestra (standard: FLAT cioè non è in rilievo)**
- **text : testo che viene inserito come titolo del frame**

Compatibilità con gli altri widget di Tkinter: Il LabelFrame può essere utilizzato insieme ad altri widget di Tkinter per creare interfacce utente complesse e personalizzate. Ad esempio, può essere utilizzato insieme a pulsanti, caselle di testo, menu , immagini e altri widget per creare un'interfaccia utente completa e funzionale.



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

Per includere le immagini con il widget tkinter il modulo PIL (Pillow) viene importato per lavorare con le immagini.

```
11
12 # Caricamento dell'immagine
13 immagine = Image.open("linux.png")
14 immagine = immagine.resize((200, 200), Image.ANTIALIAS)
15 immagine_tk = ImageTk.PhotoImage(immagine)
```

quando si lavora con le immagini è sempre consigliato quando si danno i parametri e consigliato passare anche il parametro ANTIALIAS che serve per evitare che quando si espande la finestra manualmente tirando un bordo l'immagine rimanga deformata

il label frame permette di modificare tramite apposite funzioni lo sfondo di una finestra sia passando un colore in una variabile sia dando subito il colore con il parametro XXX.TLabelframe (XXX staa per il colore selezionato)

```
label_frame.configure(style="Custom.TLabelframe")  
  
# Creazione di uno stile personalizzato per il LabelFrame  
style = ttk.Style()  
style.configure("Custom.TLabelframe", background="lightblue")
```