

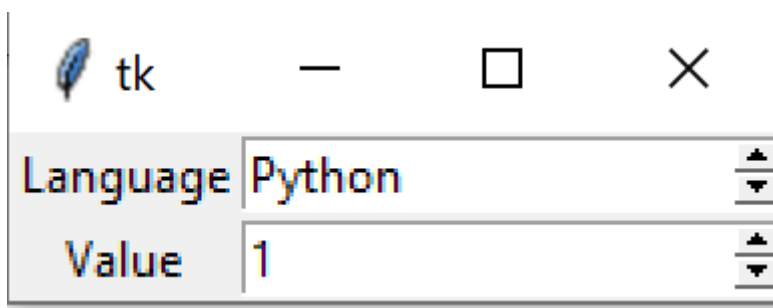
## BARRA LEONARDO (SpinBox)

Lo spinbox è un widget di tkinter che consente all'utente di selezionare un valore da una lista predefinita di opzioni. Tuttavia, a differenza della casella di testo, lo spinbox ha dei bottoni di incremento e decremento che consentono all'utente di selezionare un valore specifico nella lista.



Il valore selezionato dallo spinbox può essere utilizzato come una variabile in un programma Python per svolgere alcune operazioni.

Lo spinbox ha diversi parametri che possono essere utilizzati per personalizzare il widget. Ad esempio, il parametro `from_` specifica il valore minimo selezionabile, mentre il parametro `to` specifica il valore massimo selezionabile. Il parametro `increment` specifica l'incremento o il decremento del valore del widget quando l'utente preme i bottoni di incremento o decremento.



## PRIMO ESEMPIO

Questo è primo esempio di codice Python che utilizza lo spinbox per selezionare un valore di temperatura in gradi Celsius e convertirlo in gradi Fahrenheit: In questo esempio, ho creato un'interfaccia utente con uno spinbox che consente all'utente di selezionare un valore di temperatura in gradi Celsius. Il valore selezionato viene quindi convertito in gradi Fahrenheit quando l'utente preme il pulsante "Converti". Il risultato viene visualizzato in una label.

```
import tkinter as tk

def converti_temperatura():
    celsius = float(celsius_var.get())
    fahrenheit = (celsius * 9/5) + 32
    fahrenheit_var.set(fahrenheit)

root = tk.Tk()
root.geometry('200x100')

celsius_var = tk.StringVar()
celsius_spinbox = tk.Spinbox(root, from_=-273.15, to=100,
textvariable=celsius_var, increment=0.1)
celsius_spinbox.pack()

converti_button = tk.Button(root, text='Converti',
command=converti_temperatura)
converti_button.pack()

fahrenheit_var = tk.StringVar()
fahrenheit_label = tk.Label(root, textvariable=fahrenheit_var)
fahrenheit_label.pack()
root.mainloop()
```

## SECONDO ESEMPIO

In questo esempio, ho creato uno spinbox con una lista di opzioni predefinite. L'utente può selezionare una delle opzioni nella lista utilizzando i bottoni di incremento e decremento. Quando l'utente preme il pulsante "Mostra opzione", il valore selezionato viene stampato nella console.

```
import tkinter as tk

def opzione_selezionata():
    opzioneselezionata = opzione_var.get()
    print(f'Opzione selezionata: {opzioneselezionata}')

root = tk.Tk()
root.geometry('200x100')

opzioni = ['Opzione 1', 'Opzione 2', 'Opzione 3']
opzione_var = tk.StringVar(value=opzioni[0])
opzione_spinbox = tk.Spinbox(root, values=opzioni,
textvariable=opzione_var)
opzione_spinbox.pack()

button = tk.Button(root, text='Mostra opzione',
command=opzione_selezionata)
button.pack()

root.mainloop()
```

## TERZO ESERCIZIO

```
import tkinter as tk

def valida_numero(value):
    if not value.isdigit():
        return False
    if int(value) < 0 or int(value) > 100:
        return False
    return True

def numeroselezionato():
    numeroselezionato = numero_var.get()
    print(f'numero selezionato: {numeroselezionato}')

root = tk.Tk()
root.geometry('200x100')

numero_var = tk.StringVar(value='0')
numero_spinbox = tk.Spinbox(root, from_=0, to=100,
textvariable=numero_var, validate='key',
validatecommand=(root.register(valida_numero), '%P'))
numero_spinbox.pack()

show_button = tk.Button(root, text='Mostra numero',
command=numeroselezionato)
show_button.pack()

root.mainloop()
```

In questo esempio, utilizziamo una funzione di validazione personalizzata per impedire all'utente di selezionare un valore non valido. La funzione di validazione controlla se il valore inserito dall'utente è un intero compreso tra 0 e 100. Se il valore inserito non è valido, la funzione di validazione restituisce False, impedendo all'utente di selezionare quel valore.

Quando l'utente preme il pulsante "Mostra numero", il valore selezionato viene stampato nella console.

Per implementare la funzione di validazione, utilizziamo il metodo `register()` del widget principale per registrare la funzione di validazione personalizzata come una funzione di callback. Poi, la passiamo come parametro alla proprietà `validatecommand` dello spinbox. In questo modo, ogni volta che l'utente digita un valore nel campo, la funzione di validazione viene chiamata per verificare se il valore è valido.

Nel dettaglio..... nella creazione dell'oggetto Spinbox, viene passata la stringa 'key' all'opzione `validate`. Questo significa che la funzione di validazione viene chiamata ogni volta che viene inserito un tasto nella casella di testo dell'oggetto Spinbox. In particolare, la funzione di validazione personalizzata registrata tramite `register()` viene chiamata con `%P` come argomento, dove `%P` rappresenta il valore attuale del widget Spinbox.

La funzione `valida_numero(value)` è la funzione di validazione personalizzata registrata con `register()`. Questa funzione prende un valore come argomento (che rappresenta il valore attuale del widget Spinbox) e controlla se il valore è un numero intero compreso tra 0 e 100

## QUARTO ESERCIZIO

```
import tkinter as tk

root = tk.Tk()

numero_var = tk.StringVar(value='0')
numero_spinbox = tk.Spinbox(root, from_=0, to=10,
textvariable=numero_var, wrap=True)
numero_spinbox.pack()

root.mainloop()
```

In questo esempio, viene creato un oggetto Spinbox con `wrap=True`. Questo significa che, quando l'utente raggiunge il limite inferiore o superiore dell'intervallo specificato (`from_` e `to`), il valore della Spinbox "avvolgerà" al limite opposto dell'intervallo.

Ad esempio, se l'intervallo è da 0 a 10 e l'utente seleziona il valore 10, attivando il parametro `wrap`, la selezione passa automaticamente a 0 quando l'utente preme il pulsante successivo. In modo simile, se l'utente seleziona il valore 0 e continua a premere il pulsante precedente, la selezione passa a 10 grazie all'attivazione del parametro `wrap`