

I.I.S. Denina sez. Rivoira - Informatica

Relazione widget Combobox

Audisio Nicolò

13 aprile 2023

Indice

1	Introduzione	3
2	Codice	3
3	Spiegazione codice	4
4	Codice concreto di esempio	5

1 Introduzione

Il componente Combobox di Tkinter rappresenta un'implementazione di un widget grafico che consente di realizzare un'interfaccia utente completa ed intuitiva in accordo con le migliori pratiche del design della User Experience (UX). In particolare, questo componente permette all'utente di selezionare una delle opzioni disponibili attraverso l'utilizzo del cosiddetto "menù a tendina", il quale rappresenta una lista di scelte visualizzata su un pannello a scomparsa. In questo modo, l'utente può accedere e navigare tra le diverse opzioni disponibili con facilità e rapidità, migliorando in tal modo l'esperienza d'uso dell'applicazione.

2 Codice

```
# Audisio Combobox
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox

class Finestra(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Audisio Combobox")
        self.geometry("500x500")
        self.resizable(0, 0)
        self.create_widgets()

    def create_widgets(self):
        opzioni = ["opzione 1", "opzione 2", "opzione 3"]

        self.combo = ttk.Combobox(self, values=opzioni)
        self.combo.pack()

        self.button = tk.Button(self, text="Scelta", command=self.visualizza_selezione)
        self.button.pack()

    def visualizza_selezione(self):
        selezione = self.combo.get()
        tk.messagebox.showinfo("Selezione", "Hai selezionato: {}".format(selezione))

root = Finestra()
root.mainloop()
```

3 Spiegazione codice

Il presente codice consente di creare una finestra contenente un Combobox mediante l'utilizzo della libreria Tkinter. In particolare, tre diverse librerie della libreria standard di Python sono importate: `tkinter`, `ttk` e `messagebox`. La libreria `tkinter` rappresenta il modulo principale per la creazione di interfacce grafiche in Python. Essa mette a disposizione un insieme di widget, che possono essere posizionati all'interno della finestra dell'applicazione, gestisce gli eventi dell'interfaccia utente e fornisce funzionalità per la gestione dei colori, delle immagini, dei font e della posizione degli oggetti nella finestra. La libreria `ttk`, invece, offre una serie di widget aggiuntivi rispetto a quelli disponibili in `tkinter`, i quali includono opzioni di stile aggiuntive e utilizzano uno stile uniforme tra piattaforme per garantire una maggiore coerenza estetica. Infine, la funzione `messagebox` rappresenta un sotto-modulo della libreria `tkinter` e fornisce una serie di funzioni per la creazione di finestre di messaggio popup.

Il codice inizia con l'importazione delle librerie `Tkinter` e `ttk`. Inoltre, viene importata la funzione `messagebox` dalla libreria `Tkinter`, la quale viene utilizzata per mostrare una finestra di messaggio. La classe `Finestra` eredita dalla classe `Tk`, e pertanto, mediante la funzione `init()`, viene creata una finestra principale. All'interno del `init()`, viene impostato il titolo della finestra tramite il comando `"self.title("")"`, le dimensioni tramite il comando `"self.geometry("")"` e la proprietà di ridimensionamento tramite il comando `"self.resizable("")"`. Successivamente, viene chiamato il metodo `create_widgets()` per creare i widget all'interno della finestra.

Nel metodo `create_widgets()`, viene creato un oggetto `Combobox` mediante la classe `ttk.Combobox()`, che richiede come argomenti il frame della finestra e una lista di opzioni per il `Combobox`. Viene quindi creato un pulsante che richiama il metodo `visualizza_selezione()` quando viene premuto. Il metodo `visualizza_selezione()` si occupa di ottenere il valore selezionato dal `Combobox` mediante il metodo `get()` e, successivamente, utilizza la funzione `messagebox.showinfo()` per mostrare una finestra di messaggio contenente la selezione effettuata dall'utente.

Infine, viene creato un oggetto `Finestra` e viene avviato il loop principale `mainloop()` per visualizzare la finestra e gestire gli eventi. Nel codice in esame, la libreria `tkinter` viene importata utilizzando l'alias `tk` mediante la sintassi `import tkinter as tk`, allo scopo di rendere più breve la sintassi di chiamata ai metodi e alle classi della libreria. La libreria `ttk`, invece, viene importata mediante la sintassi `from tkinter import ttk`, che consente di chiamare direttamente i widget specifici di `ttk` senza utilizzare l'alias `ttk`. In particolare, viene utilizzato il widget `Combobox` di `ttk` per creare una lista a discesa. Infine, la funzione `showinfo()` della libreria `messagebox` viene importata mediante la sintassi `from tkinter import messagebox`, che permette di utilizzarla direttamente senza utilizzare.

Il seguente codice è presente su GitHub ¹

¹GitHub: <https://github.com/NicoloAudisio/comboBox/blob/main/combobox.py>

4 Codice concreto di esempio

In seguito viene presentato un esempio concreto di codice che illustra l'utilizzo del componente Combobox all'interno di un modulo di prenotazione per un Hotel. Attraverso tale esempio è possibile visualizzare l'interazione dell'utente con la parte front-end del programma e il conseguente salvataggio della selezione effettuata. Tale selezione verrà successivamente utilizzata per generare un resoconto completo della prenotazione

```
# Audisio esempio utilizzo Combobox
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox

class Finestra(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Prenotazione Hotel")
        self.geometry("700x500")
        self.resizable(0, 0)

    # --- FUNZIONI ---
    def prenota(self):
        try:
            with open("prenotazioni.txt", "a") as f:
                servizi = []
                if self.piscina.get() == True:
                    servizi.append("Piscina")
                if self.parcheggio.get():
                    servizi.append("Parcheggio")
                if self.wifi.get():
                    servizi.append("Wi-Fi")
                servizi_str = ",".join(servizi)
                prenotazione = f"Ripilogo prenotazione\nIl sig. {self.cognome_entry.get()}
                {self.nome_entry.get()} ha prenotato una stanza
                {self.tipologia_camera.get()} per {self.notti.get()} notti\nServizi
                richiesti: {servizi_str}\nTipologia pensione: {self.combo.get()}\n"
                f.write(prenotazione)
                self.lbl_conferma.config(text = "Prenotazione effettuata con successo!")
                self.lbl_conferma.config(fg = 'green')
        except ValueError:
            messagebox.showerror("Errore", "Errore nell'inserimento dei dati")

    def crea_widgets(self):
        # --- LABEL HOTEL BELLAVISTA
        lbl_hotel = tk.Label(self, text="Hotel Rimini", font=("Time New Romans", 20))
        lbl_hotel.pack(pady=20)
```

```

# --- COGNOME E NOME---
frm_dati = tk.Frame(self)
lbl_cognome = tk.Label(frm_dati, text="Cognome")
lbl_cognome.pack(side=tk.LEFT, padx=10)
self.cognome_entry = tk.Entry(frm_dati)
self.cognome_entry.pack(side=tk.LEFT)

lbl_nome = tk.Label(frm_dati, text="Nome")
lbl_nome.pack(side=tk.LEFT, padx=10)
self.nome_entry = tk.Entry(frm_dati)
self.nome_entry.pack(side=tk.LEFT)
frm_dati.pack(pady=10)

# --- SCELTA CAMERA ---
lbl_camera = tk.Label(self, text="Tipologia camera: ")
lbl_camera.pack()
self.tipologia_camera = tk.StringVar()
self.tipologia_camera.set("Singola")
rbt_singola = tk.Radiobutton(self, text="Singola",
variable=self.tipologia_camera, value="Singola")
rbt_singola.pack()
rbt_matrimoniale = tk.Radiobutton(self, text="Matrimoniale",
variable=self.tipologia_camera, value="Matrimoniale")
rbt_matrimoniale.pack()

# --- NUMERO NOTTI ---
lbl_notti = tk.Label(self, text="Numero notti")
lbl_notti.pack()
self.notti = tk.StringVar()
self.notti.set("1")
spin_notti = tk.Spinbox(self, from_=1, to=30, textvariable=self.notti)
spin_notti.pack()

# --- SCLTA PENSIONE ---
lbl_piano = tk.Label(self, text="Tipologia di pensione")
lbl_piano.pack()
opzioni = ["Solo colazione", "Mezza pensione", "Pensione completa"]
self.combo = ttk.Combobox(self, values=opzioni)
self.combo.pack()

```

```

# --- SERVIZI EXTRA ---
lbl_serviziExtra = tk.Label(self, text="Servizi extra")
lbl_serviziExtra.pack()
self.piscina = tk.BooleanVar()
chk_piscina = tk.Checkbutton(self, text="Piscina", variable=self.piscina)
chk_piscina.pack()
self.parcheggio = tk.BooleanVar()
chk_parcheggio = tk.Checkbutton(self, text="Parcheggio", variable=self.parcheggio)
chk_parcheggio.pack()
self.wifi = tk.BooleanVar()
chk_wifi = tk.Checkbutton(self, text="Wi-Fi", variable=self.wifi)
chk_wifi.pack()

# --- BOTTONE PRENOTA ---
btn_prenota = tk.Button(self, text="Prenota", command=self.prenota)
btn_prenota.pack(pady=20)

# --- LABEL CONFERMA ---
self.lbl_conferma = tk.Label(self, text="")
self.lbl_conferma.pack()

root = Finestra()
root.crea_widgets()
root.mainloop()

```

Il seguente codice è presente su GitHub ²

Immagine dell'esempio

Nell'immagini sopra riportata è possibile osservare l'intera pagina del modulo di registrazione, nonché la selezione effettuata mediante il componente ComboBox.

²GitHub: <https://github.com/NicoloAudisio/comboBox/blob/main/esempioConcretoCombobox.py>