

Combobox リファレンス

ドロップダウンリストから項目を選択できる **Combobox** ウィジェットについての詳細なリファレンスです。

概要

Combobox ウィジェットは、複数の選択肢をドロップダウンリストとして表示し、ユーザーが一つの項目を選択できるコントロールです。Entry ウィジェットの機能も併せ持ち、リストにない値をユーザーが直接入力することも可能です。 **tkinter.ttk** モジュールの一部として提供されます。

基本的な使用方法

シンプルなコンボボックス

```
import tkinter as tk
from tkinter import ttk

def on_selection(event):
    selection = combobox.get()
    print(f"選択された項目: {selection}")

app = tk.Tk()
app.title("Comboboxの例")
app.geometry("300x200")

# コンボボックスの作成
values = ["りんご", "みかん", "バナナ", "ぶどう", "いちご"]
combobox = ttk.Combobox(app, values=values)
combobox.set("りんご") # 初期値を設定
combobox.bind("<<ComboboxSelected>>", on_selection)
combobox.pack(pady=20)

app.mainloop()
```

クラスベースでのコンボボックス

```
import tkinter as tk
from tkinter import ttk

class ComboboxApp(tk.Tk):
    def __init__(self):
        super().__init__()

        self.title("Comboboxの例 (クラスベース) ")
        self.geometry("300x200")

        self.create_widgets()

    def create_widgets(self):
        # コンボボックスの作成
        values = ["りんご", "みかん", "バナナ", "ぶどう", "いちご"]
        self.combobox = ttk.Combobox(self, values=values)
        self.combobox.set("りんご") # 初期値を設定
        self.combobox.bind("<<ComboboxSelected>>", self.on_selection)
        self.combobox.pack(pady=20)

    def on_selection(self, event):
        selection = self.combobox.get()
        print(f"選択された項目: {selection}")

if __name__ == "__main__":
    app = ComboboxApp()
    app.mainloop()
```

主要なオプション

オプション	説明
values	ドロップダウンリストに表示される項目のタプルまたはリスト。
textvariable	tk.StringVar などの変数を指定し、選択値と同期します。
state	コンボボックスの状態 (normal , readonly , disabled)。
width	コンボボックスの幅を文字数で指定します。
font	フォントを指定。
justify	テキストの配置 (left , center , right)。

オプション	説明
<code>exportselection</code>	選択をクリップボードにエクスポートするかどうか。
<code>postcommand</code>	ドロップダウンが開かれる前に実行される関数。

主要なメソッド

メソッド	説明
<code>get()</code>	現在の値を取得します。
<code>set(value)</code>	値を設定します。
<code>current(index=None)</code>	現在の選択インデックスを取得または設定します。
<code>configure(values=...)</code>	valuesオプションを動的に変更します。

イベント

イベント	説明
<code><<ComboboxSelected>></code>	ユーザーがドロップダウンから項目を選択したときに発生します。

実用的な例

動的な選択肢の変更

```
import tkinter as tk
from tkinter import ttk

class DynamicComboboxApp(tk.Tk):
    def __init__(self):
        super().__init__()

        self.title("動的コンボボックス")
        self.geometry("400x300")

        self.create_widgets()

    def create_widgets(self):
        # カテゴリ選択
        tk.Label(self, text="カテゴリを選択:", font=("Arial", 12)).pack(pady=10)

        self.category_var = tk.StringVar()
        self.category_combo = ttk.Combobox(
            self,
            textvariable=self.category_var,
            values=["果物", "野菜", "肉類", "魚類"],
            state="readonly"
        )
        self.category_combo.bind("<<ComboboxSelected>>", self.on_category_changed)
        self.category_combo.pack(pady=5)

        # 商品選択
        tk.Label(self, text="商品を選択:", font=("Arial", 12)).pack(pady=(20, 10))

        self.product_var = tk.StringVar()
        self.product_combo = ttk.Combobox(
            self,
            textvariable=self.product_var,
            state="disabled"
        )
        self.product_combo.bind("<<ComboboxSelected>>", self.on_product_selected)
        self.product_combo.pack(pady=5)

        # 結果表示
        self.result_label = tk.Label(self, text="", font=("Arial", 11), fg="blue")
        self.result_label.pack(pady=20)

        # 商品データ
        self.products = {
            "果物": ["りんご", "みかん", "バナナ", "ぶどう", "いちご"],
            "野菜": ["キャベツ", "人参", "玉ねぎ", "じゃがいも", "トマト"],
            "肉類": ["牛肉", "豚肉", "鶏肉", "ラム肉"],
            "魚類": ["サーモン", "マグロ", "アジ", "サバ", "イワシ"]
        }

    def on_category_changed(self, event):
        category = self.category_var.get()
        if category in self.products:
            # 商品選択肢を更新
            self.product_combo.configure(values=self.products[category])
            self.product_combo.set("") # 選択をリセット
            self.product_combo.configure(state="readonly")
            self.result_label.config(text="")

    def on_product_selected(self, event):
        category = self.category_var.get()
        product = self.product_var.get()
        self.result_label.config(text=f"選択: {category} - {product}")
```

```
if __name__ == "__main__":
    app = DynamicComboboxApp()
    app.mainloop()
```

検索可能なコンボボックス

```
import tkinter as tk
from tkinter import ttk

class SearchableComboboxApp(tk.Tk):
    def __init__(self):
        super().__init__()

        self.title("検索可能コンボボックス")
        self.geometry("400x350")

        self.create_widgets()

    def create_widgets(self):
        tk.Label(self, text="国名を入力または選択:", font=("Arial", 12)).pack(pady=10)

        # 国名リスト
        self.countries = [
            "日本", "アメリカ", "イギリス", "フランス", "ドイツ", "イタリア",
            "スペイン", "カナダ", "オーストラリア", "ブラジル", "インド",
            "中国", "韓国", "タイ", "ベトナム", "シンガポール", "マレーシア"
        ]

        self.country_var = tk.StringVar()
        self.country_combo = ttk.Combobox(
            self,
            textvariable=self.country_var,
            values=self.countries,
            width=30
        )
        self.country_combo.pack(pady=5)

        # キー入力に応じて候補を絞り込む
        self.country_combo.bind('<KeyRelease>', self.on_key_release)
        self.country_combo.bind("<<ComboboxSelected>>", self.on_selection)

        # 結果表示エリア
        tk.Label(self, text="選択結果:", font=("Arial", 12)).pack(pady=(20, 5))

        self.result_text = tk.Text(self, width=40, height=10, font=("Arial", 10))
        self.result_text.pack(pady=5)

        # 情報表示ボタン
        info_button = tk.Button(self, text="国情報を表示", command=self.show_country_info)
        info_button.pack(pady=10)

        # 国の情報（サンプル）
        self.country_info = {
            "日本": "首都: 東京\n人口: 約1億2500万人\n通貨: 円",
            "アメリカ": "首都: ワシントンD.C.\n人口: 約3億3000万人\n通貨: ドル",
            "イギリス": "首都: ロンドン\n人口: 約6700万人\n通貨: ポンド",
            "フランス": "首都: パリ\n人口: 約6800万人\n通貨: ユーロ",
            "ドイツ": "首都: ベルリン\n人口: 約8300万人\n通貨: ユーロ"
        }

    def on_key_release(self, event):
        # 入力値に基づいて候補を絞り込む
        typed = self.country_var.get().lower()
        if typed == '':
            self.country_combo.configure(values=self.countries)
        else:
            filtered = [country for country in self.countries
                        if typed in country.lower()]
            self.country_combo.configure(values=filtered)

    def on_selection(self, event):
        selected = self.country_var.get()
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(1.0, f"選択された国: {selected}")

    def show_country_info(self):
        selected = self.country_var.get()
        if selected in self.country_info:
            info = self.country_info[selected]
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(1.0, f"国名: {selected}\n\n{info}")
        else:
            self.result_text.delete(1.0, tk.END)
            self.result_text.insert(1.0, f"'{selected}' の情報は登録されていません。")

if __name__ == "__main__":
    app = SearchableComboboxApp()
    app.mainloop()
```

ベストプラクティス

プラクティス	説明
適切な状態の設定	<code>readonly</code> 状態を使用して、リストからの選択のみを許可することができます。

プラクティス	説明
イベントハンドリング	<<ComboboxSelected>> イベントを使用して選択変更を検知します。
動的な選択肢	<code>configure(values=...)</code> を使用して選択肢を動的に変更できます。
初期値の設定	<code>set()</code> メソッドを使用して適切な初期値を設定します。
検索機能の実装	キー入力イベントを使用して検索機能を実装できます。

参考リンク

- [Python Docs - tkinter.ttk.Combobox](#)
- [TkDocs - Combobox](#)