Label リファレンス

テキストや画像を表示するための `Label` ウィジェットについての詳細なリファレンスです。

概要

`Label` ウィジェットは、ユーザーに情報を表示するための基本的なコントロールです。一行または複数行のテキスト、および画像を表示することができます。表示専用であり、ユーザーからの入力を受け付けることはありません。

基本的な使用方法

テキストラベルの作成

```
import tkinter as tk

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

# ラベルを作成し、ウィンドウに配置
label = tk.Label(app, text="こんにちは, tkinter!")
label.pack(pady=20) # pack()でウィジェットを配置

app.mainloop()
```

クラスベースでのラベル作成

```
import tkinter as tk

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

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

        self.create_widgets()

    def create_widgets(self):
        # ラベルを作成し、ウィンドウに配置
        self.label = tk.Label(self, text="こんにちは, tkinter!")
        self.label.pack(pady=20)

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

主要なオプション

`Label` ウィジェットは、生成時または `config()` メソッドで多くのオプションを設定できます。

- `text`: ラベルに表示するテキスト。
- `textvariable`: `tk.StringVar` などの変数を指定し、その変数の値が変更されるとラベルのテキストも自動的に更新されます。
- `image`: 表示する画像を `tk.PhotoImage` オブジェクトで指定します。
- `font`: フォントを指定します。タプル `("フォント名", サイズ, "スタイル")` や文字列で指定できます。
- `fg` (または `foreground`): テキストの色を指定します。
- `bg` (または `background`): ラベルの背景色を指定します。
- `width`: ラベルの幅をテキスト単位で指定します。
- `height`: ラベルの高さをテキスト単位で指定します。
- `padx`, `pady`: ラベル内のテキストと境界線の間の水平・垂直方向の余白。
- `relief`: 境界線のスタイル (`flat`, `raised`, `sunken`, `groove`, `ridge`)。
- `borderwidth` (または `bd`): 境界線の幅。
- `anchor`: ラベル内でテキストを配置する位置 (`n`, `s`, `e`, `w`, `center` など)。
- `justify`: 複数行のテキストの行揃え (`left`, `center`, `right`)。
- `wraplength`: テキストがこの長さを超える場合に自動的に改行されるピクセル単位の幅。

実用的な例

テキストの動的更新

`textvariable` を使用して、ボタンクリックでラベルのテキストを更新する例です。

```
 import tkinter as tk
import random

def update_text():
    # StringVarの値を更新すると、ラベルの表示も変わる
    random_number = random.randint(1, 100)
    text_variable.set(f"ランダムな数字: {random_number}")

app = tk.Tk()
app.title("動的ラベル")
app.geometry("300x200")

# StringVarを作成
text_variable = tk.StringVar()
text_variable.set("ボタンを押してください")

label = tk.Label(
    app,
    textvariable=text_variable,
    font=("Helvetica", 14),
    pady=20
)
label.pack()

button = tk.Button(app, text="更新", command=update_text)
button.pack()

app.mainloop()
```

画像の表示

`PhotoImage` を使ってラベルに画像を表示します。 **注意**: `PhotoImage` オブジェクトは、参照が失われるとガベージコレクションによって消去されてしまうため、インスタンス変数（`label.image` など）に保持しておく必要があります。

```
 import tkinter as tk

app = tk.Tk()
app.title("画像ラベル")

# このスクリプトと同じディレクトリに 'python_logo.png' があることを想定
# tkinterが標準でサポートしているのはGIFとPGM/PPM形式です。
# PNGやJPEGなどを扱うには、Pillowライブラリ(pip install Pillow)が必要です。
try:
    # Pillowを使ったPNGの読み込み
    from PIL import Image, ImageTk
    img = Image.open("python_logo.png") # Pillowで画像を開く
    photo_image = ImageTk.PhotoImage(img) # tkinterで使える形式に変換
except (ImportError, FileNotFoundError):
    # Pillowがない、またはファイルがない場合の代替
    # (tk.PhotoImageはPNGを直接読めないことが多い)
    # 代わりに標準の画像を使うか、エラーメッセージを表示
    try:
        photo_image = tk.PhotoImage(file="python_logo.gif") # GIFの場合
    except tk.TclError:
        photo_image = None
        print("画像ファイルが見つからないか、非対応の形式です。")


if photo_image:
    label = tk.Label(app, image=photo_image)
    label.image = photo_image # 参照を保持
    label.pack(pady=20)
else:
    label = tk.Label(app, text="画像を表示できませんでした")
    label.pack(pady=20)


app.mainloop()
```

参考リンク

- Python Docs - tkinter.Label
- TkDocs - Label