Radiobutton リファレンス

複数の選択肢から一つを選択する `Radiobutton` ウィジェットについての詳細なリファレンスです。

概要

`Radiobutton` ウィジェットは、複数の選択肢の中から一つだけを選択するためのコントロールです。同じ変数を共有する複数のラジオボタンは相互排他的に動作し、一つのボタンが選択されると他のボタンは自動的に選択解除されます。

基本的な使用方法

シンプルなラジオボタン

```
import tkinter as tk

def show_selection():
    selection = selected_option.get()
    print(f"選択された項目: {selection}")

app = tk.Tk()
app.title("Radiobuttonの例")
app.geometry("300x250")

# StringVar で選択状態を管理
selected_option = tk.StringVar(value="option1")

# ラジオボタンの作成
tk.Radiobutton(
    app,
    text="オプション 1",
    variable=selected_option,
    value="option1",
    command=show_selection
).pack(anchor="w", padx=20, pady=5)

tk.Radiobutton(
    app,
    text="オプション 2",
    variable=selected_option,
    value="option2",
    command=show_selection
).pack(anchor="w", padx=20, pady=5)

tk.Radiobutton(
    app,
    text="オプション 3",
    variable=selected_option,
    value="option3",
    command=show_selection
).pack(anchor="w", padx=20, pady=5)

button = tk.Button(app, text="選択を確認", command=show_selection)
button.pack(pady=20)

app.mainloop()
```

クラスベースでのラジオボタン

```
import tkinter as tk

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

        self.title("Radiobuttonの例（クラスベース）")
        self.geometry("300x250")

        self.create_widgets()

    def create_widgets(self):
        # StringVar で選択状態を管理
        self.selected_option = tk.StringVar(value="option1")

        # ラジオボタンの作成
        self.radio1 = tk.Radiobutton(
            self,
            text="オプション 1",
            variable=self.selected_option,
            value="option1",
            command=self.show_selection
        )
        self.radio1.pack(anchor="w", padx=20, pady=5)
```

```
        self.radio2 = tk.Radiobutton(
            self,
            text="オプション 2",
            variable=self.selected_option,
            value="option2",
            command=self.show_selection
        )
        self.radio2.pack(anchor="w", padx=20, pady=5)

        self.radio3 = tk.Radiobutton(
            self,
            text="オプション 3",
            variable=self.selected_option,
            value="option3",
            command=self.show_selection
        )
        self.radio3.pack(anchor="w", padx=20, pady=5)

        self.button = tk.Button(self, text="選択を確認", command=self.show_selection)
        self.button.pack(pady=20)

    def show_selection(self):
        selection = self.selected_option.get()
        print(f"選択された項目: {selection}")

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

主要なオプション

オプション	説明
`text`	ラジオボタンに表示するテキスト。
`variable`	選択状態を管理する変数（`tk.StringVar`, `tk.IntVar` など）。
`value`	このラジオボタンが選択されたときに変数に設定される値。
`command`	ラジオボタンが選択されたときに実行される関数。
`font`	フォントを指定。タプル `("フォント名", サイズ, "スタイル")` や文字列で指定。
`fg`（または `foreground`）	テキストの色。
`bg`（または `background`）	ラジオボタンの背景色。
`activeforeground`	アクティブ時のテキスト色。
`activebackground`	アクティブ時の背景色。
`selectcolor`	ラジオボタンの色。
`state`	ラジオボタンの状態（`normal`, `active`, `disabled`）。
`anchor`	テキストの配置位置（`n`, `s`, `e`, `w`, `center` など）。
`justify`	複数行テキストの行揃え（`left`, `center`, `right`）。
`indicatoron`	ラジオボタンのインジケータを表示するかどうか（`True` / `False`）。

主要なメソッド

メソッド	説明
`select()`	このラジオボタンを選択状態にします。
`deselect()`	このラジオボタンの選択を解除します。
`invoke()`	ラジオボタンをクリックしたときと同じ動作を実行します。

実用的な例

設定選択画面

```
 import tkinter as tk
from tkinter import messagebox

class ConfigurationApp(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("設定選択")
        self.geometry("500x700")

        self.create_widgets()

    def create_widgets(self):
        # タイトル
        title_label = tk.Label(self, text="アプリケーション設定", font=("Arial", 16, "bold"))
        title_label.pack(pady=10)

        # テーマ設定
```

```
        theme_frame = tk.LabelFrame(self, text="テーマ選択", font=("Arial", 12, "bold"), padx=10, pady=10)
        theme_frame.pack(fill="x", padx=20, pady=10)

        self.theme_var = tk.StringVar(value="light")

        themes = [
            ("ライトテーマ", "light"),
            ("ダークテーマ", "dark"),
            ("ハイコントラスト", "high_contrast"),
            ("システム設定に従う", "system")
        ]

        for text, value in themes:
            tk.Radiobutton(
                theme_frame,
                text=text,
                variable=self.theme_var,
                value=value,
                command=self.on_theme_changed
            ).pack(anchor="w", pady=2)

        # 言語設定
        language_frame = tk.LabelFrame(self, text="言語選択", font=("Arial", 12, "bold"), padx=10, pady=10)
        language_frame.pack(fill="x", padx=20, pady=10)

        self.language_var = tk.StringVar(value="ja")

        languages = [
            ("日本語", "ja"),
            ("English", "en"),
            ("Español", "es"),
            ("Français", "fr"),
            ("Deutsch", "de")
        ]

        for text, value in languages:
            tk.Radiobutton(
                language_frame,
                text=text,
                variable=self.language_var,
                value=value,
                command=self.on_language_changed
            ).pack(anchor="w", pady=2)

        # ファイルサイズ設定
        filesize_frame = tk.LabelFrame(self, text="デフォルトファイルサイズ", font=("Arial", 12, "bold"), padx=10, pady=10)
        filesize_frame.pack(fill="x", padx=20, pady=10)

        self.filesize_var = tk.IntVar(value=1)

        filesizes = [
            ("小 (1MB未満)", 1),
            ("中 (1-10MB)", 2),
            ("大 (10-100MB)", 3),
            ("特大 (100MB以上)", 4)
        ]

        for text, value in filesizes:
            tk.Radiobutton(
                filesize_frame,
                text=text,
                variable=self.filesize_var,
                value=value,
                command=self.on_filesize_changed
            ).pack(anchor="w", pady=2)

        # ボタンフレーム
        button_frame = tk.Frame(self)
        button_frame.pack(pady=20)

        apply_button = tk.Button(button_frame, text="適用", command=self.apply_settings, bg="lightblue")
        apply_button.pack(side="left", padx=5)

        reset_button = tk.Button(button_frame, text="リセット", command=self.reset_settings, bg="lightcoral")
        reset_button.pack(side="left", padx=5)

        info_button = tk.Button(button_frame, text="現在の設定", command=self.show_current_settings, bg="lightgreen")
        info_button.pack(side="left", padx=5)

    def on_theme_changed(self):
        theme = self.theme_var.get()
        print(f"テーマが変更されました: {theme}")

    def on_language_changed(self):
        language = self.language_var.get()
        print(f"言語が変更されました: {language}")

    def on_filesize_changed(self):
        filesize = self.filesize_var.get()
        print(f"ファイルサイズ設定が変更されました: {filesize}")

    def apply_settings(self):
        settings = self.get_current_settings()
        messagebox.showinfo("設定適用", f"以下の設定が適用されました:\n\n{settings}")

    def reset_settings(self):
        self.theme_var.set("light")
        self.language_var.set("ja")
        self.filesize_var.set(1)
        messagebox.showinfo("リセット完了", "設定がデフォルト値にリセットされました")
```

```
    def show_current_settings(self):
        settings = self.get_current_settings()
        messagebox.showinfo("現在の設定", settings)

    def get_current_settings(self):
        theme_names = {
            "light": "ライトテーマ",
            "dark": "ダークテーマ",
            "high_contrast": "ハイコントラスト",
            "system": "システム設定に従う"
        }

        language_names = {
            "ja": "日本語",
            "en": "English",
            "es": "Español",
            "fr": "Français",
            "de": "Deutsch"
        }

        filesize_names = {
            1: "小 (1MB未満)",
            2: "中 (1-10MB)",
            3: "大 (10-100MB)",
            4: "特大 (100MB以上)"
        }

        theme = theme_names.get(self.theme_var.get(), "不明")
        language = language_names.get(self.language_var.get(), "不明")
        filesize = filesize_names.get(self.filesize_var.get(), "不明")

        return f"テーマ: {theme}\n言語: {language}\nファイルサイズ: {filesize}"

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

インジケータなしのラジオボタン（ボタン風）

```
import tkinter as tk

class ButtonStyleRadioApp(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("ボタン風ラジオボタン")
        self.geometry("400x300")

        self.create_widgets()

    def create_widgets(self):
        # タイトル
        title_label = tk.Label(self, text="難易度を選択してください", font=("Arial", 16, "bold"))
        title_label.pack(pady=20)

        # 難易度選択
        self.difficulty_var = tk.StringVar(value="normal")

        difficulty_frame = tk.Frame(self)
        difficulty_frame.pack(pady=20)

        difficulties = [
            ("初心者", "beginner", "lightgreen"),
            ("普通", "normal", "lightblue"),
            ("上級者", "advanced", "orange"),
            ("エキスパート", "expert", "lightcoral")
        ]

        self.radio_buttons = []
        for i, (text, value, color) in enumerate(difficulties):
            rb = tk.Radiobutton(
                difficulty_frame,
                text=text,
                variable=self.difficulty_var,
                value=value,
                indicatoron=False,  # インジケータを非表示にしてボタン風にする
                width=12,
                height=2,
                bg=color,
                font=("Arial", 12, "bold"),
                command=self.on_difficulty_changed
            )
            rb.grid(row=0, column=i, padx=5)
            self.radio_buttons.append(rb)

        # ゲームモード選択
        mode_label = tk.Label(self, text="ゲームモードを選択してください", font=("Arial", 14, "bold"))
        mode_label.pack(pady=(40, 10))

        self.mode_var = tk.StringVar(value="single")

        mode_frame = tk.Frame(self)
        mode_frame.pack()

        modes = [
            ("シングルプレイヤー", "single"),
            ("マルチプレイヤー", "multi"),
```

```
            ("協力プレイ", "coop"),
            ("対戦モード", "versus")
        ]

        for text, value in modes:
            tk.Radiobutton(
                mode_frame,
                text=text,
                variable=self.mode_var,
                value=value,
                font=("Arial", 11),
                command=self.on_mode_changed
            ).pack(anchor="w", pady=2)

        # 開始ボタン
        start_button = tk.Button(
            self,
            text="ゲーム開始",
            command=self.start_game,
            bg="darkgreen",
            fg="white",
            font=("Arial", 14, "bold"),
            height=2
        )
        start_button.pack(pady=30)

    def on_difficulty_changed(self):
        difficulty = self.difficulty_var.get()
        print(f"難易度が選択されました: {difficulty}")

        # 選択されたボタンの見た目を変更
        for rb in self.radio_buttons:
            if rb['value'] == difficulty:
                rb.config(relief="sunken", bd=2)
            else:
                rb.config(relief="raised", bd=1)

    def on_mode_changed(self):
        mode = self.mode_var.get()
        print(f"ゲームモードが選択されました: {mode}")

    def start_game(self):
        difficulty = self.difficulty_var.get()
        mode = self.mode_var.get()

        difficulty_names = {
            "beginner": "初心者",
            "normal": "普通",
            "advanced": "上級者",
            "expert": "エキスパート"
        }

        mode_names = {
            "single": "シングルプレイヤー",
            "multi": "マルチプレイヤー",
            "coop": "協力プレイ",
            "versus": "対戦モード"
        }

        difficulty_text = difficulty_names.get(difficulty, difficulty)
        mode_text = mode_names.get(mode, mode)

        message = f"ゲームを開始します!\n\n難易度: {difficulty_text}\nモード: {mode_text}"

        # 簡単な確認ダイアログ風の表示
        result_window = tk.Toplevel(self)
        result_window.title("ゲーム開始")
        result_window.geometry("300x150")
        result_window.transient(self)
        result_window.grab_set()

        tk.Label(result_window, text=message, font=("Arial", 12)).pack(expand=True)
        tk.Button(result_window, text="OK", command=result_window.destroy).pack(pady=10)

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

ベストプラクティス

プラクティス	説明
変数の共有	同じグループのラジオボタンは同じ変数（`tk.StringVar` など）を共有して、相互排他的な動作を実現します。
適切な値の設定	各ラジオボタンの `value` には識別しやすい一意の値を設定します。
初期値の設定	変数に初期値を設定して、デフォルトで選択されるオプションを明確にします。
グループ化	関連するラジオボタンは `Frame` や `LabelFrame` でグループ化して整理します。
カスタマイズ	`indicatoron=False` を使用してボタン風の見た目にするなど、用途に応じてカスタマイズします。

参考リンク

- Python Docs - tkinter.Radiobutton

- TkDocs - Radiobutton