

Overview

This document is part lesson and part reference document. At this point, the code that is shown cannot be used yet, **but** it can be helpful to know about **before** diving into some of the most common widgets that **tkinter** uses. Then, once you are creating an interface, this page can be helpful to refer back to for different ways to create styles in a GUI.

Important Information

Configuring A Style

In **tkinter**, styles are managed using the **ttk.Style** class. A style lets you define how a widget should look **once**, and then reuse that look across many widgets.

Styles are created and configured like this:

```
# Create a Style object
style = ttk.Style()

# Configure a new style
style.configure(
    "My.TButton",
    background="blue",
    foreground="white",
    padding=10
)
```

The name of a style is important. It usually follows this pattern:

```
CustomName.WidgetType
```

For example:

- **My.TButton**
- **Danger.TFrame**
- **Title.TLabel**

These are the widget types that you will most commonly use:

- **TFrame**
- **TLabel**
- **TButton**
- **TCheckbutton**
- **TRadiobutton**
- **TEntry**
- **TCombobox**

Using A Style on a Widget

Once a style has been configured, it can be applied to any compatible widget using the **style** option.

```
button = ttk.Button(root, text="Click Me", style="My.TButton")
button.grid()
```

If a widget uses a style, all visual configuration should be done **through the style**, not directly on the widget.

Default Style Names

Every **ttk** widget already has a default style name. When you do not specify a style, the widget automatically uses its default.

Common default style names include:

- **TFrame**
- **TLabel**
- **TButton**
- **TCheckbutton**
- **TRadiobutton**
- **TEntry**
- **TCombobox**

Custom styles are created by adding a prefix to these names, such as **My.TButton**.

Common Style Options

These options work across **most** themed widgets.

Background

Controls the background color of the widget.

```
style.configure("My.TFrame", background="lightgray")
```

Foreground

Controls the text color.

```
style.configure("My.TLabel", foreground="blue")
```

Font

Controls the font used for text.

The first parameter for **font** is the typeface name.

The second parameter for **font** is the text size in points.

```
style.configure("My.TLabel", font=("Arial", 14))
```

You can specify the font used to display the label's text using the font configuration option. Here are the names of some predefined fonts you can use:

Font	Description
TkDefaultFont	Default for all GUI items not otherwise specified.
TkTextFont	Used for entry widgets, listboxes, etc.
TkFixedFont	A standard fixed-width font.
TkMenuFont	The font used for menu items.
TkHeadingFont	A font for column headings in lists and tables.
TkCaptionFont	A font for window and dialog caption bars.
TkSmallCaptionFont	Smaller captions for subwindows or tool dialogs.
TkIconFont	A font for icon captions.
TkTooltipFont	A font for tooltips.

Padding

Adds space inside the widget.

```
style.configure("My.TButton", padding=10)
```

Advanced Padding

Padding can be defined with multiple values:

```
# Left/right = 5, top/bottom = 10
style.configure("My.TButton", padding=(5, 10))

# Left, top, right, bottom
style.configure("My.TButton", padding=(5, 7, 10, 12))
```

Relief

Controls the border appearance.

```
style.configure("My.TFrame", relief="raised")
```

Common values include:

- **flat**
- **raised**
- **sunken**
- **groove**
- **ridge**

Border Width

Controls the thickness of the border.

```
style.configure("My.TFrame", borderwidth=5)
```

Anchor

Controls where content is placed inside the widget.

```
style.configure("My.TLabel", anchor="center")
```

Common values:

- **n, s, e, w**
- **ne, nw, se, sw**
- **center**

Compound

Controls how text and images are combined.

```
style.configure("My.TButton", compound="left")
```

Common values:

- **left**
- **right**
- **top**
- **bottom**
- **center**

Uncommon Style Options (Usually Specific to a Few Types of Widgets)

Button

Dark Color

Used on some platforms to control the darker edge of a button.

```
style.configure("My.TButton", darkcolor="#444444")
```

Highlight Color

Controls highlight accents used by some themes.

```
style.configure("My.TButton", highlightcolor="yellow")
```

Checkbutton

Indicator Color

Controls the color of the checkbox indicator.

```
style.configure("My.TCheckbutton", indicatorcolor="green")
```

Indicator Margin

Controls spacing around the checkbox indicator.

```
style.configure("My.TCheckbutton", indicatormargin=5)
```

Entry

Field Background

Controls the background color of the entry field.

```
style.configure("My.TEntry", fieldbackground="lightyellow")
```

Combobox

Arrow Color

Controls the dropdown arrow color.

```
style.configure("My.TCombobox", arrowcolor="blue")
```

Focus Fill

Controls the color shown when the combobox has focus.

```
style.configure("My.TCombobox", focusfill="lightblue")
```

Field Background

Controls the background of the text area.

```
style.configure("My.TCombobox", fieldbackground="white")
```