Tkinter 8.5 reference: a GUI for Python

7. The `Button` widget

To create a pushbutton in a top-level window or frame named *parent*:

```
w = tk.Button(parent, option=value, ...)
```

The constructor returns the new `Button` widget. Its options include:

Table 5. `Button` widget options

`activebackground`	Background color when the button is under the cursor.
`activeforeground`	Foreground color when the button is under the cursor.
`anchor`	Where the text is positioned on the button. See Section 5.5, "Anchors". For example, `anchor=tk.NE` would position the text at the top right corner of the button.
`bd` or `borderwidth`	Width of the border around the outside of the button; see Section 5.1, "Dimensions". The default is two pixels.
`bg` or `background`	Normal background color.
`bitmap`	Name of one of the standard bitmaps to display on the button (instead of text).
`command`	Function or method to be called when the button is clicked.
`cursor`	Selects the cursor to be shown when the mouse is over the button.
`default`	`tk.NORMAL` is the default; use `tk.DISABLED` if the button is to be initially disabled (grayed out, unresponsive to mouse clicks).
`disabledforeground`	Foreground color used when the button is disabled.
`fg` or `foreground`	Normal foreground (text) color.
`font`	Text font to be used for the button's label.
`height`	Height of the button in text lines (for textual buttons) or pixels (for images).
`highlightbackground`	Color of the focus highlight when the widget does not have focus.
`highlightcolor`	The color of the focus highlight when the widget has focus.
`highlightthickness`	Thickness of the focus highlight.

`image`	Image to be displayed on the button (instead of text).
`justify`	How to show multiple text lines: `tk.LEFT` to left-justify each line; `tk.CENTER` to center them; or `tk.RIGHT` to right-justify.
`overrelief`	The relief style to be used while the mouse is on the button; default relief is `tk.RAISED`. See Section 5.6, "Relief styles".
`padx`	Additional padding left and right of the text. See Section 5.1, "Dimensions" for the possible values for padding.
`pady`	Additional padding above and below the text.
`relief`	Specifies the relief type for the button (see Section 5.6, "Relief styles"). The default relief is `tk.RAISED`.
`repeatdelay`	See `repeatinterval`, below.
`repeatinterval`	Normally, a button fires only once when the user releases the mouse button. If you want the button to fire at regular intervals as long as the mouse button is held down, set this option to a number of milliseconds to be used between repeats, and set the `repeatdelay` to the number of milliseconds to wait before starting to repeat. For example, if you specify "`repeatdelay=500, repeatinterval=100`" the button will fire after half a second, and every tenth of a second thereafter, until the user releases the mouse button. If the user does not hold the mouse button down at least `repeatdelay` milliseconds, the button will fire normally.
`state`	Set this option to `tk.DISABLED` to gray out the button and make it unresponsive. Has the value `tk.ACTIVE` when the mouse is over it. Default is `tk.NORMAL`.
`takefocus`	Normally, keyboard focus does visit buttons (see Section 53, "Focus: routing keyboard input"), and a space character acts as the same as a mouse click, "pushing" the button. You can set the `takefocus` option to zero to prevent focus from visiting the button.
`text`	Text displayed on the button. Use internal newlines to display multiple text lines.
`textvariable`	An instance of `StringVar()` that is associated with the text on this button. If the variable is changed, the new value will be displayed on the button. See Section 52, "Control variables: the values behind the widgets".
`underline`	Default is `-1`, meaning that no character of the text on the button will be underlined. If nonnegative, the corresponding text character will be underlined. For example, `underline=1` would underline the second character of the button's text.
`width`	Width of the button in letters (if displaying text) or pixels (if displaying an image).
`wraplength`	If this value is set to a positive number, the text lines will be wrapped to fit within this length. For possible values, see Section 5.1, "Dimensions".

Methods on `Button` objects:

`.flash()`

> Causes the button to flash several times between active and normal colors. Leaves the button in
> the state it was in originally. Ignored if the button is disabled.

`.invoke()`

> Calls the button's `command` callback, and returns what that function returns. Has no effect if the
> button is disabled or there is no callback.

Next: 8. The `Canvas` widget
Contents: *Tkinter* 8.5 reference: a GUI for Python
Previous: 6. Exception handling
Help: Tech Computer Center: Help System
Home: About New Mexico Tech

John W. Shipman
Comments welcome: tcc-doc@nmt.edu
Last updated: 2013-12-31 17:59
URL: `http://www.nmt.edu/tcc/help/pubs/tkinter/web/button.html`