

Python - Tkinter Menubutton

A menubutton is the part of a drop-down menu that stays on the screen all the time. Every menubutton is associated with a Menu widget that can display the choices for that menubutton when the user clicks on it.

Syntax

Here is the simple syntax to create this widget –

```
w = Menubutton ( master, option, ... )
```

Parameters

- **master** – This represents the parent window.
- **options** – Here is the list of most commonly used options for this widget. These options can be used as key-value pairs separated by commas.

Sr.No.	Option & Description
1	activebackground The background color when the mouse is over the menubutton.
2	activeforeground The foreground color when the mouse is over the menubutton.
3	anchor This options controls where the text is positioned if the widget has more space than the text needs. The default is anchor=CENTER, which centers the text.
4	bg The normal background color displayed behind the label and indicator.
5	bitmap To display a bitmap on the menubutton, set this option to a bitmap name.
6	bd The size of the border around the indicator. Default is 2 pixels.
7	cursor The cursor that appears when the mouse is over this menubutton.
8	direction Set direction=LEFT to display the menu to the left of the button; use direction=RIGHT to display the menu to the right of the button; or use direction='above' to place the menu above the button.
9	disabledforeground The foreground color shown on this menubutton when it is disabled.
10	fg

	The foreground color when the mouse is not over the menubutton.
11	height The height of the menubutton in lines of text (not pixels!). The default is to fit the menubutton's size to its contents.
12	highlightcolor Color shown in the focus highlight when the widget has the focus.
13	image To display an image on this menubutton,
14	justify This option controls where the text is located when the text doesn't fill the menubutton: use justify=LEFT to left-justify the text (this is the default); use justify=CENTER to center it, or justify=RIGHT to right-justify.
15	menu To associate the menubutton with a set of choices, set this option to the Menu object containing those choices. That menu object must have been created by passing the associated menubutton to the constructor as its first argument.
16	padx How much space to leave to the left and right of the text of the menubutton. Default is 1.
17	pady How much space to leave above and below the text of the menubutton. Default is 1.
18	relief Selects three-dimensional border shading effects. The default is RAISED.
19	state

	Normally, menubuttons respond to the mouse. Set state=DISABLED to gray out the menubutton and make it unresponsive.
20	text To display text on the menubutton, set this option to the string containing the desired text. Newlines ("\n") within the string will cause line breaks.
21	textvariable You can associate a control variable of class StringVar with this menubutton. Setting that control variable will change the displayed text.
22	underline Normally, no underline appears under the text on the menubutton. To underline one of the characters, set this option to the index of that character.
23	width The width of the widget in characters. The default is 20.
24	wraplength Normally, lines are not wrapped. You can set this option to a number of characters and all lines will be broken into pieces no longer than that number.

Example

Try the following example yourself –

```
from Tkinter import *
import tkMessageBox
import Tkinter

top = Tk()

mb= Menubutton ( top, text="condiments", relief=RAISED )
mb.grid()
mb.menu = Menu ( mb, tearoff = 0 )
mb["menu"] = mb.menu

mayoVar = IntVar()
```

```
ketchVar = IntVar()

mb.menu.add_checkbutton ( label="mayo",
                           variable=mayoVar )
mb.menu.add_checkbutton ( label="ketchup",
                           variable=ketchVar )

mb.pack()
top.mainloop()
```

When the above code is executed, it produces the following result –

