Python - Tkinter Checkbutton

The Checkbutton widget is used to display a number of options to a user as toggle buttons. The user can then select one or more options by clicking the button corresponding to each option.

You can also display images in place of text.

Syntax

Here is the simple syntax to create this widget −

```
w = Checkbutton ( 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** Background color when the checkbutton is under the cursor.
2	**activeforeground** Foreground color when the checkbutton is under the cursor.
3	**bg** The normal background color displayed behind the label and indicator.
4	**bitmap** To display a monochrome image on a button.
5	**bd** The size of the border around the indicator. Default is 2 pixels.
6	**command** A procedure to be called every time the user changes the state of this checkbutton.
7	**cursor** If you set this option to a cursor name (*arrow, dot etc.*), the mouse cursor will change to that pattern when it is over the checkbutton.
8	**disabledforeground** The foreground color used to render the text of a disabled checkbutton. The default is a stippled version of the default foreground color.
9	**font** The font used for the text.
10	**fg** The color used to render the text.

11	**height** The number of lines of text on the checkbutton. Default is 1.
12	**highlightcolor** The color of the focus highlight when the checkbutton has the focus.
13	**image** To display a graphic image on the button.
14	**justify** If the text contains multiple lines, this option controls how the text is justified: CENTER, LEFT, or RIGHT.
15	**offvalue** Normally, a checkbutton's associated control variable will be set to 0 when it is cleared (off). You can supply an alternate value for the off state by setting offvalue to that value.
16	**onvalue** Normally, a checkbutton's associated control variable will be set to 1 when it is set (on). You can supply an alternate value for the on state by setting onvalue to that value.
17	**padx** How much space to leave to the left and right of the checkbutton and text. Default is 1 pixel.
18	**pady** How much space to leave above and below the checkbutton and text. Default is 1 pixel.
19	**relief** With the default value, relief=FLAT, the checkbutton does not stand out from its background. You may set this option to any of the other styles

20	**selectcolor** The color of the checkbutton when it is set. Default is selectcolor="red".
21	**selectimage** If you set this option to an image, that image will appear in the checkbutton when it is set.
22	**state** The default is state=NORMAL, but you can use state=DISABLED to gray out the control and make it unresponsive. If the cursor is currently over the checkbutton, the state is ACTIVE.
23	**text** The label displayed next to the checkbutton. Use newlines ("\n") to display multiple lines of text.
24	**underline** With the default value of -1, none of the characters of the text label are underlined. Set this option to the index of a character in the text (counting from zero) to underline that character.
25	**variable** The control variable that tracks the current state of the checkbutton. Normally this variable is an *IntVar*, and 0 means cleared and 1 means set, but see the offvalue and onvalue options above.
26	**width** The default width of a checkbutton is determined by the size of the displayed image or text. You can set this option to a number of characters and the checkbutton will always have room for that many characters.
27	**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.

Methods

Following are commonly used methods for this widget −

Sr.No.	Method & Description
1	**deselect()** Clears (turns off) the checkbutton.
2	**flash()** Flashes the checkbutton a few times between its active and normal colors, but leaves it the way it started.
3	**invoke()** You can call this method to get the same actions that would occur if the user clicked on the checkbutton to change its state.
4	**select()** Sets (turns on) the checkbutton.
5	**toggle()** Clears the checkbutton if set, sets it if cleared.

Example

Try the following example yourself −

```python
from Tkinter import *
import tkMessageBox
import Tkinter

top = Tkinter.Tk()
CheckVar1 = IntVar()
CheckVar2 = IntVar()
C1 = Checkbutton(top, text = "Music", variable = CheckVar1, \
                 onvalue = 1, offvalue = 0, height=5, \
                 width = 20)
C2 = Checkbutton(top, text = "Video", variable = CheckVar2, \
```

```
                   onvalue = 1, offvalue = 0, height=5, \
                   width = 20)
C1.pack()
C2.pack()
top.mainloop()
```

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