Python - Tkinter Label

This widget implements a display box where you can place text or images. The text displayed by this widget can be updated at any time you want.

It is also possible to underline part of the text (like to identify a keyboard shortcut) and span the text across multiple lines.

Syntax

Here is the simple syntax to create this widget −

```
w = Label ( 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	**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 in the available space.
2	**bg** The normal background color displayed behind the label and indicator.
3	**bitmap** Set this option equal to a bitmap or image object and the label will display that graphic.
4	**bd** The size of the border around the indicator. Default is 2 pixels.
5	**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.
6	**font** If you are displaying text in this label (with the text or textvariable option, the font option specifies in what font that text will be displayed.
7	**fg** If you are displaying text or a bitmap in this label, this option specifies the color of the text. If you are displaying a bitmap, this is the color that will appear at the position of the 1-bits in the bitmap.
8	**height** The vertical dimension of the new frame.
9	**image** To display a static image in the label widget, set this option to an image object.
10	**justify** Specifies how multiple lines of text will be aligned with respect to each other: LEFT for flush left, CENTER for centered (the default), or RIGHT for right-justified.
11	**padx** Extra space added to the left and right of the text within the widget. Default is 1.
12	**pady** Extra space added above and below the text within the widget. Default is 1.

13	**relief** Specifies the appearance of a decorative border around the label. The default is FLAT; for other values.
14	**text** To display one or more lines of text in a label widget, set this option to a string containing the text. Internal newlines ("\n") will force a line break.
15	**textvariable** To slave the text displayed in a label widget to a control variable of class *StringVar*, set this option to that variable.
16	**underline** You can display an underline (_) below the nth letter of the text, counting from 0, by setting this option to n. The default is underline=-1, which means no underlining.
17	**width** Width of the label in characters (not pixels!). If this option is not set, the label will be sized to fit its contents.
18	**wraplength** You can limit the number of characters in each line by setting this option to the desired number. The default value, 0, means that lines will be broken only at newlines.

Example

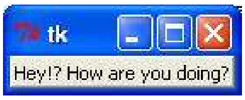
Try the following example yourself −

```
from Tkinter import *

root = Tk()
var = StringVar()
label = Label( root, textvariable=var, relief=RAISED )

var.set("Hey!? How are you doing?")
label.pack()
root.mainloop()
```

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



Hey!? How are you doing?