

Python - Tkinter Entry

The Entry widget is used to accept single-line text strings from a user.

- If you want to display multiple lines of text that can be edited, then you should use the *Text* widget.
- If you want to display one or more lines of text that cannot be modified by the user, then you should use the *Label* widget.

Syntax

Here is the simple syntax to create this widget –

```
w = Entry( 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	bg The normal background color displayed behind the label and indicator.
2	bd The size of the border around the indicator. Default is 2 pixels.
3	command A procedure to be called every time the user changes the state of this checkbutton.
4	cursor If you set this option to a cursor name (<i>arrow</i> , <i>dot etc.</i>), the mouse cursor will change to that pattern when it is over the checkbutton.
5	font The font used for the text.
6	exportselection By default, if you select text within an Entry widget, it is automatically exported to the clipboard. To avoid this exportation, use exportselection=0.
7	fg The color used to render the text.
8	highlightcolor The color of the focus highlight when the checkbutton has the focus.
9	justify If the text contains multiple lines, this option controls how the text is justified: CENTER, LEFT, or RIGHT.
10	relief

	<p>With the default value, <code>relief=FLAT</code>, the <code>checkboxbutton</code> does not stand out from its background. You may set this option to any of the other styles</p>
11	<p><code>selectbackground</code></p> <p>The background color to use displaying selected text.</p>
12	<p><code>selectborderwidth</code></p> <p>The width of the border to use around selected text. The default is one pixel.</p>
13	<p><code>selectforeground</code></p> <p>The foreground (text) color of selected text.</p>
14	<p><code>show</code></p> <p>Normally, the characters that the user types appear in the entry. To make a <code>.password</code> entry that echoes each character as an asterisk, set <code>show=""</code>.</p>
15	<p><code>state</code></p> <p>The default is <code>state=NORMAL</code>, but you can use <code>state=DISABLED</code> to gray out the control and make it unresponsive. If the cursor is currently over the <code>checkboxbutton</code>, the state is <code>ACTIVE</code>.</p>
16	<p><code>textvariable</code></p> <p>In order to be able to retrieve the current text from your entry widget, you must set this option to an instance of the <code>StringVar</code> class.</p>
17	<p><code>width</code></p> <p>The default width of a <code>checkboxbutton</code> is determined by the size of the displayed image or text. You can set this option to a number of characters and the <code>checkboxbutton</code> will always have room for that many characters.</p>
18	<p><code>xscrollcommand</code></p> <p>If you expect that users will often enter more text than the onscreen size of the widget, you can link your entry widget to a scrollbar.</p>

Methods

Following are commonly used methods for this widget –

Sr.No.	Method & Description
1	<p>delete (first, last=None)</p> <p>Deletes characters from the widget, starting with the one at index first, up to but not including the character at position last. If the second argument is omitted, only the single character at position first is deleted.</p>
2	<p>get()</p> <p>Returns the entry's current text as a string.</p>
3	<p>icursor (index)</p> <p>Set the insertion cursor just before the character at the given index.</p>
4	<p>index (index)</p> <p>Shift the contents of the entry so that the character at the given index is the leftmost visible character. Has no effect if the text fits entirely within the entry.</p>
5	<p>insert (index, s)</p> <p>Inserts string s before the character at the given index.</p>
6	<p>select_adjust (index)</p> <p>This method is used to make sure that the selection includes the character at the specified index.</p>
7	<p>select_clear()</p> <p>Clears the selection. If there isn't currently a selection, has no effect.</p>
8	<p>select_from (index)</p> <p>Sets the ANCHOR index position to the character selected by index, and selects that character.</p>
9	<p>select_present()</p> <p>If there is a selection, returns true, else returns false.</p>

10	select_range (start, end) Sets the selection under program control. Selects the text starting at the start index, up to but not including the character at the end index. The start position must be before the end position.
11	select_to (index) Selects all the text from the ANCHOR position up to but not including the character at the given index.
12	xview (index) This method is useful in linking the Entry widget to a horizontal scrollbar.
13	xview_scroll (number, what) Used to scroll the entry horizontally. The what argument must be either UNITS, to scroll by character widths, or PAGES, to scroll by chunks the size of the entry widget. The number is positive to scroll left to right, negative to scroll right to left.

Example

Try the following example yourself –

```
from Tkinter import *  
  
top = Tk()  
L1 = Label(top, text="User Name")  
L1.pack( side = LEFT)  
E1 = Entry(top, bd =5)  
E1.pack(side = RIGHT)  
  
top.mainloop()
```

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

