**Tkinter Text**

The Text widget is used to show the text data on the Python application. However, Tkinter provides us the Entry widget which is used to implement the single line text box. The Text widget is used to display the multi-line formatted text with various styles and attributes. The Text widget is mostly used to provide the text editor to the user. The Text widget also facilitates us to use the marks and tabs to locate the specific sections of the Text. We can also use the windows and images with the Text as it can also be used to display the formatted text.

SYNTAX

**w = Text(top, options)**

A list of possible options that can be used with the Text widget is given below.

|  |  |  |
| --- | --- | --- |
| **SN** | **Option** | **Description** |
| 1 | bg | The background color of the widget. |
| 2 | bd | It represents the border width of the widget. |
| 3 | cursor | The mouse pointer is changed to the specified cursor type, i.e. arrow, dot, etc. |
| 4 | exportselection | The selected text is exported to the selection in the window manager. We can set this to 0 if we don't want the text to be exported. |
| 5 | font | The font type of the text. |
| 6 | fg | The text color of the widget. |
| 7 | height | The vertical dimension of the widget in lines. |
| 8 | highlightbackground | The highlightcolor when the widget doesn't has the focus. |
| 9 | highlightthickness | The thickness of the focus highlight. The default value is 1. |
| 10 | highlighcolor | The color of the focus highlight when the widget has the focus. |
| 11 | insertbackground | It represents the color of the insertion cursor. |
| 12 | insertborderwidth | It represents the width of the border around the cursor. The default is 0. |
| 13 | insertofftime | The time amount in Milliseconds during which the insertion cursor is off in the blink cycle. |
| 14 | insertontime | The time amount in Milliseconds during which the insertion cursor is on in the blink cycle. |
| 15 | insertwidth | It represents the width of the insertion cursor. |
| 16 | padx | The horizontal padding of the widget. |
| 17 | pady | The vertical padding of the widget. |
| 18 | relief | The type of the border. The default is SUNKEN. |
| 19 | selectbackground | The background color of the selected text. |
| 20 | selectborderwidth | The width of the border around the selected text. |
| 21 | spacing1 | It specifies the amount of vertical space given above each line of the text. The default is 0. |
| 22 | spacing2 | This option specifies how much extra vertical space to add between displayed lines of text when a logical line wraps. The default is 0. |
| 23 | spacing3 | It specifies the amount of vertical space to insert below each line of the text. |
| 24 | state | It the state is set to DISABLED, the widget becomes unresponsive to the mouse and keyboard unresponsive. |
| 25 | tabs | This option controls how the tab character is used to position the text. |
| 26 | width | It represents the width of the widget in characters. |
| 27 | wrap | This option is used to wrap the wider lines into multiple lines. Set this option to the WORD to wrap the lines after the word that fit into the available space. The default value is CHAR which breaks the line which gets too wider at any character. |
| 28 | xscrollcommand | To make the Text widget horizontally scrollable, we can set this option to the set() method of Scrollbar widget. |
| 29 | yscrollcommand | To make the Text widget vertically scrollable, we can set this option to the set() method of Scrollbar widget. |

## **Methods**

We can use the following methods with the Text widget.

|  |  |  |
| --- | --- | --- |
| **SN** | **Method** | **Description** |
| 1 | delete(startindex, endindex) | This method is used to delete the characters of the specified range. |
| 2 | get(startindex, endindex) | It returns the characters present in the specified range. |
| 3 | index(index) | It is used to get the absolute index of the specified index. |
| 4 | insert(index, string) | It is used to insert the specified string at the given index. |
| 5 | see(index) | It returns a boolean value true or false depending upon whether the text at the specified index is visible or not. |

## **Mark handling methods**

Marks are used to bookmark the specified position between the characters of the associated text.

|  |  |  |
| --- | --- | --- |
| **SN** | **Method** | **Description** |
| 1 | index(mark) | It is used to get the index of the specified mark. |
| 2 | mark\_gravity(mark, gravity) | It is used to get the gravity of the given mark. |
| 3 | mark\_names() | It is used to get all the marks present in the Text widget. |
| 4 | mark\_set(mark, index) | It is used to inform a new position of the given mark. |
| 5 | mark\_unset(mark) | It is used to remove the given mark from the text. |

## **Tag handling methods**

The tags are the names given to the separate areas of the text. The tags are used to configure the different areas of the text separately. The list of tag-handling methods along with the description is given below.

|  |  |  |
| --- | --- | --- |
| **SN** | **Method** | **Description** |
| 1 | tag\_add(tagname, startindex, endindex) | This method is used to tag the string present in the specified range. |
| 2 | tag\_config | This method is used to configure the tag properties. |
| 3 | tag\_delete(tagname) | This method is used to delete a given tag. |
| 4 | tag\_remove(tagname, startindex, endindex) | This method is used to remove a tag from the specified range. |

# Tkinter Toplevel

The Toplevel widget is used to create and display the toplevel windows which are directly managed by the window manager. The toplevel widget may or may not have the parent window on the top of them. The toplevel widget is used when a python application needs to represent some extra information, pop-up, or the group of widgets on the new window. The toplevel windows have the title bars, borders, and other window decorations.

The syntax to use the Toplevel widget is given below.

**w = Toplevel(options)**

A List of possible options is given below.

|  |  |  |
| --- | --- | --- |
| **SN** | **Options** | **Description** |
| 1 | bg | It represents the background color of the window. |
| 2 | bd | It represents the border size of the window. |
| 3 | cursor | The mouse pointer is changed to the cursor type set to the arrow, dot, etc. when the mouse is in the window. |
| 4 | class\_ | The text selected in the text widget is exported to be selected to the window manager. We can set this to 0 to make this behavior false. |
| 5 | font | The font type of the text inserted into the widget. |
| 6 | fg | The foreground color of the widget. |
| 7 | height | It represents the height of the window. |
| 8 | relief | It represents the type of the window. |
| 9 | width | It represents the width of the window, |

## **Methods**

The methods associated with the Toplevel widget is given in the following list.

|  |  |  |
| --- | --- | --- |
| **SN** | **Method** | **Description** |
| 1 | deiconify() | This method is used to display the window. |
| 2 | frame() | It is used to show a system dependent window identifier. |
| 3 | group(window) | It is used to add this window to the specified window group. |
| 4 | iconify() | It is used to convert the toplevel window into an icon. |
| 5 | protocol(name, function) | It is used to mention a function which will be called for the specific protocol. |
| 6 | state() | It is used to get the current state of the window. Possible values are normal, iconic, withdrawn, and icon. |
| 7 | transient([master]) | It is used to convert this window to a transient window (temporary). |
| 8 | withdraw() | It is used to delete the window but doesn't destroy it. |
| 9 | maxsize(width, height) | It is used to declare the maximum size for the window. |
| 10 | minsize(width, height) | It is used to declare the minimum size for the window. |
| 11 | positionfrom(who) | It is used to define the position controller. |
| 12 | resizable(width, height) | It is used to control whether the window can be resizable or not. |
| 13 | sizefrom(who) | It is used to define the size controller. |
| 14 | title(string) | It is used to define the title for the window. |

# Python Tkinter Spinbox

The Spinbox widget is an alternative to the Entry widget. It provides the range of values to the user, out of which, the user can select the one. It is used in the case where a user is given some fixed number of values to choose from. We can use various options with the Spinbox to decorate the widget. The syntax to use the Spinbox is given below.

**w = Spinbox(top, options)**

A list of possible options is given below.

|  |  |  |
| --- | --- | --- |
| **SN** | **Option** | **Description** |
| 1 | activebackground | The background color of the widget when it has the focus. |
| 2 | bg | The background color of the widget. |
| 3 | bd | The border width of the widget. |
| 4 | command | The associated callback with the widget which is called each time the state of the widget is called. |
| 5 | cursor | The mouse pointer is changed to the cursor type assigned to this option. |
| 6 | disabledbackground | The background color of the widget when it is disabled. |
| 7 | disabledforeground | The foreground color of the widget when it is disabled. |
| 8 | fg | The normal foreground color of the widget. |
| 9 | font | The font type of the widget content. |
| 10 | format | This option is used for the format string. It has no default value. |
| 11 | from\_ | It is used to show the starting range of the widget. |
| 12 | justify | It is used to specify the justification of the multi-line widget content. The default is LEFT. |
| 13 | relief | It is used to specify the type of the border. The default is SUNKEN. |
| 14 | repeatdelay | This option is used to control the button auto repeat. The value is given in milliseconds. |
| 15 | repeatinterval | It is similar to repeatdelay. The value is given in milliseconds. |
| 16 | state | It represents the state of the widget. The default is NORMAL. The possible values are NORMAL, DISABLED, or "readonly". |
| 17 | textvariable | It is like a control variable which is used to control the behaviour of the widget text. |
| 18 | to | It specify the maximum limit of the widget value. The other is specified by the from\_ option. |
| 19 | validate | This option controls how the widget value is validated. |
| 20 | validatecommand | It is associated to the function callback which is used for the validation of the widget content. |
| 21 | values | It represents the tuple containing the values for this widget. |
| 22 | vcmd | It is same as validation command. |
| 23 | width | It represents the width of the widget. |
| 24 | wrap | This option wraps up the up and down button the Spinbox. |
| 25 | xscrollcommand | This options is set to the set() method of scrollbar to make this widget horizontally scrollable. |

## **Methods**

There are the following methods associated with the widget.

|  |  |  |
| --- | --- | --- |
| **SN** | **Option** | **Description** |
| 1 | delete(startindex, endindex) | This method is used to delete the characters present at the specified range. |
| 2 | get(startindex, endindex) | It is used to get the characters present in the specified range. |
| 3 | identify(x, y) | It is used to identify the widget's element within the specified range. |
| 4 | index(index) | It is used to get the absolute value of the given index. |
| 5 | insert(index, string) | This method is used to insert the string at the specified index. |
| 6 | invoke(element) | It is used to invoke the callback associated with the widget. |