Unit 4- GUI with Python

GUI Programming Toolkits for Python

Python provides various options for developing graphical user interfaces (GUIs). Most important are listed below.

Tkinter − Tkinter is the Python interface to the Tk GUI toolkit shipped with Python.

wxPython − This is an open-source Python interface for wxWindows http://wxpython.org.

JPython − JPython is a Python port for Java which gives Python scripts seamless access to Java class libraries on the local machine http://www.jython.org.

Tkinter Programming

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

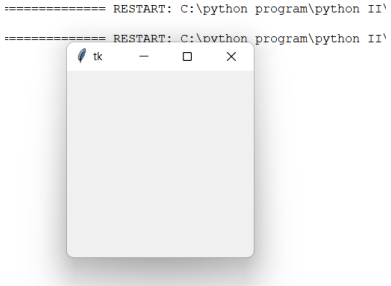
Creating a GUI application perform following steps

1. Import the Tkinter module.
2. Create the GUI application main window.
3. Add one or more of the above-mentioned widgets to the GUI application.
4. Enter the main event loop to take action against each event triggered by the user

Example

```
from tkinter import *
top =Tk()
# Code to add widgets will go here...
top.mainloop()
```

Output:



Creating GUI Widgets with Tkinter

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

Sr. No	Widget	Description
1	Button	The Button is used to add various kinds of buttons to the python application.
2	Canvas	The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.
3	Checkbutton	The Checkbutton is used to display the CheckButton on the window.
4	Entry	The entry widget is used to display the single-line text field to the user. It is commonly used to accept user values.
5	Frame	It can be defined as a container to which, another widget can be added and organized.
6	Label	A label is a text used to display some message or information about the other widgets.
7	ListBox	The ListBox widget is used to display a list of options to the user.
8	Menubutton	The Menubutton is used to display the menu items to the user.
9	Menu	It is used to add menu items to the user.
10	Message	The Message widget is used to display the message-box to the user.
11	Radiobutton	The Radiobutton is different from a checkbutton. Here, the user is provided with various options and the user can select only one option among them.
12	Scale	It is used to provide the slider to the user.
13	Scrollbar	It provides the scrollbar to the user so that the user can scroll the window up and down.
14	Text	It is different from Entry because it provides a multi-line text field to the user so that the user can write the text and edit the text inside it.
14	Toplevel	It is used to create a separate window container.
15	Spinbox	It is an entry widget used to select from options of values.
16	PanedWindow	It is like a container widget that contains horizontal or vertical panes.
17	LabelFrame	A LabelFrame is a container widget that acts as the container
18	MessageBox	This module is used to display the message-box in the desktop based applications.

Resizing the Widget

The Frames widget in tkinter is generally used to display widgets in the form of a container. The Frame widget works similar to the default window container. The

geometry and size of the frame widget can be configured using various geometry managers available in the tkinter library.

Python Tkinter Geometry

The geometry method is a fundamental one which decides the size, position and some other attributes of the screen layout we are going to create.There are three methods

1. The pack() method
2. The grid() method
3. The place() method

1) Python Tkinter pack() method

The pack() widget is used to organize widget in the block. The positions widgets added to the python application using the pack() method can be controlled by using the various options specified in the method call. However, the controls are less and widgets are generally added in the less organized manner.

Syntax-

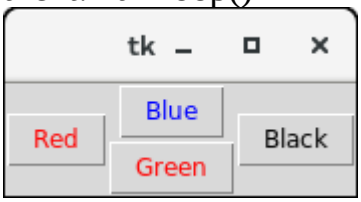
widget.pack(options)

options

1. **expand:** If the expand is set to true, the widget expands to fill any space.
2. **Fill:** By default, the fill is set to NONE. However, we can set it to X or Y to determine whether the widget contains any extra space.
3. **size:** it represents the side of the parent to which the widget is to be placed on the window

   Example-

```
from tkinter import *
parent = Tk()
redbutton = Button(parent, text = "Red", fg = "red")
redbutton.pack( side = LEFT)
greenbutton = Button(parent, text = "Black", fg = "black")
greenbutton.pack( side = RIGHT )
bluebutton = Button(parent, text = "Blue", fg = "blue")
bluebutton.pack( side = TOP )
blackbutton = Button(parent, text = "Green", fg = "red")
blackbutton.pack( side = BOTTOM)
parent.mainloop()
```



2) The grid() method

The grid() geometry manager organizes the widgets in the tabular form. We can specify the rows and columns as the options in the method call. We can also specify the column span (width) or rowspan(height) of a widget.

Syntax:

widget.grid(options)

options

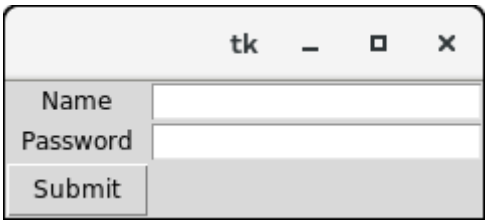
1. **column**
   The column number in which the widget is to be placed. The leftmost column is represented by 0.
2. **Columnspan**

The width of the widget. It represents the number of columns up to which, the column is expanded.

3. **ipadx, ipady**
   It represents the number of pixels to pad the widget inside the widget's border.
4. **padx, pady**
   It represents the number of pixels to pad the widget outside the widget's border.
5. **row**
   The row number in which the widget is to be placed. The topmost row is represented by 0.
6. **rowspan**
   The height of the widget, i.e. the number of the row up to which the widget is expanded.
7. **Sticky**
   If the cell is larger than a widget, then sticky is used to specify the position of the widget inside the cell. It may be the concatenation of the sticky letters representing the position of the widget. It may be N, E, W, S, NE, NW, NS, EW, ES.

   Example

```
from tkinter import *
parent = Tk()
name = Label(parent,text = "Name").grid(row = 0, column = 0)
e1 = Entry(parent).grid(row = 0, column = 1)
password = Label(parent,text = "Password").grid(row = 1, column = 0)
e2 = Entry(parent).grid(row = 1, column = 1)
submit = Button(parent, text = "Submit").grid(row = 4, column = 0)
parent.mainloop()
```



3) The place() method

The place() geometry manager organizes the widgets to the specific x and y coordinates.

Syntax-

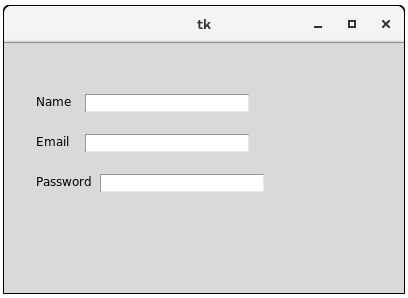
widget.place(options)

Option-

1. **Anchor**: It represents the exact position of the widget within the container. The default value (direction) is NW (the upper left corner)
2. **bordermode:** The default value of the border type is INSIDE that refers to ignore the parent's inside the border. The other option is OUTSIDE.
3. **height, width**: It refers to the height and width in pixels.
4. **relheight**, **relwidth**: It is represented as the float between 0.0 and 1.0 indicating the fraction of the parent's height and width.
5. **relx, rely**: It is represented as the float between 0.0 and 1.0 that is the offset in the horizontal and vertical direction.
6. **x, y:** It refers to the horizontal and vertical offset in the pixels.

Example-

```
from tkinter import *
top = Tk()
top.geometry("400x250")
name = Label(top, text = "Name").place(x = 30,y = 50)
email = Label(top, text = "Email").place(x = 30, y = 90)
password = Label(top, text = "Password").place(x = 30, y = 130)
e1 = Entry(top).place(x = 80, y = 50)
e2 = Entry(top).place(x = 80, y = 90)
e3 = Entry(top).place(x = 95, y = 130)
top.mainloop()
```



Widges-

1) Python Tkinter Button

The button widget is used to add various types of buttons to the python application. Python allows us to configure the look of the button according to our requirements.

Syntax-

W = Button(parent, options)

Option-

SN	Option	Description
1	activebackground	It represents the background of the button when the mouse hover the button.
2	activeforeground	It represents the font color of the button when the mouse hover the button.
3	Bd	It represents the border width in pixels.
4	Bg	It represents the background color of the button.
5	Command	It is set to the function call which is scheduled when the function is called.

6	Fg	Foreground color of the button.
7	Font	The font of the button text.
8	Height	The height of the button. The height is represented in the number of text lines for the textual lines or the number of pixels for the images.
10	Highlightcolor	The color of the highlight when the button has the focus.
11	Image	It is set to the image displayed on the button.
12	Justify	It illustrates the way by which the multiple text lines are represented. It is set to LEFT for left justification, RIGHT for the right justification, and CENTER for the center.
13	Padx	Additional padding to the button in the horizontal direction.
14	Pady	Additional padding to the button in the vertical direction.
15	Relief	It represents the type of the border. It can be SUNKEN, RAISED, GROOVE, and RIDGE.
17	State	This option is set to DISABLED to make the button unresponsive. The ACTIVE represents the active state of the button.
18	Underline	Set this option to make the button text underlined.
19	Width	The width of the button. It exists as a number of letters for textual buttons or pixels for image buttons.
20	Wraplength	If the value is set to a positive number, the text lines will be wrapped to fit within this length.

Example-

```
#python application to create a simple button
from tkinter import *

top = Tk()

top.geometry("200x100")

b = Button(top,text = "Simple")

b.pack()

top.mainloop()
```

2) Python Tkinter Canvas

The canvas widget is used to add the structured graphics to the python application. It is used to draw the graph and plots to the python application

Syntax-

w = canvas(parent, options)

Options

SN	Option	Description
1	Bd	The represents the border width. The default width is 2.
2	Bg	It represents the background color of the canvas.
3	Confine	It is set to make the canvas unscrollable outside the scroll region.
4	Cursor	The cursor is used as the arrow, circle, dot, etc. on the canvas.
5	Height	It represents the size of the canvas in the vertical direction.
6	highlightcolor	It represents the highlight color when the widget is focused.
7	Relief	It represents the type of the border. The possible values are SUNKEN, RAISED, GROOVE, and RIDGE.
8	scrollregion	It represents the coordinates specified as the tuple containing the area of the canvas.
9	Width	It represents the width of the canvas.
10	xscrollincrement	If it is set to a positive value. The canvas is placed only to the multiple of this value.
11	xscrollcommand	If the canvas is scrollable, this attribute should be the .set() method of the horizontal scrollbar.
12	yscrollincrement	Works like xscrollincrement, but governs vertical movement.
13	yscrollcommand	If the canvas is scrollable, this attribute should be the .set() method of the vertical scrollbar.

A canvas object has several add_* methods. These methods allow you to place items on it

Item	Method
Line	create_line()
Rectangle	create_rectangle()
Oval	create_oval()
Arc	create_arc()
Polygon	create_polygon()
Text	create_text(()
Image	create_image()

Example-

1) creating a simple canvas

```
c = Canvas(top,bg = "pink",height = "200",width = 200)

arc = c.create_arc((5,10,150,200),start = 0,extent = 150, fill= "white")

c.pack()
```

2) Creating a line

```
canvas.create_line((50, 50), (100, 100), width=4, fill='red')
```

3) Creating a rectangle

canvas = tk.Canvas(root, width=600, height=400, bg='white')

4) Create an image

To place an image on a canvas, you use the create_image()

```
from tkinter import *
root = Tk()
canvas = Canvas(root, width = 300, height = 300)
canvas.pack()
img = PhotoImage(file='download1.png')
canvas.create_image(20,20, anchor=NW, image=img)
mainloop()
```

3) Python Tkinter Check button

The Checkbutton is used to track the user's choices provided to the application. In other words, Checkbutton is used to implement the on/off selections.

Syntax-

w = checkbutton(top, options)

Options

SN	Option	Description
1	activebackground	It represents the background color when the checkbutton is under the cursor.
2	activeforeground	It represents the foreground color of the checkbutton when the checkbutton is under the cursor.
3	Bg	The background color of the button.
4	bitmap	It displays an image (monochrome) on the button.
5	Bd	The size of the border around the corner.
6	command	It is associated with a function to be called when the state of the checkbutton is changed.
7	Cursor	The mouse pointer will be changed to the cursor name when it is over the checkbutton.
8	disableforeground	It is the color which is used to represent the text of a disabled checkbutton.
9	Font	It represents the font of the checkbutton.
10	Fg	The foreground color (text color) of the checkbutton.
11	height	It represents the height of the checkbutton (number of lines). The default height is 1.
12	highlightcolor	The color of the focus highlight when the checkbutton is under focus.
13	Image	The image used to represent the checkbutton.
14	Justify	This specifies the justification of the text if the text contains multiple lines.

15	offvalue	The associated control variable is set to 0 by default if the button is unchecked. We can change the state of an unchecked variable to some other one.
16	onvalue	The associated control variable is set to 1 by default if the button is checked. We can change the state of the checked variable to some other one.
17	Padx	The horizontal padding of the checkbutton
18	Pady	The vertical padding of the checkbutton.
19	Relief	The type of the border of the checkbutton. By default, it is set to FLAT.
20	selectcolor	The color of the checkbutton when it is set. By default, it is red.
21	selectimage	The image is shown on the checkbutton when it is set.
22	State	It represents the state of the checkbutton. By default, it is set to normal. We can change it to DISABLED to make the checkbutton unresponsive. The state of the checkbutton is ACTIVE when it is under focus.
24	underline	It represents the index of the character in the text which is to be underlined. The indexing starts with zero in the text.
25	variable	It represents the associated variable that tracks the state of the checkbutton.
26	Width	It represents the width of the checkbutton. It is represented in the number of characters that are represented in the form of texts.
27	wraplength	If this option is set to an integer number, the text will be broken into the number of pieces.

Methods

SN	Method	Description
1	deselect()	It is called to turn off the check button.
2	flash()	The checkbutton is flashed between the active and normal colors.
3	invoke()	This will invoke the method associated with the checkbutton.
4	select()	It is called to turn on the checkbutton.
5	toggle()	It is used to toggle between the different Checkbuttons.

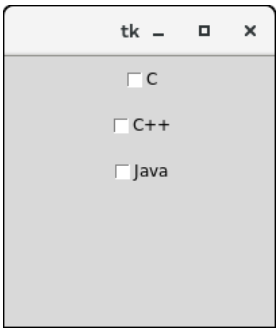
A variable defined using IntVar() function holds integer data where we can set integer data and can retrieve it as well using getter and setter methods

Example

```
from tkinter import *

top = Tk()

top.geometry("200x200")
```

```
checkvar1 = IntVar()

checkvar2 = IntVar()

checkvar3 = IntVar()

chkbtn1 = Checkbutton(top, text = "C", variable = checkvar1, onvalue = 1, offvalue = 0, height = 2, width = 10)

chkbtn2 = Checkbutton(top, text = "C++", variable = checkvar2, onvalue = 1, offvalue = 0, height = 2, width = 10)

chkbtn3 = Checkbutton(top, text = "Java", variable = checkvar3, onvalue = 1, offvalue = 0, height = 2, width = 10)

chkbtn1.pack()

chkbtn2.pack()

chkbtn3.pack()

top.mainloop()
```

OutPut



5) Python Tkinter Entry

The Entry widget is used for the single line text-box to the user to accept a value from the user. We can use the Entry widget to accept the text strings from the user. It can only be used for one line of text from the user. For multiple lines of text, we must use the text widget.

Syntax

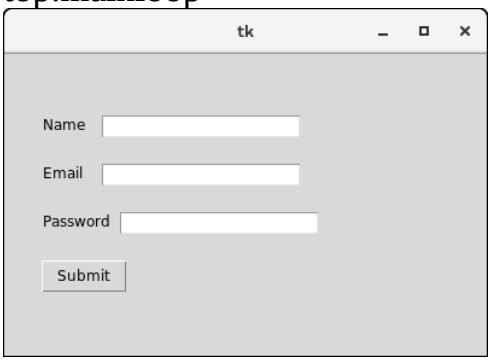
w = Entry (parent, options)

SN	Option	Description
1	Bg	The background color of the widget.
2	Bd	The border width of the widget in pixels.
3	Cursor	The mouse pointer will be changed to the cursor type set to the arrow, dot, etc.
4	exportselection	The text written inside the entry box will be automatically copied to the clipboard by default. We can set the exportselection to 0 to not copy this.
5	Fg	It represents the color of the text.
6	Font	It represents the font type of the text.
7	highlightbackground	It represents the color to display in the traversal highlight region when the widget does not have the input focus.
8	highlightcolor	It represents the color to use for the traversal highlight rectangle that is drawn around the widget when it has the input focus.
9	highlightthickness	It represents a non-negative value indicating the width of the highlight rectangle to draw around the outside of the widget when it has the input focus.
10	insertbackground	It represents the color to use as background in the area covered by the insertion cursor. This color will normally override either the normal background for the widget.
11	insertborderwidth	It represents a non-negative value indicating the width of the 3-D border to draw around the insertion cursor. The value may have any of the forms acceptable to Tk_GetPixels.
12	insertofftime	It represents a non-negative integer value indicating the number of milliseconds the insertion cursor should remain "off" in each blink cycle. If this option is zero, then the cursor doesn't blink: it is on all the time.
13	insertontime	Specifies a non-negative integer value indicating the number of milliseconds the insertion cursor should remain "on" in each blink cycle.
14	insertwidth	It represents the value indicating the total width of the insertion cursor. The value may have any of the forms acceptable to Tk_GetPixels.
15	Justify	It specifies how the text is organized if the text contains multiple lines.
16	Relief	It specifies the type of the border. Its default value is FLAT.
17	selectbackground	The background color of the selected text.
18	selectborderwidth	The width of the border to display around the selected task.
19	selectforeground	The font color of the selected task.
20	Show	It is used to show the entry text of some other type instead of the string. For example, the password is typed using stars (*).
21	textvariable	It is set to the instance of the StringVar to retrieve the text from the entry.
22	Width	The width of the displayed text or image.

23	xscrollcommand	The entry widget can be linked to the horizontal scrollbar if we want the user to enter more text then the actual width of the widget.

Example-

```
from tkinter import *

top = Tk()

top.geometry("400x250")

name = Label(top, text = "Name").place(x = 30,y = 50)

email = Label(top, text = "Email").place(x = 30, y = 90)

password = Label(top, text = "Password").place(x = 30, y = 130)

sbmitbtn = Button(top, text = "Submit",activebackground = "pink", activeforeground = "blue").place(x = 30, y = 170)

e1 = Entry(top).place(x = 80, y = 50)


e2 = Entry(top).place(x = 80, y = 90)


e3 = Entry(top).place(x = 95, y = 130)

top.mainloop
```



6) Python Tkinter Frame

Python Tkinter Frame widget is used to organize the group of widgets. It acts like a container which can be used to hold the other widgets. The rectangular areas of the screen are used to organize the widgets to the python application

Syntax

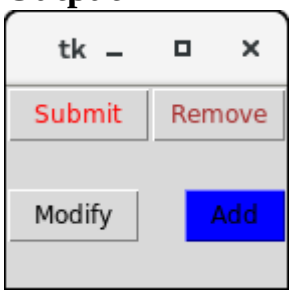
w = Frame(parent, options)

SN	Option	Description
1	Bd	It represents the border width.
2	Bg	The background color of the widget.
3	cursor	The mouse pointer is changed to the cursor type set to different values like an arrow, dot, etc.
4	height	The height of the frame.
5	highlightbackground	The color of the background color when it is under focus.
6	highlightcolor	The text color when the widget is under focus.
7	highlightthickness	It specifies the thickness around the border when the widget is under the focus.
8	relief	It specifies the type of the border. The default value if FLAT.
9	width	It represents the width of the widget

Example-

```
from tkinter import *

top = Tk()
top.geometry("140x100")
frame = Frame(top)
frame.pack()

leftframe = Frame(top)
leftframe.pack(side = LEFT)

rightframe = Frame(top)
rightframe.pack(side = RIGHT)

btn1 = Button(frame, text="Submit", fg="red",activebackground = "red")
btn1.pack(side = LEFT)

btn2 = Button(frame, text="Remove", fg="brown", activebackground = "brown")
btn2.pack(side = RIGHT)

btn3 = Button(rightframe, text="Add", fg="blue", activebackground = "blue")
btn3.pack(side = LEFT)

btn4 = Button(leftframe, text="Modify", fg="black", activebackground = "white")
btn4.pack(side = RIGHT)

top.mainloop()
```

Output



7) Label

The Label is used to specify the container box where we can place the text or images**.**

Syntax-

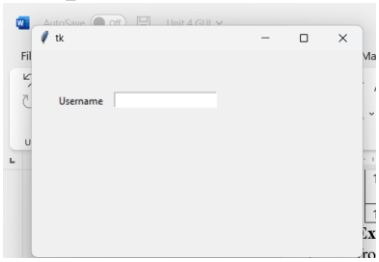
w = Label (top, options)

SN	Option	Description
1	Anchor	It specifies the exact position of the text within the size provided to the widget. The default value is CENTER, which is used to center the text within the specified space.
2	Bg	The background color displayed behind the widget.
3	Bitmap	It is used to set the bitmap to the graphical object specified so that, the label can represent the graphics instead of text.
4	Bd	It represents the width of the border. The default is 2 pixels.
5	Cursor	The mouse pointer will be changed to the type of the cursor specified, i.e., arrow, dot, etc.
6	Font	The font type of the text written inside the widget.
7	Fg	The foreground color of the text written inside the widget.
8	Height	The height of the widget.
9	Image	The image that is to be shown as the label.
10	Justify	It is used to represent the orientation of the text if the text contains multiple lines. It can be set to LEFT for left justification, RIGHT for right justification, and CENTER for center justification.
11	Padx	The horizontal padding of the text. The default value is 1.
12	Pady	The vertical padding of the text. The default value is 1.
13	Relief	The type of the border. The default value is FLAT.
14	Text	This is set to the string variable which may contain one or more line of text.
15	textvariable	The text written inside the widget is set to the control variable StringVar so that it can be accessed and changed accordingly.
16	underline	We can display a line under the specified letter of the text. Set this option to the number of the letter under which the line will be displayed.
17	Width	The width of the widget. It is specified as the number of characters.

Example

```
from tkinter import *

top = Tk()

top.geometry("400x250")

#creating label
uname = Label(top, text = "Username").place(x = 30,y = 50)
e1 = Entry(top,width = 20).place(x = 100, y = 50)
top.mainloop()
```

Output



8) Listbox

The Listbox widget is used to display the list items to the user. We can place only text items in the Listbox and all text items contain the same font and color

Syntax

w = Listbox(parent, options)

Options

SN	Option	Description
1	Bg	The background color of the widget.
2	Bd	It represents the size of the border. Default value is 2 pixel.
3	Cursor	The mouse pointer will look like the cursor type like dot, arrow, etc.
4	Font	The font type of the Listbox items.
5	Fg	The color of the text.
6	Height	It represents the count of the lines shown in the Listbox. The default value is 10.
7	highlightcolor	The color of the Listbox items when the widget is under focus.
8	highlightthickness	The thickness of the highlight.
9	Relief	The type of the border. The default is SUNKEN.
10	selectbackground	The background color that is used to display the selected text.
11	Selectmode	It is used to determine the number of items that can be selected from the list. It can set to BROWSE, SINGLE, MULTIPLE, EXTENDED.
12	Width	It represents the width of the widget in characters.
13	xscrollcommand	It is used to let the user scroll the Listbox horizontally.

Method-

SN	Method	Description
1	activate(index)	It is used to select the lines at the specified index.
2	curselection()	It returns a tuple containing the line numbers of the selected element or elements, counting from 0. If nothing is selected, returns an empty tuple.
3	delete(first, last = None)	It is used to delete the lines which exist in the given range.
4	get(first, last = None)	It is used to get the list items that exist in the given range.
5	index(i)	It is used to place the line with the specified index at the top of the widget.
6	insert(index, *elements)	It is used to insert the new lines with the specified number of elements before the specified index.
7	nearest(y)	It returns the index of the nearest line to the y coordinate of the Listbox widget.
8	see(index)	It is used to adjust the position of the listbox to make the lines specified by the index visible.
9	size()	It returns the number of lines that are present in the Listbox widget.
10	xview()	This is used to make the widget horizontally scrollable.
11	xview_moveto(fraction)	It is used to make the listbox horizontally scrollable by the fraction of width of the longest line present in the listbox.
12	xview_scroll(number, what)	It is used to make the listbox horizontally scrollable by the number of characters specified.
13	yview()	It allows the Listbox to be vertically scrollable.
14	yview_moveto(fraction)	It is used to make the listbox vertically scrollable by the fraction of width of the longest line present in the listbox.
15	yview_scroll (number, what)	It is used to make the listbox vertically scrollable by the number of characters specified.

Example-

```
from tkinter import *

top = Tk()

top.geometry("200x250")

lbl = Label(top,text = "A list of favourite countries...")

listbox = Listbox(top)

listbox.insert(1,"India")

listbox.insert(2, "USA")

listbox.insert(3, "Japan")

listbox.insert(4, "Austrelia")

lbl.pack()
```

listbox.pack()

top.mainloop()

Output

9) Menu

The Menu widget is used to create various types of menus (top level, pull down, and pop up) in the python application

Syntax-

w = Menu(top, options)

SN	Option	Description
1	activebackground	The background color of the widget when the widget is under the focus.
2	activeborderwidth	The width of the border of the widget when it is under the mouse. The default is 1 pixel.
3	activeforeground	The font color of the widget when the widget has the focus.
4	bg	The background color of the widget.
5	bd	The border width of the widget.
6	cursor	The mouse pointer is changed to the cursor type when it hovers the widget. The cursor type can be set to arrow or dot.
7	disabledforeground	The font color of the widget when it is disabled.
8	font	The font type of the text of the widget.
9	fg	The foreground color of the widget.
10	postcommand	The postcommand can be set to any of the function which is called when the mourse hovers the menu.
11	relief	The type of the border of the widget. The default type is RAISED.
12	image	It is used to display an image on the menu.
13	selectcolor	The color used to display the checkbutton or radiobutton when they are selected.
14	tearoff	By default, the choices in the menu start taking place from position 1. If we set the tearoff = 1, then it will start taking place from 0th position.
15	title	Set this option to the title of the window if you want to change the title of the window.

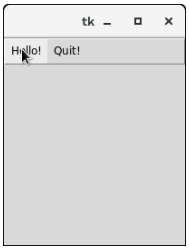
Methods

SN	Option	Description
1	add_command(options)	It is used to add the Menu items to the menu.

2	add_radiobutton(options)	This method adds the radiobutton to the menu.
3	add_checkbutton(options)	This method is used to add the checkbuttons to the menu.
4	add_cascade(options)	It is used to create a hierarchical menu to the parent menu by associating the given menu to the parent menu.
5	add_seperator()	It is used to add the seperator line to the menu.
6	add(type, options)	It is used to add the specific menu item to the menu.
7	delete(startindex, endindex)	It is used to delete the menu items exist in the specified range.
8	entryconfig(index, options)	It is used to configure a menu item identified by the given index.
9	index(item)	It is used to get the index of the specified menu item.
10	insert_seperator(index)	It is used to insert a seperator at the specified index.
11	invoke(index)	It is used to invoke the associated with the choice given at the specified index.
12	type(index)	It is used to get the type of the choice specified by the index.

Example-

```
from tkinter import *

top = Tk()

def hello():
   print("hello!")

# create a toplevel menu
menubar = Menu(top)
menubar.add_command(label="Hello!", command=hello)
menubar.add_command(label="Quit!", command=top.quit)

# display the menu
top.config(menu=menubar)

top.mainloop()
```



10) Python Tkinter Message

The Message widget is used to show the message to the user regarding the behaviour of the python application. The message widget shows the text messages to the user which can not be edited.

Syntax-

w = Message(parent, options)

SN	Option	Description
1	anchor	It is used to decide the exact position of the text within the space provided to the widget if the widget contains more space than the need of the text. The default is CENTER.
2	bg	The background color of the widget.
3	bitmap	It is used to display the graphics on the widget. It can be set to any graphical or image object.
4	bd	It represents the size of the border in the pixel. The default size is 2 pixel.
5	cursor	The mouse pointer is changed to the specified cursor type. The cursor type can be an arrow, dot, etc.
6	font	The font type of the widget text.
7	fg	The font color of the widget text.
8	height	The vertical dimension of the message.
9	image	We can set this option to a static image to show that onto the widget.
10	justify	This option is used to specify the alignment of multiple line of code with respect to each other. The possible values can be LEFT (left alignment), CENTER (default), and RIGHT (right alignment).
11	padx	The horizontal padding of the widget.
12	pady	The vertical padding of the widget.
13	relief	It represents the type of the border. The default type is FLAT.
14	text	We can set this option to the string so that the widget can represent the specified text.
15	textvariable	This is used to control the text represented by the widget. The textvariable can be set to the text that is shown in the widget.
16	underline	The default value of this option is -1 that represents no underline. We can set this option to an existing number to specify that nth letter of the string will be underlined.
17	width	It specifies the horizontal dimension of the widget in the number of characters (not pixel).
18	wraplength	We can wrap the text to the number of lines by setting this option to the desired number so that each line contains only that number of characters.

Example-

```
from tkinter import *

top = Tk()
top.geometry("100x100")
```

```
msg = Message( top, text = "Welcome to Javatpoint")

msg.pack()
top.mainloop()
```

11) Radiobutton

The Radiobutton widget is used to implement one-of-many selection in the python application. It shows multiple choices to the user out of which, the user can select only one out of them.

w = Radiobutton(top, options)

SN	Option	Description
1	activebackground	The background color of the widget when it has the focus.
2	activeforeground	The font color of the widget text when it has the focus.
3	anchor	It represents the exact position of the text within the widget if the widget contains more space than the requirement of the text. The default value is CENTER.
4	bg	The background color of the widget.
5	bitmap	It is used to display the graphics on the widget. It can be set to any graphical or image object.
6	borderwidth	It represents the size of the border.
7	command	This option is set to the procedure which must be called every-time when the state of the radiobutton is changed.
8	cursor	The mouse pointer is changed to the specified cursor type. It can be set to the arrow, dot, etc.
9	font	It represents the font type of the widget text.
10	fg	The normal foreground color of the widget text.
11	height	The vertical dimension of the widget. It is specified as the number of lines (not pixel).
12	highlightcolor	It represents the color of the focus highlight when the widget has the focus.
13	highlightbackground	The color of the focus highlight when the widget is not having the focus.
14	image	It can be set to an image object if we want to display an image on the radiobutton instead the text.
15	justify	It represents the justification of the multi-line text. It can be set to CENTER(default), LEFT, or RIGHT.
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 value is FLAT.

19	selectcolor	The color of the radio button when it is selected.
20	selectimage	The image to be displayed on the radiobutton when it is selected.
21	state	It represents the state of the radio button. The default state of the Radiobutton is NORMAL. However, we can set this to DISABLED to make the radiobutton unresponsive.
22	text	The text to be displayed on the radiobutton.
23	textvariable	It is of String type that represents the text displayed by the widget.
24	underline	The default value of this option is -1, however, we can set this option to the number of character which is to be underlined.
25	value	The value of each radiobutton is assigned to the control variable when it is turned on by the user.
26	variable	It is the control variable which is used to keep track of the user's choices. It is shared among all the radiobuttons.
27	width	The horizontal dimension of the widget. It is represented as the number of characters.
28	wraplength	We can wrap the text to the number of lines by setting this option to the desired number so that each line contains only that number of characters.

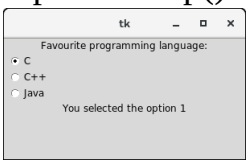
Methods

SN	Method	Description
1	deselect()	It is used to turn of the radiobutton.
2	flash()	It is used to flash the radiobutton between its active and normal colors few times.
3	invoke()	It is used to call any procedure associated when the state of a Radiobutton is changed.
4	select()	It is used to select the radiobutton.

Example-

```
from tkinter import *

def selection():
   selection = "You selected the option " + str(radio.get())
   label.config(text = selection)

top = Tk()
top.geometry("300x150")
radio = IntVar()
lbl = Label(text = "Favourite programming language:")
lbl.pack()
R1 = Radiobutton(top, text="C", variable=radio, value=1, command=selection)
R1.pack( anchor = W )
```

```
R2 = Radiobutton(top, text="C++", variable=radio, value=2,   command=selection)
R2.pack( anchor = W )

R3 = Radiobutton(top, text="Java", variable=radio, value=3,  command=selection)
R3.pack( anchor = W)

label = Label(top)
label.pack()
top.mainloop()
```



12) Messagebox

The messagebox module is used to display the message boxes in the python applications. There are the various functions which are used to display the relevant messages depending upon the application requirements**.**

Syntax-

messagebox.function_name(title, message [, options])

Parameters

function_name: It represents an appropriate message box function.

title: It is a string which is shown as a title of a message box.

message: It is the string to be displayed as a message on the message box.

options: There are various options which can be used to configure the message dialog box.

options

1. **default**-The default option is used to mention the types of the default button, i.e. ABORT, RETRY, or IGNORE in the message box.
2. **Parent-** The parent option specifies the parent window on top of which, the message box is to be displayed.

Functions-

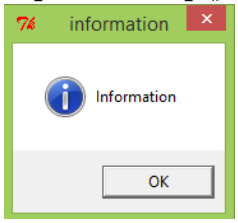
1. **showinfo()**

The showinfo() messagebox is used where we need to show some relevant information to the user.

Example-

```
from tkinter import *

from tkinter import messagebox

top = Tk()
```

```
top.geometry("100x100")

messagebox.showinfo("information","information")

top.mainloop()
```



2. **showwarning()**
   This method is used to display the warning to the user**.**
   Example
   messagebox.showwarning("warning","Warning")
3. **showerror()**
   This method is used to display the error message to the user**.**
   messagebox.showerror("error","Error")
4. **askquestion()**
   This method is used to ask some question to the user which can be answered in yes or no
   messagebox.askquestion("Confirm","Are you sure?")
5. **askokcancel()**
   This method is used to confirm the user's action regarding some application activity.
   messagebox.askokcancel("Redirect","Redirecting you to www.javatpoint.com")
6. **askyesno()**
   This method is used to ask the user about some action to which, the user can answer in yes or no
   messagebox.askyesno("Application","Got It?")
7. **askretrycancel()**
   This method is used to ask the user about doing a particular task again or not.
   messagebox.askretrycancel("Application","try again?")

Configuring Widget-

When you create a widget in Tkinter, you have to pass in several values as parameters, used to configure certain features for that Widget**.** If you have change the setting after create the widget For this we have the Tkinter Config() function, which can be used on any widget to change settings that you may have applied earlier, or haven't applied yet.

Python Tkinter Config() function, used to simply change the text on a label.

Example-

The button is linked to a function that calls the config() on the label when the button is pressed. All you have to do is assign a new value to the option you want changed in the config() function.

```
from tkinter import *

root = Tk()
frame = Frame(root)

def dosomething():
    mylabel.config(text = "Goodbye World")


mylabel =Label(root,text = "Hello World", bg = "red")
mylabel.pack(padx = 5, pady = 10)

mybutton =Button(root,text = "Click Me", command = dosomething)
mybutton.pack(padx = 5, pady = 10)


frame.pack(padx = 5, pady = 5)
root.mainloop()
```