NFSU

विद्यया अमृतं अश्नुते

National Forensic Sciences University

Knowledge | Wisdom | Fulfilment

An Institution of National Importance
(Ministry of Home Affairs, Government of India)

TA 2 Assignment

FEB 2022

Under the Guidance of **Prof. Sarang Rajvansh**

PREPARED BY

Sagar Shah

ENROLLMENT NUMBER:

101CTMTCS2122040

Introduction & Background of Tkinter

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. Tkinter is lightweight and relatively painless to use compared to other frameworks. This makes it a compelling choice for building GUI applications in Python, especially for applications where a modern sheen is unnecessary, and the top priority is to build something that's functional and cross-platform quickly.

Tkinter Widgets

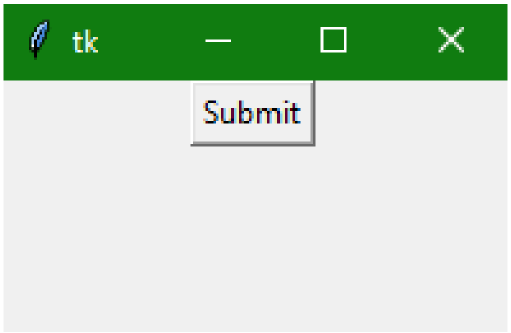
Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

Button

The Button gadget in Tkinter is essentially used to add a button in any GUI Application. In Python, while utilising the Tkinter button gadget, we can undoubtedly alter the style of the button like adding a foundation tone to it, changing the tallness and width of the button, or the situation of the button, and so forth without any problem.
Options: activebackground, bd, bg, command, activeforeground, fg, font, height, image, highlightcolour, justify, padx, pady, underline, width, Wraplength, state

Syntax:
Button(master, options)

Example:

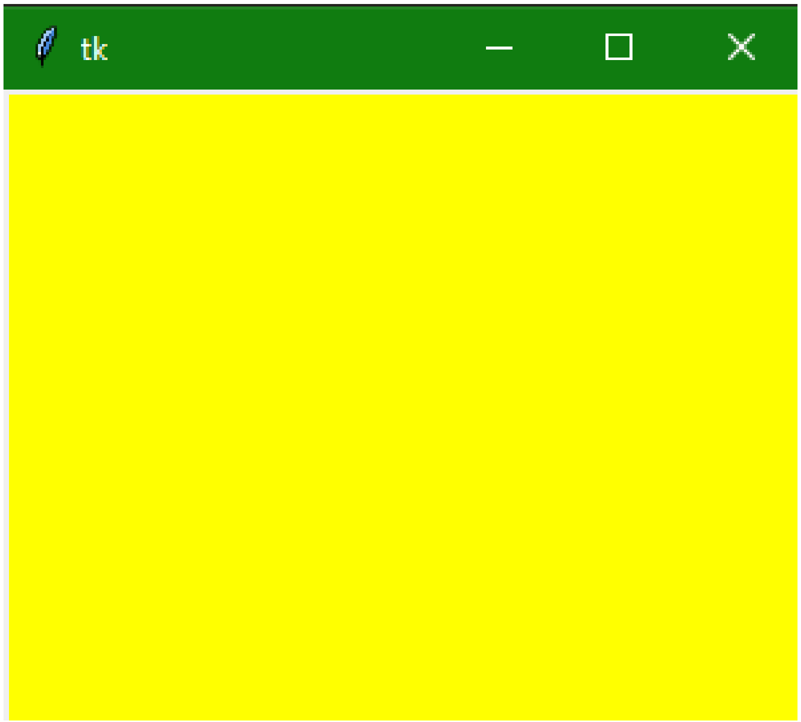


Canvas

Tkinter Canvas gadget is mostly utilised as a broadly useful gadget which is utilised to attract anything on the application window Tkinter. This gadget is essentially used to draw illustrations and plots, drawings, diagrams, and showing pictures. You can draw a few complex formats with the assistance of material, for instance, polygon, square shape, oval, text, curve bitmap, designs, and so on. Material gadgets are likewise used to make graphical editors.
Options: bd, bg, cursor, confine, height, width,highlightcolour, xscrollcommand, yscrollcommand, scrollregion, xscrollincrement, yscrollincrement

Syntax:
w = Canvas(master, option=value)

Example:

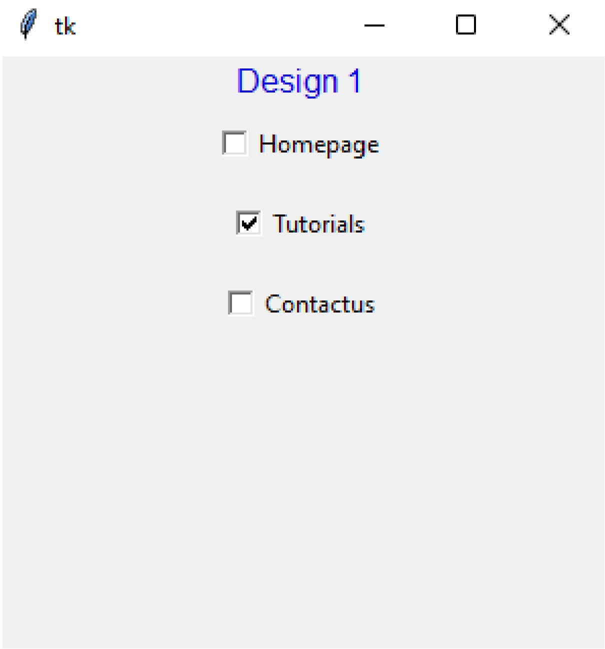


CheckButton

If you want to display a number of options in a form, where users can check to select any option, we can use the Tkinter Checkbutton widget. It allows you to select multiple options or a single option at a time just by clicking the button corresponding to each option.
Options: activebackground, bd, bg, bitmap, command, activeforeground, fg, font, height, image, cursor, disableforeground, higlightcolor, justify, padx, pady, underline, width, Wraplength, Variable, offvalue, onvalue, text, state, selectcolor, selectimage

Syntax:
w = CheckButton(master, option=value)

Example:



Design 1

☐ Homepage

☑ Tutorials

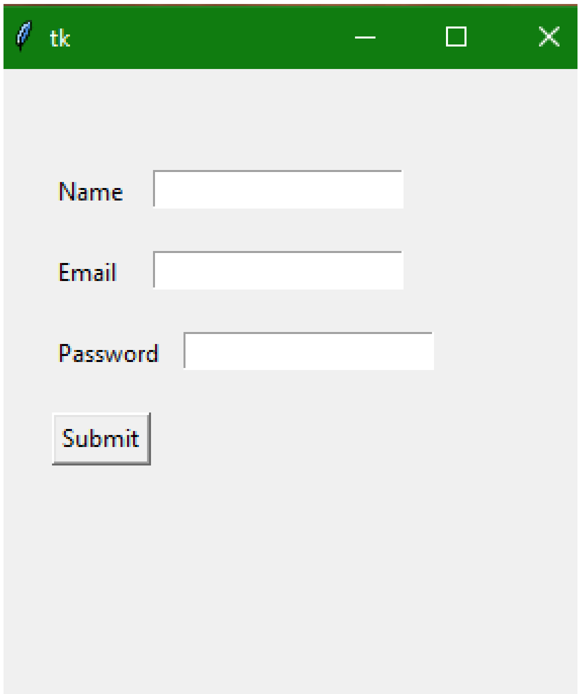
☐ Contactus

Entry

If you need to get a little bit of text from a user, like a name or an email address or a contact number then use an Entry widget. The Entry widget is mainly used to display a small text box that the user can type some text into. There are the number of options available to change the styling of the Entry Widget. It is important to note that the Entry widget is only used to get a single-line text from the user because in the case of multiline text the text widget will be used. This widget is mainly used to accept text strings from the user.
Options: bg, bd, cursor, exportselection, fg, font, highlightbackground, highlightcolor, justify, relief, selectbackground, selectforeground, selectborderwidth, width, textvariable, show, xscrollcommand, insertackground

Syntax:
w = Entry(master, option=value)

Example:

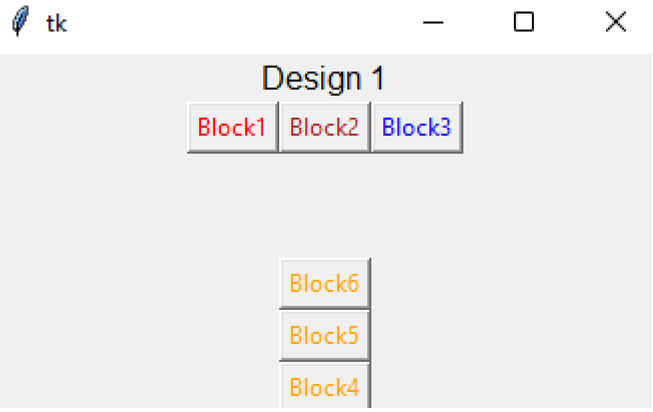


Frame

The Tkinter Frame widget is used to group and organise the widgets in a better and friendly way.
The Frame widget is basically a container (an invisible container) whose task is to hold other widgets and arrange them with respect to each other.
The Tkinter frame widget makes up a rectangular region on the screen.
Options: bd, bg, cursor, height, width, highlightbackground, highlightthickness, relief, highlightcolor

Syntax:
w = Frame(master, option=value)

Example:

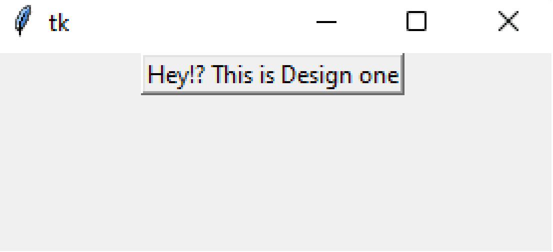


Label

The label widget in Tkinter is used to display boxes where you can place your images and text.
The label widget is mainly used to provide a message about the other widgets used in the Python Application to the user.
You can change or update the text inside the label widget anytime you want.
This widget uses only one font at the time of displaying some text.
You can perform other tasks like underline some part of the text and you can also span text to multiple lines.
There are various options available to configure the text or the part of the text shown in the Label.

Syntax:
w = Label(master, option=value)

Example:

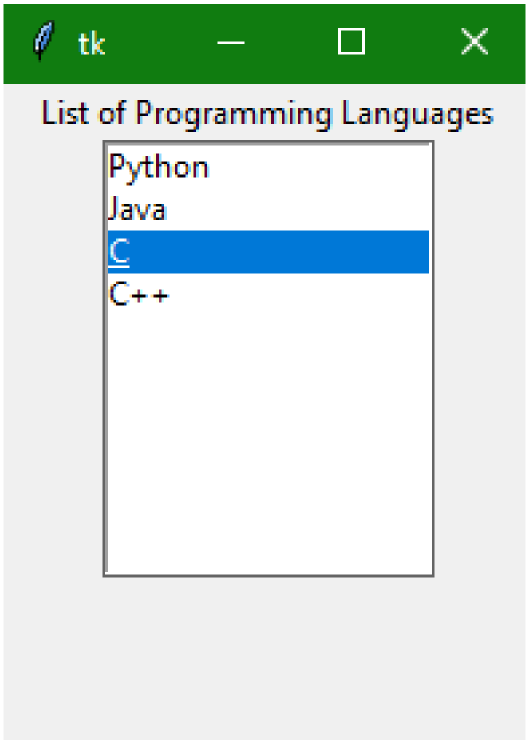


Listbox

Tkinter Listbox widget in Python which is used to display different types of items to the user in the form of a List inside a box and the user can select the items.
The items contain the same type of font and the same font colour.
It is important to note here that only text items can be placed inside a Listbox widget.
From this list of items, the user can select one or more items according to the requirements.

Syntax:
w = Listbox(master, option=value)

Example:



List of Programming Languages

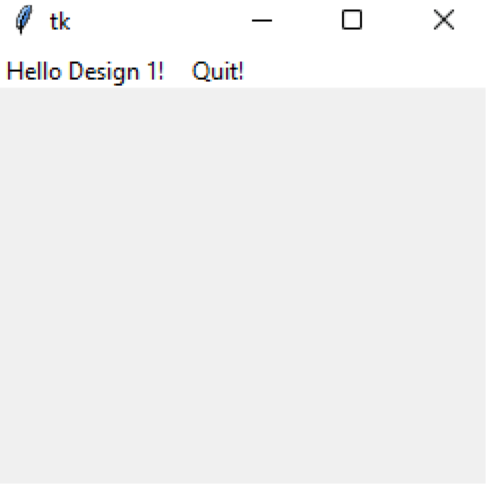
Python
Java
C
C++

Menu

The Tkinter Menu widget is used to create different types of menus in Python Application. The following types of menus can be created using the Tkinter Menu widget: pop-up, pull-down, and top level.
Top-level menus are those menus that are displayed just under the title bar of the root or any other top-level windows. For example, all the websites have a top navigation menu just below the URL bar in the browser.
Menus are commonly used to provide convenient access to options like opening any file, quitting any task, and manipulating data in an application.

Syntax:
w = Menu(master, option=value)

Example:



MenuButton

Menubutton widget in Python which is used to create a dropdown menu which can be clicked by the user to see the options

This widget is used to provide various types of menus in the Python Application.

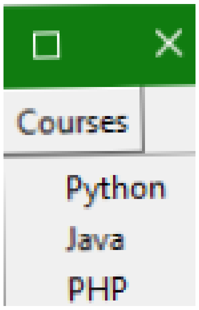
It is important to note that every Menubutton in an application is associated with a Menu widget and that in return can display the choices for that menubutton whenever the user clicks on it.

The Tkinter Menubutton widget provides the user with an option to select the appropriate choice that exists within the application.

Syntax:

w = Menubutton(master, option=value)

Example:

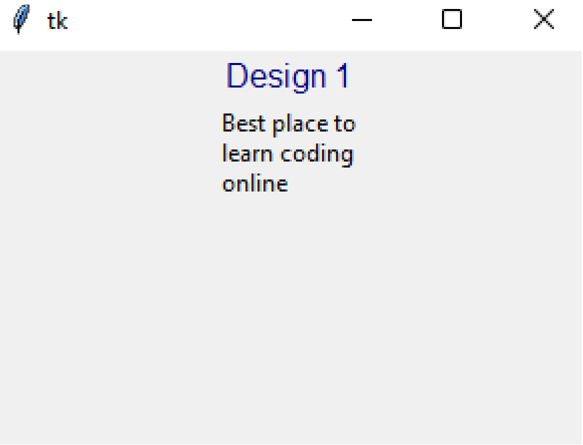


Message

The Tkinter Message Widget in Python is mainly used to show some message to the user who is using the application.
The message displayed by the Tkinter Message Widget is of non-editable type and it can be in multiline.
The message displayed by the Tkinter Message widget contains single font text.
The functionality of this widget is very similar to the Tkinter Label widget, but there is a difference and that is the message widget can automatically wrap the text.

Syntax:
w = Message(master, option=value)

Example:



Design 1

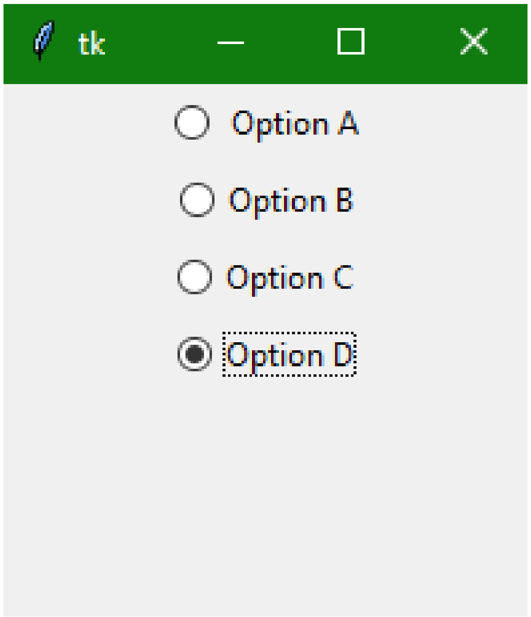
Best place to
learn coding
online

RadioButton

Tkinter radiobutton widget is used to implement multiple-choice options that are mainly created in user input forms.
This widget offers multiple selections to the user and allows the user to select only one option from the given ones. Thus it is also known as implementing one-of-many selection in a Python Application.
Also, different methods can also be associated with radiobutton.
You can also display multiple line text and images on the radiobutton.
Each radiobutton displays a single value for a particular variable.
You can also keep a track of the user's selection of the radiobutton because it is associated with a single variable

Syntax:
w = Radiobutton(master, option=value)

Example:



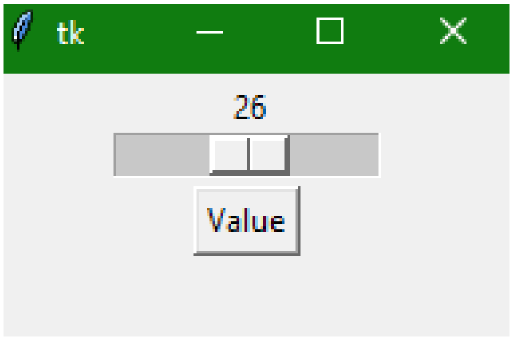
○ Option A
○ Option B
○ Option C
● Option D

Scale

Scale widget in Python which is used to add a graphical slider object which the user can slide and choose a number, as a numeric value is attached to this slider scale and as you move the slider up/down or right/left the numeric value attached to it increases or decreases and you can set the slider to the value you wish to select.
The sliding bar provided by the scale widget is helpful in selecting the values just by sliding from left to right or top to bottom depending upon the orientation of the sliding bar in our application.
The scale widget is used as an alternative to the Entry widget if the purpose of the Entry widget is to take numeric input from the user within a given range of values.
You can also control minimum and maximum values along with the resolution of the scale.

Syntax:
w = Scale(master, option=value)

Example:

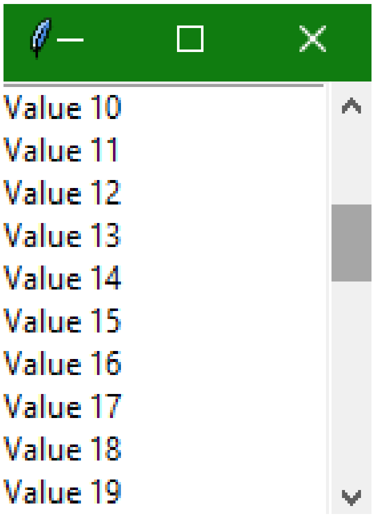


Scrollbar

To scroll up or down or right or left the content in a Python desktop application, the Tkinter Scrollbar widget is used.
To scroll the content of other widgets like Listbox, canvas, etc we use this widget.
Both Horizontal and Vertical scrollbars can be created in the Trinket Entry widget.

Syntax:
w = Scrollbar(master, option=value)

Example:



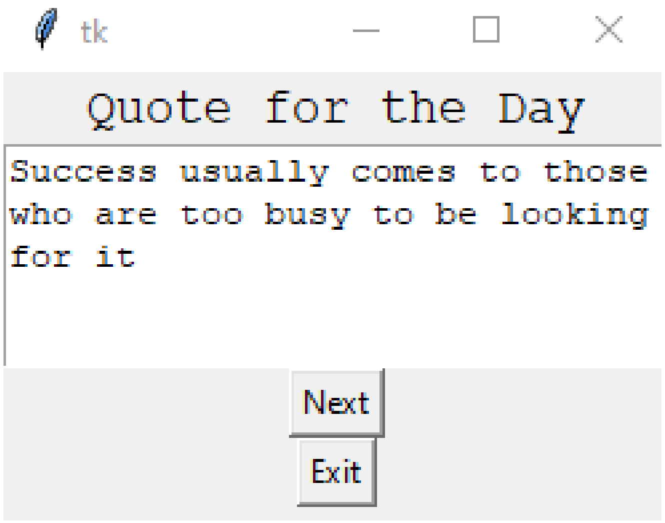
Value 10
Value 11
Value 12
Value 13
Value 14
Value 15
Value 16
Value 17
Value 18
Value 19

Text

The text widget is used to provide a multiline textbox (input box) because in Tkinter single-line textbox is provided using Entry widget.
You can use various styles and attributes with the Text widget.
You can also use marks and tabs in the Text widget to locate the specific sections of the text.
Media files like images and links can also be inserted in the Text Widget.
There are some variety of applications where you need multiline text like sending messages or taking long inputs from users, or to show editable long format text content in application, etc. use cases are fulfilled by this widget.
Thus in order to show textual information, we will use the Text widget.

Syntax:
w = Text(master, option=value)

Example:



Quote for the Day

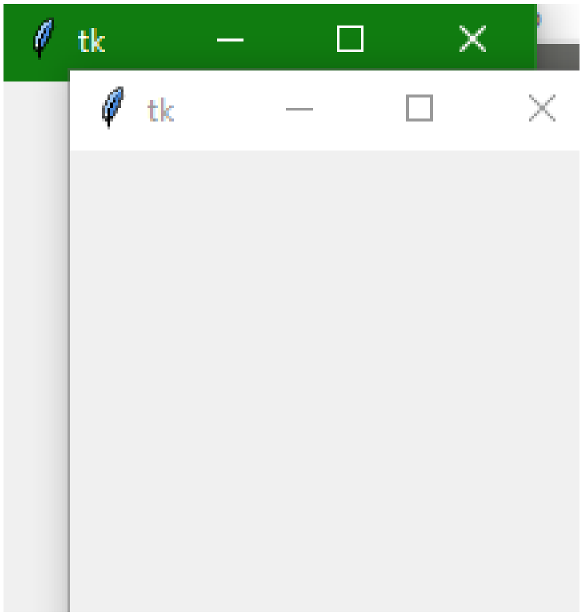
Success usually comes to those who are too busy to be looking for it

Toplevel

With the help of the Tkinter Toplevel widget, you can provide extra information to the user in a separate window on top of the parent window.
This top-level window created using the Toplevel widget is directly organised and managed by the window manager.
It is not necessary for the top-level windows to have parents on their top.
You can create multiple top-level windows one over the other.

Syntax:
w = Toplevel(master, option=value)

Example:

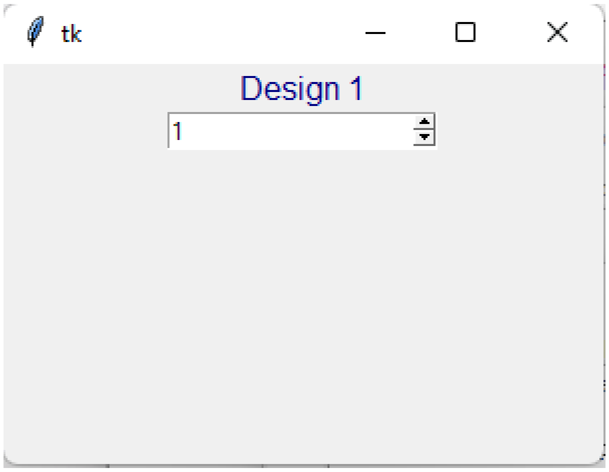


Spinbox

The Spinbox widget in Tkinter in Python is used to select a value from the specified given range of values.
It is different from the Tkinter Scale widget (Scale widget being more stylish) in terms of style, but more or less, fulfils the same purpose.
For example, when you want to have a dropdown of numerical values like year of birth (from 1950 to 2020) or a dropdown for users to choose their age, we can use the Tkinter Spinbox widget.
This widget is an alternative to Entry widget, when we want the user to enter a numeric value within a specific range.

Syntax:
w = Spinbox(master, option=value)

Example:

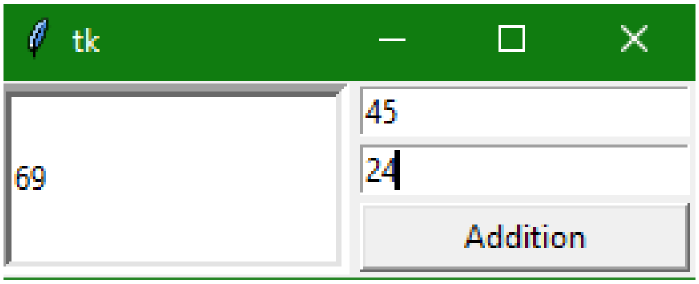


PanedWindow

PanedWindow widget which is mainly a container widget containing one or more than one child widgets which are also known as Panes.
This widget arranges child widgets either in a vertical or in a horizontal manner.
It is also known as the Geometry Manager widget.
This widget is used to implement different layouts in a Python desktop application created using the Tkinter module.
The child widgets inside the PanedWindow widget can be resized by the user by moving separator lines sashes using the mouse.
You can implement multiple panes using the PanedWindow widget.

Syntax:
w = PanedWindow(master, option=value)

Example:



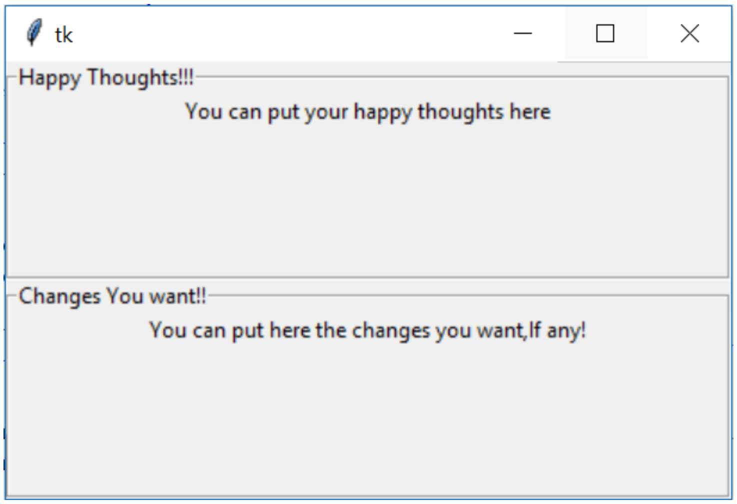
```
69
45
24
```

LabelFrame

The LabelFrame widget is mainly used to draw borders around the child widgets.
This widget is a bordered container widget and is used to group the related widgets in a Tkinter application to provide a better user experience to the user.
For example, we can group the radiobutton widgets used in an application using the labelframe widget.
One can also add a title for the LabelFrame widget.

Syntax:
w = LabelFrame(master, option=value)

Example:



Happy Thoughts!!!

You can put your happy thoughts here

Changes You want!!

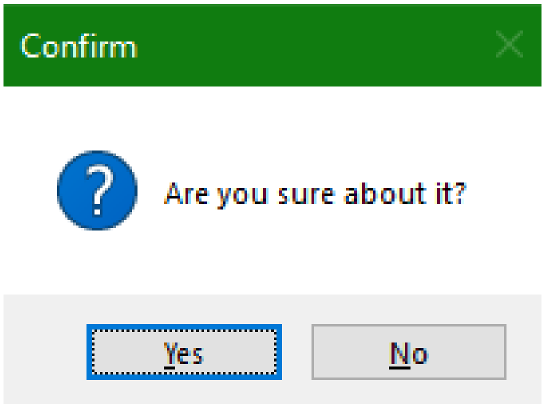
You can put here the changes you want,If any!

MessageBox

In order to display message boxes in a desktop application, we use the MessageBox module in Tkinter.
There are various functions present in this module which helps to provide an appropriate type of message according to the requirement.
With the help of this module, we can create pop-up message boxes to take user input.
The functions of the MessageBox module are as follows: showError(), askretrycancel(), showwarning(), etc., all of which are used to create a messagebox.

Syntax:
w = messagebox.function_name(title, message [, options])

Example:



Confirm

Are you sure about it?

Thank You