RAJ COMPUTERSSince 1994

Indira Nagar, College Road, Vinay Nagar

**GUI Base =>**

**MVC Pattern=>**

**Model View Controller**

**1.Components/ Controls/ Widget**

**2.Look=How to Appear/view**

**3.Task to do on event/trigger**

**How to create a tkinter:**

1. Importing the module – tkinter
2. Create the main window (container)
3. Add any number of widgets to the main window
4. Apply the event Trigger on the widgets.

import tkinter//Here it also Imports all modules

from tkinter import \*//Here also we can handle all modules.

**How to Create a Window/container**

M=Tk()

#M.loop()#this is used open a window still window is not closed.

**Layout Setting Methods-**

1. **pack() method:**It organizes the widgets in blocks before placing in the parent widget.
2. **grid() method:**It organizes the widgets in grid (table-like structure) before placing in the parent widget.
3. **place() method:**It organizes the widgets by placing them on specific positions (x,y) directed by the programmer.

**Components or Widget-**

**Creating a Way-**

var = Widget(masterContainter, option=value)

**Button :** It is used to draw pictures and other complex layout like graphics, text and widgets.

* **activebackground**: to set the background color when button is under the cursor.
* **activeforeground**: to set the foreground color when button is under the cursor.
* **bg**: to set he normal background color.
* **command**: to call a function.
* **font**: to set the font on the button label.
* **image**: to set the image on the button.
* **width**: to set the width of the button.
* **height**: to set the height of the button.

|  |
| --- |
|  |

1. **CheckButton:**
   * **Title**: To set the title of the widget.
   * **activebackground**: to set the background color when widget is under the cursor.
   * **activeforeground**: to set the foreground color when widget is under the cursor.
   * **bg**: to set he normal background
   * **command**: to call a function.
   * **font**: to set the font on the button label.
   * **image**: to set the image on the widget.
2. **Entry:**It is used to input the single line text entry from the user.. For multi-line text input, Text widget is used.

w=Entry(master, option=value)

* + **bd**: to set the border width in pixels.
  + **bg**: to set the normal background color.
  + **cursor**: to set the cursor used.
  + **command**: to call a function.
  + **highlightcolor**: to set the color shown in the focus highlight.
  + **width**: to set the width of the button.
  + **height**: to set the height of the button.

1. **Label**: It refers to the display box where you can put any text or image which can be updated any time as per the code.  
   * **bg**: to set he normal background color.
   * **fg** to set he normal foreground color.
   * **command**: to call a function.
   * **font**: to set the font on the button label.
   * **image**: to set the image on the button.
   * **width**: to set the width of the button.
   * **height**” to set the height of the button.
2. **Listbox**: It offers a list to the user from which the user can accept any number of options.  
   * **highlightcolor**: To set the color of the focus highlight when widget has to be focused.
   * **bg**: to set he normal background color.
   * **bd**: to set the border width in pixels.
   * **font**: to set the font on the button label.
   * **image**: to set the image on the widget.
   * **width**: to set the width of the widget.
   * **height**: to set the height of the widget.

\*\* insert function is used to insert in Listbox

|  |
| --- |
| Lb = Listbox(top)  Lb.insert(1, 'Python') |

1. **RadioButton:**It is used to offer multi-choice option to the user. It offers several options to the user and the user has to choose one option.  
   * **activebackground**: to set the background color when widget is under the cursor.
   * **activeforeground**: to set the foreground color when widget is under the cursor.
   * **bg**: to set he normal background color.
   * **command**: to call a function.
   * **font**: to set the font on the button label.
   * **image**: to set the image on the widget.
   * **width**: to set the width of the label in characters.
   * **height**: to set the height of the label in characters.

|  |
| --- |
| from tkinter import \*  root = Tk()  v = IntVar()  Radiobutton(root, text='GfG', variable=v, value=1).pack(anchor=W)  Radiobutton(root, text='MIT', variable=v, value=2).pack(anchor=W)  mainloop() |

Scrollbar=>

master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

* + **width**: to set the width of the widget.
  + **activebackground**: To set the background when mouse is over the widget.
  + **bg**: to set he normal background color.
  + **bd**: to set the size of border around the indicator.
  + **cursor**: To appear the cursor when the mouse over the menubutton.

|  |
| --- |
| from tkinter import \*  root = Tk()  scrollbar = Scrollbar(root)  scrollbar.pack( side = RIGHT, fill = Y )  mylist = Listbox(root, yscrollcommand = scrollbar.set )  for line in range(100):     mylist.insert(END, 'This is line number' + str(line))  mylist.pack( side = LEFT, fill = BOTH )  scrollbar.config( command = mylist.yview )  mainloop() |

1. **Text:**
   * **highlightcolor**: To set the color of the focus highlight when widget has to be focused.
   * **insertbackground**: To set the background of the widget.
   * **bg**: to set he normal background color.
   * **font**: to set the font on the button label.
   * **image**: to set the image on the widget.
   * **width**: to set the width of the widget.
   * **height**: to set the height of the widget.

|  |  |
| --- | --- |
| **Event** | **Description** |
| <Button> | A mouse button is pressed with the mouse pointer over the widget. The detail part specifies which button, e.g. The left mouse button is defined by the event <Button-1>, the middle button by <Button-2>, and the rightmost mouse button by <Button-3>.  <Button-4> defines the scroll up event on mice with wheel support and and <Button-5> the scroll down.  If you press down a mouse button over a widget and keep it pressed, Tkinter will automatically "grab" the mouse pointer. Further mouse events like Motion and Release events will be sent to the current widget, even if the mouse is moved outside the current widget. The current position, relative to the widget, of the mouse pointer is provided in the x and y members of the event object passed to the callback. You can use ButtonPress instead of Button, or even leave it out completely: , , and <1> are all synonyms. |
| <Motion> | The mouse is moved with a mouse button being held down. To specify the left, middle or right mouse button use <B1-Motion>, <B2-Motion> and <B3-Motion> respectively. The current position of the mouse pointer is provided in the x and y members of the event object passed to the callback, i.e. event.x, event.y |
| <ButtonRelease> | Event, if a button is released. To specify the left, middle or right mouse button use <ButtonRelease-1>, <ButtonRelease-2>, and <ButtonRelease-3> respectively. The current position of the mouse pointer is provided in the x and y members of the event object passed to the callback, i.e. event.x, event.y |
| <Double-Button> | Similar to the Button event, see above, but the button is double clicked instead of a single click. To specify the left, middle or right mouse button use <Double-Button-1>, <Double-Button-2>, and <Double-Button-3> respectively. You can use Double or Triple as prefixes. Note that if you bind to both a single click (<Button-1>) and a double click (<Double-Button-1>), both bindings will be called. |
| <Enter> | The mouse pointer entered the widget. Attention: This doesn't mean that the user pressed the Enter key!. <Return> is used for this purpose. |
| <Leave> | The mouse pointer left the widget. |
| <FocusIn> | Keyboard focus was moved to this widget, or to a child of this widget. |
| <FocusOut> | Keyboard focus was moved from this widget to another widget. |
| <Return> | The user pressed the Enter key. You can bind to virtually all keys on the keyboard: The special keys are Cancel (the Break key), BackSpace, Tab, Return(the Enter key), Shift\_L (any Shift key), Control\_L (any Control key), Alt\_L (any Alt key), Pause, Caps\_Lock, Escape, Prior (Page Up), Next (Page Down), End, Home, Left, Up, Right, Down, Print, Insert, Delete, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, Num\_Lock, and Scroll\_Lock. |
| <Key> | The user pressed any key. The key is provided in the char member of the event object passed to the callback (this is an empty string for special keys). |
| a | The user typed an "a" key. Most printable characters can be used as is. The exceptions are space (<space>) and less than (<less>). Note that 1 is a keyboard binding, while <1> is a button binding. |
| <Shift-Up> | The user pressed the Up arrow, while holding the Shift key pressed. You can use prefixes like Alt, Shift, and Control. |
| <Configure> | The size of the widget changed. The new size is provided in the width and height attributes of the event object passed to the callback. On some platforms, it can mean that the location changed. |
|  |  |

tkMessageBox -

import Tkinter

import tkMessageBox

Here is the simple syntax to create this widget −

tkMessageBox.FunctionName(title, message [, options])

Parameters

* **FunctionName** − This is the name of the appropriate message box function.
* **title** − This is the text to be displayed in the title bar of a message box.
* **message** − This is the text to be displayed as a message.
* **options** − options are alternative choices that you may use to tailor a standard message box. Some of the options that you can use are default and parent. The default option is used to specify the default button, such as ABORT, RETRY, or IGNORE in the message box. The parent option is used to specify the window on top of which the message box is to be displayed.

You could use one of the following functions with dialogue box −

* showinfo()
* showwarning()
* showerror ()
* askquestion()
* askokcancel()
* askyesno ()
* askretrycancel ()