

[COURSES](#)[Login](#)[HIRE WITH US](#)

Python GUI – tkinter

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

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.

Importing tkinter is same as importing any other module in the python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x is 'tkinter'.

```
import tkinter
```

There are two main methods used you the user need to remember while creating the Python application with GUI.



Ad closed by Google

Stop seeing this ad

Why this ad? ⓘ

1. **Tk(screenName=None, baseName=None, className='Tk', useTk=1)**: To create a main window, tkinter offers a method 'Tk(screenName=None, baseName=None, className='Tk', useTk=1)'. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:

```
m=tkinter.Tk() where m is the name of the main window object
```

2. **mainloop()**: There is a method known by the name mainloop() is used when you are ready for the application to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event till the window is not closed.

```
m.mainloop()
```

```
import tkinter
m = tkinter.Tk()
...
widgets are added here
...
m.mainloop()
```

tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.



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 directed by the programmer.

There are a number of widgets which you can put in your tkinter application. Some of the major widgets are explained below:

1. **Button:** To add a button in your application, this widget is used.

The general syntax is:

```
w=Button(master, option=value)
```

master is the parameter used to represent the parent window.

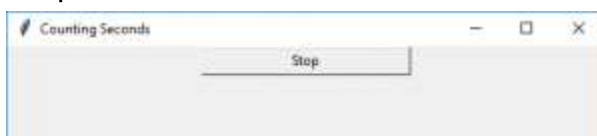
There are number of options which are used to change the format of the Buttons.

Number of options can be passed as parameters separated by commas. Some of them are listed below.

- **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 the 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.

```
import tkinter as tk
r = tk.Tk()
r.title('Counting Seconds')
button = tk.Button(r, text='Stop', width=25, command=r.destroy)
button.pack()
r.mainloop()
```

Output:



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

The general syntax is:

```
w = Canvas(master, option=value)
```

master is the parameter used to represent the parent window.

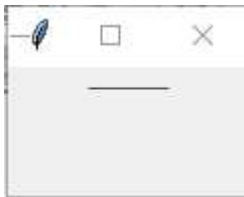
There are number of options which are used to change the format of the widget.

Number of options can be passed as parameters separated by commas. Some of them are listed below.

- **bd:** to set the border width in pixels.
- **bg:** to set the normal background color.
- **cursor:** to set the cursor used in the canvas.
- **highlightcolor:** to set the color shown in the focus highlight.
- **width:** to set the width of the widget.
- **height:** to set the height of the widget.

```
from tkinter import *  
master = Tk()  
w = Canvas(master, width=400, height=600)  
w.pack()  
canvas_height=200  
canvas_width=200  
y = int(canvas_height / 2)  
w.create_line(0, y, canvas_width, y )  
mainloop()
```

Output:



3. **CheckButton:** To select any number of options by displaying a number of options to a user as toggle buttons. The general syntax is:

```
w = CheckButton(master, option=value)
```

There are number of options which are used to change the format of this widget.

Number of options can be passed as parameters separated by commas. Some of them are listed below.

- **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 the normal background color.

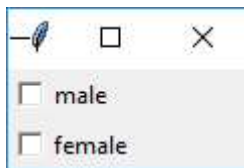
Secret Code:

Attach a File:nd color.

- **command**: to call a function.
- **font**: to set the font on the button label.
- **image**: to set the image on the widget.

```
from tkinter import *
master = Tk()
var1 = IntVar()
Checkbutton(master, text='male', variable=var1).grid(row=0, sticky=W)
var2 = IntVar()
Checkbutton(master, text='female', variable=var2).grid(row=1, sticky=W)
mainloop()
```

Output:



4. **Entry**: It is used to input the single line text entry from the user.. For multi-line text input, Text widget is used.

The general syntax is:



```
w=Entry(master, option=value)
```

master is the parameter used to represent the parent window.

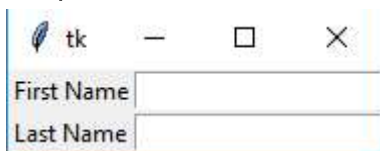
There are number of options which are used to change the format of the widget.

Number of options can be passed as parameters separated by commas. Some of them are listed below.

- **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.

```
from tkinter import *
master = Tk()
Label(master, text='First Name').grid(row=0)
Label(master, text='Last Name').grid(row=1)
e1 = Entry(master)
e2 = Entry(master)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)
mainloop()
```

Output:



5. **Frame**: It acts as a container to hold the widgets. It is used for grouping and organizing the widgets. The general syntax is:

```
w = Frame(master, option=value)
```

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.

- **highlightcolor**: To set the color of the focus highlight when widget has to be focused.



- **bd**: to set the border width in pixels.
- **bg**: to set the normal background color.
- **cursor**: to set the cursor used.
- **width**: to set the width of the widget.
- **height**: to set the height of the widget.

```
from tkinter import *
```

```
root = Tk()
frame = Frame(root)
frame.pack()
bottomframe = Frame(root)
bottomframe.pack( side = BOTTOM )
redbutton = Button(frame, text = 'Red', fg='red')
redbutton.pack( side = LEFT)
greenbutton = Button(frame, text = 'Brown', fg='brown')
greenbutton.pack( side = LEFT )
bluebutton = Button(frame, text ='Blue', fg='blue')
bluebutton.pack( side = LEFT )
blackbutton = Button(bottomframe, text ='Black', fg='black')
blackbutton.pack( side = BOTTOM)
root.mainloop()
```

Output:



6. **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.

The general syntax is:

```
w=Label(master, option=value)
```

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.

- **bg**: to set he normal background color.
- **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.

```
from tkinter import *
root = Tk()
w = Label(root, text='GeeksForGeeks.org!')
w.pack()
root.mainloop()
```

Output:



7. **Listbox**: It offers a list to the user from which the user can accept any number of options.

The general syntax is:

```
w = Listbox(master, option=value)
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.

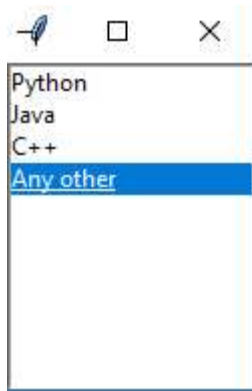
Number of options can be passed as parameters separated by commas. Some of them are listed below.

- **highlightcolor**: To set the color of the focus highlight when widget has to be focused.
- **bg**: to set the 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.

```
from tkinter import *
...
top = Tk()
lb = Listbox(top)
lb.insert(1, 'Python')
lb.insert(2, 'Java')
lb.insert(3, 'C++')
lb.insert(4, 'Any other')
lb.pack()
top.mainloop()
```



Output:



8. **MenuButton**: It is a part of top-down menu which stays on the window all the time. Every menubutton has its own functionality. The general syntax is:

```
w = MenuButton(master, option=value)
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.

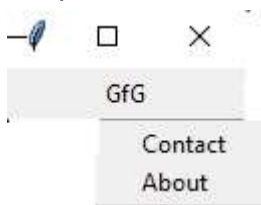
- **activebackground**: To set the background when mouse is over the widget.
- **activeforeground**: To set the foreground when mouse is over the widget.
- **bg**: to set the normal background color.
- **bd**: to set the size of border around the indicator.
- **cursor**: To appear the cursor when the mouse over the menubutton.
- **image**: to set the image on the widget.
- **width**: to set the width of the widget.
- **height**: to set the height of the widget.
- **highlightcolor**: To set the color of the focus highlight when widget has to be focused.

```
from tkinter import *
```

```
top = Tk()
mb = Menubutton ( top, text = '&GfG&');
mb.grid()
mb.menu = Menu ( mb, tearoff = 0 )
mb['&menu&'] = mb.menu
cVar = IntVar()
aVar = IntVar()
mb.menu.add_checkbutton ( label = 'Contact', variable = cVar )
mb.menu.add_checkbutton ( label = 'About', variable = aVar )
mb.pack()
top.mainloop()
```



Output:



9. **Menu:** It is used to create all kinds of menus used by the application.

The general syntax is:

```
w = Menu(master, option=value)
```

master is the parameter used to represent the parent window.

There are number of options which are used to change the format of this widget.

Number of options can be passed as parameters separated by commas. Some of them are listed below.

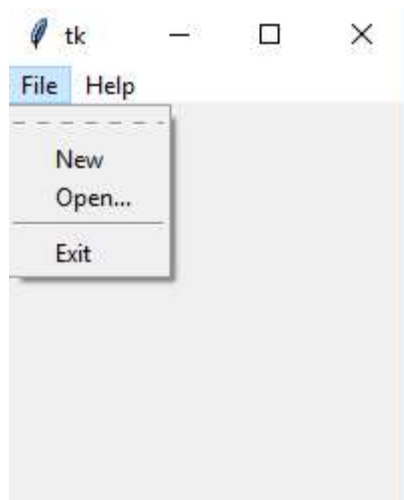
- **title:** To set the title of the widget.
- **activebackground:** to set the background color when widget is under the cursor.
- **activeforeground:** to set the foreground color when widget is under the cursor.
- **bg:** to set the normal background color.
- **command:** to call a function.
- **font:** to set the font on the button label.
- **image:** to set the image on the widget.

```
from tkinter import *
```

```
root = Tk()
menu = Menu(root)
root.config(menu=menu)
filemenu = Menu(menu)
menu.add_cascade(label='File', menu=filemenu)
filemenu.add_command(label='New')
filemenu.add_command(label='Open...')
filemenu.add_separator()
filemenu.add_command(label='Exit', command=root.quit)
helpmenu = Menu(menu)
menu.add_cascade(label='Help', menu=helpmenu)
helpmenu.add_command(label='About')
mainloop()
```



Output:



10. **Message:** It refers to the multi-line and non-editable text. It works same as that of Label.

The general syntax is:



```
w = Message(master, option=value)
```

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.

- **bd:** to set the border around the indicator.
- **bg:** to set the 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.

```
from tkinter import *
main = Tk()
ourMessage = 'This is our Message'
messageVar = Message(main, text = ourMessage)
messageVar.config(bg='lightgreen')
messageVar.pack( )
main.mainloop( )
```

Output:



11. **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.

The general syntax is:

```
w = RadioButton(master, option=value)
```

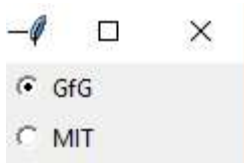
There are number of options which are used to change the format of this widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

- **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 the 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()
```



Output:



12. **Scale:** It is used to provide a graphical slider that allows to select any value from that scale. The general syntax is:

```
w = Scale(master, option=value)
```

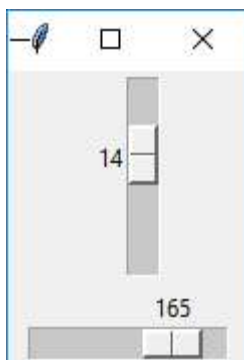
master is the parameter used to represent the parent window.

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.

- **cursor:** To change the cursor pattern when the mouse is over the widget.
- **activebackground:** To set the background of the widget when mouse is over the widget.
- **bg:** to set the normal background color.
- **orient:** Set it to HORIZONTAL or VERTICAL according to the requirement.
- **from_:** To set the value of one end of the scale range.
- **to:** To set the value of the other end of the scale range.
- **image:** to set the image on the widget.
- **width:** to set the width of the widget.

```
from tkinter import *
master = Tk()
w = Scale(master, from_=0, to=42)
w.pack()
w = Scale(master, from_=0, to=200, orient=HORIZONTAL)
w.pack()
mainloop()
```

Output:



13. **Scrollbar**: It refers to the slide controller which will be used to implement listed widgets.

The general syntax is:

```
w = Scrollbar(master, option=value)
```

master is the parameter used to represent the parent window.

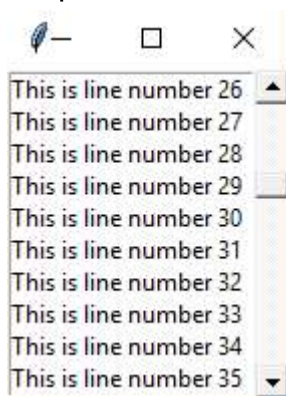
There are number of options which are used to change the format of the widget.

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 the 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()
```

Output:



14. **Text**: To edit a multi-line text and format the way it has to be displayed.

The general syntax is:

```
w =Text(master, option=value)
```

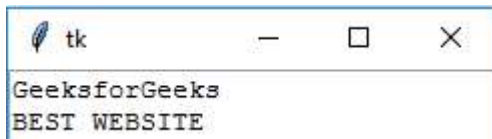


There are number of options which are used to change the format of the text. Number of options can be passed as parameters separated by commas. Some of them are listed below.

- **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 the 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.

```
from tkinter import *
root = Tk()
T = Text(root, height=2, width=30)
T.pack()
T.insert(END, 'GeeksforGeeks\nBEST WEBSITE\n')
mainloop()
```

Output:



15. **TopLevel**: This widget is directly controlled by the window manager. It don't need any parent window to work on. The general syntax is:

```
w = TopLevel(master, option=value)
```

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.

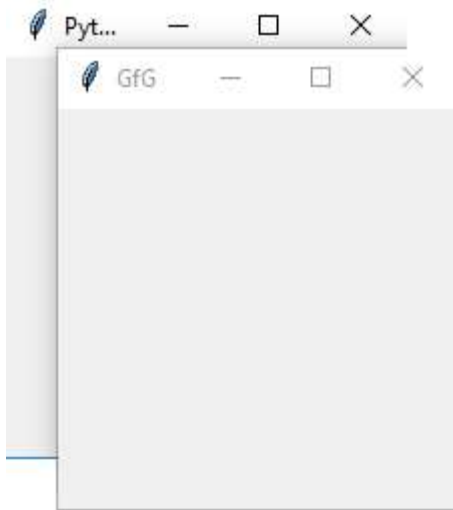
- **bg**: to set the normal background color.
- **bd**: to set the size of border around the indicator.
- **cursor**: To appear the cursor when the mouse over the menubutton.
- **width**: to set the width of the widget.
- **height**: to set the height of the widget.

```
from tkinter import *
root = Tk()
root.title('GfG')
```



```
top = Toplevel()
top.title('Python')
top.mainloop()
```

Output:



16. **SpinBox**: It is an entry of 'Entry' widget. Here, value can be input by selecting a fixed value of numbers. The general syntax is:

```
w = SpinBox(master, option=value)
```

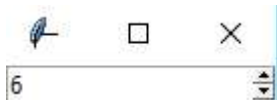
There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

- **bg**: to set the normal background color.
- **bd**: to set the size of border around the indicator.
- **cursor**: To appear the cursor when the mouse over the menubutton.
- **command**: To call a function.
- **width**: to set the width of the widget.
- **activebackground**: To set the background when mouse is over the widget.
- **disabledbackground**: To disable the background when mouse is over the widget.
- **from_**: To set the value of one end of the range.
- **to**: To set the value of the other end of the range.

```
from tkinter import *
master = Tk()
w = Spinbox(master, from_ = 0, to = 10)
w.pack()
mainloop()
```



Output:



17. **PanedWindow** It is a container widget which is used to handle number of panes arranged in it. The general syntax is:

```
w = PanedWindow(master, option=value)
```

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.

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

```
from tkinter import *
m1 = PanedWindow()
m1.pack(fill = BOTH, expand = 1)
left = Entry(m1, bd = 5)
m1.add(left)
m2 = PanedWindow(m1, orient = VERTICAL)
m1.add(m2)
top = Scale( m2, orient = HORIZONTAL)
m2.add(top)
mainloop()
```

Output:



This article is contributed by **Rishabh Bansal**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](https://www.geeksforgeeks.org/contribute) or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

