

Python is a language which can be used across different applications including GUI programs. There are many ways to create GUI application like ipywidgets,tkinter,Wxpython,jpython etc. Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

Tkinter has a variety of commonly used GUI elements (like ,button,menus,labels,entry,areas etc.) ,these are called Widgets

Widgets Widgets are basic building blocks of GUI programming and are used to display information or get input from the user.Some are as follows

- 1.Button The Button widget is used to display buttons in your 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 widget is used to display a number of options as checkboxes. The user can select multiple options at a time.
- 4.Entry The Entry widget is used to display a single-line text field for accepting values from a user.
- 5.Frame The Frame widget is used as a container widget to organize other widgets.
- 6.Label The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
- 7.Listbox The Listbox widget is used to provide a list of options to a user.
- 8.Menubutton The Menubutton widget is used to display menus in your application.
- 9.Menu The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.
- 10.Message The Message widget is used to display multiline text fields for accepting values from a user.
- 11.Radiobutton The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.
- 12.Scale The Scale widget is used to provide a slider widget.
- 13.Scrollbar The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.
- 14.Text The Text widget is used to display text in multiple lines.
- 15.Toplevel The Toplevel widget is used to provide a separate window container.

16.Spinbox The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values.

17.PanedWindow A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically.

18.LabelFrame A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.

19.tkMessageBox This module is used to display message boxes in your applications.

```
In [1]: from tkinter import *
```

```
In [2]: window=Tk()#creates the instance of tk class or window creation  
window.mainloop()#is a method on the main window which is used to execute the  
application infinitely till user interrupts or close it manually
```

```
In [3]: root=Tk()  
root.geometry("750x450")# it is used to set the size of tk window by default it  
will appear on (150,150)  
root.mainloop()
```

```
In [4]: root=Tk()  
root.geometry("750x250+350+350")# it is used to set the size of tk window now it  
will appear on (350,350)  
root.mainloop()
```

```
In [5]: root=Tk()  
root.attributes('-fullscreen',True)#to set window as fullscreen  
root.mainloop()
```

```
In [6]: root=Tk()  
root.geometry("750x350")  
root.title("this is my First Window")#to set title of the window  
root.mainloop()
```

```
In [7]: #Creating a Label widget and adding to the window  
root=Tk()  
root.geometry("750x350")  
root.title("Adding label")  
#creating a Label widget  
mylabel=Label(root,text="hello students")  
#place the Label in the window  
mylabel.pack(pady=50)#used to place the widget in the window  
root.mainloop()
```

```
In [8]: #Creating a Label and changing the foreground(font) and background color  
root=Tk()
```

```

root.geometry("750x350")
root.title("Adding label")
#creating a Label widget
mylabel=Label(root,text="hello students",foreground='white',background='black')
#place the Label in the window
mylabel.pack(pady=50)#used to place the widget in the window
root.mainloop()

```

In [9]:

```

#creating two labels and placing it in the window

root=Tk()
root.geometry("750x550")
root.title("Adding label")
#creating a Label widget
mylabel1=Label(root,text="hello students",background='pink')
mylabel2=Label(root,text="NIET welcomes you all",background='green')
#place the Label in the window
mylabel1.pack(pady=50)
mylabel2.pack(pady=120)
root.mainloop()

```

In [10]:

```

#placing the labels in the grid
root=Tk()
root.geometry("750x550")
root.title("Adding label")
#creating a Label widget
mylabel1=Label(root,text="hello students",background='pink')
mylabel2=Label(root,text="NIET welcomes you all",background='green')
#place the Label in the window using grid
mylabel1.grid(row=0,column=0)
mylabel2.grid(row=0,column=1)
root.mainloop()

```

In [11]:

```

# creating and placing together
root=Tk()
root.geometry("750x550")
root.title("Adding label")
#creating a Label widget
mylabel1=Label(root,text="hello students",background='pink').grid(row=0,column=0)
mylabel2=Label(root,text="NIET welcomes you all",background='green').grid(row=0,column=1)
#place the Label in the window using grid
root.mainloop()

```

In [12]:

```

#button widget: clickable to trigger the event
root=Tk()
root.geometry("750x550")
root.title("Button")
#creating a disabled button
mybutton=Button(root,text='Click me',padx=30,state=DISABLED)
mybutton.pack(pady=50)
root.mainloop()

```

In [13]:

```

#button widget: clickable to trigger the event

```

```

root=Tk()
root.geometry("750x550")
root.title("Button")
#creating a clickable button
mybutton=Button(root,text='Click me',padx=30)
mybutton.pack(pady=50)
root.mainloop()

```

```

In [14]: #button widget: clickable to trigger the event
root=Tk()
root.geometry("750x550")
root.title("Event Triggering Button")
def click():#we have to create a function for the event to trigger by button
    mylabel=Label(root,text="you clicked",fg='white',bg='black',pady=20,font=25)
    mylabel.pack()
#command parameter has to be set as the crated function
mybutton=Button(root,text='Click me',command=click,padx=30)
mybutton.pack(pady=50)
root.mainloop()

```

```

In [15]: #button click counter
root=Tk()
root.geometry("750x550")
root.title("counter Button")
counter=0
def click():#we have to create a function for the event to trigger by button
    global counter
    global mylabel
    counter=counter+1
    #configure is used to attach a value with Label
    mylabel.configure(text=f'you clicked {counter} times')

mylabel=Label(root,text="no click",fg='white',bg='black',pady=20,font=25)
mylabel.pack()
mybutton=Button(root,text='Click me',command=click,padx=30,font=25)
mybutton.pack(pady=50)
root.mainloop()

```

```

In [16]: #button click counter with stop condition
root=Tk()
root.geometry("750x550")
root.title("counter Button")
counter=0
def click():#we have to create a function for the event to trigger by button
    global counter
    global mylabel
    counter=counter+1
    #configure is used to attach a value with Label
    mylabel.configure(text=f'you clicked {counter} times')
    if counter>10:
        mybutton.configure(state=DISABLED)
        mylabel.configure(text="Sorry!! No more clicks allowed")

mylabel=Label(root,text="no click",fg='white',bg='black',pady=20,font=25)
mylabel.pack()
mybutton=Button(root,text='Click me',command=click,padx=30,font=25)

```

```
mybutton.pack(pady=50)
root.mainloop()
```

```
In [17]: #Entry widgets(text field): used to accept single line user inputs
#create textbox to get data from user
root=Tk()
root.geometry("750x550")
root.title("Entry widget (textbox)")
#Entry function is used to create single line text box
ent=Entry(root,borderwidth=5,width=30)
ent.pack()
root.mainloop()
```

```
In [18]: #we can change bg and fg color
root=Tk()
root.geometry("750x550")
root.title("Entry widget (textbox)")
#Entry function is used to create single line text box
ent=Entry(root,borderwidth=5,width=30,fg='white',bg='black')
ent.pack()
root.mainloop()
```

```
In [20]: #to get data from textbox:get() method
root=Tk()
root.geometry("750x550")
root.title("Entry widget (textbox)")
ent=Entry(root,borderwidth=5,width=30,fg='white',bg='black')
ent.pack()
def click():
    x=ent.get()
    mylabel=Label(root,text="NIET WELCOMES "+x,width=120)
    mylabel.pack()
    ent.delete(0,END)#to remove data from textbox
mybutton=Button(root,text="Enter",command=click,width=20)
mybutton.pack()
root.mainloop()
```

```
In [23]: #to insert data in entry box: insert()
root=Tk()
root.geometry("750x550")
root.title("Entry widget (textbox)")
ent=Entry(root,borderwidth=5,width=30,fg='white',bg='black')
ent.pack()
ent.insert(0,"Enter your Name")
def click():
    x=ent.get()
    mylabel=Label(root,text="NIET WELCOMES "+x,width=120)
    mylabel.pack()
    ent.delete(0,END)#to remove data from textbox
mybutton=Button(root,text="Enter",command=click,width=20)
mybutton.pack()
root.mainloop()
```

```
In [22]:
```

```
#text widget can be used for multi line user inputs
root=Tk()
root.geometry("750x550")
root.title("Text widget (textbox)")
txt=Text(root)
txt.configure(bg='SteelBlue2',height=10)
txt.pack()
root.mainloop()
```

In [27]:

```
#text widget insert,get and delete
root=Tk()
root.geometry("750x550")
root.title("Text widget (textbox)")
txt=Text(root)
txt.configure(bg='SteelBlue2',height=2)
txt.pack()
txt.insert('1.0',"Enter your Data")
name=''
def submit():
    global name
    name=txt.get("1.0","1.3")#Line.column
    mylabel=Label(root,text=name,width=120)
    mylabel.pack()
    txt.delete('1.0','1.3')
button=Button(root,text='submit',command=submit)
button.pack()
root.mainloop()
```

In [29]:

```
# create frame widget
win=Tk() # instance of Tk() frame
win.geometry("750x250")

# create frame
myframe=LabelFrame(win,text='MY frame',padx=5,pady=5)
myframe.pack(padx=140,pady=10) # to check the location of frame

# create Label
mylabel=Label(myframe,text="this is label")
mylabel.pack()

# create button
mybutton=Button(myframe,text="This is Button")
mybutton.pack()
win.mainloop()
```

In [ ]:

```
# create radio button
win=Tk()
r=IntVar() # used to provide variable in tkinter
r.set('2')
def onclick(value):
    global mylabel
    mylabel.destroy()
    mylabel=Label(win,text=value)
    mylabel.pack()

# create radio button
```

```

Radiobutton(win,text='Option 1',variable=r,value=1,command=lambda:onclick(r.get())).pack
Radiobutton(win,text='Option 2',variable=r,value=2,command=lambda:onclick(r.get())).pack
Radiobutton(win,text='Option 3',variable=r,value=3,command=lambda:onclick(r.get())).pack

mylabel=Label(win,text=r.get())
mylabel.pack()

win.mainloop()

```

In [ ]:

```

# message box
from tkinter import *
win=Tk()
win.geometry("1500x300")

def messagedisplay():
    messagebox.showinfo("Warning","you have clicked the button")
mylabel=Label(win,text="hello",font=20)
mylabel.pack(pady=50)

mybutton=Button(win,text="dont click me",command=messagedisplay)
mybutton.pack()

win.mainloop()

```

In [ ]:

```

# destroy the widget or window
win=Tk()
win.geometry("750x250")

def click():
    mylabel.destroy()
    win.destroy()
mylabel=Label(win,text="hello")
mylabel.pack()

mybutton=Button(win,text="cancel",command=click)
mybutton.pack()

win.mainloop()

```

In [ ]:

```

# canvas : to draw the shape or figure or style
from tkinter import *
win=Tk()
win.geometry("750x250")

def style():
    label.configure(font=('impact',25,'italic'),fg='white',bg='black')
    button.configure(text='close',command=lambda:win.destroy())
# create canvas
canvas=Canvas(win,width=600,height=200,bg='bisque')
canvas.pack(fill=BOTH,expand=True)

# create rectangle
canvas.create_rectangle(50,20,80,40,fill='red')

#create Label inside of canvas
label=Label(canvas,text="welcome students",font=20)

```

```
label.pack(pady=14)
# create button insdie of canvas
button=Button(canvas,text='click',command=style)
button.pack(pady=20)
win.mainloop()
```

## Simple Calculator app

In [2]:

```
from tkinter import *
root=Tk()
root.title("Simple Calculator")
# creating main frame
mainframe=Frame(root,width=45,bd=10,relief=RIDGE,bg="blue")
mainframe.pack()
inner=Frame(mainframe,width=45,bd=10,relief=RIDGE,bg="black")
inner.pack()
e=Entry(inner,width=65,borderwidth=5)
e.grid(row=0,column=0,columnspan=4,padx=10,pady=1)

def onclick(num):
    x=e.get()
    e.delete(0,END)
    e.insert(0,str(x)+str(num))

def clear():
    e.delete(0,END)

def add():
    global first,op
    op='+'
    first=e.get()
    e.delete(0,END)

def sub():
    global first,op
    op='-'
    first=e.get()
    e.delete(0,END)

def mul():
    global first,op
    op='*'
    first=e.get()
    e.delete(0,END)

def div():
    global first,op
    op='/'
    first=e.get()
    e.delete(0,END)

def equal():
    second=e.get()
    if op=='+':
        result=float(first)+float(second)
    elif op=='-':
        result=float(first)-float(second)
```



```

elif op=='*':
    result=float(first)*float(second)
elif op=='/':
    result=float(first)/float(second)
e.delete(0,END)
e.insert(0,result)

#####
#####Button widget#####
button_1=Button(inner,text='1',padx=30,pady=10,relief=RIDGE,font=15,width=4,
    command=lambda:onclick(1))
button_2=Button(inner,text='2',padx=30,pady=10,relief=RIDGE,font=15,width=4,
    command=lambda:onclick(2))
button_3=Button(inner,text='3',padx=30,pady=10,relief=RIDGE,font=15,width=4,
    command=lambda:onclick(3))
button_4=Button(inner,text='4',padx=30,pady=10,relief=RIDGE,font=15,width=4,
    command=lambda:onclick(4))
button_5=Button(inner,text='5',padx=30,pady=10,relief=RIDGE,font=15,width=4,
    command=lambda:onclick(5))
button_6=Button(inner,text='6',padx=30,pady=10,relief=RIDGE,font=15,width=4,
    command=lambda:onclick(6))
button_7=Button(inner,text='7',padx=30,pady=10,relief=RIDGE,font=15,width=4,
    command=lambda:onclick(7))
button_8=Button(inner,text='8',padx=30,pady=10,relief=RIDGE,font=15,width=4,
    command=lambda:onclick(8))
button_9=Button(inner,text='9',padx=30,pady=10,relief=RIDGE,font=15,width=4,
    command=lambda:onclick(9))
button_0=Button(inner,text='0',padx=30,pady=10,relief=RIDGE,font=15,width=4,
    command=lambda:onclick(0))

# operation button
button_add=Button(inner,text='+',padx=30,pady=10,relief=RIDGE,font=15,width=4,
    command=add)
button_sub=Button(inner,text='-',padx=30,pady=10,relief=RIDGE,font=15,width=4,command=s
button_mul=Button(inner,text='*',padx=30,pady=10,relief=RIDGE,font=15,width=4,command=m
button_div=Button(inner,text='/',padx=30,pady=10,relief=RIDGE,font=15,width=4,command=d
button_equal=Button(inner,text='=',padx=30,pady=10,relief=RIDGE,font=15,width=4,command
button_clear=Button(inner,text='C',padx=30,pady=10,relief=RIDGE,font=15,width=4,command

#####place the number button#####
button_1.grid(row=3,column=0)
button_2.grid(row=3,column=1)
button_3.grid(row=3,column=2)
button_4.grid(row=2,column=0)
button_5.grid(row=2,column=1)
button_6.grid(row=2,column=2)
button_7.grid(row=1,column=0)
button_8.grid(row=1,column=1)
button_9.grid(row=1,column=2)
button_0.grid(row=4,column=0)

button_add.grid(row=1,column=3)
button_sub.grid(row=2,column=3)
button_mul.grid(row=3,column=3)
button_div.grid(row=4,column=3)
button_equal.grid(row=4,column=2)
button_clear.grid(row=4,column=1)
root.mainloop()

```

In [3]:

```
# Change the color upon hovering over Button in Tkinter

win= Tk()
win.geometry("750x250")
#Define functions
def on_enter(e):
    button.config(background='OrangeRed3', foreground= "white")

def on_leave(e):
    button.config(background= 'SystemButtonFace', foreground= 'black')

button= Button(win, text= "Click Me", font= ('Helvetica 13 bold'))
button.pack(pady= 20)

#Bind the Enter and Leave Events to the Button
button.bind('<Enter>', on_enter)
button.bind('<Leave>', on_leave)

win.mainloop()
```

In [84]:

```
# Automatically close window after a certain time in Tkinter

win = Tk()
win.geometry("750x270")

Label(win, text= "This window will get closed after 5 seconds...",
font=('Helvetica 20 bold')).pack(pady=20)

#Automatically close the window after 5 seconds
win.after(5000,lambda:win.destroy())

win.mainloop()
```

In [87]:

```
#creating a listbox widgets
from tkinter import *

import tkinter

win = Tk()
win.geometry("750x270")
Lb1 = Listbox(win)
Lb1.insert(1, "Python")
Lb1.insert(2, "Perl")
Lb1.insert(3, "C")
Lb1.insert(4, "PHP")
Lb1.insert(5, "JSP")
Lb1.insert(6, "Ruby")

Lb1.pack()
win.mainloop()
```

In [89]:

```
# Combobox widget using tkinter

import tkinter as tk
from tkinter import ttk
```

```

# Creating tkinter window
window = tk.Tk()
window.title('Combobox')
window.geometry('500x250')

# Label text for title
ttk.Label(window, text = "GFG Combobox Widget",
          background = 'green', foreground = "white",
          font = ("Times New Roman", 15)).grid(row = 0, column = 1)

# Label
ttk.Label(window, text = "Select the Month :",
          font = ("Times New Roman", 10)).grid(column = 0,
          row = 5, padx = 10, pady = 25)

# Combobox creation
n = tk.StringVar()
monthchoosen = ttk.Combobox(window, width = 27, textvariable = n)

# Adding combobox drop down list
monthchoosen['values'] = (' January',
                        ' February',
                        ' March',
                        ' April',
                        ' May',
                        ' June',
                        ' July',
                        ' August',
                        ' September',
                        ' October',
                        ' November',
                        ' December')

monthchoosen.grid(column = 1, row = 5)
window.mainloop()

```

## ipywidgets

Simple Widget Introduction What are widgets? Widgets are eventful python objects that have a representation in the browser, often as a control like a slider, textbox, etc.

What can they be used for? You can use widgets to build interactive GUIs for your notebooks. You can also use widgets to synchronize stateful and stateless information between Python and JavaScript. Using widgets To use the widget framework, you need to import ipywidgets.

```
In [2]: import ipywidgets as widgets
```

```
In [92]: widgets.IntSlider(
        value=4,
        min=-5,
        max=25,
        step=1,
```

```
description='Integer slider:',  
)
```

In [95]:

```
widgets.FloatSlider(  
    value=7.5,  
    min=0,  
    max=10.0,  
    step=0.01,  
    description='Test:',  
    disabled=False,  
    continuous_update=False,  
    orientation='horizontal',  
    readout=True,  
    readout_format='.2f',  
)
```

In [96]:

```
widgets.FloatLogSlider(  
    value=10,  
    base=10,  
    min=-10, # max exponent of base  
    max=10, # min exponent of base  
    step=0.2, # exponent step  
    description='Log Slider'  
)
```

In [97]:

```
widgets.IntRangeSlider(  
    value=[5, 7],  
    min=0,  
    max=10,  
    step=1,  
    description='Test:',  
    disabled=False,  
    continuous_update=False,  
    orientation='horizontal',  
    readout=True,  
    readout_format='d',  
)
```

In [98]:

```
widgets.IntProgress(  
    value=7,  
    min=0,  
    max=10,  
    description='Copying:',  
    bar_style='info',  
    style={'bar_color': 'Green'},  
    orientation='horizontal'  
)
```

```
In [99]: widgets.BoundedIntText(  
    value=7,  
    min=0,  
    max=10,  
    step=1,  
    description='Text:',  
    disabled=False  
)
```

```
In [100... widgets.Checkbox(  
    value=False,  
    description='Check me',  
    disabled=False,  
    indent=False  
)
```

```
In [101... widgets.RadioButton(  
    options=['pepperoni', 'pineapple', 'anchovies'],  
    value='pineapple', # Defaults to 'pineapple'  
    description='Pizza topping:',  
    disabled=False  
)
```

```
In [103... w=widgets.Dropdown(  
    options=[('One', 1), ('Two', 2), ('Three', 3)],  
    value=2,  
    description='Number:',  
)  
  
display(w)
```

```
In [104... widgets.Text(  
    value='Hello World',  
    placeholder='Type something',  
    description='Data:',  
    disabled=False  
)
```

```
In [55]: widgets.Textarea(  
    value='Hello World',  
    placeholder='Type something',  
    description='String:',  
    disabled=False  
)
```

```
In [106... widgets.Password(  
    value='',  
    description='Password:',  
    disabled=False  
)
```

```
value='password',  
placeholder='Enter password',  
description='Password:',  
disabled=False  
)
```

<https://ipywidgets.readthedocs.io/en/latest/examples/Widget%20List.html>