GUI

Lokesh Payasi

Introduction

- Python provides various options for developing Graphical User Interfaces (GUIs):

- **Tkinter:-** Tkinter is the Python interface to the Tk GUI toolkit shipped with Python.
- **wxPython:-** This is an open-source Python interface for wxWindows.
- **JPython:-** JPython is a Python port for Java which gives Python scripts access to Java class libraries on the local machine.

Tkinter

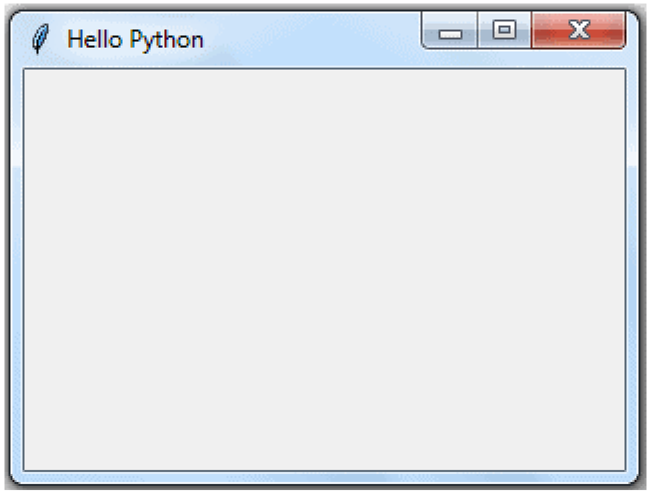
- Tkinter is the Python port for **Tcl-Tk GUI toolkit** developed by Fredrik Lundh. This module is bundled with standard distributions of Python for all platforms.
- PyQtis, the Python interface to Qt, is a very popular cross-platform GUI framework.
- PyGTK is the module that ports Python to another popular GUI widget toolkit called GTK.
- WxPython is a Python wrapper around WxWidgets, another cross-platform graphics library.

First Box- Basic GUI Application

```
from tkinter import *
window=Tk()
# add widgets here

window.title('Hello Python')
window.geometry("300x200+10+20")
window.mainloop()
```

output



Button- Button(window, attributes)

```
from tkinter import *
window=Tk()
btn=Button(window, text="Python Advanced
Course", fg='blue')
btn.place(x=250, y=150)
window.title('Hello Python')
window.geometry("500x500+10+10")
window.mainloop()
```

Button-CheckButton(master, option=value)

```
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()
```

Frame: Frame(master, option=value)

```
•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()
```

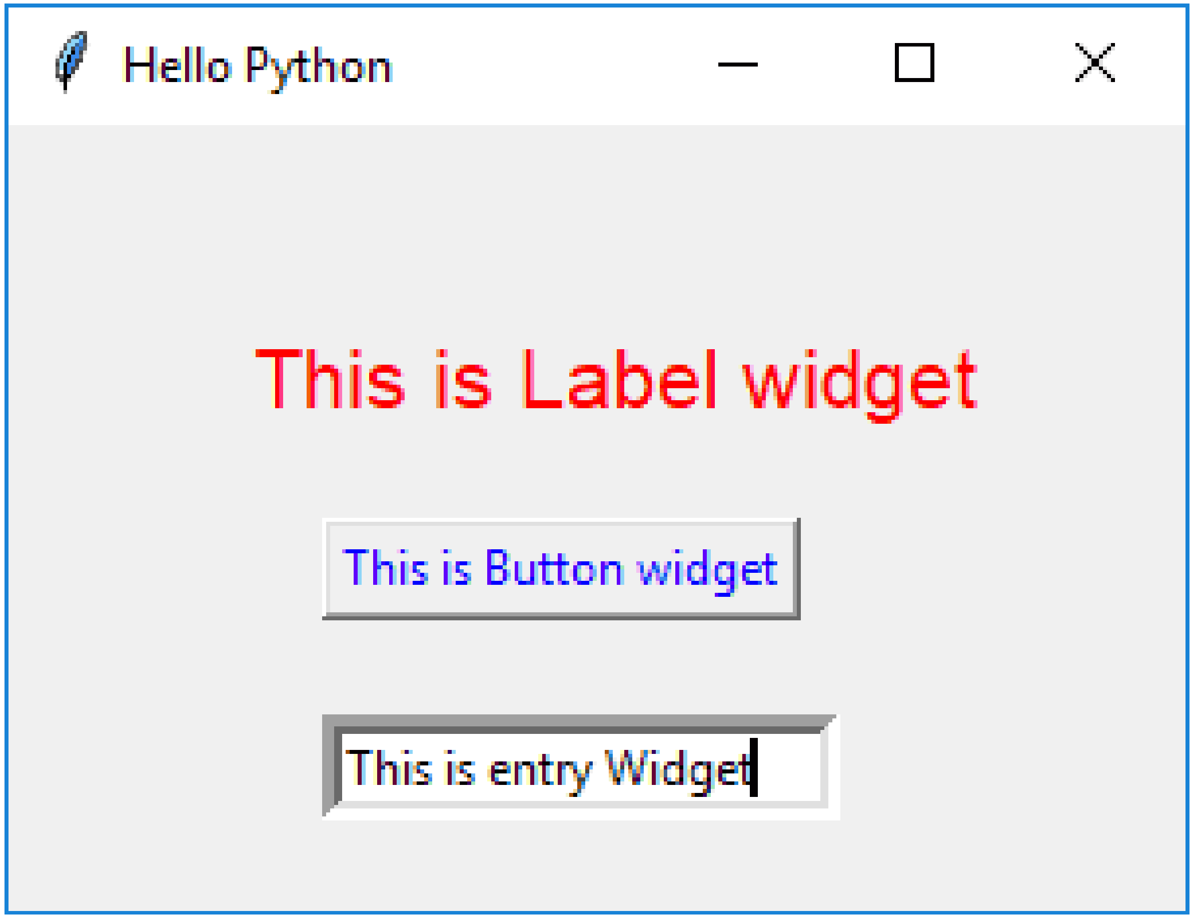
Label

```
•from tkinter import *
•window=Tk()
•lbl=Label(window, text="This is Label widget",
fg='red', font=("Helvetica", 16))
•lbl.place(x=60, y=50)
•window.title('Hello Python')
•window.geometry("300x200+10+10")
•window.mainloop()
```

Entry

txtfld=Entry(window, text="This is Entry Widget", bg='black',fg='white', bd=5)

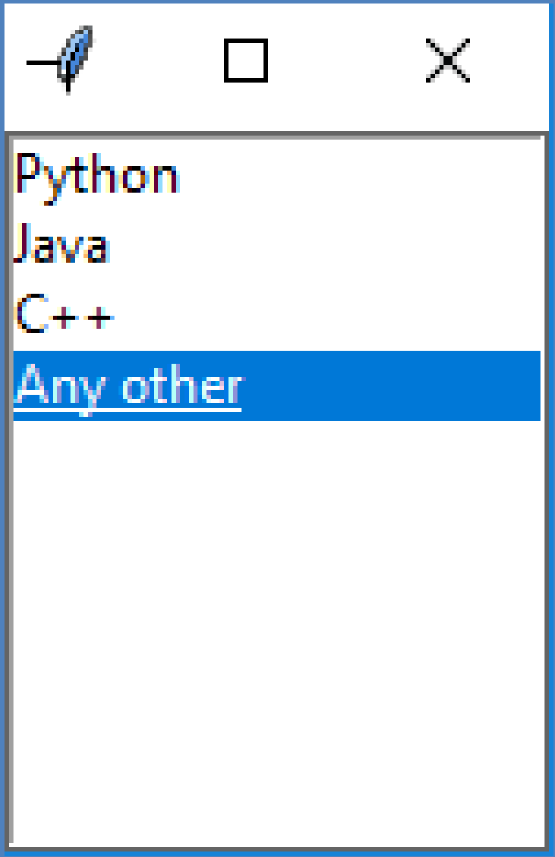
- from tkinter import *
- window=Tk()
- btn=Button(window, text="This is Button widget", fg='blue')
- btn.place(x=80, y=100)
- lbl=Label(window, text="This is Label widget", fg='red', font=("Helvetica", 16))
- lbl.place(x=60, y=50)
- txtfld=Entry(window, text="This is Entry Widget", bd=5)
- txtfld.place(x=80, y=150)
- window.title('Hello Python')
- window.geometry("300x200+10+10")
- window.mainloop()


Hello Python
This is Label widget
This is Button widget
This is entry Widget

Listbox

```
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()
```



```
Python
Java
C++
Any other
```

MenuButton

MenuButton(master, option=value)

```
from tkinter import *

top = Tk()
mb = Menubutton ( top, text = "LOK")
mb.grid()
mb.menu = Menu ( mb, tearoff = 0 )

mb.pack()
top.mainloop()
```

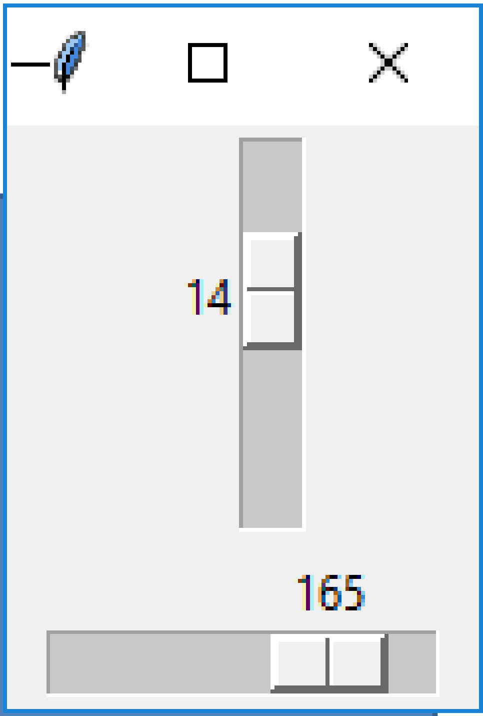
Menu It is used to create all kinds of menus used by the application.

```
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()
```

Scale

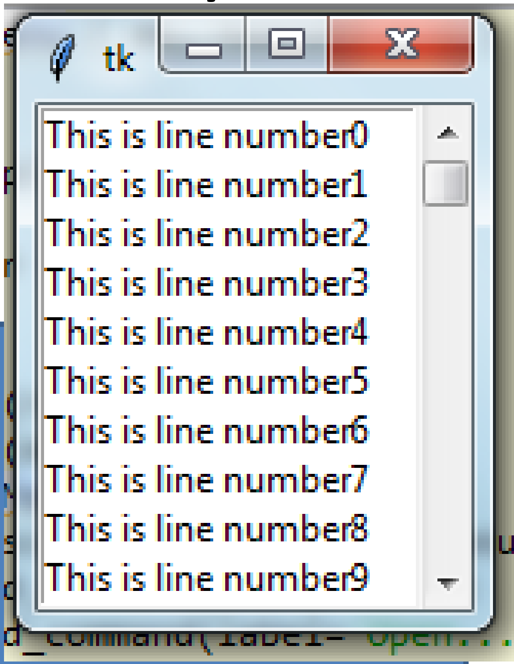
Scale(master, option=value)

```
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()
```



Scrollbar

Scrollbar(master, option=value)



tk

```
This is line number0
This is line number1
This is line number2
This is line number3
This is line number4
This is line number5
This is line number6
This is line number7
This is line number8
This is line number9
```

```
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()
```

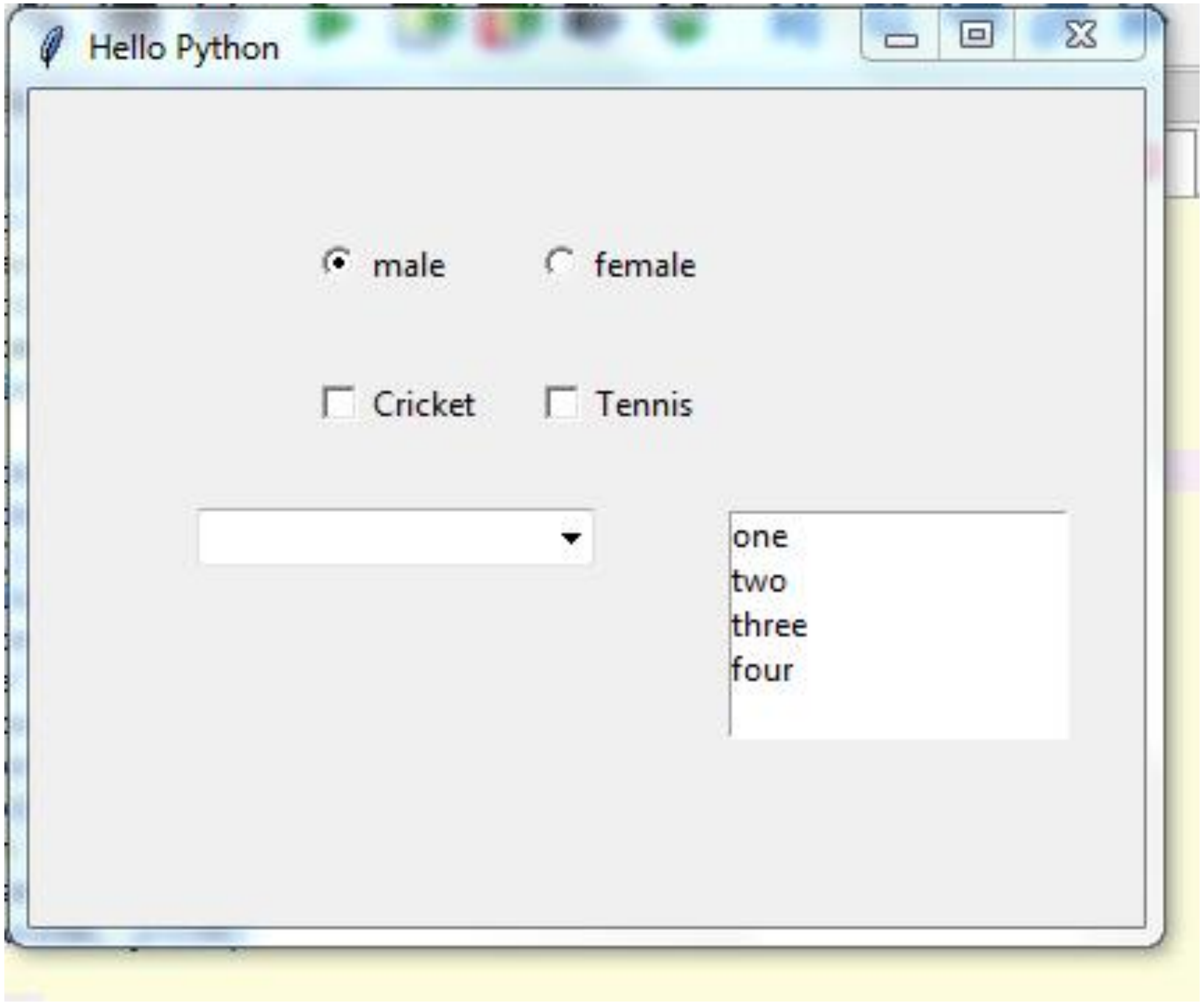
Selection Widgets

```
from tkinter import *
from tkinter.ttk import Combobox
window=Tk()
var = StringVar()
var.set("one")
data=("one", "two", "three", "four")
cb=Combobox(window, values=data)
cb.place(x=60, y=150)
lb=Listbox(window, height=5, selectmode='multiple')
for num in data:
    lb.insert(END,num)
lb.place(x=250, y=150)
```

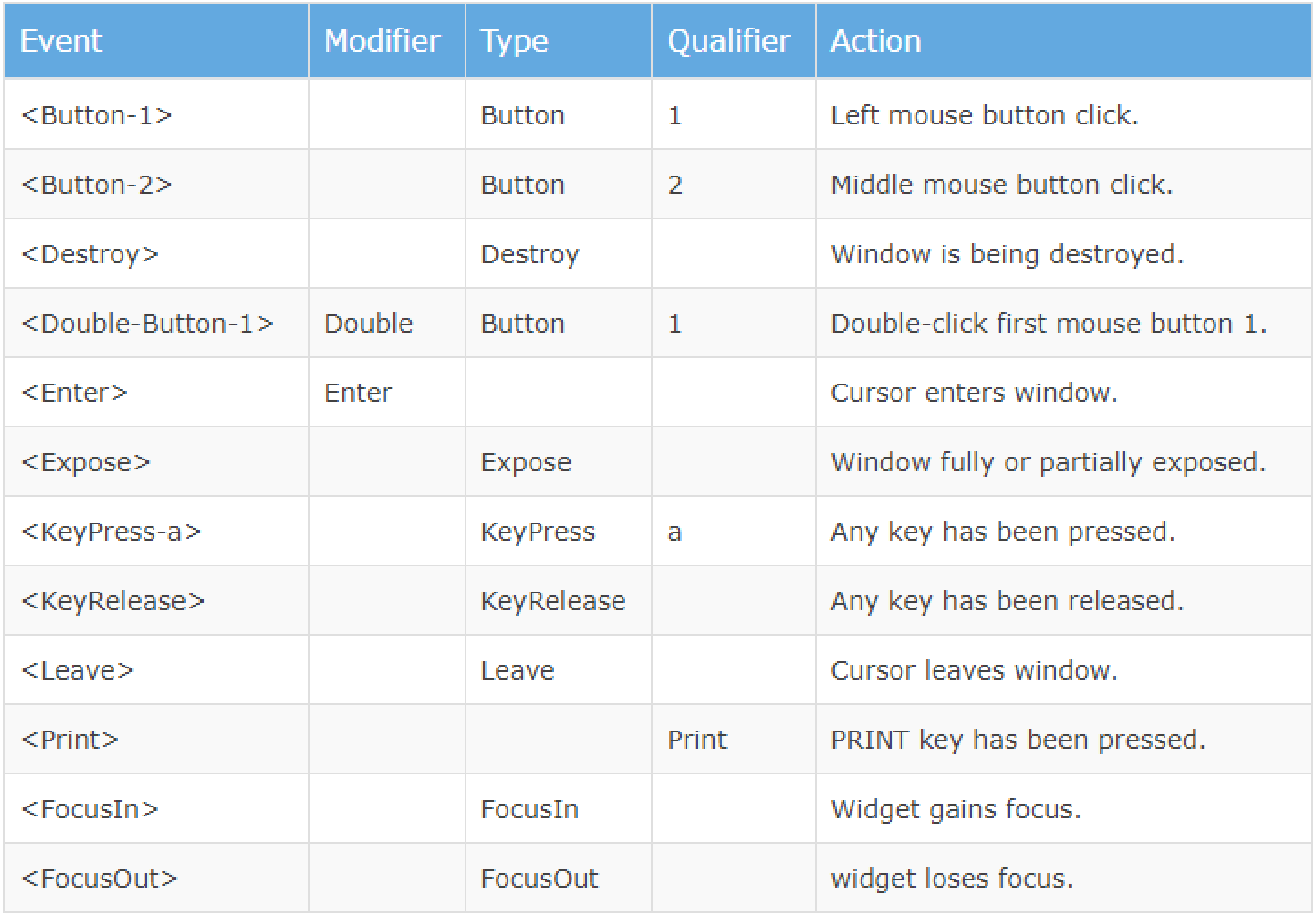
Selection Widget

- v0=IntVar()
- v0.set(1)
- r1=Radiobutton(window, text="male", variable=v0,value=1)
- r2=Radiobutton(window, text="female", variable=v0,value=2)
- r1.place(x=100,y=50)
- r2.place(x=180, y=50)
- v1 = IntVar()
- v2 = IntVar()
- C1 = Checkbutton(window, text = "Cricket", variable = v1)
- C2 = Checkbutton(window, text = "Tennis", variable = v2)
- C1.place(x=100, y=100)
- C2.place(x=180, y=100)
- window.title('Hello Python')
- window.geometry("400x300+10+10")
- window.mainloop()

Output



Event Handling



Event	Modifier	Type	Qualifier	Action
<Button-1>		Button	1	Left mouse button click.
<Button-2>		Button	2	Middle mouse button click.
<Destroy>		Destroy		Window is being destroyed.
<Double-Button-1>	Double	Button	1	Double-click first mouse button 1.
<Enter>	Enter			Cursor enters window.
<Expose>		Expose		Window fully or partially exposed.
<KeyPress-a>		KeyPress	a	Any key has been pressed.
<KeyRelease>		KeyRelease		Any key has been released.
<Leave>		Leave		Cursor leaves window.
<Print>			Print	PRINT key has been pressed.
<FocusIn>		FocusIn		Widget gains focus.
<FocusOut>		FocusOut		widget loses focus.

Bind() Method- Widget.bind(event, callback)

- The bind() method associates an event.
- The event object is characterized by many properties such as source widget.

Let’s Make a Calculator Command Parameter

- btn = Button(window, text='OK', command = myEventHandlerFunction)

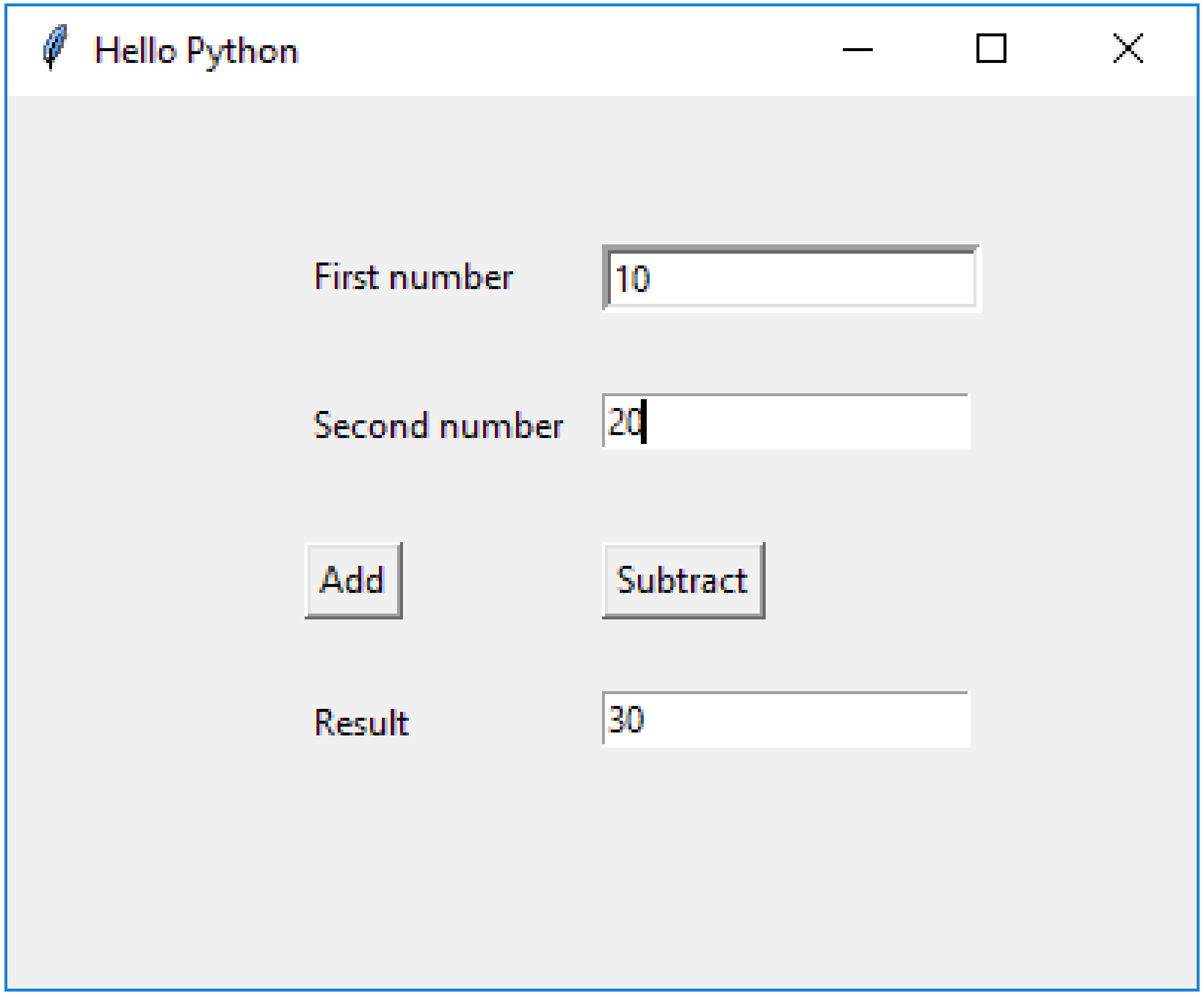
Codes

```
class MyWindow:
    def __init__(self, win):
        self.lbl1=Label(win, text='First number')
        self.lbl2=Label(win, text='Second number')
        self.lbl3=Label(win, text='Result')
        self.t1=Entry(bd=3)
        self.t2=Entry()
        self.t3=Entry()
        self.btn1 = Button(win, text='Add')
        self.btn2=Button(win, text='Subtract')
        self.lbl1.place(x=100, y=50)
        self.t1.place(x=200, y=50)
        self.lbl2.place(x=100, y=100)
        self.t2.place(x=200, y=100)
        self.b1=Button(win, text='Add', command=self.add)
        self.b2=Button(win, text='Subtract')
        self.b2.bind('<Button-1>', self.sub)
        self.b1.place(x=100, y=150)
        self.b2.place(x=200, y=150)
        self.lbl3.place(x=100, y=200)
        self.t3.place(x=200, y=200)
```

Codes cont..

```
def add(self):
      self.t3.delete(0, 'end')
      num1=int(self.t1.get())
      num2=int(self.t2.get())
      result=num1+num2
      self.t3.insert(END, str(result))
   def sub(self, event):
      self.t3.delete(0, 'end')
      num1=int(self.t1.get())
      num2=int(self.t2.get())
      result=num1-num2
      self.t3.insert(END, str(result))
```

The output


Hello Python
First number
10
Second number
20
Add
Subtract
Result
30

Canvas: Canvas(master, option=value)

```
from tkinter import *
master = Tk()
w = Canvas(master, width=40, height=60)
w.pack()
canvas_height=20
canvas_width=200
y = int(canvas_height / 2)
w.create_line(0, y, canvas_width, y )
mainloop()
```

Conclusion