GUI Programming

Graphical User Interfaces

A graphical user interface allows the user to interact with the operating system and other programs using graphical elements such as icons, buttons, and dialog boxes.

Using the tkinter Module

In Python you can use the tkinter module to create simple GUI programs.

Tkinter=Tk interface

Table 13-1 `tkinter` Widgets

Widget	Description
`Button`	A button that can cause an action to occur when it is clicked.
`Canvas`	A rectangular area that can be used to display graphics.
`Checkbutton`	A button that may be in either the "on" or "off" position.
`Entry`	An area in which the user may type a single line of input from the keyboard.
`Frame`	A container that can hold other widgets.
`Label`	An area that displays one line of text or an image.
`Listbox`	A list from which the user may select an item
`Menu`	A list of menu choices that are displayed when the user clicks a `Menubutton` widget.
`Menubutton`	A menu that is displayed on the screen and may be clicked by the user
`Message`	Displays multiple lines of text.
`Radiobutton`	A widget that can be either selected or deselected. `Radiobutton` widgets usually appear in groups and allow the user to select one of several options.
`Scale`	A widget that allows the user to select a value by moving a slider along a track.
`Scrollbar`	Can be used with some other types of widgets to provide scrolling ability.
`Text`	A widget that allows the user to enter multiple lines of text input.
`Toplevel`	A container, like a `Frame`, but displayed in its own window.

Window displayed by Program

Tkinter module

```
import tkinter
def main():
  main_window=tkinter.Tk() #creates an instance ofthe tkinter module's Tk class and assigns it to the
                           main_window variable
  main_window.mainloop()
main()
```

By Class approach we also create the window

```
import tkinter
class MyGUI:
  def __init__(self):
    self.main_window=tkinter.Tk()
```

```
        tkinter.mainloop()
Myob=MyGUI()
```

Display the label widget

```
import tkinter
class MyGUI:
    def __init__(self):
        self.main_window=tkinter.Tk()
        self.label=tkinter.Label(self.main_window, text='hello world')
        self.label.pack() # call the label widget call method, it uses where label is displayed
        tkinter.mainloop()
Myob=MyGUI()
```

Organizing widget with frames

```
import tkinter
class MyGUI:
    def __init__(self):
        self.main_window=tkinter.Tk()
        self.top_frame=tkinter.Frame(self.main_window)
        self.bottom_frame=tkinter.Frame(self.main_window)
        self.label=tkinter.Label(self.top_frame, text='hello world')
        self.label1=tkinter.Label(self.bottom_frame, text='hello world')
        self.label.pack(side='top')
        self.label1.pack(side='left')
        self.top_frame.pack(side='left')
        self.bottom_frame.pack()
        tkinter.mainloop()
Myob=MyGUI()
```

Button Widget

You use the Button widget to create a standard button in a window. When the user clicks a button, a specified function or method is called. An info dialog box is a simple window that displays a message to the user and has an OK button that dismisses the dialog box. You can use the tkinter.messagebox module's show info function to display an info dialog box.

```
import tkinter
import tkinter.messagebox
class MyGUI:
    def __init__(self):
        self.main_window=tkinter.Tk()
        self.top_frame=tkinter.Frame(self.main_window)
        self.bottom_frame=tkinter.Frame(self.main_window)
        self.label=tkinter.Label(self.top_frame, text='hello world')
        self.my_button=tkinter.Button(self.top_frame,text='click me', command=self.do_something)
```

```
        self.label1=tkinter.Label(self.bottom_frame, text='hello world')
        self.my_button.pack()
        self.label.pack(side='top')
        self.label1.pack(side='left')
        self.top_frame.pack(side='left')
        self.bottom_frame.pack()
        tkinter.mainloop()
    def do_something(self):
        tkinter.messagebox.showinfo('Response', 'Thanks for clicking the button.')
Myob=MyGUI()
```

Get Input with the Entry Widget

An Entry widget is a rectangular area that the user can type input into. You use the Entry widget's get method to retrieve the data that has been typed into the widget

```
import tkinter
import tkinter.messagebox
class MyGUI:
    def __init__(self):
        self.main_window=tkinter.Tk()
        self.top_frame=tkinter.Frame(self.main_window)
        self.bottom_frame=tkinter.Frame(self.main_window)
        self.label=tkinter.Label(self.top_frame, text='enter the kilometers')
        self.my_button=tkinter.Button(self.top_frame,text='click me', command=self.do_something)
        self.label1=tkinter.Label(self.bottom_frame, text='hello world')
        self.kilo_entry=tkinter.Entry(self.top_frame, width=10)
        self.my_button.pack()
        self.label.pack(side='top')
        self.kilo_entry.pack(side='top')
        self.label1.pack(side='left')
        self.top_frame.pack(side='left')
        self.bottom_frame.pack()
        tkinter.mainloop()
    def do_something(self):

        kilo=float(self.kilo_entry.get())
        miles=kilo*0.624
        tkinter.messagebox.showinfo('Response', 'here the miles are' +str(miles))
Myob=MyGUI()
```

Creating the Quit Button

```
self.quit_button=tkinter.Button(self.top_frame,text='Quit', command=self.main_window.destroy)
```

Create the program to enter the user value through the entry widget and also uses the quit frame

Using labels as output field

```
import tkinter
import tkinter.messagebox
class MyGUI:
   def __init__(self):
      self.main_window=tkinter.Tk()
      self.top_frame=tkinter.Frame(self.main_window)
      self.bottom_frame=tkinter.Frame(self.main_window)
      self.value=tkinter.StringVar()
      # We need a StringVar object to associate with
       # an output label. Use the object's set method
       # to store a string of blank characters.
      self.label=tkinter.Label(self.top_frame, text='enter the kilometers')
      self.my_button=tkinter.Button(self.top_frame,text='click me', command=self.do_something)
      self.quit_button=tkinter.Button(self.top_frame,text='Quit', command=self.main_window.destroy)

      self.label1=tkinter.Label(self.bottom_frame, text='hello world')
      self.label2=tkinter.Label(self.top_frame, textvariable=self.value)
      # Value presented as label value

      self.kilo_entry=tkinter.Entry(self.top_frame, width=10)
      self.my_button.pack()
      self.quit_button.pack()
      self.label.pack(side='top')

      self.kilo_entry.pack(side='top')
      self.label1.pack(side='left')
      self.label2.pack(side='right')
      self.top_frame.pack(side='left')
      self.bottom_frame.pack()
      tkinter.mainloop()
   def do_something(self):

      kilo=float(self.kilo_entry.get())
      miles=kilo*0.624
      self.value.set(kilo)  # specify the value to show as label
      tkinter.messagebox.showinfo('Response', 'here the miles are' +str(miles))
Myob=MyGUI()
```