1. import tkinter-to import tkinter

in tkinter everything (GUI) is a widget.

1. <window\_name>=Tk()-used for creating a window /root widget.

Two steps in tkinter are first define the thing and then dispay it on the screen

1. <labelname>=Label(<name of window you want to display on>,text=<text you want to display>,anchor=[E or W],bd=<number>,relief=SUNKEN,font=(‘<font name>’,<size>))-label widget is used to define the text you want to display in a certain window,anchor used to set text at east or west,bd adds border and relief gives sunken effect.
2. <label\_name>.pack()-it just put the text on the window .
3. <window\_name>.mainloop()-creates a loop to track of movement
4. <label\_name>.grid(row=<number row>,column=<number column>,columspan=<number of colum to take>,rowspan=<number of row you want to take>,padx=<number>,pady=<number>,sticky={N,[+]E,[+]W,[+]S})-it prints take in given row and column note it make column and row as big as highest text it print if there is no text for a column/row it just size it zero. Padx and pady just give pixel space between it next objects.sticky stretches it to the specified direction
5. <buttonname>=Button(<window>,text=<text of button>[,padx=<breadth>,pady=<length>, state=DISABLED,fg=<text color>,bg=<button color>,command=<function name>],font=(‘<font name>’,<size>))-used to create button command do the function you define when click button(don’t use paranthesis) and state disables the button. If command is <window>.quit then on clicking the button the program ends. <window>.destroy closes the program
6. <name>=Entry(<window>,fg=<text color>,bg=<background color>,width=<number>,borderwidth=<number>,font=(‘<font name>’,<size>))-print a box to input data.
7. <name>.get()-gets the input data in entry box.
8. <name>.insert(0,<text>)-print the text in input box.
9. <window>.title(<text>)-gives title to window.
10. <window>.iconbitmap(<location of image>)-makes image as icon
11. To add image-

Ensure you have pillow module in the system

Code-

from tkinter import \*

from PIL import ImageTk,Image

<window>=Tk()

<image>=ImageTk.PhotoImage(Image.open('<image name>'))

<label>=Label(<window>,image=<image>)

<label>.pack()

1. <name>.grid\_forget()-removes all the items in existing row and column to avoid overlap.
2. <entryname>.delete(0,END)-delete number from entry box
3. <frame>=LabelFrame(<window>,[text=<string>[,padx=<num>,pady=<num>]])

<frame>.pack/grid()-this creates a frame to divide screen for controlling things. To add widgets inside frame just name the frame instead of window .

1. Radiobutton-

These are round buttons for giving values to the tkinter to make them write-

<name>=Radiobutton(<window>,text=<str>,variable=<tkvar>,value=<value to be given to var>,command=<function>,padx=<num>,pady=<num>,bg=<color>,fg=<color>,font=(‘<font name>’,<size>))

<name>.grid/pack()-

For radiobutton special predefined variables are needed.to create them use-

<tkvar>= IntVar() or StringVar()- for string or int type value ensure that value that button gave is of type given, also is same variable is used in n buttons then out of them only one could be clicked at a time, otherwise for different variable or button can be clicked.

To get the value use –

<tkvar>.get()

And to set a value use-

<tkvar>.set(<value>)

{command=lambda:<function>(<value>)-use this for giving value}

1. To create messagebox –

From tkinter import messagebox-to import the function

There are 6 types of message box-

showinfo,showerror,showwarning,askquestion,askokcancel,askyesno

<variable>=messagebox.showwarning(<title>,<message>)-gives a message box , returns ok if button pressed if closed then return ok

<variable>=messagebox.showerror(<title>,<message>)-gives an error box,returns ok if button pressed if closed then return ok

<variable>=messagebox.showinfo(<title>,<message>)-gives info box, returns ok if button pressed if closed then return ok

<variable>=messagebox.askquestion(<title>,<message>)-gives a yes no box, returns yes if yes pressed and no if no pressed

<variable>=messagebox.askokcancel(<title>,<message>)-gives an ok cancel box, returns True if ok pressed and False if cancel pressed if closed then return False

<variable>=messagebox.askyesno(<title>,<message>)-gives an yes no box, returns True if yes pressed and False if no pressed

1. <2window>=Toplevel()-used for creating a second window.
2. To get the location of a file interactively use-

from tkinter import filedialog

<window>.filename=filedialog.askopenfilename(initialdir=<location to start search>,title=<title to search box>,filetypes=((<name displayed>,<type>),(<name displayed>,<type>)..))-minimum two arguments in filetypes an use \*.<extension> for every file of that extension and \*.\* for all files of all extension.

1. <window>geometry(‘<horizontal length>x<vertical length>’)-resize your screen.
2. <slidername>=Scale(<window>,from\_=<start>,to=<end>[,orient=HORIZONTAL], width=<number>,bg=<color>,fg=<color>,bd=<number>)-makes a glider vertical default.

<slidername>.get()-use to get current value of slider

1. To create checkboxes-

<variable>=IntVar() or StringVar()-for creating an int or string variable

<name>=Checkbutton(<window>,variable=<variable>[,onvalue=<value>,offvalue=<value>,command=<function>],padx=<num>,pady=<num>,bg=<color>,fg=<color>,font=(‘<font name>’,<size>))-makes a check box of onvalue and off value are given the it gives onvalue on on and offvalue on off

To get value use-<variable>.get()

Note=by default int variable have value 0 and string variable as ‘’. Now int check button are on only on 1 value if onvalue is not given and off at other .If onvalue given then on at that value only, but string variable are on at onvalue(if given otherwise default one) and ‘’ and off at other.

By default for int onvalue is 1 and off value is 0 and for string it is ‘1’ for on and ‘0’ for off.

1. To crete drop box-

<variable>=IntVar() or StringVar()-for creating an int or string variable

<name>=OptionMenu(<window>,<variable>,<value1>,...)-create a dropbox in which you can choose one value and it is shown and assigned to variable .to enhance readability create ist of value and use \*<list> instead of <value1>...

Use <variable>.get() to get a value.