```
#OPENING A WINDOW

import tkinter

top=tkinter.Tk()

top.mainloop()
```

```
#ADDING A TITLE

from tkinter import *

~~#master=Tk()~~

master=Tk(className="Welcome")

mainloop()
```

```
#ARC

import tkinter

top = tkinter.Tk()

C = tkinter.Canvas(top, bg="blue", height=250, width=300)

coord = 10, 50, 240, 210

arc = C.create_arc(coord, start=0, extent=150, fill="red")

C.pack()

top.mainloop()
```

This line imports the tkinter module, allowing you to create graphical user interfaces in Python.

Here, you create a new instance of the Tk class, representing the main window of your tkinter application

This line creates a canvas widget (C) inside the top window. A canvas is a rectangular area that you can use to draw shapes, text, and images. In this case, the canvas has a blue background (bg="blue"), a height of 250 pixels, and a width of 300 pixels.

coord variable represents a tuple of four values: (x1, y1, x2, y2),

his line creates an arc on the canvas C using the specified coordinates (coord). The start=0 parameter indicates that the arc starts from the 0-degree angle, and extent=150 specifies that the arc should cover a 150-degree angle. The fill="red" parameter sets the fill color of the arc to red

This line packs the canvas widget into the top window. The pack() method organizes widgets in blocks before placing them in the parent widget.

Finally, this line starts the tkinter event loop (mainloop()), allowing the application to respond to user interactions.

```
#rectangle
from tkinter import *
master = Tk()
w = Canvas(master, width=200, height=100,bg="green")
w.pack()
w.create_rectangle(10, 20, 150, 80, fill="blue")
mainloop()
```

```
#oval
##
from tkinter import *
master = Tk()
w = Canvas(master, width=200, height=200,bg="blue")
w.pack()
w.create_oval(10, 10, 80, 180, outline="#f11",fill="#1f1", width=2)
mainloop()
```

```
#polygon
from tkinter import *
master = Tk()
w = Canvas(master, width=400, height=400,bg="blue")
w.create_polygon(150,175,260,100,200,250,300,200,100,150,outline='red', fill='green', width=2)
w.pack()
mainloop()
```

```
#line
from tkinter import *
master = Tk()
w = Canvas(master, width=200, height=100)
w.pack()
w.create_line(15, 25, 200, 25)
mainloop()
```

```
#button
from tkinter import *
top = Tk()
top.geometry("1400x1000")
b = Button(top,text = "Register",fg="red",activeforeground="blue",activebackground="pink")
b.place(x=100,y= 200)

b1 = Button(top,text = "cancel",fg="red",activeforeground="blue",activebackground="pink")
b1.grid(row=200,column=200)
mainloop()
```

```
#checkbutton
from tkinter import *
top = Tk()
top.geometry("300x300")

checkvar1 = IntVar()
checkvar2 = IntVar()
checkvar3 = IntVar()

chkbtn1 = Checkbutton(top, text = "Cpp", variable = checkvar1, onvalue = 1, offvalue = 0, height = 1, width = 3)
chkbtn2 = Checkbutton(top, text = "C++", variable = checkvar2, onvalue = 1, offvalue = 0, height = 1, width = 10)
chkbtn3 = Checkbutton(top, text = "Java", variable = checkvar3, onvalue = 1, offvalue = 0, height = 1, width = 10)

chkbtn1.pack()
chkbtn2.pack()
chkbtn3.pack()

top.mainloop()
##

#entry,label

from tkinter import *
top = Tk()
top.geometry("400x250")

email = Label(top, text = "Email").place(x = 30, y = 90)
name = Label(top, text = "Name").place(x = 30,y = 60)
```

```
password = Label(top, text = "Password").place(x = 30, y = 120)


sbmitbtn = Button(top, text = "Register",activebackground = "pink", activeforeground = "blue").place(x =
30, y = 170)


e1 = Entry(top).place(x = 80, y = 60)

e2 = Entry(top).place(x = 80, y = 90)

e3 = Entry(top).place(x = 95, y = 120)



top.mainloop()





#frame


from tkinter import *


top = Tk()

top.geometry("540x500")


frame1 = Frame(top,bg='green',bd=50)

frame1.pack()


leftframe = Frame(top,bg='yellow',bd=30)  #bd = size of border around indicator

leftframe.pack(side = LEFT)


rightframe = Frame(top,bg='orange',bd=160)

rightframe.pack(side = RIGHT)
```

```
btn1 = Button(frame1, text="Submit", fg="red",activebackground = "red")

btn1.pack(side = LEFT)


btn2 = Button(frame1, text="Remove", fg="brown", activebackground = "brown")

btn2.pack(side = RIGHT)


btn3 = Button(rightframe, text="Add", fg="green", activebackground = "black")

btn3.pack(side = LEFT)


btn4 = Button(leftframe, text="Modify", fg="black", activebackground = "white")

btn4.pack(side = RIGHT)

top.mainloop()
```

```
#listbox
```

```
from tkinter import *
```

This line imports all classes, functions, and variables from the tkinter module, which is a standard Python interface to the Tk GUI toolkit.

```
top = Tk()
```

This line creates the main window of the application. Tk() is a class within the tkinter module that creates a top-level window

```
top.geometry("200x250")
```

This line sets the initial dimensions of the window to 200 pixels in width and 250 pixels in height.

```
lbl = Label(top,text = "A list of favourite countries...")
```

This line creates a label widget with the text "A list of favourite countries..." and associates it with the top window. Labels are used to display text or images.

```
lis1 = Listbox(top)
```

This line creates a listbox widget and associates it with the top window. A listbox is a widget that allows the user to select one or more items from a list.

```
lis1.insert(1,"India")
```

This line inserts the string "India" at position 1 in the listbox.

```
lis1.insert(2, "USA")
```

This line inserts the string "USA" at position 2 in the listbox.

```
lis1.insert(3, "Japan")
```

This line inserts the string "Japan" at position 3 in the listbox.

```
lis1.insert(4, "Austrelia")
```

This line inserts the string "Australia" at position 4 in the listbox.

```
lbl.pack()
```

This line organizes the label widget within the window. The pack() method is used to automatically fit the widget into the parent window

```
lis1.pack()

top.mainloop()
```

This line starts the main event loop of the Tkinter application. It continuously listens for events such as button clicks, keypresses, etc., and updates the GUI accordingly. The program will run indefinitely until the user closes the window

```
#menu

from tkinter import *

top = Tk()


def hello1():
    print("hellooo!")
```

This line defines a function named hello1 that prints "hellooo!" to the console when called.

```
menubar = Menu(top)
```

This line creates a menu bar (Menu widget) that will be attached to the top window. Menu bars are used to create menus in a GUI application

```
menubar.add_command(label="Hello!",command=hello1)
```

This line adds a menu item labeled " Hello!" to the menu bar. When the "Hello!" menu item is clicked, it will execute the hello1 function, which prints "hellooo!" to the console

```
menubar.add_command(label="Quit!", command=top.destroy)
```

This line adds another menu item labeled "Quit!" to the menu bar. When the "Quit!" menu item is clicked, it will execute the top.destroy method, which closes the main window and terminates the application

```
# display the menu

top.config(menu=menubar)
```

This line associates the menubar (the created menu bar) with the top window. It sets the menu bar for the top window, so the menu items will be displayed within the window.

```
top.mainloop()
```

```
#menubutton

from tkinter import *

top = Tk()

top.geometry("200x250")

menubutton = Menubutton(top,text = "Language", relief = RIDGE )  #relief -enable 3d effect to the menubutton
```

This line creates a Menubutton widget. A Menubutton is a button that, when clicked, drops down a menu. The relief=RIDGE argument provides a 3D effect to the button.

```
menubutton.pack()
```

This line organizes the Menubutton widget within the window

```
menubutton.menu = Menu(menubutton)    #to create new menu
```

This line creates a new Menu widget and associates it with the Menubutton widget. The menu will be attached to the Menubutton and displayed when the button is clicked

```
menubutton['menu']=menubutton.menu    #['menu']:-menu specified in the menubutton
```

This line sets the menu attribute of the Menubutton widget to the Menu widget created in the previous line. This associates the menu with the Menubutton

```
check1=IntVar()
```

This line creates an IntVar variable, which is used to hold integer values, and will be associated with the first check button

```
check2=IntVar()

menubutton.menu.add_checkbutton(label = "Hindi", variable=check1) #adding checkbutton to menu
```

This line adds a check button labeled "Hindi" to the menu. When this check button is selected, the check1 variable will be set to 1.

```
menubutton.menu.add_checkbutton(label = "English", variable = check2)
```

This line adds another check button labeled "English" to the menu. When this check button is selected, the check2 variable will be set to 1.

```
menubutton.pack()

top.mainloop()
```

```
#radio button

from tkinter import *

def selection():
  sel = "You selected the option " + radio.get()
  label.config(text = sel)
```

This defines a function named selection which will be called when a radio button is selected. This function sets the text of the label widget based on the radio button selected.

```
top = Tk()
```

This creates the main window of the application

```
top.geometry('300x150')
```

This sets the initial dimensions of the window to be 300 pixels wide and 150 pixels tall

```
radio = StringVar()
```

his creates a StringVar variable. StringVar is a special variable in Tkinter that can hold a string value. It will be associated with the selected radio button.

```
lbl = Label(text = "Favourite programming language:")
```

This creates a label widget with the specified text

```
lbl.pack()
```

his organizes the label widget within the window

```
R1 = Radiobutton(top, text="C", variable=radio, value="C",
            command=selection)
```

This creates a radio button labeled "C". The variable parameter is set to the radio variable, and the value parameter is set to "C". The command parameter specifies the function to be called when this radio button is selected.

```
R1.pack( anchor = N )
```

This organizes the first radio button within the window, anchoring it to the north (top) side of the available space

```
R2 = Radiobutton(top, text="C++", variable=radio, value="C++",
            command=selection)
```

This creates a second radio button labeled "C++" with similar parameters to the first radio button.

```
R2.pack( anchor = E )
```

This organizes the second radio button within the window, anchoring it to the east (right) side of the available space.

```
R3 = Radiobutton(top, text="Java", variable=radio, value="Java",
            command=selection)
```

his creates a third radio button labeled "Java" with similar parameters to the previous radio buttons.

```
R3.pack( anchor = W)
```

This organizes the third radio button within the window, anchoring it to the west (left) side of the available space

```
label = Label(top)
```

This creates another label widget that will be used to display the selection made by the user.

```
label.pack()
```

This organizes the label widget within the window.

```
top.mainloop()
```