**TUTORIAL1.**

**PYTHON GUI**

**USER INTERFACE USED TO MAKE APPLICATION DESKTOP.BOTH WEB SERVER AND CLIENT SERVER.BASED APLICATIONS CAN BE MADE.**

* **WX PYTHON ,QT,JPYTHON,KIVY ,TKINTER ETC.. ARE USED TO MAKE APPLICATIONS**
* **PACKAGING AND DISTRIBUTIONS IS DONE THEN**

**EX ADOBE PHOTOSHOP,CHROME SOFTWARES CAN BE MADE AND CAN BE SOLD.**

**TK IS A WIDGET TOOL**

**TKINTER IS A INTERFACE FOR PYTHON.**

**Import tkinter**

**Or**

**From tkinter import \***

**Root is a tk class instance**

**This makes a basic GUI**

**Naresh\_root = Tk()**

**Event loop**

**Naresh\_root.mainloop()**

**GUI ia a program which uses api to display buttons menu.. tkinter is a cross platform gui**

**ATTRIBUTES ARE IN TKINTER TO DISPLAY BUTTONS.**

**TO PACK ALL THESE WIDGETS WE USE PACK,GRID ,PLACE**

**\*PACK PLACES BLOCKWISE ALL WIDGETS**

**\*GRID MAKES GRID LIKE EXCEL SHEET**

**\*.PLACE METHOD PLACES UR WIDGET IN A SPECIF POSITION.**

**TUTORIAL 4**

**LABELS,GEOMETRY,MAXSIZE,MINSIZE**

**THESE ARE TO QUICKSTART APPLICATION**

1. **LABEL IS A WIDGET WITH AHICH USER DOES NOT INTERACT**
2. **.geometry(“”) IS A FUNCTION WIDTHxHEIGHT IS PARAMETER**
3. **.minsize(200,100) minimum size it can become. width,height is argument**
4. **.maxsize(200,100) maximum size it can become. width,height is argument**
5. **Var=Label(text=”naresh is a good boy”)**
6. **Var.pack() We should pack label**

**TUTORIAL 5**

**DISPLAYING IMAGES USING LABEL**

**TO ADD PHOTO**

**SYNTAX: caps p i**

**Var=PhotoImage(file=”1.png”)**

**ADD IT TO LABEL**

**Var2=Label(image=var)**

**PACT IT**

**Var2.pack()**

**Pip install pillow in terminal:**

**Pillow python image in library**

1. **from PIL import Image,ImageTk**
2. **Image Tk helps to add import jpeg files**
3. **FOR JPEG IMAGES FIRST OPEN IMAGES**
4. **Var=Image.open(“photo.jpg”)**
5. **ADD IMAEG TO TK**
6. **Var1=ImageTk.PhotoImage(var)**
7. **LABEL IT**
8. **Var3=Label(image=var1)**
9. **PACK IT.var3.pack()**

**TUTORIAL 6**

**ATTRIBUTES OF LABEL AND PACK**

**\*TO CHANGE TITLE**

**Var.title(“my gui”)**

**2.IMPORTANT LABEL OPTIONS**

* **text=adds text to gui user does not interact**
* **image=adds image to gui png**
* **Bg or bd= background**
* **fg=foreground**
* **font-sets the font**
* **padx= xpadding seen in html**
* **pady = ypadding**
* **borderwidth-chnage border size**
* **relief -border styling-SUNKEN,RAISED,GROOVE,RIDGE**

1. **IMPORTANT PACK ATTRIUTES**

* **anchor = nw,ne,se,sw**
* **side =left,right,top,bottom**
* **fill = X,Y**
* **padx,pady =100,200**

from tkinter import \*  
from PIL import Image,ImageTk  
naresh=Tk()  
*# gui attributes of label  
# font="Timesroman 12 italic"  
# font=("TimesRoman""15""bold")*naresh.geometry(**"700x500"**)  
title\_label=Label(text=**'''In 1976, if you had told fourteen-year-old Franciscan seminary student Thomas Cruise Mapother IV**\n **that one day in the not too distant future he would be Tom Cruise, one of the top 100 movie**\n**stars of all time, he would have probably grinned and told you that his ambition was to join**\n**the priesthood. Nonetheless, this sensitive, deeply religious youngster who was born in 1962 in Syracuse,**\n**New York, was destined to become one of the highest paid and most sought after actors in screen history'''**,bg=**"red"**,  
fg=**"white"**,padx=20,pady=200,font=(**"TimesRoman""15""bold"**),borderwidth=10,relief=GROOVE)  
*# title\_label.pack(side="bottom",anchor="se",fill="x")*title\_label.pack(side=**"left"**,fill=**"y"**,padx=**"100"**,pady=**"50"**)  
  
naresh.mainloop()

**SOURCE CODE AND CHEAT SHEET**

**<https://www.codewithharry.com/videos/python-gui-tkinter-hindi-6>**

* **Attributes:** A set of properties of a widget that defines its visual appearance on the computer screen and how it responds to user events. Here we’ll discuss about the **attributes** of Label and Pack.
* **Attributes of Label:** The **Label**widget is a standard Tkinter widget used to display a text or image on the screen.  There are a lot of attributes of Label widget. Some important attributes are discussed below:
  1. **bg:** The normal background color displayed behind the label and indicator.
  2. **fg:**This option specifies the color of the text (foreground color). If you are displaying a bitmap, this is the color that will appear at the position of the 1-bits in the bitmap.
  3. **padx:** Extra space added to the left and right of the text within the widget. **Default is 1.**
  4. **pady:** Extra space added above and below the text within the widget. **Default is 1.**
  5. **relief:** Specifies the appearance of a decorative border around the label. There are five types of reliefs, such that FLAT, RAISED, SUNKEN, GROOVE, RIDGE. **The default is FLAT.**
  6. **font:** If you are displaying text in this label (with the text or textvariable option), the font option specifies the style, size and other characteristics (i.e. bold, italic etc.) of the font and in this style the text will be displayed.
  7. **text:** To display one or more lines of text in a label widget, set this option to a string containing the text. Internal newlines (“\n”) will force a line break.
  8. **justify:** Specifies how multiple lines of text will be aligned with respect to each other: **LEFT** for flush left, **CENTER**for centered (**the default**), or **RIGHT** for right-justified.
  9. **height:** The vertical dimension of the new frame.
  10. **width:** The horizontal dimension of the new frame. If this option is not set, the label will be sized to fit its contents.
* **Attributes of Pack:**The Pack geometry manager packs widgets in rows or columns. We can use options like **fill**, **expand**, and **side**to control this geometry manager.
  1. **fill:**Determines whether widget fills any extra space allocated to it by the packer, or keeps its own minimal dimensions: NONE (default), X (fill only horizontally), Y (fill only vertically), or BOTH (fill both horizontally and vertically).
  2. **side:**Determines which side of the parent widget packs against: TOP, BOTTOM, LEFT, or RIGHT. **The default is TOP.**
  3. **anchor:** Anchors are used to define where text is positioned relative to a reference point. The anchor attributes are: n, s, e, w, center, nw, sw, ne and se. **The default is center.**

**TUTORIAL 8**

**FRAMES IN TKINTER**

**FRAMES ARE LIKE PREPLAN FOR MAKING AGUI.LIKE DIV TAGS SPAN WE CREATE FIRST IN HTML,THEN WE FILL WITH COLORS LATER WITH CODE.LIKE THAT PREDESIGN.**

**SYNTAX.**

**Var=Frame(root,bg=”red”,borderwidth=6)**

**Root is where frame is packed.**

1. **then GIVE POSITION TO FRAME**

**Var.pack(side= LEFT)**

1. **TO WRITE INSIDE A FRAME**

**L=Label(var,text=”hello mars”)**

**Here var is a frame ,inside frame we write hello mars.**

1. **PACK THE FRAME**
2. **pack()**

  
A **Frame** widget is a rectangular region on the screen which can be used as a foundation class to implement complex widgets. This widget is very important for the process of grouping and organizing other widgets in a friendly way. It works like a container, which is responsible for arranging the position of other widgets. We can use bg, relief, borderwidth, bd as the attributes of Frame widget. Some important attributes are discussed below:

**bg:** The normal background color displayed behind the label and indicator.

**relief:** The type of the border of the frame. **Its default value is set to FLAT**. We can set it to any other styles i.e. FLAT, RAISED, SUNKEN, GROOVE, RIDGE.

**borderwidth:** **Tkinter**Label **doesn't have any border by default**. We need to assign the borderwidth option to add a border around Label widget along with the relief option to be any option rather than flat to make visible.

**bd:** The size of the border around the indicator. **Default is 2 pixels**.

**from tkinter import \***

**root = Tk()**

**root.geometry("655x333")**

**f1 = Frame(root, bg="grey", borderwidth=6, relief=SUNKEN)**

**f1.pack(side=LEFT, fill="y")**

**f2 = Frame(root, borderwidth=8, bg="grey", relief=SUNKEN)**

**f2.pack(side=TOP, fill="x")**

**l = Label(f1, text="Project Tkinter - Pycharm")**

**l.pack( pady=142)**

**l = Label(f2, text="Welcome to sublime text", font="Helvetica 16 bold", fg="red", pady=22)**

**l.pack()**

**root.mainloop()**

**TUTORIAL 9**

**PACKING BUTTONS IN TKINTER**

1. **FIRST CREATE FRAME**

**F1=Frame(root,fg=”red”,bd=5)**

**F1.pack(side=LEFT)**

1. **CREATE BUTTON**

**B1=Button(f1,fg=”red”,text=”printnow”,command=hello)**

**B1.pack(side=LEFT)**

**Command is a for a prg,hello is a function u previously made.**

**TUTORIAL 10:**

**ENTRY WIDGETS AND GRID LAYOUT IN TKINTER.**

* **Entry()** **widgets** are the basic widgets of Tkinter which is used to get input, i.e. text strings, from the user of an application. This widget allows the user to enter a single line of text.

Note: **If the user enters a string, which is longer than the available display space of the widget, the content will be scrolled.**This means that the string cannot be seen in its entirety. The arrow keys can be used to move to the invisible parts of the string. **If you want to enter multiple lines of text, you have to use the text widget. An entry widget is also limited to single font.**

* The **Grid()**manager is the most flexible of the geometry managers in Tkinter. The Grid() geometry manager puts the widgets in a 2-dimensional table. The master widget is split into a number of rows and columns, and each “cell” in the resulting table can hold a widget.

Note:**We can also create layouts using Pack() manager**butit takes a number of extra frame widgets, and a lot of work to make things look good. If you use the grid manager instead, you only need one call per widget to get everything laid out properly.

**from tkinter import \***

**def getvals():**

**print(f"The value of username is {uservalue.get()}")**

**print(f"The value of password is {passvalue.get()}")**

**root = Tk()**

**root.geometry("655x333")**

**user = Label(root, text="Username")**

**password = Label(root, text="Password")**

**user.grid()**

**password.grid(row=1)**

**# Variable classes in tkinter**

**# BooleanVar, DoubleVar, IntVar, StringVar**

**uservalue = StringVar()**

**passvalue = StringVar()**

**userentry = Entry(root, textvariable = uservalue)**

**passentry = Entry(root, textvariable = passvalue)**

**userentry.grid(row=0, column=1)**

**passentry.grid(row=1, column=1)**

**Button(text="Submit", command=getvals).grid()**

**root.mainloop()**

**TUTORIAL 11**

**TRAVEL FORM USING CHECK BOX AND ENTRY WIDGET**

from tkinter import \*  
root=Tk()  
  
root.geometry(**"400x500"**)  
def getval():  
print(**"its working"**)  
  
Label(text=**"welcome to canteen"**,font=**"sans-serif 15 bold"**,pady=**"10"**,bg=**"yellow"**).grid(row=0,column=3)  
Label(text=**"name"**).grid(row=1,column=0)  
Label(text=**"number"**).grid(row=2,column=0)  
Label(text=**"gender"**).grid(row=3,column=0)  
Label(text=**"age"**).grid(row=4,column=0)  
Label(text=**"mealname"**).grid(row=5,column=0)  
*# entry names variables*nameentry=StringVar()  
numberentry=StringVar()  
genderentry=StringVar()  
ageentry=StringVar()  
mealnameentry=StringVar()  
foodorder=IntVar()  
*# add entryes*Entry(textvariable=nameentry).grid(row=1,column=3)  
Entry(textvariable=numberentry).grid(row=2,column=3)  
Entry(textvariable=genderentry).grid(row=3,column=3)  
Entry(textvariable=ageentry).grid(row=4,column=3)  
Entry(textvariable=mealnameentry).grid(row=5,column=3)  
*# checkbutton and pack*Checkbutton(text=**"book now"**,variable=foodorder).grid(row=6,column=3)  
*# sunbit button and grid*Button(text=**"submit"**,command=getval).grid(row=7,column=3)  
root.mainloop()

**TUTORIAL 12**

**ACCEPTING USER INPUT FROM USER**

**from tkinter import \*  
root=Tk()  
  
root.geometry("400x500")  
def getval():  
 print("its SUBMITTING")  
 print(f"{nameentry.get(),numberentry.get(),genderentry.get(),ageentry.get(),mealnameentry.get(),foodorder.get()}")  
  
 with open("records.txt","a") as f:  
 f.write(f"{nameentry.get(),numberentry.get(),genderentry.get(),ageentry.get(),mealnameentry.get(),foodorder.get()}\n")  
  
Label(text="welcome to canteen",font="sans-serif 15 bold",pady="10",bg="yellow").grid(row=0,column=3)  
Label(text="name").grid(row=1,column=0)  
Label(text="number").grid(row=2,column=0)  
Label(text="gender").grid(row=3,column=0)  
Label(text="age").grid(row=4,column=0)  
Label(text="mealname").grid(row=5,column=0)  
# entry names variables  
nameentry=StringVar()  
numberentry=StringVar()  
genderentry=StringVar()  
ageentry=StringVar()  
mealnameentry=StringVar()  
foodorder=IntVar()  
# add entryes  
Entry(textvariable=nameentry).grid(row=1,column=3)  
Entry(textvariable=numberentry).grid(row=2,column=3)  
Entry(textvariable=genderentry).grid(row=3,column=3)  
Entry(textvariable=ageentry).grid(row=4,column=3)  
Entry(textvariable=mealnameentry).grid(row=5,column=3)  
# checkbutton and pack  
Checkbutton(text="book now",variable=foodorder).grid(row=6,column=3)  
# sunbit button and grid  
Button(text="submit",command=getval).grid(row=7,column=3)  
root.mainloop()**

**TUTORIAL 13 CANVAS WIDGET IN PYTHON TKINTER**

When it comes to Graphical User Interface based application, image(s) play a vital role. From the application icon to animation, it's useful.

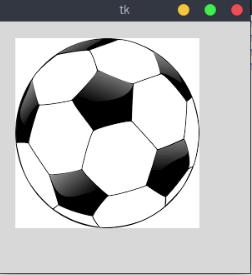
To display images in labels, buttons, canvases, and text widgets, the PhotoImage class is used, which is present in tkinter package.

**Example Code**

1. **from** tkinter **import** \*
2. root = Tk()
3. canvas = Canvas(root, width = 300, height = 300)
4. canvas.pack()
5. img = PhotoImage(file="ball.ppm")
6. canvas.create\_image(20,20, anchor=NW, image=img)
7. mainloop()

"PhotoImage()" function returns the image object.

**Output**



To display image in Python is as simple as that. But, the problem is PhotoImage class only supports GIF and PGM/PPM formats.

The more generalized formats are JPEG/JPG and PNG. To open and display with those formats, we need help of ImageTk and Image classes from PIL(photo imaging Library) package.

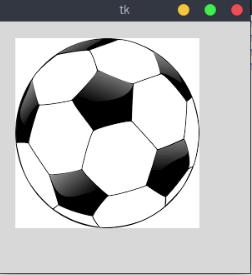
With the help of PIL(photo imaging Library), we can load images in over 30 formats and convert them to image objects, even base64-encoded GIF files from strings!!

**Example Code**

1. **from** tkinter **import** \*
2. **from** PIL **import** ImageTk,Image
3. root = Tk()
4. canvas = Canvas(root, width = 300, height = 300)
5. canvas.pack()
6. img = ImageTk.PhotoImage(Image.open("ball.png"))
7. canvas.create\_image(20, 20, anchor=NW, image=img)
8. root.mainloop()

Image class has an attribute "open()" which would open the not-so-directly-supported image formats and along with "ImageTk.PhotoImage()", we can return the image object and use it.

**Output**



**Note**

If you display an image inside a function, then make sure to keep reference to the image object in your Python program, either by storing it in a global variable or by attaching it to another object.

1. **global** img
2. **def** foo(self):
3. img = ImageTk.PhotoImage(Image.open("ball.png"))
4. **def** foo(self):
5. self.img = ImageTk.PhotoImage(Image.open("ball.png"))
6. self.canvas.create\_image(20,20, anchor=NW, image=self.img)
7. self.canvas.image = self.img

It's important because when you return from the function and if the image object is stored in a variable local to that function, the image is cleared by the garbage collector even if it's displayed by tkinter.

**TUTORIAL 14**

**HANDING EVENTS IN TKINTER**

from tkinter import \*  
from PIL import Image,ImageTk  
root =Tk()  
def hello(event):  
print(**f"u touched click me at** {event.x}**,**{event.y}**"**)  
def sizes(event):  
 root.geometry(**"900x800"**)  
root.title(**"events"**)  
root.geometry(**"500x400"**)  
widget=Button(root,text=**"click me"**)  
widget.pack()  
widget.bind(**'<Motion>'**,hello)  
widget.bind(**'<Leave>'**,sizes)  
root.mainloop()

**Events types in tkinter**

**<https://www.python-course.eu/tkinter_events_binds.php>**

**TUTORIAL 15**

**NEWSPAPER SOLUTION**

**from tkinter import \***

**from PIL import ImageTk, Image**

**def every\_100(text):**

**final\_text = ""**

**for i in range(0, len(text)):**

**final\_text +=text[i]**

**if i%100==0 and i!=0:**

**final\_text += "\n"**

**return final\_text**

**root = Tk()**

**root.title("CodeWithHarry News - Aapka Apna Akhabaar")**

**root.geometry("1000x1000")**

**texts = []**

**photos = []**

**for i in range(0, 3):**

**with open(f"{i+1}.txt") as f:**

**text = f.read()**

**texts.append(every\_100(text))**

**image = Image.open(f"{i+1}.png")**

**#TODO: Resize these images**

**image = image.resize((225, 265), Image.ANTIALIAS)**

**photos.append(ImageTk.PhotoImage(image))**

**f0 = Frame(root, width=800, height=70)**

**Label(f0, text="Code With Harry News", font="lucida 33 bold").pack()**

**Label(f0, text="January 19, 2050", font="lucida 13 bold").pack()**

**f0.pack()**

**f1 = Frame(root, width=900, height=200, pady=14)**

**Label(f1, text=texts[0], padx=22, pady=22).pack(side="left")**

**Label(f1, image=photos[0], anchor="e").pack()**

**f1.pack(anchor="w")**

**f2 = Frame(root, width=900, height=200, pady=14, padx=22)**

**Label(f2, text=texts[1], padx=22, pady=22).pack(side="right")**

**Label(f2, image=photos[1], anchor="e", padx=22).pack()**

**f2.pack(anchor="w")**

**f3 = Frame(root, width=900, height=200, pady=34)**

**Label(f3, text=texts[2], padx=22, pady=22).pack(side="left")**

**Label(f3, image=photos[2], anchor="e").pack()**

**f3.pack(anchor="w")**

**root.mainloop()**

**ANOTHER SOLUTION SAME RESULT**

from tkinter import \*  
from PIL import ImageTk, Image  
def every\_100(text):  
 final\_text = **""**for i in range(0, len(text)):  
 final\_text +=text[i]  
if i%100==0 and i!=0:  
 final\_text += **"**\n**"**return final\_text  
  
  
  
  
root = Tk()  
root.title(**"CodeWithHarry News - Aapka Apna Akhabaar"**)  
root.geometry(**"1000x1000"**)  
  
  
texts = []  
photos = []  
for i in range(0, 3):  
with open(**f"**{i+1}**.txt"**) as f:  
 text = f.read()  
 texts.append(every\_100(text))  
  
 image = Image.open(**f"**{i+1}**.jpg"**)  
*#TODO: Resize these images*image = image.resize((225, 265), Image.ANTIALIAS)  
 photos.append(ImageTk.PhotoImage(image))  
  
f0 = Frame(root, width=800, height=70)  
Label(f0, text=**"Code With Harry News"**, font=**"lucida 33 bold"**).pack()  
Label(f0, text=**"January 19, 2050"**, font=**"lucida 13 bold"**).pack()  
f0.pack()  
  
*# f=[]*for i in range(0,3):  
 f= Frame(root, width=900, height=200, pady=14)  
 Label(f, text=texts[i], padx=22, pady=22).pack(side=**"left"**)  
 Label(f, image=photos[i], anchor=**"e"**).pack()  
 f.pack(anchor=**"w"**)  
  
*#  
# f2 = Frame(root, width=900, height=200, pady=14, padx=22)  
# Label(f2, text=texts[1], padx=22, pady=22).pack(side="right")  
# Label(f2, image=photos[1], anchor="e", padx=22).pack()  
# f2.pack(anchor="w")  
#  
#  
# f3 = Frame(root, width=900, height=200, pady=34)  
# Label(f3, text=texts[2], padx=22, pady=22).pack(side="left")  
# Label(f3, image=photos[2], anchor="e").pack()  
# f3.pack(anchor="w")*root.mainloop()

**TUTORIAL 16**

**WINDOWS RESIZE TASK**

from tkinter import \*  
from PIL import Image,ImageTk  
def attack(event):  
  
 root.geometry(**f"**{width.get()}**x**{height.get()}**"**)  
root=Tk()  
root.title(**"window resize"**)  
root.geometry(**"900x700"**)  
width=IntVar()  
height=IntVar()  
width\_entry=Entry(root,textvariable=width)  
height\_entry=Entry(root,textvariable=height)  
width\_entry.pack()  
height\_entry.pack()  
resize=Button(root,text=**"apply"**)  
resize.pack()  
resize.bind(**'<Button-1>'**,attack)  
root.mainloop()

**TUTORIAL 17**

**MENUS AND SUBMENUS**

from tkinter import \*  
from PIL import Image,ImageTk  
def myfunc():  
print(**"file saved"**)  
root=Tk()  
root.title(**"menus and submenus"**)  
root.geometry(**"400x500"**)  
mainmenu=Menu(root)  
m1=Menu(mainmenu,tearoff=0)  
m1.add\_command(label=**"open"**,command=myfunc)  
m1.add\_command(label=**"save"**,command=myfunc)  
m1.add\_command(label=**"save as"**,command=myfunc)  
m1.add\_separator()  
m1.add\_command(label=**"exit"**,command=myfunc)  
mainmenu.add\_cascade(label=**"file"**,menu=m1)  
m2=Menu(mainmenu,tearoff=0)  
m2.add\_command(label=**"insert"**,command=myfunc)  
m2.add\_command(label=**"delete"**,command=myfunc)  
m2.add\_command(label=**"change"**,command=myfunc)  
  
mainmenu.add\_cascade(label=**"edit"**,menu=m2)  
  
root.config(menu=mainmenu)  
  
root.mainloop()

**TUTORIAL 18**

**MESSAGE BOX,SHOW INFO,WARNING**

from tkinter import \*  
import tkinter.messagebox as lil  
from PIL import Image, ImageTk  
  
  
def myfunc():  
print(**"file saved"**)  
  
  
def help():  
print(**"narehs will help you"**)  
a = lil.showinfo(**"help"**, **"need help"**)  
  
  
def leave():  
 b = lil.askyesno(**"leave"**, **"wanna leave"**)  
if b == True:  
print(**"u can leave"**)  
else:  
print(**"no u cant leave"**)  
  
  
def close():  
 c = lil.askretrycancel(**"close"**, **"unale to close"**)  
if c == True:  
print(**"try again"**)  
else:  
print(**"cancel please"**)  
  
  
  
  
  
root = Tk()  
root.title(**"menus and submenus"**)  
root.geometry(**"400x500"**)  
def animals():  
x = lil.askyesno(**"question"**, **"which is animal"**)  
 lil.showwarning(**"warning"**,**"u cant ask sillly questions"**)  
 i = lil.askyesnocancel(**"question"**,**"quit app"**)  
if i==True:  
print(**"quitting"**)  
elif i == False:  
print(**"not quitting"**)  
else:  
print(**"you cancelled"**)  
mainmenu = Menu(root)  
m1 = Menu(mainmenu, tearoff=0)  
m1.add\_command(label=**"open"**, command=myfunc)  
m1.add\_command(label=**"save"**, command=myfunc)  
m1.add\_command(label=**"save as"**, command=myfunc)  
m1.add\_separator()  
m1.add\_command(label=**"exit"**, command=myfunc)  
mainmenu.add\_cascade(label=**"file"**, menu=m1)  
m2 = Menu(mainmenu, tearoff=0)  
m2.add\_command(label=**"insert"**, command=myfunc)  
m2.add\_command(label=**"delete"**, command=myfunc)  
m2.add\_command(label=**"change"**, command=myfunc)  
mainmenu.add\_cascade(label=**"edit"**, menu=m2)  
m3 = Menu(mainmenu, tearoff=0)  
m3.add\_command(label=**"help"**, command=help)  
m3.add\_command(label=**"leave"**, command=leave)  
m3.add\_separator()  
m3.add\_command(label=**"close"**, command=close)  
m3.add\_command(label=**"animal"**, command=animals)  
mainmenu.add\_cascade(label=**"quit"**, menu=m3)  
  
root.config(menu=mainmenu)  
  
root.mainloop()

**TUTORIAL 19**

**SLIDERS SCALE ,ORIENT,TICKINTERVAL IN TKINTER**

from tkinter import \*  
import tkinter.messagebox as m  
from tkinter import Scale  
def amount():  
 m.showinfo(**"amount"**,**f"amount** {sliders.get()} **is credited"**)  
root=Tk()  
root.title(**"sliders tutorial"**)  
root.geometry(**"400x500"**)  
Label(root,text=**"how many dollars u want?"**).pack()  
sliders = Scale(root , from\_=**"0"**,to=**"100"**,orient=HORIZONTAL,tickinterval=50)  
sliders.set(20)  
  
sliders.pack()  
Button(root,text=**"credit amount"**,command=amount).pack()  
  
root.mainloop()

**Challenge of rating restaurent**

from tkinter import \*  
import tkinter.messagebox as m  
from tkinter import Scaleroot=Tk()def store():  
with open(**"records.txt"**,**"a"**) as f:  
 f.write(**f"**{myslider.get()}\n**"**)  
 m.showinfo(**"submit"**, **"Thanks for rating our restaurent :)"**)  
 root.quit()  
root.title(**"rate restaurent"**)  
root.geometry(**"400x500"**)Label(root,text=**"please give a rating about your experience "**).pack()  
myslider=Scale(root,from\_=1,to=10,orient=HORIZONTAL)  
myslider.pack()  
Button(root,text=**"post"**,command=store).pack()  
root.mainloop()

**TUTORIAL 20**

**RADIO BUTTONS IN TKINTER**

from tkinter import \*  
from PIL import ImageTk,Image  
import tkinter.messagebox as lil  
from tkinter import Scale  
root=Tk()  
def ordered():  
 lil.showinfo(**"ordered"**,**f"your order** {var.get()} **is placed.Thank you for ordering:)"**)  
 root.quit()  
root.title(**"radio buttons"**)  
root.geometry(**"400x500"**)  
*# var=IntVar()*var=StringVar()  
var.set(**"rabbit"**)  
Label(root,text=**"what u like to order"**,font=**"TimesRoman 14 bold"**,pady=10).pack(anchor=**"n"**)  
radio=[**"dosa"**,**"idly"**,**"samosa"**,**"icecream"**]  
for i,item in enumerate(radio):  
 Radiobutton(root,text=item,variable=var,value=item).pack(anchor=**"w"**)  
Button(root,text=**"order"**,command=ordered).pack(anchor=**"n"**)  
root.mainloop()

**TUTORIAL 21**

**LIST BOX IN TKINTER**

from tkinter import \*  
from PIL import Image,ImageTk  
from tkinter import Scale  
import tkinter.messagebox as ms  
root=Tk()  
def select():  
global i  
 box.insert(ACTIVE,**f"**{i}**"**)  
 i=i+1  
  
i=0  
root.title(**"list boxes"**)  
root.geometry(**"400x500"**)  
box=Listbox(root)  
box.pack()  
box.insert(END,**"what is u want"**)  
box.insert(END,**"what is they want"**)  
Button(root,text=**"select this"**,command=select).pack()  
root.mainloop()

**TUTORIAL 22**

**SCROLL BAR IN TKINTER**

**from tkinter import \***

**root = Tk()**

**root.geometry("455x233")**

**root.title("Scrollbar tutorial")**

**# For connecting scrollbar to a widget**

**# 1. widget(yscrollcommand = scrollbar.set)**

**# 2 scrollbar.config(command=widget.yview)**

**scrollbar = Scrollbar(root)**

**scrollbar.pack(side=RIGHT, fill=Y)**

**listbox = Listbox(root, yscrollcommand = scrollbar.set)**

**for i in range(344):**

**listbox.insert(END, f"Item {i}")**

**listbox.pack(fill="both",padx=22)**

**#text = Text(root, yscrollcommand = scrollbar.set)**

**#text.pack(fill=BOTH)**

**scrollbar.config(command=listbox.yview)**

**#scrollbar.config(command=text.yview)**

**root.mainloop()**

**exercise**

from tkinter import \*  
  
root = Tk()  
root.geometry(**"900x600"**)  
root.title(**"First scrollbar"**)  
scrollbar = Scrollbar(root)  
scrollbar.pack(side=RIGHT,fill=Y)  
text = Text(root,yscrollcommand = scrollbar.set,font=**"lucida 13 italic"**,height=100,fg=**"brown"**)  
text.pack(fill=BOTH)  
scrollbar.config(command=text.yview)  
root.mainloop()

**TUTORIAL 23**

**STATUS BAR IN TKINTER**

from tkinter import \*  
from PIL import Image,ImageTk  
from tkinter import Scale  
import tkinter.messagebox as lil  
root=Tk()  
def uploaded():  
 var.set(**"busy.."**)  
 label.update()  
import time  
 time.sleep(2)  
 var.set(**"Ready now"**)  
root.title(**"statusbar"**)  
root.geometry(**"400x500"**)  
var=StringVar()  
color=StringVar()  
color.set(**"yellow"**)  
var.set(**"Ready"**)  
label=Label(root,textvariable=var,relief=GROOVE,anchor=**"w"**,bg=color.get())  
label.pack(side=BOTTOM,fill=**"x"**)  
Button(root,text=**"upload"**,bg=**"white"**,command=uploaded).pack()  
root.mainloop()

**TUTORIAL 24**

**CLASSES OBJECTS IN TKINTER**

**TO AVOID REPEATING CODE ALWAYS USE OOPS CONCEPTS ,ENCAPSULATION**

from tkinter import \*  
class GUI(Tk):  
def \_\_init\_\_(self):  
super().\_\_init\_\_()  
self.geometry(**"744x500"**)  
def status(self):  
self.var1=StringVar()  
self.var1.set(**"Ready"**)  
self.var=Label(self,textvar=self.var1,relief=GROOVE,anchor=**"w"**)  
self.var.pack(side=BOTTOM,fill=X)  
def click(self):  
print(**"button clicked"**)  
self.var1.set(**"busy.."**)  
import time  
 time.sleep(2)  
self.var1.set(**"ready now"**)  
self.geometry(**"400x200"**)  
def update(self,btext):  
 Button(text=btext,bg=**"pink"**,command=self.click).pack()  
  
  
  
  
if \_\_name\_\_==**"\_\_main\_\_"**:  
 window=GUI()  
 window.status()  
 window.update(**"click me"**)  
 window.mainloop()

**TUTORIAL 24**

**ICONS,QUIT,CONFIGURE**

from tkinter import \*  
from PIL import Image,ImageTk  
from tkinter import Scale  
import tkinter.messagebox as lil  
root=Tk()  
root.title(**"note"**)  
root.geometry(**"400x500"**)  
*# to add icon to application*root.wm\_iconbitmap(**"4.ico"**)  
*# to make changes in middle*root.configure(bg=**"red"**)  
*# laptop screen width and height*width=root.winfo\_screenwidth()  
height=root.winfo\_screenheight()  
print(str(width)+**"x"**+str(height))  
print(**f"**{width}**x**{height}**"**)  
Button(text=**"close"**,bg=**"yellow"**,command=root.quit).pack()  
root.mainloop()

**TUTORIAL 25**

**BUILD A CLACULATOR**

from tkinter import \*  
  
def click(event):  
 global scvalue  
 text = event.widget.cget(**"text"**)  
 if text == **"="**:  
 if scvalue.get().isdigit():  
 value = int(scvalue.get())  
 else:  
 try:  
 value = eval(screen.get())  
  
 except Exception as e:  
 print(e)  
 value = **"Error"** scvalue.set(value)  
 screen.update()  
  
 elif text == **"C"**:  
 scvalue.set(**""**)  
 screen.update()  
  
 else:  
 scvalue.set(scvalue.get() + text)  
 screen.update()  
  
root = Tk()  
root.geometry(**"644x970"**)  
root.title(**"Calculator By CodeWithHarry"**)  
root.wm\_iconbitmap(**"4.ico"**)  
  
scvalue = StringVar()  
scvalue.set(**""**)  
screen = Entry(root, textvar=scvalue, font=**"lucida 40 bold"**)  
screen.pack(fill=X, ipadx=8, pady=10, padx=10)  
  
f = Frame(root, bg=**"grey"**)  
b = Button(f, text=**"9"**, padx=28, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
b = Button(f, text=**"8"**, padx=28, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
b = Button(f, text=**"7"**, padx=28, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
f.pack()  
  
  
f = Frame(root, bg=**"grey"**)  
b = Button(f, text=**"6"**, padx=28, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
b = Button(f, text=**"5"**, padx=28, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
b = Button(f, text=**"4"**, padx=28, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
f.pack()  
  
  
f = Frame(root, bg=**"grey"**)  
b = Button(f, text=**"3"**, padx=28, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
b = Button(f, text=**"2"**, padx=28, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
b = Button(f, text=**"1"**, padx=28, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
f.pack()  
  
  
f = Frame(root, bg=**"grey"**)  
b = Button(f, text=**"0"**, padx=31, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
b = Button(f, text=**"-"**, padx=31, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
b = Button(f, text=**"\*"**, padx=31, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
f.pack()  
  
  
f = Frame(root, bg=**"grey"**)  
b = Button(f, text=**"/"**, padx=33, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
b = Button(f, text=**"%"**, padx=21, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
b = Button(f, text=**"="**, padx=27, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
f.pack()  
  
f = Frame(root, bg=**"grey"**)  
b = Button(f, text=**"C"**, padx=26, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
b = Button(f, text=**"."**, padx=26, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
b = Button(f, text=**"00"**, padx=26, pady=18, font=**"lucida 35 bold"**)  
b.pack(side=LEFT, padx=18, pady=5)  
b.bind(**"<Button-1>"**, click)  
  
f.pack()  
  
root.mainloop()

**TUTORIAL 26**

**BUILD A NOTEPAD**

from tkinter import \*  
from tkinter.messagebox import showinfo  
from tkinter.filedialog import askopenfilename, asksaveasfilename  
import os  
  
def newFile():  
 global file  
 root.title(**"Untitled - Notepad"**)  
 file = None  
 TextArea.delete(1.0, END)  
  
  
def openFile():  
 global file  
 file = askopenfilename(defaultextension=**".txt"**,  
 filetypes=[(**"All Files"**, **"\*.\*"**),  
 (**"Text Documents"**, **"\*.txt"**)])  
 if file == **""**:  
 file = None  
 else:  
 root.title(os.path.basename(file) + **" - Notepad"**)  
 TextArea.delete(1.0, END)  
 f = open(file, **"r"**)  
 TextArea.insert(1.0, f.read())  
 f.close()  
  
  
def saveFile():  
 global file  
 if file == None:  
 file = asksaveasfilename(initialfile = **'Untitled.txt'**, defaultextension=**".txt"**,  
 filetypes=[(**"All Files"**, **"\*.\*"**),  
 (**"Text Documents"**, **"\*.txt"**)])  
 if file ==**""**:  
 file = None  
  
 else:  
 *#Save as a new file* f = open(file, **"w"**)  
 f.write(TextArea.get(1.0, END))  
 f.close()  
  
 root.title(os.path.basename(file) + **" - Notepad"**)  
 print(**"File Saved"**)  
 else:  
 *# Save the file* f = open(file, **"w"**)  
 f.write(TextArea.get(1.0, END))  
 f.close()  
  
  
def quitApp():  
 root.destroy()  
  
def cut():  
 TextArea.event\_generate((**"<>"**))  
  
def copy():  
 TextArea.event\_generate((**"<>"**))  
  
def paste():  
 TextArea.event\_generate((**"<>"**))  
  
def about():  
 showinfo(**"Notepad"**, **"Notepad by Code With Harry"**)  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 *#Basic tkinter setup* root = Tk()  
 root.title(**"Untitled - Notepad"**)  
 root.wm\_iconbitmap(**"4.ico"**)  
 root.geometry(**"644x788"**)  
  
 *#Add TextArea* TextArea = Text(root, font=**"lucida 13"**)  
 file = None  
 TextArea.pack(expand=True, fill=BOTH)  
  
 *# Lets create a menubar* MenuBar = Menu(root)  
  
 *#File Menu Starts* FileMenu = Menu(MenuBar, tearoff=0)  
 *# To open new file* FileMenu.add\_command(label=**"New"**, command=newFile)  
  
 *#To Open already existing file* FileMenu.add\_command(label=**"Open"**, command = openFile)  
  
 *# To save the current file* FileMenu.add\_command(label = **"Save"**, command = saveFile)  
 FileMenu.add\_separator()  
 FileMenu.add\_command(label = **"Exit"**, command = quitApp)  
 MenuBar.add\_cascade(label = **"File"**, menu=FileMenu)  
 *# File Menu ends  
  
 # Edit Menu Starts* EditMenu = Menu(MenuBar, tearoff=0)  
 *#To give a feature of cut, copy and paste* EditMenu.add\_command(label = **"Cut"**, command=cut)  
 EditMenu.add\_command(label = **"Copy"**, command=copy)  
 EditMenu.add\_command(label = **"Paste"**, command=paste)  
  
 MenuBar.add\_cascade(label=**"Edit"**, menu = EditMenu)  
  
 *# Edit Menu Ends  
  
 # Help Menu Starts* HelpMenu = Menu(MenuBar, tearoff=0)  
 HelpMenu.add\_command(label = **"About Notepad"**, command=about)  
 MenuBar.add\_cascade(label=**"Help"**, menu=HelpMenu)  
  
 *# Help Menu Ends* root.config(menu=MenuBar)  
  
 *#Adding Scrollbar using rules from Tkinter lecture no 22* Scroll = Scrollbar(TextArea)  
 Scroll.pack(side=RIGHT, fill=Y)  
 Scroll.config(command=TextArea.yview)  
 TextArea.config(yscrollcommand=Scroll.set)  
  
 root.mainloop()