07/01/2020

PRACTICAL - 5

Aim: GUI components

Algorithm:

Step 1 : Use the tkinter library for importing features of text widget

Step 2 : Create an object using Tk()

Step 3 : Use the pack method along with object created from the text method

Step 4 : Use the mainloop method for triggering of the corresponding events

Step 5 : Use the tkinter library for importing features of text widget

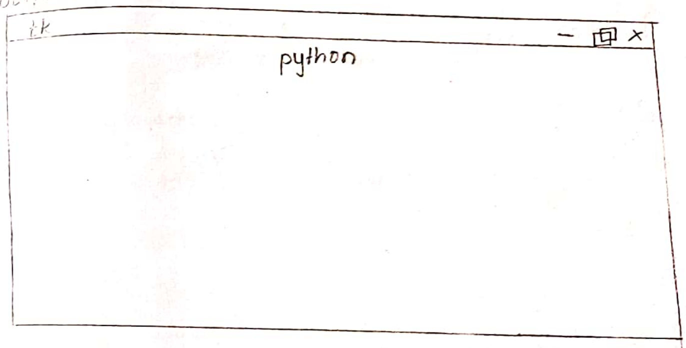
Step 6 : Create a variable from text method and position it onto the parent window

Step 7 : Use the pack method along with object created from the

#a Label Method

```
from tkinter import *
root = Tk()
l = Label(root, text = "python")
l.pack()
root.mainloop()
```

Output :



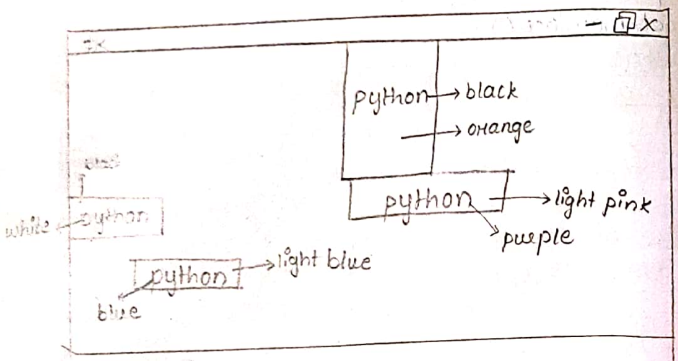
```
tk
python
```

#b

```
from tkinter import *
root = Tk()
l = Label(root, text = "python", bg = "black", fg = "white")
l.pack(side = LEFT, pady = 30)
l1 = Label(root, text = "python', bg = "orange", fg = "black")
l1.pack(side = TOP, ipady = 50)
l2 = Label(root, text = "python", bg = "light blue", fg = "blue")
l2.pack(side = LEFT, padx = 20)
l3 = Labe( root, text = "python", bg = "light pink", fg = "purple")
```

```
l3.pack(side=TOP, ipadx=40)
root.mainloop()
```

Output:



C Radiobutton

```
From tkinter import *
root=Tk()
def sel():
    selection="you selected the option"+str(var.get())
    l=Label.config(root, text="selection", justify=LEFT)
    l.pack(anchor=s)
    v=Intvar().
r1=Radiobutton(text="option 1", variable="v",
    value="1", command=sel).
r1.pack()
r2=Radiobutton(text="option 2", variable="v",
    value="2", command=sel)
```

text method and use the parameter
1. side=LEFT, padx=20 2. side=LEFT, pady=30
3. side=TOP, ipadx=40, 4. side=Top, ipady=50

Step 8: Use the mainloop method for triggering of the corresponding events.

Step 9: Now repeat above steps with label method which takes the following arguments 1. Name of parent window. 2. Text attribute which defines the string 3. The background colour 4. The foreground colour and then use pack method with the relevant padding attributes.

Step 10: Use the tkinter module to import the relevant method

Step 11: Define a function which tells the user about the given selection made from the multiple options available.

Step 12: Use the config method with label object, call the variable as an argument within the method.

Step 13: Now define the parent window and option using the control variable.

Step 14 Now create an object from Radiobutton method which will take arguments positioning on parent window, define the text variable, define the variable argument and trigger given function

Step 15: Now call the pack method for radio object so created and specify the argument as an anchor attribute

Step 16: Now define the label object from corresponding method subsequently use pack method and make use of mainloop method

Step 17: Import relevant methods from tkinter library.

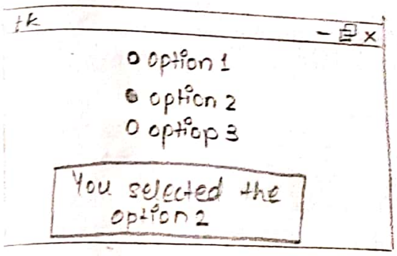
Step 18: Define object corresponding to parent window along with its size

Step 19: Now define the frame object placing it onto parent window. Create another frame object and put it onto: parent window on its left side

Step 20: Similarly define the right frame and subsequently define the button

```
r2.pack()
r3 = Radiobutton(text="option 3", variable="v",
      value="3", command=sel)
r3.pack()
root.mainloop()
```

Output::



○ option 1
● option 2
○ optiop 3

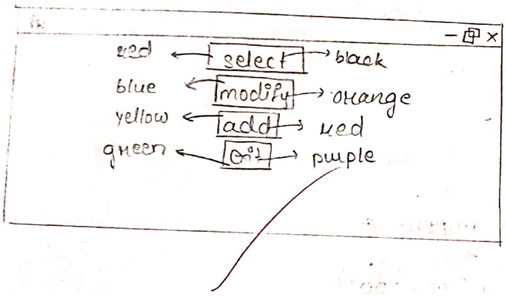
You selected the option 2

Frame object

```
from tkinter import *
top = Tk()
top.geometry('100 x 200')
frame= Frame(top)
frame.pack()
leftframe = Frame(top)
leftframe.pack(side=LEFT)
b1= Button(frame, text="Select", activebackground="red",
     fg=black)
b1.pack()
b2 = Button(frame, text="Modify", activebackground="blue"
     fg-"orange")
```

```
b2.pack()
b3=Button(frame, text="Add", activebackground="yellow",
     fg="red")
b3.pack()
b4=Button(frame, text="Exit", activebackground="green",
     fg="purple")
b4.pack()
top.mainloop()
```

Output:



object placed onto the given frame with the attribute as text activebackground and fg.

Step 21 : Now use pack method along with side attribute. Similarly create the button object corresponding to modify operation and put it into frame object with side='RIGHT' attribute.

Step 22 : Create another button object and place it onto the right and label the button ~~as~~ as Add next add another button and put it onto right frame object and term it as exit.

Step 23 : Use the pack method simultaneously for all the objects and finally use the mainloop method.

21/1

* Message box Method

Step 1: Import the relevant method from tkinter library

Step 2: Define a function and use the message box along with different methods available which contains one or more arguments

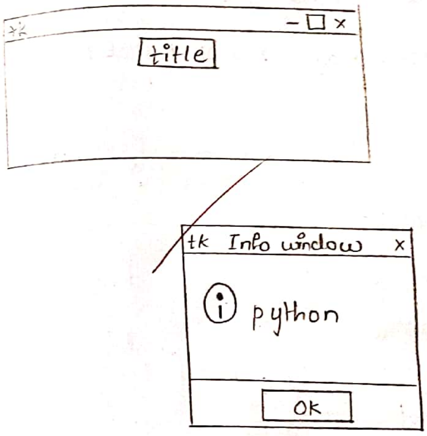
Step 3: Thus different options which are available are showinfo(), showwarning(), showerror(), askYesNo(), askquestion(), askokcancel().

Step 4: Create object from button method and place it onto the parent window with the title of the button specified and the corresponding event called for triggering.

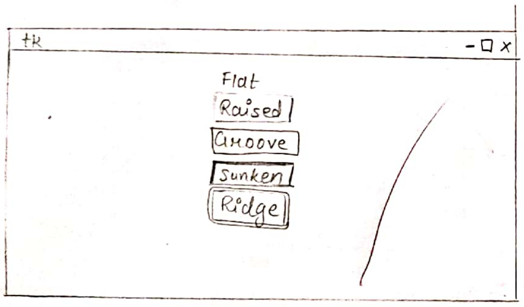
Step 5: Use the pack method to display the button widget and finally use mainloop method.

Step 6: If the user wants to hide the parent window and only the info window should be visible corresponding to the six options given above the withdraw method is used.

```
from Tkinter import *
import tkMessageBox
root = Tk()
def fun():
      tkMessageBox.showinfo("info window", "python")
b1=Button(root, text="title", command=fun)
b1.pack()
root.mainloop()
```



```
from Tkinter import *
root = Tk()
b1=Button (root, text = "Flat", relief = FLAT)
b1.pack()
b2 = Button(root, text="Raised", relief=RAISED)
b2.pack()
b3=Button (root, text= "Groove", relief= GROOVE)
b3.pack()
b4= Button (root, text="Sunken", relief = SUNKEN)
b4.pack()
b5= Button(root, text="Ridge", relief=RIDGE)
b5.pack()
root.mainloop()
```



Relief style

Step 1 : Use the button object with the following attributes
1. The parent window
2. Text attribute
3. Relief

Step 2: Use the corresponding pack method for the respective button objects and trigger the corresponding event

Step 3: Finally use the mainloop method.

- Traversing and making use of geometry layout manager method.

Step 1 : Define a function and create a object of the given window by using the three methods ~~n~~ namely config, title and minsize

Step 2 : Create a button object and use the text and command attribute for triggering the given event and used grid method along with internal and external padding specified similarly create another button object which will allow application to terminate.

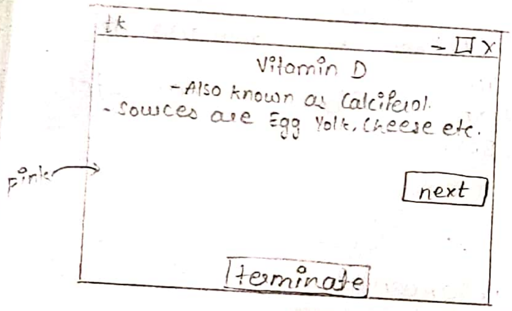
Step 3 : Define second function corresponding to second window with attributes config, title, minsize for the window object and define one button object which will shift the focus onto the third window.

Step 4 : Create third window object and in this create two button object for moving on to first window for restarting the process and second button for terminating.

```
from tkinter import *
root = Tk()
def main():
    roo' = Tk()
    roo.config(bg="pink")
    roo.title("main")
    roo.minsize(200,200)
    l = Label(roo, text="Vitamin D")
    l.pack()
    l1 = Label (roo, text="-Also known as calciferol \n
                 -Sources are Egg Yolk, Cheese etc ")
    l1.pack()
    b1=Button(roo, text="~~seto~~ next", command=se )
    b1.pack(side=RIGHT)
    b2=Button(roo, text="terminate", command=ter)
    b2.pack(side=BOTTOM)
    roo.mainloop()
def se();
    ro = Tk()
    ro.config(bg="green")
    ro.title("2")
    ro.minsize(400,200)
    l2 = Label(ro, text="Vitamin E")
    l2.pack()
    l3 = Label(ro, text="-Also known as Tocopherol \n
                 -Sources are Almonds, Peanuts etc.")
    l3.pack()
    b3=Button(ro, text="back", command=main)
    b3.pack(side=LEFT)
```

Scanned by CamScanner

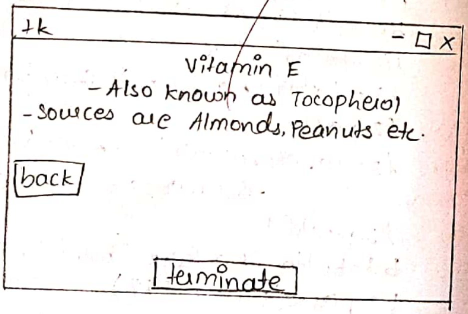
```
        b4 = Button(ro, text= "terminate", command=ter)
        b4.pack(side=BOTTOM)
        ro.mainloop()
def ter():
    quit().

b5 = Button(root, text= "Know Your Vitamins",
        command = main)
b5.pack()
root.mainloop()
```



Vitamin D

– Also known as Calciferol.

- Sources are Egg Yolk, Cheese etc.



Vitamin E

– Also known as Tocopherol

- Sources are Almonds, Peanuts etc.

Step 5 : Define a function for termination and call the quit method and finally call the first function created and trigger mainloop method.

22/7.

* Displaying the image

Algorithm:

Step 1: Create an object corresponding to the parent window and use the following 3 methods. • Title • Maxsize • Config

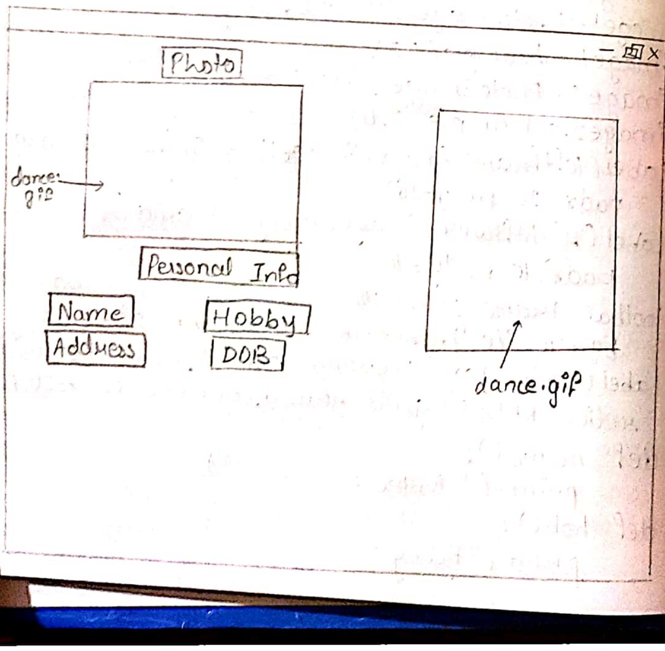
Step 2: Create a leftframe object from the frame method and place it onto the parent window with the height, width and the bg specified. Subsequently use the grid method with the row, column, padx, pady specified

Step 3: Now create a rightframe object from ~~fro~~ the frame method with the width, height specified and the row and the column value should be specified.

Step 4: Create a label object from the label method and place it onto the leftframe with text attribute denoting the original image with relief attribute used as RAISED value and subsequently use grid method with row, column value specified as (0,0) with some external padding values

```
from tkinter import *
root = Tk()
root.title("Python")
root.maxsize(1000,900)
root.config(bg="black")
leftframe = Frame(root, bg="pink", height="400",
    width="200")
leftframe.grid(row=0, column=0)
rightframe = Frame(root, bg="light green", height="400",
        width="250")
rightframe.grid(row=0, column=0)
Label(leftframe, text="Photo", height=2, width=20).
    grid(row=0, column=0)
image1 = PhotoImage(file="dance.gif")
image1.subsample(1,2)
image2 = PhotoImage(file="dance.gif")
image2.subsample(3,4)
Label(leftframe, image=image1).grid(row=0, column=0,
    padx=20, pady=10)
Label(rightframe, image=image2).grid(row=0, column=1,
    padx=10, pady=10)
toolbar = Frame(leftframe, width=200, height=400,
    bg="white").grid(row=2, column=0)
Label(toolbar, text="Personal Info", height=2, width=20,
   relief=RAISED).grid(row=0, column=0, padx=20, pady=10)
def name();
    print(" Name :          ")
def hob():
    print("Hobby :          ")
```

```
def add():
    print("Address: Mumbai")
def dob():
    print("DOB: 24/08/1982")
Button(toolbar,text="Name", height=1, width=16, command
    name).grid(row=1, column=0)
Button(toolbar,text="Hobby", height=1, width=16,
  command=hob).grid(row=1, column=1)
Button(toolbar,text="Address"; height=1, width=16,
   command=add).grid(row=2, column=0)
Button(toolbar, text="DOB", height=1, width=16,
   command=dob).grid(row=2; column=1)
root.mainloop()
```



Step 5: Now use the photo image method with the file attribute specified.

Step 6: Use the sub sample method with the object of the image and give the x, y co-ordinate values.

Step 7: Use the label method and position it onto the left frame and placing the image after the sampling and use the grid method for the positioning in the first row.

Step 8: Create another label object positioning it onto the rightframe and specifying the image and background attribute with row and column attribute specify it as (0,0)

Step 9: Now create a toolbar object from the frame method and position it onto the leftframe with the height and width specified and position it onto the second row.

Step 10: Now define the various function for different tool bar options provided in the leftframe

Step 11: From the label method position the text on the toolbar use the relief attribute and corresponding grid value and incorporate the internal padding as well

Step 12: Create the label method position ~~in~~ it on the toolbar with the next title as personal information and position it on same row but next column

Step 13: Now make use of mainloop method.