Q8 Explain Grid manager of Tkinter with examples.

Ans 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.

Using the grid manager is easy. Just create the widgets, and use the **grid** method to tell the manager in which row and column to place them

The grid manager is especially convenient to use when designing dialog boxes.

9c..widget :Label &Entry

The Entry widget is used **to provde the single line text-box to the user to accept a value from the user**.

#9c1

from tkinter import \*

window = Tk()

window.title("WELCOME TO PYTHON PROG")

lbl = Label(window,text="firstname",bg="red",fg="yellow")

lbl.grid(column=0,row=0)

e1 = Entry(window)

e1.grid(row=0,column=1)

window.mainloop()

OUTPUT:



Q9 Explain ‘sticky’,’columnspan’ and ‘rowspan’ options of ‘Grid’ manager.

Ans The widgets are centered in their cells. We can use the **sticky** option to change this; this option takes one or more values from the set **N**, **S**, **E**, **W**. To align the labels to the left border, we could use **W** (west).

We can also have the widgets span more than one cell. The **columnspan** option is used to let a widget span more than one column, and the **rowspan** option lets it span more than one row.

#9c2

from tkinter import \*

window = Tk()

window.title("Welcome to LikeGeeks app")

lbl = Label(window, text="first name",bg="red",fg="yellow")

lbl.grid(column=0,row=0,columnspan=1,rowspan=1)

lb2 = Label(window, text="last name",bg="red",fg="yellow")

lb2.grid(column=0,row=1,columnspan=2,rowspan=2,sticky=W)

e1 = Entry(window)

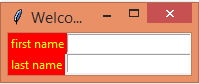
e2 = Entry(window)

e1.grid(row=0,column=1)

e2.grid(row=1,column=1)

window.mainloop()

OUTPUT:



Q10 Explain ‘Place’ manager of Tkinter in python.

Ans The Tkinter place geometry manager allows you to precisely position widgets within its parent window using the (x, y) coordinate system.

To access the place geometry manager, you use the place() method on all the standard widgets like this:

Widget.place(options)

Absolute and relative positions

The place geometry manager provides you with both absolute and relative positioning options.

* Absolute positioning is specified by the x and y options.
* Relative positions is specified by the relx and rely options.

Specifying width and height

To set the absolute width and height of the widget in pixels, you use the width and height options.

The place geometry manager also provides you with relative width and height using the relwidth and relheight options.

The relwidth and relheight has a value of a floating-point number between 0.0 and 1.0. This value represents a fraction of the width and height of the container.

anchor

To specify the exact position of the widget, you use the anchor option.

The value of anchor can be N, E, S, W, NW, SE,or SW.

The anchor defaults to NW which is the upper left corner of the parent container.

#9d pract on Label & Button

#9d1

from tkinter import \*

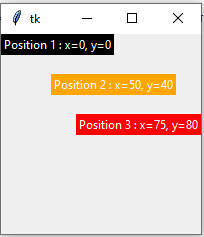
window= Tk()

Label(window,text="Position 1 : x=0, y=0", bg="black", fg="white").place(x=0, y=0)

Label(window, text="Position 2 : x=50, y=40", bg="orange", fg="white").place(x=50, y=40)

Label(window, text="Position 3 : x=75, y=80", bg="red", fg="white").place(x=75, y=80)

window.mainloop()



Q11Using place layout manager of Tkinter Create GUI application with Label and clicked button in python.

Ans #9d2

from tkinter import\*

window = Tk()

window.title("Welcome")

Label(window,text="Position 1 : x=0, y=0", bg="black", fg="white").place(x=0, y=0)

def clicked():

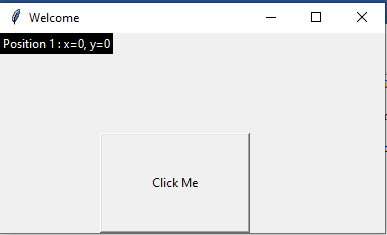
btn.configure(text="Button was clicked !!")

btn = Button(window, text="Click Me", command=clicked)

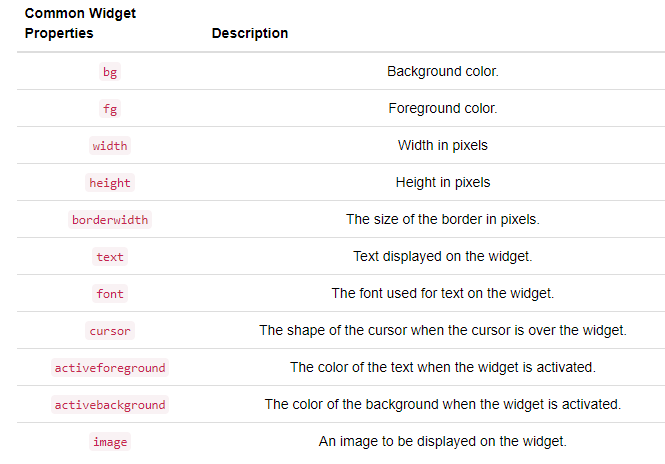
btn.place( height=100, width=150,x=100, y=100)

window.mainloop()

Output

Q13)What are general Common Widget Properties/attributes in Tkinter?



Q13 Write code on different widgets of Tkinter in python

1)Widget...Label

### Tkinter Label Widget Options

| **Name of the option** | **Description** |
| --- | --- |
| anchor | This option is mainly used for controlling the**position of text in the provided widget size**. The default value is **CENTER which is used to align the text in**center in the provided space. |
| bd | This option is used **for the border width of the widget**. Its default value is 2 pixels. |
| bg | This option is used for the background color of the widget. |
| fg | This option is used to specify the foreground color of the text that is written inside the widget. |
| font | This option specifies the font type of text inside the label. |
| height | This option indicates the height of the widget |

# code

from tkinter import \*

window = Tk()

window.title("WELCOME TO PYTHON PROG")

lbl = Label(window,text="firstname",bg="red",fg="yellow")

lbl.grid(column=0,row=0)

window.mainloop()

Output:



2) Widget...Entry

### Tkinter Entry Widget Options:

| **Option Name** | **Description** |
| --- | --- |
| bg | This option is used for the **background color** of the widget. |
| bd | This option is used for the**width of the border** in pixels. Its default value is**2 pixels.** |
| fg | This option is used to indicate the color of the text. |
| font | This option is used to represent the font type of the text |
| justify | This option is used to specify how the text is organized in the case if the text contains multiple lines. |
| relief | This option is used to indicate the type of border. The default value of this option is FLAT. It has more values like GROOVE, RAISED,RIGID. |
| selectbackground | This option is used to indicate the background color of the selected text. |
| selectforeground | It is used to set the font of the selected task. |

from tkinter import \*

window = Tk()

window.title("WELCOME TO PYTHON PROG")

e1 = Entry(window)

e1.grid(row=0,column=1)

window.mainloop()

Output:



3) Widget...Button

### Tkinter Button Widget Options:

| **Option name** | **Description** |
| --- | --- |
| **bd** | **This option is used to represent the width of the border in pixels.** |
| **bg** | **This option is used to represent the background color of the button.** |
| **command** | **The command option is used to set the function call which is scheduled at the time when the function is called.** |
| **fg** | **This option represents the foreground color of the button.** |
| **font** | **This option indicates the font of the button.** |
| **height** | **This option indicates the height of the button. This height indicates the number of text lines in the case of text lines and it indicates the number of pixels in the case of images.** |
| **activebackground** | **This option indicates the background of the button at the time when the mouse hovers the button.** |
| **activeforeground** | **This option mainly represents the font color of the button when the mouse hovers the button.** |

#code

from tkinter import\*

window = Tk()

window.title("Welcome to LikeGeeks app")

def clicked():

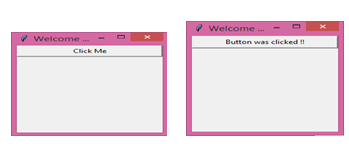
btn.configure(text="Button was clicked !!")

btn = Button(window, text="Click Me", command=clicked)

btn.place(bordermode=OUTSIDE, height=100, width=150)

window.mainloop()

Output



4) Widget:Canvas

from tkinter import\*

window = Tk()

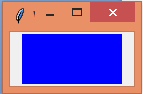
window.title("Welcome to LikeGeeks app")

canvas\_widget = Canvas(window,bg="blue",width=100,height= 50)

canvas\_widget.pack()

window.mainloop()

Output:



from tkinter import\*

window = Tk()

window.title("Welcome to LikeGeeks app")

coord = 10,40,60,80

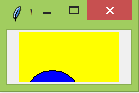
canvas\_widget = Canvas(window,bg="yellow",width=100,height=50)

canvas\_widget.create\_arc(coord,start=10,extent=200,fill="blue")

canvas\_widget.pack()

window.mainloop()

Output:



5) Widget:Listbox

from tkinter import\*

window = Tk()

window.title("Welcome to LikeGeeks app")

Lb1 = Listbox(window,bg="blue",font=12,fg="red")

Lb1.insert(1, "Python")

Lb1.insert(2, "Perl")

Lb1.insert(3, "C")

Lb1.insert(4, "PHP")

Lb1.insert(5, "JSP")

Lb1.insert(6, "Ruby")

Lb1.pack()

window.mainloop()



6) Widget:check button

| **Option name** | **Description** |
| --- | --- |
| activebackground | This option indicates the background color of the checkbutton at the time**when the checkbutton is under the cursor**. |
| bd | This option indicates the **size of the border around the corner**. The default size is 2 pixels. |
| bg | This option is used to **represent the background color of the checkbutton**. |
| command | The command option is used to set the function call which is scheduled at the time when the state of checkbutton is changed. |
| activeforeground | This option mainly represents the foreground color of the button when the checkbutton is under the cursor. |
| fg | This option represents the text color of the checkbutton. |
| font | This option indicates the font of the checkbutton. |
| height | This option indicates the height of the button. This height indicates the number of text lines in the case of text lines and it indicates the number of pixels in the case of images. the default value is 1. |

#code

from tkinter import\*

window = Tk()

window.title("Welcome to python programming")

checkbutton\_widget1 = Checkbutton(window,text="Checkbutton1")

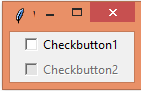
checkbutton\_widget2=Checkbutton(window,text="Checkbutton2",state="disabled")

checkbutton\_widget1.pack()

checkbutton\_widget2.pack()

window.mainloop()

Output:



7) Widget...menubar

from tkinter import \*

window=Tk()

window.title("WELCOME TO PYTHON PROG")

def hello():

print("hi!")

def quit():

print("exit!")

# create a toplevel menu

menubar = Menu(window)

menubar.add\_command(label="Hello!", command=hello)

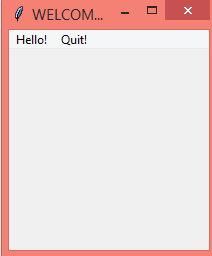
menubar.add\_command(label="Quit!", command=quit)

# display the menu

window.config(menu=menubar)

window.mainloop()

output:



8) Widget..frame

|  |  |
| --- | --- |
| **Name of the option** | **Description** |
| bd | This option is used to represent the **width of the border**. Its default value is **2 pixels**. |
| bg | This option is used to indicate the **normal background color**of a widget. |
| height | This option is used to indicate the height of the frame. |
| width | This option is used to indicate the width of the frame. |
| highlightbackground | This option denotes the color of the background color when it is under focus. |
| highlightthickness | This option is used to specify the thickness around the border when the widget is under the focus. |
| relief | This option specifies the type of the border of the frame. Its default value is FLAT |
| highlightcolor | This option is mainly used to represent the color of the focus highlight when the frame has the focus. |

from tkinter import \*

top = Tk()

top.geometry("140x100")

frame = Frame(top)

frame.pack()

leftframe = Frame(top)

leftframe.pack(side = LEFT)

rightframe = Frame(top)

rightframe.pack(side = RIGHT)

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

btn1.pack(side = LEFT)

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

btn2.pack(side = RIGHT)

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

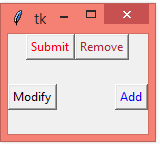
btn3.pack(side = LEFT)

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

btn4.pack(side = RIGHT)

top.mainloop()

output:



9) Widget..combobox

Combobox is **a combination of Listbox and an entry field**. It is one of the Tkinter widgets where it contains a down arrow to select from a list of options. It helps the users to select according to the list of options displayed

from tkinter import \*

from tkinter.ttk import Combobox

window=Tk()

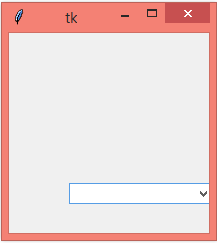
data=("one", "two", "three", "four")

cb=Combobox(window, values=data)

cb.place(x=60, y=150)

window.mainloop()

Output:



10) Widget..radiobutton

| **Name of the option** | **Description** |
| --- | --- |
| anchor | This option is used to **represent the exact position of the text** within the widget, in the case of the **widget contains more space than the requirement of the text**. The default value of this option is CENTER. |
| bg | This option represents**the background color of the widget**. |
| activebackground | This option **represents the background color of the widget**when it is under focus. |
| activeforeground | This option **represents the font color of the widget**when it is under focus. |
| borderwidth | This option is used **to represent the size of the border**. |
| command | This option is used to set the procedure which must be called every time when the state of the radiobutton is changed. |
| font | This option is used to represent the font type of the text of the widget. |
| fg | This option is used to represent the foreground color of the text of the widget. |
| height | This option indicates the vertical dimension of the widget |
| width | This option indicates the horizontal dimension of the widget and it is represented as the number of characters. |

from tkinter import \*

window = Tk()

window.title("Welcome to python programming")

rad1 = Radiobutton(window,text='First', value=1)

rad2 = Radiobutton(window,text='Second', value=2)

rad3 = Radiobutton(window,text='Third', value=3)

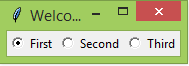
rad1.grid(column=0, row=0)

rad2.grid(column=1, row=0)

rad3.grid(column=2, row=0)

window.mainloop()

Output:



11) Widget..scrollbar

from tkinter import \*

root = Tk()

scrollbar = Scrollbar(root)

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

mylist = Listbox(root, yscrollcommand=scrollbar.set)

for line in range(100):

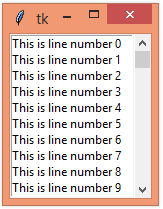
mylist.insert(END, "This is line number " + str(line))

mylist.pack(side=LEFT, fill=BOTH)

scrollbar.config(command=mylist.yview)

root.mainloop()

output:



12) Widget:messagebox/tkmessagebox

from tkinter import \*

from tkinter import messagebox

window = Tk()

window.title("Welcome to python programming")

def hello():

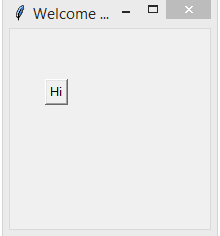
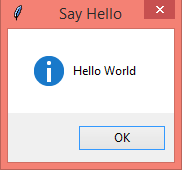
messagebox.showinfo("Say Hello","Hello World")

B1=Button(window,text="Hi",command=hello)

B1.place(x=35,y=50)

window.mainloop()

Output:

13) Widget..scale

from tkinter import \*

window = Tk()

window.title("Welcome to python programming")

w = Scale(window, from\_=0, to=100)

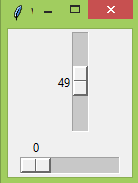
w.pack()

w = Scale(window, from\_=0, to=200, orient=HORIZONTAL)

w.pack()

window.mainloop()

Output:



14) Widget...spinbox

from tkinter import \*

window = Tk()

window.title("Welcome to python programming")

w = Spinbox(window, from\_=0, to=10)

w.pack()

window.mainloop()

Output:



15) **Widget:PanedWindow**

* The **PanedWindow** widget is a geometry manager widget, which can contain one or more child widgets **panes**. The child widgets can be resized by the user, by moving separator lines **sashes** using the mouse.
* **Syntax:** PanedWindow(master, \*\*options)
* **Parameters:**  
  **master**: parent widget   
  **options:** which are passed in config method or directly in the constructor
* PanedWindow can be used to implement common 2-panes or 3-panes but multiple panes can be used.

**Code :**PanedWindow with only twopanes

# Importing everything from tkinter module

from tkinter import \*

# main tkinter window

root = Tk()

# panedwindow object

pw = PanedWindow(orient ='vertical')

# Button widget

top = Button(pw, text ="Click Me,I'm a Button")

top.pack(side = TOP)

# This will add button widget to the panedwindow

pw.add(top)

# Checkbutton Widget

bot = Checkbutton(pw, text ="Choose Me !")

bot.pack(side = TOP)

# This will add Checkbutton to panedwindow

pw.add(bot)

# expand is used so that widgets can expand

# fill is used to let widgets adjust itself

# according to the size of main window

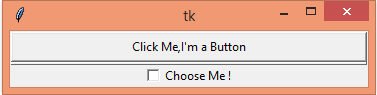
pw.pack(fill = BOTH, expand = True)

# This method is used to show sash

pw.configure(sashrelief = RAISED)

root.mainloop()

output:



Q14 Write steps of Connecting to a MySQL database from Python ….Refer practical

Q15 Create a table in MySQL database and insert data in it from python. ….Refer practical

Q16 Write a code in python to fetch data from MySQL database. ….Refer practical

Q17 What is function of Update, Delete commands in MySQL from python?

Ans

1)Update: UPDATE Operation on any database updates one or more records, which are already available in the database. We can update the values of existing records in MySQL using the UPDATE statement.

2)Delete: We can delete records from an existing table by using the "DELETE FROM" statement