

\*\*\*\*\*

## Internet Kaynaklarından Derleyen YASİN MEYDAN

\*\*\*\*\*

Daire oluşturmak için ya da daire içinde bölümler oluşturmak için kullanılan Arc fonksiyonunun kullanımı

```
from Tkinter import *

root = Tk()
root.title('Canvas')
canvas = Canvas(root, width =400, height=400)

xy = 10, 105, 100, 200
canvas.create_arc(xy, start=0, extent=270, fill='gray60')

canvas.pack()
root.mainloop()
```

\*\*\*\*\*

Resim açmak ve pencereye yüklemek için kullanılan PhotoImage fonksiyonu

```
from Tkinter import *

root = Tk()
root.title('Canvas')
canvas = Canvas(root, width =400, height=400)

img = PhotoImage(file='./logo.gif')
canvas.create_image(145,280, image=img, anchor=CENTER)

canvas.pack()
root.mainloop()
```

\*\*\*\*\*

pencere üzerine yüklenen küçük resimler(bitmap)

```
from Tkinter import *

class LabelDemo( Frame ):
    def __init__( self ):
        Frame.__init__( self )

        self.pack( expand = YES, fill = BOTH )
        self.master.title( "Labels" )

        self.Label3 = Label( self, bitmap = "warning" )
        self.Label3.pack( side = LEFT )

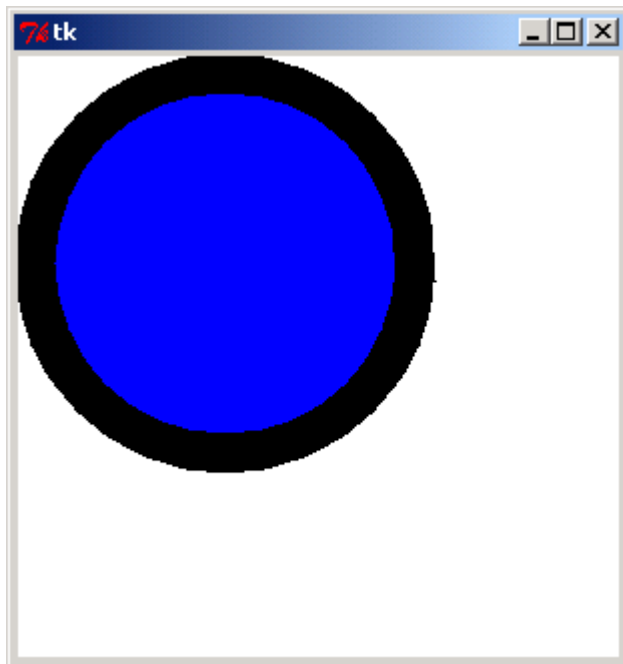
if __name__ == "__main__":
    LabelDemo().mainloop()
```

```
from Tkinter import *

root = Tk()
root.title('Canvas')
canvas = Canvas(root, width =400, height=400)
canvas.create_bitmap(355, 53, bitmap='questhead')

canvas.pack()
root.mainloop()
```

\*\*\*\*\*



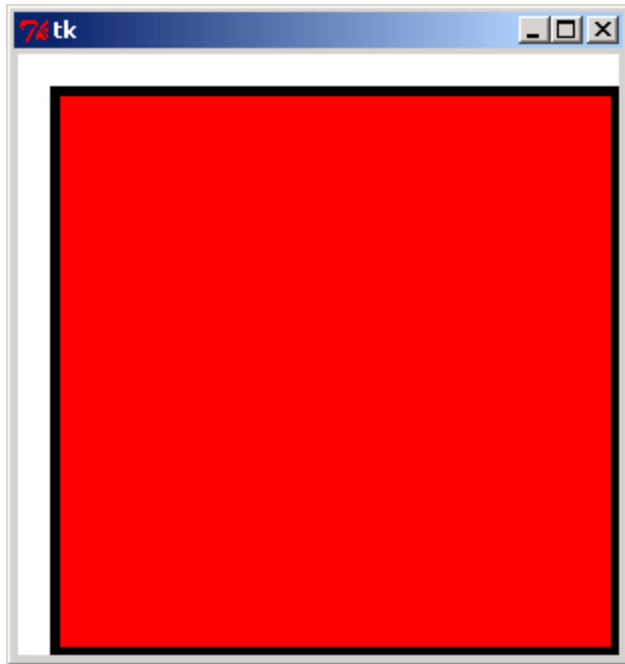
```
from Tkinter import *

canvas = Canvas(width=300, height=300, bg='white')
canvas.pack(expand=YES, fill=BOTH)

canvas.create_oval(10, 10, 200, 200, width=20, fill='blue')

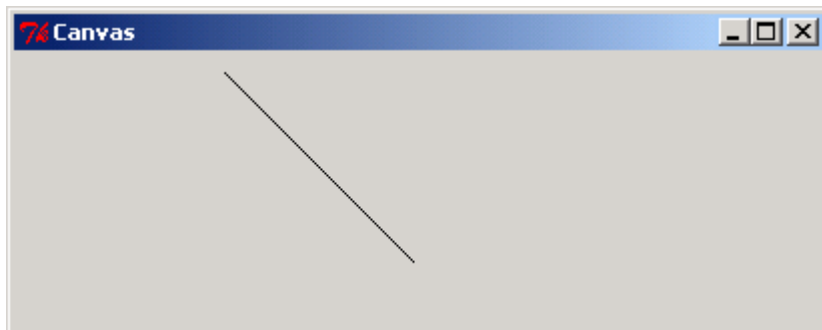
mainloop()
```

**\*\* çerçevelerin etrafında siyah bir çizgi oluşturmak için (width) argümanına çizgi kalınlığını belirleyen bir değer atanır**

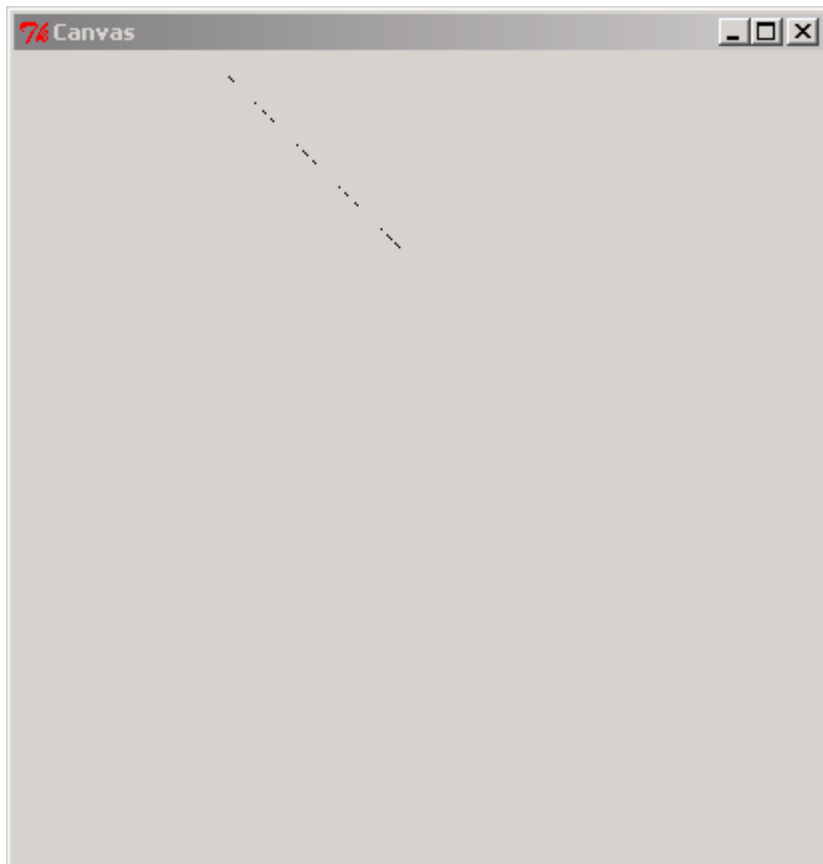


```
from Tkinter import *  
  
canvas = Canvas(width=300, height=300, bg='white')  
canvas.pack(expand=YES, fill=BOTH)  
  
canvas.create_rectangle(20, 20, 300, 300, width=5, fill='red')  
  
mainloop()
```

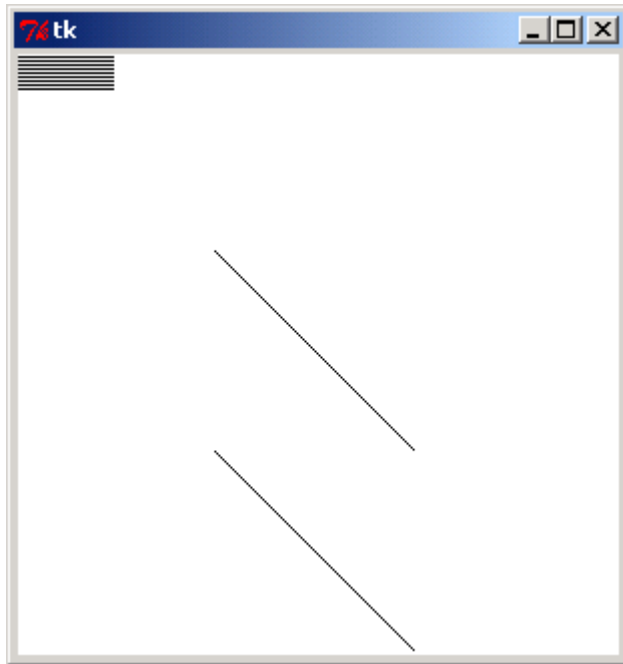
\*\*\*\*\*



```
from Tkinter import *  
  
root = Tk()  
root.title('Canvas')  
canvas = Canvas(root, width =400,  
height=400)  
canvas.create_line(105,10,200,105)  
  
canvas.pack()  
root.mainloop()
```



```
from Tkinter import *  
  
root = Tk()  
root.title('Canvas')  
canvas = Canvas(root, width =400, height=400)  
canvas.create_line(105,10,200,105, stipple='questhead')  
  
canvas.pack()  
root.mainloop()
```

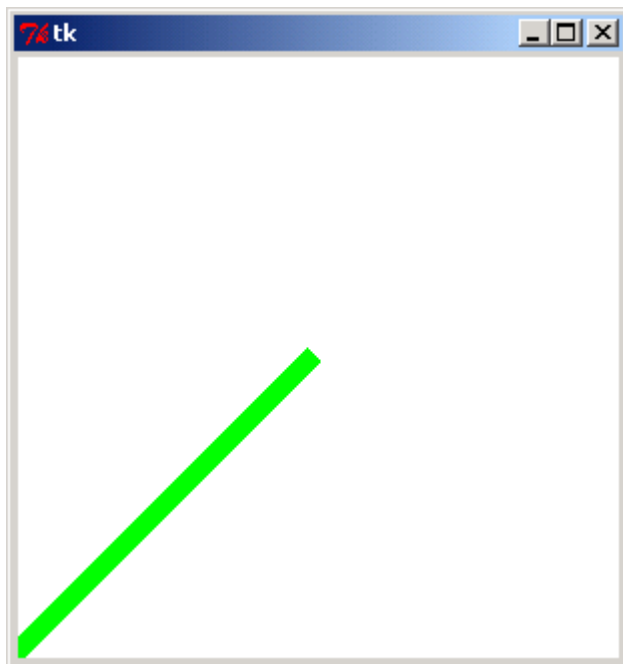


```
from Tkinter import *

canvas = Canvas(width=300, height=300, bg='white')
canvas.pack(expand=YES, fill=BOTH)

canvas.create_line(100, 100, 200, 200)
canvas.create_line(100, 200, 200, 300)
for i in range(1, 20, 2):
    canvas.create_line(0, i, 50, i)

mainloop()
```



```

from Tkinter import *

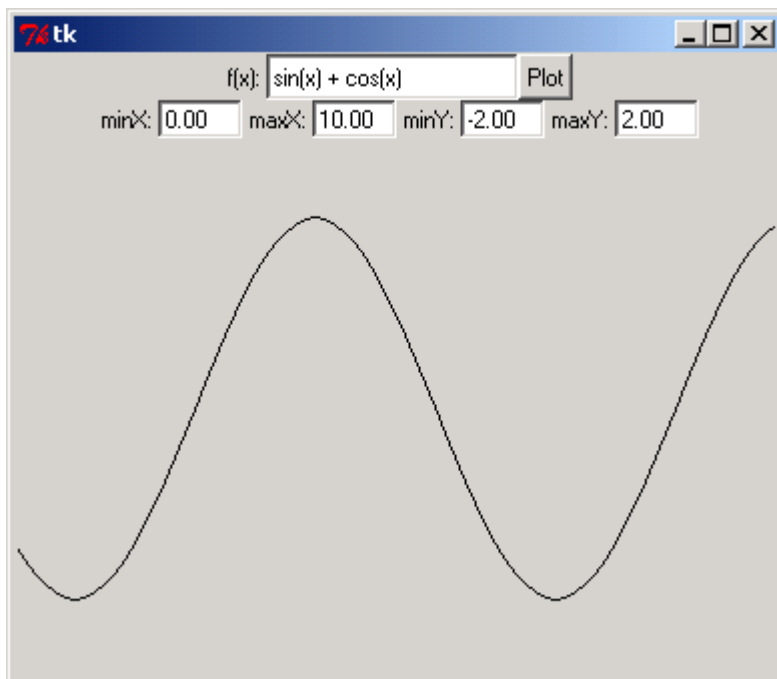
canvas = Canvas(width=300, height=300, bg='white')
canvas.pack(expand=YES, fill=BOTH)

canvas.create_line(0, 300, 150, 150, width=10, fill='green')

mainloop()

*****
*****

```



```

from Tkinter import *
import math

root = Tk()

fram = Frame(root)
Label(fram, text='f(x):').pack(side=LEFT)
func = Entry(fram)
func.pack(side=LEFT, fill=BOTH, expand=1)
butt = Button(fram, text='Plot')
butt.pack(side=RIGHT)
fram.pack(side=TOP)

fram = Frame(root)
bounds = []
for label in 'minX', 'maxX', 'minY', 'maxY':
    Label(fram, text=label+':').pack(side=LEFT)

```

```

    edit = Entry(fram, width=6)
    edit.pack(side=LEFT)
    bounds.append(edit)
fram.pack(side=TOP)

c = Canvas(root)
c.pack(side=TOP, fill=BOTH, expand=1)

def minimax(values=[0.0, 1.0, 0.0, 1.0]):
    for i in range(4):
        edit = bounds[i]
        try: values[i] = float(edit.get())
        except: pass
        edit.delete(0, END)
        edit.insert(END, '%.2f'%values[i])
    return values

def plot():
    minx, maxx, miny, maxy = minimax()

    f = func.get()
    f = compile(f, f, 'eval')

    CX = c.winfo_width()
    CY = c.winfo_height()

    coords = []
    for i in range(0, CX, 5):
        coords.append(i)
        x = minx + ((maxx-minx)*i)/CX
        y = eval(f, vars(math), {'x':x})
        j = CY*(y-miny)/(maxy-miny)
        coords.append(j)

    c.delete(ALL)
    c.create_line(*coords)

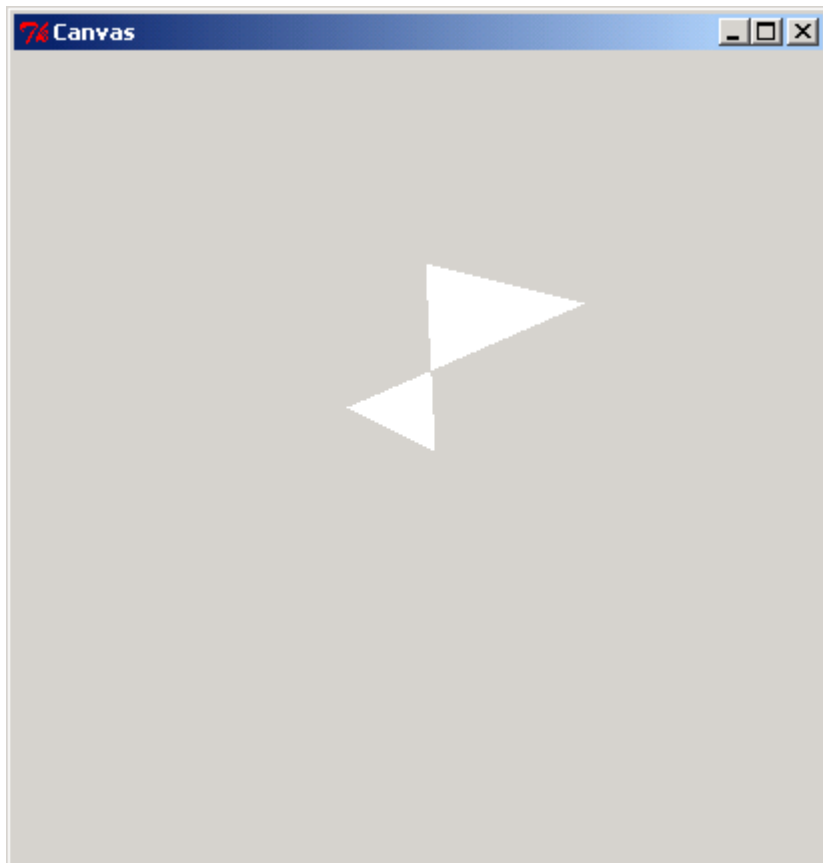
butt.config(command=plot)

f = 'sin(x) + cos(x)'
func.insert(END, f)
minimax([0.0, 10.0, -2.0, 2.0])

root.mainloop()

```

---



---

```
from Tkinter import *  
  
root = Tk()  
  
root.title('Canvas')  
canvas = Canvas(root, width =400, height=400)  
  
canvas.create_polygon(205,105,285,125,166,177,210,199,205,105,  
fill='white')  
  
canvas.pack()  
root.mainloop()
```

---