

Hesap Makinası

Python Tkinter, masaüstü uygulama oluşturmanın hızlı ve kolay bir yolunu sağlayan Python için standart bir GUI (Grafiksel Kullanıcı Arayüzü) yöntemidir.

Tkinter(tikincır), Python ile birlikte gelen GUI kütüphanesidir.

Toplama, çıkarma, çarpma ve bölme gibi temel aritmetik işlemleri gerçekleştirebilen Python Tkinter modülünü kullanarak GUI tabanlı basit bir hesap makinesi oluşturalım.

tkinter modülünden her şeyi içe aktarın

```
from tkinter import *
```

bir GUI penceresi oluşturun

```
pencere = Tk()
```

#GUI penceresinin arka plan rengini ayarla yın

```
pencere.configure(background="gray90")
```

Not: tkinter colors aratarak tkinter in kullandığı renk isimlerini bulabilirsiniz.

GUI penceresinin başlığını ayarlayın

```
pencere.title("Hesap Makinesi")
```

GUI penceresinin boyutunu ayarlayın

```
pencere.geometry("350x400")
```

#Ekran Boyutunun ayarlanmasını kapatın

```
pencere.resizable(False, False)
```

resizable(risayzıbil)=yeniden boyutlandırılabilir

GUI'yi başlat(en alt satırda olması lazım)

```
pencere.mainloop()
```

İlk önce tasarım yapalım.



ekran için metin giriş kutusu oluştur.

#justify ,bu seçenek metnin nasıl hizalanacağını denetler: ORTA, SOL veya SAĞ.

```
ekran = Entry(pencere, font="Arial 20",justify=RIGHT)
justify(castıfayl)
```

grid yöntemi, tablo benzeri bir yapıdır

widget'ları ilgili pozisyonlara

yerleştirmek için kullanılır.

```
ekran.grid(columnspan=3, ipadx=15)
```

#columnspan – widget'ın kapladığı sütun sayısı; varsayılan 1.

#ipadx, ipady – Widget'ın sınırları içerisinde yatay ve dikey olarak kaç piksel doldurulacağı.

#Birinci Satır

```
button1 = Button(pencere,text=1,fg="white",bg="gray50",font="Helvetica 20
bold",width=6,command=lambda: islem(1))
```

```
button1.grid(row=1,column=0)
```

```
button2 = Button(pencere,text=2,fg="white",bg="gray50",font="Helvetica 20
bold",width=6,command=lambda: islem(2))
```

```
button2.grid(row=1,column=1)
```

```
button3 = Button(pencere,text=3,fg="white",bg="gray50",font="Helvetica 20
bold",width=6,command=lambda: islem(3))
```

```
button3.grid(row=1,column=2)
```

#İkinci Satır

```
button4 = Button(pencere,text=4,fg="white",bg="gray50",font="Helvetica 20  
bold",width=6,command=lambda: islem(4))
```

```
button4.grid(row=2,column=0)
```

```
button5 = Button(pencere,text=5,fg="white",bg="gray50",font="Helvetica 20  
bold",width=6,command=lambda: islem(5))
```

```
button5.grid(row=2,column=1)
```

```
button6 = Button(pencere,text=6,fg="white",bg="gray50",font="Helvetica 20  
bold",width=6,command=lambda: islem(6))
```

```
button6.grid(row=2,column=2)
```

#Üçüncü Satır

```
button7 = Button(pencere,text=7,fg="white",bg="gray50",font="Helvetica 20  
bold",width=6,command=lambda: islem(7))
```

```
button7.grid(row=3,column=0)
```

```
button8 = Button(pencere,text=8,fg="white",bg="gray50",font="Helvetica 20  
bold",width=6,command=lambda: islem(8))
```

```
button8.grid(row=3,column=1)
```

```
button9 = Button(pencere,text=9,fg="white",bg="gray50",font="Helvetica 20  
bold",width=6,command=lambda: islem(9))
```

```
button9.grid(row=3,column=2)
```

#Dördüncü Satır

```
button0 = Button(pencere,text=0,fg="white",bg="gray50",font="Helvetica 20  
bold",width=6,command=lambda: islem(0))
```

```
button0.grid(row=4,column=0)
```

```
plus = Button(pencere,text="+",fg="white",bg="gray25",font="Helvetica 20  
bold",width=6,command=lambda: islem("+"))
```

```
plus.grid(row=4,column=1)
```

```
divide = Button(pencere,text="/",fg="white",bg="gray25",font="Helvetica 20  
bold",width=6,command=lambda: islem("/"))
```

```
divide.grid(row=4,column=2)
```

#Beşinci Satır

```
decimall= Button(pencere, text='.', fg='white', bg='gray25', font= ('Helvetica  
20 bold'),width=6,command=lambda: islem("."))  
decimall.grid(row=5, column=0)
```

```
minus = Button(pencere, text=' - ', fg='white', bg='gray25', font= ('Helvetica  
20 bold'),width=6,command=lambda: islem("-"))  
minus.grid(row=5, column=1)
```

```
multiply = Button(pencere, text=' * ', fg='white', bg='gray25', font= ('Helvetica  
20 bold'),width=6,command=lambda: islem("*"))  
multiply.grid(row=5, column=2)
```

#Altıncı Satır

```
sil = Button(pencere, text='Sil', fg='white', bg='Orange red', font= ('Helvetica  
20 bold'),width=6,command=silmek)  
sil.grid(row=6, column=0)
```

```
equal = Button(pencere, text=' = ', fg='white', bg='Orange red',font=  
( 'Helvetica 20 bold'),command=sonuc)  
equal.grid(row=6, column=1,columnspan=2,ipadx=85)
```

Şimdi işlemleri yaptıralım.

StringVar() değişken sınıfıdır

```
denklem = StringVar()
```

StringVar()

Tkinter, Label ve Entry gibi widget'ların değerlerini daha etkili bir şekilde işlemek için python değişkeni gibi çalışan ve ek özelliklere sahip yapılar içerir. Bu değişkenler ayrıca değerlerine erişmek ve bunları değiştirmek için getter ve setter yöntemleri içerir. StringVar bunlara bir örnektir.

Not:Eğer (pencere = Tk())pencere başlatılmadan oluşturulursa çalışmayacaktır.

RuntimeError: Too early to create variable: no default root window

RuntimeError: Değişken oluşturmak için çok erken: varsayılan kök penceresi yok

entry ye ekleyelim.

textvariable(metin değişkeni),Giriş widget'ınızdan geçerli metni alabilmek için bu seçeneği StringVar sınıfının bir örneğine ayarlamamız gerekir.

```
ekran = Entry(pencere, textvariable=denklem,font="Arial  
20",justify=RIGHT)
```

Girilen değerleri saklayacağımız değişkenimizi oluşturalım.

ifade değişkenini global olarak bildir

```
gelen_ifade= ""
```

Girilen değerleri birleştirmek için fonksiyonumuzu yazalım

gelen_ifade değişkene ilk gelen değeri , sonrasında yapılacak işlemi ve ikinci sayıyı atamak için değişkenimizi güncelleyeceğiz.

gelen_ifade güncelleme fonksiyonu

```
def islem(sayi):
```

```
    global gelen_ifade
```

dize birleştirme

```
    gelen_ifade = gelen_ifade + str(sayi)
```

set metodunu kullanarak ifadeyi güncelle

```
    denklem.set(gelen_ifade)
```

Not:Bir fonksiyon içerisindeki global değişkeni değiştirmek istiyorsanız global anahtar sözcüğünü kullanın. Eğer global anahtar kelimesini kullanmazsanız aşağıdaki hatayı alırsınız.

UnboundLocalError: cannot access local variable 'gelen_ifade' where it is not associated with a value

UnboundLocalError: Bir değerle ilişkilendirilmemiş yerel değişken 'gelen_ifade'a erişilemiyor

Eşittir tuşu için fonksiyon yazalım

Son ifadeyi değerlendirmek için fonksiyon

def sonuc():

#oluşabilecek hataları engelleyin.

try:

global gelen_ifade

eval fonksiyonu ifadeyi değerlendirir

ve str fonksiyonu sonucu dönüştürür

total = str(eval(gelen_ifade))

denklem.set(total)

gelen_ifade içerisini boşaltalım

gelen_ifade = ""

eğer hata oluşursa, o zaman işlem yap

except:

denklem.set(" Hata oluştu ")

gelen_ifade = ""

eval()

x = 5

print(eval("x * 2"))

İfade argümanı, Python ifadesi (teknik olarak bir koşul listesi) olarak ayrıştırılır ve değerlendirilir.

Not: Bu çok güçlüdür, ancak güvenilmeyen girdiden değerlendirmek için dizeleri kabul ederseniz çok tehlikelidir.

https://nedbatchelder.com/blog/201206/eval_really_is_dangerous.html

İçeriği temizleme işlevi

metin giriş kutusu

```
def silmek():  
    global gelen_ifade  
    gelen_ifade = ""  
    denklem.set("")
```