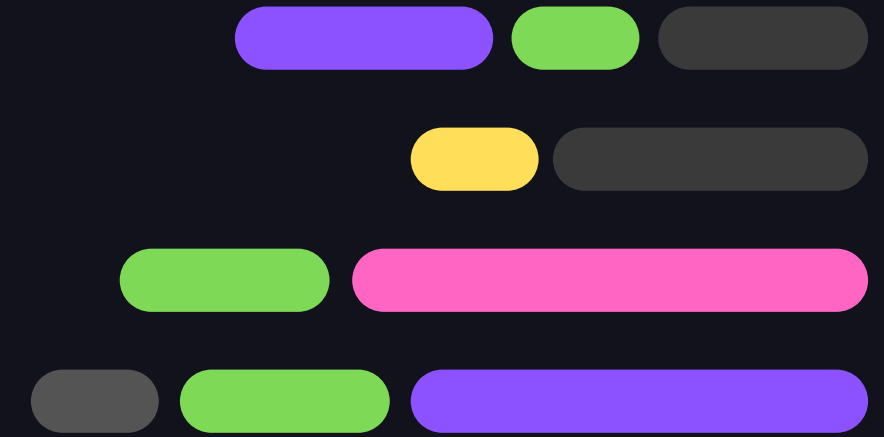
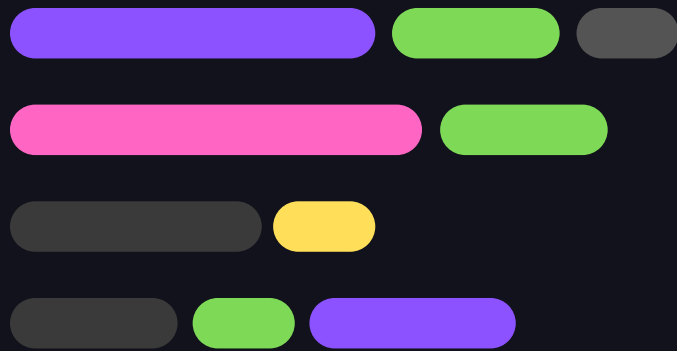




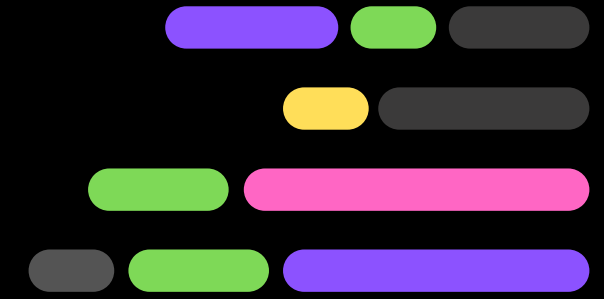
{ TEMEL PROGRAMLAMA II }

Samsun Üniversitesi
Teknik Bilimler Meslek Yüksekokulu
Arka-Yüz Yazılım Geliştirme Pr.





Bu Haftanın Ders Kazanımları



01

Dosya İşlemleri

02

Uygulama Saati

03

Tkinter

Dosya

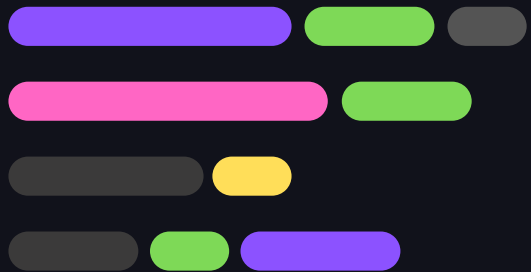
işlemleri



01

Dosya İşlemleri

Bilgisayarımızda kullandığımız hangi programlar dosya okuma/yazma işlemi yapıyor olabilir?





01

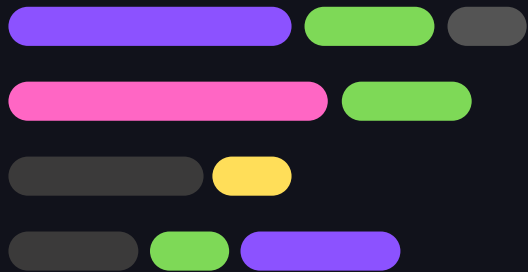
Dosya İşlemleri

WhatsApp / Telegram Desktop

Google Chrome / Tarayıcılar

VS Code

Microsoft Ürünleri vs.





01

Dosya İşlemleri

Gerçek uygulamaların büyük kısmı veriyi saklamak, okumak, güncellemek ve yönetmek zorundadır. İşte burada dosya işlemleri devreye girer.

Python'da dosya işlemleri sayesinde:

- veri kaydedebiliriz,
- eski verileri okuyabiliriz,
- rapor oluşturabiliriz,
- log tutabiliriz,
- yedek alabiliriz,
- kullanıcı bilgilerini saklayabiliriz.



01

Dosya İşlemleri

Dosya okuma ve yazma işlemlerine geçmeden önce Python'un çalışma dizini ile ilgili ayarlarının yapılması gerekmektedir.

Python dilinin varsayılan çalışma dizini, programın kurulduğu klasördür.

Python'un bilgisayarın dosya sistemlerine erişmesi için "**OS**" adındaki Python modülünün çağrılması gerekir. Komut satırından bu modüle ait

getcwd() fonksiyonu çağrılarak programın mevcut çalışma dizini öğrenilebilir.



01

Dosya İşlemleri

```
import os
dizin = os.getcwd()
print("Bu proje şuan : " + dizin + " dizininde çalışıyor.")
```

Yandaki kodu çalıştırınız.



01

Dosya İşlemleri

```
import os
dizin = os.getcwd()
print("Bu proje şuan : " + dizin + " dizininde çalışıyor.")
```

Yandaki kodu çalıştırınız.

getcwd() fonksiyonu çağrılarak programın mevcut çalışma dizini öğrenilebilir.



01

Dosya İşlemleri

```
import os
dizin = os.getcwd()
print(dizin)
os.chdir('c:\\Program Files')

yeni_dizin = os.getcwd()
print("Bu proje şuan : " + yeni_dizin + " dizininde çalışıyor.")
```

Yandaki kodu çalıştırınız.



01

Dosya İşlemleri

```
import os
dizin = os.getcwd()
print(dizin)
os.chdir('c:\\Program Files')

yeni_dizin = os.getcwd()
print("Bu proje şuan : " + yeni_dizin + " dizininde çalışıyor.")
```

Yandaki kodu çalıştırınız.

Mevcut çalışma dizinini değiştirmek için **chdir()** fonksiyonu kullanılır.



01

Dosya İşlemleri

```
import os
dizin = os.getcwd()
print(dizin)
os.chdir('c:\\test')

yeni_dizin = os.getcwd()
print("Bu proje şuan : " + yeni_dizin + " dizininde çalışıyor.")
```

Yandaki kodu çalıştırınız.



01

Dosya İşlemleri

Kod bloğu çalıştırıldığında hata ile karşılaşılacaktır.

C: sürücüsünde "test" isimli bir klasör olmadığı için Python belirtilen dizini bulamaz. Bu nedenle çalışma dizini değiştirme işlemi başarısız olur. Bilgisayarın c: sürücüsünde ilgili klasör oluşturularak bu sorun çözülebilir ancak klasör ve dosya oluşturma işlemlerini kod bloğu içinde yapmak daha doğru bir yöntemdir.

```
import os
dizin = os.getcwd()
print(dizin)
os.chdir('c:\\test')

yeni_dizin = os.getcwd()
print("Bu proje şuan : " + yeni_dizin + " dizininde çalışıyor.")
```

01

Dosya İşlemleri

Yol (**Path**), dosya veya klasörün bilgisayarın dosya sistemindeki konumunu belirtir. Doğal olarak bir dosyaya erişmek, bir dosyayı silmek veya değiştirmek için o dosyanın yolunun bilinmesi gerekir. Yazılan kodların platformdan bağımsız olarak Windows, Linux, Unix ve Mac OS gibi bütün işletim sistemlerinde çalışabilmesi için dosya ve dizin işlemlerini platformdan bağımsız yollar tanımlayarak yapmak gerekecektir. Python, yol tanımlama işlemleri için kullanışlı birçok fonksiyonu barındıran

os.path alt modülünü içerir.



01

Dosya İşlemleri

join() fonksiyonu, Windows işletim sistemi için ters bölü (\), Unix işletim sistemi için bölü (/) işaretini yol bileşenlerinin arasına ekleyerek birleştirir. Tek bir fonksiyon ile hem Windows hem de Unix işletim sisteminde doğru çalışacak "path" bilgisi oluşturulmuş olur.

split() fonksiyonu ise verilen yol bilgisini parçalara ayırarak liste hâlinde verir.

```
import os
yol = os.path.join('test', 'python')
print(yol)
pc = os.path.split(yol)
print(pc) |
```



01

Dosya İşlemleri

Program yazılırken dosya yolu verilen bir dosyanın veya dizinin var olup olmadığının bazı durumlarda kontrol edilmesi gerekir. Bunun için yolun var olup olmadığını belirten **exists()** ve dizin olup olmadığını kontrol eden **isdir()** fonksiyonları kullanılır.

```
import os
yol = os.path.join("C:\\", "Program files")
print(yol)
if os.path.exists(yol):
    print(yol + ' : var')
    if os.path.isdir(yol):
        print(yol + ' : bir dizin')
else:
    print("Yol bulunamadı.")
```

01

Dosya İşlemleri

Yandaki kodu çalıştırınız.

```
import os
yol = os.path.join("C:\\", "test")
print(yol)
if os.path.exists(yol):
    print(yol + ' : var')
    if os.path.isdir(yol):
        print(yol + ' : bir dizin')
else:
    print("Yol bulunamadı.")
```



01

Dosya İşlemleri

Yandaki kodu çalıştırınız ve C Klasörünü kontrol ediniz.

```
import os

yol = os.path.join("C:\\", "test")
if os.path.exists(yol):
    print(yol + ' : var')
    if os.path.isdir(yol):
        print(yol + ' : bir dizin')
else:
    print("Yol bulunamadı!")
    os.mkdir(yol)
    print("oluşturuluyor...")
```



01

Dosya İşlemleri

mkdir() Sadece belirtilen tek bir klasörü oluşturur.

```
import os

yol = os.path.join("C:\\", "test")
if os.path.exists(yol):
    print(yol + ' : var')
    if os.path.isdir(yol):
        print(yol + ' : bir dizin')
else:
    print("Yol bulunamadı!")
    os.mkdir(yol)
    print("oluşturuluyor...")
```



01

Dosya İşlemleri

Yandaki kodu çalıştırınız.

```
import os

yol = os.path.join("C:\\", "test2", "kodlar", "python")
if os.path.exists(yol):
    print(yol + ' : var')
    if os.path.isdir(yol):
        print(yol + ' : bir dizin')
else:
    print("Yol bulunamadı!")
    os.mkdir(yol)
    print("oluşturuluyor...")
```



01

Dosya İşlemleri

Kod hata aldı çünkü `mkdir()` sadece belirtilen **tek bir** klasörü oluşturur.

```
import os

yol = os.path.join("C:\\", "test2", "kodlar", "python")
if os.path.exists(yol):
    print(yol + ' : var')
    if os.path.isdir(yol):
        print(yol + ' : bir dizin')
else:
    print("Yol bulunamadı!")
    os.mkdir(yol)
    print("oluşturuluyor...")
```



01

Dosya İşlemleri

Kod hata aldı çünkü `mkdir()` sadece belirtilen tek bir klasörü oluşturur. Belirtilen yol boyunca eksik olan tüm klasörleri (üst klasörler dahil) oluşturmak için **`makedirs()`** kullanılır.

```
import os

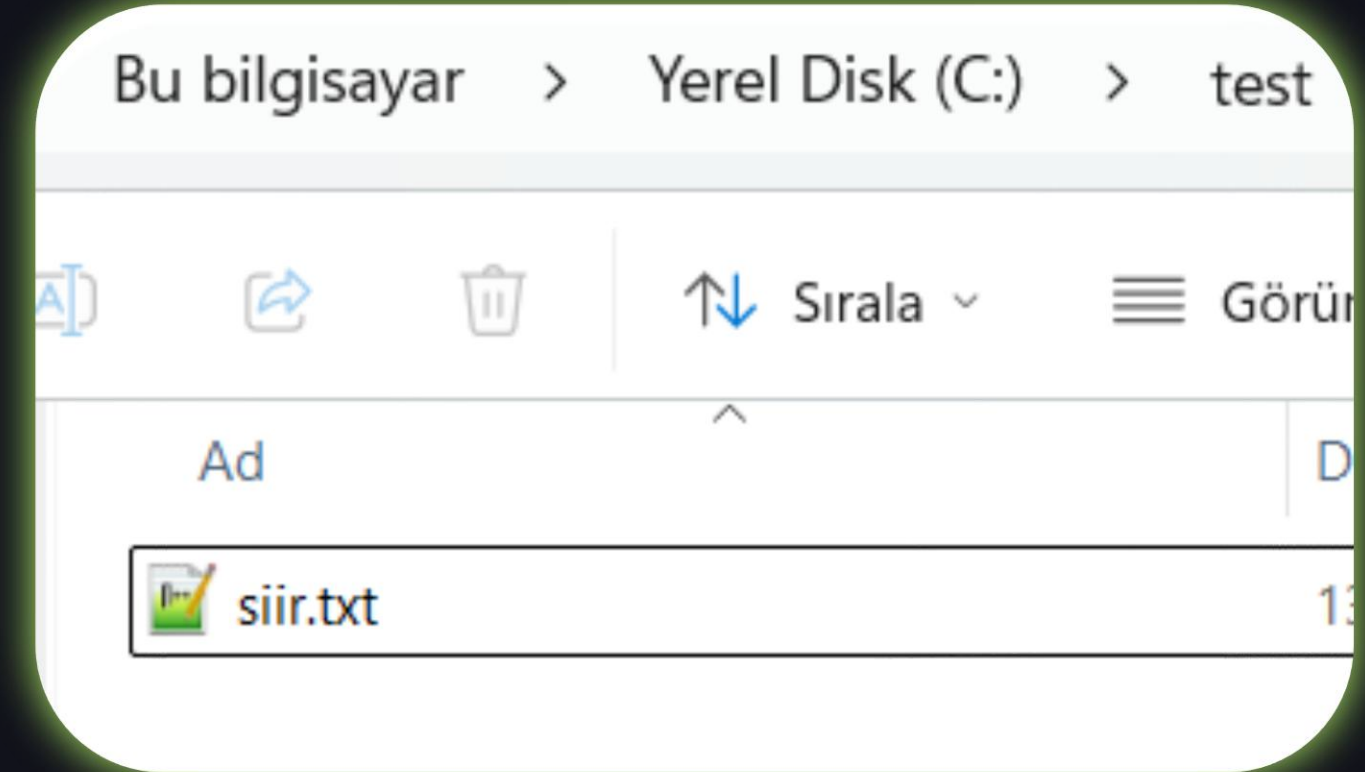
yol = os.path.join("C:\\", "test2", "kodlar", "python")
✓ if os.path.exists(yol):
    print(yol + ' : var')
✓ if os.path.isdir(yol):
    print(yol + ' : bir dizin')
✓ else:
    print("Yol bulunamadı!")
    os.makedirs(yol)
    print("oluşturuluyor...")
```



01

Dosya İşlemleri

C:\test klasörünü açın ve internetten bulduğunuz bir şiiri test klasörünün içine not defteri ile **siir.txt** olarak kayıt ediniz.





01

Dosya İşlemleri

Yandaki kodu çalıştırınız.

```
import os

os.chdir('C:\\test')
dosya = open("siir.txt")
print(dosya.read())
```



01

Dosya İşlemleri

Yandaki kodu çalıştırınız.

Mevcut bir dosyayı açmak için

open() fonksiyonu kullanılır.

open() fonksiyonu, belirtilen yolda bulunan dosyayı açar ve dosya nesnesi olarak programa döndürür.

```
import os  
  
os.chdir('C:\\test')  
dosya = open("siir.txt")  
print(dosya.read())
```

01

Dosya İşlemleri

```
import os  
  
os.chdir('C:\\test')  
dosya = open("siir.txt")  
print(dosya.read())
```

Korkma, sığınmez bu âfaklarda yâzen al sancak;
Sığınmeden yurdumun âstânde tâten en son ocak.
O benim milletimin yıldâzâdır, parlayacak;
O benimdir, o benim milletimindir ancak.

Karakter problemi oluşmuş olabilir, bunu çözmek için utf
belirlemeliyiz.



01

Dosya İşlemleri

Karakter problemi oluşmuş olabilir, bunu çözmek için utf belirlemeliyiz.

```
import os

os.chdir('C:\\test')
dosya = open("siir.txt", encoding="utf-8")
print(dosya.read())
```



01

Dosya İşlemleri

`Dosya.read()` tipi string olduğu için `split` ile kelime kelime bölme yapıp bir listeye atayabiliriz.

```
import os

os.chdir('C:\\test')
dosya = open("siir.txt", "r", encoding="utf-8")
kelimeler = dosya.read().split()
print(kelimeler)
```



01

Dosya İşlemleri

`open()` fonksiyonu, dosya ismi ve mod (dosya açma biçimi) olarak iki parametre alır. Dosyayı açmak için 4 farklı mod kullanılır.

“**r**” - Okuma - Okumak için bir dosya açar. Dosya yoksa hata verir.

“**a**” - Ekleme - Olan dosyayı düzenleme yapmak için açar yoksa oluşturulur.

“**w**” - Yazma - Yazma modunda bir dosya açar. Dosya yoksa oluşturulur.

“**x**” - Oluştur - Belirtilen dosyayı oluşturur. Dosya varsa hata döndürür.



01

Dosya İşlemleri

“**r**” - Okuma - Okumak için bir dosya açar. Dosya yoksa hata verir.

```
import os

os.chdir('C:\\test')

with open("siir.txt", "r", encoding="utf-8") as f:
    print(f.read())
```



01

Dosya İşlemleri

“**r**” - Okuma - Okumak için bir dosya açar. Dosya yoksa hata verir.

```
import os

os.chdir('C:\\test')

with open("siir.txt", "r", encoding="utf-8") as f:
    print(f.read())
```

with open() Python'da dosya işlemlerini güvenli ve otomatik şekilde yönetmek için kullanılır.



01

Dosya İşlemleri

“a” - Ekleme - Olan dosyayı düzenleme yapmak için açar yoksa oluşturulur.
Yandaki kodu üst üste 3 kez çalıştırıp c:\test klasörünü açınız.

```
import os  
  
os.chdir('C:\\test')  
✓ with open("test.txt", "a") as f:  
    f.write("Merhaba\nDersimiz Python\n")
```



01

Dosya İşlemleri

“**W**” – Yazma – Yazma
modunda bir dosya açar.
Dosya yoksa oluşturulur.
Varsa içeriğini komple
günceller.
Yandaki kodu 1 kez
çalıştırıp c:\test
klasörünü açınız.

```
import os  
  
os.chdir('C:\\test')  
  
✓ with open("test.txt", "w") as f:  
    f.write("Merhaba\nDersimiz Django")
```



01

Dosya İşlemleri

“**X**” - Oluştur -
Belirtilen dosyayı
oluşturur. Dosya varsa
hata döndürür.

Yandaki kodu çalıştırınız.

```
import os  
  
os.chdir('C:\\test')  
  
✓ with open("test.txt", "x") as f:  
    f.write("Bu dosya yeni oluşturuldu\n")
```



01

Dosya İşlemleri

Yandaki kodu çalıştırınız.

```
import os

os.chdir('C:\\test')

with open("test.txt", "w", encoding="utf-8") as f:
    f.write("Merhaba\nDersimize devam edelim\n")

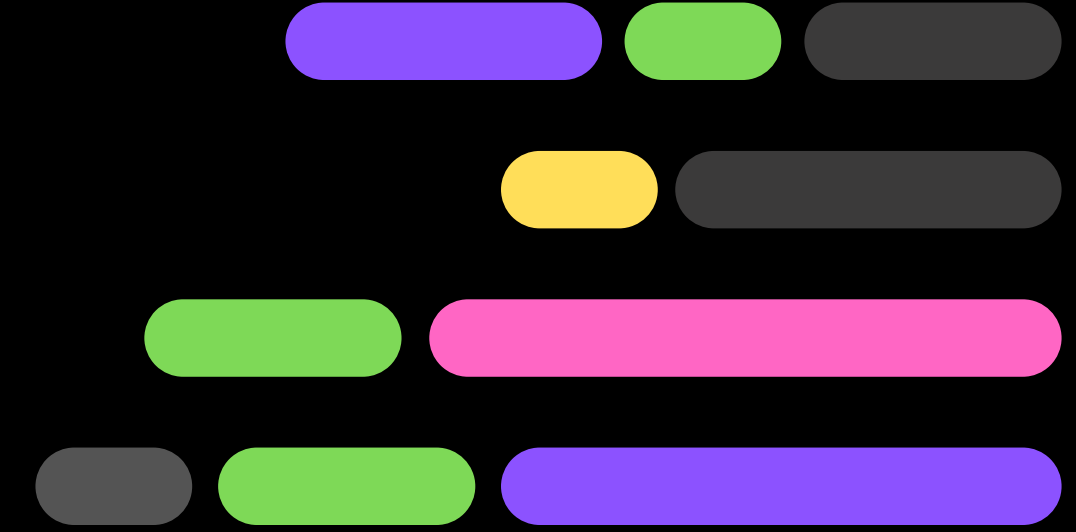
with open("test.txt", "r", encoding="utf-8") as f:
    print(f.read())
```

Uygulama Saati



02

Uygulama Saati



Kullanıcıdan kullanıcı adı ve şifre alınacaktır, eğer kullanıcı adı 'admin', şifre '1234' ise giriş başarılı, yanlış ise giriş hatalı mesajı verip program kapanmalı. Program her kapanışından önce c:\test içerisine log.txt dosyasına girişin başarılı olup olmadığı tarih saat ile eklemelidir.

```
2026-05-13 15:54:54.024469 - admin - HATALI GİRİŞ
2026-05-13 15:54:57.487553 - asd - HATALI GİRİŞ
2026-05-13 15:55:00.587660 - admin - BAŞARILI GİRİŞ
```



02

Uygulama Saati

Kullanıcıdan kullanıcı adı ve şifre alınacaktır, eğer kullanıcı adı 'admin', şifre '1234' ise giriş başarılı, yanlış ise giriş hatalı mesajı verip program kapanmalı. Program her kapanışından önce c:\test içerisine log.txt dosyasına girişin başarılı olup olmadığı tarih saat ile eklemelidir.

```
import os
import datetime

os.chdir('C:\\test')

dogru_kullanici = "admin"
dogru_sifre = "1234"

kullanici = input("Kullanıcı adı: ")
sifre = input("Şifre: ")

zaman = datetime.datetime.now()

if kullanici == dogru_kullanici and sifre == dogru_sifre:
    durum = "BAŞARILI GİRİŞ"
    print("Giriş başarılı")
else:
    durum = "HATALI GİRİŞ"
    print("Giriş başarısız")

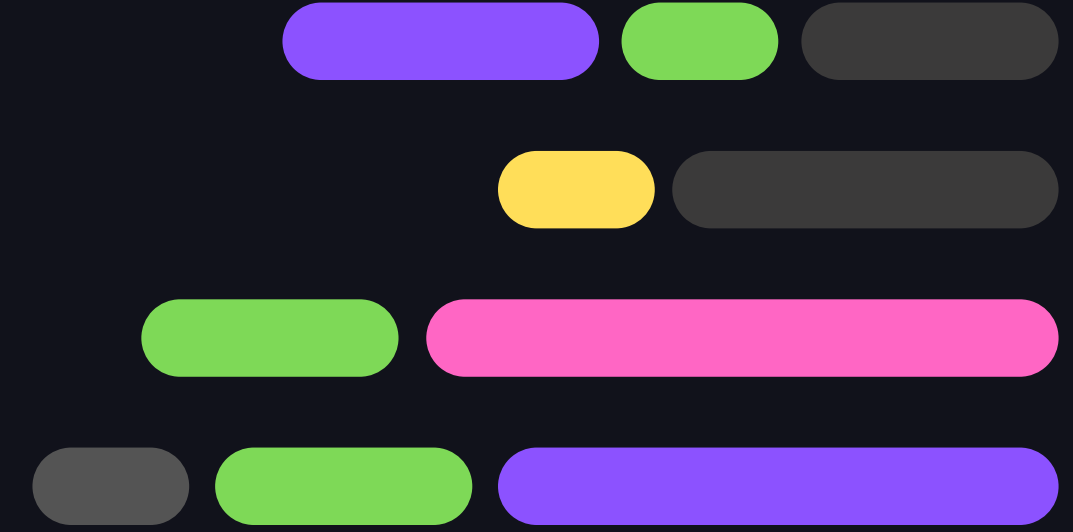
with open("log.txt", "a", encoding="utf-8") as f:
    f.write(f"{zaman} - {kullanici} - {durum}\n")
```

Tkinter

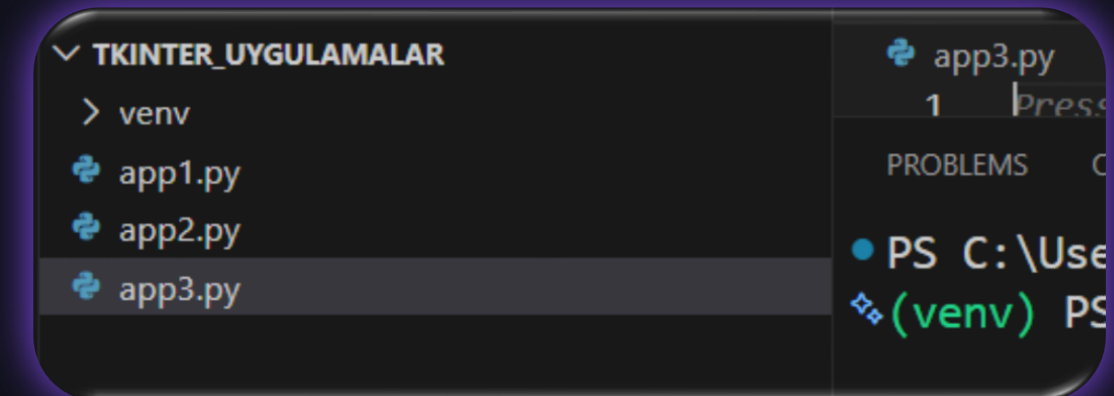


03

Tkinter



- 1- Masaüstünde 'Tkinter_Uygulamalar' isminde bir klasör oluşturun.
- 2- vs code'da open folder ile bu klasörü açın.
- 3- Vs code terminalden sanal ortam oluşturun. `python -m venv venv`
- 4- Sanal ortamı başlatın. `venv/scripts/activate`
- 5- Birazdan yazacağınız her uygulamayı farklı bir py ye kaydedin.





03

Tkinter

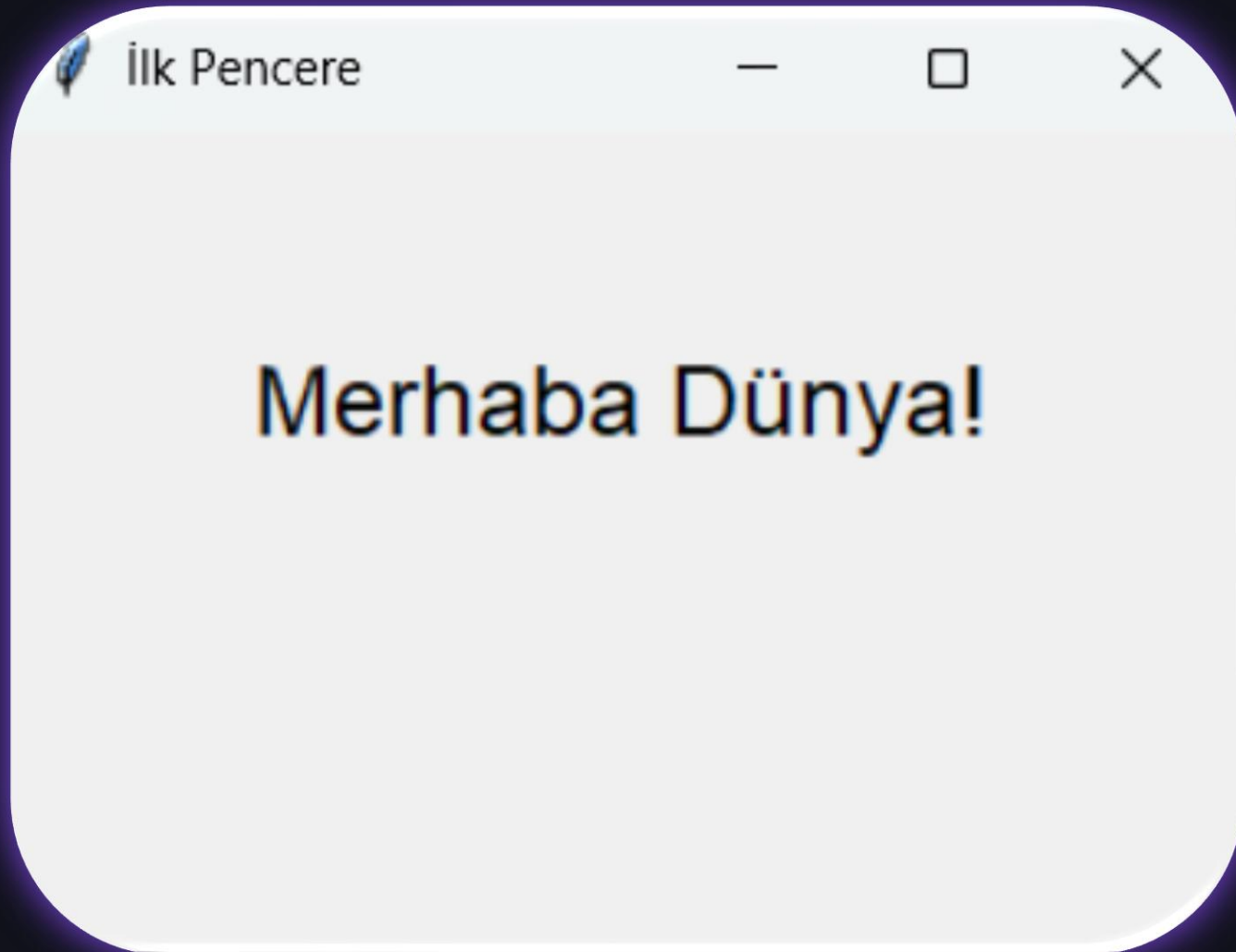
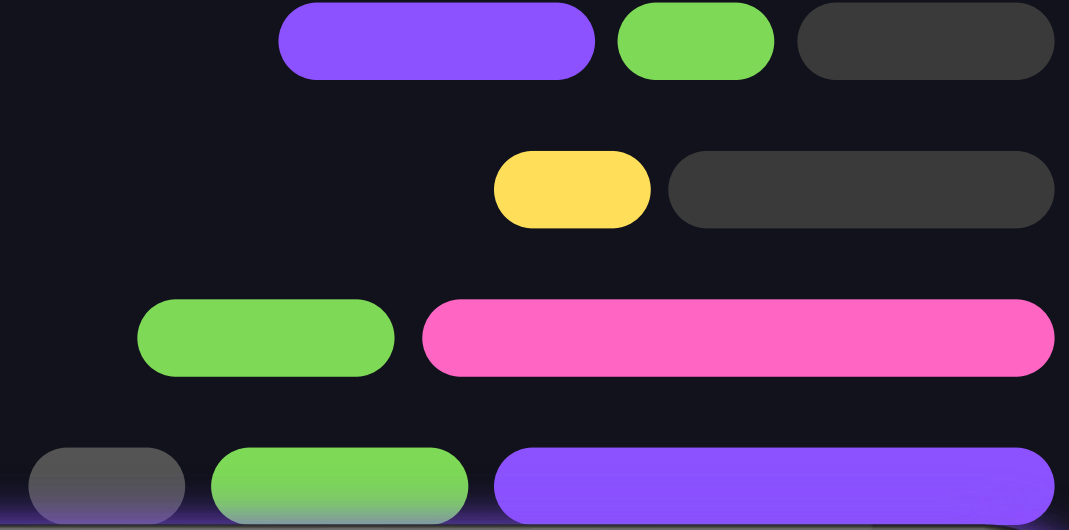
Yandaki kodu çalıştırınız.

```
test.py > ...
1   import tkinter as tk
2
3   pencere = tk.Tk()
4   pencere.title("İlk Pencere")
5   pencere.geometry("300x200")
6
7   etiket = tk.Label(pencere, text="Merhaba Dünya!", font=("Arial", 18))
8   etiket.pack(pady=50)
9
10  pencere.mainloop()
```



03

Tkinter



```
test.py > ...
1 import tkinter as tk
2
3 pencere = tk.Tk()
4 pencere.title("İlk Pencere")
5 pencere.geometry("300x200")
6
7 etiket = tk.Label(pencere, text="Merhaba Dünya!", font=("Arial", 18))
8 etiket.pack(pady=50)
9
10 pencere.mainloop()
```



03

Tkinter

Yandaki kodu çalıştırınız.

```
import tkinter as tk

def tikla():
    etiket.config(text="Butona tıklandı!")

pencere = tk.Tk()
pencere.title("Buton Örneği")
pencere.geometry("300x200")

etiket = tk.Label(pencere, text="Merhaba!", font=("Arial", 14))
etiket.pack(pady=30)

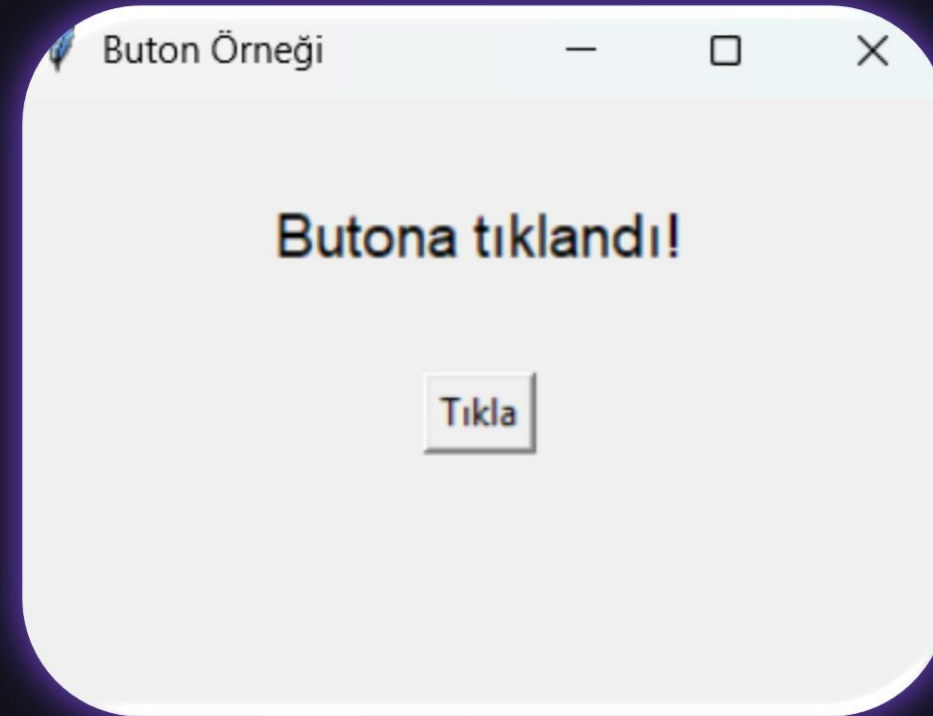
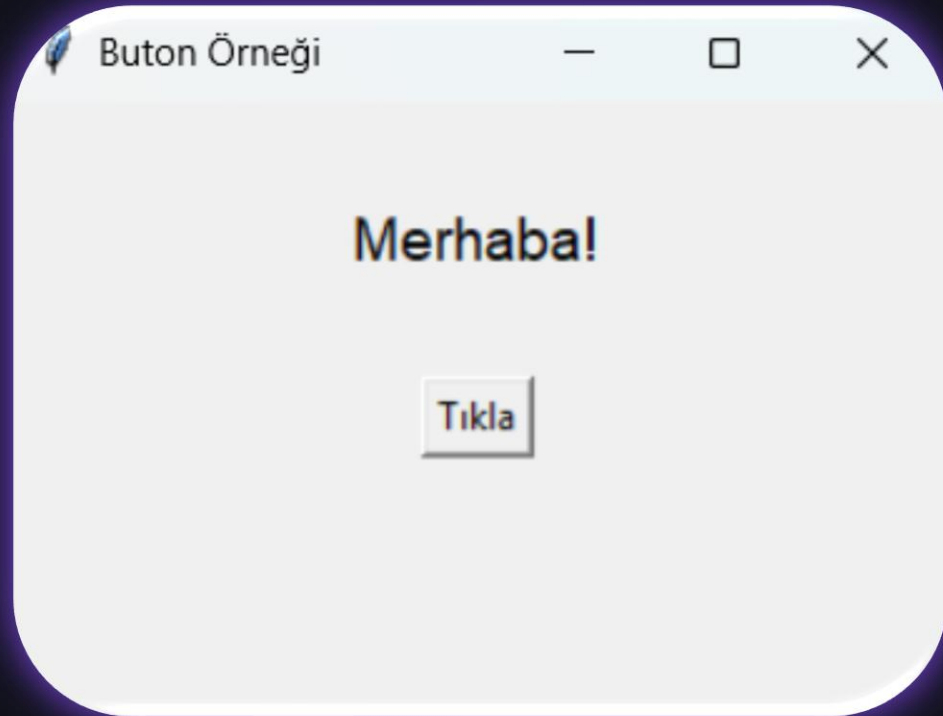
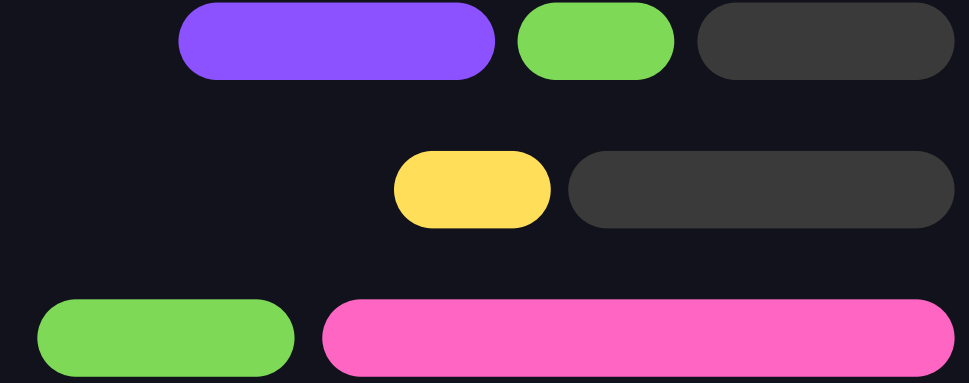
buton = tk.Button(pencere, text="Tıkla", command=tikla)
buton.pack()

pencere.mainloop()
```



03

Tkinter



```
import tkinter as tk

def tikla():
    etiket.config(text="Butona tıklandı!")

pencere = tk.Tk()
pencere.title("Buton Örneği")
pencere.geometry("300x200")

etiket = tk.Label(pencere, text="Merhaba!", font=("Arial", 14))
etiket.pack(pady=30)

buton = tk.Button(pencere, text="Tıkla", command=tikla)
buton.pack()

pencere.mainloop()
```



03

Tkinter

Finans Takip

May 2026

CSV Dışa Aktar

YENİ İŞLEM

Gelir Gider

Açıklama

Miktar (₺)

Tarih (GG.AA.YYYY)

13.05.2026

Kategori

Kira

+ İşlem Ekle

Son 12 Ay Özeti

Gelir Gider

1417K 1063K 709K 354K

2025 2025 2025 2025 2025 2025 2025 2026 2026 2026 2026 2026

Gider Dağılımı

Kira (100%)

Toplam

₺1,232,464.00

İŞLEMLER

Tümü Tümü

4 İşlem

Tarih	Açıklama	Kategori	Tür	Miktar
2026-05-13	ek gelir	Diğer Gelir	Gelir	+₺1,500.00
2026-05-13	asd	Kira	Gider	-₺1,111.00
2026-05-13	a	Kira	Gider	-₺1,231,231.00
2026-05-13	123	Kira	Gider	-₺122.00

BU AY

Gelir

₺1,500.00

Gider

₺1,232,464.00

Bakiye

₺-1,230,964.00



03

Tkinter

Tkinter, Python programlama diliyle birlikte gelen, grafiksel kullanıcı arayüzü (GUI) geliştirmek için kullanılan standart kütüphanedir. Python ile bir pencere, buton, giriş alanı veya menü oluşturmak istediğinizde ilk başvuracağınız araç genellikle budur.





03

Tkinter

Yandaki kodu çalıştırınız.

```
test.py > ...
1  import tkinter as tk
2  import random
3
4  def uret():
5      label.config(text=str(random.randint(1, 100)))
6
7  pencere = tk.Tk()
8  pencere.title("Rastgele Sayı Üretici")
9  pencere.geometry("300x100")
10
11 tk.Button(pencere, text="Üret", command=uret).pack(pady=20)
12
13 label = tk.Label(pencere, text="")
14 label.pack()
15
16 pencere.mainloop()
```



03

Tkinter

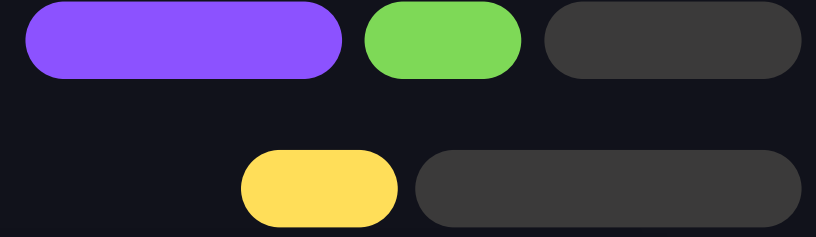


```
test.py > ...
1  import tkinter as tk
2  import random
3
4  def uret():
5      label.config(text=str(random.randint(1, 100)))
6
7  pencere = tk.Tk()
8  pencere.title("Rastgele Sayı Üretici")
9  pencere.geometry("300x100")
10
11 tk.Button(pencere, text="Üret", command=uret).pack(pady=20)
12
13 label = tk.Label(pencere, text="")
14 label.pack()
15
16 pencere.mainloop()
```



03

Tkinter



Config -> mevcut widget'ı günceller

Tk.Tk() -> Pencere oluşturur

Title -> Oluşturulan pencerenin başlığı

Geometry -> Pencerenin başlangıç boyutu

Button -> Buton oluşturur

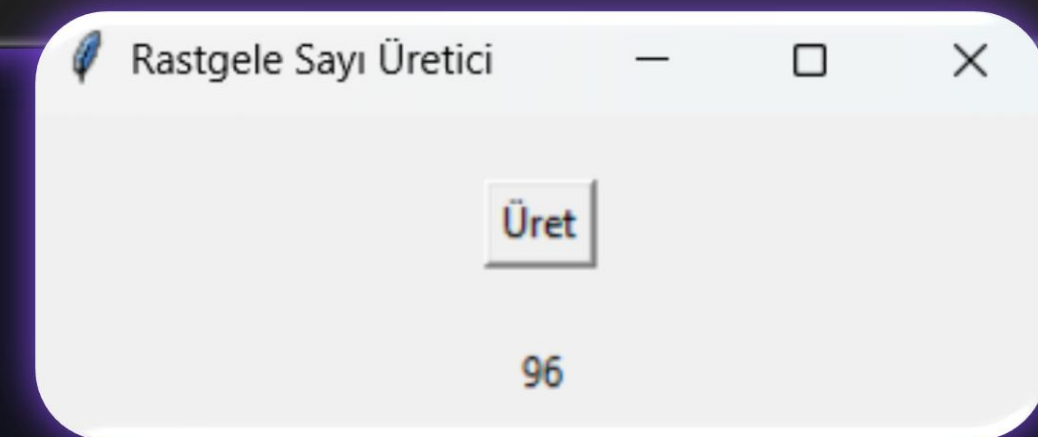
Pack -> Oluşan widget'ı pencereye ekler

Pady -> Html'deki margin

Label -> Pencereye text alanı ekler

Mainloop -> Bu şartlarda pencereyi oluşturur

```
test.py > ...
1 import tkinter as tk
2 import random
3
4 def uret():
5     label.config(text=str(random.randint(1, 100)))
6
7 pencere = tk.Tk()
8 pencere.title("Rastgele Sayı Üretici")
9 pencere.geometry("300x100")
10
11 tk.Button(pencere, text="Üret", command=uret).pack(pady=20)
12
13 label = tk.Label(pencere, text="")
14 label.pack()
15
16 pencere.mainloop()
```





03

Tkinter

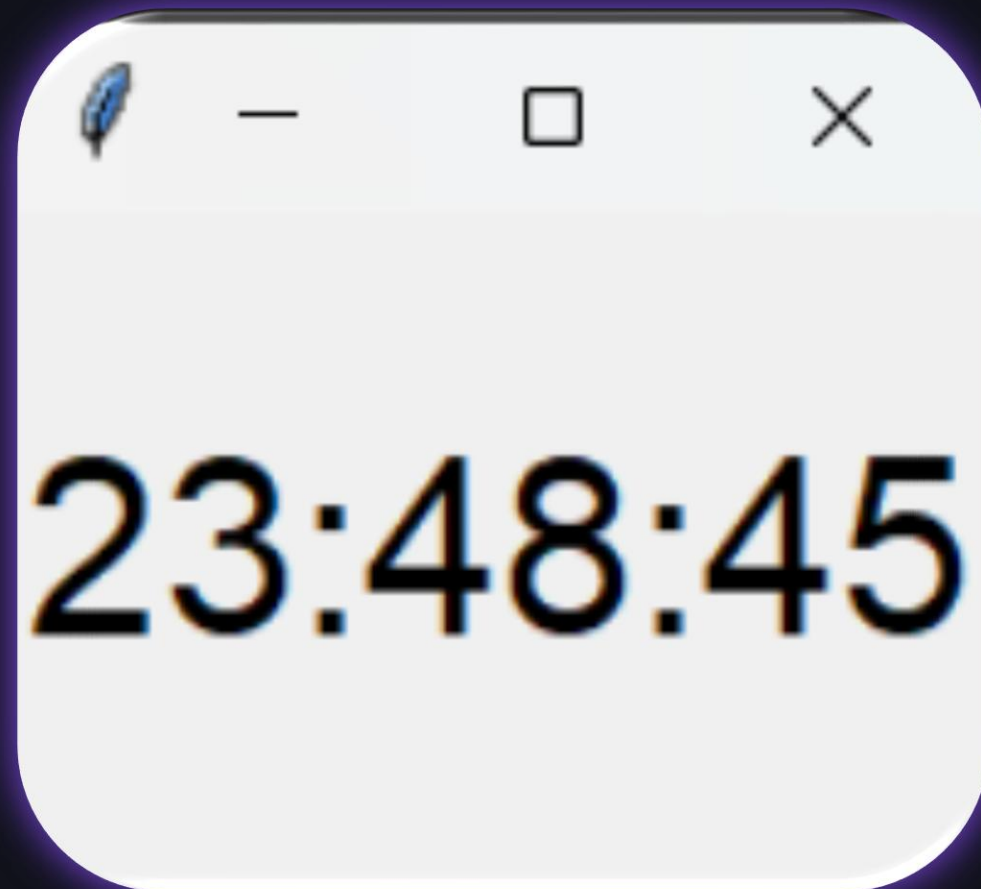
Yandaki kodu çalıştırınız.

```
test.py > ...
1 import tkinter as tk
2 from datetime import datetime
3
4 def saat():
5     simdi = datetime.now().strftime("%H:%M:%S")
6     label.config(text=simdi)
7     pencere.after(1000, saat)
8
9 pencere = tk.Tk()
10
11 label = tk.Label(pencere, font=("Arial", 30))
12 label.pack(pady=30)
13
14 saat()
15
16 pencere.mainloop()
```



03

Tkinter



```
test.py > ...
1  import tkinter as tk
2  from datetime import datetime
3
4  def saat():
5      simdi = datetime.now().strftime("%H:%M:%S")
6      label.config(text=simdi)
7      pencere.after(1000, saat)
8
9  pencere = tk.Tk()
10
11  label = tk.Label(pencere, font=("Arial", 30))
12  label.pack(pady=30)
13
14  saat()
15
16  pencere.mainloop()
```



03

Tkinter

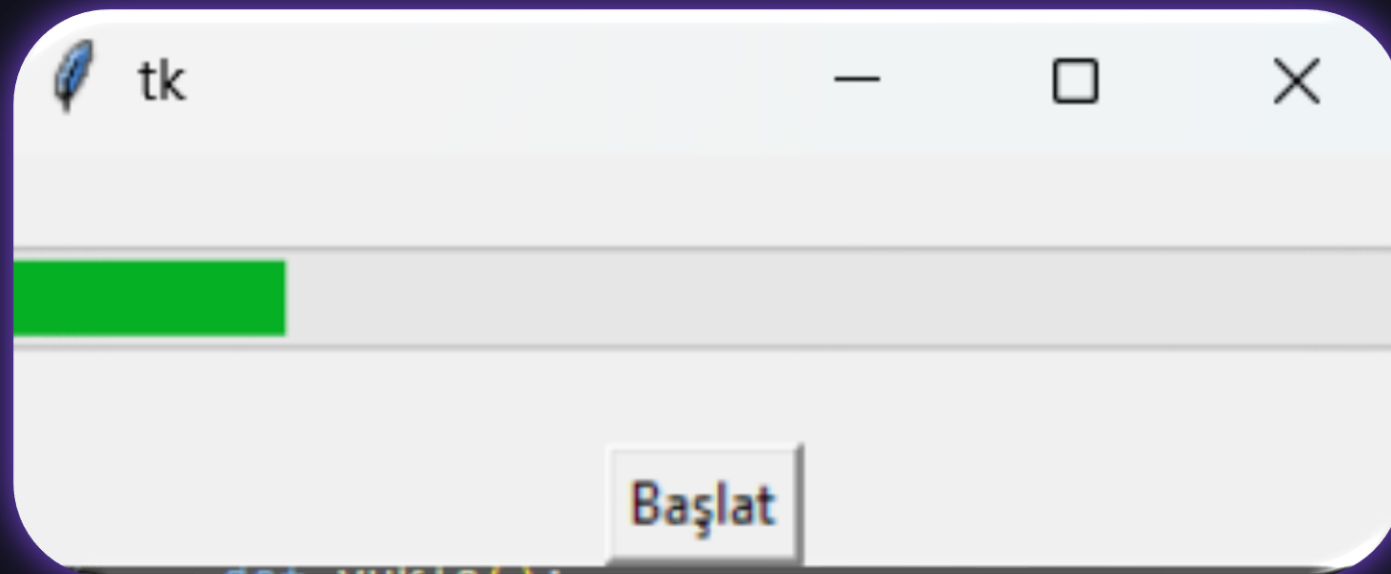
Yandaki kodu çalıştırınız.

```
test.py > ...
1  import tkinter as tk
2  from tkinter import ttk
3
4  deger = 0
5
6  def yukle():
7      global deger
8
9      if deger < 100:
10         deger += 10
11         bar.config(value=deger)
12         pencere.after(500, yukle)
13
14  pencere = tk.Tk()
15
16  bar = ttk.Progressbar(pencere, length=300)
17  bar.pack(pady=20)
18
19  tk.Button(pencere, text="Başlat", command=yukle).pack()
20
21  pencere.mainloop()
```



03

Tkinter

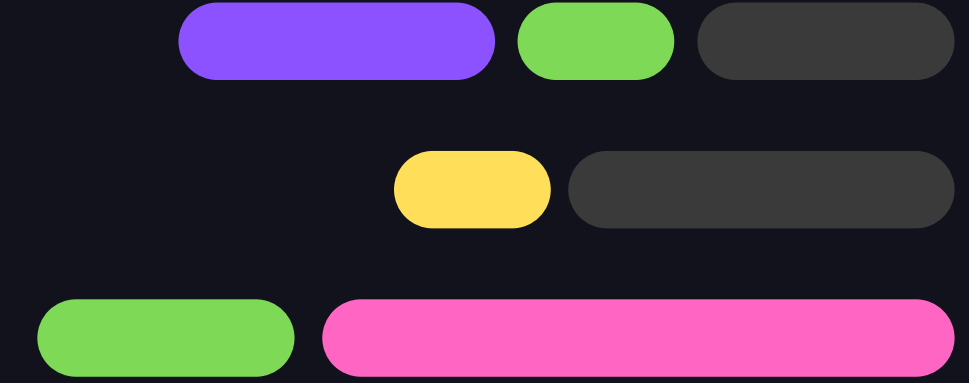


```
test.py > ...
1  import tkinter as tk
2  from tkinter import ttk
3
4  deger = 0
5
6  def yukle():
7      global deger
8
9      if deger < 100:
10         deger += 10
11         bar.config(value=deger)
12         pencere.after(500, yukle)
13
14  pencere = tk.Tk()
15
16  bar = ttk.Progressbar(pencere, length=300)
17  bar.pack(pady=20)
18
19  tk.Button(pencere, text="Başlat", command=yukle).pack()
20
21  pencere.mainloop()
```



03

Tkinter



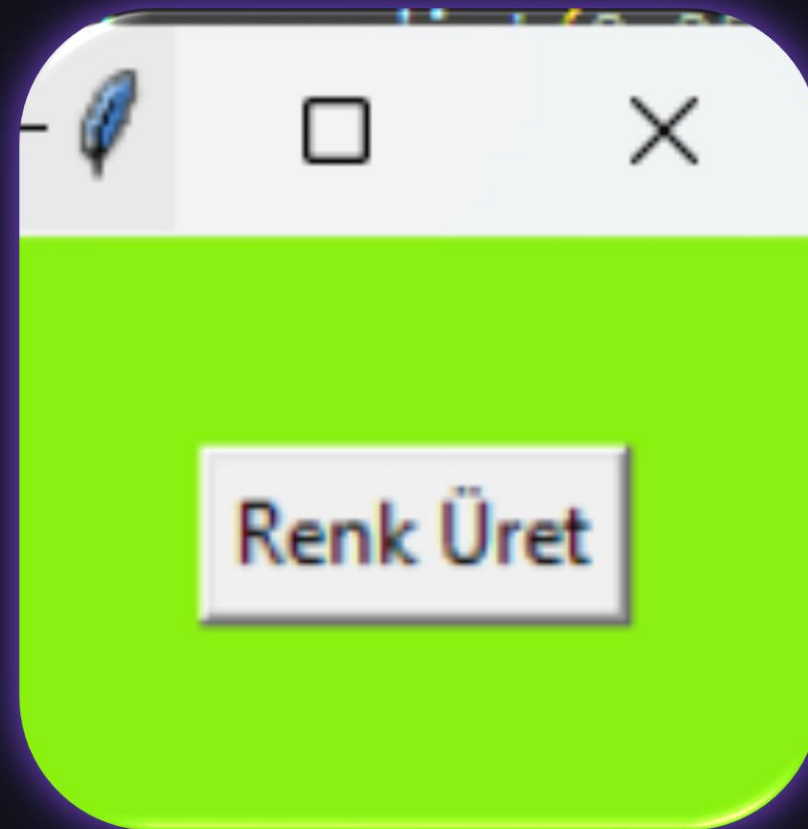
Yandaki kodu çalıştırınız.

```
test.py > ...
1  import tkinter as tk
2  import random
3
4  def renk():
5      r = random.randint(0,255)
6      g = random.randint(0,255)
7      b = random.randint(0,255)
8
9      renk_kodu = f"#{r:02x}{g:02x}{b:02x}"
10
11     pencere.config(bg=renk_kodu)
12
13     pencere = tk.Tk()
14
15     tk.Button(pencere, text="Renk Üret", command=renk).pack(pady=30)
16
17     pencere.mainloop()
18
```



03

Tkinter



```
test.py > ...
1  import tkinter as tk
2  import random
3
4  def renk():
5      r = random.randint(0,255)
6      g = random.randint(0,255)
7      b = random.randint(0,255)
8
9      renk_kodu = f"#{r:02x}{g:02x}{b:02x}"
10
11     pencere.config(bg=renk_kodu)
12
13     pencere = tk.Tk()
14
15     tk.Button(pencere, text="Renk Üret", command=renk).pack(pady=30)
16
17     pencere.mainloop()
18
```



03

Tkinter

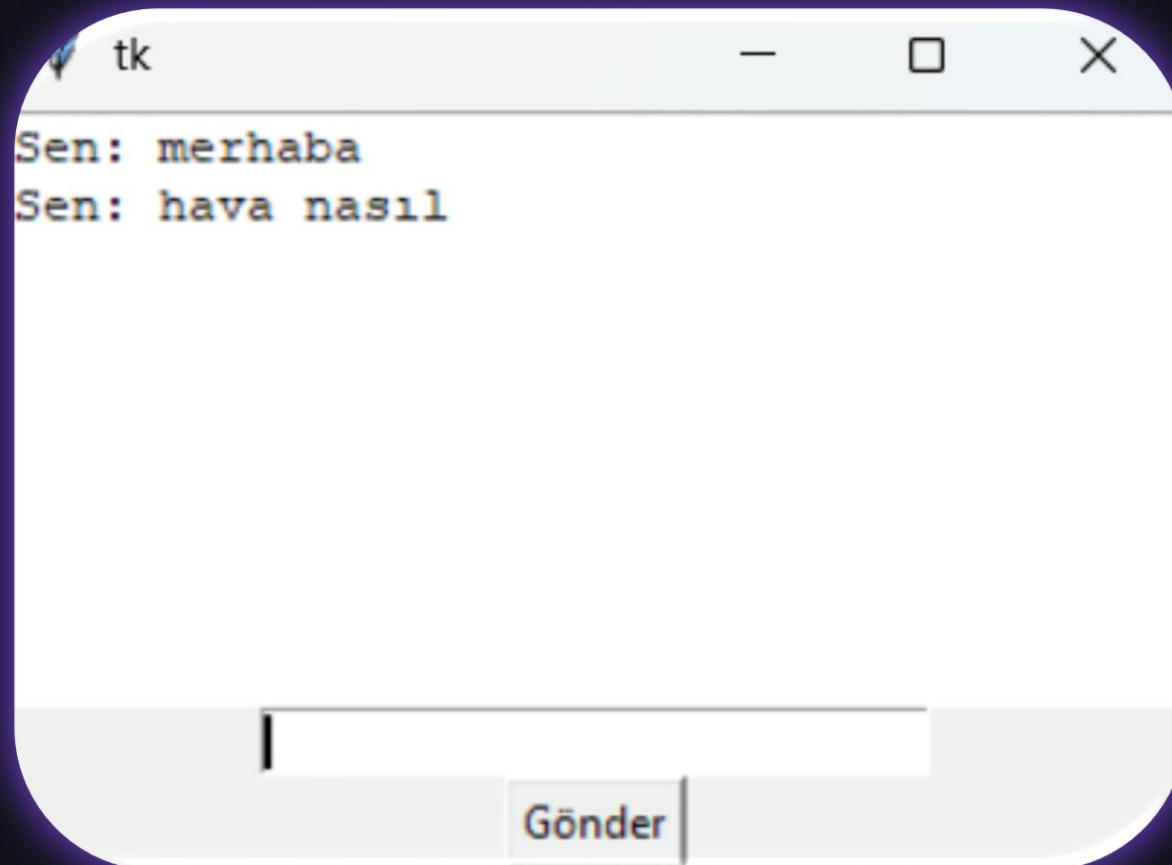
Yandaki kodu çalıştırınız.

```
1 import tkinter as tk
2
3 def gonder():
4     mesaj = entry.get()
5     alan.insert(tk.END, "Sen: " + mesaj + "\n")
6     entry.delete(0, tk.END)
7
8 pencere = tk.Tk()
9
10 alan = tk.Text(pencere, height=10, width=40)
11 alan.pack()
12
13 entry = tk.Entry(pencere, width=30)
14 entry.pack()
15
16 tk.Button(pencere, text="Gönder", command=gonder).pack()
17
18 pencere.mainloop()
```



03

Tkinter



```
1 import tkinter as tk
2
3 def gonder():
4     mesaj = entry.get()
5     alan.insert(tk.END, "Sen: " + mesaj + "\n")
6     entry.delete(0, tk.END)
7
8 pencere = tk.Tk()
9
10 alan = tk.Text(pencere, height=10, width=40)
11 alan.pack()
12
13 entry = tk.Entry(pencere, width=30)
14 entry.pack()
15
16 tk.Button(pencere, text="Gönder", command=gonder).pack()
17
18 pencere.mainloop()
```



03

Tkinter

Yandaki kodu çalıştırınız.

```
import tkinter as tk

sayi = 0

def arttir():
    global sayi
    sayi += 1
    label.config(text=sayi)

pencere = tk.Tk()
pencere.title("Sayı Arttırıcı")
pencere.geometry("200x100")

label = tk.Label(pencere, text="0")
label.pack(pady=10)

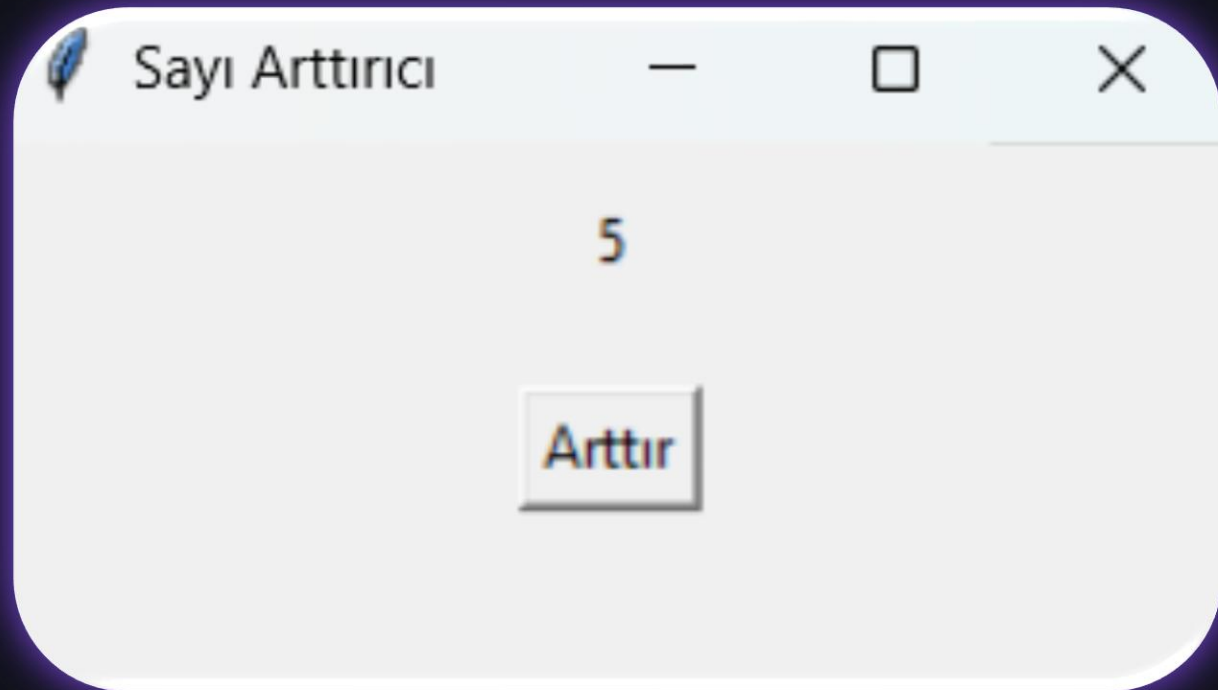
button = tk.Button(pencere, text="Arttır", command=arttir)
button.pack(pady=10)

pencere.mainloop()
```



03

Tkinter



```
import tkinter as tk

sayi = 0

def arttir():
    global sayi
    sayi += 1
    label.config(text=sayi)

pencere = tk.Tk()
pencere.title("Sayı Arttırıcı")
pencere.geometry("200x100")

label = tk.Label(pencere, text="0")
label.pack(pady=10)

button = tk.Button(pencere, text="Arttır", command=arttir)
button.pack(pady=10)

pencere.mainloop()
```



03

Tkinter

Yandaki kodu çalıştırınız.

```
import tkinter as tk

def say():
    text = entry.get()
    label.config(text=len(text))

pencere = tk.Tk()
pencere.title("Karakter Sayıcı")
pencere.geometry("250x150")

entry = tk.Entry(pencere)
entry.pack(pady=10)

tk.Button(pencere, text="Say", command=say).pack(pady=10)

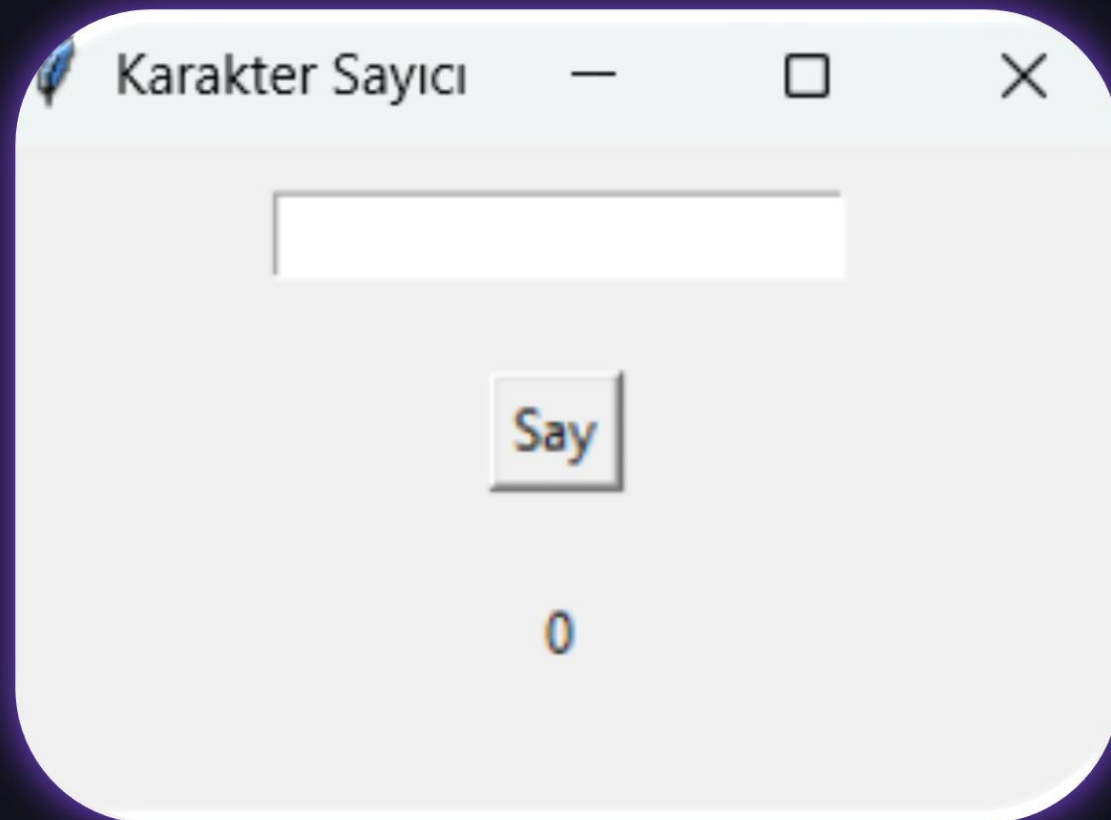
label = tk.Label(pencere, text="0")
label.pack(pady=10)

pencere.mainloop()
```



03

Tkinter



```
import tkinter as tk

def say():
    text = entry.get()
    label.config(text=len(text))

pencere = tk.Tk()
pencere.title("Karakter Sayıcı")
pencere.geometry("250x150")

entry = tk.Entry(pencere)
entry.pack(pady=10)

tk.Button(pencere, text="Say", command=say).pack(pady=10)

label = tk.Label(pencere, text="0")
label.pack(pady=10)

pencere.mainloop()
```



03

Tkinter

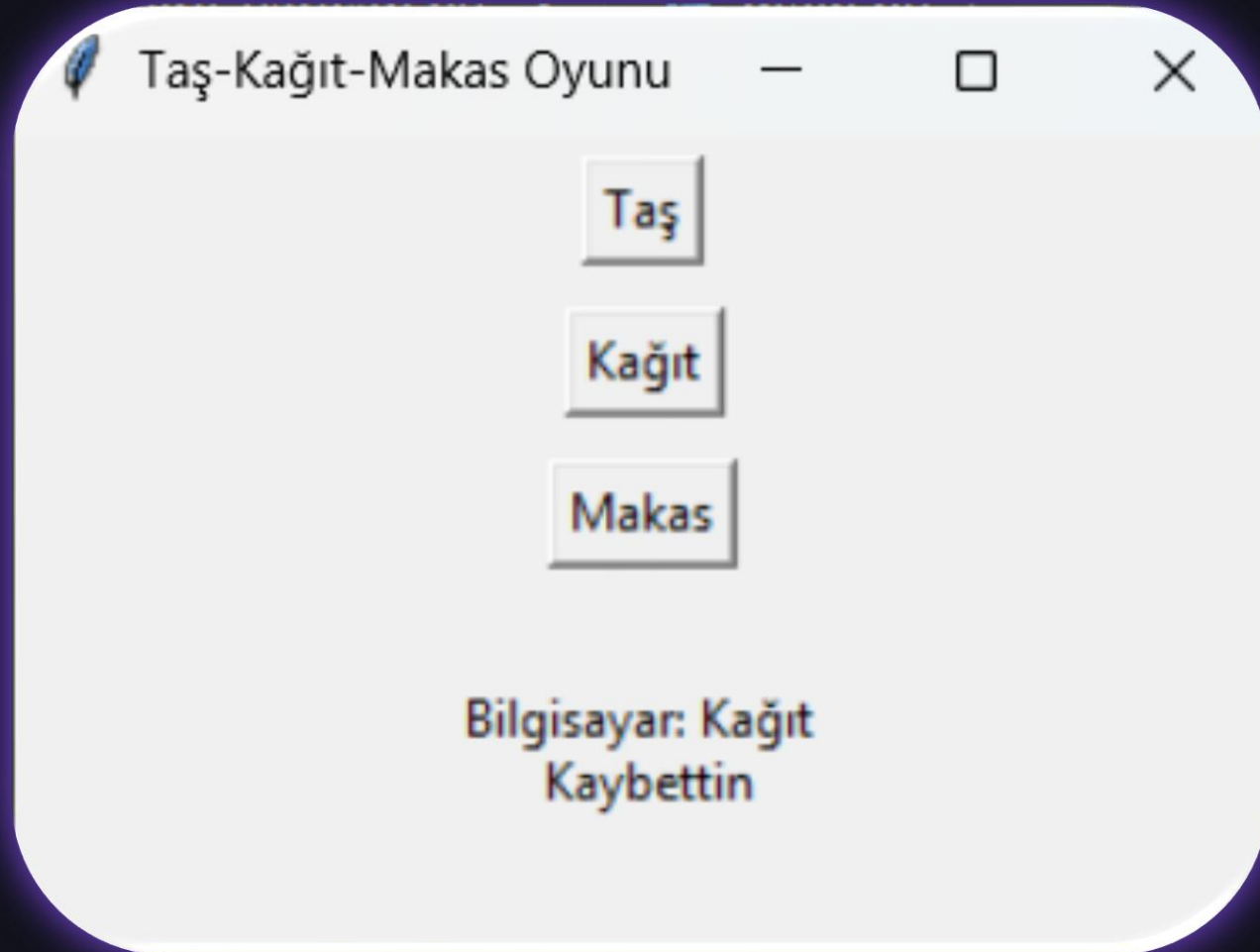
Yandaki kodu çalıştırınız.

```
st.py > ...
1 import tkinter as tk
2 import random
3
4 secenekler = ["Taş", "Kağıt", "Makas"]
5
6 def oyna(secim):
7     pc = random.choice(secenekler)
8
9     if secim == pc:
10        sonuc.config(text=f"Bilgisayar: {pc} \n Berabere")
11    elif (
12        (secim == "Taş" and pc == "Makas") or
13        (secim == "Kağıt" and pc == "Taş") or
14        (secim == "Makas" and pc == "Kağıt")
15    ):
16        sonuc.config(text=f"Bilgisayar: {pc} \n Kazandın")
17    else:
18        sonuc.config(text=f"Bilgisayar: {pc} \n Kaybettin")
19
20 pencere = tk.Tk()
21 pencere.title("Taş-Kağıt-Makas Oyunu")
22 pencere.geometry("300x200")
23
24 tk.Button(pencere, text="Taş", command=lambda: oyna("Taş")).pack(pady=5)
25 tk.Button(pencere, text="Kağıt", command=lambda: oyna("Kağıt")).pack(pady=5)
26 tk.Button(pencere, text="Makas", command=lambda: oyna("Makas")).pack(pady=5)
27
28 sonuc = tk.Label(pencere, text="")
29 sonuc.pack(pady=20)
30
31 pencere.mainloop()
```



03

Tkinter



```
1 import tkinter as tk
2 import random
3
4 secenekler = ["Taş", "Kağıt", "Makas"]
5
6 def oyna(secim):
7     pc = random.choice(secenekler)
8
9     if secim == pc:
10        sonuc.config(text=f"Bilgisayar: {pc} \n Berabere")
11    elif (
12        (secim == "Taş" and pc == "Makas") or
13        (secim == "Kağıt" and pc == "Taş") or
14        (secim == "Makas" and pc == "Kağıt")
15    ):
16        sonuc.config(text=f"Bilgisayar: {pc} \n Kazandın")
17    else:
18        sonuc.config(text=f"Bilgisayar: {pc} \n Kaybettin")
19
20 pencere = tk.Tk()
21 pencere.title("Taş-Kağıt-Makas Oyunu")
22 pencere.geometry("300x200")
23
24 tk.Button(pencere, text="Taş", command=lambda: oyna("Taş")).pack(pady=5)
25 tk.Button(pencere, text="Kağıt", command=lambda: oyna("Kağıt")).pack(pady=5)
26 tk.Button(pencere, text="Makas", command=lambda: oyna("Makas")).pack(pady=5)
27
28 sonuc = tk.Label(pencere, text="")
29 sonuc.pack(pady=20)
30
31 pencere.mainloop()
```



03

Tkinter

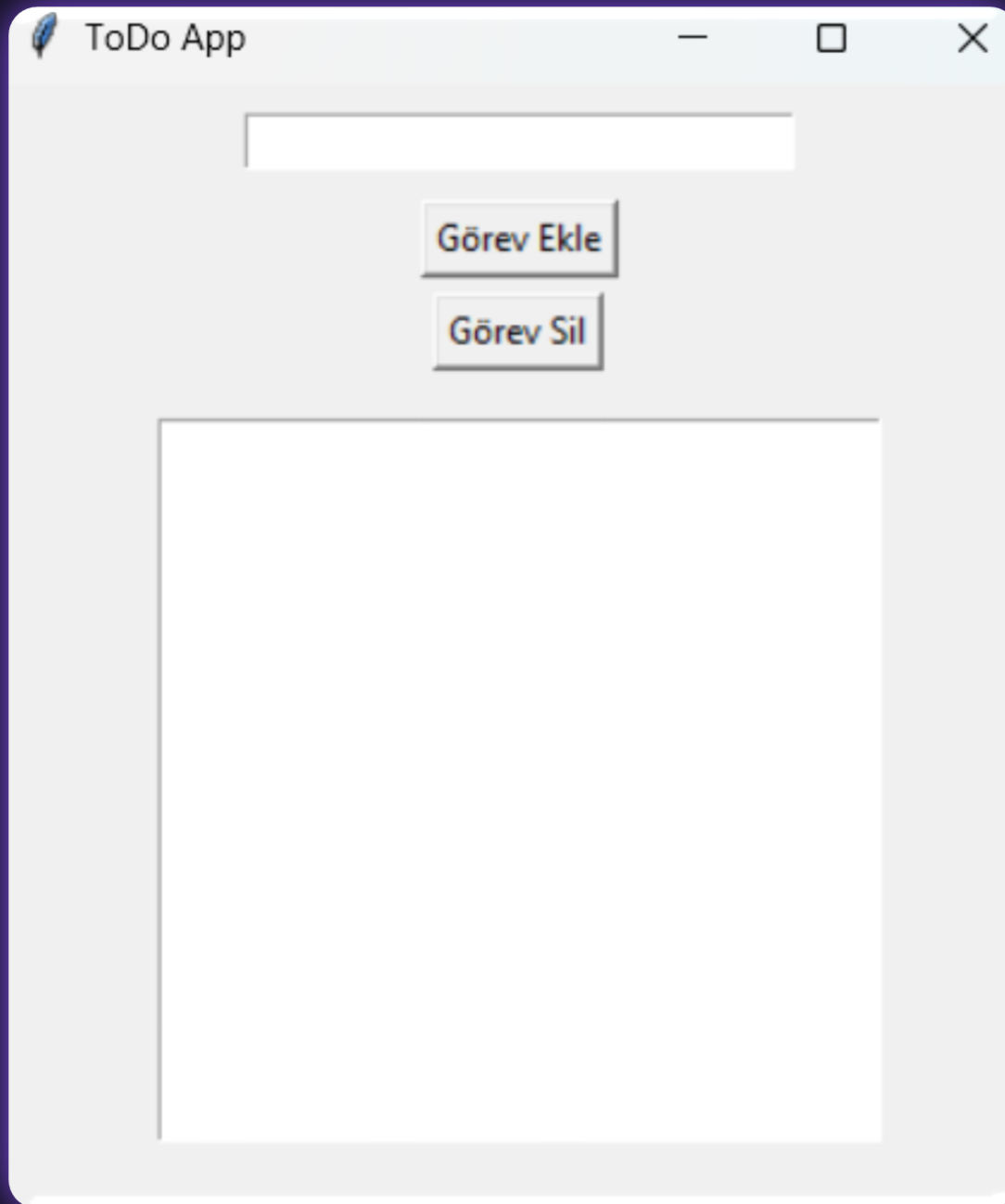
Yandaki kodu çalıştırınız.

```
import tkinter as tk
2
3 def gorev_ekle():
4     gorev = entry.get()
5
6     if gorev != "":
7         liste.insert(tk.END, gorev)
8         entry.delete(0, tk.END)
9
10 def gorev_sil():
11     secili = liste.curselection()
12     if secili:
13         liste.delete(secili)
14
15
16 pencere = tk.Tk()
17 pencere.title("ToDo App")
18 pencere.geometry("350x400")
19 entry = tk.Entry(pencere, width=30)
20 entry.pack(pady=10)
21 tk.Button(pencere, text="Görev Ekle", command=gorev_ekle).pack()
22 tk.Button(pencere, text="Görev Sil", command=gorev_sil).pack(pady=5)
23 liste = tk.Listbox(pencere, width=40, height=15)
24 liste.pack(pady=10)
25 pencere.mainloop()
```



03

Tkinter



```
import tkinter as tk
2
3 def gorev_ekle():
4     gorev = entry.get()
5
6     if gorev != "":
7         liste.insert(tk.END, gorev)
8         entry.delete(0, tk.END)
9
10 def gorev_sil():
11     secili = liste.curselection()
12     if secili:
13         liste.delete(secili)
14
15
16 pencere = tk.Tk()
17 pencere.title("ToDo App")
18 pencere.geometry("350x400")
19 entry = tk.Entry(pencere, width=30)
20 entry.pack(pady=10)
21 tk.Button(pencere, text="Görev Ekle", command=gorev_ekle).pack()
22 tk.Button(pencere, text="Görev Sil", command=gorev_sil).pack(pady=5)
23 liste = tk.Listbox(pencere, width=40, height=15)
24 liste.pack(pady=10)
25 pencere.mainloop()
```



03

Tkinter

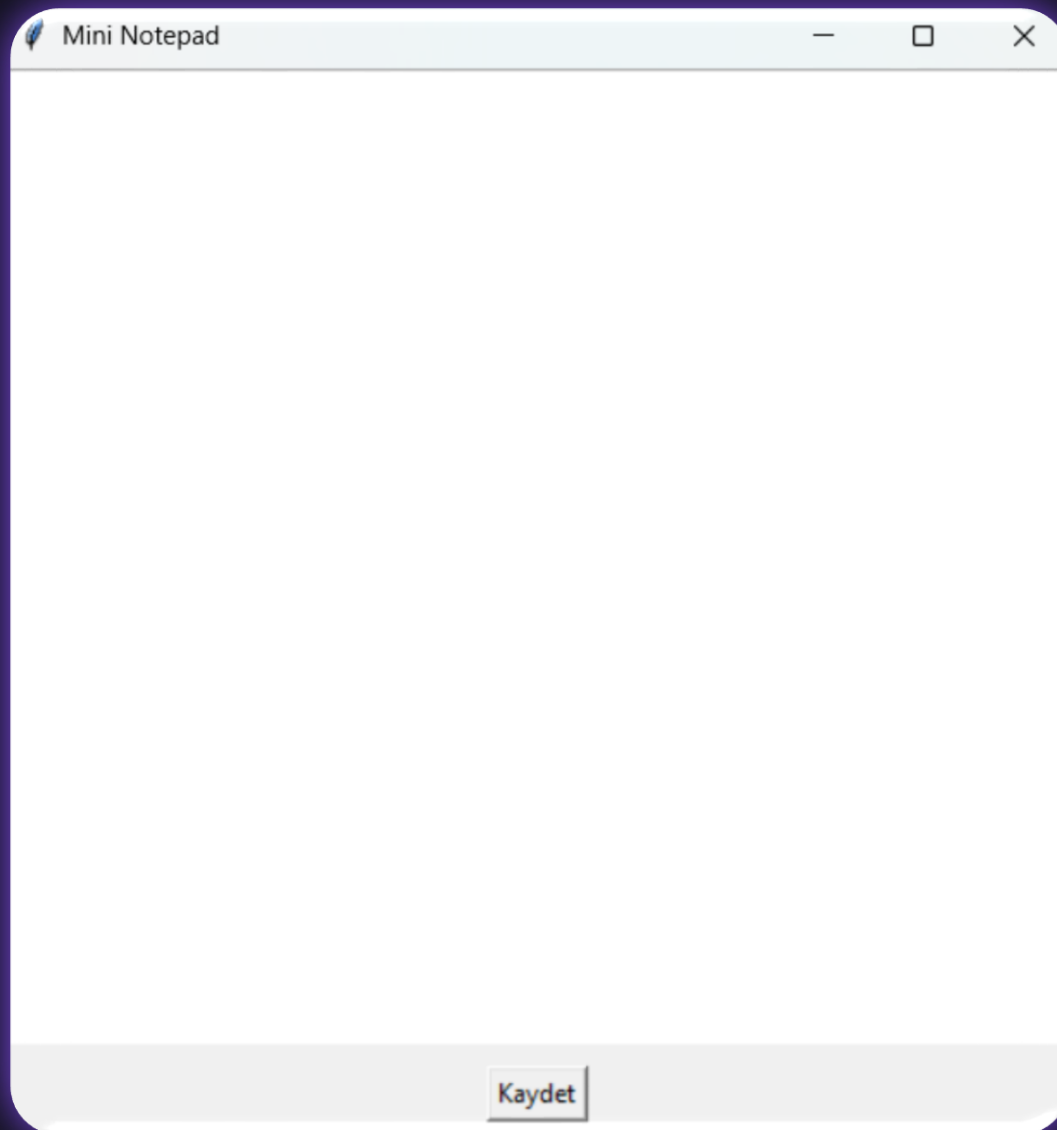
Yandaki kodu çalıştırınız.

```
test.py > ...
1  import tkinter as tk
2  from tkinter import filedialog
3
4  def kaydet():
5      dosya = filedialog.asksaveasfile(
6          defaultextension=".txt"
7      )
8
9      if dosya:
10         dosya.write(text_alani.get("1.0", tk.END))
11         dosya.close()
12
13
14     pencere = tk.Tk()
15     pencere.title("Mini Notepad")
16     pencere.geometry("500x500")
17     text_alani = tk.Text(pencere)
18     text_alani.pack(expand=True, fill="both")
19     tk.Button(pencere, text="Kaydet", command=kaydet).pack(pady=10)
20     pencere.mainloop()
```



03

Tkinter



```
test.py > ...
1  import tkinter as tk
2  from tkinter import filedialog
3
4  def kaydet():
5      dosya = filedialog.asksaveasfile(
6          defaultextension=".txt"
7      )
8
9      if dosya:
10         dosya.write(text_alani.get("1.0", tk.END))
11         dosya.close()
12
13
14  pencere = tk.Tk()
15  pencere.title("Mini Notepad")
16  pencere.geometry("500x500")
17  text_alani = tk.Text(pencere)
18  text_alani.pack(expand=True, fill="both")
19  tk.Button(pencere, text="Kaydet", command=kaydet).pack(pady=10)
20  pencere.mainloop()
```



03

Tkinter

Terminalde sırayla alttaki kütüphaneleri yükleyin:

```
pip install qrcode
```

```
pip install qrcode[pil]
```

Ardından

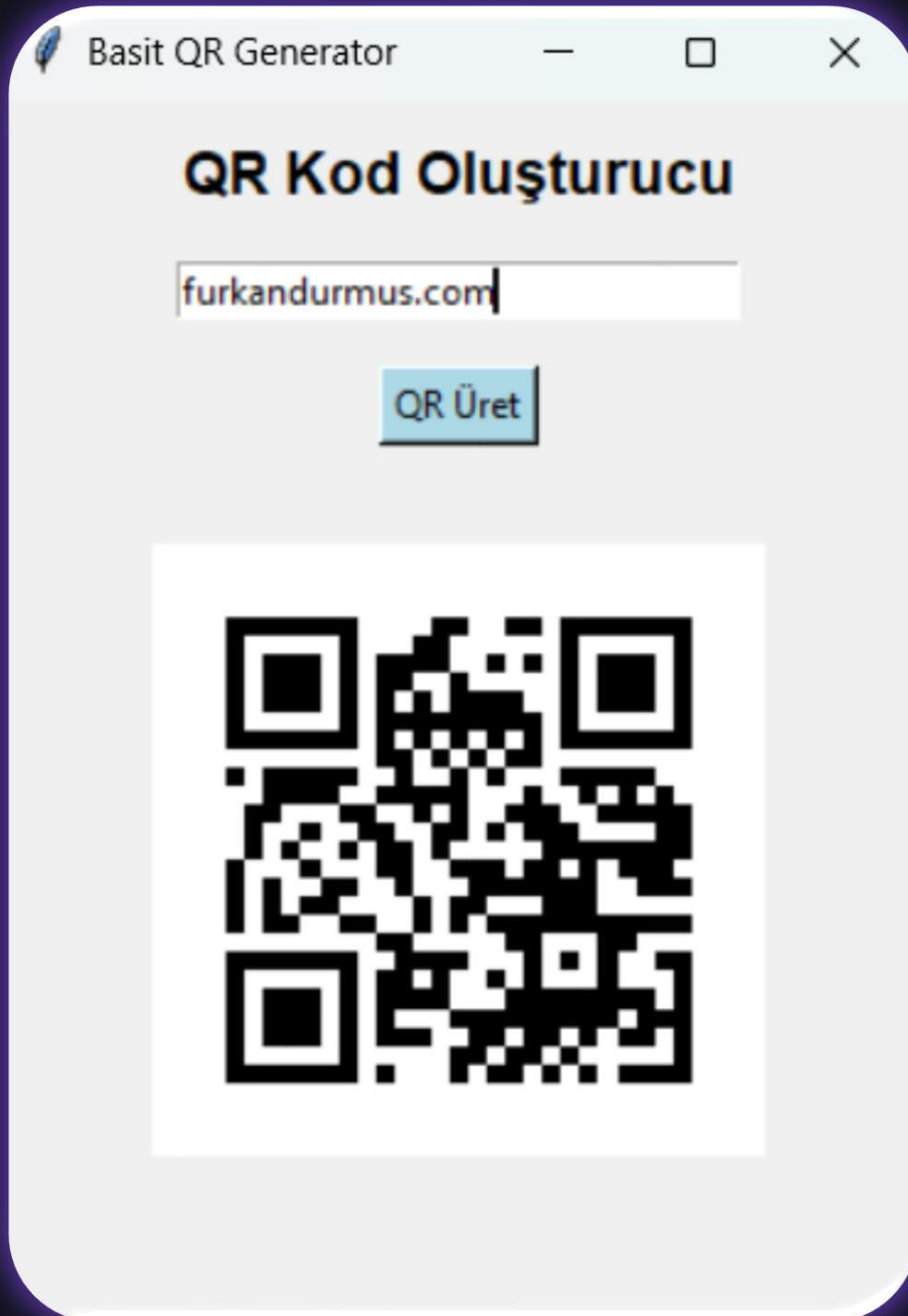
Yandaki kodu çalıştırınız.

```
test.py > ...
1 import tkinter as tk
2 from tkinter import messagebox
3 import qrcode
4 from PIL import ImageTk
5
6 def qr_olustur():
7     veri = giris_kutusu.get()
8     if veri:
9         img = qrcode.make(veri)
10        img = img.resize((200, 200))
11        tk_img = ImageTk.PhotoImage(img)
12        qr_etiket.config(image=tk_img)
13        qr_etiket.image = tk_img
14    else:
15        messagebox.showwarning("Uyarı", "Lütfen bir metin veya URL girin!")
16
17 pencere = tk.Tk()
18 pencere.title("Basit QR Generator")
19 pencere.geometry("300x400")
20
21 baslik = tk.Label(pencere, text="QR Kod Oluşturucu", font=("Arial", 14, "bold"))
22 baslik.pack(pady=10)
23
24 giris_kutusu = tk.Entry(pencere, width=30)
25 giris_kutusu.pack(pady=5)
26
27 buton = tk.Button(pencere, text="QR Üret", command=qr_olustur, bg="lightblue")
28 buton.pack(pady=10)
29
30 qr_etiket = tk.Label(pencere)
31 qr_etiket.pack(pady=20)
32
33 pencere.mainloop()
```



03

Tkinter



```
test.py > ...
1 import tkinter as tk
2 from tkinter import messagebox
3 import qrcode
4 from PIL import ImageTk
5
6 def qr_olustur():
7     veri = giris_kutusu.get()
8     if veri:
9         img = qrcode.make(veri)
10        img = img.resize((200, 200))
11        tk_img = ImageTk.PhotoImage(img)
12        qr_etiket.config(image=tk_img)
13        qr_etiket.image = tk_img
14    else:
15        messagebox.showwarning("Uyarı", "Lütfen bir metin veya URL girin!")
16
17 pencere = tk.Tk()
18 pencere.title("Basit QR Generator")
19 pencere.geometry("300x400")
20
21 baslik = tk.Label(pencere, text="QR Kod Oluşturucu", font=("Arial", 14, "bold"))
22 baslik.pack(pady=10)
23
24 giris_kutusu = tk.Entry(pencere, width=30)
25 giris_kutusu.pack(pady=5)
26
27 buton = tk.Button(pencere, text="QR Üret", command=qr_olustur, bg="lightblue")
28 buton.pack(pady=10)
29
30 qr_etiket = tk.Label(pencere)
31 qr_etiket.pack(pady=20)
32
33 pencere.mainloop()
```



Son...