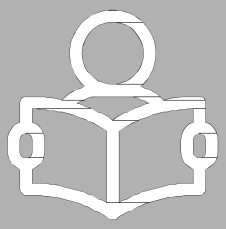
PYTHON

Dosyalarla Çalışmak

Kapsam

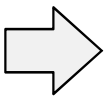


- Dosya İşlemleri
- Dosya Kipleri
- Dosya Okuma Yazma İşlemleri
- Arama ve Sıralama Algoritmaları ile Dosyadan Okuma ve Dosyaya Yazma
- Dosyaların Özel Metotları

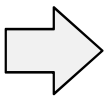
Dosyalar

Dosyalar verilerin kalıcı olarak depolandığı ve ihtiyaç duyulduğunda okunabildiği temel yapılardandır.

- Dosyayı aç
  - Dosya adı
  - Açma modu



- İşlem yap
  - Oku, arat, yaz



- Dosyayı kapat

Dosya İşlemleri

Dosya açma	dosya=open(dosya adresi, dosya açma türü)
Hangi dosyayı açayım?	Cevabı adresini belirttiğiniz dosyadır
Hangi amaç için açayım?	Veri okuma, veri yazma veya ikisinin (okuma ve yazma) birlikte olduğu durumlardır
Dosya kapatma	Dosya ile işimiz bittiği zaman dosyayı close() fonksiyonu ile kapatırız
Ek bilgi	Karakter kodlaması içinde 3. parametre olarak encoding kullanılır. Türkçe karakterlerde sorun yaşamamak için encoding="utf-8" ifadesi eklenir.

Dosya İşlemleri

Dosya işlemlerini yaparken iki farklı yöntem kullanabiliriz. Aşağıda gösterildiği gibi dosyayı kendimizin açıp kendimizin kapattığı yöntem ya da with open parametresi kullanılarak dosyanın sürecin bitiminde otomatik kapandığı yöntem kullanılabilir. İki yöntemi de kullanabilirsiniz.

Dosya işlemleri süreci yöntemleri

1

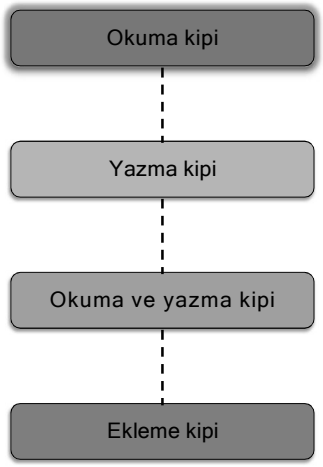
```
dosya=open(dosya yolu, açma kipi)
işlemler
dosya.close ()
```

2

```
with open(dosya yolu, açma kipi) as dosya:
    işlemler # girinti kullanmaya dikkat
```

Dosya işlemlerine başlamadan önce ilerleyen örneklerde kullanılmak üzere deneme.txt, veri.txt gibi dosyaları Python kodlarının olduğu ve .py uzantılı olan kod dosyası ile aynı klasörde oluşturunuz. Böylece dosyanın sadece adını yazarak bilgisayarda bulunduğu yeri yazmaya ihtiyaç duymadan açabiliriz.

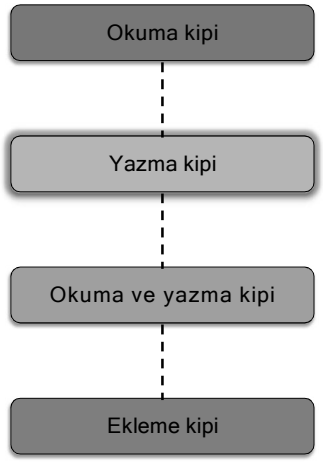
Dosya Kipleri



Dosyadan sadece veri okumak istiyor isek r (read) kipinde açarız.
Dosya yok ise hata verir.

```
dosya=open("deneme.txt","r")
işlemler
dosya.close( )
```

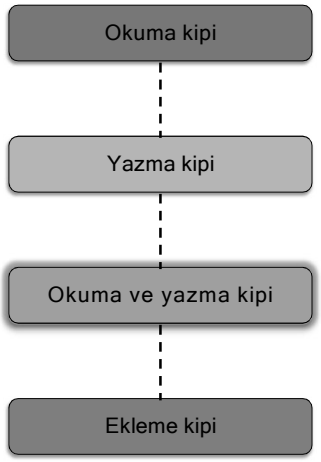
Dosya Kipleri



Dosyaya sadece veri kaydetmek/yazmak istiyorsak w (write) kipinde açarız.
Dosya yoksa oluşturulur, varsa dosya oluşturulur ve eski dosya silinir.
Eski dosyadaki bilgiler gider.

```
dosya=open("deneme.txt","w")
işlemler
dosya.close( )
```

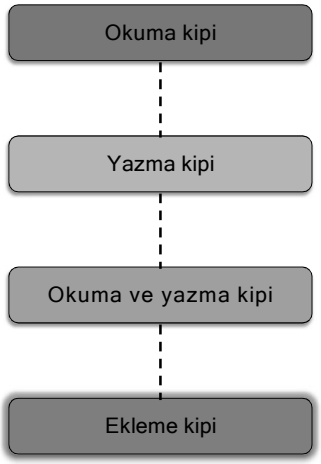
Dosya Kipleri



Dosyadan hem veri okumak hem de dosyaya veri yazmak istiyorsak r+ kipinde açarız.
Dosya yoksa hata üretilir, varsa dosya oluşturulmaz.
Dosyanın istenen bölümlerine eklemeler yapılır.
Eski bilgiler silinmez.

```
dosya=open("deneme.txt","r+")
işlemler
dosya.close( )
```

Dosya Kipleri



Dosyadan hem veri okumak hem de dosyanın sonuna veri yazmak istiyor isek a (append) kipinde açarız.
Dosya yoksa oluşturulur, varsa dosya oluşturulmaz.
Dosyanın sonuna ekleme yapılır.
Eski bilgiler silinmez.

```
dosya=open("deneme.txt","a")
işlemler
dosya.close( )
```

Dosya Okuma Yazma İşlemleri

Bu bölümde dosya okuma ve yazma örnekleri bulunmaktadır. Öncelikle deneme.txt isimli bir dosya oluşturarak içine "bu basit bir dosyadır" cümlesini yazmalısınız.
Örnekleri uyguladıktan sonra deneme.txt dosyasında nelerin değiştiğini görmek için dosyayı açmalısınız. Dosyayı açıp inceledikten sonra kapatmayı unutmayınız.

Spyder editörde çalışma dizinini deneme.txt dosyasının bulunduğu dizine getirmeyi unutmayınız.

Not: Python'da dosyalara yazılan veriler string tipinde kaydedilmektedir.

Dosyayı okuma kipinde açma

İlk olarak oluşturduğumuz deneme.txt dosyası okuma kipinde açılacak ve içindeki verileri ekranda görüntülenecektir.

Kod

```
dosya = open("deneme.txt","r")
belge=dosya.read()
print(belge)
dosya.close()
```

Dosyayı yazma kipinde açma

İkinci aşamada deneme.txt dosyası yazma kipinde açılıp içine veri kaydedilecektir. Yazma işlemini yaptıktan sonra dosyada önceden bulunan verilerin silindiğini fark ettiniz mi?

Kod

```
dosya = open("deneme.txt","w")
dosya.write("Dosya silinip sıfırdan yazıldı.")
dosya.close()
```

Dosyayı okuma ve yazma kipinde açma

Şimdi dosyayı okuma ve yazma kipinde açıp neler olduğunu inceleyim.

Kod

```
dosya = open("deneme.txt","r+")
belge=dosya.read()
print(belge)
dosya.write("\n bu metin en sona eklendi")
dosya.close()
```

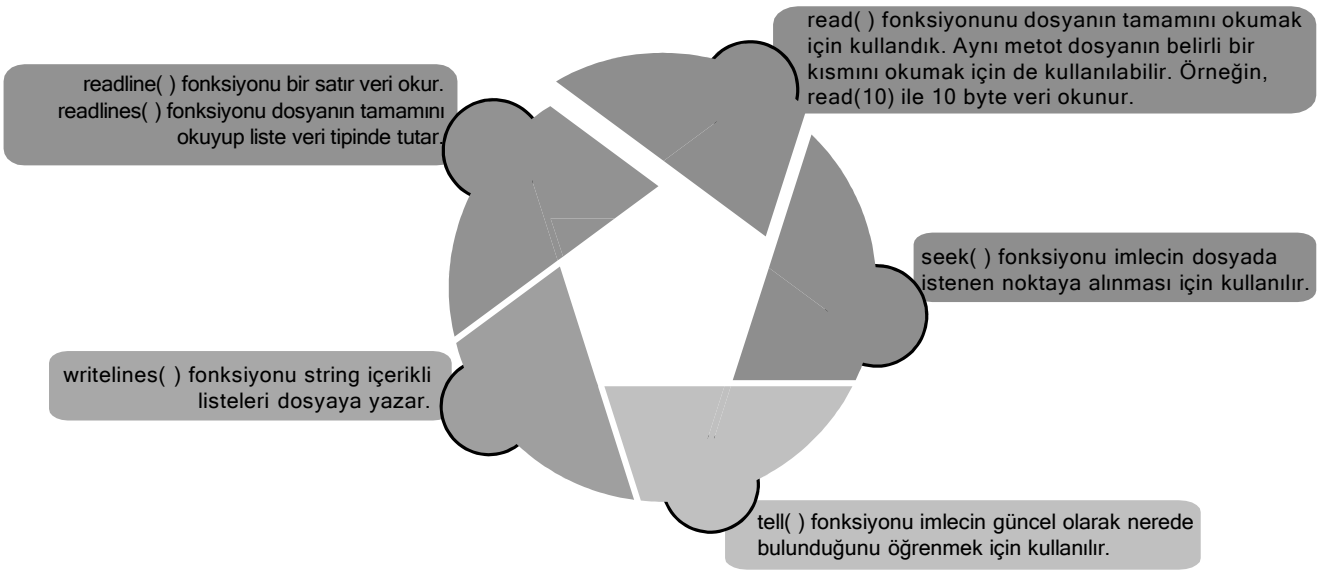
Dosyayı ekleme kipinde açma

Son olarak dosyayı ekleme kipinde açıp neler olduğunu inceleyim.

Kod

```
dosya = open("deneme.txt","a")
dosya.write ("bu metin dosyanın sonuna yazıldı")
dosya.close()
```

Dosyaların Özel Metotları



Örnek 1

1'den 1000'e kadar olan asal sayıları bulup dosyaya kaydeden program.

Kod

```
asal_sayı=[2]
for sayı in range (3,1001):
    for bölen_sayı in range (2,sayı):
        sayı_asalmı=False
        if sayı%bölen_sayı==0:
            sayı_asalmı=True
            break
    if sayı_asalmı==False:
        asal_sayı.append(sayı)
veri=" "
for i in asal_sayı:
    veri+=str(i)
    veri+=" "
dosya=open ("asalsayı.txt","w")
dosya.write(veri)
dosya.close()
```

Örnek 2

1'den 1000 arasında girilen bir sayının asal olup olmadığını dosyadan karşılaştıran program.

Kod

```
with open("asalsayı.txt","r") as dosya :
    veri=dosya.read()
    asal_sayılar=veri.split(" ")
kontrol_sayısı=input("istediğiniz sayıyı giriniz: ")
if kontrol_sayısı in asal_sayılar :
    print("bu bir asal sayıdır.")
else:
    print("bu bir asal sayı değildir.")
```

Örnek 3

tell( ), seek( ), read(girilen bayt) ve for döngüsü ile dosya okuma örneği.

Kod

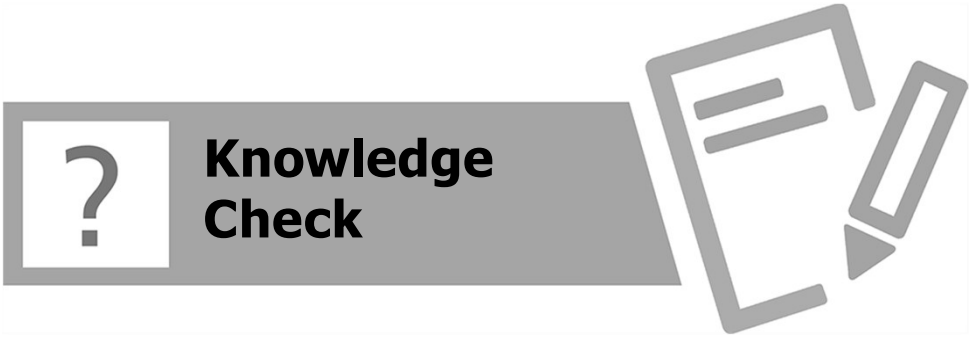
```
dosya=open("deneme.txt","r")
for veri in dosya: # dosyamızı for döngüsü ile okuyoruz
    print(veri)
print(dosya.tell()) #imlecin nerede olduğunu ekrana yazdırıyoruz
dosya.seek(10) #imleci 10. bayta taşıyoruz
print(dosya.read(20)) #imlecin bulunduğu yerden 20 bayt veri okuyoruz
print(dosya.tell()) #imlecin nerede olduğunu görüntülüyoruz
dosya.close()
```

Örnek 4

tell( ), seek( ), writelines( ) uygulaması.

Kod

```
dosya=open("deneme.txt","r+")
dosya.seek(20) #dosyada 20. bayta gittik
dosya.write("20. bayttan itibaren yazdık")
print(dosya.tell()) #imlecin 47. bayta geldiğini öğreneceğiz
print(dosya.read()) #burda dosyadan okuma yaparsak 47. bayttan sonrası okunacak
liste=["1","2","3","4"] #listedeki veriler string olmazsa hata alırız.
dosya.writelines (liste) #listemizdeki verileri en sona yazıyoruz
dosya.close()
```



1	**Elli adet rastgele sayı üretip sayı.txt dosyasına kaydeden program uygulamasını yapınız.**

1	**Elli adet rastgele sayı üretip sayı.txt dosyasına kaydeden program uygulamasını yapınız.**

```
import random
sayılar=[ ]
for i in range (50):
    sayı=str(random.randint(0,1000))+"\n"
    sayılar.append(sayı)
dosya=open("sayı.txt","w")
dosya.writelines(sayılar)
dosya.close()
```

Doğru Cevap

2	**Birinci uygulamadaki rastgele üretilen sayıları alıp büyükten küçüğe doğru sıralayıp sıralı.txt dosyasına kaydeden program uygulamasını yapınız.**

2	**Birinci uygulamadaki rastgele üretilen sayıları alıp büyükten küçüğe doğru sıralayıp sıralı.txt dosyasına kaydeden program uygulamasını yapınız.**

```
dosya=open("sayı.txt","r")
sayılar=dosya.readlines()
dosya.close()
# sayıları string olarak aldık sayıya dönüştürüp sıralayıp tekrardan stringe dönüştüreceğiz
for i in range(len(sayılar)):
    sayılar[i]=int(sayılar[i])
sayılar.sort()
for i in range(len(sayılar)):
    sayılar[i]=str(sayılar[i])+"\n"
dosya=open("sıralı.txt","w")
dosya.writelines(sayılar)
dosya.close()
```

Teşekkürler