

Dosya İşlemleri

Table of contents

Bölüm 7: Dosya İşlemleri - Verilerinizi Kalıcı Hale Getirin	1
7.1 Dosyalar: Bilgileri Saklama ve Paylaşma	1
7.2 Python ile Dosya İşlemleri	2
7.2.1 <code>open()</code> Fonksiyonu	2
7.2.2 Dosyayı Kapatma	3
7.3 Dosya Okuma İşlemleri	3
7.3.1 <code>read()</code> Metodu	3
7.3.2 <code>readline()</code> Metodu	4
7.3.3 <code>readlines()</code> Metodu	4
7.3.4 Döngü ile Dosya Okuma	4
7.4 Dosya Yazma İşlemleri	4
7.4.1 <code>write()</code> Metodu	4
7.4.2 <code>writelines()</code> Metodu	5
7.4.3 Dosya Konumlandırma (File Positioning)	5
7.5 Farklı Dosya Formatları	6
7.6 Ek Örnekler ve Uygulamalar	6
7.7 Bölüm Sonu Quiz	6

Bölüm 7: Dosya İşlemleri - Verilerinizi Kalıcı Hale Getirin

7.1 Dosyalar: Bilgileri Saklama ve Paylaşma

Bilgisayar programları, verileri işlemek için kullanılır. Bu veriler, program çalışırken bilgisayarın belleğinde (RAM) saklanır. Ancak, program sona erdiğinde, bellekteki veriler kaybolur. Verileri kalıcı olarak saklamak ve daha sonra tekrar kullanmak veya paylaşmak için **dosyaları** kullanırız.

Dosyalar, bilgisayarın sabit diskinde veya diğer depolama ortamlarında saklanan veri koleksiyonlarıdır. Dosyalar, metin, resim, müzik, video veya diğer herhangi bir türde veri içerebilir.

7.2 Python ile Dosya İşlemleri

Python, dosyaları okumak, yazmak ve yönetmek için birçok fonksiyon sunar. Bu fonksiyonlar, `open()` fonksiyonu aracılığıyla erişilir.

7.2.1 `open()` Fonksiyonu

`open()` fonksiyonu, bir dosyayı açar ve bir **dosya nesnesi** döndürür. Dosya nesnesi, dosya üzerinde okuma, yazma veya diğer işlemleri yapmak için kullanılır.

Sözdizimi:

```
open(dosya_adi, mod)
```

- **dosya_adi:** Açmak istediğiniz dosyanın adı (ve yolu).
- **mod:** Dosyayı hangi modda açacağınızı belirten bir string. Yaygın olarak kullanılan modlar şunlardır:
 - **'r':** Okuma modu (varsayılan).
 - **'w':** Yazma modu. Eğer dosya yoksa, yeni bir dosya oluşturulur. Eğer dosya varsa, içeriği silinir.
 - **'a':** Ekleme modu. Eğer dosya yoksa, yeni bir dosya oluşturulur. Eğer dosya varsa, yeni veriler dosyanın sonuna eklenir.
 - **'x':** Oluşturma modu. Eğer dosya zaten varsa, `FileExistsError` hatası verir.
 - **'b':** İkili (binary) mod. İkili dosyaları (örneğin, resimler) okumak veya yazmak için kullanılır.
 - **'t':** Metin (text) modu (varsayılan). Metin dosyalarını okumak veya yazmak için kullanılır.

Örnekler:

```
# "dosya.txt" adlı dosyayı okuma modunda açar
dosya = open("dosya.txt", "r")

# "veri.txt" adlı dosyayı yazma modunda açar. Dosya yoksa oluşturur, varsa içeriğini siler.
dosya = open("veri.txt", "w")

# "log.txt" adlı dosyayı ekleme modunda açar. Dosya yoksa oluşturur.
dosya = open("log.txt", "a")
```

7.2.2 Dosyayı Kapatma

Bir dosya üzerinde işlemlerinizi tamamladıktan sonra, dosyayı kapatmanız önemlidir. Bunu, dosya nesnesinin `close()` metodunu kullanarak yapabilirsiniz.

Örnek:

```
dosya = open("dosya.txt", "r")

# Dosya üzerinde işlemler yapın...

dosya.close()
```

with İfadesi ile Otomatik Kapatma

`with` ifadesi, bir dosyayı otomatik olarak kapatmamızı sağlar. `with` bloğu sona erdiğinde, dosya otomatik olarak kapatılır.

Örnek:

```
with open("dosya.txt", "r") as dosya:
    # Dosya üzerinde işlemler yapın...
```

Bu örnekte, `with` bloğu sona erdiğinde `dosya.txt` dosyası otomatik olarak kapatılır.

Dosyayı Kapatmanın Önemi:

Dosyaları açık bıraktığınızda, işletim sistemi kaynakları gereksiz yere kullanılır ve potansiyel sorunlara yol açabilir. Örneğin, başka bir program dosyaya erişmeye çalıştığında sorunlarla karşılaşabilir veya dosyada yapılan değişiklikler kaydedilmeyebilir. Bu nedenle, dosya işlemlerini tamamladıktan sonra `close()` metodunu kullanarak dosyayı kapatmak veya `with` ifadesini kullanarak otomatik kapatma sağlamak **iyi bir programlama pratiğidir**.

7.3 Dosya Okuma İşlemleri

7.3.1 `read()` Metodu

`read()` metodu, dosyanın tamamını okur ve bir string olarak döndürür.

Örnek:

```
with open("dosya.txt", "r") as dosya:
    icerik = dosya.read()
    print(icerik)
```

7.3.2 readline() Metodu

readline() metodu, dosyanın bir satırını okur ve bir string olarak döndürür.

Örnek:

```
with open("dosya.txt", "r") as dosya:
    satir1 = dosya.readline()
    satir2 = dosya.readline()
    print(satir1, end="") # Satır sonu karakterini yazdırmamak için end="" kullanıyoruz
    print(satir2, end="")
```

7.3.3 readlines() Metodu

readlines() metodu, dosyanın tüm satırlarını okur ve her satırı bir eleman olarak içeren bir liste döndürür.

Örnek:

```
with open("dosya.txt", "r") as dosya:
    satirlar = dosya.readlines()
    for satir in satirlar:
        print(satir, end="")
```

7.3.4 Döngü ile Dosya Okuma

Bir dosyayı satır satır okumak için, bir for döngüsü kullanabiliriz.

Örnek:

```
with open("dosya.txt", "r") as dosya:
    for satir in dosya:
        print(satir, end="") # end="" ile satır sonu karakterini yazdırmayı engelliyoruz
```

7.4 Dosya Yazma İşlemleri

7.4.1 write() Metodu

write() metodu, dosyaya bir string yazar.

Örnek:

```
with open("dosya.txt", "w") as dosya:
    dosya.write("Merhaba Dünya!\n") # \n ile yeni satır karakterini ekliyoruz
    dosya.write("Python ile dosya işlemleri.")
```

7.4.2 writelines() Metodu

writelines() metodu, bir string listesi dosyaya yazar.

Örnek:

```
satirlar = ["Satır 1\n", "Satır 2\n", "Satır 3\n"]

with open("dosya.txt", "w") as dosya:
    dosya.writelines(satirlar)
```

7.4.3 Dosya Konumlandırma (File Positioning)

tell() ve seek() metotlarını kullanarak dosya içinde istediğimiz konuma gidebiliriz. Bu, dosyanın belirli bir bölümünü okumak veya yazmak istediğimizde faydalı olabilir.

- **tell()**: Dosya içindeki geçerli konumu (byte cinsinden) döndürür. Dosyanın başlangıcı 0. bayttır.
- **seek(offset, from_what)**: Dosya imlecini belirtilen konuma taşır.
 - **offset**: Taşınacak bayt sayısı.
 - **from_what**: Konumun nereden itibaren hesaplanacağını belirtir.
 - * 0: Dosyanın başlangıcından (varsayılan)
 - * 1: Geçerli konumdan
 - * 2: Dosyanın sonundan

Örnekler:

```
with open("dosya.txt", "r") as dosya:
    # Geçerli konumu yazdır
    print(dosya.tell()) # Çıktı: 0

    # 5 karakter oku
    dosya.read(5)
    print(dosya.tell()) # Çıktı: 5

    # Dosya başlangıcına git
```

```
dosya.seek(0)
print(dosya.tell()) # Çıktı: 0

# Dosyanın sonundan 10 bayt önceki konuma git
dosya.seek(-10, 2)
print(dosya.tell()) # Dosya boyutuna bağlı olarak değişir
```

7.5 Farklı Dosya Formatları

Python, farklı dosya formatları ile çalışmak için farklı modüller sunar. Örneğin, CSV dosyaları ile çalışmak için `csv` modülü, JSON dosyaları ile çalışmak için `json` modülü kullanılır.

7.6 Ek Örnekler ve Uygulamalar

Aşağıdaki örnekleri ve uygulamaları deneyerek dosya işlemleri konusundaki bilgilerinizi pekiştirebilirsiniz:

1. **Bir dosyadaki satır sayısını bulma:** Bir dosyayı açın ve satır satır okuyarak toplam satır sayısını hesaplayın.
2. **Bir dosyadaki en uzun satırı bulma:** Bir dosyayı açın ve satır satır okuyarak en uzun satırı bulun ve uzunluğunu yazdırın.
3. **Bir dosyadaki belirli bir kelimeyi arama:** Kullanıcıdan bir kelime alın ve bir dosyada bu kelimeyi arayın. Kelime bulunursa, bulunduğu satır numarasını yazdırın.
4. **İki dosyayı birleştirme:** İki dosyayı açın ve içeriklerini birleştirerek yeni bir dosyaya yazın.
5. **Bir dosyanın içeriğini tersine çevirme:** Bir dosyayı açın, içeriğini okuyun ve tersten yeni bir dosyaya yazın.

7.7 Bölüm Sonu Quiz

1. Aşağıdaki kod bloğunda boşluğa hangi ifade gelmelidir?

```
with open("dosya.txt", "_____") as dosya:
    icerik = dosya.read()
    print(icerik)
```

2. `dosya.txt` adlı bir dosyanın içeriğini okuyan ve her satırı büyük harfe dönüştürerek ekrana yazdıran bir Python kodu yazın.

3. Aşağıdaki kod bloğunun çıktısı nedir?

```
with open("dosya.txt", "w") as dosya:
    dosya.write("Merhaba\n")
    dosya.write("Dünya!")
```

4. Kullanıcıdan alınan bir metni `metin.txt` adlı bir dosyaya yazan bir program yazın.

5. `sayilar.txt` adlı bir dosyada her satırda bir sayı olduğunu varsayalım. Bu dosyadaki sayıları okuyan ve toplamını hesaplayan bir program yazın.

6. Aşağıdaki kod bloğunun çıktısı ne olur?

```
with open("dosya.txt", "r") as dosya:
    dosya.read(10)
    konum = dosya.tell()
print(konum)
```

7. `metin.txt` adlı bir dosyanın içeriğini okuyarak her satırın başına satır numarasını ekleyen ve yeni bir dosya (`numaralanmis_metin.txt`) oluşturan bir program yazın.