

## Python'da Dosya Yönetimi (File Handling)

### Tanım:

Python'da dosya yönetimi, dış kaynaklardaki verilerle (örneğin .txt, .csv, .json gibi dosyalarla) etkileşim kurmayı sağlayan işlemlerdir. Bu süreçte dosyalar açılır, okunur, içine veri yazılır ve işlem sonunda kapatılır. Bu işlemler sayesinde veriler kalıcı hale getirilir ve programlar arasında paylaşılabilir.

### Ana Amaçlar:

- Kalıcı veri saklamak
- Programlar arası veri alışverişi sağlamak
- Loglama ve raporlama yapmak

### Dosya Türleri:

1. **Metin Dosyaları (Text files):** .txt, .csv, .json
2. **İkili Dosyalar (Binary files):** Görüntüler, videolar, ses dosyaları vb.

### Dosya Erişim Modları:

- **Read ("r"):** Dosya sadece okunur.
- **Write ("w"):** Dosya yazılır, eski veri silinir.
- **Append ("a"):** Mevcut verinin sonuna ekleme yapılır.
- **Read and Write ("r+"):** Hem okuma hem yazma yapılabilir.

### Güvenli Dosya Kullanımı:

Dosyaların doğru şekilde açılması ve işlem sonunda kapatılması gerekir. Aksi takdirde bellek sızıntısı, veri kaybı veya sistem hataları yaşanabilir. Bu yüzden dosya işlemleri dikkatle yapılmalıdır.

## Python'da Hata Yönetimi (Exception Handling)

### Tanım:

Hata yönetimi, bir programın çalışma sırasında karşılaşılabileceği beklenmedik durumlara (örneğin kullanıcı hataları, dosya bulunamaması, sıfıra bölme, internet bağlantı hataları vb.) karşı önlem almayı sağlayan yapıdır. Python'da bu amaçla `try`, `except`, `else` ve `finally` blokları kullanılır.

### Hatalar Neden Oluşur?

- Kullanıcının yanlış veri girmesi
- Var olmayan bir dosyanın açılmaya çalışılması
- Sıfıra bölme gibi matematiksel hatalar

- İnternete erişim problemi gibi çevresel hatalar

### **Hata Türleri:**

- **SyntaxError:** Yazım hatası
- **NameError:** Tanımlanmamış bir değişkenin kullanılması
- **TypeError:** Yanlış veri tipiyle işlem yapılması
- **FileNotFoundError:** Dosya bulunamadığında oluşan hata
- **ZeroDivisionError:** Sıfıra bölme hatası

### **Hata Yönetiminin Önemi:**

- Programın çökmesini engeller
- Kullanıcı deneyimini iyileştirir
- Hataların kontrol altına alınarak raporlanmasını sağlar
- Kritik işlemlerde veri kaybını önler

### **Hata Yönetiminde Yapılar:**

- **try:** Hatanın oluşabileceği kodlar buraya yazılır.
- **except:** Hata olursa yapılacak işlemler burada yer alır.
- **else:** Hata oluşmazsa çalışacak kod bloğudur.
- **finally:** Hata olsun ya da olmasın her durumda çalışacak kısımdır (genellikle dosya kapatma, bağlantı sonlandırma gibi işlemler için kullanılır).