


# EĞİTİM PLANI

Hafta	Konu Başlıkları
1. Hafta: Python'a Giriş	<ul style="list-style-type: none"><li>- Python Nedir? Kullanım Alanları</li><li>- Python Kurulumu ve IDE'ler</li><li>- Python Çalışma Mantığı: İfadeler, Yorumlayıcı</li><li>- <code>print()</code> ve <code>input()</code> Fonksiyonları</li><li>- Veri Türleri (<code>int</code>, <code>float</code>, <code>string</code>, <code>bool</code>)</li><li>- Yorum Satırları</li><li>- Basit Matematiksel İşlemler</li><li>- Casting (Tür Dönüşümü)</li><li>- Koşullu İfadeler ( <code>if</code> , <code>else</code> , <code>elif</code> )</li></ul>
2. Hafta: Döngüler ve Veri Yapıları	<ul style="list-style-type: none"><li>- Döngüler: <code>for</code> , <code>while</code></li><li>- Döngülerde <code>break</code> , <code>continue</code> , <code>range()</code></li><li>- Listeler ( <code>list</code> )</li><li>- Demetler ( <code>tuple</code> )</li><li>- Sözlükler ( <code>dictionary</code> )</li></ul>
3. Hafta: Fonksiyonlar ve Değişkenler	<ul style="list-style-type: none"><li>- Fonksiyonlar ve Fonksiyon Tanımlama</li><li>- <code>return</code> İfadesi</li><li>- Global ve Yerel Değişkenler</li><li>- Fonksiyonlarda Parametreler</li></ul>
4. Hafta: Dosya İşlemleri	<ul style="list-style-type: none"><li>- Dosya Açma, Okuma, Yazma</li><li>- Dosya Kapatma</li><li>- Dosya Modları ( <code>r</code> , <code>w</code> , <code>a</code> )</li><li>- Hata Yönetimi ve Dosya İşlemleri</li></ul>

# DOSYA MODLARI

- Dosyayı sadece okuma modunda açar.
- Dosya yoksa hata verir.
- Dosya üzerinde değişiklik yapılmaz.



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

# YAZMA MODU (W)

- Dosyayı sadece yazma modunda açar.
- Dosya mevcutsa içeriği siler, dosya yoksa yeni bir dosya oluşturur.
- Eski veriler silinir, sadece yeni yazılan veriler kalır.



```
with open("dosya.txt", "w") as dosya:  
    dosya.write("Yeni bir veri")
```

# EKLEME MODU (A)

- Dosyayı ekleme modunda açar.
- Dosya yoksa yeni bir dosya oluşturur, mevcutsa içeriğin sonuna veri ekler.
- Dosyanın mevcut içeriği korunur ve yeni veri sonuna eklenir.



```
with open("dosya.txt", "a") as dosya:  
    dosya.write("\nEklenen yeni satır")
```

# OKUMA VE YAZMA MODU (R+)

- Dosyayı hem okuma hem de yazma modunda açar.
- Dosya yoksa hata verir.
- Dosyanın içeriği değiştirilebilir ve okunabilir.

```
with open("dosya.txt", "r+") as dosya:  
    veri = dosya.read()  
    print(veri)  
    dosya.write("\nYeni bir satır")
```

# YAZMA VE OKUMA MODU (W+)

- Dosyayı hem yazma hem de okuma modunda açar.
- Dosya mevcutsa içeriği silinir, dosya yoksa yeni bir dosya oluşturur.

```
with open("dosya.txt", "w+") as dosya:  
    dosya.write("Yeni bir içerik")  
    dosya.seek(0) # Dosya başına geri döner  
    print(dosya.read())
```

# EKLEME VE OKUMA MODU (A+)

- Dosyayı hem ekleme hem de okuma modunda açar.
- Dosya yoksa yeni bir dosya oluşturur.
- Dosya sonuna veri eklenebilir ve içeriği okunabilir.

```
with open("dosya.txt", "a+") as dosya:  
    dosya.write("\nYeni bir satır")  
    dosya.seek(0) # Dosya başına geri döner  
    print(dosya.read())
```

# READLINE()

- Dosyadan bir satır okur.
- Çok satırlı dosyalarda her çağrıldığında bir sonraki satırı okur.



```
with open("dosya.txt", "r") as dosya:  
    satir = dosya.readline()  
    print(satir)
```



# READLINES()

- Dosyadaki tüm satırları okur ve bir liste olarak döner.
- Her bir satır liste elemanı olarak saklanır.

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

# TELLO

- **Mevcut dosya konumunu (byte olarak) döner.**



```
with open("dosya.txt", "r") as dosya:  
    dosya.read(10)  
    print(dosya.tell()) # Şu anki konumu verir
```

# SEEK()

- Dosya içindeki okuma/yazma pozisyonunu değiştirir.
- **offset:** Pozisyon kaydırma miktarıdır.
- **from\_what:** Başlangıç noktasıdır (0: dosya başı, 1: mevcut konum, 2: dosya sonu).

```
with open("dosya.txt", "r") as dosya:  
    dosya.seek(5) # 5. byte'a gider  
    print(dosya.read())
```

# CLOSE()

- Dosyayı manuel olarak kapatır.
- Dosya with bloğunun dışında açıldığında kullanılması önemlidir.

```
dosya = open("dosya.txt", "r")  
print(dosya.read())  
dosya.close() # Dosyayı manuel olarak kapatır
```

# HATA YÖNETİMİ: TRY-EXCEPT BLOKLARI

## try-except Yapısının Temelleri

- try bloğunda hata alabilecek kodları yazarız.
- except bloğunda ise hata oluştuğunda yapılacak işlemler yer alır.
- Dosya işlemlerinde sıkça karşılaşılan hatalar (örneğin, dosyanın bulunamaması) için kullanılır.

```
try:
    with open("olmayan_dosya.txt", "r") as dosya:
        print(dosya.read())
except FileNotFoundError:
    print("Hata: Dosya bulunamadı.")
```

# BIRDEN FAZLA EXCEPT BLOĞU İLE HATA TÜRLERİNİ AYIRMA

- Farklı hata türleri için ayrı except blokları kullanabiliriz.
- Dosya işlemlerinde FileNotFoundError ve PermissionError gibi farklı hataları ayırabiliriz.

```
try:
    with open("dosya.txt", "r") as dosya:
        print(dosya.read())
except FileNotFoundError:
    print("Hata: Dosya bulunamadı.")
except PermissionError:
    print("Hata: Dosyaya erişim izniniz yok.")
```

# ELSE VE FINALLY BLOKLARI

- **else:** try bloğunda hata oluşmazsa çalışır.
- **finally:** Hata oluşsa da oluşmasa da her durumda çalışır. Genellikle dosyayı kapatmak gibi işlemler için kullanılır.

```
try:
    with open("dosya.txt", "r") as dosya:
        print(dosya.read())
except FileNotFoundError:
    print("Hata: Dosya bulunamadı.")
else:
    print("Dosya başarıyla okundu.")
finally:
    print("İşlem tamamlandı.")
```