



PROGRAMLAMA TEMELLERİ

Öğr. Gör. Erhan AKAGÜNDÜZ

DOSYA İŞLEMLERİ

Çalışma Dizini Ayarları ve Klasör Oluşturma

- ❑ 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.

DOSYA İŞLEMLERİ

Çalışma Dizini Ayarları ve Klasör Oluşturma

```
>>> import os
>>> os.getcwd()
'C:\\Users\\Ahmet\\AppData\\Local\\Programs\\Python\\Python38'
>>>
```

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

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

DOSYA İŞLEMLERİ

Yol (Path) Tanımlama

- ❑ 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.
- ❑ Python, yol tanımlama işlemleri için kullanışlı bir çok fonksiyonu barındıran **os.path** alt modülünü içerir.

DOSYA İŞLEMLERİ

Yol (Path) Tanımlama

- ❑ **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) # test\python (Windows Sistemde)

pc = os.path.split(yol)
print(pc) # ('test', 'python')

test\python
('test', 'python')
```

DOSYA İŞLEMLERİ

Yolu Bilinen Klasör veya Dosyanın Var Olup Olmadığını Kontrol Etme

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

C:\test : var

C:\test : bir dizin

DOSYA İŞLEMLERİ

Klasör Oluşturma

- ❑ **mkdir()** veya **makedirs()** fonksiyonları kullanılarak belirlediğiniz konumda yeni bir klasör oluşturulabilir.
- ❑ Yeni bir klasör oluşturulurken öncelikle belirtilen konumda aynı klasörün var olup olmadığı kontrol edilmelidir.

```
import os
yol = os.path.join("C:\\", "test2")
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...")
```

C:\test2 : var

C:\test2 : bir dizin

```
import os

# Mevcut klasör yolu
ana_dizin = "C:\\\\"
eski_klasor_adi = "test"
yeni_klasor_adi = "yeni_test"

# Eski ve yeni klasör yollarını oluşturma
eski_klasor_yolu = os.path.join(ana_dizin, eski_klasor_adi)
yeni_klasor_yolu = os.path.join(ana_dizin, yeni_klasor_adi)

# Klasör adını değiştirme
try:
    os.rename(eski_klasor_yolu, yeni_klasor_yolu)
    print(f"Klasör başarıyla {eski_klasor_adi} -> {yeni_klasor_adi} olarak değiştirildi.")
except FileNotFoundError:
    print(f"Eski klasör yolu bulunamadı: {eski_klasor_yolu}")
except PermissionError:
    print("Bu işlemi gerçekleştirmek için gerekli izinler yok.")
except Exception as e:
    print(f"Bir hata oluştu: {e}")
```

Klasör başarıyla test -> yeni_test olarak değiştirildi.

DOSYA İŞLEMLERİ

Dosyalara Erişme ve Okuma

- ❑ Python dili dosya işlemleri için oldukça kullanışlı gömülü fonksiyonlar içerir.
- ❑ Bu fonksiyonları herhangi bir modülü içe aktarmadan kullanabilirsiniz.
- ❑ Önce var olan bir dosyayı açıp içeriğinin nasıl okunabileceği öğrenilmelidir.
- ❑ Çalışma dizini "**C:\test**" klasörü olarak ayarlanmıştı.
- ❑ Örneklerle başlamadan önce bu dizinde **milli.txt** isminde bir metin dosyası oluşturunuz ve içeriğini yazıp kaydediniz.

DOSYA İŞLEMLERİ

Dosyalara Erişme ve Okuma

- ❑ Mevcut bir dosyayı açmak için **open()** fonksiyonu kullanılır. **open()** fonksiyonu, belirtilen yolda bulunan dosyayı açar ve dosya nesnesi olarak programa döndürür.
- ❑ `open()` fonksiyonu, dosya ismi ve mod (dosya açma biçimi) olarak iki parametre alır. Dosyayı açmak için 4 farklı mod kullanılır.

DOSYA İŞLEMLERİ

Dosyalara Erişme ve Okuma

- ❑ **"r" - Okuma** – Okumak için bir dosya açar. Dosya hedefte yoksa hata verir(Varsayılan).
- ❑ **"a" - Ekleme** – Var olan dosyada düzenleme yapmak için açar. Dosya hedefte yoksa oluşturulur.
- ❑ **"w" - Yazma** – Yazma modunda bir dosya açar. Dosya hedefte yoksa oluşturulur.
- ❑ **"x" - Oluştur** – Belirtilen dosyayı oluşturur. Dosya varsa hata döndürür.

DOSYA İŞLEMLERİ

Dosyalara Erişme ve Okuma

- ❑ **read()** fonksiyonu, açılmış dosyanın içeriğini tek bir metin bilgisi olarak okur.

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

```
dosya = open("milli.txt", encoding="utf-8")  
print(dosya.read())
```

```
Korkma, sönmez bu şafaklarda yüzen al sancak;  
Sönmeden yurdumun üstünde tüten en son ocak.  
O benim milletimin yıldızıdır, parlayacak;  
O benimdir, o benim milletimindir ancak.
```

DOSYA İŞLEMLERİ

Dosyalara Erişme ve Okuma

❑ **readlines()** fonksiyonu, içeriği satırlara ayırır ve satırlardan oluşmuş bir liste döndürür.

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

dosya = open("milli.txt", encoding="utf-8")
print(dosya.readlines())
```

```
['Korkma, sönmez bu şafaklarda yüzen al sancak;\n', 'Sönmeden yurdumun üstünde tüten en son ocak.\n', 'O benim milletimin yıldızıdır, parlayacak;\n', 'O benimdir, o benim milletimindir ancak.']
```

DOSYA İŞLEMLERİ

Dosyalara Erişme ve Okuma

- ❑ **readline()** fonksiyonu, mevcut satırı okur ve her çağrıldığında bir sonraki satırı getirir.

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

dosya = open("milli.txt", encoding="utf-8")
print(dosya.readline())
print(dosya.readline())
print(dosya.readline())

Korkma, sönmez bu şafaklarda yüzen al sancak;
Sönmeden yurdumun üstünde tüten en son ocak.
O benim milletimin yıldızıdır, parlayacak;
```

DOSYA İŞLEMLERİ

Dosyalara Erişme ve Okuma

- ❑ **readlines()** veya **readline()** fonksiyonlarını döngü ile beraber kullanarak dosyalar satır satır işlenebilir.
- ❑ Açılan her dosya belleğe atılır ve hafızada yer kaplar. Bu nedenle verimli bir bellek yönetimi için dosyalar çalışma sonunda **close()** fonksiyonu kullanılarak kapatılmalı ve bellekten kaldırılmalıdır.

```
import os
os.chdir('C:\\test')
dosya = open("milli.txt", encoding="utf-8")
for satir in dosya:
    print(satir.upper(), end="")
dosya.close()
```

```
KORKMA, SÖNMEZ BU ŞAFAKLARDA YÜZEN AL SANCAK;
SÖNMEYEN YURDUMUN ÜSTÜNDE TÜTEN EN SON OCAK.
O BENİM MİLLETİMİN YILDIZIDIR, PARLAYACAK;
O BENİMDİR, O BENİM MİLLETİMİNDİR ANCAK.
```

DOSYA İŞLEMLERİ

Dosya Oluşturma ve Yazma

- ❑ Dosya açmak için kullanılan **open()** fonksiyonu **“w” (write)** parametresi ile beraber kullanılırsa dosya yazma modunda açılır ve veri yazımına hazır hâle gelir.

```
import os
os.chdir('C:\\test')
dosya = open("hitabe.txt", "w")
dosya.write("Duydum ki bizi bırakmaya azmediyorsun, etme\n")
dosya.write("Başka bir yar, başka bir dosta meylediyorsun, etme.\n")
dosya.write("Sen yadeller dünyasında ne arıyorsun yabancı, \n")
dosya.write("Hangi hasta gönüllüyü kastediyorsun, etme.")
dosya.close()
```


DOSYA İŞLEMLERİ

Dosya Oluşturma ve Yazma

- ❑ Dosya içeriğine mevcut veriyi silmeden ekleme yapmak için dosyanın “a” parametresi kullanılarak **append (ekle)** modunda açılması gerekmektedir.
- ❑ Bu modda açılan dosyaya eklediğiniz her yeni satır son verinin altına yazılacaktır.
- ❑ **isfile()** fonksiyonu, erişilen konumun dosya olup olmadığını kontrol eder.

DOSYA İŞLEMLERİ

Dosya Silme ve Yedekleme

- ❑ Mevcut bir dosyayı silmek için os modülünün **remove()** fonksiyonu kullanılır.
- ❑ Aşağıdaki örnekte, belirtilen dosyanın var olup olmadığı kontrol edilerek dosya varsa silinmektedir.

```
import os
os.chdir('C:\\test')
if os.path.exists("tarih.txt"):
    print("Dosya mevcut, siliniyor...")
    os.remove("tarih.txt")
else:
    print("Dosya mevcut değil.")
```

Dosya mevcut, siliniyor...

DOSYA İŞLEMLERİ

Dosya Silme ve Yedekleme

- ❑ Çalışılan dosyaların herhangi bir olumsuz duruma karşı yedeklenmesi gerekir.
- ❑ Bu sayede veri ve emek kaybının önüne geçilebilir.
- ❑ Python'da **os** modülü ile birlikte dosya kopyalama, taşıma ve üst düzey dosya işlemlerinde kullanabileceğiniz **shutil** adlı modül bulunmaktadır.
- ❑ Aşağıdaki örnekte **os** ve **shutil** modülleri kullanılarak daha önce oluşturulan **test** klasöründeki **tarih.txt** dosyası, yeni oluşturulacak **yedek** isimli klasöre **tarihyedek.txt** ismi ile kopyalanarak yedeklenecektir.

```
import os, shutil
os.chdir('C:\\test')
yedek_dizini = os.path.join("C:\\", "yedek")
dosya = "tarih.txt"
if os.path.exists(yedek_dizini):
    print(yedek_dizini + ' : var')
    if os.path.isdir(yedek_dizini):
        print(yedek_dizini + ' : bir dizin')
else:
    print("Yedek dizini bulunamadı!")
    os.mkdir(yedek_dizini)
    print("oluşturuluyor...")

yedek_dosya = os.path.join(yedek_dizini, "tarihyedek.txt")

if os.path.exists(dosya):
    print("Dosya mevcut, yedekleniyor.")
    shutil.copy(dosya, yedek_dosya)
if os.path.exists(yedek_dosya):
    print("Dosya başarı ile yedeklendi.")
```

```
Yedek dizini bulunamadı!
oluşturuluyor...
```

DOSYA İŞLEMLERİ

```
import os
import shutil

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

yedek_dizini = os.path.join("C:\\", "yedek")
dosya = "tarih.txt"
yedek_dosya = os.path.join(yedek_dizini, "tarihyedek.txt")

# Yedek dizinini kontrol et ve oluştur
if os.path.exists(yedek_dizini):
    print(yedek_dizini + ' : var')
    if os.path.isdir(yedek_dizini):
        print(yedek_dizini + ' : bir dizin')
else:
    print("Yedek dizini bulunamadı!")
    os.makedirs(yedek_dizini) # Tüm ara dizinleri oluşturur
    print("oluşturuluyor...")
```

DOSYA İŞLEMLERİ

```
# Dosyanın varlığını kontrol et ve yedekle
if os.path.exists(dosya):
    print("Dosya mevcut, yedekleniyor.")
    shutil.copy(dosya, yedek_dosya)
    if os.path.exists(yedek_dosya):
        print("Dosya başarı ile yedeklendi.")
else:
    print("Yedeklenecek dosya bulunamadı.")
```

```
C:\yedek : var
C:\yedek : bir dizin
Dosya mevcut, yedekleniyor.
Dosya başarı ile yedeklendi.
```