

# ÖĞRENME BİRİMİ 9

## DOSYA İŞLEMLERİ

### Neler Öğreneceksiniz?

#### Bu öğrenme birimi ile;

- Python dilini kullanarak dosya ve dizin oluşturabilecek,
- Dosyalara erişip içeriğini okuyabilecek ve değiştirebilecek,
- Dosyaları silme ve yedekleme işlemlerini yapabileceksiniz.

### Anahtar Kelimeler:

Dosya, dizin, veri yazma, kopyalama



**Hazırlık Çalışmaları**

1. İnternet üzerinden dosya, klasör, dosya sistemi konularını araştırınız.
2. Günümüzde bilgisayarlarda kullanılan dosya sistemlerini araştırınız.

## 9. DOSYA İŞLEMLERİ

### 9.1. 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 çağırılması gerekir. Komut satırından bu modüle ait **getcwd()** fonksiyonu çağırılarak programın mevcut çalışma dizini öğrenilebilir.

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

Aynı komutlar aşağıdaki gibi bir Python dosyasına yazılarak da çalıştırılabilir. Klasör, dosya oluşturma ve silme gibi işlemlerin çoğunlukla program içinden yapılması gerekir. Bu nedenle bu yöntemi kullanmanız önerilir.

**Örnek 1:**

```
import os
dizin = os.getcwd()
print(dizin)
```

**Çıktı:**

```
C:\\Users\\Ahmet\\AppData\\Local\\Programs\\Python\\Python38
```

Mevcut çalışma dizinini değiştirmek için **chdir()** fonksiyonu kullanılır. Yukarıdaki kod bloğu, çalışma dizinini c sürücüsünde test klasörü ("c:\\test") olarak değiştirecek şekilde geliştirilebilir.

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

**Çıktı:**

```
Traceback (most recent call last):
  File "C:/Users/Ahmet/AppData/Local/Programs/Python/Python38/2.py", line 4, in
<module>
    os.chdir('c:\\test')
FileNotFoundError: [WinError 2] The system cannot find the file specified: 'c:\\test'
```

Kod bloğu çalıştırıldığında yukarıdaki hata ile karşılaşılacaktır. C: sürücüsünde “test” isimli bir klasör olmadığı için Python belirtilen dizini bulamaz. Bu nedenle çalışma dizini değiştirme işlemi başarısız olur. Bilgisayarın c: sürücüsünde ilgili klasör oluşturularak bu sorun çözülebilir ancak klasör ve dosya oluşturma işlemlerini kod bloğu içinde yapmak daha doğru bir yöntemdir. Python dili ile dosya veya dizin oluşturmak için gerekli bilgiler ve fonksiyonlar, aşağıdaki konu başlıklarında ve örneklerde adım adım incelenecektir.

**9.1.1. 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. Yazılan kodların platformdan bağımsız olarak Windows, Linux, Unix ve Mac OS gibi bütün işletim sistemlerinde çalışabilmesi için dosya ve dizin işlemlerini platformdan bağımsız yollar tanımlayarak yapmak gerekecektir. Python, yol tanımlama işlemleri için kullanışlı bir çok fonksiyonu barındıran `os.path` alt modülünü içerir.

**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.

**Örnek 2:**

```
import os
yol = os.path.join('test','python')
print(yol) # test\python (Windows Sistemde)
pc = os.path.split(yol)
print(pc) # ('test', 'python')
```

**Çıktı:**

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

**Önemli Not:** Yukarıdaki kod bloğunda “print(yol)” komut satırı Windows işletim sisteminde “test\python” şeklinde çıktı üretirken Unix işletim sisteminde “test/python” şeklinde çıktı üretecektir. Ayraçların birbirinden farklı olduğuna dikkat ediniz.

**9.1.2. 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.

**Örnek 3:**

```
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.")
```

**Çıktı:**

Yol bulunamadı.

Kod bloğu çalıştığında program belirtilen yolun varlığını araştırmış ve bulamadığı için hata vermek yerine ekrana **"Yol bulunamadı"** çıktısı vermiştir.

**9.1.3. 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. Yukarıdaki kodları eğer klasör yoksa oluşturacak şekilde düzenleyebilirsiniz.

```
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...")
```

**Çıktı:**

Yol bulunamadı!  
oluşturuluyor...

Kodunuzu bir kez daha çalıştırdığınızda klasörün artık oluşturulduğunu ve ekran çıktısının değiştiğini görsünüz.

**Çıktı:**

```
C:\test : var
C:\test : bir dizin
```

Yukarıdaki örnek tekrar çalıştırıldığında çalışma dizini değiştirilmiş olacaktır.

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

**Çıktı:**

```
C:\Users\Ahmet\AppData\Local\Programs\Python\Python38
c:\test
```

Programın çalışma dizini "c:\test" olarak değiştirilmiş oldu. Sonraki birimlerde dosyaları oluşturmak ve saklamak için bu dizin kullanılacaktır.

**Sıra Sizde:** **os** modülü ile gelen **rename()** fonksiyonu dosya / dizin isim değiştirme işlemleri için, **rmdir()** fonksiyonu ise dizin silme işlemleri için kullanılır. Siz de bu fonksiyonları kullanarak deneme isimli bir dizin oluşturup daha sonra bu dizinin ismini değiştirip silen programı öğretmenlerinizle birlikte yazınız.

#### 9.1.4. 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.

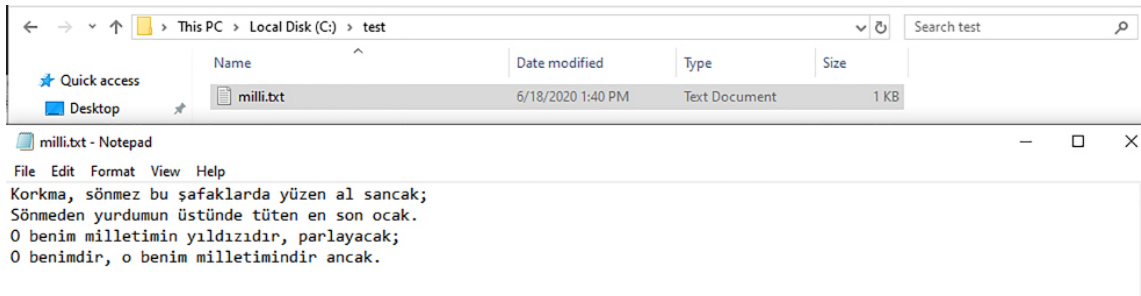
Çalışma dizini "C:\test" klasörü olarak ayarlanmıştı. Örneklere başlamadan önce bu dizinde milli.txt isiminde bir metin dosyası oluşturunuz ve içeriğini yazıp kaydediniz.

"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."



**Görsel 9.1:** Test klasöründe milli.txt dosyası

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.

**"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.

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

#### Örnek 4:

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

dosya = open("milli.txt")
print(dosya.read())
```

#### Çıktı:

```
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.
```

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

#### Örnek 5:

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

dosya = open("milli.txt")
print(dosya.readlines())
```

#### Çıktı:

```
['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.']
```

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

**Örnek 6:**

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

dosya = open("milli.txt")
print(dosya.readline())
print(dosya.readline())
```

**Çıktı:**

Korkma, sönmez bu şafaklarda yüzen al sancak;  
  
Sönmeden yurdumun üstünde tüten en son ocak.

**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.

**Örnek 7:**

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

dosya = open("milli.txt")
for satir in dosya:
    print(satir.upper(),end="")
dosya.close()
```

**Çıktı:**

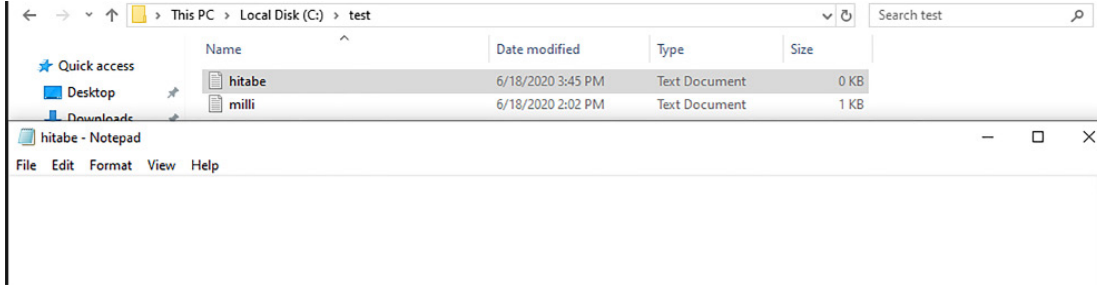
KORKMA, SÖNMEZ BU ŞAFAKLARDA YÜZEN AL SANCAK;  
SÖNMEYEN YURDUMUN ÜSTÜNDE TÜTEN EN SON OCAK.  
O BENİM MİLLETİMİN YILDIZIDIR, PARLAYACAK;  
O BENİMDİR, O BENİM MİLLETİMİNDİR ANCAK.

**9.2. 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.

```
dosya = open("hitabe.txt", "w")
```

satırı dosyanın mevcut olması durumunda önceki içeriği siler ve hitabe.txt dosyasını yazma modunda açar. Eğer böyle bir dosya bulunmuyorsa dizinde bu isimle boş bir dosya oluşturur.



**Görsel 9.2:** Oluşturulan boş hitabe.txt dosyası

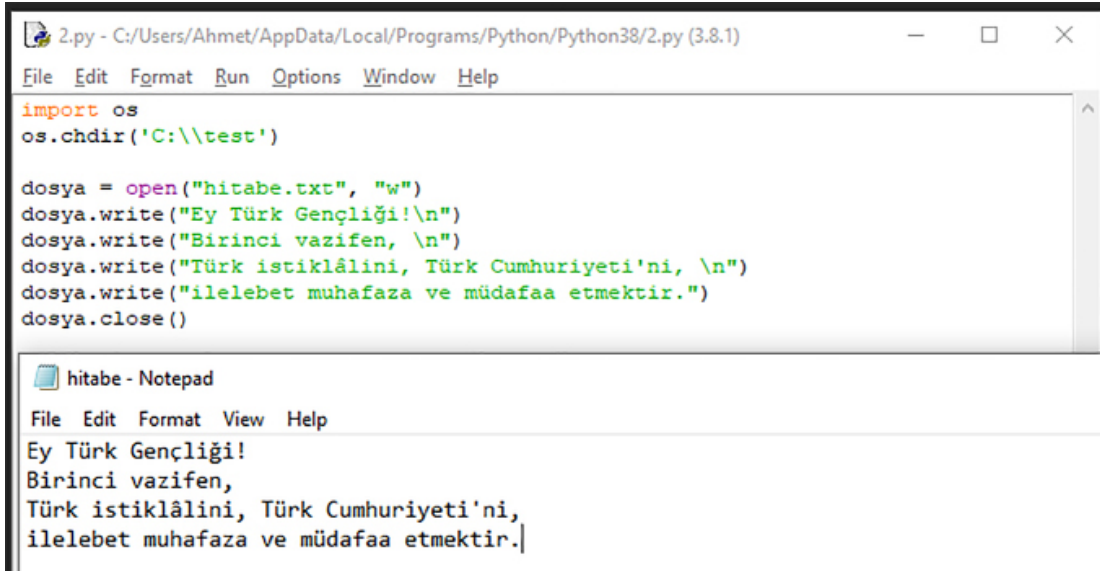
Açılan dosyaya veri yazdırmak için **write()** fonksiyonu kullanılır.

**Örnek 8:**

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

dosya = open("hitabe.txt", "w")
dosya.write("Ey Türk Gençliği!\n")
dosya.write("Birinci vazifen, \n")
dosya.write("Türk istiklâlîni, Türk Cumhuriyeti'ni, \n")
dosya.write("ilelebet muhafaza ve müdafaa etmektir.")
dosya.close()
```

Kod bloğu çalıştırıldığında hitabe.txt dosyasının içeriğine “Gençliğe Hitabe”nin dört satırının yazıldığı görülecektir.



**Görsel 9.3:** hitabe.txt dosyası

Dosya **write** (yazma) modunda açıldığından dosyanın içeriğini okumak için kapatılıp tekrar açılması gerekmektedir. Dosya kapatılmadan dosyanın içeriği okunmak istenirse program hata verecektir.

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.



**Örnek 9:**

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

dosya = open("tarih.txt", "w")
tarih = datetime.date.today()
dosya.write(str(tarih) + "\n")
dosya.close()

dosya = open("tarih.txt")
print(dosya.read())
dosya.close()

dosya = open("tarih.txt", "a")
saat = datetime.datetime.now().time()
dosya.write(str(saat) + "\n")
dosya.close()

dosya = open("tarih.txt")
print(dosya.read())
dosya.close()
```

**Çıktı:**

2020-06-18

2020-06-18

16:18:52.916795

Kod bloğunda yeni bir dosya oluşturulmuş, içeriğine tarih bilgisi içeren bir satır eklenmiştir. Daha sonra düzenlemek için açılan dosyaya saat bilgisi eklenmiş ve dosya okunup ekrana yazdırılmıştır.

Belirtilen konumda dosyanın var olup olmadığının kontrolü `os.path.exists()` fonksiyonu ve `os.path.isfile()` fonksiyonları ile birlikte yapılabilir. `os.path.isfile()` fonksiyonu, erişilen konumun dosya olup olmadığını kontrol eder.

**Örnek 10:**

```
import os
yol = os.path.join("C:\\", "test", "tarih.txt")
if os.path.exists(yol):
    print(yol + ' : konumu var.')
    if os.path.isfile(yol):
        print(yol + ' : bir dosya')
    else:
        print("bir dosya değil")
else:
    print("Yol bulunamadı!")
```

**Çıktı:**

```
C:\test\tarih.txt : konumu var.
C:\test\tarih.txt : bir dosya
```

**9.3. Dosya Silme ve Yedekleme****Görsel 9.4:** Yedekleme

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

**Örnek 11:**

```
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.")
```

**Çıktı:**

Dosya mevcut, siliniyor...

Kod çalıştırıldığında daha önce oluşturulan tarih.txt dosyası silinecektir. Kod bir daha çalıştırılırsa dosya silindiği için **"Dosya mevcut değil."** çıktısı verir.

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.

**Örnek 12:**

```
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.")
```

**Çıktı:**

```
Yedek dizini bulunamadı!
oluşturuluyor...
Dosya mevcut, yedekleniyor.
Dosya başarı ile yedeklendi.
```

## ÖLÇME VE DEĞERLENDİRME 9

1. Verilen konumun var olup olmadığı bilgisini veren ..... veren fonksiyonu, dosya olup olmadığı bilgisini veren ise.. ..... fonksiyonudur.
2. Açılmış dosyalar üzerinde farklı modlarda işlem yapılabilmesi için önce kapatılmaları gerekir. (D/Y)
3. Aşağıdakilerden hangisi dosya sistemlerine erişmek için gerekli modüldür?
  - A) os
  - B) math
  - C) for
  - D) time
  - E) timedelta
4. Verilen yolun klasör olup olmadığı bilgisini veren fonksiyon hangisidir?
  - A) getcwd()
  - B) isdir()
  - C) exists()
  - D) split()
  - E) sin()
5. Aşağıdakilerden hangisi dosya işlemleri ile ilgili bir fonksiyon değildir?
  - A) getcwd()
  - B) join()
  - C) mkdir()
  - D) exists()
  - E) print()

**NOT:** Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları veya faaliyetleri geri dönerek tekrarlayınız.

## ÖĞRENME BİRİMLERİ ÖLÇME VE DEĞERLENDİRME CEVAP ANAHTARLARI

### ÖĞRENME BİRİMİ 1'İN CEVAP ANAHTARI

1. E
2. B
3. B
4. D
5. Adım1: başla, Adım 2: tencereye su koy, Adım 3: tuz ekle, Adım 4: suyu kaynat, Adım 5: makarnayı ekle, Adım 6: makarnayı pişir, Adım 7: makarnanın suyunu süz, Adım 8: bitir

### ÖĞRENME BİRİMİ 2'NİN CEVAP ANAHTARI

1. E
2. C
3. A
4. D
5. A
6. E

### ÖĞRENME BİRİMİ 3'ÜN CEVAP ANAHTARI

1. Hayır. Çünkü; Python programlarını çalıştırabilmek için bir işletim sistemi gereklidir. Dolayısıyla Python ile işletim sistemi yazılamaz.
2. En son versiyonu kullanmak gerekliliği ile ilgili şunlar söylenebilir.
  - Yeni işlev ve özellikler gelmiştir.
  - Hatalardan arındırılmıştır.
  - Daha az hafıza ve işlemci gücü gereksinimi vardır.
3. Şu IDE'ler tercih edilebilir.
  - PyCharm: Dünyanın en çok kullanılan IDE'si olduğu için tercih edilebilir.
  - Spyder: Açık kaynak kod olduğu için tercih edilebilir.
  - Visual Studio Code: Sadece Python'a değil, birçok dile destek verdiği için tercih edilebilir.
4. Komutların işlevleri:
  - VI.  $10 + 20$  : İki sayıyı toplar.
  - VII.  $4 * 30$  : İki sayıyı çarpar.
  - VIII.  $2 ** 1000$  : 2 üzeri 1000'i hesaplar.
  - IX. `print("merhaba")` : Ekranı "merhaba" yazdırır.
  - X.  $36 / 4 * (3 + 2) * 4 + 2$  : İşlem önceliğine göre hesaplama yapar.
5. Evet.

#### **ÖĞRENME BİRİMİ 4' ÜN CEVAP ANAHTARI**

1. B
2. E
3. C
4. A
5. E
6. B
7. C
8. A
9. D
10. D

#### **ÖĞRENME BİRİMİ 5'İN CEVAP ANAHTARI**

1. B
2. True
3. Break
4. Random
5. Range
6. D
7. B
8. C
9. A
10. E

#### **ÖĞRENME BİRİMİ 6'NIN CEVAP ANAHTARI**

- 1 – Modül
- 2 – def kare\_al(sayı) :
- 3 – Özyinelemeli fonksiyon
- 4 – D
- 5 – C
- 6 – E
- 7 – B
- 8 – A

#### **ÖĞRENME BİRİMİ 7'NİN CEVAP ANAHTARI**

- 1 – C
- 2 – D
- 3 – C
- 4 – find()
- 5 – B

## ÖĞRENME BİRİMİ 8'İN CEVAP ANAHTARI

1. İstenildiği kadar kullanılabilir. Her hata için birden fazla except bloğu da kullanılabilir. Sadece try-finally şeklindeki kullanımlarda except bloğu kullanılmayabilir.

2. Evet geçerli bir kullanımdır.

3. Evet yakalanabilir.

```
try:
    pass
except ValueError:
    pass
except (TypeError, ZeroDivisionError):
    pass
except:
    # handle all other exceptions
    pass
```

4. C) 2

5. B) False

6.

```
print("Program başladı") # Ekrana "Program başladı" yazar.
try: # try bloğu başlangıcı
    raise Exception('Bir hata oluştu!') # kodla hata fırlatılır.
except: # fırlatılan hata burada yakalanır.
    print("Except bloğuna geldik.") # ekrana "Except bloğuna geldik" yazar.
    raise # hata tekrar fırlatılır
finally: # bu kod mutlaka çalıştırılır.
    print("Finally bloğuna geldik.") # ekrana "Finally bloğuna geldik" yazar
print("Program bitti") # Bu satır çalışmaz. Çünkü raise ifadesi ile hata tekrar fırlatılır ve program yarıda kesilir.
```

## ÖĞRENME BİRİMİ 9' UN CEVAP ANAHTARI

1. exists() , isfile()

2. Doğru

3. A

4. B

5. E



## KAYNAKÇA

Algan, S. (2008). Her Yönüyle C#. İstanbul: Pusula Yayıncılık.

El Harezmi. (2014, 17 Mayıs). Erişim adresi: <http://matematik.dpu.edu.tr/index/sayfa/3119/el-harezmi>

Hunt, J. (2019). A Beginners Guide to Python 3 Programming. Springer.

Python Documentation. (2020, 14 Nisan). Erişim adresi: <https://docs.python.org/3/>

Su, G. (2018). Scratch ile Programlama. İstanbul: Kodlab.

Tanimoto, S. (2017). Game Design for Problem Solving with Python. Erişim adresi: <https://courses.cs.washington.edu/courses/cse190d/17sp/materials/GDFPSP-ch1-3.pdf>

Topal, A.D., Alkan, A. (2010). Mayer'in Bilimsel ve Matematiksel Mesaj Tasarım İlkelerine

Göre Tasarlanmış Öğrenme Ortamının Öğrenci Başarısı Üzerine Etkisi. Kocaeli Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 20(2), 93-106.

Türk Dil Kurumu Sözlükleri. (2020, 10 Mayıs). Erişim adresi: <https://sozluk.gov.tr/>

Scratch web sitesi. Erişim adresi: <http://scratch.mit.edu/about>, kaynağından Nisan 27, 2020 tarihinde erişilmiştir.

**\*Kaynakça kısmı APA6 referanslama sistemi kullanılarak oluşturulmuştur.**

## GÖRSEL KAYNAKÇALARI

GÖRSEL NO	ERİŞİM ADRESİ		ERİŞİM TARİHİ
ÖĞRENME BİRİMİ 1			
Öğrenme Birimi Kapak Resimleri	123rf.com	İd:45725682	
	123rf.com	İd:88225741	
	123rf.com	İd:111714055	
Görsel 1.1	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 104723360	
Görsel 1.2	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 599345642	
Görsel 1.3	https://courses.cs.washington.edu/courses/cse190d/17sp/materials/GDFPSP-ch1-3.pdf sitesindeki figure 1.4 resmi ile <a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a> sitesindeki <b>id: 320117219</b> Resimlerinin birleştirilmesi ile elde edilmiştir.		30.04.2020
Görsel 1.4	<a href="https://www.cs.princeton.edu/courses/archive/spring18/cos226/assignments/8puzzle/index.html">https://www.cs.princeton.edu/courses/archive/spring18/cos226/assignments/8puzzle/index.html</a>		15.04.2020
Görsel 1.5	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 1135956632	
Görsel 1.6	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 127871471	
ÖĞRENME BİRİMİ 2			
Öğrenme Birimi Kapak Resimleri	123rf.com	İd:145813081	
	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:726501535	
	123rf.com	İd:86327943	
ÖĞRENME BİRİMİ 3			
Öğrenme Birimi Kapak Resimleri	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:604363835	
	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:1646703295	
	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:164090489	
Görsel 3.1	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 772227652	
Görsel 3.2	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 701798683	
Görsel 3.3	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 288042338	
Görsel 3.4	<a href="https://www.tiobe.com/tiobe-index/">https://www.tiobe.com/tiobe-index/</a>		14.04.2020
ÖĞRENME BİRİMİ 4			
Öğrenme Birimi Kapak Resimleri	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:1681185799	
	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:1188132619	
		İd:649380919	
Görsel 4.1	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:516701455	
Görsel 4.2	123rf.com	image ID: 89201019- resmi-Komisyon- Görsel- Tasarım-uzmanı- tarafından-düzenerek- kullanılmıştır.	

	ÖĞRENME BİRİMİ 5		
Öğrenme Birimi Kapak Resimleri	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:1673512366	
	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 708208342	
	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:424439902	
Görsel 5.1	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 1188320323	
	ÖĞRENME BİRİMİ 6		
Öğrenme Birimi Kapak Resimleri	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:1478984780	
	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:39409432	
	123rf	İd:82427415	
Görsel 6.2	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 771551869	
	ÖĞRENME BİRİMİ 7		
Öğrenme Birimi Kapak Resimleri	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 766847608	
	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:1388181686	
		İd:442939405	
Görsel 7.2	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 316441487	
	ÖĞRENME BİRİMİ 8		
Öğrenme Birimi Kapak Resimleri	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:1147086383	
	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:173105192	
	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:1728589483	
	ÖĞRENME BİRİMİ 9		
Öğrenme Birimi Kapak Resimleri	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 279072458	
	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd.:543238333	
	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd:116342143	
Görsel 9.2	<a href="https://www.shutterstock.com/">https://www.shutterstock.com/</a>	İd: 730338076	
Kitap Kapağı Görseli	Kitap kapağı görseli grafik tasarımcı tarafından tasarlanmıştır.		

