

Girdi Çıktı İşlemleri

Prof.Dr. Bahadır AKTUĞ
JFM212 Python ile Mühendislik Uygulamaları

**Kaynakça bölümünde verilen kaynaklardan derlenmiştir.*

Ekrandan Okuma ve Ekranaya Yazma

- ▶ Python'da ekrandan okuma için versiyon 3'den itibaren "input" komutu kullanılır.
- ▶ Versiyon 3 öncesinde raw_input() komutu bulunmaktadır.
- ▶ Komutun genel hali;

```
>>> a = input()
```

şeklindedir.

- ▶ "input" komutu ile okunan tüm veriler "metin" tipindedir. Eğer okunan veri ile daha sonra aritmetik işlem yapılacaksa, ilgili sayı formatına (integer, float vb.) dönüştürülmelidir.
- ▶ Ekranaya yazma için "print" fonksiyonu kullanılır.

Dosyadan Okuma

- ▶ Python'da dosya işlemleri diğer birçok programlama dillerine benzer şekilde yapılmaktadır.
- ▶ Dosyadan okuma yapmak için öncelikle;

```
>>> fid = open("ogrenci_listesi.txt", "r")
```

komutu verilir.

- ▶ 'r' dosyanın açma modudur. Burada okuma amaçlı açılacağını belirtmektedir.
- ▶ Dosya açma komutunda, dosya modu belirtilmezse varsayılan olarak okuma modunda açılır.

Dosyadan Okuma

- ▶ Dosya açıldıktan sonra, Python'da dosyadan okuma için
 - `read(byte-sayısı)`
 - `readline()`
 - `readlines()`

komutları vardır.

- ▶ `a = read(n)` komutu ile dosyadan n byte veri okunur ve a değişkenine atanır.
- ▶ `a = readline()` komutu ile dosyadan bir satır veri okunur ve a değişkenine atanır.
- ▶ `a = readlines()` ile dosyadaki tüm satırlar okunur ve a değişkenine atanır.

Dosyadan Okuma

- ▶ Açılan dosya, okuma işlemi yapıldıktan sonra kapatılmalıdır.

- ▶ Kapatma işlemi için;

```
>>> fid.close()
```

komutu verilir.

- ▶ 'r' dosyanın açma modudur. Burada okuma amaçlı açılacağını belirtmektedir.
- ▶ Dosya açma komutunda, dosya moduna bir şey yazılmazsa varsayılan olarak okuma modunda açılır.

Dosya Açma Modu

Mode	Açıklama
r	Okuma modunda açar ve dosya işaretçisi dosyanın başına gider.
rb	Binary bir dosyayı okuma modunda açar ve dosya işaretçisi dosyanın başına gider
r+	Hem okuma hem de yazma modunda açar. Dosya işaretçisi dosyanın başına gider.
rb+	Binary bir dosyayı hem okuma hem de yazma modunda açar. Dosya işaretçisi dosyanın başına gider.
w	Yazma modunda açar. Dosya mevcutsa üzerine yazar, yoksa yeni dosya yaratır.
wb	Binary bir dosya olarak yazma modunda açar. Dosya mevcutsa üzerine yazar, yoksa yeni dosya yaratır.
w+	Hem okuma hem de yazma modunda açar. Dosya mevcutsa üzerine yazar, yoksa yeni dosya yaratır.
wb+	Binary bir dosya olarak hem okuma hem de yazma modunda açar. Dosya mevcutsa üzerine yazar, yoksa yeni dosya yaratır.
a	Ekleme modunda açar. Dosya varsa, dosya işaretçisi dosyanın sonundadır. Dosya yoksa yeniden yaratır.
ab	Binary bir dosya olarak ekleme modunda açar. Dosya varsa, dosya işaretçisi dosyanın sonundadır. Dosya yoksa yeniden yaratır.
a+	Hem ekleme hem de okuma modunda açar. Dosya mevcutsa, dosya işaretçisi dosyanın sonundadır. Dosya yoksa, yeniden yaratır.
ab+	Binary bir dosya olarak hem ekleme hem de okuma modunda açar. Dosya mevcutsa, dosya işaretçisi dosyanın sonundadır. Dosya yoksa, yeniden yaratır.

Dosyaya Yazma

- ▶ Python'da dosyaya yazmak için;

```
>>> fid.write()
```

komutu kullanılır.

- ▶ "write" komutu parametre olarak;
 - ▶ Yazım formatı
 - ▶ Yazılacak değişkenler

alır.

- ▶ Yazım formatı, yazılacak değişkeninin tipi, boşluk ve satır sonu karakteri, ondalık bir sayıysa noktadan sonra basılacak hane sayısı vb. içerir.

Formatlı Yazım

```
>>> write("Adet: %5d, Birim fiyat: %8.2f" % (12,54.70))
```

▶ Burada

- ▶ %5d, 5 haneye kadar bir tamsayı yazılacağını
- ▶ %8.2f, noktadan sonra iki adet olmak üzere toplam sekiz haneli ondalık bir sayı yazılacağını belirtiyor.
- ▶ Değişkenler bir demet (tuple) şeklinde verilmelidir.
- ▶ "Format" ile değişkenlerden arasında yer alan "%" operatörü unutulmamalıdır.

Dosyanın tüm satırlarını okuma

- ▶ Dosyanın tamamı okunacaksa for veya while döngüleri kullanılabilir:

- ▶ "while" ile;

```
line = fid.readline()
while line:
    line = fid.readline()
    print(line)
```

- ▶ "for" ile;

```
for line in fid.readlines():
    print(line)
```

Format Komutları

d	İşaretili tamsayı
i	İşaretili tamsayı
o	İşaretsiz sekiz tabanında sayı
u	İşaretsiz ondalık
x	İşaretsiz onaltı tabanında sayı (küçük)
X	İşaretsiz onaltı tabanında sayı (küçük)
e	Kayan noktalı sayı (küçük)
E	Kayan noktalı sayı (büyük)
f	Ondalık sayı
F	Ondalık sayı
g	Sayının içeriğine bağlı olarak "e" ve "f"
G	Sayının içeriğine bağlı olarak "e" ve "f"
c	Tek karakter
r	Metin tipi (Python nesnelere)
s	Metin tipi (Python nesnelere)
%	Yüzde işareti

Örnekler

```
>>> print("%10.3e"% (356.08977))
3.561e+02
>>> print("%10.3E"% (356.08977))
3.561E+02
>>> print("%10o"% (25))
31
>>> print("%10.3o"% (25))
031
>>> print("%10.5o"% (25))
00031
>>> print("%5x"% (47))
2f
>>> print("%5.4x"% (47))
002f
>>> print("%5.4X"% (47))
002F
>>> print("Only one percentage sign: %% " % ())
%
```

Örnekler

```
>>> print("%#5X"% (47))
```

```
0X2F
```

```
>>> print("%5X"% (47))
```

```
2F
```

```
>>> print("%#5.4X"% (47))
```

```
0X002F
```

```
>>> print("%#5o"% (25))
```

```
0o31
```

```
>>> print("%+d"% (42))
```

```
+42
```

```
>>> print("% d"% (42))
```

```
42
```

```
>>> print("%+2d"% (42))
```

```
+42
```

```
>>> print("% 2d"% (42))
```

```
42
```

```
>>> print("%2d"% (42))
```

```
42
```

Metin Tipi Değişkeni Formatlama

- ▶ Diğer bir yöntem ise, yazdırılacak metnin format komutuyla önceden hazırlanmasıdır.
- ▶ Bunun için metin tipi bir değişkenin içeriğinde {} parantezleri arasına sırasıyla argümanın numarası ve formatlanması istenen biçimi yazılır.

```
>>> "Adet: {0:5d}, Birim fiyat: {1:8.2f} ".format(12,54.70)
```

- ▶ Burada
 - ▶ İlk argümanın {0}, 5d olarak formatlanacağı,
 - ▶ İkinci argümanın {1} ise 8.2f olarak formatlanacağı belirtiliyor.
- ▶ Elde edilen formatlı metin doğrudan "write" komutu ile yazdırılabilir.

► Kaynakça

- 1 Wentworth, P., Elkner, J., Downey, A.B., Meyers, C. (2014). *How to Think Like a Computer Scientist: Learning with Python (3rd edition)*.
- 2 Pilgrim, M. (2014). *Dive into Python 3 by*. Free online version: DiveIntoPython3.org ISBN: 978-1430224150.
- 3 Summerfield, M. (2014) *Programming in Python 3 2nd ed (PIP3) :- Addison Wesley* ISBN: 0-321-68056-1.
- 4 Summerfield, M. (2014) *Programming in Python 3 2nd ed (PIP3) :- Addison Wesley* ISBN: 0-321-68056-1.
- 5 Jones E, Oliphant E, Peterson P, et al. *SciPy: Open Source Scientific Tools for Python, 2001-*, <http://www.scipy.org/>.
- 6 Millman, K.J., Aivazis, M. (2011). *Python for Scientists and Engineers, Computing in Science & Engineering, 13, 9-12*.
- 7 John D. Hunter (2007). *Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95*.
- 8 Travis E. Oliphant (2007). *Python for Scientific Computing, Computing in Science & Engineering, 9, 10-20*.
- 9 Goodrich, M.T., Tamassia, R., Goldwasser, M.H. (2013). *Data Structures and Algorithms in Python, Wiley*.
- 10 <http://www.diveintopython.net/>
- 11 <https://docs.python.org/3/tutorial/>
- 12 <http://www.python-course.eu>
- 13 <https://developers.google.com/edu/python/>
- 14 <http://learnpythonthehardway.org/book/>