

# YAPAY ZEKA VE PYTHON PROGRAMLAMA

*String İşlemleri*  
*Emir Öztürk*

Dr. Öğr. Üyesi Altan MESUT'un "Programlama Dillerine Giriş" ders slaytları kaynak alınmıştır.

## PYTHON STRINGLERİ

- Metin saklamak için
- Tek tırnak - çift tırnak
- input

Tek tırnak veya çift tırnak içinde yazılan her şey rakam bile olsa str (string) olarak saklanır.

Birçok programlama dilinde birden fazla karakterden oluşan diziler için string tanımı kullanılırken, Python'da tek bir karakterin saklandığı bir değişken bile str türünde kabul edilir (C'deki char gibi bir veri tipi yok).  
Input() fonksiyonunun döndürdüğü değer bir string'tir.

## STRING ÜZERİNDE + VE \* OPERATÖRLERİ

- String birleştirme +
- String tekrarlama \*

Python stringler ile "+" ve "\*" operatörlerinin kullanımına izin verir.

"+" operatörü solunda ve sağında yer alan string'leri birleştirmek için kullanılır.  
"\*" operatörü ise bir yanında str, diğer yanında int verilerek kullanılır ve verilen string'in verilen sayı kadar tekrarlanmasını sağlar.

## STRING - LİSTE

- String - Karakter listesi
- Elemanlara erişme
- Elemanları değiştirme

Listelerde olduğu gibi string’lerde de indisler ile işlem yapılabilir. Buna rağmen string’ler “immutable” oldukları için herhangi bir indisindeki değer değiştirilemez.

## STRING ELEMANLARI ÜZERİNDE GEZME

- for döngüsü
- stringin kendisi veya range
- range ile aralık verilebilmesi

String’in elemanlarına sıra ile erişilmek istendiğinde for döngüsü string’in kendisi üzerinden kurulabilir.

Eğer döngüde indisler ile işlem yapılmak istenirse, örneğin tüm karakterler yerine 2’şer atlayarak gezinmek istenirse; `range(0, len(string_adı), 2)` kullanılabilir.

## STRING YÖNTEMLERİ

- `metin.lower()`
- `metin.upper()`
- `metin.capitalize()`
- `metin.title()`
- `metin.replace()`
- `metin.split()`

`s.lower()`: String’in tümünü küçük harfe çevirir.

`s.upper()`: String’in tümünü büyük harfe çevirir.

`s.capitalize()`: Sadece ilk karakteri büyük harfe çevirir.

`s.title()`: Tüm kelimelerin ilk karakterlerini büyük harfe çevirir.

`s.replace(old, new, count)`: String içerisinde «old» ile verilen alt-string’i, «new» ile verilen alt-string ile değiştirir. 3. argüman olarak bir sayı verilirse bu işlemi sadece ilk «count» adet karakter için yapar.

`s.split(sep)`: String’i «sep» ile verilen bir karakter veya alt-string’lerden böler ve bir liste olarak döndürür. «sep» verilmezse boşluk karakterlerinden böler

(kelimelere ayırır).

#### STRING YÖNTEMLERİ

- `metin.center()`
- `metin.join()`
  - `join(liste)`
  - `join(string)`
- `metin.strip()`
  - `lstrip()`
  - `rstrip()`

`s.center(width, fillchar)`: String'i verilen genişlik içinde ortalar. İkinci parametre olarak sağına ve soluna doldurulacak karakter verilebilir (verilmediyse boşluk karakteri kullanılır)

`s.join(iterable)`: String'i verilen bir listenin her elemanı arasına (başka bir string verilirse her karakteri arasına) ekleyerek, oluşan yeni string'i döndürür.

`s.strip(chars)`: Argüman verilmezse string'in solunda ve sağındaki boşluk karakterlerini siler. Eğer «chars» verilirse, verilen karakterleri siler.

`lstrip()` yalnızca soldaki, `rstrip()` ise yalnızca sağdakileri siler

#### BOOL DEĞER DÖNDÜREN YÖNTEMLER

- `metin.isalnum()`
- `metin.isalpha()`
- `metin.isdecimal()`
- `metin.isdigit()`
- `metin.isnumeric()`
- `metin.isspace()`
- `metin.islower()`
- `metin.isupper()`

`s.isalnum()`: String'in alfanümerik olup olmadığını döndürür.

`s.isalpha()`: String'in alfabetik karakterlerden oluşup oluşmadığını döndürür.

`s.isdecimal()`: String'in içerisindeki tüm karakterlerin sayı olup olmadığını döndürür.

`s.isdigit()`: `isdecimal()`'den farkı Unicode kullanılarak alt simge ve üst simge biçiminde yazılan karakterler de dâhildir (Örn: "23").

`s.isnumeric()`: `isdigit()`'ten farkı Unicode kullanılarak yazılan "½" gibi özel sayısal karakterler de dâhildir.

`s.isspace()`: String'in boşluk veya TAB karakterlerinden oluşup oluşmadığını

döndürür.

s.islower() / s.isupper(): String'in tamamının küçük / büyük karakterlerden oluşup oluşmadığını döndürür.

#### ARAMA İLE İLGİLİ YÖNTEMLER

- `metin.find()`
- `metin.index()`
- `metin.count()`

s.find(sub, start, end): String içerisinde aranan başka bir string'in ilk bulunduğu noktanın indisini döndürür. Bulunamaz ise -1 döndürür. İstenirse «start» ve «end» değerleri verilerek o indisler arasında arama gerçekleştirilebilir.

s.index(sub, start, end): find() yönteminden farkı, aranan bulunamadığı zaman -1 değil hata döndürmesidir (Listelerde olduğu gibi).

s.count(sub, start, end): String içerisinde aranan başka bir string'in kaç defa yer aldığını döndürür.

#### KAÇIŞ KARAKTERLERİ

- String içerisinde belirtilmesi gereken durumlar
  - `\n`
  - `\\`
  - `\'`
  - `\"`
  - `\t`
- Kaçış tanımlaması - r harfi

`\n`: String içerisinde yeni satıra geçmek için kullanılır.

`\\`: `"\"` karakterini string içerisine yazdırmak için kullanılır.

`\'`: `'''` karakterini string içerisine yazdırmak için kullanılır.

`\"`: `"""` karakterini string içerisine yazdırmak için kullanılır.

`\t`: String içerisinde sekme koymak için kullanılır.

Bir string içerisinde kaçış karakterinin olmadığını belirtmek için stringin başına «r» harfi koyulur.

### S.FORMAT()

- Yazma sırasında + ile birleştirme yerine format
- Değişkenler için { }
- {Sayı} -> indis
- {İsim} -> değişken adı

Stringler yazılırken + ile birleştirmek yerine .format() yöntemi kullanılarak değişkenlerin geleceği yerler «{}» sembolleri ile belirtilebilir.

Değişkenlerin gelmesi istenen yere string içerisinde «{}» koyulur. Daha sonra format yönteminden sonra karşılık gelen değişkenler yazılır.

Küme parantezleri üç farklı şekilde kullanılabilir.

{ } içerisine hiçbir şey yazılmazsa format parametreleri sırası ile yazdırılır.

Küme parantezleri içerisine sayı yazılırsa ({sayi}) format parametrelerinin indislerine bakılarak yazma işlemi gerçekleştirilir.

Küme parantezlerinin içerisine tanımlayıcı yazılırsa ({isim}) format parametreleri bu tanımlara eşitlenerek yazma işlemi gerçekleştirilir.

### ÖRNEK

- Verilen bir cümledeki sesli karakter sayısını veren uygulama

## ÖRNEK

- Bir cümlede kaç kelime geçtiğini bulan uygulama

## ÖRNEK

- Bir kelimde her harften kaç tane olduğunu sayan uygulama