String Metodları

Şimdi karakter dizilerine (string) ait metotları inceleyeceğiz.Python'da bu metodları kullanarak istediğiniz karakter dizisi üstünde farklı farklı şekillerde oynayabilirsiniz.

1.1. LEN( ) METODU

Karakter dizisinin veya listelerin (ileride göreceğiz) uzunluğunu/boyutunu ölçmek için kullandığımız yöntemdir.
liste = [1,2,"hey", 'a']
len(liste) = 4

ornek = "DSC Galatasaray"
len(ornek) yazdığımızda 15 çıktısını alırız. Çünkü 14 tane harf ve bir de boşluk vardır. Boşluklar da birer karakter olarak sayılırlar.

1.2. UPPER( ) LOWER( ) ISUPPER( ) ISLOWER( ) METODLARI

Bir karakter dizisinin harfleriyle oynama imkanı sağlar. Harfleri büyük ya da küçük yazmamıza veya sorgulamamıza yardımcı olur.

Bir karakter dizisinin bütün harflerini büyük yazmak istersek kullanacağımız metod upper( ) 'dır.
ornek = "dsc galatasaray"
ornek.upper() dediğimizde programın bize DSC GALATASARAY diye bir çıktı vereceğini görebiliriz. upper( ) fonksiyonu, harflerin küçük/büyük harf olup olmadığına bakmaksızın o karakter dizisindeki bütün harfleri büyük yazar. Yani
ornek = "DSC Galatasaray" yazar ve ornek.upper() ile çalıştırırsak yine bir önceki gibi DSC GALATASARAY çıktısını alırız.

lower( ) metodu da bunun tam tersidir. Karakter dizisinin bütün harflerini küçük harfe çevirir.
ornek = "DSC GALATASARAY"
ornek.lower() diyip çalıştırdığımızda dsc galatasaray çıktısı ile karşılaşırız. lower( ) fonksiyonu yine upper( ) 'da olduğu gibi , harflerin küçük/büyük harf olup olmadığına bakmaksızın o karakter dizisindeki bütün harfleri küçük yazar. Yani
ornek = "DSC Galatasaray" yazar ve ornek.lower() ile çalıştırırsak bir önceki gibi dsc galatasaray çıktısını alırız.

isupper( ) ve islower( ) ikilisi ise bir çeşit sorgulama amaçlı kullanılır. Python'dan çıktı olarak True ya da False değerlerini alırız.

Kullanacağımız/elde ettiğimiz karakter dizisinin büyük harflerden mi yoksa küçük harflerden mi oluştuğuna emin olamıyorsak bu iki metod yardımımıza koşar.

isupper( ) bahsedilen dizinin bütün elemanlarının büyük mü olduğunu programa sorar.
ornek = "DSC GALATASARAY"
ornek.isupper() diyerek çalıştırırsak program bize True değerini döndürecektir. Ancak bu sefer sorgulanan karakter dizisinde bir tane bile bir harf küçük olursa False çıktısını alırız.
ornek = "Dsc Galatasaray"
ornek.isupper() diyip çalıştıralım ve aldığımız değere bakalım : False. Çünkü bütün harfleri büyük harf değil.

islower( )) da tam tersi iş yapar. Bu sefer programa, baktığımız dizinin bütün harflerinin küçük harf olup olmadığını sorarız.
ornek = "Dsc GALATASARAY"
ornek.lower() diyerek çalıştırırsak program bize False değerini döndürecektir. Çünkü bütün harfler küçük değildir.
ornek = "dsc galatasaray"
ornek.islower() diyip çalıştıralım ve aldığımız değere bakalım : True. Çünkü bütün harfleri küçük.

1.3. TITLE( ) CAPITALIZE( ) METODLARI

Yine bir karakter dizisindeki harflerin boyutları(büyüklük/küçüklük) oynamak istediğimizde başvurduğumuz metodlardandır.

title() metodu, bir karakter dizisindeki bütün kelimelerin baş harfini büyük yazar.
a = "python eğitim serisi"
a.title() bize Python Eğitim Serisi şeklinde bir çıktı verecektir.

Capitalize( ) metodu ise bir karakter dizisindeki ilk harfi büyük yazar.
a = "python eğitim serisi"

a.capitalize() yazıp çalıştırırsak Python eğitim serisi sonucunu elde ederiz.

1.4. STRIP( ) REPLACE( ) METODLARI

Bir karakter dizisinin düzeni üzerine oynama yapmak istediğimizde aklımıza gelen metodlardan ikisi de replace( ) ve strip( ) 'tir.

Dizideki harflerin yerini değiştirmek, daha doğrusu dizinin içindeki bir harfi başka bir harfle değiştirmek için replace( ) metodunu kullanırız.
a = "dsc galatasaray" karakter dizisindeki "a" harflerini "e" ile değiştirmek istersem şunu yaparım;
a.replace("a", "e") elde ettiğim sonuç şu şekilde olur : dsc geleteserey
Metodu yazarken sıralamaya dikkat etmemiz gerekiyor. Parantez içinde önce değiştirmek istediğimiz harfi sonra da yerine koymak istediğimiz harfi yazmalıyız. Ve çift tırnak içinde yazdığınıza emin olun.

Dizideki istenmeyen boşluk, sembol vs kırpmak için strip( ) metodunu kullanacağız.
a = " dsc galatasaray " bu sefer karakter dizisinin başında ve sonunda bir boşluk var. Hadi şimdi o boşluğu yok edelim.
a.strip(" ") tırnak arasına boşluk koyduk ve çalıştırdık. Aldığımız sonuç : 'dsc galatasaray' Fark ettiyseniz sadece başındaki ve sonundaki boşluklar yok oldu ancak dizinin ortasında (veya başka yerinde) bulunan boşluk tuşuna dokunulmadı. Bunun sebebi strip( ) metodunun sadece baştaki ve sondaki istenmeyen karakterleri kırpmasıdır.

b = "*dsc galatasaray" Bu sefer dizimin sadece başına * koydum ve şimdi onu kırpacağım.
b.strip("*") dsc galatasaray şeklinde * sembolünün kırpılmış halini elde ettik.

del metodunu kullanarak yapmış olduğumuz bir atama işlemini silebiliriz.
a = "dsc galatasaray"
del a
dediğimizde hafızadaki a değişkeni silinecektir.

Split( ) metodu, karakter dizisinde belirtilen bir karaktere göre parçalama işlemi yapar.
a = "dsc galatasaray"
a.split(" ") yazarsak program bize " " yani boşluk karakterinden önce ve sonra diye karakter dizisini ayırıp bir liste olarak verecektir.
Yani çıktımız şudur : ["dsc" , "galatasaray"]

EXTRA KAYNAKLAR:

PYTHON STRING METHODS:

1. https://www.programiz.com/python-programming/methods/string

DIR( )

Doğal olarak bütün metodları bilmemiz pek olası değil. Bazen sadece istediğimiz şeye göre metod araştırması yapmamız gerekiyor. Python'ın bu konuda bize sağladığı güzel bir özelliği var.
dir( ) fonksiyonunu kullanarak yazmış olduğumuz değişken üzerinde yapabileceğimiz işlemleri görebiliyoruz.

Mesela
a = "dsc galatasaray"
dir(a) dediğimizde program bize kullanabileceğimiz metodları verir.

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
```

```
'__hash__',
'__init__',
'__init_subclass__',
'__iter__',
'__le__',
'__len__',
'__lt__',
'__mod__',
'__mul__',
'__ne__',
'__new__',
'__reduce__',
'__reduce_ex__',
'__repr__',
'__rmod__',
'__rmul__',
'__setattr__',
'__sizeof__',
'__str__',
'__subclasshook__',
'capitalize',
'casefold',
'center',
'count',
'encode',
'endswith',
'expandtabs',
'find',
'format',
'format_map',
'index',
'isalnum',
'isalpha',
'isascii',
'isdecimal',
'isdigit',
'isidentifier',
'islower',
'isnumeric',
'isprintable',
'isspace',
'istitle',
'isupper',
'join',
'ljust',
'lower',
'lstrip',
'maketrans',
'partition',
'replace',
'rfind',
'rindex',
'rjust',
'rpartition',
'rsplit',
'rstrip',
'split',
'splitlines',
'startswith',
'strip',
'swapcase',
'title',
'translate',
'upper',
'zfill']
```

KENDINIZ DENEYIN

O halde, biraz oturup kendi istedğinize göre bir değişkene bir karakter dizisi (string) atayıp ve dir( ) fonksiyonu kullanarak çıkardığınız metodlardan bir kaçını seçip deneyin ve ne işe yaradıklarını görün.

İsterseniz buraya yazmış olduğum metodları kullanın ve ne işe yaradıklarını keşfedin.

'center'

'isnumeric'

'istitle'

'find'