

# 2\_Hafta Konu

March 1, 2021

## 0.0.1 2. Hafta Python Ders icerigi

Dr. Ibrahim KÖK (ikok@pau.edu.tr)

#### Veri Türleri

- String (String)
- Liste (List)
- Demet (Tuple)
- Sözlükler(Dictionary)

## 0.0.2 String Veri Türü

- String veri türü metinleri temsil etmek için kullanılmaktadır.
- Metinler tek veya çift tırnak ile tanımlanabilir.
- Tek tırnak içinde değiştirme karakteri (escape character) gerek kalmadan çift tırnak sembolü yada tersi kullanılabilir.
- Değiştirme karakteri geçersiz olmak için metnin önüne r sembolü

```
In [1]: print('merhaba')
        print("dunya")
```

```
merhaba
dunya
```

```
In [2]: print('Bana "merhaba" dedi!')
        print("Bana \"merhaba\" dedi!")
        print(r"c:\users\pau") # c:\users\pau
```

```
Bana "merhaba" dedi!
Bana "merhaba" dedi!
c:\users\pau
```

```
In [3]: print('Çok\nsatır\ntext') # 3 satır içerir \n new line-yeni satır
        print("""Çok
        satır
        text""")
```

```
Çok
satrl
metin
Çok
satrl
metin
```

## Aritmetik ilem ve indexleme/Dilimleme

- Aritmetik operatörler string veri tipinde farklı işlevlere sahiptir.
- Bir metnin herhangi bir karakterine ya da bir kısma köşeli parantezler ile erişilebilir.
- Köşeli parantez içerisinde tek bir indis değeri veya : sembolü ile ayrılmış bir aralık verilebilir.

```
In [4]: s1="Python"
        s2="Dersi"
```

```
In [5]: print(s1+s2)
```

```
PythonDersi
```

```
In [6]: print('Python' 'Dersi')
```

```
PythonDersi
```

```
In [7]: print(s1+" "+s2)
```

```
Python Dersi
```

```
In [8]: print(3*s1)
```

```
PythonPythonPython
```

## Indexleme/Dilimleme

	P		y		t		h		o		n		# string değişken sd
	0		1		2		3		4		5		# index no
	-6		-5		-4		-3		-2		-1		# index no

- [başlangıç : bitiş]

```
In [9]: sd="Python"
        print(sd[2])
```

```
t
```

```
In [11]: sd[6]="f" # string nesnesi içine eleman atamayı desteklemez.
```

```
-----  
TypeError                                Traceback (most recent call last)  
  
  <ipython-input-11-c35830b234b1> in <module>()  
----> 1 sd[6]="f"  
  
TypeError: 'str' object does not support item assignment
```

```
In [12]: print(sd[-4])
```

t

```
In [13]: print(sd[1:4]) # Dikkat: parantez ici ilk index degeri dahil, son deer dahil DEGL  
yth
```

```
In [14]: print(sd[2:])
```

thon

```
In [15]: print(sd[-4:])
```

thon

```
In [16]: print(sd[:4])
```

Pyth

### String veri türünün hazır metotlar / Built-in Functions

- **lower(), upper():** Metnin srasıyla küçük ve büyük harfli hallerini verir.
- **startswith(str), endswith(str):** Metnin baska bir metinle basladgn ve bittigini True ve False olarak verir.
- **find(str):** Metnin içinde baska bir metni arar; bulursa konumunu, bulamazsa -1 döndürür.
- **count(str):** Metnin içinde baska bir metnin kaç defa geçtiğini verir.
- **replace(eski, yeni):** Metnin içinde eslesen degerleri baska bir metinle degistirir.
- **split(ayraç):** Metni ayraca göre parçalara ayır ve sonucu liste olarak döndürür.
- **join(liste):** Bir listenin elemanların aralarına metni koyarak birleştirir.
- **len(str):** string uzunluğunu verir

- **strip()**: Metni basndaki ve sonundaki bosluklar olmadan verir.
- **capitalize(str)** ilk harfi büyük yapar

```
In [17]: sd2="Pamukkale üniversitesi"
```

```
In [18]: #lower(), upper(): Metnin srasyla kucuk ve buyuk harfli hallerini verir.
print(sd2.upper())
```

PAMUKKALE ÜNİVERSİTESİ

```
In [19]: print(sd2.lower())
```

pamukkale üniversitesi

```
In [21]: #startswith(str), endswith(str): Metnin baska bir metinle basladgn ve bittigini True ve
print(sd2.startswith("Pamuk"))
```

True

```
In [24]: print(sd2.endswith("tesiii"))
```

False

```
In [25]: #find(str): Metnin içinde baska bir metni arar; bulursa konumunu, bulamazsa -1 döndürür
print(sd2.find("üni")) # bulursa konumunu, bulamazsa -1 döndürür.
```

10

```
In [ ]: print(sd2.find("uni"))
```

```
In [ ]: #count(str): String içinde bir metnin kaç defa geçtiğini verir.
print(sd2.count('i'))
```

```
In [ ]: #replace(eski, yeni): Metnin icinde eslesen degerleri baska bir metinle degistirir.
print(sd2.replace('üniversitesi', 'University'))
```

```
In [31]: sd3="Pamukkale Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendislii"
```

```
In [32]: #strip(): Metni basndaki ve sonundaki bosluklar olmadan verir.
sd3="      Pamukkale üniversitesi      "
print(sd3.strip())
```

Pamukkale üniversitesi

```
In [33]: #split(ayraç): Metni ayraca göre parçalara ayrr ve sonucu liste olarak döndürür.
print(sd3.split(","))
```

```
['Pamukkale Üniversitesi', ' Mühendislik Fakültesi', ' Bilgisayar Mühendislii']
```

```
In [34]: liste=sd3.split(",")
        print(liste)
```

```
['Pamukkale Üniversitesi', ' Mühendislik Fakültesi', ' Bilgisayar Mühendislii']
```

```
In [35]: #join(liste): Bir listenin elemanlarn aralarına metni koyarak birleştirir.
        print(';'.join(liste))
```

```
Pamukkale Üniversitesi; Mühendislik Fakültesi; Bilgisayar Mühendislii
```

```
In [36]: ikinokta=';'.join(liste)
        print(ikinokta)
```

```
Pamukkale Üniversitesi; Mühendislik Fakültesi; Bilgisayar Mühendislii
```

```
In [37]: print(len(sd3))
```

```
70
```

### 0.0.3 List Veri Türü

```
In [39]: l = ['krmz', 'yesil', 'mavi', 'sar']
        print(l)
```

```
['krmz', 'yesil', 'mavi', 'sar']
```

```
In [40]: print(l[0]) # krmz
        print(l[1:3]) # ['yesil', 'mavi']
        print(len(l)) # 4
```

```
krmz
['yesil', 'mavi']
4
```

```
In [41]: l2 = l # yeni liste oluşturulmaz, her ikisi de ayn listeyi,! gösterir
        print(l2)
```

```
['krmz', 'yesil', 'mavi', 'sar']
```

```
In [42]: bos_liste = [] # bos liste olusturur
        bos_liste = list() # bos liste olusturur
        l3 = ['siyah', 'mor']
        print(l[2:] + l3) # ['mavi', 'sar', 'siyah', 'mor']

['mavi', 'sar', 'siyah', 'mor']
```

Liste elemanlar **for** döngüsü ile dolasabilir. Bir eleman listede olup olmadı da **in** isleci ile sorgulanabilir. Asagda **for** döngüsü ve **in** islecinin listelerde kullanm verilmistir.

```
In [43]: renkler = ['krmz', 'yesil', 'mavi']
```

```
In [44]: for renk in renkler:
        print(renk)
```

```
krmz
yesil
mavi
```

```
In [45]: print('yesil' in renkler) # True
        print('sar' in renkler) # False
```

```
True
False
```

### List veri türünün hazır metotlar / Built-in Functions

- **append(eleman):** Listenin sonuna eleman ekler.
- **insert(indis, eleman):** Listenin belirtilen indisine araya eleman ekler.
- **extend(list):** Listenin sonuna baska bir listeyi ekler. Orjinal liste degisir.
- **index(eleman):** Eleman listede arar, varsa konumunu döndürür. Yoksa ValueError hatas verir.
- **remove(eleman):** Verilen eleman listeden siler. Döngu içinde yapılmamalıdır.
- **pop():** method removes the specified index.
- **sort():** Listeyi sıralar. Orjinal liste degisir. Liste degistirilmeden sıralanmak isteniyorsa sorted(liste) kullanılmalıdır.
- **reverse():** Elemanların sırasını tersine çevirir.
- **pop(indis):** Verilen indisteki eleman siler ve geri döndürür. Indis verilmemisse son eleman için bu işlemi yapar.

```
In [46]: liste = ['Ali', 'Ahmet', 'Mehmet']
        liste.append('Burak') # sona Burak eklenir
        liste.insert(0, 'Hasan') # basa Hasan eklenir
        liste.extend(['Ayse', 'Fatma']) # listenin sonuna Ayse ve Fatma,! eklenir.
        print(liste) # ['Hasan', 'Ali', 'Ahmet', 'Mehmet', 'Burak', ,! 'Ayse', 'Fatma']
```

```
['Hasan', 'Ali', 'Ahmet', 'Mehmet', 'Burak', 'Ayse', 'Fatma']
```

```
In [47]: print(liste.index('Ahmet'))
```

```
2
```

```
In [48]: liste.remove('Burak')
         print(liste)
```

```
['Hasan', 'Ali', 'Ahmet', 'Mehmet', 'Ayse', 'Fatma']
```

```
In [49]: liste.pop(1) # Ali silinir
         print(liste)
```

```
['Hasan', 'Ahmet', 'Mehmet', 'Ayse', 'Fatma']
```

```
In [50]: print(liste) #
         print(sorted(liste)) #
         #liste.sort() # Liste sıralanms hali ile degisir
```

```
['Hasan', 'Ahmet', 'Mehmet', 'Ayse', 'Fatma']
```

```
['Ahmet', 'Ayse', 'Fatma', 'Hasan', 'Mehmet']
```

### **reverse = True/False**

- büyükten küçü-e-sondan baa isteniyorsa- True
- küçükten büyü-e-batan sona isteniyorsa- False

```
In [75]: thislist = [75,100, 50, 65, 82, 23]
         thislist.sort(reverse = True)
         print(thislist)
```

```
[100, 82, 75, 65, 50, 23]
```

```
In [76]: thislist = ["aabc", "xyz", "def", "bbaa", "klmm"]
         thislist.sort(reverse = True)
         print(thislist)
```

```
['xyz', 'klmm', 'def', 'bbaa', 'aabc']
```

```
In [104]: thislist[1]="oooo" # degistirilebilir.
          print(thislist)
```

```
['xyz', 'oooo', 'def', 'bbaa', 'aabc']
```

#### 0.0.4 Tuple Veri Türü

- İçerisinde birden fazla eleman saklayabilen fakat \*\* oluşturulduktan sonra degistirilemeyen veri tipleridir\*\*.
- Tuple oluşturmak için parantez sembolleri kullanılır.

```
In [77]: tup1 = tuple()
         tup2 = ('bir', 'iki')
         print(tup2)

('bir', 'iki')
```

```
In [78]: tup1 = ('physics', 'chemistry', 1997, 2000)
         tup2 = (11, 20, [55,66,77,88], 66, 77)
         tup3 = ("a", "b", "c", "d")
```

```
In [79]: print(tup2[2])

[55, 66, 77, 88]
```

```
In [80]: print(tup2[2][0])

55
```

```
In [81]: tpl = (1, 3.14, 'merhaba')
         for el in tpl: # Tüm elemanlar yazar
             print(el)

1
3.14
merhaba
```

```
In [82]: print(tpl[1]) # 3.14

3.14
```

```
In [83]: tpl[2] = 0 # Hata: tuple degistirilemez
```

-----

TypeError

Traceback (most recent call last)

```
<ipython-input-83-258b36678f4b> in <module>()
----> 1 tpl[2] = 0 # Hata: tuple degistirilemez
```

TypeError: 'tuple' object does not support item assignment



## Tuple veri türünün hazır metotlar / Built-in Functions

```
In [87]: # count()   aranan elemanın sayısını verir
         tupS1=tuple("deneme çalması")
         tupS2=("deneme çalması")
```

```
In [88]: print(tupS1)
```

```
('d', 'e', 'n', 'e', 'm', 'e', ' ', 'ç', 'a', 'l', 'ı', ' ', ' ', 'm', 'a', 's', 'ı')
```

```
In [90]: print(tupS2)
```

```
deneme çalması
```

```
In [60]: tupS.count('a')
```

```
Out[60]: 2
```

```
In [61]: # index() aranan elemanın index numarasını döndürür
         tupS.index('ç')
```

```
Out[61]: 7
```

### 0.0.5 Dictionary Veri Türü

- Sözlükler **anahtar:değer** çiftlerinin tutulduğu bir veri türüdür.
- Köşeli parantezlerin içine anahtarlar yazılarak değerlere ulaşılabilir. Tanımlarken süslü parantezler ya da dict() fonksiyonu kullanılır.

```
In [91]: # empty dictionary
         my_dict = {}
```

```
         # dictionary with integer keys
         my_dict = {'name': 'jack', 'age': 29}
```

```
         # dictionary with mixed keys
         my_dict2 = {'name': 'John', 1: [2, 4, 3]}
```

```
In [92]: print(my_dict)
```

```
{'name': 'jack', 'age': 29}
```

```
In [94]: print(my_dict2["name"])
```

```
John
```

```

In [63]: sozluk=dict()
          sozluk['a']='alpha'
          sozluk['o']='omega'
          sozluk['g']='gamma'
          print(sozluk)

{'a': 'alpha', 'o': 'omega', 'g': 'gamma'}

In [64]: for anahtar in sozluk:
          print(anahtar + ':' + sozluk[anahtar])

a:alpha
o:omega
g:gamma

In [65]: yas=dict()
          yas['ali']='30'
          yas['veli']='34'
          yas['seda']='22'

In [66]: print(yas['ali'])

30

In [95]: hgun=dict()
          hgun[1]='Pazartesi'
          hgun[2]='Sal'
          hgun[3]='Çaramba'
          hgun[4]='Perembe'
          hgun[5]='Cuma'
          hgun[6]='Cumartesi'
          hgun[7]='Pazar'

In [96]: print(hgun[3])

Çaramba

In [101]: hgun[2]="tanmsz"    # degistirilebilir

In [102]: print(hgun[2])

tanmsz

```