

2. String veri tipi

```
x1 = 'Mustafa Kemal' #tek tırnak ile string tanımlama
x1

'Mustafa Kemal'

x2 = "Mustafa Kemal" # çift tırnak ile string tanımlama
x2

'Mustafa Kemal'

x3 = """Mustafa Kemal""" # üç tane çift tırnak ile string tanımlama
x3

'Mustafa Kemal'

print(x1)
print(x2)
print(x3)

Mustafa Kemal
Mustafa Kemal
Mustafa Kemal

'Kaan'ın bugün dersi var # tek tırnak ve kesme işareti sorunsalı
Cell In[5], line 1
    'Kaan'ın bugün dersi var # tek tırnak ve kesme işareti sorunsalı
    ^
SyntaxError: invalid syntax

"Kaan'ın bugün dersi var" # sorunsalı çift tırnak ile aşmak
"Kaan'ın bugün dersi var"

'Kaan\'ın bugün dersi var' # \ (ters slash ile sorunsalı aşmak)
"Kaan'ın bugün dersi var"
```

string değişkenler toplanabilir ve çarpılabilir.

```
a = "Fener"
b = "Bahçe"

print(a, b)

Fener Bahçe
```

```
c = a + b
c
'FenerBahçe'

d = a + " " + b
d
'Fener Bahçe'

c * 5
'FenerBahçeFenerBahçeFenerBahçeFenerBahçeFenerBahçe'
```

Bazı string metotları

```
x = "çay"
x
'çay'

x.capitalize()
'Çay'

y = x.capitalize()
y
'Çay'

y * 4
'ÇayÇayÇayÇay'

x.center(9)
'   çay   '

x.join("***")
'*çay*'

x.join("*+-+*")
'*çay+çay-çay-çay+çay*'
```

Python'da çok sayıda string metodu vardır. Bunlardan bazıları şunlardır:

- `upper()`: String nesnesindeki tüm karakterleri büyük harfe çevirir.
- `lower()`: String nesnesindeki tüm karakterleri küçük harfe çevirir.

- `capitalize()`: String nesnesindeki ilk karakteri büyük harfe, geri kalanını küçük harfe çevirir.
- `title()`: String nesnesindeki her kelimenin ilk harfini büyük harfe çevirir.
- `strip()`: String nesnesinin başında ve sonunda bulunan boşluk karakterlerini siler.
- `replace(old, new)`: String nesnesindeki old parametresiyle belirtilen karakterleri new parametresiyle belirtilen karakterlerle değiştirir.
- `find(sub)`: String nesnesinde sub parametresiyle belirtilen alt karakter dizisinin ilk bulunduğu indeksi döndürür. Eğer bulamazsa -1 döndürür.
- `index(sub)`: String nesnesinde sub parametresiyle belirtilen alt karakter dizisinin ilk bulunduğu indeksi döndürür. Eğer bulamazsa ValueError hatası verir.
- `split(sep)`: String nesnesini sep parametresiyle belirtilen ayraç karakterine göre bölümlere ayırır ve bir liste olarak döndürür. Eğer sep parametresi verilmezse boşluk karakterine göre bölümler.
- `join(iterable)`: Iterable parametresiyle belirtilen bir dizi veya liste gibi yinelenebilir bir veri tipinin elemanlarını string nesnesiyle birleştirerek yeni bir string nesnesi döndürür.

```
x
'çay'
x.upper()
'ÇAY'
y = x.upper()
y
'ÇAY'
z = y.lower()
z
'çay'
u = "vatanını en çok seven görevini en iyi yapandır"
u
'vatanını en çok seven görevini en iyi yapandır'
u.title()
'Vatanını En Çok Seven Görevini En İyi Yapandır'
```

```
t = " strip "  
t  
' strip '  
t.strip()  
'strip'  
x.replace("ç", "t") # x değişmez !  
'tay'  
x.replace("çay", "haydar") # x değişmez !  
'haydar'  
x.find("y")  
2  
x.index("y")  
2
```

String indeksleme ve parçalama

Python'daki tüm veri tipleri indekslidir ve diğer veri tipleri için aynı kurallar geçerlidir.

```
u = "Disk"  
u  
'Disk'  
u[2]  
's'  
u[4]  
-----  
-----  
IndexError                                Traceback (most recent call  
last)  
Cell In[34], line 1  
----> 1 u[4]
```

```
IndexError: string index out of range
```

soldan sağa doğru artan sırada, sağdan sola doğru -1 ile başlayarak azalan sırada

```
u[-1]
```

```
'k'
```

```
u[-2]
```

```
's'
```

u = Disk olarak tanımlanan string bir kelime dizesidir ve dize 0 indisinden başlar. Yani dizinin 0. elemanı, D, 1. elemanı i, 2. elemanı s ve üçüncü elemanı da k' dir. Dizide 4. eleman olmadığından u[4] komutu hata vermiştir. Dizinin elemanları çağrılırken, eleman numarası köşeli parantez ile çağırılır.

```
G = "Ali topu at"
```

```
G
```

```
'Ali topu at'
```

```
G[0]
```

```
'A'
```

```
G[3]
```

```
' '
```

```
W = "Zara"
```

```
W
```

```
'Zara'
```

```
W[-1]
```

```
'a'
```

```
W[-2]
```

```
'r'
```

String parçalama

Uzun bir string yapısının sadece belirli bir parçası alınmak istenirse aşağıdaki komut yapısı kullanılır: [başlangıç indeksi : bitiş indeksi : atlama değeri]

```
Q = "Çiğdem Talu & Melih Kibar "  
Q
```

```
'Çiğdem Talu & Melih Kibar '
```

```
len(Q)
```

```
26
```

```
Q[0:3]
```

```
'Çiğ'
```

```
Q[7:14]
```

```
'Talu & '
```

```
Q
```

```
'Çiğdem Talu & Melih Kibar '
```

```
Q[0:26:3]
```

```
'Çd l&ehir'
```

```
Q[::3]
```

```
'Çd l&ehir'
```

```
Q[:14]
```

```
'Çiğdem Talu & '
```

```
Q[4:]
```

```
'em Talu & Melih Kibar '
```

```
Q[:]
```

```
'Çiğdem Talu & Melih Kibar '
```

```
Q[:-1]
```

```
'Çiğdem Talu & Melih Kibar'
```

```
Q[::2]
```

```
'Çğe au&MlhKbr'
```

```
Q[::1]
```

```
'Çiğdem Talu & Melih Kibar '
```

```
Q[::4]
```

```
'Çea&lKr'
```

```
Q[::-1]
```

```
' rabiK hileM & ulaT medğiÇ'
```

Bir stringin uzunluğunu nasıl belirleyebiliriz? Bu amaçla len() fonksiyonu kullanılır. (len=length kelimesinin kısaltmasıdır.)

```
S = "Diğer makine de hesap yapıyor"  
S
```

```
'Diğer makine de hesap yapıyor'
```

```
len(S)
```

```
7
```

Bir stringin değişimi kolay değildir. Aşağıda buna bir örnek verilmiştir.

```
S = "Mertaba"  
S
```

```
'Mertaba'
```

```
S[3] = "h"
```

```
-----  
-----
```

```
TypeError                                Traceback (most recent call  
last)
```

```
Cell In[62], line 1
```

```
----> 1 S[3] = "h"
```

```
TypeError: 'str' object does not support item assignment
```

```
ss = S.replace("t", "h")
```

```
print(S)
```

```
print(ss)
```

```
Mertaba
```

```
Merhaba
```

veri tipi dönüşümleri

Tam sayıyı ondalıklı sayıya dönüştürme

float() fonksiyonu, tam sayının ondalıklı yazılmasını olanaklı kılar

```
a = 97
float(a)
```

```
97.0
```

```
b = float(a)
print(a)
print(b)
```

```
97
97.0
```

```
bb = int(b)
print(a)
print(bb)
```

```
97
97
```

```
c = "Kıraç"
float(c)
```

```
-----
-----
ValueError                                Traceback (most recent call
last)
Cell In[67], line 2
      1 c = "Kıraç"
----> 2 float(c)
```

```
ValueError: could not convert string to float: 'Kıraç'
```

Ondalıklı sayıyı tam sayıya dönüştürme

```
a = 3.14
int(a)
```

```
3
```


Sayıların stringe dönüştürülmesi

```
type(23022020)
int
str(23022020)
'23022020'
type(str(23022020))
str
int(str(23022020))
type(int(str(23022020)))
int
j = str(23022020)
len(j)
8
h = str(3.14)
h
'3.14'
len(h)
4
```

Stringi tam sayıya dönüştürme

int() fonksiyonu kullanılarak string değişkeni tam sayıya dönüştürülür.

```
a = "4398467"
a
'4398467'
print(a)
type(a)
4398467
str
int(a)
```

4398467

Karakterler 10' luk tabansa sayı olmalıdır

```
a = "23ab"  
aa = int(a)  
aa
```

```
-----  
-----  
ValueError                                Traceback (most recent call  
last)
```

```
Cell In[79], line 2  
      1 a = "23ab"  
----> 2 aa = int(a)  
      3 aa
```

ValueError: invalid literal for int() with base 10: '23ab'

```
type(aa)
```

```
-----  
-----  
NameError                                Traceback (most recent call  
last)
```

```
Cell In[80], line 1  
----> 1 type(aa)
```

NameError: name 'aa' is not defined

```
type(a)
```

```
str
```