PYTHON PROGRAMLAMA DİLİ

VERİ TİPLERİ
int-float-str

Dr. Öğr. Üyesi M. Ozan AKI

VERİ TİPLERİ

Python dilinde, diğer programlama dillerinde olduğu gibi farklı veri tipleri bulunmaktadır.

int, float ve **str** sadece bir değişken tanımlarken,

List, **Tuple**, **Dictionary** ve **Set** veri tipleri, çok sayıda veriyi sistematik bir şekilde saklamak ve işlemek için kullanılmaktadır.

Veri Tipi	Açıklama
`int`	Tam sayı 123
`float`	Ondalıklı sayılar 12.34
`str`	Karakter dizileri '…' "…"
`List`	Listeler […]
`Tuple`	Demetler (…)
`Dictionary`	Sözlük {…}
`Set`	Küme {…}

VERİ TİPLERİ

int

(integer)

Tamsayıları saklamak ve üzerinde işlem yapmak için kullanılır.

```
>>> a=3
>>> b=5
>>> vize=75
>>> sayac=1500
>>> x,y=12,34
>>> p,q,r,s=15,34,65,87
>>>
```

VERİ TİPLERİ

float

(floating point)

Ondalıklı sayıları saklamak ve üzerinde işlem yapmak için kullanılır.

Ondalık nokta her zaman noktadır.
(virgül kullanılmaz!)

Bilimsel formatta da giriş yapılabilir.

1e6 == $1x10^{6}$

1e-3 == $1x10^{-3}$

```
>>> x=5.67
>>> pi=3.14159
>>> ortalama=83.4
>>> volt,akim=12.5,0.25
>>> carpan=0.000001
>>> print(carpan)
    1e-06
>>> katsayi=2.5e-3
>>> print(katsayi)
    0.0025
>>> guc=3.45e6
>>> print(guc)
    3450000.0
>>>
```

VERİ TİPLERİ

str

(string)

Karakter dizilerinden oluşan string ifadeleri saklamak ve üzerinde işlem yapmak için kullanılır.

String veri tipleri '…' (tek tırnak) ya da "…" çift tırnak içerisinde tanımlanır.

```
>>> ad_soyad="Müzeyyen Sözügüzel"
>>> bolum='Elektronik ve Otomasyon'
>>> okul_adi="Ipsala Meslek Yuksekokulu"
>>> ConnectionString="Server=192.168.1.22;Database=myDB;User Id=dbUser;Password=1234;"
>>> Cumle="Baris Manco'nun "+'"Lahburger" isimli eseri "24 Ayar" albumundedir'
>>> print(Cumle)
Baris Manco'nun "Lahburger" isimli eseri "24 Ayar" albumundedir
>>>
```

VERİ TİPİ DÖNÜŞÜMLERİ

Veri tiplerinin dönüştürülmesi

Bazı durumlarda veri tiplerinin birbirine dönüştürülmesi gerekir.

Veri tipi adının yanına parantez açılırsa, parantez içindeki değer yazılan tipe dönüştürülür.

Dönüşüm mümkün değilse hata oluşur ve programın çalışması kesilir.

```
>>> a=3
>>> print(a)
3
>>> b=float(a)
>>> print(b)
3.0
>>>
>>> x=3.14
>>> y=int(x)
>>> print(y)
3
>>>
```

```
>>> giris=input("sayi girin:")
sayi girin:78
>>> type(giris)
<class 'str'>
>>> sayi=int(giris)
>>> type(sayi)
<class 'int'>
>>> print(sayi)
78
>>>
>>> giris=input("Volt girin:")
Volt girin:4.5
>>> type(giris)
<class 'str'>
>>> volt=float(giris)
>>> type(volt)
<class 'float'>
>>> print(volt)
4.5
>>>
```

STR (STRING) VERİ TİPİ

str (string) Veri Tipi ile İşlemler

String ifadeler üzerinde bazı operatörler ve string nesnesinin metotları kullanılarak işlemler yapılabilir.

+: iki yanına yazılan string ifadeleri birleştirmek için kullanılır.

*: yanına yazılan sayı kadar string ifade kendini tekrar eder.

```
>>> ad='Mustafa'
>>> soyad='Kemal'
>>> ad_soyad=ad+' '+soyad
>>> print(ad_soyad)
Mustafa Kemal
>>>
>>> cumle="Ali'nin"+' bilgisayari "Asus" marka.'
>>> print(cumle)
Ali'nin bilgisayari "Asus" marka.
>>>
>>> gul='Ha '*5
>>> print(gul)
Ha Ha Ha Ha Ha
>>>
>>> cizgi=20*'-'
>>> print(cizgi)
--------------------
>>>
```

STR (STRING) VERİ TİPİ - METOTLAR

str (string) Veri Tipi Metotları

String veri tipinde, elemanalar üzerinde işlem yapan metotlar bulunur.

Bu metotlar, string karakterleri üzerinde işlem yaparak sonucu yeni bir string değişken olarak döndüdürler.

```
str.lower()
str.upper()
str.capitalize()
str.replace()
str.split()
str.splitlines()
str.center()
str.join()
str.strip()
str.lstrip()
str.rstrip()
str.find()
str.index()
str.count()
```

```
str.isalnum()
str.isalpha()
str.isdecimal()
str.isdigit()
str.isnumeric()
str.isspace()
str.islower()
str.isupper()
str.isascii()
str.isidentifier()
str.isprintable()
str.istitle()
```

STR (STRING) VERİ TİPİ - METOTLAR

str.lower(): Tümünü küçük harfe çevirir.

str.upper(): Tümünü BÜYÜK harfe çevirir.

str.capitalize(): Sadece ilk harfleri büyük yapar.

str.title(): Her Kelimenin İlk Harfini Büyük Yapar.

str.replace(eski, yeni): eski ile verilen ifadeyi yeni ile değiştirir.

str.center(width, fillchar): strin ifadeyi, with ile verilen genişlik içine ortalar. Fillcahr belirtildiyse, sağ ve solda kalan boşluklar bu karater ile doldurulur.

str.ljust(width,fillchar): string ifadeyi width ile belirtilen genişlik içine sola yaslı yazar. fillchar verildiyse boşlukları bu karaterle doldurur.

str.rjust(width,fillchar): string ifadeyi width ile belirtilen genişlik içine sağa yaslı yazar. fillchar verildiyse boşlukları bu karaterle doldurur.

s.join(x): s ile verilen strin ifadeyi x ile verilen string ifadenin her bir elemanının arasına yerleştirir ve yeni string döndürür.

```
>>> cumle='Trakya Universitesi'
>>> print(cumle)
Trakya Universitesi
>>> print(cumle.lower())
trakya universitesi
>>> print(cumle.upper())
TRAKYA UNIVERSITESI
>>> print(cumle.capitalize())
Trakya universitesi
>>> print(cumle.title())
Trakya Universitesi
>>> print(cumle.center(30))
      Trakya Universitesi
>>> print(cumle.ljust(30,'_'))
Trakya Universitesi__________
>>> print(cumle.rjust(30,'-'))
----------Trakya Universitesi
>>> print("_".join(cumle))
T_r_a_k_y_a_ _U_n_i_v_e_r_s_i_t_e_s_i
>>>
```

STR (STRING) VERİ TİPİ - METOTLAR

str.split(): İfadeyi boşluklardan bölerek kelimelerden oluşan bir liste olarak döndürür.

str.split(k): İfadeyi k bulunan yerlerden böler ve bir liste olarak döndürür.

str.splitlines(): İfadeyi satır sonlarından bölerek satırlardan oluşan liste döndürür.

str.strip(): String ifadenin başında ve sonundaki boşlukları siler.

str.lstrip(): String ifadenin başındaki boşlukları siler.

str.lstrip(): String ifadenin sonundaki boşlukları siler.

str.strip(c): String ifadenin başında ve sonundaki c ile belirtilen karakterleri siler.

str.lstrip(c): String ifadenin başındaki c ile belirtilen karakterleri siler.

str.rstrip(c): String ifadenin sonundaki c ile belirtilen karakterleri siler.

```
>>> cumle='   Trakya Universitesi   '
>>> print(cumle.split())
['Trakya', 'Universitesi']
>>> print(cumle.split('e'))
['   Trakya Univ', 'rsit', 'si   ']
>>> print(cumle.strip())
Trakya Universitesi
>>> print(cumle.lstrip())
Trakya Universitesi
>>> print(cumle.rstrip())
   Trakya Universitesi
>>>
```

STR (STRING) VERİ TİPİ - METOTLAR

str.startswith(s): string ifadenin s ile başlayıp başlamadığını döndürür.

str.endswith(s): string ifadenin s ile bitip bitmediğini döndürür.

str.replace(eski, yeni): eski ile verilen ifadeyi yeni ile değiştirir.

```
>>> cumle='Trakya Universitesi'
>>> print(cumle.startswith('Tra'))
True
>>> print(cumle.startswith('Nam'))
False
>>> print(cumle.endswith('si'))
True
>>> print(cumle.endswith('mek'))
False
>>> print(cumle.replace('Trakya','Anadolu'))
Anadolu Universitesi
>>> print(cumle.replace('i','o'))
Trakya Unoversoteso
>>>
```

STR (STRING) VERİ TİPİ - METOTLAR

str.find(sub): String ifade içinde sub ile verilen diğer string ifadeyi arar. Bulursa başlangıç indeks numarasını döndürür. Bulamazsa -1 döndürür.

str.rfind(sub): String ifade içinde sub ile verilen diğer string ifadeyi sağdan sola doğru arar. Bulursa başlangıç indeks numarasını döndürür. Bulamazsa -1 döndürür.

str.find(sub,start,end): String ifade içinde sub ile verilen diğer string ifadeyi indeksinden end indeksine kadar olan kısmında arar. Bulursa başlangıç indeks numarasını döndürür. Bulamazsa -1 döndürür.

```
>>> cumle='Trakya Universitesi'
>>> cumle.find('ya')
4
>>> cumle.find('si')
13
>>> cumle.rfind('si')
17
>>> cumle.find('ya',6)
-1
>>> cumle.find('ni',6)
8
>>> cumle.find('si',6,14)
-1
>>> cumle.find('si',6,16)
13
>>>
```

STR (STRING) VERİ TİPİ - METOTLAR

str.index(sub): String ifade içinde sub ile verilen diğer string ifadeyi arar. Bulursa başlangıç indeks numarasını döndürür. Bulamazsa hata verir.

str.index(sub,start,end): String ifade içinde sub ile verilen diğer string ifadeyi indeksinden end indeksine kadar olan kısmında arar. Bulursa başlangıç indeks numarasını döndürür. Bulamazsa hata verir döndürür.

str.count(sub): String ifade içinde kaç tane sub ile verilen diğer string ifade olduğunu sayarak döndürür.

str.count(sub, start, end): String ifade içinde, start ile end indisleri arasında kaç tane sub ile verilen diğer string ifade olduğunu sayarak döndürür.

```
>>> cumle='Trakya Universitesi'
>>> cumle.index('Universitesi')
7
>>> cumle.index('si')
13
>>> cumle.index('si',6)
13
>>> cumle.index('si',14)
17
>>> cumle.count('i')
3
>>> cumle.count('e')
2
>>> cumle.count('a')
2
>>> cumle.count('a',6)
0
>>> cumle.count('i',6,15)
2
>>>
```

STR (STRING) VERİ TİPİ - METOTLAR

str.isalnum(): string ifadenin alfanumerik olup olmadığını döndürür.
(alfanumerik: sadece harf ve rakamlardan oluşan ifade, içinde özel semboller yok)

str.isalpha(): string ifadenin sadece alfabetik karakterlerden oluşup oluşmadığını döndürür.

str.isdecimal(): string içindeki tüm karakerlerin sayı olup olmadığını döndüdür.

str.isdigit(): string içindeki tüm karakerlerin sayı olup olmadığını döndüdür.

(Unicode içindeki 2_3 ve 2^3 gibi alt ve üst karakter sayıları da dahildir dahildir)

str.isnumeric(): string içindeki tüm karakerlerin sayı olup olmadığını döndüdür. *(Unicode içindeki ½, ¼ gibi karaterler de dahildir)*

str.isspace(): string ifadenin boşluk ve/veya tab karakterlerinden oluşup oluşmadığını döndürür.

str.islower(): string ifadenin tüm karakterlerinin küçük harflerden oluşup oluşmadığını döndüdür.

str.isupper(): string ifadenin tüm karakterlerinin BÜYÜK harflerden oluşup oluşmadığını döndüdür.

STR (STRING) VERİ TİPİ - METOTLAR

```
>>> kelime='trakya'
>>> kelime.isalnum()
True
>>> kelime.isalpha()
True
>>> kelime.isdecimal()
False
>>> kelime.isdigit()
False
>>> kelime.isnumeric()
False
>>> kelime.isspace()
False
>>> kelime.islower()
True
>>> kelime.isupper()
False
>>>
```

```
>>> kelime='EDIRNE22'
>>> kelime.isalnum()
True
>>> kelime.isalpha()
False
>>> kelime.isdecimal()
False
>>> kelime.isdigit()
False
>>> kelime.isnumeric()
False
>>> kelime.isspace()
False
>>> kelime.islower()
False
>>> kelime.isupper()
True
>>>
```

```
>>> kelime='123'
>>> kelime.isalnum()
True
>>> kelime.isalpha()
False
>>> kelime.isdecimal()
True
>>> kelime.isdigit()
True
>>> kelime.isnumeric()
True
>>> kelime.isspace()
False
>>> kelime.islower()
False
>>> kelime.isupper()
False
>>>
```

STR (STRING) VERİ TİPİ – KAÇIŞ KARAKTERİ

Kaçış Karaklerleri (Escape Characters)

\n: String içinde yeni satıra geçer

\t: String içine **Tab** karakerini ekler.

****: String içine **** (ters bölü) karakterini ekler.

\': String içine **'** (tek tırnak) karakterini ekler

\": String içine **"** (çift tırnak) karakerini ekler

```
>>> cumle='bir\niki\nuc'
>>> print(cumle)
bir
iki
uc
>>> cumle='bir\tiki\tuc'
>>> print(cumle)
bir     iki     uc
>>> cumle='\'bir\'\t\"iki\"\t\'uc\''
>>> print(cumle)
'bir'   "iki"   'uc'
>>>
```

Bir string ifadenin başına **r** konursa, o string içinde Escape (kaçış) karakeri olmadığı anlamına gelir.

Bir string ifadenin başına **f** konursa, formatlı string olduğu anlaşılır ve içindeki {…} küme parantezleri değişken yer tutucu olarak değerlendirilir.