



PROGRAMLAMA TEMELLERİ

Öğr. Gör. Erhan AKAGÜNDÜZ

VERİ TİPLERİ

- ❑ Python'da genel olarak;
 - string (metinsel),
 - numbers (sayısal),
 - list (liste),
 - tuple (demet),
 - dictionary (sözlük) ve
 - set (küme)
- ❑ veri tipleri bulunmaktadır.

STRING (METİNSEL) VERİ TİPİ

- ❑ Tek ya da çift tırnak içlerine yazılan karakter dizileridir.
- ❑ Burada karakter harf (t,c), rakam (1,9,2,3) ya da özel semboller (&,/) olabilir.
- ❑ String veri tipleri tek ya da çift tırnak içinde yazılır.
- ❑ Örneğin aşağıdaki iki ifade birbirinin aynısıdır.
- ❑ `print ("Bütün ümidim gençliktedir.")`
- ❑ `print ('Bütün ümidim gençliktedir.')`
- ❑ Her iki kod satırı çalıştırıldığında aynı çıktı üretilir.

STRING (METİNSEL) VERİ TİPİ

- ❑ String bir değişken tanımlama işlemi aşağıdaki gibi yapılır:
 - **yasadiginiz_sehir** = "Van"
 - **ad_soyad** = "Ahmet Mehmetoğlu"
- ❑ Python'da **type()** fonksiyonu kullanılarak veri tipi öğrenilebilir.

```
okul_turu = "Meslek Lisesi"  
print(type(okul_turu))
```

```
<class 'str'>
```

NUMBERS (SAYISAL) VERİ TİPLERİ

- ❑ Sayısal verileri tutan veri tiplerine verilen addır.
- ❑ Python'da sayısal veri tipleri genel olarak **int**, **float** ve **complex** veri tipleridir.
- ❑ **int veri tipi tam sayı** değerleri tutar.
- ❑ **float veri tipi ondalıklı** değerleri tutar.
- ❑ Bu noktada tüm tam sayıların da ondalıklı olarak ifade edilebileceğini unutmayınız.
- ❑ Örneğin 3 tam sayısı (normalde int) 3.00 şeklinde ifade edildiğinde float olarak da tanımlanabilir.

NUMBERS (SAYISAL) VERİ TİPLERİ

- ❑ **Örnek:** **sayi** isminde değeri 1919 olan bir değişken tanımlayarak ekrana yazdırınız.

```
sayi = 1919  
print(sayi)
```

1919

- ❑ **Örnek:** **pi_degeri** isminde değeri 3.14 olan bir değişken tanımlayarak ekrana yazdırınız.

```
pi = 3.14  
print(pi)
```

3.14

NUMBERS (SAYISAL) VERİ TİPLERİ

- ❑ Python'da kullanılan bir diğer veri tipi de **bool** veri tipidir.
- ❑ Kod yazarken bazı ifadelerin **doğru** ya da **yanlış** olarak değerlendirilmesi istenebilir.
- ❑ Bu durumlarda yalnızca **True (doğru)** ve **False (yanlış)** değerlerini döndüren **bool** veri tipi kullanılır.

INPUT() FONKSİYONU İLE KULLANICIDAN VERİ ALMA

- ❑ Programlamada bazı değerlerin kullanıcılar tarafından girilmesi gerekebilir.
- ❑ Kullanıcıdan değer almak için **input()** fonksiyonu kullanılır.
- ❑ **Örnek:** Kullanıcıya yaşını sorunuz ve girilen yaşı ekrana yazdırınız.

```
yas = int(input("Yaşınızı giriniz: "))  
print(yas)
```

Yaşınızı giriniz:

```
yas = int(input("Yaşınızı giriniz: "))  
print(yas)
```

Yaşınızı giriniz: 25
25

VERİ TİPİ DÖNÜŞÜMLERİ

- ❑ Python'da bir değişkenin ya da değerin tipini başka bir veri tipine dönüştürmeniz gerekebilir.
- ❑ Örneğin **float** olarak tanımlanan bir değişkeni (örneğin 3.14) programın herhangi bir yerinde **tam sayı** olarak (3) kullanmanız gerekebilir.
- ❑ Bu durumda float olan bu değeri int'e çevirmeniz gerekmektedir.

YORUM SATIRLARI

- ❑ Programlamada bazen kod satırına açıklama yapmak ya da yorum yazmak gerekebilir.
- ❑ Python'da yorum satırları için **#** işareti kullanılır.

```
faiz_orani = 1.24 # float türünde bir değişken tanımlandı.
```

LIST (LİSTELER)

- ❑ Farklı verilerin bir dizi hâlinde tutulduğu koleksiyonlara **liste** adı verilir.
- ❑ Daha önce **int**, **float**, **string** gibi veri türlerini öğrenmiştiniz.
- ❑ Bu veri tiplerini kullanarak tek bir veriyi tutabilirsiniz.
- ❑ Birden fazla veriyi sıralı ve değiştirilebilen bir yapıda tutmak için listeler kullanılır.
- ❑ Listeler ile farklı veri tiplerini tutabilirsiniz.
- ❑ Python programlama dilinde listeler iki köşeli parantez [] ile tanımlanmaktadır.

LIST (LİSTELER)

❑ Örnek:

```
ilk_listemiz = ["Van", 65, 0.6]
print(ilk_listemiz)

['Van', 65, 0.6]
```

- ❑ Liste veri tipi için tanımlanan ilk liste incelendiğinde sırasıyla **string**, **integer** ve **float** tiplerinin bir arada kullanıldığı görülmektedir.
- ❑ Üç elemanı olan bu listeyi yazdırmak için daha önce kullanılan **print()** fonksiyonunu kullanmanız gerekmektedir.
- ❑ Bu örneğin liste veri tipinde olduğunu doğrulamak için **type()** fonksiyonu kullanılabilir.
- ❑ **type(ilk_listemiz)** kod satırı çalıştırıldığında ekran çıktısı **<class 'list'>** olacaktır.

İNDEKS KULLANIMI

- ❑ Liste içindeki elemanlara erişmek için ilgili elemanın **indeksi** kullanılır.
- ❑ Bazı kaynaklarda indis olarak da karşınıza çıkabilir.
- ❑ İlk elemanın indisi her zaman **0 (sıfır)** olarak kabul edilir.

İNDEKS KULLANIMI

- ❑ Şehirler isimli listenin ilk elemanı olan “Ankara” dır.
- ❑ İndeksi ise sıfırdır.
- ❑ Aşağıdaki tabloda indeksleri ve değerleri bir arada görebilirsiniz.

İndeksi	0	1	2	3	4
Değeri	Ankara	Bursa	Çanakkale	Denizli	Eskişehir

İNDEKS KULLANIMI

- ❑ **Örnek:** İndeksi 2 olan elemanı ekrana yazdırınız.

```
sehirler=["Ankara", "Bursa", "Çanakkale", "Denizli", "Eskişehir"]  
print(sehirler[2])
```

Çanakkale

İNDEKS KULLANIMI

- ❑ İndeksler negatif olarak da yazılabilir.
- ❑ Örneğin; -1 indeksi sondaki elemanı gösterirken -2 indeksi sondan bir öncekini gösterir.
- ❑ **Örnek:**

```
sehirler=["Ankara", "Bursa", "Çanakkale", "Denizli", "Eskişehir"]  
print(sehirler[-2])
```

Denizli

İNDEKS KULLANIMI

- ❑ Listelerde indekslerle birlikte **iki nokta (:)** operatörü kullanılarak istenilen elemanlara ulaşılabilir.
- ❑ Bu işlem için **liste[başlangıç indeksi:bitiş indeksi]** yapısı kullanılır.
- ❑ **Örnek:**

```
asal_sayilar = [2, 3, 5, 7, 11, 13, 17, 19, 23]  
print(asal_sayilar[1:4])
```

```
[3, 5, 7]
```

- ❑ Bu örnekte indeksi 1 olan elemandan başlayarak indeksi 4 olan elemana (4 dâhil değil) kadar ekrana yazdırır.

İNDEKS KULLANIMI

❑ Örnek:

```
asal_sayilar = [2, 3, 5, 7, 11, 13, 17, 19, 23]  
print(asal_sayilar[5:])
```

```
[13, 17, 19, 23]
```

- ❑ Buradaki kullanımda dikkat edilirse **başlangıç olarak** 5 indeksi verilip bitiş indeksi ise verilmemiştir.
- ❑ Bu kullanımda indeksi 5 olan elemandan başlayarak son elemana kadar yazılır.

İNDEKS KULLANIMI

❑ Örnek:

```
asal_sayilar = [2, 3, 5, 7, 11, 13, 17, 19, 23]  
print(asal_sayilar[:5])  
  
[2, 3, 5, 7, 11]
```

- ❑ Bu kullanımda da başlangıç indeksi verilmemiş **bitiş indeksi** olarak 5 verilmiştir.
- ❑ Başlangıç indeksinin verilmediği durumda indeksi 0 (sıfır) olan elemandan başlayarak yazdırılır.

İNDEKS KULLANIMI

❑ Örnek:

```
asal_sayilar = [2, 3, 5, 7, 11, 13, 17, 19, 23]  
print(asal_sayilar[0:6:2])
```

```
[2, 5, 11]
```

- ❑ Bu kullanımda ise sırasıyla **başlangıç indeksi**, **bitiş indeksi** ve **atlama değeri** verilmiştir.
- ❑ Yani **0.** indeksten başlayarak **6.** indekse kadar **ikişer artarak** ekrana yazdırılır.