

1. Grup Tekrar Notu

1) Python Veri Tipleri

- a) Tamsayı(Integer): Pozitif ve negatif tüm sayıları kapsar (1, 100, -1000, 400 ...)
- b) Ondalıklı Sayı(Float): Ondalıklı sayıları kapsar. (3.14, 5.22 ...)
- c) Karakter Dizileri(String): Gerçek hayatta kullandığımız yazıların aynısıdır.

NOT: Python'da değişkenler kendilerine verilen değişkenin tipinin otomatik olarak almaktadır.

2) Tip Dönüşümleri:

- a) Tam Sayıya Çevirme: Tam sayıya çevirmek için int() fonksiyonunu kullanıyoruz.
 - ◆ print(int("55")) #Çıktı: 55
 - ◆ print(int("06")) #Çıktı: 6
 - ◆ print(int(5.5)) #Çıktı: 5 ondalıklı sayıyı tam sayıya çevirme.
 - ◆ print(int("test")) #Çıktı: hata
- b) Tamsayıdan Ondalıklı Sayıya Çevirme:
 - ◆ Print(float(5)) #Çıktı: 5.0
- c) Sayıları String'e Çevirme:
 - ◆ Print(str(1923)) #Çıktı: '1923'

3) Yorum Satırları

- a) Tek Satırlık Yorum Satırı: Yazdığımız yazının önüne “#” karakteri koyarak kullanılır.
ÖR: #Tek satır yorum satırım
- b) Çok Satırlı Yorum Satırı:


```
"""
Yorum satırı 1
Yorum Satırı 2
"""
```

4) Matematik Operatörleri

- a) Dört İşlem Operatörleri (+, -, *, /):
- b) Tamsayı Bölme (//):
Ör: print(17//5) #Çıktı: 3
- c) Üs Bulma (**) :
Print(2**3) #Çıktı: 8
- d) Kalan(Mod) Bulma(%):
print(17%5) #Çıktı: 2
- e) Operatörleri Birlikte Kullanırken İşlem Önceliği
 - ◆ Parantez içi henzaman önce yapılır.
 - ◆ Çarpma ve bölme işlemleri toplama ve çıkarma işleminden önce yapılır.
 - ◆ İşlem öncelikleri eşitse işlemler soldan sağa doğru yapılır.
ÖR: 5 + 5*7/(8-3) #Çıktı: 12.0

5) String Oluşturma:

- a) Tek Tırnak İle : 'Burası Btk Burası Zirve'
- b) Çift Tırnak İle : "Burası Btk Burası Zirve"
- c) Üç Tırnak İle : Birden fazla satırdan oluşan bir metin yazdırılmak istendiğinde kullanılır.

```
print("""
Burası Btk Burası Zirve
Btk Python Kursu
""")
```

6) String Birleştirme: İki string ifadeyi birleştirmek için “+” operatörünü kullanıyoruz.

ÖR: print("Btk" + " " + "Akademi") # Btk Akademi

7) Bir string ifadenin karakter sayısını bulma: Bu işlem için len fonksiyonu kullanılıyor.

ÖR: print(len("Hey Gidi Dünya Hey")) # 18

8) String ifadeler üzerinde çarpma işlemi:

ÖR: print("Btk"*4) #Çıktı: BtkBtkBtkBtk

9) String Parçalama ve Indexleme

- a) Belirli Bir String’in Karakterlerine Ulaşma: Stringler bir karakter dizisidir ve her karakterin bir index numarası vardır. Bu index numarası 0 dan başlamaktadır.

Örneğin kurum adında bir değişken oluşturup içerisine “BTK” atayalım.

```
print(kurum[0]) #Çıktı: B
print(kurum[1]) #Çıktı: T
print(kurum[2]) #Çıktı: K
print(kurum[-1]) #Çıktı: K
print(kurum[-2]) #Çıktı: T
print(kurum[-3]) #Çıktı: B
```

- b) Belirli Bir Stringin Belirli Bir Kısımına Ulaşma

değişken_adı[başlangıç index değeri : bitiş index değeri : atlama değeri]

NOT: Bu tanımlamada başlangıç index’inden başlar ve bitiş index’ine kadar alır yani bitiş indexindeki karakteri almaz.

Örnekler:

```
site="Seçmek özgürlüktür. güvenlinet.org"
print(site[0:7]) #Çıktı: Seçmek
print(site[20:]) #Çıktı: güvenlinet.org
print(site[::-1]) #Çıktı: gro.tenilnevüg .rütkülügöz kemçeS
print(site[7:18]) #Çıktı: özgürlüktür
```

10) Print Fonksiyonu Ve Formatlama: Print fonksiyonu ekrana değer basmak için kullandığımız bir fonksiyondur.

a) Genel Kullanımı:

ÖR:

```
print("Burası Btk Burası Zirve")
```

b) Aynı Satırda Birden Fazla Değer Bastırma: Print fonksiyonu ile birden fazla değer aralarına virgül konularak yazdırılırken yazdırılan değerler arasına default olarak boşluk karakteri eklenir ve bu karakteri sep parametresi ile değiştirebilir.

ÖR:

◆ `print("Burası","Btk","Burası","Zirve")` #Çıktı:Burası Btk Burası Zirve

◆ `print("Burası","Btk","Burası","Zirve", sep="-")`

#Çıktı:Burası-Btk-Burası-Zirve

◆ `print("Burası","Btk","Burası","Zirve",sep = "\n")`

#Çıktı:

Burası

Btk

Burası

Zirve

◆ `print("Burası","Btk","Burası","Zirve",sep = "\t")` #"\t" yazdırılan değişkenlerin #arasına bir tab boşluk bırakır.

Çıktı:Burası Btk Burası Zirve

c) Format Fonksiyonu ile Formatlama İşlemi: Format fonksiyonunu bir string'in istediğimiz bir yerine int, float veya string tiplerinden bir değer yerleştirmek istediğimiz zaman kullanırız.

◆ Ör:

```
mesaj = "Btk'nın {}. yılı kutlu olsun.".format(18)
```

#Süslü parantezin olduğu yere format fonksiyonu içerisine yerleştirdiğimiz

#parametreyi ekler.

```
print(mesaj) #Çıktı:Btk'nın 18. yılı kutlu olsun.
```

◆ Ör:

```
a=3
```

```
b=4
```

```
print("{} + {} 'nin toplamı {} 'dir".format(a,b,a+b))
```

```
# Çıktı:3 + 4 'nin toplamı 7 'dir
```

◆ Ör:

```
mesaj = "{1} ile Programlama Sanatı - {0}".format("BTK", "Python")
```

#Süslü parantezlerin içerisindeki sayılar format fonksiyonu içerisindeki

#parametrelerin string içerisinde yerini göstermek için kullanılır.

```
print(mesaj)
```

```
# Çıktı:Python ile Programlama Sanatı - BTK
```

11) Listeler: Listeler içinde farklı türden veriler barındıran taşıyıcılarımızdır. Listelerde her bir eleman bir indis(index) numarasına sahiptir ve her listenin başlangıç indisi 0 (sıfır) dır.

a) Listeler ile değişik tipte veriler saklayabiliyoruz.

```
ÖR: liste = [3,4,5,6,"Elma",3.14,5.324]
```

b) Boş liste oluşturma

```
bos_liste = []
```

```
bos_liste = list()
```

c) Bir string'in karakterlerini bir diziye aktarmak

```
mesaj = "Merhaba"
```

```
liste = list(mesaj)
```

```
print(liste) # Çıktı: ['M', 'e', 'r', 'h', 'a', 'b', 'a']
```

d) Index numarası ile listenin elemanlarına erişme:

```
# 0. eleman
```

```
liste = [1,2,3,4,5,6,7,8,9,10]
```

```
print(liste[2]) # Çıktı: 3
```

```
#Sonuncu Eleman
```

```
liste = [1,2,3,4,5,6,7,8,9,10]
```

```
print(liste[len(liste)-1])# Çıktı: 10
```

```
# Baştan 4. indeks e kadar (4 dahil değil)
```

```
liste = [1,2,3,4,5,6,7,8,9,10]
```

```
Print(liste[:4]) veya Print(liste[0:4]) #Çıktı: [1, 2, 3, 4]
```

```
#2.indeksten 5.indekse kadar
```

```
liste = [1,2,3,4,5,6,7,8,9,10]
```

```
print(liste[2:5]) # Çıktı: [3, 4, 5]
```

e) İki listeyi toplama(+) oparörörü ile birleştirmek

```
liste1 = [1,2,3,4,5]
```

```
liste2 = [6,7,8,9,10]
```

```
print(liste1 + liste2)
```

```
# Çıktı: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

f) Listenin sonuna "btk" 'yı ekliyoruz.

```
liste = [1,2,3,4]
```

```
liste = liste + ["BTK"]
```

```
print(liste)
```

```
# Çıktı: [1, 2, 3, 4, 'BTK']
```

g) Listenin Birden Fazla Elemanını Birlikte Değiştirme İşlemi:

```
liste = [1,2,3,4,5,6,7,8,9]
```

```
liste[0:3] = [10,20,30]
```

```
print(liste)
```

```
# Çıktı: [10, 20, 30, 4, 5, 6, 7, 8, 9]
```

h) Bir listeyi bir sayıyla çarpma işlemi:

```
liste = [1,2,3]
liste *=3 #liste = liste *3
print(liste)
#Çıktı: [1, 2, 3, 1, 2, 3, 1, 2, 3]
```

i) Append metodu ile listenin sonuna yeni bir eleman ekleme işlemi:

```
meyveListesi = ["Elma", "Armut", "Portakal"]
meyveListesi.append("Muz")
print(meyveListesi)
#Çıktı: ['Elma', 'Armut', 'Portakal', 'Muz']
```

j) Pop metodu ile dizinin son karakterini silme:

```
liste = [1,2,3,4,5]
liste.pop()
print(liste)
#Çıktı: [1, 2, 3, 4]
```

k) Pop metodu ile index numarasını kullanarak kayıt silme:

Kullanımı: liste_adi.pop(index_numarasi)

ÖR:

```
liste = [1,2,3,4,5]
liste.pop(2)
print(liste)
#Çıktı: [1, 2, 4, 5]
```

l) Sort metodu ile sıralama işlemi:

◆ Küçükten büyüğe sıralama:

ÖR:

```
liste = [34,1,56,334,23,2,3,19]
liste.sort()
print(liste)#Çıktı: [1, 2, 3, 19, 23, 34, 56, 334]
```

◆ Büyükten küçüğe sıralama:

ÖR:

```
liste = [34,1,56,334,23,2,3,19]
liste.sort(reverse=True)
print(liste)
#Çıktı: [334, 56, 34, 23, 19, 3, 2, 1]
```

◆ Alfabetik olarak küçükten büyüğe sıralama işlemi:

ÖR:

```
liste = ["Elma","Armut","Muz","Kiraz"]
liste.sort()
print(liste)
#Çıktı: ['Armut', 'Elma', 'Kiraz', 'Muz']
```

◆ Alfabetik olarak büyükten küçüğe sıralama işlemi:

ÖR:

```
liste = ["Elma","Armut","Muz","Kiraz"]
liste.sort(reverse=True)
print(liste)
#Çıktı: ['Muz', 'Kiraz', 'Elma', 'Armut']
```

◆ İç içe listeler

ÖR:

```
sayilar = [1,2,3]
meyveler = ["Elma", "Armut", "Muz"]
hayvanlar = ["Köpek","Kedi","Fare"]
yeniListe = [sayilar,meyveler,hayvanlar]
print(yeniListe)
print(yeniListe[0][2])#Çıktı:3
print(yeniListe[1][1])#Çıktı:Armut
print(yeniListe[2][2])#Çıktı:Fare
```

◆ index metodu ile arama yapma: Aranılan dizi elemanının index numarasını verir.

ÖR:

```
hayvanlarim = ["Köpek","Kedi","Fare"]
bulunanIndex = hayvanlarim.index("Fare")
print(bulunanIndex)# Çıktı:2
```