# PYTHON MATEMATİKSEL İŞLEMLER

Python’da kütüphaneler olmadan da matematiksel işlemler yapabiliriz. İlerleyen günlerde daha ileri seviye işlemler yapacağız ve kütüphaneler (Pandas,NumPy) konusuna değineceğiz.

## Python’da Toplama İşlemi (addition)

“+” operatörü ile toplama işlemi yapabiliriz.

ÖRNEK:

print(6+5)

ÇIKTI:

11

## Python’da Çıkarma İşlemi (subtraction)

“-” operatörü ile çıkarma işlemi yapabiliriz.

ÖRNEK:

print(5-2)

ÇIKTI:

3

## Python’da Çarpma İşlemi(multiplatication)

“\*” operatörü ile çarçma işlemi yapabiliriz.

ÖRNEK:

print(5\*7)

ÇIKTI:

35

## Python’da Bölme İşlemi(division)

“/” operatörü ile bölme işlemi yapabiliriz.

ÖRNEK:

print(9/3)

ÇIKTI:

3

Yukarıda yaptıklarımızı farklı bir şekilde değişken tanımlayarak da gerçekleştirebiliriz.

ÖRNEK:

x=20

y=80

carpma=(x\*y)

print(carpma)

ÇIKTI:

1600

ÖRNEK:

gelir=87000

gider=5674

net=(gelir-gider)

print(net)

ÇIKTI:

81326

• Bölme işlemini düşünürsek “Bölen, Bölünen, Kalan, Bölüm” kısımlarından oluşur. Python’da hangi operatörler ile bölüm ve kalana ulaştığımızı aşağıda görebilirsiniz.

### **% OPERATÖRÜ (modulus)**

Bölme işlemindeki kalan bulma için kullanılan bir operatördür ve dört işleme dahil olarak ele alınır. % operatörü birinci sayıyı ikinci sayıya böler ve ikinci sayıdan kalanı yazar.

ÖRNEK:

print(3%2)

ÇIKTI:

1

### **// OPERATÖRÜ (Floor Division Operator)**

Bölme işleminde birinci sayıyı ikinci sayıya böler ve kalan sayıyı yazar.

ÖRNEK:

print(9//2)

ÇIKTI:

4

## Python ÜSLÜ İFADELER(EXPONENTİAL)

### **• “\*\*” operatörü ile üslü işlemi yapabiliriz.**

Üs alma işlemini basitçe yapan komuttur.

ÖRNEK:

print(6\*\*2)

36

### **• Pow() komutu**

Üstel değerin hesaplanmasına izin veren bir komuttur. Fonksiyonun genel hali;

pow(taban,üs)

ÖRNEK:

taban=8

üs=2

print("üstel değer:",pow(taban,üs))

ÇIKTI:

64

# PYTHON VERİ TÜRLERİ (DATA TYPES)

Programı kullanırken veri tipleri önemlidir. Değişkene atadığımız veri farklı veri türlerinde depolanabilir. Farklı veri tiplerinde tanımlanan değişkenler farklı işler yapabilir. Genel olarak veri türlerini aşağıdaki tabloda görmekteyiz.

Metin Veri Türü (String) : Str

Sayısal Veri Türü (Numeric) : İnt,float,complex

Sıra Veri Türleri (Sequence) : list,tuple

Haritalama Veri Türü (Mapping) : Dict

## SAYISAL VERİ TÜRLERİ

•int() float() complex() komutları ile sayısal veriyi tanımlar.

int(): tamsayı sayısal veri tipi

float():ondalık sayısal veri tipi

complex():karmaşık sayısal veri tipi

ÖRNEK:

x=78 # int tanımlanır.

y=27.85 # float tanımlanır.

z=2+3j # complex tanımlanır.

## METİN VERİ TÜRÜ

Metin halindeki veri tipleri burada görebiliriz.

ÖRNEK:

first\_name='büşra'

## SIRA VERİ TİPİ

Üç tip sıra veri tipini ele alırsak farklı gösterimler ile adlandırıldıklarını görmekteyiz

ÖRNEK:

tek\_sayılar=[1,3,5,7,9] # list tanımlanır.

paket\_ağırlık={9.2, 8.7, 8.14} # set tanımlanır.

paket\_fiyat=(10.2,11.5,12.7) # tuple yanımlanır.

## HARİTALAMA VERİ TÜRÜ

ÖRNEK:

x={'AD':'Büşra'}

print(x)

# VERİ TÜRÜNÜ ALMA

typ() komutu ile verinin hangi tip olduğunu anlayabiliriz.

## GENEL GÖSTERİMİ

print(type('istenien veri'))

### GENEL GÖSTERİME ÖRNEK

A=157

B=57.8777712

C=5+9J

D='BÜŞRA'

E=[1,3,5,7,9]

F={9.2, 8.7, 8.14}

G=(10.2,11.5,12.7)

H={'ŞEHİR':'İSTANBUL'}

Değişkenler tanımlandıktan sonra hangi veri türüne ait olduklarını aşağıdaki komut yardımıyla öğrenelim.

print(type(A))

print(type(B))

print(type(C))

print(type(D))

print(type(E))

print(type(F))

print(type(G))

print(type(H))

ÇIKTI:

<class 'int'>

<class 'float'>

<class 'complex'>

<class 'str'>

<class 'list'>

<class 'set'>

<class 'dict'>