***Pythonda Dışardan Veri Girme Ve Matematiksel Operatörler***

|  |
| --- |
| **ÖMER ÖZYER ANADOLU LİSESİ****2023-2024 EĞİTİM ÖĞRETİM YILI** **BİLİŞİM TEKNOLOJİLERİ** **ÇALIŞMA KAĞIDI** |

x = int(input(“1.sayı: “))

y = int(input(“2.sayı: “))

Burada int komutu integer yani sayısal veri girişini sağlar. input komutu ise dışardan veri girişini sağlar. Kullanıcıdan aldığımız 2 sayıyı x, y değişkenleri içerisine attıktan sonra 2 sayıyı aşağıdaki gibi toplarsak,

toplam = x + y

print(toplam)

Veya Ortalama hesaplayacaksak;

Ortalama=(x+y)/2 şeklinde değişken tanımdalıktan sonra ekranda yazdırabiliriz.

***% (Mod) Komutu (Kalanı bulma)***

Bir Sayının herhangi bir sayıya bölümünden kalanı bulmak için kullandığımız koddur.

Örneğin x adında bir değişken girdik ve 2’ye bölümünden kalanı bulmak için aşağıdaki kodu yazabiliriz;

X%2 x değişkenin 2’ye bölümünden kalanı bulmamızı sağlar.

Pythonda Koşul İfadeleri

Pythonda koşul durumlar ile bir koşuldan elde ettiğimiz True(doğru) ya da False(yanlış) değere göre programın farklı yönlerden ilerlemesini sağlayabiliriz.

Örneğin; kullanıcıdan aldığımız bir parolanın ya da email adresinin doğru olup olmadığını kontrol ederek 'uygulamaya giriş yapıldı' ya da 'parolanız yanlış' şeklinde mesajlar verebiliriz.

Pythonda If Bloğu

Bir koşulun True ya da False olan sonucuna göre farklı kod blokları oluşturmak için If komutunu kullanırız.

Aşağıda bazı koşul ifadeleri verilmiştir. Bu koşullar bize True ya da False değer üretirler.

Eşit mi : a == b,

Eşit değil mi: a != b

Küçük mü : a < b

Küçük ya da eşit mi: a <= b

Büyük mü: a > b

Büyük ya da eşit mi: a >= b

### **Örnek**

numara = **int**(input(“sayı: '))

**if** numara> 0:

 print(“sayı pozitif”)

elif numara < 0: # burda elif komutu başka bir şart koymamızı sağlıyor.

 print(“sayı negatif”)

**else**: # else komutu ise başka bir ihtimal kalmadığında kullanılmaktadır.

 print(“sayı sıfır”)

 Örnek : Girilen bir sayının pozitif sayı olup olmadığını kontrol eden python uygulaması.

sayi = **int**(input('sayı: '))

**if** sayi > 0:

 print(“girilen sayı pozitif sayıdır.”)

**elif sayı< 0**:

 print(“girilen sayı negatif sayı.”)

else:

 print(“Girilen Sayı sıfırdır.”)

# **Python List**

Python Programlama Dilinde 4 tür dize (array) bulunur. Bunlar;

**List**, sıralanabilir ve değiştirilebilir dizeler. Bu tarz dizelere aynı üyeler tekrar eklenebilir.

* **Tuple**, sıralanabilir ancak değiştirilemez dizelerdir. Aynı üyeler birden fazla eklenebilir.
* **Set** sıralanamaz ve aynı üyeleri kabul etmez.
* **Dictionary** sıralanamaz ancak değiştirilebilir ve listelenebilir. Aynı üyeler eklenemez.

Bir dize kullanırken dizelerin özelliğini bilmek ve ona göre kullanmak hem zaman kazandırır, hem işimizi kolaylaştırır.
**List**

Sıralanabilir ve değiştirilebilir veriler topluluğudur. Nesneler köşeli parantezler arasına yazılır.

Kendimiz istediğimiz isimde istediğimiz listeyi oluşturabiliriz. Mesela şehirlerle ilgili aşağıdaki gibi bir liste oluşturalım.

şehirler=[“ankara”,”İstanbul”,”izmir”,”muğla”] # 4 elemanlı bir liste oluşturduk.

print(şehirler) # bu komut şehirler listemizi ekrana yazdırmamızı sağlar.

Yukarıdaki liste elemanlarımızın sıralaması sıfırdan başlayarak devam eder. Muğla ili listemizde 3. Sırada bulunan elmandır.

## Dizeye Değer Eklemek

Dizeye yeni bir değer eklemek için **append()** fonksiyonundan faydalanabiliriz. Değer, dizenin en sonuna eklenecektir.

şehirler=[“ankara”,”İstanbul”,”izmir”,”muğla”]

şehirler.append(“yozgat”)

print(şehirler) # burada listemizde Yozgat ilimizi de eklemiş olduk.

## Dizeden Bir Değeri Silmek

Bunun için **remove()** fonksiyonu kullanılır. Parantez içinde silinecek değer belirtilir:

şehirler=[“ankara”,”İstanbul”,”izmir”,”muğla”]

şehirler.remove("izmir")

print(şehirler)

Eğer dizenin belli bir sırasındaki nesneyi sileceksek **pop()** fonksiyonundan faydalanabiliriz. Parantez içinde sıra numarası belirtebiliriz. Eğer hiçbir şey belirtmezsek dizenin en sonundaki nesne silinir.

şehirler=[“ankara”,”İstanbul”,”izmir”,”muğla”]

şehirler.pop(0)

print(şehirler)

## Bir Dizenin Kaç Elemanlı Olduğunu Bulmak

Dizelerin eleman sayısı yani boyutu/genişliği **len()** fonksiyonu ile tespit edilebilir:

şehirler=[“ankara”,”İstanbul”,”izmir”,”muğla”]

print(len(şehirler))

## Pythonda For Döngüleri

Python for döngülerini bir eleman grubundaki (list, tuple, dictionary, set ya da string) her bir elemana ulaşmak için kullanırız.

sayilar = [1,2,3,4,5,-9,-4]

**for** sayi **in** sayilar:

 print(sayi)

Örneğin bu kod liste elemanlarımızın hepsini sırayla ekrana yazdırır. Sadece belirli şartı sağlayan elemanları yazdırmak istiyorsak o zaman for döngüsü içine if bloğu açmamız gerekir.

sayilar = [1,2,3,4,5,-9,-4]

**for** sayi **in** sayilar:

 if sayi<0:

 print(sayı) Bu komutlarda sıfırdan küçük olan sayıları yazdırmak için for döngüsü içine if bloğu açtık. Eğer sayı sıfırdan küçükse o zaman yazdır komutu ekledik.