



ÖĞRENME BİRİMİ 5

KARAR VE DÖNGÜ YAPILARI

Neler Öğreneceksiniz?

Bu öğrenme birimi ile;

- Karar yapısı kullanımlarını öğrenebilecek,
- If ve if-elif yapılarını kullanabilecek,
- Döngü mantığını anlayabilecek,
- Döngü türlerini kullanabileceksiniz.

Anahtar Kelimeler:

Karar yapısı, döngüler.



Hazırlık Çalışmaları

1. Karar yapılarını araştırarak günlük hayatta nasıl kullanıldığını tartışınız.
2. Programlama dillerinde kullanılan döngülerin sağladığı kolaylıkları araştırınız.

5. KARAR VE DÖNGÜ YAPILARI

5.1. Karar Yapıları

Günlük hayatta sık sık karar vermeyi gerektiren durumlarla karşılaşmaktadır. Programlamada da benzer olarak karar yapıları kullanılmaktadır. Örneğin teneffüste çay ya da kahve arasında bir seçim yapma karar verme sürecidir. Karar verme sürecinde eldeki verilerle bir değerlendirme yapılmaktadır. Bir önceki teneffüste çay içilmesi bu teneffüsteki kararı etkiler ve belki de kahve tercihinin daha cazip hâle getirir. Bu gibi örnekleri çoğaltmak mümkündür.

5.1.1. If-Else Yapısı

Python programlama dilinde (ve birçok diğer dilde) karar yapıları if (eğer) ile temsil edilmektedir. Bu yapıda bir durumun doğru (true) ya da yanlış (false) olma durumuna göre bazı eylemler icra edilmektedir. if yapısı tek başına kullanıldığı gibi else ile birlikte de kullanılabilir. Else anahtar sözcüğü tek başına kullanılmaz. Özetle; if “**eğer**” olarak, else ise “**değilse**” olarak düşünülebilir. Kullanıcının girdiği yaş 18 ve daha büyükse ekrana “**ehliyet alabilir**”; değilse ekrana “**ehliyet alamaz**” gibi uyarılar vermek bu yapının bir örneğidir. Bu örnek, programlama dilinde şu şekilde yazılır:

```
yas=int(input("Yaşınızı girin: "))
if yas>= 18:
    print("Ehliyet alabilirsiniz")
else:
    print("Ehliyet alamazsınız")
```

Örnekte int veri tipinde yas isimli bir değişken tanımlandı. input() fonksiyonu ile kullanıcıdan veri alındı. if satırında ise yas>=18 şartı sorgulandı. Bu şart doğru ise “**Ehliyet alabilirsiniz**” yanlış ise “**Ehliyet alamazsınız**” uyarılarının ekran çıktısı olması sağlandı. Burada else, doğru değilse anlamında kullanıldı. Ayrıca kod yapısı incelendiğinde farklı bir girinti yapısı da görülmektedir. Python programlama dilinde girinti yapısı örnekteki gibidir. if ve else satırının sonunda : (iki nokta) kullanıldığına dikkat edilmelidir. print satırlarında ise satırın içten başladığı görülmektedir. Klavyede bulunan **tab** tuşu ile bu girinti ayarlanabilir.

Örnek 1 : Kullanıcının girdiği sayı çift ise “Çift sayı”; değilse “Tek sayı” uyarılarını veren kodu yazınız.

```
girilen_sayi=int(input("Bir sayı girin: "))
if girilen_sayi%2==0:
    print("Çift sayı")
else:
    print("Tek sayı")
```

Bu örnekte kullanıcıdan bir sayı alınmıştır. Bir sayının çift olması, o sayının ikiye kalansız bölünmesi ile açıklanır. Bu nedenle bir mod alma işlemi yapılmıştır. girilen_sayi%2==0 şartı sayının 2'ye bölündüğünde kalanın 0 (sıfır) olması anlamına gelmektedir.

Önemli Not: Bu örnekteki == karşılaştırma operatörüdür. Atama operatörü ile karıştırılmaması gerekir.

Sıra Sizde:

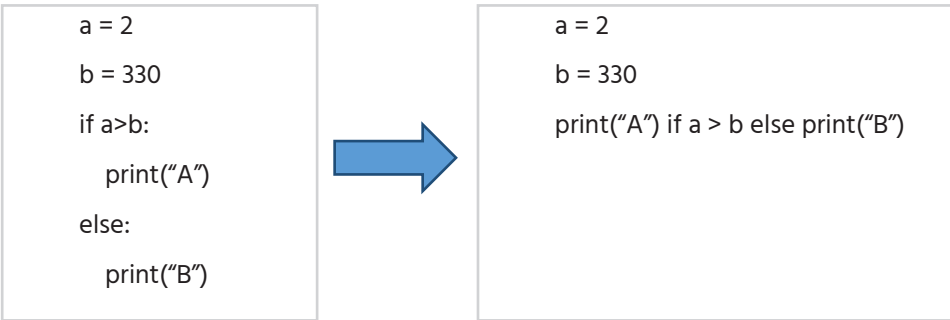
- a) Kullanıcıdan iki sınav ve bir performans notu girmesini isteyiniz. Girilen 3 notun ortalaması 50 ve daha büyükse "Başarılı"; değilse "Başarısız" çıktıları veren kodu yazınız.
- b) Bir üçgenin iç açıları toplamı 180 derecedir. Kullanıcının girdiği üç açı değerine göre "Bu bir üçgendir." ya da "Bu bir üçgen değildir." çıktıları veren kodu yazınız.
- c) Bir hava yolu firması en fazla 20 kilogram bagaj hakkı vermektedir. 20 kilogramdan sonraki her kilogram için 10 TL ek ücret almaktadır. Buna göre bagajı 20 kg ya da daha az olan yolculara "Herhangi bir ücret ödemeniz gerekmiyor."; 20 kg'den fazla olanlar için de ne kadar ek ücret ödeneceğini hesaplayarak "Fazla bagaj için TL ödemelisiniz." çıktıları veren kodu yazınız.

Not: Bu soruda kilogram hesabında sadece tam sayıları dikkate alınız. Örneğin 28,70 kilogram olan bagaj için sadece 8 kg için ek ücret ödenmesi yeterlidir.

- ç) Kullanıcının girdiği iki ürünün toplam fiyatı 200 TL ve altıysa "Ödenecek miktar=.... TL"; 200 TL'yi geçerse %25 indirim yaparak "Ödenecek miktar, indirimden sonra TL'dir." çıktıları veren kodu yazınız.

Önemli Not:

Tek satırlık ifadeler, Python'da tek satırlık kodların if ifadesinin yanına yazılmasını desteklemektedir. Aşağıdaki örnekte solda verilen kod bloğu sağdaki gibi tek satır hâlinde de yazılabilir.

**Örnek 2:**

cikis_birimleri isimli bir liste oluşturarak yazıcı, hoparlör ve ekran elemanları eklensin. If yapısı kullanarak ekran elemanı listede varsa "**Eleman bulundu.**"; yoksa "**Eleman bulunamadı.**" çıktıları veren kodu yazınız.

```
cikis_birimleri=["yazıcı", "hoparlör", "ekran"]
```

```
if "ekran" in cikis_birimleri:
```

```
    print("Eleman bulundu.")
```

```
else:
```

```
    print("Eleman bulunamadı.")
```

Ekran Çıktısı: Eleman bulundu.

Sıra Sizde:

haftaici isimli bir liste oluşturarak Pazartesi, Salı, Çarşamba, Perşembe ve Cuma elemanları eklensin. If yapısı kullanarak Cumartesi elemanı listede varsa "**Listede bulundu.**"; yoksa "**Listede bulunamadı.**" çıktıları veren kodu yazınız.

Örnek 3:

Kullanıcıya yabancı dil ve ofis programlarını bilip bilmediği sorulsun. Her iki soruya da "**Evet**" cevabı verilirse "**İşe alınıyorsunuz.**"; diğer durumlarda ise "**İşe alınmadınız.**" çıktıları veren programı yazınız.

```
yabanci_dil=input("Yabancı dil biliyor musunuz? (Evet/Hayır): ")
```

```
ofis_programlari=input("Ofis programlarını biliyor musunuz? (Evet/Hayır): ")
```

```
if yabanci_dil=="Evet" and ofis_programlari=="Evet":
```

```
    print("İşe alınıyorsunuz.")
```

else:

```
print("İşe alınmadınız.")
```

Bu örnekte iki değişken tanımlanmıştır. Her iki değişkene verilen cevabın da **"Evet"** olması durumunda koşullar sağlanarak **"İşe alındınız."** çıktısı verilmiştir. Burada **and** operatörünün kullanıldığı görülmektedir. Bu operatör tüm şartların doğru olmasını gerektirir. Dikkat edilmesi gereken başka bir nokta da string ifadeler karşılaştırılırken tırnak işaretinin kullanılmasıdır.

Sıra Sizde:

- Kullanıcıdan kullanıcı adı ve şifre girilmesi istensin. Kullanıcı adı "Türkiye"; şifre 1923 ise "Giriş başarılı"; değilse "Kullanıcı adı ya da şifre yanlış" çıktıları veren kodu yazınız.
- Girilen sayı hem 3 hem de 5'e tam bölünüyorsa "15'e tam bölünür."; bölünmüyorsa "15'e tam bölünmez." çıktıları veren kodu yazınız.

Örnek 4 : Bir mülakatta katılımcının başarılı olabilmesi için İngilizce ya da Fransızcadan birini bilmesi ve yaşının 40'tan küçük olması gerekmektedir. Katılımcıya yukarıdaki bilgileri, adını ve soyadını sorarak mülakat sonucunu **"Başarılı"** ya da **"Başarısız"** çıktıları ile gösteriniz.

```
ad_soyad=input("Adınız-Soyadınız: ")
yabanci_dil=input("Bildiğiniz yabancı dil: ")
yas=int(input("Yaşınız: "))
if ((yabanci_dil=="İngilizce" or yabanci_dil=="Fransızca") and yas<40):
    print("Sayın "+ad_soyad+", sonuç başarılı")
else:
    print("Sayın "+ad_soyad+", sonuç başarısız")
```

Ekran Çıktısı:

```
Adınız-Soyadınız: Canan Yılmaz
Bildiğiniz yabancı dil: Fransızca
Yaşınız: 32
Sayın Canan Yılmaz, sonuç başarılı
```

Bu örnekte mantıksal operatörlerden **and** ve **or** bir arada kullanılmıştır. **or** operatöründe şartlardan birinin doğru olması yeterlikken **and** operatöründe tüm şartların doğru olması gerekmektedir. **if** satırına bakıldığında yabanci_dil değişkeninin İngilizce ya da Fransızca olması; ayrıca yas değişkeninde de 40 değerinden küçük olması istenmektedir. **print** satırında ise + operatörü ifadeleri birleştirmek amacıyla kullanılmıştır.

Sıra Sizde:

Bir programın bilgisayara kurulması için i7 işlemci ya da en az 8 GB RAM belleğe ihtiyaç duyulmaktadır. Şartlar sağlanıyorsa **"Kurulum uygun"**; sağlanmıyorsa **"Kurulum uygun değil"** çıktıları veren programı yazınız.

5.1.2. If-Elif-Else Yapısı

Daha önce yapılan karar yapısı örneklerinde eğer-değilse yapısı kullanıldı. Başka bir ifadeyle şart doğru ise bir durum, yanlış ise başka bir durum vardı. Bazen tek bir şartın değil de daha fazla şartın olduğu durumlar da ortaya çıkmaktadır. Bu gibi durumlarda **if-elif-else** yapısı kullanılır. Bu yapıda ilk şart **if**; aradaki şartlar **elif**; değilse kısmında da **else** sıralaması bulunmaktadır. Örneğin bir sayının pozitif olup olmadığının öğrenilmeye çalışıldığı bir durumda if-else yapısı yetersiz kalacaktır. Çünkü sayı 0'dan büyükse **"Pozitif"**, küçükse **"Negatif"** olacağı gibi sayı sıfıra eşit de olabilir. Örnekte bu durumun if-elif-else yapısı ile kodlanması görülmektedir:

```
sayi=int(input("Bir sayı girin: "))
if sayi>0:
```

```

print("Pozitif")
elif sayi<0:
    print("Negatif")
else:
    print("Sayı sıfıra eşittir")

```

Ekran Çıktısı: Bir sayı girin: -5

Negatif

Bu örnekte birden fazla şart olduğu için **if-elif-else** yapısı kullanılmıştır. Bu yapıda ilk şart olan `sayi>0` if satırına; `sayi<0` ise elif satırına yazılmıştır. Örnekte üç durum yaşanabilir. Sayı 0'dan büyük, 0'dan küçük ya da 0'a eşit olabilir. Bu nedenle ilk iki şartı yazdıktan sonra başka bir ihtimal olmayacağı için `sayi==0` yazmak yerine else anahtar sözcüğünü yazmak yeterlidir.

Sıra Sizde: Girilen plaka kodu 06 ise ekrana Ankara; 07 ise Antalya; 08 ise Artvin; bunların dışında girilen tüm değerlerde ise Türkiye çıktısı veren kodu yazınız.

Örnek 5: Girilen iki sayıya ve operatöre (+,-,*,/) göre toplama, çıkarma, çarpma ya da bölme işlemlerini yapan; bu operatörler dışında bir değer girildiğinde **"Yanlış işlem girdiniz."** uyarısı veren kodu yazınız.

```

sayi1=int(input("Birinci sayıyı girin: "))
sayi2=int(input("İkinci sayıyı girin: "))
islem=input("İşlem seçin (+,-,*,/): ")
if islem=="+":
    sonuc=sayi1+sayi2
    print(sonuc)
elif islem=="-":
    sonuc=sayi1-sayi2
    print(sonuc)
elif islem=="*":
    sonuc=sayi1*sayi2
    print(sonuc)
elif islem=="/":
    sonuc=sayi1/sayi2
    print(sonuc)
else:
    print("Yanlış işlem girdiniz")

```

Ekran Çıktısı:

```

Birinci sayıyı girin: 10
İkinci sayıyı girin: 20
İşlem seçin (+,-,*,/): -
-10

```

Örnek 6: Yaşam süresinin artmasından sonra yaş grupları aşağıdaki gibi değerlendirilmeye başlanmıştır. Girilen doğum tarihine göre kişinin yaş grubunu ekrana yazdıran kodu yazınız.

```

0-17 yaş arası: Çocuk
18-65 yaş arası: Genç
66-79 yaş arası: Orta Yaşlı
80 yaş ve üstü: Yaşlı
dogum_tarihi=int(input("Doğum yılınızı girin: "))
yas=2020-dogum_tarihi
if yas>=0 and yas<=17:
    print("Çocuk")
elif yas>=18 and yas<=65:
    print("Genç")
elif yas>=66 and yas<=79:
    print("Orta Yaşlı")
elif yas>=80:
    print("Yaşlı")
else:
    print("Yanlış değer girdiniz")

```

Ekran Çıktısı:

Doğum yılınızı girin: 1965

Genç

Sıra Sizde:

- Kullanıcı tarafından girilen hava sıcaklığı 5 °C ve altındaysa "Soğuk"; 6-14 °C arasındaysa "İlık"; 15 °C ve daha fazlaysa "Sıcak" çıktılarını veren kodu yazınız.
- Bir otoparkın ücret tarifesini aşağıdaki gibidir:
 - 1 saate kadar: 5 TL
 - 1-5 saat arası: Saat başı 4 TL
 - 5 saatten fazla: Saat başı 3 TL
 Buna göre kullanıcının girdiği otoparkta kalınan saat süresine göre ödenecek miktarı bularak ekrana yazdırınız.
- Üçgenler kenar uzunluklarına göre üçe ayrılmaktadır: Eşkenar, İkizkenar ve Çeşitkenar. Kullanıcının girdiği 3 kenar uzunluğuna göre üçgenin türünü ekrana yazdırınız.
- Kullanıcının girdiği boy ve ağırlık değerlerine göre vücut kitle indeksini ($VKI = \text{ağırlık} / (\text{boy} * \text{boy})$), boy metre cinsinden verilmeli) hesaplayınız.
 - VKI 18 ile < 25 aralığındaysa normal,
 - VKI 25 ile <30 aralığındaysa kilolu,
 - VKI 30 ve daha yüksekse obez,
 - VKI 35 ve daha fazlaysa ciddi obez olarak kabul edilir.
 VKİ'ni hesaplayarak kişinin durumunu yazdırınız.
- Kullanıcıdan adını, maaşını ve çalışma yılını girmesini isteyiniz. 0-5 yıl arası çalışanlara %10; 6-10 yıl arası

çalışanlara %15; 11 ve daha fazla yıl çalışanlara %25 zam yapılmaktadır. Buna göre “Sayın, zamlı maaşınız TL” çıktısı veren kodu yazınız.

e) Girilen üç sayıdan en büyüğünü bulan kodu yazınız.

5.1.3. İç İçe İfadeler

Önceki konuda mantıksal operatörleri kullanarak birden fazla durumun kontrolü sağlanmıştır. İç içe ifadeler de birden fazla durumun kontrol edilmesi gerektiğinde kullanılır.

Örnek 7: Bir firma işe alımlarda 40 yaş altı kişileri tercih etmektedir. Bu şartı sağlayan kişilerde de sürücü belgesi olan üniversite mezunlarını tercih etmektedir. Buna göre kullanıcıya önce yaşı sorulsun. Yaşı 40 altı olmayanlara “**Üzgünüz, kriterlerimize uymuyorsunuz.**” uyarısı verilerek programdan çıkılırken; yaş şartı uyanlara diğer iki soruyu sorarak işe alınıp alınmadıklarını çıktı olarak veren kodu yazınız.

```
yas=int(input("Yaşınızı girin: "))
if yas<40:
    mezuniyet=input ("Üniversite mezunu musunuz? (E/H):")
    surucu_belgesi=input ("Sürücü belgeniz var mı? (E/H):")
    if mezuniyet=="E" and surucu_belgesi=="E":
        print("Tebrikler, işe alındınız")
    else:
        print("İşe alınmadınız")
else:
    print("Üzgünüz, kriterlerimize uymuyorsunuz")
elif yas>=80:
    print("Yaşlı")
else:
    print("Yanlış değer girdiniz")
```

Ekran Çıktısı 1 :

Yaşınızı girin: 46
Üzgünüz, kriterlerimize uymuyorsunuz.

Ekran Çıktısı 2 :

Yaşınızı girin: 35
Üniversite mezunu musunuz? (E/H):E
Sürücü belgeniz var mı? (E/H):E
Tebrikler, işe alındınız.

Bu örnekte birden fazla durumun kontrolü **iç içe if yapısı** ile sağlanmıştır. Kullanıcıya sorulan yaş bilgisine göre; yaşı 40 altı olmayanlara herhangi bir soru sormadan programdan çıkılırken diğer adaylara ek sorular sorulmuştur.

Sıra Sizde: Kullanıcıya sinema ya da tiyatro tercihi sorulsun. Sinema izlemek için 15 TL, tiyatro için 10 TL ödenmesi gerekmez. Öğrencilere %50 indirim yapıldığı düşünülerek öğrenci ise indirim yapılan; öğrenci değilse indirimsiz tutarı hesaplayarak ekrana yazdıran kodu yazınız.

5.2. Döngüler

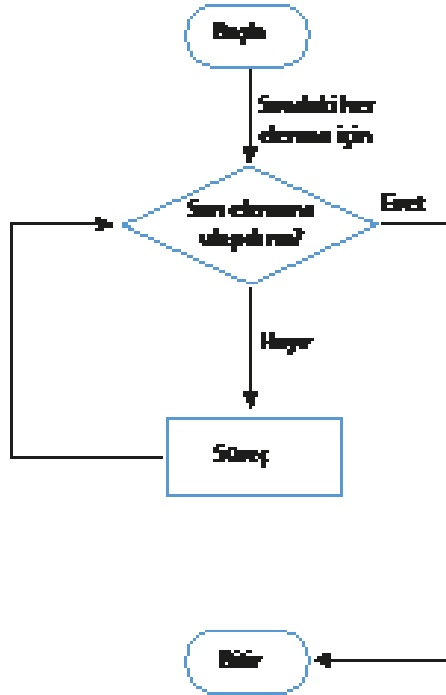
Programlama dillerinde karar yapıları gibi sık kullanılan başka bir yapı da döngülerdir. Program içinde kod bloklarının istenen sayıda tekrar etmesini sağlayan yapılara **döngü** adı verilir. Python programlama dilinde **for** ve **while** döngüleri bulunmaktadır.



Görsel 5.1: Döngüler

5.2.1. For Döngüsü

Şart doğru olduğu sürece işlemlerin tekrarını sağlayan döngü yapısıdır. For döngüsü belirli bir şart sağlanana kadar belirlenen kod bloklarını tekrarlar. For döngüsünün yapısı aşağıdaki gibidir:



Şekil 5.2: For döngüsünün yapısı

5.2.1.1. Range Kullanımı

Döngünün başlangıç ve bitiş değeri belli olan durumlarda kullanılan fonksiyondur. Varsayılan olarak 0'dan (sıfır) başlayarak birer birer artar. Range fonksiyonunun bitiş değeri döngü dışında kabul edilir.

Örnek 8:

```
for sayılar in range(10):
    print(sayılar)
```


Ekran Çıktısı:

0
1
2
3
4
5
6
7
8
9

Bu örnekte bir başlangıç değeri verilmediği için döngü 0'dan (sıfır) başlar ve 10'a kadar devam eder (10 hariç). sayılar ismiyle oluşturulan değişken print fonksiyonu ile ekrana yazdırılmıştır.

```
for sayilar in range(5,10):  
    print(sayilar)
```

Ekran Çıktısı:

5
6
7
8
9

Bu örnekte başlangıç ve bitiş değerleri birlikte verilmiştir. Ekran çıktısına bakıldığında başlangıç değeri olan 5'ten başlayarak bitiş değerine kadar olan sayılar (bitiş değeri dâhil değil) ekrana yazdırılmıştır.

Örnek 9:

```
for sayilar in range(5,20,3):  
    print(sayilar)
```

Ekran Çıktısı:

5
8
11
14
17

Bu örnekte başlangıç ve bitiş değerleri ile artış değeri de verilmiştir. Başka bir ifadeyle döngünün 5'ten başlayarak 20'ye kadar 3'er 3'er artması sağlanmıştır. Burada yine dikkat edilmesi gereken nokta 17 sayısından sonra 20 sayısının son değer olduğu için çıktıda görülmemesidir.

Örnek 10 :

```
for sayilar in range(20,5,-3):  
    print(sayilar)
```

Ekran Çıktısı:

```
20  
17  
14  
11  
8
```

Bu örnekte 20'den başlayarak 5'e kadar (5 dâhil değil) 3'er azalan sırada sayılar yazdırılmıştır.

Sıra Sizde:

- a) 0-20 arası çift sayıları for döngüsü ile ekrana yazdırınız.
- b) 1-30 arası tek sayıları for döngüsü ile ekrana yazdırınız.
- c) 3'ten başlayarak 41'e kadar olan sayıları 5'er arttırarak for döngüsü ile ekrana yazdırınız.
- ç) 50'den 20'ye kadar olan sayıları 3'er azaltarak for döngüsü ile ekrana yazdırınız.

Range fonksiyonu ile else anahtar sözcüğü de kullanılarak döngü sonunda bir mesaj verilebilir.

Kullanımı şu şekildedir:

```
for sayilar in range(10):  
    print(sayilar)  
else:  
    print("Döngü bitti")
```

Ekran Çıktısı:

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
Döngü bitti
```

Önemli Not: Toplama işleminde, 0 etkisiz eleman olduğundan toplam değişkenine başlangıçta 0 (sıfır) atanır. Çarpma işleminde ise 1 etkisiz eleman olduğundan çarpım değişkenine başlangıçta 1 atanır.

Örnek 11 : for döngüsü ile 1'den 10'a kadar olan sayıların toplamını bularak ekrana yazdırınız.

```
toplam=0
for sayilar in range(11):
    toplam=toplam+sayilar
print("Sayıların toplamı= ",toplam)
```

Ekran Çıktısı:

Sayıların toplamı= 55

Bu örnekte toplama 0 ilk değeri atandı. 1-10 arasındaki sayılar toplanacağı için range değeri 11 olarak verildi ve sayilar isimli bir değişken oluşturuldu. Döngü her döndüğünde sayilar değişkeni toplam değişkenine eklendi. Print fonksiyonu ile toplam ekrana yazdırıldı.

Önemli Not: Python programlama dilinin girinti yapısından bahsedilmişti. Yukarıdaki kodun son satırı bir önceki satırla aynı hizada yazılırsa for döngüsü içinde kabul edilir. Bu durumda toplam, döngü her başa döndüğünde adım adım ekrana yazılır. Lütfen deneyerek sonucunu görünüz.

Sıra Sizde:

- Girilen iki sayı arasındaki sayıların toplamını bularak ekrana yazdırınız.
- Girilen iki sayının arasındaki sayıların ortalamasını bularak ekrana yazdırınız.
- Girilen sayının faktöriyelini bularak ekrana yazdırınız.
- Elemanları sırasıyla 4, 12, 18, 33 olan sayılar ile **sayilar** isminde bir liste oluşturunuz. Listenin elemanlarını for döngüsü kullanarak toplayınız ve ekrana yazdırınız.

5. 2. 1. 2. In kullanımı:

in operatörü bir elemanın listede olup olmadığını kontrol eder. For döngüsü ile kullanımı şu şekildedir:

```
meyveler=["çilek", "muz", "şeftali"]
```

```
for meyve in meyveler:
```

```
    print(meyve)
```

Ekran Çıktısı:

```
çilek
muz
şeftali
```

Bu örnekte meyveler listesi içinde meyve isimli bir değişken oluşturularak ekrana yazdırılmıştır.

in operatörü metinsel (string) ifadeleri de harf harf ekrana yazdırabilir. Kullanımı şu şekildedir:

```
for harfler in "Döngü":
```

```
    print(harfler)
```

Ekran Çıktısı:

D
ö
n
g
ü

Sıra Sizde:

- a) Yukarıdaki örnekte ikinci satırı `print(harfler*10)` şeklinde değiştirerek çalıştırınız.
b) Çıktısı aşağıdaki gibi olan kodu yazınız.

```
PPPPPPPPPP
YYYYYYYYYY
TTTTTTTTTT
HHHHHHHHHH
OOOOOOOOOO
NNNNNNNNNN
```

Örnek 12 : 10-20 arası sayılardan oluşan sayılar isimli bir liste oluşturarak liste içinde 3'e tam bölünen sayıları ekrana yazdırınız.

```
sayilar=[10,11,12,13,14,15,16,17,18,19,20]
```

```
for sayi in sayilar:
```

```
    if sayi%3==0:
```

```
        print(sayi)
```

Ekran Çıktısı:

12
15
18

Sıra Sizde:

- a) Yukarıdaki listede bulunan çift sayıları ekrana yazdırınız.
b) Yukarıdaki liste ile `sayilar2=[21,22,23,24,25]` listesini birleştirerek 4'e tam bölünen sayıları ekrana yazdırınız.

Örnek 13 : `alan_adi` isimli, değeri bilişim olan bir değişken tanımlayarak içinde kaç adet "i" harfi olduğunu bulup ekrana yazdırınız.

```
alan_adi="bilişim"
```

```
toplam=0
```

```
for aranan in alan_adi:
```

```
    if aranan=="i":
```

```
        toplam=toplam+1
```

```
print("Bu metinde toplam ",toplam," adet i vardır")
```

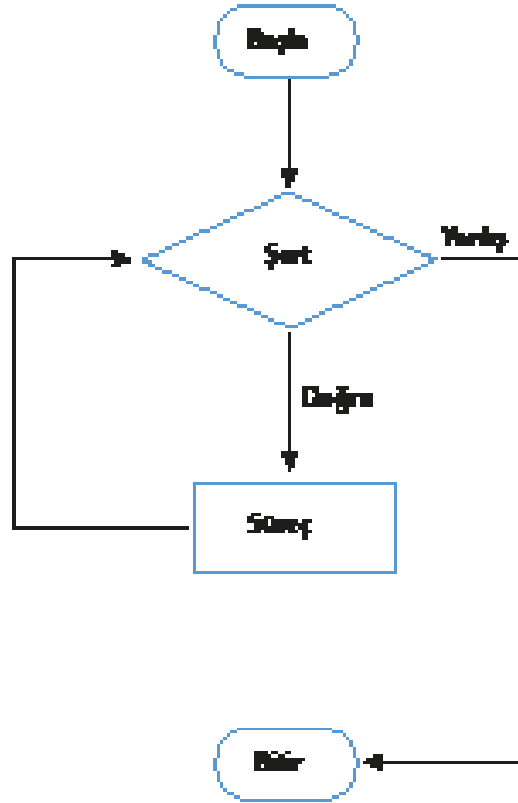
Ekran Çıktısı:

Bu metinde toplam 3 adet "i" vardır.

Sıra Sizde: Kullanıcıdan bir ifade ve aranacak harf girmesini isteyiniz. Girilen ifadede kaç tane "a" harfi olduğunu bularak ekrana yazdırınız.

5.2.2. While Döngüsü

While döngüsü hemen hemen tüm programlama dillerinde bulunmaktadır. Test edilen ifade doğru (true) olduğu sürece kodları tekrarlamaktadır. While yapısı genellikle kod bloğunun kaç kez tekrar edileceğinin bilinmediği durumlarda kullanılmaktadır. While döngüsünün yapısı aşağıdaki gibidir:



Şekil 5.3: While döngüsünün yapısı

Örnek 14 :

```

i=0
while (i<5):
    print("Kodlama")
    i=i+1
  
```

Ekran Çıktısı:

```

Kodlama
Kodlama
Kodlama
Kodlama
Kodlama
  
```

Bu örnekte *i* değişkenine ilk değer olarak 0 (sıfır) atanmıştır. `while (i<5):` satırı ile *i* değeri 5 olana kadar (5 dâhil değil) döngü devam eder. Her adımda bir kez **“Kodlama”** ifadesi ekrana yazdırılır ve *i* değeri 1 arttırılır. *i* değeri sırasıyla 0,1,2,3 ve 4 olur. Yani döngü 5 kez döner ve program sonlanır.

Sıra Sizde: Yukarıdaki kodda `while` satırındaki şartı `i<=5` yaparak çalıştırınız ve çıktıları arasındaki farkı sınıfta tartışınız.

Örnek 15 :

```
i=0
while (i<=20):
    print(i)
    i=i+2
print("Döngü sonu")
```

Ekran Çıktısı:

```
0
2
4
6
8
10
12
14
16
18
20
Döngü sonu
```

Bu örnekte *i* değeri 0'dan başlayarak 20'ye kadar 2'şer artarak (`i=i+2`) ekrana yazdırılmıştır.

Sonsuz döngü: Programlama dillerinde döngü oluştururken yapılacak bir mantık hatası sonsuz döngüye neden olabilir. Sonsuz döngüde program sürekli çalışacaktır. Sonsuz döngüden çıkmak için **Ctrl+C** tuş kombinasyonu kullanılabilir.

Örnek 16 :

```
i=15
while (i<20):
    print(i)
    i=i-1
```

Bu örnekte *i* değişkeni 15'ten başlayarak 1'er azalır (`i=i-1`). `while (i<20)` şartına göre *i* sürekli azalarak devam ettiği için sonsuza doğru gitmektedir ve bu döngü hiçbir zaman bitmeyecektir.

Örnek 17 :

```
while True:
    print("Sonsuz döngüye girildi")
```

Bu örnekte de while True satırı aksi belirtilmediği sürece devam edilmesi anlamı taşır ve bu nedenle kullanıcı döngüden çıkana kadar print satırında bulunan ifadeyi yazar.

Sıra Sizde:

- 1-30 (30 dâhil) arasındaki tek sayıları while döngüsü ile ekrana yazdırınız.
- 60-30 (30 dâhil değil) arasındaki çift sayıları azalan sırada while döngüsü ile ekrana yazdırınız.
- 0-100 (100 dâhil) arasındaki sayılardan 5'e tam bölünenleri while döngüsü ile ekrana yazdırınız.
- Ekran çıktısı aşağıdaki gibi olan kodu while döngüsü ile yazınız.

1 . sınıf
2 . sınıf
3 . sınıf
4 . sınıf
5 . sınıf
6 . sınıf
7 . sınıf
8 . sınıf
9 . sınıf
10 . sınıf
11 . sınıf
12 . sınıf

Örnek 18 :

```
i=1
sonuc=1
faktoriyel=int(input("Faktoriyeli hesaplanacak sayıyı giriniz: "))
while (i<=faktoriyel):
    sonuc=i*sonuc
    i=i+1
print("Sonuc=",sonuc)
```

Ekran Çıktısı:

Faktoriyeli hesaplanacak sayıyı giriniz: 5
Sonuc= 120

Bu örnekte sonuç değişkenine ilk değer olarak 1 atanmıştır. Faktoriyel hesaplama bir çarpma işlemi olduğundan, çarpmadaki etkisiz eleman olan 1 verilmiştir. Faktoriyel hesaplama 1 sayısından başlayacağı için i değişkenine de ilk değer olarak 1 atandı. Faktoriyel kendinden önceki sayıların çarpımı olduğundan sonuc=i*sonuc ifadesi eklendi.

Sıra Sizde:

- 1 ile 20 arasındaki (20 dâhil) sayıların toplamını bulan programı while döngüsü ile yazınız.
- Girilen iki sayı arasındaki sayıları toplayan programı while döngüsü ile yazınız.
- Girilen iki sayı arasındaki sayıların ortalamasını bulan programı while döngüsü ile yazınız.
- 20 ile 50 arasındaki (50 dâhil) çift sayıların toplamını bulan programı while döngüsü ile yazınız.

Örnek 19 : Girilen sayı 0 (sıfır) olana kadar girilen tüm sayıları toplayan ve ekranda gösteren programı yazınız.

```
toplam=0
sayi=1
while (sayi!=0):
    sayi=int(input("Bir sayı giriniz: "))
    toplam=toplam+sayi
print("Sonuc=",toplam)
```

Ekran Çıktısı:

```
Bir sayı giriniz: 5
Bir sayı giriniz: 6
Bir sayı giriniz: 7
Bir sayı giriniz: 0
Sonuc= 18
```

Bu örnekte toplam değişkenine ilk değer olarak 0 atandı. *sayi* değişkenine de 1 değeri verilmiştir (0 dışında herhangi bir değer verilebilir). *sayi!=0* ifadesi ile sayı 0 olmadığı sürece while bloğunun çalışması sağlandı. Girilen her sayı toplama eklendi ve 0 girildiğinde döngüden çıkılarak sonuç ekrana yazdırıldı.

Sıra Sizde:

- Klavyeden 1 girilene kadar girilen sayıların ortalamasını alan kodu yazınız.
- Girilen şifre "Python" olana kadar "Tekrar deneyiniz" uyarısı veren, "Python" girildiğinde "Giriş başarılı" uyarısı veren kodu yazınız.

Programlama dillerinde döngüler iç içe de kullanılabilir. Örneğin 3*3'lük matris ve adreslerini oluşturmak için iç içe döngü yapısı kullanılır.

	0	1	2
0	0,0	0,1	0,2
1	1,0	1,1	1,2
2	2,0	2,1	2,2

Örnek 20 :

```
for i in range(0,3):
    for j in range(0,3):
        print([i,j])
```

Ekran Çıktısı:

[0, 0]

[0, 1]

[0, 2]

[1, 0]

[1, 1]

[1, 2]

[2, 0]

[2, 1]

[2, 2]

Bu örnekte 0 ile 3 aralığında iki farklı döngü oluşturularak iç içe kullanılmıştır. Birinci for döngüsündeki **i** değeri 0 olduğunda; içindeki döngüde **j** değeri sırayla 0,1 ve 2 olmaktadır.

Sıra Sizde:

Çıktısı aşağıdaki gibi devam eden çarpım tablosunu iç içe döngü kurarak kodlayınız.

1 x 1 = 1

1 x 2 = 2

1 x 3 = 3

1 x 4 = 4

1 x 5 = 5

1 x 6 = 6

1 x 7 = 7

1 x 8 = 8

1 x 9 = 9

2 x 1 = 2

2 x 2 = 4

2 x 3 = 6

2 x 4 = 8

2 x 5 = 10

...

5.2.3. Break ve Continue Deyimleri

Break komutu döngüleri sonlandırır. Programlamada öngörülemeyen bir durum gerçekleştiğinde break komutu döngüden çıkılmasına imkân tanır. Döngüden çıkıldıktan sonra ise döngü sonrasındaki ilk satırdan kod çalıştırılmaya devam eder.

Örnek 21 :

```
i=1
while True:
    if (i==5):
        print("Döngüden çıkıldı")
        break
    print(i)
    i=i+1
```

Ekran Çıktısı:

```
1
2
3
4
Döngüden çıkıldı
```

Örnekte **while True:** ifadesi görülmektedir. Aksi belirtilmediği sürece döngünün devam etmesi anlamına gelen bu ifade sonsuz döngü anlatılırken de örnek olarak verilmişti. Burada kullanılan break ifadesi sayesinde while True: satırı ile sürekli devam etmesi istenen döngüden çıkılmıştır. Görüldüğü üzere i değişkeni 1 değerini alır. i değişkeni 5 değerine eşit olduğunda ise döngüden çıkılır. break ifadesi print(i) satırından önce olduğu için 5 değeri ekran çıktısında görülmez.

Sıra Sizde:

- Kullanıcıdan 1 ile 5 arasında bir sayı girmesi isteyiniz. Kullanıcı 3 sayısını girdiğinde break komutu ile döngüden çıkılarak "3 sayısı girildi ve döngü sona erdi" çıktısı veren kodu yazınız.
- Kullanıcıdan 8 karakterlik bir şifre girmesini isteyiniz. Kullanıcı 8'den az ya da daha fazla karakter içeren bir şifre girdiğinde "**Şifreniz 8 karakter olmalıdır.**" şeklinde uyarı verdiriniz. Kullanıcı şartlara uygun bir şifre girdiğinde de "**Şifreniz kaydedildi.**" uyarısı verdiriniz.

Örnek 22 :

```
sayi = int(input("Bir sayı girin: "))
for i in range(1, 10):
    if i == sayi:
        break
    print(i)
print("Döngü sona erdi")
```

Ekran Çıktısı:

```
Bir sayı girin: 8
1
2
```

3
4
5
6
7

Döngü sona erdi

Bu örnekte kullanıcı tarafından bir sayı girildi. **For** komutu kullanılarak 1 ile 10 aralığında bir döngü oluşturuldu. Kullanıcı 1-10 arası bir sayı girerse döngü kullanıcının girdiği sayıya kadar devam eder. Örneğin 8 sayısı girildiğinde döngü 8'e kadar çalıştıktan sonra durdurulur. 8 dâhil olmadığından ekranda en son 7 sayısı görülür.

Sıra Sizde: Yukarıdaki kodu çalıştırarak 1-10 aralığı dışında bir sayı giriniz ve ekran çıktısını sınıfta tartışınız.

Örnek 23 :

metin = "Ankara"

for i in metin:

if i == 'r':

break

print(i)

Ekran Çıktısı:

A
n
k
a

Bu örnekte "Ankara" metni for döngüsü ile ekran çıktısı olacakken r harfinde break ile döngü sonlandırılmıştır.

Örnek 24 :

sayilar=[10, 11, 12, 13, 14, 15, 16]

for aranan in sayilar:

print(aranan)

if(aranan==14):

print("14 sayısı bulundu")

break

Ekran Çıktısı:

10
11
12
13
14

14 sayısı bulundu

Bu örnekte, oluşturulan dizide bulunan 14 sayısı bulunduğunda döngüden çıkmıştır.

Sıra Sizde: Elemanları alfabadeki ilk 8 harf olan bir liste oluşturarak “e” harfine gelindiğinde döngüden çıkan kodu yazınız.

Örnek 25 :

```
import random

while True:

    n = random.randint(1, 20)
    print("Rastgele seçilen ", n)

    if n % 2 == 0:
        print("Çift sayı seçildi, döngü bitti")
        break
```

Ekran Çıktısı:

```
Rastgele seçilen 5
Rastgele seçilen 13
Rastgele seçilen 15
Rastgele seçilen 8
Çift sayı seçildi, döngü bitti.
```

Bu örnekte kullanılan **random** komutu ile 1-20 arasında rastgele bir sayının hafızaya alınması sağlanmıştır. Seçilen bu rastgele sayı çift bir sayı olana kadar döngü devam eder. Çift sayı tutulduğunda ise döngü sona erer.

Sıra Sizde:

- 1 ile 100 arasında rastgele 6 sayı seçerek ekrana yazdırınız.
- Random metodu ile 0-20 arası bir sayı seçerek kullanıcının bu sayıyı tahmin etmesini isteyiniz. Kullanıcının tahminine göre arttır ve azalt şeklinde uyarılar verdirerek doğru sonuca ulaşılmasını sağlayınız.

Continue komutu döngüyü başa döndürerek continue sonrasında yazılan kod bloğunun göz ardı edilmesini sağlar. Başka bir ifadeyle döngünün o anki adımını atlayarak kaldığı yerden devam eder.

Örnek 26 :

```
i = 0
while i < 10:
    i=i+1
    if i == 5:
        continue
    print(i)
```

Ekran Çıktısı:

```
1
2
3
4
6
7
```

8
9
10

Bu örnekte **i** değişkenin değeri 5 olduğunda continue ifadesi ile döngü başa dönerek 5 değerini yazmadan devam etmiştir.

Sıra Sizde: Aşağıdaki kod nasıl bir çıktı üretir? Yazınız.

```
i = 0
while i < 50:
    i=i+1
    if i>10 and i<45:
        continue
    print(i)
```

Örnek 27 :

```
while True:
    sifre = input("Bir şifre giriniz: ")
    if len(sifre) < 4 or len(sifre) > 4:
        print("4 karakterden oluşan bir şifre girmelisiniz.")
        continue
    else:
        print("Şifreniz oluşturuldu: ", sifre)
        break;
print("Şifrenizi while döngüsü içinde oluşturdunuz. ")
```

Ekran Çıktısı:

```
Bir şifre giriniz: 45
4 karakterden oluşan bir şifre girmelisiniz.
Bir şifre giriniz: 459876
4 karakterden oluşan bir şifre girmelisiniz.
Bir şifre giriniz: 4554
Şifreniz oluşturuldu: 4554
Şifrenizi while döngüsü içinde oluşturdunuz.
```

Bu örnekte kullanıcıdan bir şifre girmesi istendi. Girilen şifre 4 karakterden az ya da fazla olduğu sürece döngü, **continue** komutu ile devam etmektedir. 4 karakter şifre girildiğinde ise "**Şifre oluşturuldu**" mesajı verilerek **break** komutu ile döngüden çıkılmaktadır.

Önemli Not: Telefon numarası, T.C. kimlik numarası gibi sayısal ifadeler üzerinde matematiksel işlemler yapılmayacağı için string olarak tanımlanabilir.

Örnek 28 :

```
sayilar=[20,23,79,88,111,65]
```

```
for sayi in sayilar:
```

```
    if sayi%2 == 0:
```

```
        continue
```

```
    print(sayi)
```

Ekran Çıktısı:

23

79

111

65

Bu örnekte listedeki sayılardan çift olanlar continue kullanıldığı için atlanmış ve ekran çıktısında listelenmemiştir.

Sıra Sizde: 1-30 arasındaki sayıları bir liste hâline getirerek continue ile sadece tek olanları ekrana yazdırınız.

ÖLÇME VE DEĞERLENDİRME 5

A) Aşağıdaki boşluk doldurma sorularını cevaplayınız.

```
if sayi%3==0:
```

```
    print("A")
```

```
else:
```

```
    print("B")
```

1. Yukarıdaki kod çalıştırıldığında ve sayi değişkeni 7 olarak girildiğinde çıktısı üretilir.
2. if ((3>5 and 5<1) or 5==5) satırı değerini döndürür.
3. Döngüleri sonlandırmak için komutu kullanılır.
4. Rastgele sayı seçmek için komutu kullanılır.
5., döngünün başlangıç ve bitiş değeri belli olan durumlarda kullanılan, varsayılan olarak 0'dan (sıfır) başlayarak birer birer artan fonksiyondur.

B) Aşağıdaki çoktan seçmeli soruları cevaplayınız.

```
if yas>18 and ehliyet=="var":
```

```
    print("Başarılı")
```

6. Yukarıdaki kod bloğu ne anlama gelmektedir?

- A) yas değişkeni 18 ve ehliyet değişkeninin değeri var ise ekrana Başarılı yazar.
- B) yas değişkeni 18'den küçük ve ehliyet değişkeninin değeri var ise ekrana Başarılı yazar.
- C) yas değişkeni 18'den büyük veya ehliyet değişkeninin değeri var ise ekrana Başarılı yazar.
- D) yas değişkeni 18'den büyük ve ehliyet değişkeninin değeri var ise ekrana Başarılı yazar.
- E) yas değişkeni 18'den küçük veya ehliyet değişkeninin değeri var ise ekrana Başarılı yazar.

```
for sayilar in range(15):
```

```
    print(sayilar)
```

7. Yukarıdaki kod çalıştırıldığında üretilecek çıktının son değeri hangisidir?

- A) 15
- B) 14
- C) 10
- D) 1
- E) 0

```
i=3
```

```
while (i<7):
```

```
    print("MEB")
```

```
    i=i+1
```

8. Yukarıdaki kod çalıştırıldığında ekrana kaç kez MEB yazar?

- A) 2
- B) 3
- C) 4
- D) 6
- E) 7

```
for sayilar in range(10,2,-3):  
    print(sayilar)
```

9. Yukarıdaki kod çalıştırıldığında sırasıyla hangi çıktıları üretir?

- A) 10 7 4
- B) 10 2 3
- C) 2 5 8
- D) 10 8 6
- E) 2 4 6

```
toplam=0  
for sayilar in range(5):  
    toplam=toplam+sayilar  
print(toplam)
```

10. Yukarıdaki kod çalıştırıldığında nasıl bir çıktı üretir?

- A) 5
- B) 15
- C) 6
- D) 8
- E) 10

NOT: Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları veya faaliyetleri geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme birimine geçiniz.