

4. KOŞULLU DURUMLAR



Bu bölümde;

- ✓ Program yazma sürecinde kullanılan koşullu durumlar konusunda bilgi sahibi olacak,
- ✓ İç içe koşul durumu ifade etmeyi kavrayacak,
- ✓ Koşul durumlarında yapılan hatalar konusunda fikir sahibi olacaksınız.

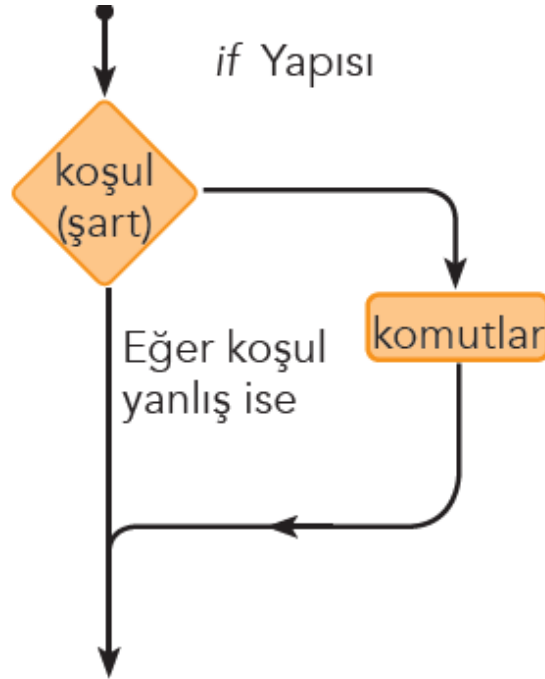
4.1. Boolean İfadesi

- Bilgisayar bilimi temelde 0 ve 1 değerleri üzerine kurulmuştur;
- 0 değeri **False**(Yanlış),
- 1 değeri **True**(Doğru) demektir.

4.2. Python'da İlişkisel Operatörler

İfade	Anlamı
$x==y$	Eğer x ve y birbirine eşitse (matematiksel olarak) doğrudur, değilse yanlıştır.
$x<y$	Eğer x, y'den küçükse doğrudur; değilse yanlıştır.
$x<=y$	Eğer x, y'den küçük ya da eşitse doğrudur; değilse yanlıştır.
$x>y$	Eğer x, y'den büyükse doğrudur; değilse yanlıştır.
$x>=y$	Eğer x, y'den büyük ya da eşitse doğrudur; değilse yanlıştır.
$x!=y$	Eğer x, y'den farklı ise (büyük ya da küçük) doğrudur; değilse yanlıştır.

4.3. “if” İfadesi



if ifadesinin yapısı:

if koşul:

ifadeler

şeklindedir.

4.3. “if” İfadesi

```
print("Lütfen bölme için iki sayı giriniz.")
bolum=int(input("Lütfen bölme için ilk sayınızı giriniz:"))
bolen=int(input("Lütfen bölme için ikinci sayınızı giriniz:"))
if bolen!=0:
    print(bolum,"/",bolen,"=", bolum/bolen)
```

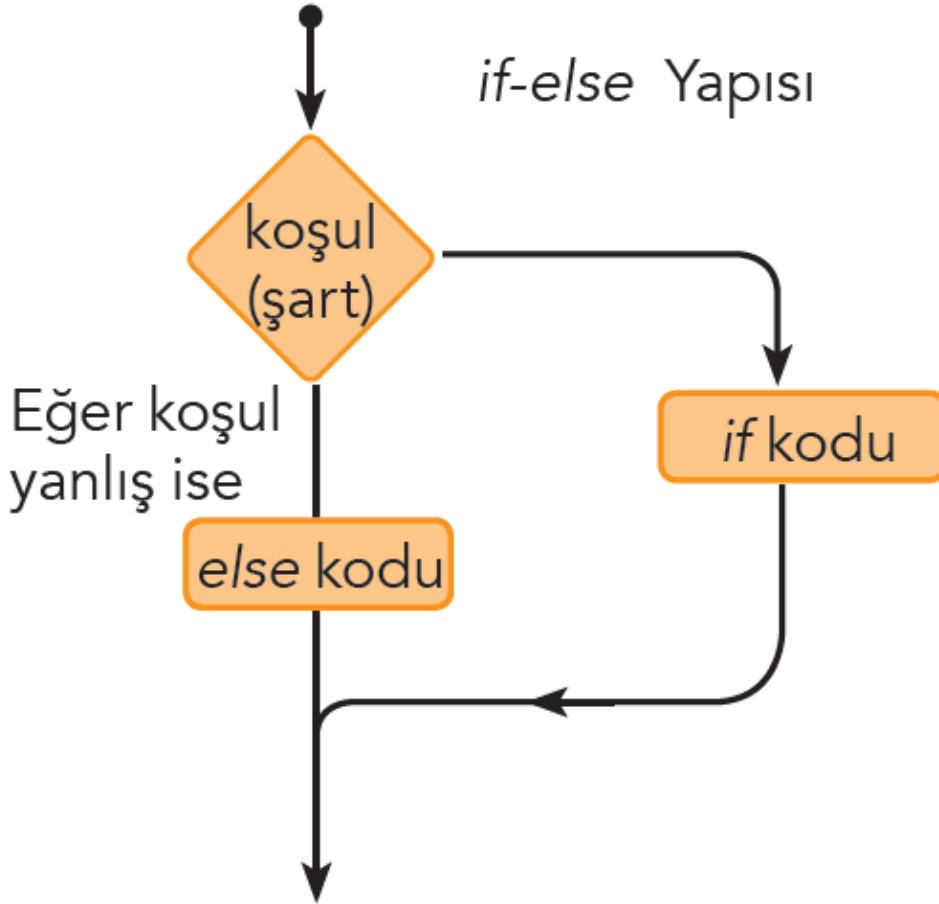
Ekran çıktısı aşağıdaki gibi olur:

```
Lütfen bölme için iki sayı giriniz.
Lütfen bölme için ilk sayınızı giriniz:32
Lütfen bölme için ikinci sayınızı giriniz:8
32/8=4.0
```

Başka değerler için test edecek olursak:

```
Lütfen bölme için iki sayı giriniz.
Lütfen bölme için ilk sayınızı giriniz:32
Lütfen bölme için ikinci sayınızı giriniz:0
```

4.4. “if/else” İfadesi



if/else ifadesinin yapısı:

if koşul:

ifadeler

else:

ifadeler

4.4. “if/else” ifadesi

```
print("Bölem işlemi için iki sayı giriniz")
bolum=int(input("ilk sayıyı giriniz="))
bolen=int(input("İkinci sayıyı giriniz="))
if bolen!=0:
    bolum=bolum/bolen
    print("Bölme işleminin sonucu=",bolum)
else:
    print("bölen sıfır olursa bölme işlemi yapılmaz")
```

4.4. “if/else” ifadesi

```
# Ağacın boyunu sorgulayan program.  
boy = input("Ağacın boyu kaç cm? ")  
if int(boy) > 150:  
    print ("Ağacın boyu uzun demek ki!")  
else:  
    print ("Ağacın boyu kısa")
```


4.5. Birleşik Boolean İfadesi

Birleşik Boolean İfadeleri oluşturan mantıksal operatörler –e1 ve e2 örneği

e1	e2	e1 and e2	e1 or e2	not e1
False (yanlış)	False (yanlış)	False (yanlış)	False (yanlış)	True (doğru)
False (yanlış)	True (doğru)	False (yanlış)	True (doğru)	True (doğru)
True (doğru)	False (yanlış)	False (yanlış)	True (doğru)	False (yanlış)
True (doğru)	True (doğru)	True (doğru)	True (doğru)	False (yanlış)

x = 10, y = 20 olarak veriliyor.

Buna göre aşağıda verilen kod örneklerini inceleyiniz.

```
b = (x == 10)
```

- # b'ye True değerini atar.

```
b = (x != 10)
```

- # b'ye False değerini atar.

```
b = (x == 10 and y == 20)
```

- # b'ye True değerini atar.

```
b = (x != 10 and y == 20)
```

- # b'ye False değerini atar.

```
b = (x == 10 and y != 20)
```

- # b'ye False değerini atar.

```
b = (x != 10 and y != 20)
```

- # b'ye False değerini atar.

```
b = (x == 10 or y == 20)
```

- # b'ye True değerini atar.

```
b = (x != 10 or y == 20)
```

- # b'ye True değerini atar.

```
b = (x == 10 or y != 20)
```

- # b'ye True değerini atar.

```
b = (x != 10 or y != 20)
```

- # b'ye False değerini atar.

4.6. Pass İfadesi

Pass ifadesi Python'da herhangi bir işlem yapmadan geçeceğimiz durumlarda kullanılır. Kısaca "Hiçbir şey yapmadan yola devam et!" anlamı katar.

```
if x == 2:
    print(x)
else:
    pass # x 2'ye eşit değilse hiçbir şey yapma
if x == 2:
    print(x) # yalnızca x 2'ye eşitse yazdır
```

4.7. Kayan Noktalı Eşitlik

```
d1 = 1.11 - 1.10
d2 = 2.11 - 2.10
print("d1 =", d1, " d2 =", d2)
if d1 == d2:
    print("Aynı")
else:
    print("Farklı")
```

4.8. İç İçe Koşul İfadeleri

```
deger = int(input("Lütfen 0...10 aralığında bir tam sayı giriniz: "))
    if deger >= 0 and deger <= 10: # İkili koşul kontrolü
        print("aralıkta")
print("tamamlandı")
```

```
deger = int(input("Lütfen 0...10 aralığında bir tam sayı giriniz: "))
    if deger >= 0: # İlk kontrol
        if deger <= 10: # İkinci kontrol
            print("aralıkta")
print("tamamlandı")
```

4.9. Çok Yönlü Karar İfadeleri

```
deger = int(input("Lütfen 0 yada 5 tam sayı değerlerinden birini girin: "))
if deger < 0:
    print("çok küçük")
else:
    if deger == 0:
        print("sıfır")
    else:
        if deger == 1:
            print("bir")
        else:
            if deger == 5:
                print("beş")
            else:
                print("çok büyük")
print("Tamamlandı")
```

4.9. Çok Yönlü Karar İfadeleri

```
# Klavyeden girilen 0-5 arasındaki sayıların yazı karşılığını veren program
deger = int(input("Lütfen 0 - 5 arasında bir değer girin: "))
if deger < 0:
    print("çok küçük")
elif deger == 0:
    print("sıfır")
elif deger == 1:
    print("bir")
elif deger == 2:
    print("iki")
elif deger == 3:
    print("üç")
elif deger == 4:
    print("dört")
elif deger == 5:
    print("beş")
else:
    print("çok büyük")
print("Tamamlandı")
```

4.10. Çok Yönlü ve Zincirleme Durum İfadeleri

Çok Yönlü Koşullu Durum İfadesi

```
value = int (input("Lütfen 0...3
arasında bir tam sayı giriniz:"))
if value==0: Kontrol 1
    print("sıfır")
elif value==1: Kontrol 2
    print("bir") Sonuç
elif value==2:
    print("iki")
elif value==3:
    print("üç")
else
    print("çok büyük")
print("Tamamlandı")
```

} Diğer kontroller atlanır.

Zincirleme Durum İfadesi

```
value = int (input("Lütfen 0...3
arasında bir tam sayı giriniz:"))
if value==0: Kontrol 1
    print("sıfır")
if value==1: Kontrol 2
    print("bir") Sonuç
if value==2: Kontrol 3
    print("iki")
if value==3: Kontrol 4
    print("üç")
if value>3: Kontrol 5
    print("çok büyük")
print("Tamamlandı")
```


4.11. Koşullu İfadeler

4.11. Koşullu İfadeler

Koşullu ifadelerin genel yapısı aşağıdaki gibidir.

Birinci Durum

if

Koşul

else

İkinci Durum

Koşul doğru ise koşullu ifadenin sonucu birinci durumdur. Koşul, “if” ifadesinde de görülebilen Boolean ifadesidir. Eğer koşul yanlış ise koşullu ifadenin sonucu ikinci durumdur.

```
n = int(input("Bir sayı giriniz: "))
print("|", n, "| = ", (-n if n < 0 else n), sep="")
```

Ekran Çıktısı

```
Bir sayı giriniz: -34
|-34| = 34
Bir sayı giriniz: 100
|100| = 100
```

4.11.1. Koşullu İfadelerde Hatalar

```
deger = int(input("Lütfen 0 - 5 arasında bir değer girin: "))
cevap="aralıkta değil" #Varsayılan Cevap
if deger == 0:
    cevap="Sıfır"
elif deger == 1:
    cevap="Bir"
elif deger == 2:
    cevap="iki"
elif deger == 3:
    cevap="üç"
elif deger == 4:
    cevap="dört"
elif deger == 5:
    cevap="beş"
print("Girdiğiniz sayı", cevap)
```

4.12. Mantık Karmaşası

- Önemli olan, mantık karmaşasına yol açmadan kullanabilmektir.
 1. `not (a == b and c != d)`
 2. `not (a == b and not (c == d))`
 3. `not (a == b) or not (c != d)`
 4. `a != b or c == d`

4.12. Mantık Karmaşası

- Çalıştırılırken en verimli yöntem basit düzeydeki mantıksal ifadelerdir.
- Basit düzeydeki mantıksal ifadeleri yazmak ve çalıştırmak daha kolaydır.
- Basit düzeydeki mantıksal ifadeler, çalıştırılırken de en verimli yöntemdir.
- Basit düzeydeki mantıksal ifadelerin değiştirilmesi, düzenlenmesi ve genişletilmesi de daha kolaydır.