

PYTHON PROGRAMLAMA DİLİ

KOŞULLU İFADELER
if-elif-else

Dr. Öğr. Üyesi M. Ozan AKI

KOŞULLU İFADELER - IF

Bazı durumlarda belirli kodları ya da kod bloklarını sadece belirli koşullar sağlandığında çalıştırmak isteyebiliriz.

Python dilinde bunun için koşullu ifadeler, yada da diğer adıyla karar yapıları bulunmaktadır: **if** *koşul*

if koşul:

koşul doğru ise yapılacaklar.

koşul doğru ise yapılacaklar.

koşul doğru ise yapılacaklar.

...

Bir koşul ;

ya **Doğrudur (True)**

ya **Yanlıştır (False)**

Koşul ifadelerinde bool (True ya da False) döndüren ifadeler kullanılır.

KOŞULLU İFADELER – IF ELSE

Bazı durumlarda koşul doğru olduğunda bir kod bloğu çalıştırılırken, yanlış olduğunda da başka bir kod bloğunu çalışması istenebilir.

Bu durumda **if** ifadesi **else** kelimesi ile devam ettirilerek **else:** bloğunun altına koşul *yanlış* olduğunda çalışmasını istediğimiz komutlar yazılır.

if koşul:

koşul doğru ise yapılacaklar.

koşul doğru ise yapılacaklar.

koşul doğru ise yapılacaklar.

...

else:

koşul yanlış ise yapılacaklar.

koşul yanlış ise yapılacaklar.

koşul yanlış ise yapılacaklar.

...

KOŞULLU İFADELER – IF ELIF

Bazı durumlarda çok sayıda koşulu kıyaslamak gerekebilir.

Bu gibi durumlarda

else ve *if* ifadelerinin birleşimi olan

elif

Kullanılarak birbirine bağlı koşul ifadeleri yazılabilir.

*Bu doğruysa bunu yap,
Değilse bu doğruysa bunu yap,
Değilse bu doğruysa bunu yap,
Değilse bu doğruysa bunu yap,
Değilse bu doğruysa bunu yap,
...*

```
if koşul_1:
    koşul_1 doğru ise yapılacaklar.
elif koşul_2:
    koşul_2 doğru ise yapılacaklar.
elif koşul_3:
    koşul_3 doğru ise yapılacaklar.
elif koşul_4:
    koşul_4 doğru ise yapılacaklar.
elif koşul_5:
    koşul_5 doğru ise yapılacaklar.
```

KOŞULLU İFADELER IF ELIF ELSE

elif ifadesiyle karşılaştırılan hiç bir koşulun doğru olmaması durumunda son koşulun altına else: bloğu açılarak tüm koşulların yanlış olması durumunda çalıştırılacak kodlar buraya yazılır

*Bu doğruysa bunu yap,
Değilse bu doğruysa bunu yap,
Değilse bu doğruysa bunu yap,
Değilse bu doğruysa bunu yap,
Değilse bu doğruysa bunu yap,
Değilse bu doğruysa bunu yap,
...
else:
Hiçbiri doğru değilse bunu yap*

```
if koşul_1:  
    koşul_1 doğru ise yapılacaklar.  
elif koşul_2:  
    koşul_2 doğru ise yapılacaklar.  
elif koşul_3:  
    koşul_3 doğru ise yapılacaklar.  
elif koşul_4:  
    koşul_4 doğru ise yapılacaklar.  
elif koşul_5:  
    koşul_5 doğru ise yapılacaklar.  
else:  
    Hiçbir koşul doğru değilse yapılacaklar
```

KOŞULLU İFADELER – KARŞILAŞTIRMA OPERATÖRLERİ

Koşul ifadelerinde;

a==b (a eşit midir **b**'ye?)

a!=b (a farklı mıdır **b**'den?)

a<b (a küçük müdür **b**'den?)

a<=b (a küçük veya eşit midir **b**'den?)

a>b (a büyük müdür **b**'den?)

a>=b (a büyük veya eşit midir **b**'den?)

Karşılaştırma operatörleri kullanılır.

Cevap Evet ise -> **True**

Hayır ise -> **False** döner.

```
>>> ortalama=65
>>> if ortalama>=50:
...     print("Gectiniz")
... else:
...     print("Kaldiniz")
...
...
Gectiniz
>>> |
```

```
>>> a=3
>>> b=5
>>> if a<b:
...     print("a kucuktur b'den")
...
...
a kucuktur b'den
>>>
>>> if a==b:
...     print("a esittir b'ye")
... else:
...     print("a farklıdır b'den")
...
...
a farklıdır b'den
>>> |
```

KOŞULLU İFADELER – MANTIKSAL OPERATÖRLER

Birden fazla koşulun kontrol edilmesi gerektiğinde Mantıksal Operatörler kullanılır:

and: VE anlamına gelir.

or: VEYA anlamına gelir.

not: DEĞİL anlamına gelir.

```
>>> a=3
>>> b=5
>>> c=7
>>> if a<b and b<c:
...     print("a < b < c")
... else:
...     print("a, b, c sirasizdir.")
...
...
... a < b < c
>>> |
```

```
>>> sicaklik=25
>>> if sicaklik>18 and sicaklik<28:
...     print("Sicaklik Normal")
... else:
...     print("Sicaklik Normal Disi")
...
...
... Sicaklik Normal
>>> |
```

İÇ İÇE KOŞULLU İFADELER

Blok girinti (indent) kurallarına uymak kaydıyla, bir koşul bloğunun altında başka koşullar, onların kod blokları altında da yine başka koşullar bulunabilir.

```
>>> a,b,c=1,4,1
>>> delta=b**2-4*a*c
>>> if delta>=0:
...     if delta==0:
...         x=(-b)/(2*a)
...         print(f"denklemin bir gerçel koku vardır: x={x}")
...     else:
...         x1=(-b+delta**0.5)/(2*a)
...         x2=(-b-delta**0.5)/(2*a)
...         print(f"denklemin iki gerçel koku vardır: x1={x1} x2={x2}")
... else:
...     print("denklemin gerçel koku yoktur.")
...
denklemin iki gerçel koku vardır: x1=-0.2679491924311228 x2=-3.732050807568877
>>> |
```

KOŞULLU İFADELER – MATCH CASE

Bazen değişkeni sadece belirli değerleri alıp almadığını kontrol etmek isteriz. *if-elif* yapısına alternatif olarak *match-case* kullanılabilir.

match yanına yazılan değişkenin değeri her bir **case** yanına yazılan değerler karşılaştırılır. Hangisi eşleşirse o **case** altındaki kodlar çalıştırılır.

```
match değişken:
    case değer_1:
        değişken, değer_1 olduğunda yapılacaklar.
    case değer_2:
        değişken, değer_2 olduğunda yapılacaklar.
    case değer_3:
        değişken, değer_3 olduğunda yapılacaklar.
    case değer_4:
        değişken, değer_4 olduğunda yapılacaklar.
    ...
```

KOŞULLU İFADELER – MATCH CASE

Hiçbir **case** değerinin eşleşmemesi durumunda ayrıca bir kod çalıştırılmak isteniyorsa, en sona

case _:

Yazılır ve hiç bir eşleşme olmazsa çalıştırılmak istenen kod bu blok altına yazılır.

match değişken:

case değer_1:

değişken, değer_1 olduğunda yapılacaklar.

case değer_2:

değişken, değer_2 olduğunda yapılacaklar.

case değer_3:

değişken, değer_3 olduğunda yapılacaklar.

case değer_4:

değişken, değer_4 olduğunda yapılacaklar.

...

case _:

hiç bir eşleşme olmazsa yapılacaklar.

KOŞULLU İFADELER – MATCH CASE

Bir **case** altında | sembolü kullanılarak birden fazla değer karşılaştırılabilir.

Herhangi değer eşleşirse o **case** altındaki kod çalışır.

```
>>> sayi=7
>>> match sayi:
...     case 1|3|5|7|9: print('tek sayi')
...     case 0|2|4|6|8: print('cift sayi')
...
...     tek sayi
>>>
```

```
>>> renk_kodu=5
>>> match renk_kodu:
...     case 0: print('siyah')
...     case 1: print('kahverengi')
...     case 2: print('kirmizi')
...     case 3: print('turuncu')
...     case 4: print('sari')
...     case 5: print('yesil')
...     case 6: print('mavi')
...     case 7: print('mor')
...     case 8: print('gri')
...     case 9: print('beyaz')
...     case _: print('hatali kod!')
...
...     yesil
>>>
```