

# YAPAY ZEKA VE PYTHON PROGRAMLAMA

*Koşullar*  
*Emir Öztürk*

*Dr. Öğr. Üyesi Altan MESUT'un "Programlama Dillerine Giriş" ders slaytları kaynak alınmıştır.*

## BOOLEAN DEĞERLER

- Bool
- İki farklı değer
- True - False

Boole cebiri sadece 0 (yanlış) ve 1 (doğru) değerleri üzerinde işlem yapmaktadır. Programlama dillerinde de sadece doğru veya yanlış değerini saklayabilen boolean türü değişkenler vardır (doğal olarak bu değişkenler için 1 bit büyüklük yeterlidir). Genellikle bu değişkenler 1 byte yer kaplarlar. Python'da bir değişkene True veya False değer atandıktan sonra type() fonksiyonu ile bu değişkenin türü kontrol edildiğinde <class 'bool'> yazdığı görülecektir.

## BOOL DEĞERLER

- Değişkene atanabilir.
- Karşılaştırılabilir
- and, or, not, ==, !=

x	y	And	Or	not(x)	==	!=
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	0	0	1
1	1	1	1	0	1	0

Bool değerler üzerinde mantıksal operatörler kullanılabilir ve bool değişkenler karşılaştırılabilirler. Aynı zamanda bir değişken içerisinde bool bir değer saklanabilmektedir.

## IF

- Koşul
- Boolean değer kontrolü
- Koşulun doğruluğu ile bloğa yönelme

Bir koşula göre işlem yapılması istendiğinde en çok kullanılan komut kümesi if & else yapısıdır:

if koşul: koşul doğru ise yapılacak işlem

else: koşul yanlış ise yapılacak işlem

Koşulun yanlış olması durumunda yapılacak bir işlem yok ise else: satırı yazılmaz.

Koşul verilirken Boole cebiri kuralları geçerlidir. Yani mantıksal (and, or, not) ve karşılaştırma (>, <, >=, <=, ==, !=) operatörleri kullanılabilir.

## KOŞUL İÇERİSİNDE MANTIKSAL OPERATÖRLER

- if a=5 and b=3
- if a<5
- if a==b
- if a != b
- if a or b
- if a == 3 or b == 5
- ~~if a=5 and b=3~~

if ile birlikte mantıksal operatörler kullanılabilir. Birden fazla koşul, and veya or ile birleştirilip kullanılabilir. Eşitlik kontrolü için == kullanılması gerekmektedir. = karakteri atama olduğu için if içerisinde kullanmak hatalı olacaktır.

## KOD BLOĞU

- Programlama dillerinin çalışma mantığı
- Bloklara ayırma
- Birden fazla satır
- Python'da bloklar (TAB)

Yapısal programlama dillerinde program kodu bloklara ayrılır. PASCAL'da kod bloğu begin ile başlar end ile biter. C, C++, C# ve JAVA gibi dillerde ise { ile başlar } ile biter.

Python'da blok başı ve sonunu belirtmek için belirli bir simge veya kelime kullanılmaz. Eğer bir satır üstündeki satırdan daha sağdan başladıysa blok başlangıcı olarak, daha soldan başladıysa üstündeki satır blok bitişi olarak düşünülür. Öncesinde eşit sayıda boşluk olan (aynı sütundan başlayan) ardışık satırlar aynı blok içinde kabul edilir.

':' karakteri genellikle bir bloğa başlanacağını gösterir. Bu nedenle ':'

karakterinden sonra ENTER tuşuna basılması, alt satırın başına bir TAB (genellikle 4 boşluk karakteri) eklenmesine neden olur. Boşluk sayısını arttırıp azaltabilirsiniz, fakat hiç boşluk kullanmadığınızda hata verecektir. Yapısal programlama dillerinde kodun daha kolay okunabilmesi için iç bloğu oluşturan satırlar dış bloktakilere göre daha sağdan başlar. Diğer dillerde bunu yapmak seçimlik iken, Python’da zorunluluktur.

#### ELSE

- Koşulun devamı
- Koşulun if içerisinde sağlanmaması ile bloğa yönelme
- Tanımlanma zorunluluğu yoktur

Bir koşula göre işlem yapılması istendiğinde en çok kullanılan komut kümesi if & else yapısıdır:

if koşul: koşul doğru ise yapılacak işlem

else: koşul yanlış ise yapılacak işlem

Koşulun yanlış olması durumunda yapılacak bir işlem yok ise else: satırı yazılmaz.

Koşul verilirken Boole cebiri kuralları geçerlidir. Yani mantıksal (and, or, not) ve karşılaştırma (>, <, >=, <=, ==, !=) operatörleri kullanılabilir.

#### IF-ELSE İLE BLOK KULLANIMI

```
if koşul:
    blok içi
    blok içi
else:
    blok içi
    blok içi
```

Verilen koşulun sağlandığı yada sağlanmadığı durumlarda yapılacak işlem birden fazla satır kapsayacak ise yeni bir blok içinde yazılması şarttır.

if koşul:

koşul doğru ise yapılacak işlem1

koşul doğru ise yapılacak işlem2

else:

koşul yanlış ise yapılacak işlem1

koşul yanlış ise yapılacak işlem2

## ELIF

- İlk tanımlanan koşulun sağlanmaması durumu
- Kendi koşuluna sahip
- Tanım zorunluluğu yoktur
- Birden fazla tanımlanabilir

Önceki slaytta olduğu gibi koşul yanlış olduğunda yapılacak herhangi bir işlem yoksa, else ifadesi kullanılmayabilir. Else ifadesinin kullanılması durumunda:

```
if bn >= 60 and bn < 70: print("(CC)")
elif bn >= 70 and bn < 80: print("(CB)")
elif bn >= 80 and bn < 90: print("(BB)")
elif bn >= 90 and bn < 95: print("(BA)")
elif bn >= 95: print("(AA)")
```

olacaktır.

•Bu şekilde yazıldığında ilk koşul doğru ise diğer 4 koşulu kontrol etmez, ikinci koşul doğru ise altındaki 3 koşulu kontrol etmez. Bu sayede program daha hızlı çalışır.

–C, C++, C# ve JAVA dilinde else if şeklinde yazılır

## ÖRNEK UYGULAMA

- Girilen sayının tek veya çift olduğunu tespit eden program

### ÖRNEK UYGULAMA

- Bir sayının mutlak değerini alıp ekrana gösteren program

### ÖRNEK UYGULAMA

- Klavyeden girilen 2 sayıdan birincisi büyük ise ikisini çarpan, değilse ikisini toplayan ve sonucu ekranda gösteren program

### ÖRNEK UYGULAMA

- Girilen 3 sayıdan en büyük olanını bulan program

#### ÖRNEK UYGULAMA

- BMI hesabı yapan program

#### ÖRNEK UYGULAMA

- Bir mağaza için sipariş tutarı ve kg değerleri ekrandan girilecek ve kg başına 1 tl kargo ücreti fiyata eklenip gösterilecektir. Bu tutar kargo hariç 100 TL'nin üstündeyse kargo ücreti hesaplanmayacaktır. Ayrıca tutar 200 TL'nin üzerinde ise toplam fiyata %10 indirim uygulanacaktır.