# Operatörler

Python dilindeki aritmetik, atama, karsılastırma ve mantıksal operatörleri isleyeceğiz.

## 1.1. ARITMETIK OPERATÖRLER

Python'da matematiksel işlemler yapmak iştediğimiz zaman kullandığımız operatörlere aritmetik operatör denir.

toplama +

çıkarma -

çarpma \*

e / Ne olursa olsun sonucu float (ondalıklı sayı) tipinden verecektir. Python'a has olan bu özelliktir. Yani 20 / 4 = 5.0

% kullanarak mod alabiliriz yani bölme işleminde kalanı bulabiliriz, örnek vermek gerekirse 20 % 6 işlemini yaptığımızda program bize 2 çıktısını verecektir.

// kullanarak bölme işlemindeki sonucu yani bölümü bulabiliriz. Bu işlem sonucunda kalan göz ardı edilerek bir tam sayı elde edilir, örnek vermek gerekirse 20 // 6 işlemini yaptığımızda program bize 3 sonucunu verecektir.

\*\* kullanarak üs alma işlemini yapabiliriz. Mesela 2 \*\* 3 işleminin sonucunu 8 olarak görürüz.

## 1.2. KARSILASTIRMA OPERATÖRLERI

Birden fazla ifadeni veya değişkeni karşılaştırmak için kullandığımız operatörlere karşılaştırma oepratörleri denir

- == eşittir (Eşit midir ?)
- 5 == 23 yazdığımız zaman alacağımız çıktı False olacaktır. Çünkü eşit değiller.
- 5 == 5 yazdığımız zaman alacağımız çıktı True olacaktır, çünkü eşit midir ifadesinin cevabı evet yani doğrudur.
- != eşit değildir. (Eşit değil midir ?)
- 5 != 23 yazdığımız zaman alacağımız çıktı True olacaktır. Çünkü eşit değil midir sorusuna yanıt olarak evet aldık ve eşit olmadıklarını gördük.
  - 5 != 5 yazdığımız zaman alacağımız çıktı False olacaktır, çünkü eşit değil midir ifadesinin cevabı hayır olduğu için yanlıştır.
  - > büyüktür (büyük müdür?)

Bu ifade, solunda yazan sayı/değeri sağında bulunan sayı/değerle karşılaştırır ve büyük olup olmadığını sorgular.

- 5 > 23 yazdığımız zaman alacağımız cevap False olur çünkü 5, 23'ten büyük değildir.
- < küçüktür (küçük müdür ?)

Bu ifade, solunda yazan sayı/değeri sağında bulunan sayı/değerle karşılaştırır ve küçük olup olmadığını sorgular.

- 5 < 23 yazdığımız zaman alacağımız cevap True olur çünkü 5, 23'ten küçüktür.
- >= büyük eşittir (büyük veya eşit midir ?)

Bu ifade, solunda yazan sayı/değeri sağında bulunan sayı/değerle karşılaştırır ve büyük veya eşit olup olmadığını sorgular.

- 5 >= 23 yazdığımız zaman alacağımız cevap False olur çünkü 5, 23'ten büyük veya eşit değildir. Ancak 5 >= 5 yazmış olsaydım True cevabını alacaktım.
- <= küçük eşittir (küçük veya eşit midir ?)

Bu ifade, solunda yazan sayı/değeri sağında bulunan sayı/değerle karşılaştırır ve küçük veya eşit olup olmadığını sorgular.

5 <= 23 yazdığımız zaman alacağımız cevap True olur çünkü 5, 23'ten küçüktür. Ancak 23 <= 22 yazmış olsaydım False çıktısını alırdım.

Operatörlerde yazılış sırası çok önemlidir. != değildir anlamına gelirken = ! bir şey ifade etmez ve program hata verir, buna dikkat edilmeli. Aynı zamanda <, <=, >, >= operatörlerinde de değerlerin hangi tarafa (sağına veya soluna ) yazıldığı önemlidir çünkü buna göre alınacak sonuç değişecektir.

#### 13. MANTIKSAL OPERATÖRLER

Python' da mantıksal operatörleri birden fazla koşulu/ifadeyi birlikte değerlendirmek için kullanırız. Özellikle döngülerle yapılan işlemlerde çok sık karşılaşırız.

Örnek vermek gerekirse dondurma almak için hem paramızın olması hem de dondurma alabileceğimiz bir yere gitmemiz gerekir. Bu tarz iki durumu kontrol etmemiz gerektiğinde mantıksal operatörleri kullanıyoruz.

and → ve İşlemlerin/ifadelerin hepsi True ise sonuç da True olur. Karşılaştırma işlemlerinden bir tanesinin bile sonucu False olursa sonuç da False olur.

(8 < 10) and (6 > 5) 8, 10'dan küçük olduğu için sol taraftan True sonucunu aldık. 6 da 5'ten büyük olduğu için sağ taraf da True oldu. Sonuç olarak bütün bu ifadenin sonucu True olur.

Eğer (8 > 10) şeklinde yani yanlış olacak şekilde yazsaydık bu sefer bir taraf True bir taraf False olacağı için sonuç False olurdu. İki taraf False olursa sonuç yine False olur.

or → veya İşlemlerden bir tanesinin sonucu True olursa sonuç da True olur. Eğer tüm işlemler False sonucunu veriyorsa alacağımız çıktı False olur.

(8 == 10) or (8 == 8) 8, 10'a eşit olmadığı için sol taraftan False dönüşü aldık, sağ tarafta ise 8, 8'e eşit olduğu için True aldık. Veya (or) kullanımında sadece bir tarafın True olması sonucun True olarak gelmesi için yeterli olacağından bütün bu ifadenin sonucu True olur.

not → değil Mantıksal işlemin sonucunu ters çevirir. True ise False, False ise True olur.
not(8 == 8) 8, 8'e eşit olduğu için True oldu. Ancak not ifadesi, alınan sonucu tersine çevireceği için sonucumuz False.

Atama operatörü = ile gösterilir ve eşittir operatörü olan == ile karıştırılmamalıdır.

x = 10 derken 10 değerini x 'e atayabilmek için = kullanarak atama işlemi yapıyoruz.

x, 10'a eşittir demek için ise x == 10 ifadesini kullanırız.

İkisi aynı şey değildir!

# 1.4. ATAMA OPERATÖRLERI

Python'da değişkenlere veri ataması yapmak istersek atama operatörlerini kullanırız.

Hepsini daha iyi görebilmek için işlemlerimizi x = 10 y = 5 üzerinden yapacağız. Burada x 'e 10 değerini, y'ye 5 değerini bir atama operatörü olan = ile atadık.

OPERATÖR	KISA KULLANIMI	UZUN KULLANIMI	SONUÇ
=	x = y	x = y	x = 5
+=	x += y	x = x + y	x = 15
-=	x -= y	x = x - y	x = 5
*=	x *= y	x = x * y	x = 50
/=	x/=y	x = x / y	x = 2.0
%=	x %= y	x = x % y	x = 0
**=	x **= y	x = x ** y	x = 100000
//=	x //= y	x = x // y	x = 2

# EXTRA KAYNAKLAR:

### PYTHON OPERATORS:

 https://dataflair.training/blogs/pythonoperator/#:~:text=Python%20operator%20is%20a%20symbol,revise%20the%20basic s%20of%20Python.