

== ve != operatörleri sadece veriyi kontrol eder.

=== ve !== operatörleri hem veriyi hem de veri tipini kontrol eder.

Bu sebeple

console.log(5 == "5") true değeri verirken

console.log(5 === "5") false değeri verir.



Karşılaştırma operatörleri

```
console.log(s1 == 5);
console.log(s1 == "5");
console.log(s1 === "5");
console.log(s1 !== 5);
console.log(s1 != "5");
console.log(s1 !== "5"); true
```

```
console.log(s1 > 5);
console.log(s1 > "4");
console.log(s1 >= 5);
console.log(s1 > "6");
```

ÖNEMLİ:

=== ve !== operatörleri hem veriyi hem de veri tipini kontrol eder.

== ve != operatörleri ise sadece veriyi kontrol

Büyük eşit ve küçük eşit işlemlerinde veri tipi kontrolü yapılmıyor.



Mantıksal operatörler

Operatör	Sembol	Örnek
And	&&	ort > 50 && ort < 70
Or	1	yas < 16 yas > 70
Not	!	! (yas > 45)

Α	В	A && B	A B
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

A	!A
1	0
0	1



Mantiksal operatörler

```
let s2 = true;
let s3 = true;
console.log(s2 && true);
console.log(s2 && s3);
console.log(s2 && s3 && false);

s3 = false;
console.log(s2 || s3 || false);
console.log(s2 || s3 || false);
s3 = null;
console.log(s2 && s3);
console.log(s2 || s3);

true
```

ÖNEMLİ:

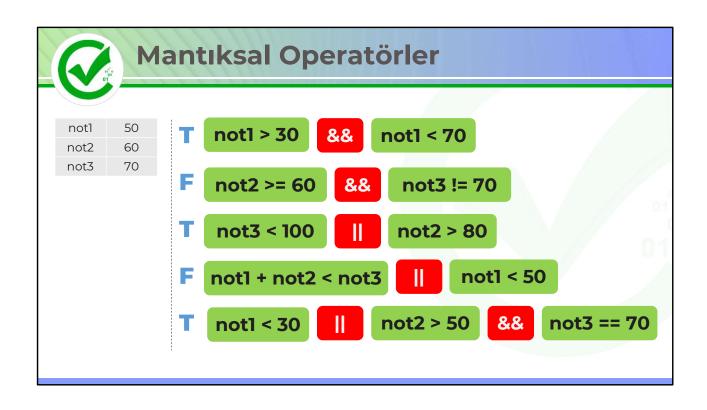
0, FALSE, NULL, undefined, "" ve **NaN** dışındaki durumlar **TRUE** kabul edilir.



Mantıksal operatörler

ÖNEMLİ:

- VEYA işleminde ilk TRUE değerin bulunması yeterlidir. Diğerlerinin kontrolüne gerek yoktur. Bu yüzden, ilk değişkenin değeri döndürülür.
- VE işleminde ise en sona kadar kontrol edilmesi gerekir. Dolayısıyla, hepsi doğru ise en sondaki değişkenin değeri döndürülür.





Atama operatörleri

Operatör	Örnek	Açıklama
=	x = y	Sağdaki değişkenin değerini soldakine kopyalar
+=	x += 1	x= x+1 işlemi gerçekleştirir.
-=	x -= 2	x= x-2 işlemi gerçekleştirir.
*=	x *= 3	x= x*3 işlemi gerçekleştirir.
/=	x/= 4	x= x/4 işlemi gerçekleştirir.
**=	x **= 2	x= x² işlemi gerçekleştirir.
%=	x %= 3	x = x mod 3 işlemi gerçekleştirir.
&=	x &= y	x = x VE y işlemi gerçekleştirir.
=	x = y	x = x VEYA y işlemi gerçekleştirir.



