

Javascript

Operatörleri

index

1. Logical Operatörler3

(OR – VEYA).....	4
&& (AND – VE).....	5
! (NOT - DEĞİL).....	5

2. Aritmetik Operatörler6

Atama operatörü.....	7
Toplama operatörü.....	7
Çıkarma operatörü.....	8
Çarpma operatörü.....	8
Bölme operatörü.....	9
Mod operatörü	9

Javascript Operatörleri



(<https://medium.com/@edisdev/algorithm-dersleri-2-42ee114f6cdf>)

1) Logical Operatörler

Javascript'te OR , AND ve NOT olmak üzere 3 tane logical (mantıksal) operatör vardır.

|| (OR – VEYA)

OR operatörü || ile gösterilir. (a||b) değeri sadece iki değerinde yanlış olduğu zaman yanlıştır. Diğer tüm durumlarda doğrudur.

```
( true || true );    // true
( false || true );   // true
( true || false );   // true
( false || false );  // false
```

if ile beraber kullanıldığında doğruysa yap anlami tasir.

```
let saat = 9;

if (saat < 10 || saat > 18) {
  alert( 'Ofis Kapalı' );}
```

if ile kullanıldığında OR operatörü her değeri boolean a çevirir ve soldan saga doğru ilk doğru değeri bulana kadar çalışır. Doğru değeri bulunca orjinal değeri gösterir ve sonraki değerlere bakmaz. Eger hepsi yanlışsa son değeri gösterir. Javascript te null,NaN,0 ,boş string ("") ve undefined yanlış olarak değerlendirilir.

```
let simdikiKullanici = null;
let varsayilanKullanici = "Gul";

let isim = simdikiKullanici || varsayilanKullanici || "isimsiz";

alert( isim ); // "Gul" seçilir - ilk doğru değeri bulduğundan
dolayı buradan dönülür ve ekrana "Gul" çıkar.
```

&& (AND – VE)

AND operatörü && ile gösterilir. Sadece her iki değerin de doğru olduğu durumlarda doğrudur.

```
( true && true );    // true
( false && true );   // false
( true && false );   // false
( false && false );  // false
```

OR dan farklı olarak ilk yanlış değeri bulur.

// Eğer ilk operand doğru ise her halükarda ikincinin değeri alınır.

```
alert( 1 && 0 ); // 0
alert( 1 && 5 ); // 5
```

// İlk operand yanlış ise ilk operandı döner ikinci operand geçilir.

```
alert( null && 5 ); // null
```

```
alert( 0 && "önemi yok" ); // 0
```

alert(1 && 2 && 3); // 3, Tüm değerler doğru ise en son olan gösterilir.

! (NOT - DEĞİL)

NOT operatörü ! ile gösterilir. Operand ın boolean değerini alıp tersine çevirir.

```
( !true ); // false
( !0 ); // true
```

!! işareti ile operand boolean değerine döner.

2) Aritmetik Operatörler

6 Adet aritmetik operatör vardır. Bunlar;

1. Atama operatörü
2. Toplama operatörü
3. Çıkarma operatörü
4. Çarpma operatörü
5. Bölme operatörü
6. Mod operatörü dır.

Aritmetiksel Operatörler

OP	ANLAMI
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	Kalan
++	Artırma
--	Eksiltme

(<https://slideplayer.biz.tr/slide/2736986/>)

1. *Atama operatörü(=):*

Değişkenlere değer atamak için kullanılır.

```
<script>  
var a = 1453;  
alert(a);  
</script>
```

2. *Toplama operatörü:*

+ sayı değişkenleri toplar, string değişkenleri birbirine ekler.

```
<script>  
  
var a = 5;  
  
var b = 6;  
  
var c = a + b;  
  
alert(c);  
</script>
```

++ değer arttırmak için kullanılır.

```
<script>  
  
var a = 2;  
  
a++;  
  
var c = a;  
  
alert(c);  
</script>
```

3. Çıkarma operatörü:

- değişkenlerde çıkartma işlemini yapar.

```
<script>  
var a = 5;  
var b = 3;  
var c = a - b;  
alert(c);  
</script>
```

-- değer azaltmak için kullanılır.

```
<script>  
var a = 2;  
  
a--;  
  
var c = a;  
  
alert(c);  
</script>
```

4. Çarpma operatörü:

* çarpma işlemi yapar.

```
<script>  
var a = 5;  
var b = 3;  
var c = a * b;  
alert(c);  
</script>
```


5. Bölme operatörü:

/ bölme işlemi yapar.

```
<script>
var a = 10;
var b = 2;
var c = a / b;
alert(c);
</script>
```

6. Mod operatörü:

% matematikteki mod işlemi yapar. Örneğin; '7%3' ifadesinin matematikteki karşılığı '7 mod 3' ifadesidir ve sonuç 1 çıkar.

Aritmetik operatörlerde işlem sırası matematikteki gibidir. Parantez kullanarak işlem sırası değiştirilebilir.

Aritmetik Operatörler: Örnek

İşlem	Operator	denklem	a'nın önceki değeri	a'nın sonraki değeri
Toplama	+	$a = a + 2$	4	6
Çıkarma	-	$a = a - 2$	4	2
Çarpma	*	$a = a * 2$	4	8
Bölme	/	$a = a / 2$	4	2
Modül	%	$a = a \% 3$	4	1

İşlem	Operator	denklem	a'nın önceki değeri	a'nın sonraki değeri
Artırma	++	$++a$	4	5
Eksiltme	--	$--a$	4	3