

Go Programlama Dili - Operatörler

Aritmetik Operatörler

+ - Toplama

```
x, y := 10, 3
fmt.Println(x + y) // 13
```

- - Çıkarma

```
fmt.Println(x - y) // 7
```

***** - Çarpma

```
fmt.Println(x * y) // 30
```

/ - Bölme

```
fmt.Println(x / y) // 3
```

% - Mod

```
fmt.Println(x % y) // 1
```

Atama Operatörleri

= - Atama

```
x := 5; x = 10
fmt.Println(x) // 10
```

+= - Toplayarak atama

```
x := 5; x += 3
fmt.Println(x) // 8
```

-= - Çıkarak atama

```
x := 5; x -= 2
fmt.Println(x) // 3
```

Karşılaştırma Operatörleri

== - Eşit mi

```
fmt.Println(10 == 20) // false
```

!= - Eşit değil mi

```
fmt.Println(10 != 20) // true
```

Mantıksal Operatörler

&& - Mantıksal AND

```
fmt.Println(true && false) // false
```

|| - Mantıksal OR

```
fmt.Println(true || false) // true
```

! - Mantıksal NOT

```
fmt.Println(!true) // false
```

Bitwise Operatörler

& - Bitwise AND

```
fmt.Println(6 & 3) // 2
```

| - Bitwise OR

```
fmt.Println(6 | 3) // 7
```

^ - Bitwise XOR

```
fmt.Println(6 ^ 3) // 5
```

&^ - AND NOT

```
fmt.Println(6 &^ 3) // 4
```

Pointer Operatörleri

& - Adres operatörü

```
x := 42; p := &x  
fmt.Println(p) // adres
```

*** - Pointer dereference**

```
fmt.Println(*p) // 42
```

Özel Operatörler

<- - Kanal operatörü

```
ch := make(chan int)  
go func() { ch <- 10 }()  
fmt.Println(<-ch) // 10
```

... - Slice genişletme

```
nums := []int{1,2,3}  
fmt.Println(sum(nums...))
```

Diğer

++ - Arttırma (postfix)

```
x := 5; x++  
fmt.Println(x) // 6
```

-- - Azaltma (postfix)

```
x := 5; x--  
fmt.Println(x) // 4
```