

işleçler

2025

İşleçler (Operator)

Sağında ve solunda bulunan değerler(operand) arasında bir ilişki kuran işaretlerdir.

- Aritmetik İşleçler
- Karşılaştırma İşleçleri
- Bool İşleçleri
- Değer Atama İşleçleri
- Aitlik İşleçleri
- Kimlik İşleçleri



Aritmetik İşleçler

Matematikte kullanılan ve sayılarla aritmetik işlemler yapmamızı sağlayan yardımcı araçlardır.

+ Toplama // Tamsayı bölme (taban bölme)

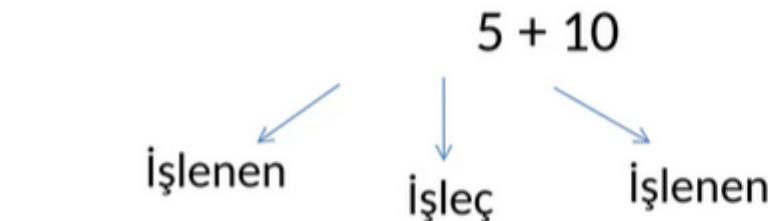
- Çıkarma % Modüler bölme (modülüs)

* Çarpma ** Kuvvet alma

/ Bölme

İşleç	İşleç Karakteri	Örnek
Toplama	+	>>> $4+4 = 8$
Çıkarma	-	>>> $3-4 = -1$
Çarpma	*	>>> $5*3 = 15$
Bölme	/	>>> $8/2 = 4$
Tam Sayı Bölme	//	>>> $19//4 = 4$
Mod (Bölümünden kalan)	%	>>> $21\%5 = 1$
Kuvvet Alma	**	>>> $8^{**2} = 64$

Aritmetik İşleçler


İşlenen İşleç İşlenen
 $>>> 45 + 60$ $5 + 10$ $>>> 25 / 4$
 105 6.25

$>>> 23 * 46$ $>>> 30 \% 4$
 1058 2

$>>> 10 - 5$ $>>> 5 // 2$
 5 2

$$\begin{array}{r} 30 \quad | \quad 4 \\ - 28 \quad | \quad 7 \\ \hline 02 \end{array}$$

Aritmetik İşleçler

```
>>> 25 ** 2  
625
```

```
>>> round(2.55)  
3
```

```
>>> pow(25, 2)  
625
```

```
>>> round(2.68, 1)  
2.7
```

```
>>> int(5/2)  
2
```

```
>>> pow(25,2,5)  
0
```

```
>>> (25**2)%5  
0
```

25 sayısının 2. kuvvetini alıp, elde ettiğimiz
sayının 5 ' e bölünmesinden kalan sayıyı hesaplamış

Karşılaştırma İşleçleri

İşlenenler arasında bir karşılaştırma ilişkisi kuran işlemlerdir.

`== > >= and not`

`!= < <= or`

`>>> 7==7`

`True`

`>>> a=5`

`>>> b=6`

`>>> a and b > 1`

`True`

`>>> |`

`>>> 5==5`

`True`

`>>> 5!=5`

`False`

`>>> 5!=6`

`True`

`>>> 5>=6`

`False`

`>>> 5>5.1`

`False`

`>>> 5>=4`

`True`

<code>==</code>	esittir
<code>!=</code>	eşit değildir
<code>></code>	büyüktür
<code><</code>	küçüktür
<code>>=</code>	büyük eşittir
<code><=</code>	küçük eşittir

`>>> (a or b) < 20`

`True`

Durum (Bool) İşleçleri

1 ve 0 gibi **True** ve **False** (Doğru ve Yanlış)

```
>>> a=1
```

```
>>> a==1
```

```
True
```

```
>>> a==2
```

```
False
```

```
>>> bool(a)
```

```
True
```

```
>>> bool(5)
```

```
True
```

```
>>> bool("Python")
```

```
True
```

```
>>> bool(" ")
```

```
True
```

```
>>> bool(0)
```

```
False
```

```
>>> bool("0")
```

```
True
```

```
>>> |
```

Değer Atama İşleçleri

= Değer atama işaretidir.

```
>>> a=5  
>>> a+=7  
+ = >>> a      =  
12  
>>> |
```

```
>>> a-=4  
>>> a  
8  
>>>
```

```
>>> a/=2  
>>> a  
* 4.0  
>>> |
```

```
>>> a*=4  
>>> a  
16.0  
>>> |
```

```
% = 16.0  
>>> a%=3    ** =  
>>> a  
1.0  
>>> |
```

```
>>> a=4  
>>> a**=2   =  
>>> a  
16  
>>> |
```

```
>>> 16/3  
5.333333333333333  
>>> a//3  
5  
>>> |
```

Aitlik İşleçleri

Bir karakter dizisi ya da sayının, herhangi bir veri tipi içinde bulunup bulunmadığını sorgulamamızı sağlayan işlemlerdir.

```
>>> a="abcd"
```

```
>>> "a" in a
```

```
True
```

```
>>> "f" in a
```

```
False
```

```
>>> adi="Çukurova"
```

```
>>> "ç" in adi
```

```
True
```

```
>>> "ç" in adi
```

```
False
```

```
>>> "r" in adi
```

```
True
```

```
>>> "Çukur" in adi
```

```
True
```

```
>>> "uv" in adi
```

```
False
```

```
>>> |
```

Kimlik İşleçleri

Eskiden Her nesnenin kimlik numarası (Identity) vardır.

id() Nesnenin bellekteki adresini gösterir.

```
>>> a=100
```

```
>>> id(a)
```

1406130748

```
>>> c=50
```

```
>>> id(c)
```

1406129984

```
>>> a=100
```

```
>>> a
```

100

```
>>> id(a)
```

1406130784

```
>>> b=100
```

```
>>> b
```

100

```
>>> id(b)
```

1406130784

```
>>>
```

-5 ... 257 arası

```
>>> a=258
```

```
>>> b=258
```

```
>>> a
```

258

```
>>> b
```

258

```
>>> id(a)
```

88906736

```
>>> id(b)
```

88906704

Aynı kimlik
Numarası (id).
100 sayısı için
bellekte iki farklı
nesne oluşturmuyor.

Alıştırmalar

Aritmetik İşleçler

Bir pizzacıkta büyük boy pizza 45 TL, küçük boy pizza 28 TL'dir. Müşteri 3 büyük, 2 küçük pizza alıyor ve %18 KDV ekleniyor. Toplam ödeme tutarını hesaplayan kodu yazın.

Cevap:

```
python

buyuk_pizza = 45
kucuk_pizza = 28
adet_buyuk = 3
adet_kucuk = 2

ara_toplam = (buyuk_pizza * adet_buyuk) + (kucuk_pizza * adet_kucuk)
kdv = ara_toplam * 0.18
toplam = ara_toplam + kdv

print(f"Ara Toplam: {ara_toplam} TL")
print(f"KDV (%18): {kdv} TL")
print(f"Toplam Ödeme: {toplam} TL")
```

Çıktı: Toplam Ödeme: 244.62 TL

Karşılaştırma İşleçleri

Bir öğrencinin vize notu 65, final notu 72'dir. Vizenin %40, finalin %60 etkisi var. Ortalama 50'den büyük veya eşitse geçti mi kontrolünü yapın.

Cevap:

```
python

vize = 65
final = 72
ortalama = (vize * 0.40) + (final * 0.60)

gecti_mi = ortalama >= 50
harfi = "AA" if ortalama >= 90 else "BA" if ortalama >= 85 else "BB" if ortalama >=
else "CC" if ortalama >= 70 else "DD" if ortalama >= 60 else "FF"

print(f"Ortalama: {ortalama}")
print(f"Geçti mi? {gecti_mi}")
print(f"Harf Notu: {harfi if gecti_mi else 'FF'}")
```

Çıktı: Ortalama: 69.2, Geçti: True

Durum (Bool) İşleçleri

Bir havaalanında bagaj kabul kriterleri: Ağırlık 20 kg'dan az VEYA business class yolcu olmalı VE tehlikeli madde taşımamalı. Kontrolü yapın.

Cevap:

python

```
bagaj_agirlik = 25
business_class = True
tehlikeli_madde = False

kabul_edilir = (bagaj_agirlik < 20 or business_class) and not tehlikeli_madde

print(f"Bagaj Ağırlığı: {bagaj_agirlik} kg")
print(f"Business Class: {business_class}")
print(f"Tehlikeli Madde: {tehlikeli_madde}")
print(f"Bagaj Kabul Edilir mi? {kabul_edilir}")
```

Çıktı: Bagaj Kabul Edilir mi? True

Değer Atama İşleçleri

Bir market kasasında başlangıç bakiyesi 1000 TL. Sırasıyla 3 müşteri 250 TL, 180 TL ve 420 TL ödeme yapıyor. Kısالتılmış atama işlemleriyle (`+=` operatörü) kasadaki toplam tutarı hesaplayın.

Cevap:

```
python

kasa_bakiye = 1000

print(f"Başlangıç Bakiyesi: {kasa_bakiye} TL")

kasa_bakiye += 250 # 1. müşteri
print(f"1. müşteri sonrası: {kasa_bakiye} TL")

kasa_bakiye += 180 # 2. müşteri
print(f"2. müşteri sonrası: {kasa_bakiye} TL")

kasa_bakiye += 420 # 3. müşteri
print(f"3. müşteri sonrası: {kasa_bakiye} TL")

print(f"\nGünlük Toplam Gelir: {kasa_bakiye - 1000} TL")
```

Çıktı: Günlük Toplam Gelir: 850 TL

Aitlik İşleçleri

Bir online mağazada "INDIRIM10", "YENIYIL", "HOSGELDIN" kupon kodları geçerli.
Kullanıcının girdiği kupon kodunun geçerli olup olmadığını kontrol edin.

Cevap:

python

```
gecerli_kuponlar = ["INDIRIM10", "YENIYIL", "HOSGELDIN"]

kullanici_kupon = "YENIYIL"
gecerli_mi = kullanici_kupon in gecerli_kuponlar

print(f"Girilen Kupon: {kullanici_kupon}")
print(f"Kupon Geçerli mi? {gecerli_mi}")

if gecerli_mi:
    print("✓ Kupon uygulandı! %10 indirim kazandınız.")
else:
    print("✗ Geçersiz kupon kodu.")

# Başka bir kontrol
kullanici_kupon2 = "ESKIKUPON"
print(f"\n'{kullanici_kupon2}' geçerli değil: {kullanici_kupon2 not in gecerli_kuponl
```

Çıktı: Kupon Geçerli mi? True

Kimlik İşleçleri

Python'da küçük sayılar için bellekte aynı nesne kullanılır (*identity*). İki değişkene aynı sayıyı atayarak hem değer eşitliği (`==`) hem kimlik eşitliğini (`is`) karşılaştırın.

Cevap:

```
python

# Küçük sayılar için
a = 100
b = 100

print("Küçük Sayılar (100):")
print(f"a == b: {a == b}") # Değer eşitliği
print(f"a is b: {a is b}") # Kimlik eşitliği
print(f"id(a): {id(a)}, id(b): {id(b)}")

print("\n" + "="*40 + "\n")

# Büyük sayılar için
x = 1000
y = 1000

print("Büyük Sayılar (1000):")
print(f"x == y: {x == y}") # Değer eşitliği
print(f"x is y: {x is y}") # Kimlik eşitliği (genelde False)
print(f"id(x): {id(x)}, id(y): {id(y)}")
```

```
# None kontrolü
deger = None
print(f"\ndeğer is None: {deger is None}") # Doğru kullanım
print(f"değer == None: {deger == None}") # Çalışır ama önerilmez
```

Çıktı: Küçük sayılarda `is` True, büyük sayılarda genelde False