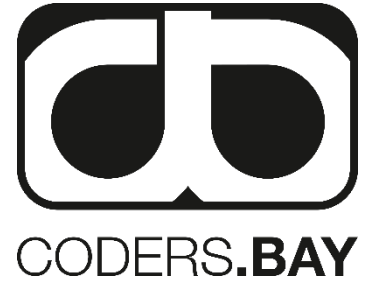


# **ARITHMETISCHE, RELATIONALE UND LOGISCHE OPERATOREN MIT BEISPIELEN ZUM TESTEN**

# ARITHMETISCHE OPERATIONEN

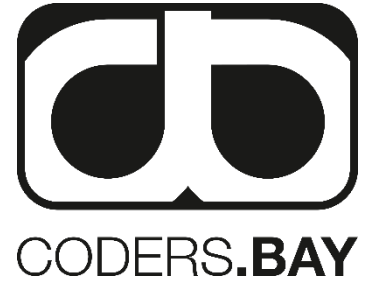


```
[...]  
int zahl1 = 5;  
int zahl2 = 10;  
System.out.println(zahl1 + zahl2);  
System.out.println(zahl1 - zahl2);  
//man kann auch mit Variablen und Zahlen gemeinsam rechnen  
System.out.println(zahl1 * 7);  
System.out.println(zahl1 / zahl2);  
[...]
```

## Ausgabe

```
15  
-5  
35  
0
```

# ARITHMETISCHE OPERATIONEN



## double als Datentyp für Kommazahlen

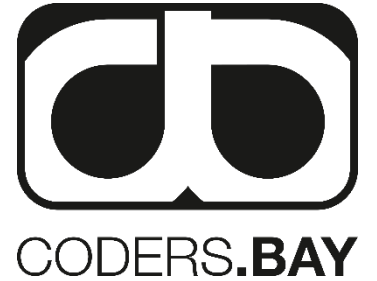
- Schlüsselwort `double`
- kein Komma, sondern Punkt

```
[...]  
int zahl1 = 5;  
double zahl2 = 10.0;  
System.out.println(zahl1 + zahl2);  
System.out.println(zahl1 / zahl2);  
[...]
```

### Ausgabe

```
15.0  
0.5
```

# KURZFORMEN



## Variablen und Operatoren

```
[...]  
int zahl1 = 5;  
zahl1 = zahl1 - 2;  
System.out.println(zahl1);  
zahl1 *= 3; //Kurzform für zahl1 = zahl1 * 3;  
System.out.println(zahl1);  
zahl1++; //Kurzform für zahl1 += 1; analog --  
System.out.println(zahl1);  
[...]
```

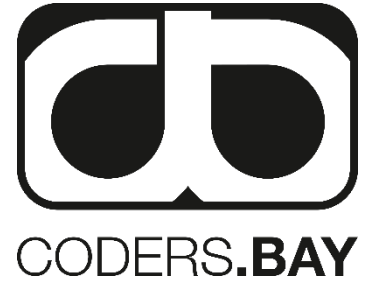
**Ausgabe**

3

9

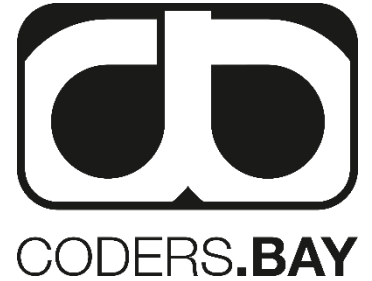
10

# RELATIONALE UND LOGISCHE OPERATOREN



Sehr häufig werden beim Programmieren auch die relationalen und logischen Operatoren verwendet. Sie werden in Kontrollstrukturen (z. B. if-Anweisung) eingesetzt, damit bestimmte Programmbereiche durchlaufen werden.

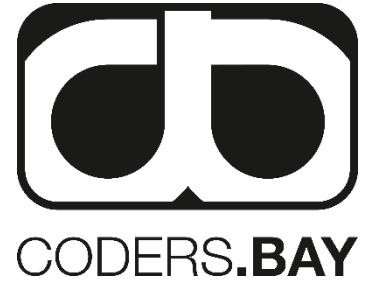
# RELATIONALE OPERATOREN



Relationale Operatoren vergleichen zwei Werte miteinander. Das Ergebnis ist immer ein boolescher Wert, also true oder false.

Operator	Beschreibung
<b>==</b>	Das doppelte Gleichheitszeichen überprüft zwei Werte auf ihre Gleichheit, kann jedoch nicht bei Zeichenketten eingesetzt werden.
<b>!=</b>	Das Gleichheitszeichen mit vorgestelltem Ausrufezeichen bedeutet "ungleich". Es überprüft zwei Werte auf Ihre Ungleichheit.
<b>&gt;</b>	Das „Größer“-Zeichen überprüft, ob ein Wert größer ist als der andere.
<b>&lt;</b>	Das „Kleiner“-Zeichen überprüft, ob ein Wert kleiner ist als der andere.
<b>&gt;=</b>	Das „Größer-Gleich“-Zeichen überprüft, ob ein Wert größer oder gleich dem anderen ist.
<b>&lt;=</b>	Das „Kleiner-Gleich“-Zeichen überprüft, ob ein Wert kleiner oder gleich dem anderen ist.

# BEISPIELE RELATIONALE OPERATOREN

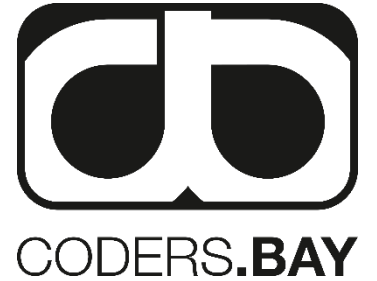


Für alle Beispiele gehen wir von **a=5** und **b=3** aus.

Hier sollten Sie zunächst selber nachdenken. Anschließend können Sie durch Betätigen der jeweiligen Schaltfläche die Lösung abrufen. Das Ergebnis der relationalen Operationen wird in der booleschen Variablen c gespeichert. Da c nur die Werte true oder false annehmen kann, ist dies der richtige Datentyp, um das Ergebnis einer relationalen oder logischen Operation zugewiesen zu bekommen.

# BEISPIELE

## RELATIONALE OPERATOREN



```
byte a=5;  
byte b=3;  
boolean c;
```

### Überprüfe das Ergebnis:

```
c = a==b;  
c = a!=b;  
c = a>b;  
c = a<b;  
c = a>=b;  
c = a<=b;
```

### Anmerkung

Zum Anzeigen der Ergebnisse, folgende Zeile hinter jeder Operation einfügen:

```
System.out.println(c);
```

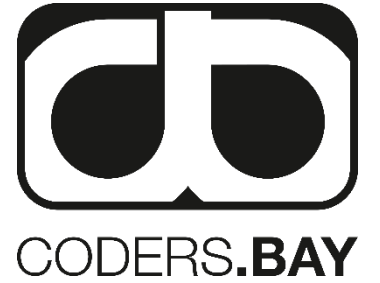


# LOGISCHE OPERATOREN

Die logischen Operatoren dienen dazu boolesche Ausdrücke auszuwerten.

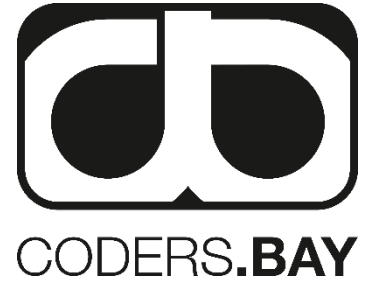
Operator	Beschreibung
<b>&amp;&amp;</b>	Doppeltes UND ist eine logische UND-Verknüpfung, bei der wir nur ein wahres Ergebnis erhalten, wenn beide Werte wahr sind. Ist an dieser Stelle bereits der erste Operator falsch (false) so wird der zweite Operand nicht mehr ausgewertet, da false und irgendwas bei einer logischen UND-Verknüpfung als Resultat immer false hat.
<b>  </b>	Doppeltes ODER ist eine logische ODER-Verknüpfung, bei der wir nur ein falsches Ergebnis erhalten, wenn beide Werte falsch sind. Ist an dieser Stelle bereits der erste Operator wahr (true) so wird der zweite Operand nicht mehr ausgewertet, da true und irgendwas bei einer logischen ODER-Verknüpfung als Resultat immer true hat.
<b>!</b>	Das Ausrufezeichen ist der Negierungsoperator in der booleschen Logik. Aus wahr (true) wird falsch (false) und umgekehrt.
<b>^</b>	Das "Dach" wird als Exklusiv-Oder bezeichnet. Entweder der erste oder der zweite Ausdruck muss wahr sein. Es dürfen aber nicht beide Ausdrücke gleich sein, damit das Ergebnis wahr wird.

# LOGISCHE UND-VERKNÜPFUNG



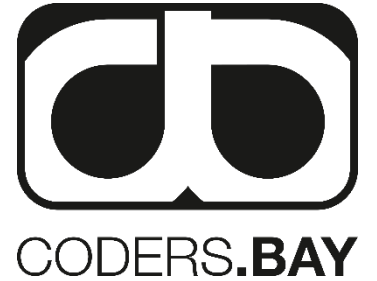
A	B	A && B
0	0	0
0	1	0
1	0	0
1	1	1

# LOGISCHE ODER-VERKNÜPFUNG



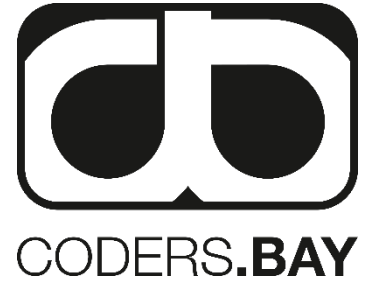
A	B	A    B
0	0	0
0	1	1
1	0	1
1	1	1

# LOGISCHE NOT-VERKNÜPFUNG (VERNEINUNG)



A	!A
0	1
1	0

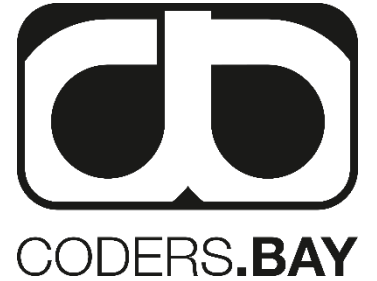
# BEISPIELE LOGISCHE OPERATOREN



Für alle Beispiele setzen wir **a=true** und **b=false**.

Hier sollten Sie erst überlegen. Anschließend können Sie durch Betätigen der jeweiligen Schaltfläche die Lösung abrufen. Das Ergebnis der booleschen Operationen wird in der booleschen Variablen c gespeichert. Da c nur die Werte true oder false annehmen kann, ist dies der richtige Datentyp, um das Ergebnis einer booleschen Operation zugewiesen zu bekommen.

# BEISPIELE LOGISCHE OPERATOREN



```
boolean a=true;  
boolean b=false;  
boolean c;
```

**Überprüfe das  
Ergebnis:**

```
c = a&&b;
```

```
c = a||b;
```

```
c = !a;
```

## **Anmerkung**

Zum Anzeigen der Ergebnisse, folgende  
Zeile hinter jeder Operation einfügen:

```
System.out.println(c);
```