

Fach: Programmieren - JavaScript
Thema: Vergleichs-, Verknüpfungsoperatoren

aus: H. Schröder, R. Steyer: JavaScript 1.8, Herdt 2014

Vergleichsoperatoren

Vergleichsoperatoren werden benutzt, um die Werte von zwei Operanden miteinander zu vergleichen. Werden Vergleichsoperatoren auf Zeichenketten angewendet, ist die Reihenfolge der Zeichen im Zeichensatz von Bedeutung. Ziffern werden z. B. niedriger eingestuft als Buchstaben und Großbuchstaben niedriger als Kleinbuchstaben. Nachfolgend sind die einzelnen Zeichen nach der Wertigkeit sortiert, wobei das Zeichen `!` das niedrigste und das Zeichen `-` das hochwertigste Zeichen ist.

```
!"#$%&'()*+,-./
0123456789
:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ
[\]^_`
abcdefghijklmnopqrstuvwxyz
{|}~
```

Folgende Vergleichsoperatoren werden unterschieden:

Name	Operator	Syntax	Beispiel	Wert
gleich	<code>==</code>	<code>Operand1 == Operand2</code>	<code>true == false</code> <code>2 + 4 == 6</code>	false true
ungleich	<code>!=</code>	<code>Operand1 != Operand2</code>	<code>"Hund" != "HUND"</code> <code>2 + 4 != 6</code>	true false
strikte (Un-) Gleichheit	<code>===</code> <code>!==</code>	<code>Operand1 === Operand2</code> <code>Operand1 !== Operand2</code>	<code>2 + 4 === 6</code> <code>"Hund" !== "HUND"</code>	true true
kleiner als	<code><</code>	<code>Operand1 < Operand2</code>	<code>4 < "5"</code> <code>"ADE" < "ABC"</code>	true false

Name	Operator	Syntax	Beispiel	Wert
größer als	<code>></code>	<code>Operand1 > Operand2</code>	<code>"abd" > "abc"</code> <code>"X" > 10</code>	true false
kleiner/gleich	<code><=</code>	<code>Operand1 <= Operand2</code>	<code>6 <= 7</code> <code>"abc" <= "abc"</code>	true true
größer/gleich	<code>>=</code>	<code>Operand1 >= Operand2</code>	<code>"6" >= "6"</code> <code>"Buch" >= "buch"</code>	true false

Wenn Sie zwei Operanden auf strikte Gleichheit prüfen, müssen diese den gleichen Wert und den gleichen Datentyp besitzen. Wird auf einfache Gleichheit geprüft, können auch Typumwandlungen zu einer Auswertung von `true` führen. Im Falle von strikter Gleichheit ist dies nicht möglich.

```
document.write(2 == "2"); // liefert nach der Typumwandlung von "2" nach 2 true
document.write(2 === "2"); // liefert false, da unterschiedliche Typen vorliegen
```

Verknüpfungsoperator

Dieser Operator, der auch Konkatenationsoperator (engl.: concatenate = verknüpfen) oder Stringverkettungsoperator genannt wird, verknüpft zwei Zeichenketten und liefert die zusammengesetzte Zeichenkette als Ergebniswert.

Name	Operator	Syntax	Beispiel	Wert
Verbinden	+	Operand1 + Operand2	"Zimmer" + "pflanze"	Zimmerpflanze
			x = 10; "x-Wert:" + x	x-Wert:10

Das Zeichen für den Verknüpfungsoperator kann **nicht** vom Berechnungsoperator für eine Addition unterschieden werden. Deshalb sollten Sie beim Programmieren darauf achten, ob Ihre Variablen Zahlen oder Zeichenketten enthalten. Wird der Operator `+` verwendet, wird er als Stringverkettungsoperator interpretiert, wenn die Variablen nicht explizit als Zahlen angegeben werden. Diese Tatsache führt zu dem Ergebnis, dass die Anweisungen `"1" + "1"` und `1 + "1"` den Wert `"11"` ergeben und nicht die Zahl 2. Dieses Verhalten wird „polymorph“ (vielgestaltig) genannt.

Logische Operatoren

Im Gegensatz zu den Vergleichsoperatoren werden mit den logischen Operatoren die booleschen Wahrheitswerte `true` und `false` miteinander verknüpft. Der Ergebniswert eines logischen Ausdrucks besteht aus einem booleschen Wert.

Name	Operator	Syntax	Beispiel	Wert
UND	<code>&&</code>	Operand1 <code>&&</code> Operand2	<code>true && true</code> <code>false && true</code> <code>true && false</code> <code>false && false</code>	<code>true</code> <code>false</code> <code>false</code> <code>false</code>
ODER	<code> </code>	Operand1 <code> </code> Operand2	<code>true true</code> <code>false true</code> <code>true false</code> <code>false false</code>	<code>true</code> <code>true</code> <code>true</code> <code>false</code>
NICHT	<code>!</code>	<code>!Operand</code>	<code>!true</code> <code>!false</code>	<code>false</code> <code>true</code>