

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

Zuweisungsoperatoren

Der einfachste Zuweisungsoperator ist das Gleichheitszeichen `=`. Sie können damit den Wert von Variablen festlegen.

```
var a = 42.1;
```

Der Zuweisungsoperator kann in Verbindung mit anderen Operatoren eingesetzt werden und gilt für alle Datentypen. Eine häufige Operation ist es, bestehende Werte in Variablen zu verändern und erneut zuzuweisen.

```
a = a + 5; // erhöhe den Wert in der Variablen a um 5
```

Da solche Anweisungen sehr häufig benötigt werden, gibt es für alle genannten binären Operatoren eine Kurzschreibweise, um die Operatorfunktion mit der Zuweisung zu verknüpfen. Das letzte Beispiel kann daher auch kürzer geschrieben werden:

```
a += 5; // erhöhe den Wert in der Variablen a um 5
```

Operator	Wirkung	Beispiel	Ergebnis
<code>=</code>	Einfache Wertzuweisung	<code>a = 5;</code>	<code>a = 5</code>
<code>+=</code>	Addieren und Zuweisen	<code>a = 5;</code> <code>a += 5;</code>	<code>5 + 5</code> <code>a = 10</code>
<code>+=</code>	Anfügen und Zuweisen	<code>a = 'Ei';</code> <code>a += 'dotter';</code>	<code>'Ei' + 'dotter'</code> <code>a = 'Eidotter'</code>
<code>-=</code>	Subtrahieren und Zuweisen	<code>a = 5;</code> <code>a -= 2;</code>	<code>5 - 2</code> <code>a = 3</code>
<code>*=</code>	Multiplizieren und Zuweisen	<code>a = 5;</code> <code>a *= 2</code>	<code>5 * 2</code> <code>a = 10</code>
<code>/=</code>	Dividieren und Zuweisen	<code>a = 20;</code> <code>a /= 4</code>	<code>20 / 4</code> <code>a = 5</code>
<code>%=</code>	Modulo-Operation und Zuweisen	<code>a = 5;</code> <code>a %= 3</code>	<code>5 % 3</code> <code>a = 2</code>

3.9 Rangfolge der Operatoren

Die Operatoren sind in ihrer Rangordnung festgelegt und werden dementsprechend nacheinander abgearbeitet:

Rangstufe (1=höchste Priorität)	Operatorzeichen	Operatortyp	Operatortyp
1	() []	Klammer	Gruppierung
2	! ++ -- - ~	NICHT Inkrement und Dekrement Negation NICHT	Logischer Operator Arithmetischer Operator Arithmetischer Operator Bit-Operator
3	* / %	Multiplikation Division Modulo-Operator	Arithmetische Operatoren
4	+	Aneinanderreihung	Konkatenationsoperator
5	+ -	Addition Subtraktion	Arithmetische Operatoren
6	<< >> >>>	Linksverschiebung Rechtsverschiebung Zero-Fill-Rechtsverschiebung	Bit-Operatoren
7	< > <= >=	Kleiner als Größer als Kleiner gleich Größer gleich	Vergleichsoperatoren
8	== === != !==	Werte sind gleich Werte sind ungleich	Vergleichsoperatoren
9	&	UND	Bit-Operator
10	^	ENTWEDER ODER	Bit-Operator
11		ODER	Bit-Operator
12	&&	UND	Logischer Operator
13		ODER	Logischer Operator
14	? :	Bedingung	Konditionaler Operator
15	= += -= *= /= %= &= ^= = <<= >>= >>>=	Zuweisungsoperatoren	Zuweisungsoperatoren
16	, (Komma)	Aneinanderreihung	

Geklammerte Ausdrücke besitzen die höchste Priorität und werden zuerst berechnet. Zur besseren Lesbarkeit und zur Vermeidung von Fehlern sollten Sie alle Ausdrücke, die mehr als zwei Operanden besitzen, durch das Setzen von Klammern erweitern.

```
a = 8 + 2 * 7 + 3      // a = 8 + 14 + 3 = 25
a = (8 + 2) * (7 + 3)  // a = 10 * 10 = 100
```