


Informationsblatt	System und- Anwendungssoftware	
Operatoren		

Operatoren

Eine Operation ist als Verknüpfung von Operanden mittels Operatoren zu verstehen.

Operation: 3 + 22;

Operand: 3 und 22


Operator: +

Arithmetische Operatoren

C# kennt die üblichen Operatoren der vier Grundrechenarten Addition, Subtraktion, Division und Multiplikation. Darüber hinaus werden in der nachfolgenden Tabelle weitere Operatoren beschrieben.

Operator	Beschreibung
+	->Als Addition bildet er die Summe zweier Operanden (x+y). ->Als Vorzeichenoperator beschreibt er eine positive Zahl (+x).
-	->Als Subtraktionsoperatoreingesetzt, bildet er die Differenz zweier Operanden (x-y). ->Als unärer Vorzeichenoperator beschreibt er eine negative Zahl (-x).
*	Multiplikationsoperator; multipliziert zwei Operanden (x*y).
/	Divisionsoperator; dividiert zwei Operanden (x/y), behält den nachkommateil der Division.
%	Restwertoperator; dividiert zwei Operanden und liefert als Ergebnis den Restwert der Operation (x%y)=x - (x / y) * y.
++	Erhöht den Inhalt des Operanden um 1. Das Ergebnis der Operation ++x ist der Wert des Operanden nach der Erhöhung.
--	Das Ergebnis der Operation x++ ist der Wert des Operanden vor der Erhöhung. Verringert den Inhalt des Operanden um 1. Das Ergebnis der Operation --x ist der Wert des Operanden nach der Verringerung. Das Ergebnis der Operation x-- ist der Wert des Operanden vor der Verringerung.

Besonderheiten der Division: Die Konsole zeigt den Wert 0, da beide Literale dem Typ ,int' zugeordnet werden. Wird ein Literal dem Typ ,double' zugeordnet, wird das Ergebnis als ,double' zurückgegeben.	<pre>double value = 3 / 4; Console.WriteLine(value); //'0' value = 3.0 / 4; Console.WriteLine(value); //'0,75'</pre>
Erhöhung von ,x' nach der Zuweisung.	<pre>int x = 5; int y = x++; Console.WriteLine(y); //x=6 //'5'</pre>
Erhöhung von ,x' vor der Zuweisung	<pre>int x = 5; int y = ++x; Console.WriteLine(y); //x=6 //'6'</pre>

Informationsblatt	System und- Anwendungssoftware	
Operatoren		

Vergleichsoperatoren

Vergleichsoperatoren vergleichen zwei Ausdrücke miteinander. Der Rückgabewert ist immer ein boolescher Wert, also entweder ‚true‘ oder ‚false‘. Vergleiche können auf Gleichheit bzw. Ungleichheit sowie auf >>größer<< und >>kleiner<< durchgeführt werden.

Operator	Beschreibung
a==b	(Vergleichsoperator) prüft, ob Ausdruck a Ausdruck b entspricht, und gibt in diesem Fall 'true' zurück
a!=b	Ergebnis der Operation ist 'true', wenn a ungleich b ist.
a>b	Ergebnis der Operation ist 'true', wenn a größer b ist.
a<b	Ergebnis der Operation ist 'true', wenn a kleiner b ist.
a<=b	Ergebnis der Operation ist 'true', wenn a kleiner oder gleich b ist.
a>=b	Ergebnis der Operation ist 'true', wenn a größer oder gleich b ist.

Vergleichsoperatoren haben eine höhere Priorität als der Zuweisungsoperatoren.	<code>bool compare; compare = 99 <= 100;</code>
---	--

Zuweisungsoperatoren

Operator	Beschreibung
=	x=y weist x den Wert von y zu.
+=	x+=y weist x den Wert von x+y zu.
-=	x-=y weist x den Wert von x-y zu.
=	x=y weist x den Wert von x*y zu.
/=	x/=y weist x den Wert von x/y zu.
%=	x%=y weist x den Wert von x%y zu.
&=	x&=y weist x den Wert von x&y zu.
=	x =y weist x den Wert von x y zu.
^=	x^=y weist x den Wert von x^y zu.
<<=	x<<=y weist x den Wert von x<<y zu.
>>=	x>>=y weist x den Wert von x>>y zu.

Logische Operatoren

Logische Operatoren führen logische Vorgänge mit booleschen-Operanden durch und geben einen booleschen-wert zurück.

Operator	Beschreibung
!	Unärer Negationsoperator. Der Ausdruck '!a' ist 'true', wenn 'a' einen unwahren Wert beschreibt, und 'false', wenn 'a' wahr ist.
&	(And-Operator, 1.Variante) Der Ausdruck 'a&b' ist dann 'true', wenn sowohl 'a' als auch 'b true' sind. Dabei werden in jedem Fall beide Ausdrücke ausgewertet.
	(Or-Operator, 1. Variante) Der Ausdruck 'a b' ist 'true', wenn entweder 'a' oder 'b' wahr ist. Dabei werden in jedem Fall beide Ausdrücke ausgewertet.
^	(Xor-Operator) Der Ausdruck 'a^b' ist 'true', wenn die beiden beteiligten Operanden unterschiedliche Wahrheitswerte haben.
&&	(And-Operator, 2.Variante) Der Ausdruck 'a&&b' ist 'true', wenn sowohl 'a' als auch 'b true' sind. Zuerst wird 'a' ausgewertet. Sollte 'a false' sein, ist in jedem Fall der Gesamtausdruck unabhängig von 'b' auch falsch. 'b' wird dann nicht mehr ausgewertet.
	(Or-Operator, 2.Variante) Der Ausdruck 'a b' ist 'true', wenn 'a' oder 'b true' ist. Zuerst wird 'a' ausgewertet. Sollte 'a' bereits 'true' sein, ist in jedem Fall der Gesamtausdruck unabhängig von 'b' auch Wahr. 'b' wird dann nicht mehr ausgewertet.

Vergleichsoperatoren können Literale miteinander vergleichen und der boolesche Rückgabewert kann als Operant der Logischen Operation genutzt werden.	<code>Console.WriteLine(true & false); //False Console.WriteLine(true & !(false)); //True Console.WriteLine(5==6 11<=10); //False</code>
--	---