

ZUSAMMENFASSUNG Operatoren

EINFÜHRUNG

Ein Operator können Variablen und Werte zu Ausdrücken verbunden werden. Ohne Operatoren, kann man nichts mit Variablen oder Werten machen.

Man kann die Operatoren in Python untergliedern.

1. Unäre Operatoren: Haben einen Operanden, bezieht sich auf eine Variable
2. Binäre Operatoren: Haben zwei Operanden, bezieht sich auf zwei Variablen
3. Ternäre Operatoren: Haben drei Operanden, bezieht sich auf drei Variablen

Bei unären Operatoren handelt es sich um Vorzeichen (also + oder -) von Zahlen. Bei einer Subtraktion von einem Wert von einem anderen, ist ein Minuszeichen allerdings ein binärer Operator da er zwei Operanden betrifft.

DER ZUWEISUNGSOPERATOR =

Mit diesem Operator kann man einer Variablen einen Wert zuweisen.

Zum Beispiel:

`result1 = 14`

➔ 14 ist hier ein einzelner Wert und wird auch als Konstante bezeichnet

`result2 = 14 + 7`

➔ 14 und 7 sind mit einem Operator verbunden, man bezeichnet das als Ausdruck

`result3 = (14+4) * 2`

➔ Das ist ein komplexer Ausdruck

Zuweisungsoperatoren arbeiten immer von rechts nach links. Der Operand rechts wird dem Operanden links zugewiesen. Es wird zuerst der komplexe Ausdruck ausgewertet und dann das Ergebnis zugewiesen. Dies geschieht aufgrund der Operanden Priorität.

ARITHMETISCHE OPERATOREN

Diese Operatoren führen arithmetische Berechnungen durch, beispielsweise die Grundrechenarten.

Die folgende Tabelle gibt einen Überblick über alle arithmetischen Operatoren von Python.

Operator	Beschreibung	Beispiel
-	Vorzeichen -	number = 3
+	Vorzeichen +	number =3 (oder number = +3)
+	Addition	number = 3 + 4
-	Subtraktion	number = 3 - 4
*	Multiplikation	number = 3 * 4
/	Division	number =12 / 4
%	Modulo Operator (Rest der Division)	number = 32 % 3 (Ergebnis: 2)
**	Exponentation	number = 3 **4 (Ausgesprochen: 3 hoch 4)
//	Ganzzahl Division (Division ohne Rest)	number = 32 // 3 (Ergebnis 10)

KOMBINIERTES ZUWEISUNGSOPERATOREN

Ist eine Kombination aus Zuweisungs- und arithmetischen Operatoren.

Zum Beispiel:

variable1 = 3

variable2 = 3

Möchte man den Wert der Variable1 um 10 erhöhen, kann man einfach 10 addieren.

variable1 = variable1 + 10

Eine kompaktere Schreibweise ist allerdings:

variable1 +=10

WICHTIG

Hierbei handelt es sich um einen kombinierten Zuweisungsoperator. Dieser funktioniert auch mit allen anderen arithmetischen Operatoren!

VERGLEICHOPERATOREN UND DER DATENTYP BOOL

Vergleichsoperatoren ermöglichen es innerhalb eines Programms Vergleiche durchzuführen und etwas über die Beziehung zwischen zwei Operanden auszusagen.

Operator	Beschreibung	Beispiel
<	kleiner als	1 < 6 (True)
>	größer als	1 > 6 (False)
<=	kleiner als oder gleich	4 <= 4 (True) 3 <= 4 (True)
>=	größer als oder gleich	4 >= 4 (True) 6 >=4 (True)
==	gleich	4 == 4 (True) 3 == 4 (False)
!=	ungleich	4 != 4 (False) und 3 !=4 (True)

Der Basisdatentyp Bool

Mit diesem Typ können Wahrheitswerte dargestellt werden. Es gibt zwei unterschiedliche Werte: True und False.

True und False sind aber auch gültige Konstanten, die immer im Code verwendet werden können, auch wenn gerade kein Vergleich durchgeführt wird. Der Wert einer Variablen kann also auch True oder False sein.

WICHTIG

Die Anfangsbuchstaben von True und False müssen immer groß geschrieben werden.

LOGISCHE OPERATOREN

Logische Operatoren geben genau wie die Vergleichsoperatoren nach der Auswertung einen Rückgabewert zurück (True oder False).

Operator	Beschreibung	Beispiel
or	logisches ODER	(a < b) or (c == d)
and	logisches UND	(a < b) and (c == d)
not	logisches NICHT	Not b

BITWEISE OPERATOREN

Bitweise Operatoren arbeiten mit dem Binärsystem. Mit ihnen kann man den Wert einzelner Bits gezielt verändern.

Operator	Beschreibung	Beispiel
<<	Linksshift (alle Bits nach links schieben)	x << 1 x << 4
>>	Rechtsshift (alle Bits nach rechts schieben)	x >> 1 x >> 4
&	UND-Operator	2 & 3
	ODER-Operator	2 3
^	Exklusiv-ODER-Operator	2 ^ 3

OPERATOREN PRIORITÄTEN

Operatoren Prioritäten regeln die Rangfolge von Operatoren.

In der Python Dokumentation ist die Rangfolge aufgelistet. Die Operatoren ganz unten in der Liste werden als erstes ausgeführt, die weiter oben werden zuletzt ausgeführt.

Die Dokumentation ist hier zu finden:

<https://docs.python.org/3/reference/expressions.html#operator-precedence>