



Parser

Inhalt

Allgemein	2
Syntax	3
Funktionen	3
Operationen	4
Konstanten	5
Anhang	6
Schlüssel	6
Weitere Beispiele	6

Allgemein

Dialog Lohn bietet bei bestimmten Berechnungs- und Anzeigefunktionen die Möglichkeit, den Wert mittels Formel umzurechnen. Wenn keine Formel angegeben wird, also das Eingabefeld leer gelassen wird, erfolgen keine Berechnung und der Wert wird nicht verändert.

Zur Berechnung der Formel kommt ein so genannter Parser zur Anwendung, welcher die Formel interpretiert und die Umrechnung vornimmt. Diese Parser-Funktionen sind in einer eigens dafür vorgesehenen Bibliothek Namens `muParser.dll` zu finden. Die Bibliothek `muParser.dll` muss sich daher auf dem Programmverzeichnis befinden, damit die Berechnung möglich ist.

Syntax

Damit die Formel erfolgreich berechnet werden kann, muss sie in eine bestimmte Syntax vorliegen. In diesem Abschnitt finden sind alle Funktionen, Operatoren und Konstanten mit Beispielcode aufgeführt.

Funktionen

Nachfolgende Funktionen werden unterstützt.

Funktion	Beschreibung
sin	Sinus Funktion <code>sin(1.0)</code> berechnet den Sinus von 1.0
cos	Cosinus Funktion <code>cos(x)</code> berechnet den Sinus von x
tan	Tangens Funktion <code>tan(0.5)</code> berechnet den Tangens von 0.5
asin	Arcus Sinus Funktion <code>asin(x)</code> berechnet den Arcus Sinus von x
acos	Arcus Cosinus Funktion <code>acos(aaa)</code> berechnet den Arcus Cosinus aaa
atan	Arcus Tangens Funktion <code>atan(x/2)</code> berechnet den Arcus Tangens x/2
sinh	Sinus Hyperbolikus Funktion <code>sinh(x)</code> berechnet den Sinus Hyperbolikus x
cosh	Cosinus Hyperbolikus Funktion <code>cosh(x)</code> berechnet den Cosinus Hyperbolikus x
tanh	Tangens Hyperbolikus Funktion <code>tanh(x)</code> berechnet den Tangens Hyperbolikus x
asinh	Arcus Sinus Hyperbolikus Funktion <code>asinh(x)</code> berechnet den Arcus Sinus Hyperbolikus x
acosh	Arcus Cosinus Hyperbolikus Funktion <code>acosh(x)</code> berechnet den Arcus Cosinus Hyperbolikus x
atanh	Arcus Tangens Hyperbolikus Funktion <code>atanh(x)</code> berechnet den Arcus Tangens Hyperbolikus x
log2	Logarithmus Funktion zur Basis 2 <code>log(x/2+0.5)</code> berechnet den Logarithmus zur Basis 2 von x/2+0.5
log10	Logarithmus Funktion zur Basis 10 <code>log10(abc)</code> berechnet den 10er Logarithmus von abc

log	Logarithmus Funktion zur Basis 10 <code>log(A*0.8)</code> berechnet den 10er Logarithmus von A*0.8
ln	Logarithmus Funktion zur Basis e <code>ln(x)</code> berechnet den Logarithmus zur Basis e von x
exp	Exponentialfunktion e^x <code>exp(2)</code> e hoch 2 (2.718281828459045235360287 hoch 2)
sqrt	Quadratwurzel Funktion <code>sqrt(9)</code> berechnet die Quadratwurzel von 9
sign	Vorzeichen bestimmen <code>sign(x-12)</code> bestimmt das Vorzeichen von x-12
rint	Runden auf nächsten ganzzahligen Wert <code>rint(z)</code> runden von z auf Ganzzahl <code>rint(z*20)/20</code> runden auf 5 Rappen
abs	Absolut Wert bestimmen <code>abs(x)</code> Absolut Wert von x bestimmen
if	if ... then ... else Funktion <code>if(x>10,1,2)</code> Wenn x grösser als 10, dann Wert 1 zurückgeben und sonst Wert 2 zurückgeben
min	Minimalwert einer Liste ermitteln <code>min(a,b,X1,Y2,12,abc)</code>
max	Maximalwert einer Liste ermitteln <code>max(x,y,1.5)</code> Bestimmt Maximalwert von x, y und 1.5
sum	Summenwert einer Liste ermitteln <code>sum(x,y,z)</code> Bildet Summe von x, y und z <code>sum(x*12,y/2)</code> Bildet Summe von x*12 und y/2
avg	Durchschnittswert einer Liste ermitteln <code>avg(x,y,a,b)</code> Bildet Durchschnittswert von x, y, a und b

Operationen

Nachfolgende Operatoren werden unterstützt.

Funktion	Beschreibung
=	Zuweisungszeichen <code>x=y*12</code> Der Variablen x den Wert von y*12 zuweisen
and	logisches AND (Anwendung im Zusammenhang mit der if-Anweisung) <code>if(x==10 and y == 12,1,2)</code> Wenn x gleich 10 und y gleich 12, dann Wert 1, sonst Wert 2
or	logisches OR (Anwendung im Zusammenhang mit der if-Anweisung) <code>if(x==10 or y == 12,1,2)</code> Wenn x gleich 10 oder y gleich 12, dann Wert 1, sonst Wert 2
xor	logisches XOR (Anwendung im Zusammenhang mit der if-Anweisung) <code>xx</code>
<=	größer oder gleich (Anwendung im Zusammenhang mit der if-Anweisung) <code>if(x<=10,1,2)</code> Wenn x kleiner oder gleich 10, dann Wert 1, sonst Wert 2

>=	kleiner oder gleich (Anwendung im Zusammenhang mit der if-Anweisung) <code>if(x>=10,1,2)</code> Wenn x grösser oder gleich 10, dann Wert 1, sonst Wert 2
!=	ungleich (Anwendung im Zusammenhang mit der if-Anweisung) <code>if(x!=10,1,2)</code> Wenn x ungleich 10, dann Wert 1, sonst Wert 2
==	gleich (Anwendung im Zusammenhang mit der if-Anweisung) <code>if(x==10,1,2)</code> Wenn x gleich 10, dann Wert 1, sonst Wert 2
>	grösser als (Anwendung im Zusammenhang mit der if-Anweisung) <code>if(x>10,1,2)</code> Wenn x grösser als 10, dann Wert 1, sonst Wert 2
<	kleiner als (Anwendung im Zusammenhang mit der if-Anweisung) <code>if(x<10,1,2)</code> Wenn x kleiner als 10, dann Wert 1, sonst Wert 2
+	Addition <code>a + 2.5 + x</code> Verschiedene Additionen
-	Subtraktion <code>a - b - 1</code> Verschiedene Subtraktionen
*	Multiplikation <code>3.5 * x</code> Multiplikation mit 3.5
/	Division <code>a / 2</code> Division durch 2
^	x^y <code>12^2</code> Zahl 12 im Quadrat <code>x^3</code> Variable x hoch 3
()	öffnende und schliessende Klammer <code>y = ((x+12)-1)*2</code> addiert zuerst x und 12 und multipliziert anschliessend mit dem Faktor 2

Konstanten

Die Formeln können auch die hier angegebenen Konstanten enthalten

Funktion	Beschreibung
<code>_e</code>	Euler Konstante (2.718281828459045235360287) <code>y = x * _e</code> berechnet Produkt aus x und 2.718...
<code>_pi</code>	Pi (3.141592653589793238462643) <code>y = x / _pi</code> dividiert x durch 3.141...

Anhang

Schlüssel

In oben aufgeführter Syntax sind verschiedene Beispiele dargestellt. Dabei wurden willkürliche Variablen wie x, y, aaa, etc. verwendet. Als gültige Variable sind die im Anhang des Benutzerhandbuches aufgeführten möglich.

Weitere Beispiele

Im Abschnitt 2 ist die Syntax mit einfachen Beispielen zu finden. Die Funktionen können aber auch kombiniert werden, was zu komplexerem Code führt.

runden auf 5 Rappen

```
rint(z*20)/20
```

Die Variabel z wird auf 5 Rappen auf- bzw. abgerundet.

Multiplikation von Maximalwerten

```
max(x,2) * max(y,12,z*2)
```

Ermittle den Maximalwert von x und 2 sowie Maximalwert von y, 12 und z*2 und multipliziere die Ergebnisse.

Berechnen des Ferienanspruchs linear

```
SF / TS
```

Dadurch wird z.B. bei einem Teilzeit-Angestellten, welcher z.B. 20 Tage Ferienanspruch (zu BG * Sollzeit) hat, auch tatsächlich 20 Tage angezeigt.