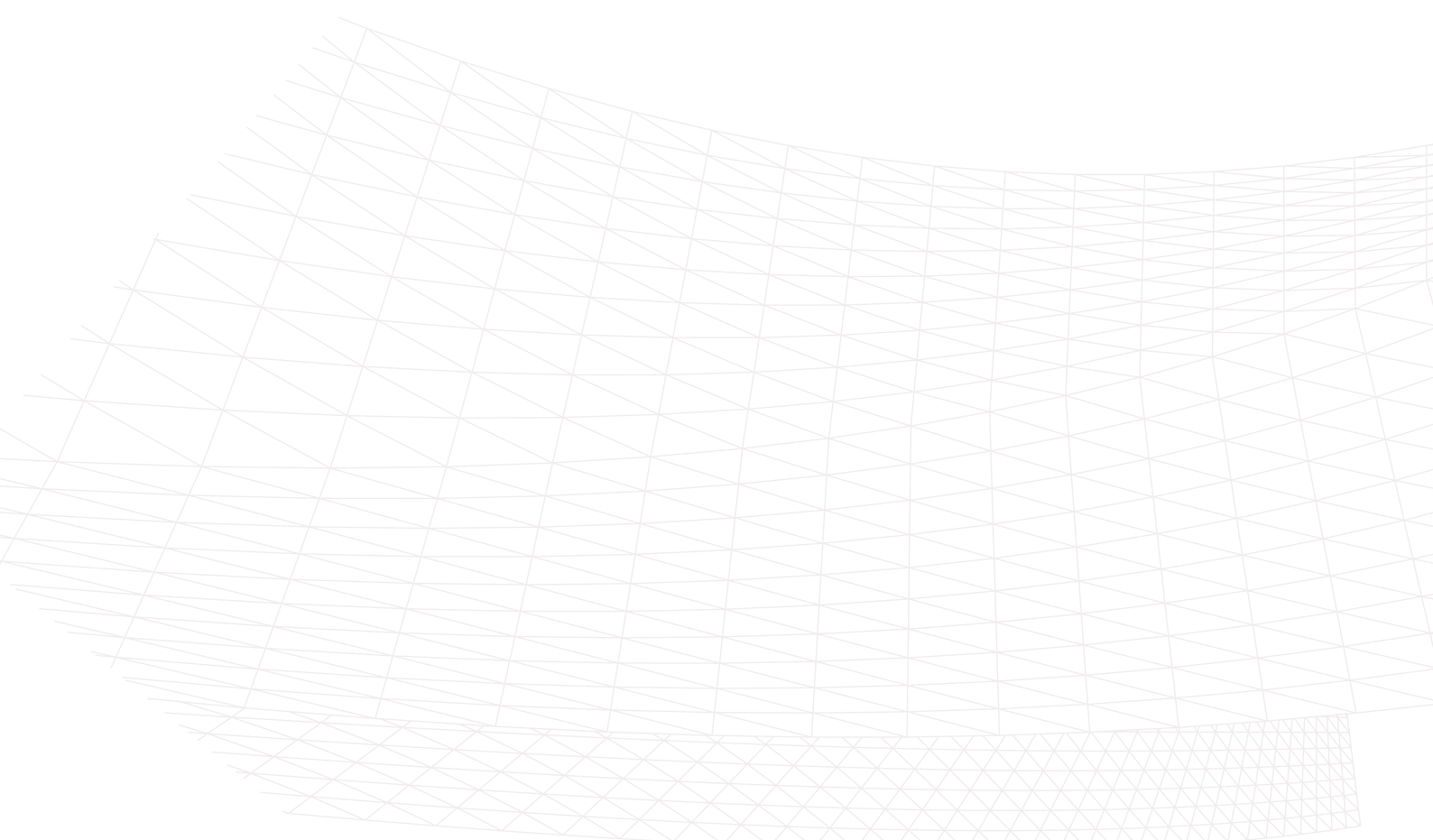


# Archicad 26 Referenzhandbuch



# Funktionen in Ausdrücken

Die in diesem Abschnitt beschriebenen Konstanten, Operatoren und Funktionen können zum Erstellen von Eigenschaften für Archicad-Elemente verwendet werden, die aus einer Berechnung (einen Ausdruck) generiert werden.

**KONSTANTEN**

**OPERATOREN**

**LOGISCHE FUNKTIONEN**

**MATHEMATISCHE FUNKTIONEN**

**TRIGONOMETRISCHE FUNKTIONEN**

**STATISTISCHE FUNKTIONEN**

**TEXTBEARBEITUNGSFUNKTIONEN**

**DATENKONVERTIERUNG**

**Verwandte Themen**

[Als Berechnung definierte Element-Eigenschaften](#)

[Als Berechnung definierte Eigenschaften: Beispiele](#)

[Weitere Details](#)

## KONSTANTEN

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

Pi [PI]

Wahr [TRUE] Falsch [FALSE]

**Pi [PI]**

**Wert**

3.141592654

**Wahr [TRUE] Falsch [FALSE]**

Gibt WAHR oder FALSCH zurück bei einer Funktion, in der das Argument Wahr oder Falsch erfordert.

**Beispiel**

Die Funktion Zeichenfolgen verbinden [TEXTJOIN] bietet die Option zum Ignorieren leerer Felder.

```
TEXTJOIN ("Begrenzer"; IgnorierenWennLeer; "Text1"; "Text2"; ...)
```

Im folgenden Ausdruck TEXTJOIN fügen wir WAHR für das Argument IgnorierenWennLeer ein; dies ist ein Wahr/Falsch-Datentyp.

**Ausdruck**

```
TEXTJOIN (" ", TRUE, Umbau-Status, Zugehöriger Raumname)
```

Der Ausdruck ignoriert daher leere Felder.

*Ein Beispiel find Sie unter [Zeichenfolgen verbinden \[TEXTJOIN\]](#).*

# OPERATOREN

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

Addieren [+]

Dividieren [/]

Gleich [=]

Größer als [>]

Größer oder Gleich [>=]

Kleiner als [<]

Kleiner oder Gleich [<=]

Multiplizieren [\*]

Nicht gleich [<>]

Hoch [^]

Subtrahieren [-]

## Addieren [+]

Fügt zwei Argumente (Operanden) hinzu.

### Syntax

```
[+]
```

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen oder Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Wie Datentyp des Arguments

### Beispiel

Summe von Steigungshöhe und Auftrittstiefe einer Treppe

### Ausdruck

```
Standard-Steigungshöhe + Standard-Auftrittstiefe
```

### Ergebnis

Fügt die beiden Längen-Werte hinzu, um ein Längenergebnis zu erhalten (z. B. 0,50)

### Anmerkungen

*Siehe auch* [Summe \[SUM\]](#).

## Dividieren [/]

Dividiert zwei Argumente (Operanden).

### Syntax

```
[ / ]
```

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen oder Winkel

### Datentyp des Ergebnisses

Hängt von den Datentypen der Argumente ab

### Beispiel

Berechnen des Fensteranteils eines Raums durch Division seiner Gesamt-Fensterfläche durch seine Nettofläche.

### Ausdruck

```
Fenster-Oberflächenbereich/Nettofläche
```

### Ergebnis

Dividiert zwei Flächenwerte für ein numerisches Ergebnis (z. B. 0,31)

### Anmerkungen

Beide Argumente sind Flächenwerte. Dieser Ausdruck dividiert sie und gibt eine Zahl ohne Einheiten zurück (das Verhältnis beider Flächen).



## Gleich [=]

Vergleicht zwei Argumente (Operanden). Gibt Wahr zurück, wenn ihre Werte gleich sind, Falsch, wenn sie nicht gleich sind.

### Syntax

```
[=]
```

### Datentypen der Argumente

Wahr/Falsch, Ganzzahl, Zahl, Länge, Fläche, Volumen, Winkel oder String (muss für alle Operanden identisch sein)

### Datentyp des Ergebnisses

Wahr/Falsch

### Beispiel

Prüfen, ob einem Raum die Kategorie Wohnen zugewiesen ist.

Bei beiden Argumenten, die verglichen werden (Klassifizierung und Wohnen) handelt es sich um String-Daten.

### Ausdruck

```
Archicad Klassifizierung -25 = "Wohnbereich"
```

### Ergebnis

Wahr für Räume, die für "Wohnen" klassifiziert sind, ansonsten Falsch

## Größer als [>]

Vergleicht zwei Argumente (Operanden).

Gibt Wahr zurück, wenn der erste Wert größer ist als der zweite, ansonsten Falsch.

### Syntax

```
[>]
```

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen oder Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Wahr/Falsch

### Ausdruck

Ermitteln, ob der Kaufpreis des Elements (eine vorhandene Eigenschaft des Typs Zahl) größer als 30 ist.

### Ausdruck

```
Kaufpreis > 30
```

### Ergebnis

Wahr für Elemente, deren Preis über 30 liegt, ansonsten Falsch

## Größer oder Gleich [>=]

Vergleicht zwei Argumente (Operanden). Gibt Wahr zurück, wenn der erste Wert größer oder gleich ist wie der zweite, ansonsten Falsch.

### Syntax

```
[>=]
```

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen oder Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Wahr/Falsch

### Beispiel

Prüfen, ob der tatsächliche Fensteranteil von Räumen (eine vorhandene Eigenschaft) dem vorgeschriebenen Fensteranteil entspricht (einem festen Wert, hier  $\geq 0,40$ ).

### Ausdruck

```
Raumfläche Fensteranteil  $\geq$  0,40
```

### Ergebnis

Wahr für Räume, deren Fensteranteil mindestens 40 Prozent ist, ansonsten Falsch

## Kleiner als [<]

Vergleicht zwei Argumente (Operanden). Gibt Wahr zurück, wenn der erste Wert kleiner ist als der zweite, ansonsten Falsch.

### Syntax

```
[<]
```

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen oder Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Wahr/Falsch

### Beispiel

Ermitteln, ob der Kaufpreis des Elements (eine vorhandene Eigenschaft des Typs Zahl) kleiner als 30 ist.

### Ausdruck

```
Kaufpreis < 30
```

### Ergebnis

Wahr für Elemente, deren Preis unter 30 liegt, ansonsten Falsch

## Kleiner oder Gleich [ $\leq$ ]

Vergleicht zwei Argumente (Operanden). Gibt Wahr zurück, wenn der erste Wert kleiner oder gleich ist wie der zweite, ansonsten Wahr.

### Syntax

```
[ $\leq$ ]
```

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen oder Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Wahr/Falsch

### Beispiel

Ermitteln, ob die Fußbodenstärke eines Raums (eine vorhandene Längen-Eigenschaft)  $\leq$  20 cm ist.

### Ausdruck

```
Fußbodenstärke  $\leq$  20 cm
```

### Ergebnis

Wahr für Raumflächen mit Fußbodenstärke gleich oder weniger als 20 cm, ansonsten Falsch

### Anmerkungen

Beide Argumente müssen den gleichen Datentyp haben - hier haben beide den Typ "Länge".

Die Längenwerte müssen eine Einheit haben. Wählen Sie eine Längeneinheit aus dem Einheiten-Popup-Menü des Berechnung-Editors aus.

## Multiplizieren [\*]

Multipliziert zwei Argumente (Operanden).

### Syntax

```
[ * ]
```

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen oder Winkel

### Datentyp des Ergebnisses

Hängt von den Datentypen der Argumente ab

### Beispiel

Ermittelt die Fläche einer gegebenen Fliese anhand ihrer festen Eigenschaftswerte mit Längen-Datentypen: horizontales Maß (20 cm) und vertikales Maß (30 cm).

### Ausdruck

```
Fliesen-Bemaßung H * Fliesen-Bemaßung V
```

### Ergebnis

600

### Anmerkungen

In diesem Beispiel haben beide Argumente den Typ Länge. Multiplizieren Sie sie, um eine Fläche zu erhalten: Stellen Sie den resultierenden Eigenschaften-Datentyp auf Fläche ein.

## Nicht gleich [<>]

Vergleicht zwei Argumente (Operanden). Gibt Wahr zurück, wenn ihre Werte nicht gleich sind, ansonsten Falsch.

### Syntax

```
[<>]
```

### Datentypen der Argumente

Wahr/Falsch, Ganzzahl, Zahl, Länge, Fläche, Volumen, Winkel oder String (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Wahr/Falsch

### Beispiel

Raumkategorien umfassen z. B. Büro, Flur, Besprechungsraum und Raucherbereich.

Rauchmelder sind in allen Räumen erforderlich außer in denen, die als "Raucherbereich" klassifiziert sind.

Bestimmen Sie für jeden Raum, ob seine Klassifizierung nicht "Raucherbereich" lautet.

In diesem Ausdruck handelt es sich bei beiden Argumenten um Strings.

### Ausdruck

```
Archicad Klassifizierung 25 <> "Raucherbereich"
```

### Ergebnis

Wahr für alle Räume, die nicht Raucherbereiche sind. (Rauchmelder sind in allen diesen Bereichen erforderlich.)

## Hoch [^]

### Syntax

```
[^]
```

oder verwenden Sie die Funktion `Hoch [POWER]`.

### Datentypen der Argumente

Basis: Ganzzahl, Zahl, Länge, Fläche, Volumen, oder Winkel

Exponent: Ganzzahl oder Zahl

### Datentyp des Ergebnisses

Hängt vom Datentyp der Argumente ab. (Siehe Anmerkung, unten).

### Ausdruck

```
5^2
```

### Ergebnis

25

### Anmerkung

Wenn der Datentyp des Basisarguments Länge ist, dann

- wird ein Exponent von 2 eine Eigenschaft zurückgeben, deren Datentyp Fläche sein muss.
- wird ein Exponent von 3 eine Eigenschaft zurückgeben, deren Datentyp Volumen sein muss.



## Subtrahieren [-]

### Syntax

```
[-]
```

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen oder Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Wie Datentyp des Arguments

### Beispiel

Subtrahieren eines Längen-Ausdrucks von einem anderen Längen-Ausdruck.

Das Ergebnis ist ebenfalls eine Länge (ihre Einheit hängt von den Berechnungseinheiten für die Länge ab: hier Meter).

### Ausdruck

```
Höhe Abgehängte Decke - Höhe Doppelboden
```

### Ergebnis

2.2

### Anmerkung

In diesem Beispiel haben beide Argumente den Typ Länge. Der resultierende Wert ist ebenfalls eine Länge.

# LOGISCHE FUNKTIONEN

## Und [AND]

Prüft, ob jedes Element in einer Reihe von Bedingungen (logische Argumente) wahr ist.

Wenn *alle* Argumente wahr sind, liefert AND die Rückmeldung WAHR.

Wenn *eines oder mehrere* der Argumente falsch sind, liefert AND die Rückmeldung FALSCH.

### Syntax

```
AND (Logikelement1|[Logikelement2],...)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Wahr/Falsch

### Datentyp des Ergebnisses

Wahr/Falsch

### Beispiel

Bestimmen, ob beide Träger-Längeneigenschaften (Länge Links und Länge Rechts) eine bestimmte Länge überschreiten.

### Ausdruck

```
AND (Länge Links > 480 cm; Länge Rechts > 480 cm)
```

### Ergebnis

Wahr oder Falsch. Wahr für alle Träger, bei denen beide Träger-Längen den Wert überschreiten, ansonsten Falsch.

### Anmerkung

Die Funktion AND untersucht eine Reihe von Argumenten. Innerhalb der Klammern muss jedes Element ein logisches Argument sein, das entweder wahr oder falsch sein kann.

## Oder [OR]

Prüft, ob mindestens eines der Elemente in einer Reihe von Bedingungen (logische Argumente) wahr ist.

Wenn *mindestens eines* der Argumente wahr ist, liefert OR die Rückmeldung WAHR.

Wenn *keines* der Argumente wahr ist, liefert OR die Rückmeldung FALSCH.

### Syntax

```
OR (Logikelement1|[Logikelement2],...)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Wahr/Falsch

### Datentyp des Ergebnisses

Wahr/Falsch

### Beispiel 1

Bestimmen, ob eine der Träger-Längeneigenschaften (Länge Links und Länge Rechts) eine bestimmte Länge überschreitet.

### Ausdruck

```
OR (Länge Links > 480 cm; Länge Rechts > 480 cm)
```

### Ergebnis

Wahr oder Falsch. Wahr für alle Träger, bei denen mindestens eine Träger-Länge den Wert überschreitet, ansonsten Falsch.

### Beispiel 2

Stellen Sie fest, ob eine Wand feuerfest ist. Um feuerfest zu sein, muss die Wand *mindestens eine* von zwei möglichen Bedingungen erfüllen: Ihr Baustoff ist "Beton, Stahlbeton C20/25", und/oder sie ist mindestens 50 cm dick.

### Ausdruck

```
OR (Baustoff = „Beton, Stahlbeton C20/25“, Dicke >= 50 cm)
```

### Ergebnis

Wahr oder Falsch. Wahr für alle Wände, deren Baustoff "Beton, Stahlbeton C20/25" ist und/oder deren Dicke mindestens 50 cm beträgt.

### Anmerkung

Die Funktion OR untersucht eine Reihe von Argumenten. Innerhalb der Klammern muss jedes Element ein logisches Argument sein, das entweder wahr oder falsch sein kann.

## Nicht [NOT]

Ändert Falsches Ergebnis in Wahres und umgekehrt.

### Syntax

```
NOT (Logikelement)
```

### Datentypen der Argumente

Wahr/Falsch

### Datentyp des Ergebnisses

Wahr/Falsch

### Beispiel

Eine vorhandene Eigenschaft gibt "Wahr" zurück für alle Elemente, deren Modelleigenschaft nicht definiert ist. Zum Ändern dieser Ergebnisse in Falsch verwenden Sie NOT.

### Ausdruck

```
NOT (ISUNDEFINED (Modell))
```

### Ergebnis

Falsch für jedes Element, dessen Modelleigenschaft <Nicht definiert> ist

## Wenn [IF]

Gibt einen definierten Wert zurück, wenn das gegebene logische Argument wahr ist. und einen anderen definierten Wert, wenn das Argument falsch ist.

### Syntax

```
IF (Logikelement, WertWennWahr, WertWennFalsch)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Logikelement: Wahr/Falsch

ValuelfTrue, ValuelfFalse: Beliebiger Datentyp, aber der Datentyp muss für beide Werte identisch sein

### Datentyp des Ergebnisses

Gleicher Datentyp wie für Werte.

### Beispiel

Definieren Sie eine Regel für den Träger-Überstand abhängig von der Träger-Länge.

Wenn der Träger kürzer als 480 cm ist, sollte der Überstand 10 cm betragen.

Wenn er länger als 480 cm ist, sollte der Überstand 12 cm betragen.

### Ausdruck

```
IF (Länge Links <=480cm, 10 cm, 12 cm)
```

### Ergebnis

10 für kürzere Träger, 12 für längere Träger

### Anmerkung

Um mehrere Bedingungen zu testen, ist es eventuell einfacher, die Funktion `IFS` zu verwenden.

Siehe [If Multiple \[IFS\]](#).

## If Multiple [IFS]

Verwenden Sie diese Funktion, um eine Reihe von logischen Bedingungen zu testen und geben Sie dann den Wert für die erste wahre Bedingung zurück.

IFS kann an die Stelle von mehreren verschachtelten IF-Anweisungen treten.

### Syntax

```
IFS (Logisch1, Wert1, [Logisch2, Wert2]...)
```

Sie können bis zu 127 logische Bedingungen definieren.

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Logikelement: Wahr/Falsch

Wert: Jeder Datentyp, aber Datentyp und Einheiten müssen für alle Werte identisch sein.

### Datentyp des Ergebnisses

Gleicher Datentyp wie für Werte.

### Beispiel 1

Geben Sie den vorgeschriebenen Prozentsatz der verglasten Fläche für jeden Raum in Abhängigkeit von seiner Klassifizierung an.

### Ausdruck

```
IFS (Archicad Klassifizierung - 25 = „Wohnfläche“, „25%“, Archicad
Klassifizierung - 25 = „Bürofläche“, „30%“,
Archicad Klassifizierung - 25 = „Lagerfläche“, „10%“)
```

### Ergebnis

Liefert den vorgeschriebenen Prozentsatz der verglasten Fläche für jeden Raum in Abhängigkeit von seiner Klassifizierung. Die Werte und die Ergebnisse haben alle den gleichen Datentyp (hier: Zeichenfolge).

### Wenn keine der Bedingungen wahr ist

Wenn *keine* der Bedingungen zutrifft, gibt die Funktion "Undefiniert" zurück.

Sie können aber auch einen Standardergebniswert angeben, wenn keine der logischen Bedingungen erfüllt ist: Geben Sie für Ihre letzte logische Bedingung TRUE ein, gefolgt vom Standardwert.

Beachten Sie, dass dieser Standardergebniswert den gleichen Datentyp haben muss wie die anderen Ergebniswerte in der Funktion.

```
IFS (Logisch1, Wert1, [Logisch2, Wert2],...[TRUE,Standardwert])
```

### Beispiel 2

Kategorisieren Sie Träger als lang (länger als 6 m) oder kurz (kürzer als 2 m). Jeder Träger, der nicht in diese Kategorien fällt, ist Mittel.

**Ausdruck**

```
IFS (Länge Links > 600 cm, „Lang“, Länge Links < 200 cm, „Kurz“, TRUE,  
„Mittel“)
```

**Ergebnis**

Träger mit einer Länge von mehr als 6 m: Lang

Träger kürzer als 2 m: Kurz

Träger mit dazwischen liegender Länge: Mittel

## Exklusiv Oder [XOR]

Gibt Wahr zurück, wenn eine ungerade Anzahl wahrer logischer Argumente vorliegt.

Bei zwei Argumenten gilt: eines der beiden Argumente muss wahr sein, es dürfen aber nicht beide wahr sein.

### Syntax

```
XOR (Logikelement1, [Logikelement2],...)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Wahr/Falsch

### Datentyp des Ergebnisses

Wahr/Falsch

### Beispiel

Büro-Dekorelemente verschiedener Größe müssen verpackt werden. Es sind drei Kistengrößen verfügbar.

Elemente mit einer Höhe und Breite von jeweils mehr als 1 Meter werden in großen Kisten verpackt.

```
AND (Breite > 1m, Höhe > 1m)
```

Elemente mit einer Höhe und Breite von jeweils weniger als 1 Meter werden in kleinen Kisten verpackt.

```
AND (Breite < 1m, Höhe < 1m)
```

Elemente, deren **Höhe oder Breite** (aber nicht beide) über 1 Meter betragen, werden in mittleren Kisten verpackt.

Bestimmen Sie mit einer Eigenschaft, welche Elemente eine mittlere Kiste benötigen.

### Ausdruck

```
XOR (Breite > 1m, Höhe > 1m)
```

### Ergebnis

Wahr für alle Elemente mit einer Breite ODER Höhe (aber nicht beides) über 1 Meter.



## Nicht definiert [ISUNDEFINED]

Prüft, ob der Wert einer bestimmten Eigenschaft undefiniert ist.

### Syntax

```
ISUNDEFINED (Wert)
```

### Datentypen des Arguments

Wahr/Falsch, Ganzzahl, Zahl, Länge Fläche, Volumen, Winkel oder String

### Datentyp des Ergebnisses

Wahr/Falsch

### Beispiel

Bestimmen, ob die String-Eigenschaft "Modell" eines Elements undefiniert ist. `ISUNDEFINED` ist ein logischer Wert.

### Ausdruck

```
ISUNDEFINED (Modell)
```

### Ergebnis

Wahr für jedes Element, dessen Modelleigenschaft <Nicht definiert> ist

# MATHEMATISCHE FUNKTIONEN

## Absoluter Wert [ABS]

Absoluter Wert.

Gibt den absoluten Wert der Zahl zurück.

### Syntax

```
ABS (Zahl)
```

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen, Winkel

### Datentyp des Ergebnisses

Wie Datentyp des Arguments

### Ausdruck

```
ABS (-4)
```

### Ergebnis

4

## e Exponent [EXP]

Exponentielle Funktion von e.

Gibt "e" hoch der angegebenen Zahl zurück.

Die Basis "e" ist eine Konstante - die Basis des natürlichen Logarithmus.

### Syntax

```
EXP (Zahl)
```

### Datentyp der Argumente

Zahl, Ganzzahl

### Datentyp des Ergebnisses

Zahl

### Ausdruck

```
EXP (2)
```

### Ergebnis

7.39

### Anmerkungen

Der Wert von e beträgt 2,71828183.

EXP ist die Umkehrung von `Natürlicher Logarithmus [LN]` (dem natürlichen Logarithmus der Zahl).

Zum Berechnen der exponentiellen Funktion einer anderen Basis verwenden Sie den Operator `^`.

Siehe [Hoch \[^\]](#).

## Natürlicher Logarithmus [LN]

Gibt den natürlichen Logarithmus der angegebenen Zahl zurück.

### Syntax

```
LN (Zahl)
```

### Datentypen des Arguments

Zahl, Ganzzahl

### Datentyp des Ergebnisses

Zahl

### Ausdruck

```
LN (86)
```

### Ergebnis

4.45

### Anmerkungen

`LN` ist die Umkehrung der Funktion `e Exponent [EXP]`.

Diese Funktion funktioniert mit positiven Zahlen.

## Basislogarithmus [LOG]

Gibt den Logarithmus zur Basis der angegebenen Zahl zurück.

### Syntax

```
LOG (Zahl, Basis)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Zahl, Ganzzahl

### Datentyp des Ergebnisses

Zahl

### Ausdruck

```
LOG (8,2)
```

### Ergebnis

3

### Anmerkungen

Diese Funktion funktioniert mit positiven Zahlen.

## Dezimaler Logarithmus [LOG10]

Gibt den Logarithmus zur Basis 10 der angegebenen Zahl zurück.

### Syntax

```
LOG10 (Zahl)
```

### Datentypen der Argumente

Zahl, Ganzzahl

### Datentyp des Ergebnisses

Zahl

### Ausdruck

```
LOG10 (10)
```

### Ergebnis

1

### Anmerkungen

Diese Funktion funktioniert mit positiven Zahlen.

## Rest [MOD]

Modulo-Operation. Ermittelt den Rest nach einer Division der Zahl durch den Divisor.

### Syntax

```
MOD (Zahl, Divisor)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Zahl, Ganzzahl

### Datentyp des Ergebnisses

Zahl

### Ausdruck

```
MOD (3, 2)
```

### Ergebnis

1

### Anmerkungen

Zum Berechnen des Quotienten (ohne den Rest) verwenden Sie `Quotient [QUOTIENT]`.

*Siehe auch [Dividieren \[1\]](#).*

## Nte Wurzel [NTHROOT]

Berechnet die n-te Wurzel des angegebenen Argumentes (bei 'Grad=3' wird beispielsweise die dritte Wurzel berechnet).

### Syntax

```
NTHROOT (Zahl, Grad)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Zahl: Zahl, Ganzzahl oder ein anderer mit der Wurzel-Funktion kompatibler Datentyp

Grad: Zahl, Ganzzahl

### Datentyp des Ergebnisses

Hängt von den Datentypen der Argumente ab

### Beispiel

Berechnen der dritten Wurzel der vorgegebenen Zahl.

### Ausdruck

```
NTHROOT (125, 3)
```

### Ergebnis

5

### Anmerkungen

Für Quadratwurzel: [Siehe Quadratwurzel \[SQRT\]](#).



## Hoch [POWER]

Gibt die angegebene Zahl hoch dem angegebenen Exponenten an.

### Syntax

```
POWER (Basis, Exponent)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Basis: Ganzzahl, Zahl, Länge, Fläche, Volumen, Winkel

Exponent: Zahl, Ganzzahl

### Datentyp des Ergebnisses

Hängt von den Datentypen der Argumente ab

### Ausdruck

```
POWER (5,2)
```

### Ergebnis

25

### Anmerkungen

Funktioniert wie Hoch [^] (Operator).

## Quotient [QUOTIENT]

Gibt den ganzzahligen Anteil einer Division zurück.

### Syntax

```
QUOTIENT (Zähler, Nenner)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen oder Winkel (muss für beide Argumente identisch sein)

### Datentyp des Ergebnisses

Zahl

### Beispiel

Division der Eigenschaft Oberfläche Decken-Oberseite durch die Eigenschaft Fliesenfläche.

Die Quotientenfunktion ermittelt nur den ganzzahligen Anteil (Integer) der resultierenden Zahl.

### Ausdruck

```
QUOTIENT (Oberfläche Oberseite; Fliesenfläche)
```

### Ergebnis

1974

Beide Eigenschaften sind Flächendaten. Ihre Division liefert eine Zahl.

### Anmerkungen

Zum Ermitteln des Rests (ohne den Quotienten) verwenden Sie `Rest [MOD]`.

[Siehe auch Dividieren \[1\].](#)

## Vorzeichen [SIGN]

Gibt +1, 0 oder -1 zurück.

Wenn die Zahl positiv ist, lautet das Ergebnis +1.

Wenn die Zahl Null ist, lautet das Ergebnis 0.

Wenn die Zahl negativ ist, lautet das Ergebnis -1.

### Syntax

```
SIGN (Zahl)
```

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen, Winkel

### Datentyp des Ergebnisses

Zahl

### Beispiel

Für ein Profilelement mit einem bearbeitbaren Basisbreiten-Abstands-Modifikator:

Bestimmen, ob sein Basisbreiten-Wert positiv oder negativ ist.

### Ausdruck

```
SIGN (Basisbreite)
```

### Ergebnis

-1 für eine negative Basisbreite

1 für eine positive Basisbreite

0 für eine Basisbreite von Null.

## Quadratwurzel [SQRT]

Quadratwurzel.

Berechnet die Quadratwurzel der gegebenen Zahl.

### Syntax

```
SQRT (Zahl)
```

### Datentypen des Arguments

Ganzzahl, Zahl, Fläche

### Datentyp des Ergebnisses

Hängt vom Datentyp des Arguments ab

### Ausdruck

```
SQRT (16)
```

### Ergebnis

4

### Anmerkungen

Zum Ziehen anderer Wurzeln: [Siehe Nte Wurzel \[NTHROOT\]](#).

## Summe [SUM]

Berechnet die Summe der angegebenen Argumente.

### Syntax

```
SUM ((Zahl1; [Zahl2],...))
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen, Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Hängt vom Datentyp der Argumente ab

### Beispiel

Addieren der Ergebnisse zweier Eigenschaften.

### Ausdruck

```
SUM (Höhe Doppelboden; Dicke Unterboden)
```

### Ergebnis

.30

Beide Argumente haben den Datentyp Länge. Das addierte Ergebnis hat ebenfalls den Datentyp Länge.

### Anmerkungen

*Siehe auch [Addieren \[+\]](#).*

## Runden auf Vielfaches [MROUND]

Rundet die angegebene Zahl auf das gewünschte Vielfache.

### Syntax

```
MROUND (Zahl, Vielfaches)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen, Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Hängt vom Datentyp der Argumente ab

### Ausdruck

```
MROUND (2567, 10)
```

### Ergebnis

2670

### Anmerkungen

Wenn der Rest größer oder gleich der Hälfte des Vielfachen ist, rundet MROUND auf.

Wenn immer auf ein bestimmtes Vielfaches aufgerundet werden soll, verwenden Sie `Aufrunden auf Vielfaches [MROUNDUP]`.

Wenn immer auf ein bestimmtes Vielfaches abgerundet werden soll, verwenden Sie `Abrunden auf Vielfaches [MROUNDDOWN]`.

Zum Runden der Zahlen auf eine bestimmte Dezimalstelle verwenden Sie `Runden [ROUND]`, `Aufrunden [ROUNDUP]`, `Abrunden [ROUNDDOWN]`.

## Abrunden auf Vielfaches [MROUNDDOWN]

Rundet die angegebene Zahl auf das gewünschte Vielfache ab.

### Syntax

```
MROUNDDOWN (Zahl, Vielfaches)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen, Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Hängt vom Datentyp der Argumente ab

### Beispiel

Die Eigenschaft "Trägerlänge" berechnet die genaue Länge der vorhandenen Träger-Elemente in dem Modell (z. B. 2.044).

Die Eigenschaft "Vorfab-Trägerlänge" ist ein bestimmter Wert (z. B. 300)

Erstellen Sie einen Ausdruck, um die Trägerlänge auf die Vorfab-Schrittweite abzurunden.

### Ausdruck

```
MROUNDDOWN ("Trägerlänge", 300)
```

### Ergebnis

1.800 (für eine Probe-Trägerlänge von 2.044)

### Anmerkungen

*Siehe auch:* [Runden auf Vielfaches \[MROUND\]](#), [Aufrunden auf Vielfaches \[MROUNDUP\]](#).

Zum Runden der Zahlen auf eine bestimmte Dezimalstelle verwenden Sie [Runden \[ROUND\]](#), [Aufrunden \[ROUNDUP\]](#), [Abrunden \[ROUNDDOWN\]](#).

## Aufrunden auf Vielfaches [MROUNDUP]

Rundet die angegebene Zahl auf das gewünschte Vielfache auf.

### Syntax

```
MROUNDUP (Zahl, Vielfaches)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen, Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Hängt vom Datentyp der Argumente ab

### Beispiel

Die Eigenschaft "Trägerlänge" berechnet die genaue Länge der vorhandenen Träger-Elemente in dem Modell (z. B. 2.044).

Die Eigenschaft "Vorfab-Trägerlänge" ist ein bestimmter Wert (z. B. 300)

Erstellen Sie einen Ausdruck, um die Trägerlänge auf die Vorfab-Schrittweite aufzurunden.

### Ausdruck

```
MROUNDUP ("Trägerlänge", 300)
```

### Ergebnis

2.100 (für eine Probe-Trägerlänge von 2.044)

### Anmerkungen

*Siehe auch:* [Runden auf Vielfaches \[MROUND\]](#), [Abrunden auf Vielfaches \[MROUNDDOWN\]](#).

Zum Runden der Zahlen auf eine bestimmte Dezimalstelle verwenden Sie [Runden \[ROUND\]](#), [Aufrunden \[ROUNDUP\]](#), [Abrunden \[ROUNDDOWN\]](#).



## Runden [ROUND]

Rundet eine Zahl auf eine angegebene Anzahl von Stellen.

### Syntax

```
ROUND (Zahl, AnzahlStellen)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Zahl: Ganzzahl, Zahl

Anzahl der Stellen: Ganzzahl

Wenn das Argument Anzahl der Stellen positiv ist, wird auf die angegebene Anzahl von Dezimalstellen gerundet. [Siehe Ausdruck 1 unten.](#)

Wenn das Argument Anzahl der Stellen Null ist, wird die Zahl auf die nächste Ganzzahl gerundet. [Siehe Ausdruck 2 unten.](#)

Wenn das Argument Anzahl der Stellen negativ ist, wird nur links von der Dezimalstelle gerundet (Dezimalstellen werden ignoriert). [Siehe Ausdruck 3 unten.](#)

### Datentyp des Ergebnisses

Zahl

Die im Ergebnis angezeigte Anzahl von Dezimalstellen hängt von der Festlegung in Optionen > Projektpräferenzen > Arbeitseinheiten ab.

### Beispiele

Ausdruck 1

```
ROUND (46.9855, 2)
```

Ergebnis

```
46.99
```

Ausdruck 2

```
ROUND (46.9855, 0)
```

Ergebnis

```
47.00
```

Ausdruck 3

```
ROUND (46.9855, -1)
```

Ergebnis

```
50.00
```

### Anmerkungen

Wenn immer aufgerundet werden soll, verwenden Sie Aufrunden [ROUNDUP]. Wenn immer abgerundet werden soll, verwenden Sie Abrunden [ROUNDDOWN].

Zum Runden auf ein bestimmtes Vielfaches verwenden Sie Runden auf Vielfaches [MROUND] (oder Aufrunden auf Vielfaches [MROUNDUP] oder Abrunden auf Vielfaches [MROUNDDOWN]).

## Abrunden [ROUNDDOWN]

Rundet eine Zahl auf eine angegebene Anzahl von Stellen ab.

Wenn das Argument Anzahl der Stellen positiv ist, wird auf die angegebene Anzahl von Dezimalstellen abgerundet.

Wenn das Argument Anzahl der Stellen Null ist, wird die Zahl auf die nächste Ganzzahl abgerundet.

Wenn das Argument Anzahl der Stellen negativ ist, wird auf die angegebene Anzahl von Stellen links vom Dezimalzeichen abgerundet.

### Syntax

```
ROUNDDOWN (Zahl, AnzahlStellen)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Zahl: Ganzzahl, Zahl

Anzahl der Stellen: Ganzzahl

### Datentyp des Ergebnisses

Zahl

### Beispiel

Ausdruck 1

```
ROUNDDOWN (46.9855, 2)
```

Ergebnis

```
46.98
```

Ausdruck 2

```
ROUNDDOWN (46.9855, 0)
```

Ergebnis

```
46.00
```

Ausdruck 3

```
ROUNDDOWN (46.9855, -1)
```

Ergebnis

```
40.00
```

### Anmerkungen

*Siehe auch:* [Runden \[ROUND\]](#), [Aufrunden \[ROUNDUP\]](#).

Zum Runden auf ein bestimmtes Vielfaches verwenden Sie Runden auf Vielfaches [MROUND] (oder Aufrunden auf Vielfaches [MROUNDUP] oder Abrunden auf Vielfaches [MROUNDDOWN]).

## Aufrunden [ROUNDUP]

Rundet eine Zahl auf eine angegebene Anzahl von Stellen.

Wenn das Argument Anzahl der Stellen positiv ist, wird auf die angegebene Anzahl von Dezimalstellen aufgerundet.

Wenn das Argument Anzahl der Stellen Null ist, wird die Zahl auf die nächste Ganzzahl aufgerundet.

Wenn das Argument Anzahl der Stellen negativ ist, wird auf die angegebene Anzahl von Stellen links vom Dezimalzeichen aufgerundet.

### Syntax

```
ROUNDUP (Zahl, AnzahlStellen)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Zahl: Ganzzahl, Zahl

Anzahl der Stellen: Ganzzahl

### Datentyp des Ergebnisses

Zahl

### Beispiel

Die Eigenschaft "Anzahl der Fliesen" teilt die Oberfläche Decken-Oberseite durch die Fläche einer einzelnen Fliese. Das Ergebnis ist: 371,35.

Erstellen einer Eigenschaft, die dieses Ergebnis auf eine Ganzzahl aufrundet.

### Ausdruck

```
ROUNDUP (Anzahl der Fliesen, 0)
```

### Ergebnis

372 (wobei das Ergebnis für die Anzahl der Fliesen 371,35 war)

### Anmerkungen

*Siehe auch:* [Runden \[ROUND\]](#), [Abrunden \[ROUNDDOWN\]](#).

Zum Runden auf ein bestimmtes Vielfaches verwenden Sie `Runden auf Vielfaches [MROUND]` (oder `Aufrunden auf Vielfaches [MROUNDUP]` oder `Abrunden auf Vielfaches [MROUNDDOWN]`).

# TRIGONOMETRISCHE FUNKTIONEN

## Arkuskosinus [ACOS]

Gibt den Arkuskosinus einer Zahl zurück.

### Syntax

```
ACOS (Zahl)
```

### Datentyp des Arguments

Zahl, Ganzzahl. Muss zwischen -1 und 1 liegen.

### Datentyp des Ergebnisses

Winkel

### Ausdruck

```
ACOS (-0.5)
```

### Ergebnis

120 Grad

### Anmerkungen

Die Winkeleinheit des Ergebnisses (z. B. Grad, Radiant, Gon) hängt davon ab, was Sie in den Projektpräferenzen - Berechnungseinheiten - für die Winkeleinheit angegeben haben.

## Arkuskotangens [ACOT]

Gibt den Arkuskotangens einer Zahl zurück.

### Syntax

```
ACOT (Zahl)
```

### Datentyp des Arguments

Zahl, Ganzzahl

### Datentyp des Ergebnisses

Winkel

### Ausdruck

```
ACOT (2)
```

### Ergebnis

27 Grad

### Anmerkungen

Die Winkeleinheit des Ergebnisses (z. B. Grad, Radiant, Gon) hängt davon ab, was Sie in den Projektpräferenzen - Berechnungseinheiten - für die Winkeleinheit angegeben haben.

## Arkussinus [ASIN]

Gibt den Arkussinus eines Winkels zurück.

### Syntax

```
ASIN (Zahl)
```

### Datentyp des Arguments

Zahl, Ganzzahl. Muss zwischen -1 und 1 liegen.

### Datentyp des Ergebnisses

Winkel

### Ausdruck

```
ASIN (-0.5)
```

### Ergebnis

-30 Grad

### Anmerkungen

Die Winkeleinheit des Ergebnisses (z. B. Grad, Radiant, Gon) hängt davon ab, was Sie in den Projektpräferenzen für die Winkeleinheit angegeben haben.

## Arkustangens [ATAN]

Gibt den Arkustangens einer Zahl zurück.

### Syntax

```
ATAN (Zahl)
```

### Datentypen der Argumente

Zahl, Ganzzahl

### Datentyp des Ergebnisses

Winkel

### Ausdruck

```
ATAN (1)
```

### Ergebnis

45 Grad

### Anmerkungen

Die Winkeleinheit des Ergebnisses (z. B. Grad, Radiant, Gon) hängt davon ab, was Sie in den Projektpräferenzen für die Winkeleinheit angegeben haben.

## Kosinus [COS]

Gibt den Kosinus eines Winkels zurück.

### Syntax

```
COS (Winkel)
```

### Datentyp des Arguments

Winkel

### Datentyp des Ergebnisses

Zahl

### Ausdruck

```
COS (60 D)
```

### Ergebnis

0.5

### Anmerkungen

Der Winkelwert des Arguments muss als Einheit (Grad, Radiant, Gon) angegeben werden. Wählen Sie eine Winkeleinheit aus dem Einheiten-Popup-Menü des Berechnung-Editors aus.



## Kotangens [COT]

Gibt den Kotangens eines Winkels zurück

### Syntax

```
COT (Winkel)
```

### Datentypen der Argumente

Winkel

### Datentyp des Ergebnisses

Zahl

### Ausdruck

```
COT (30D)
```

### Ergebnis

1.73

### Anmerkungen

Der Winkelwert des Arguments muss als Einheit (Grad, Radiant, Gon) angegeben werden.

Wählen Sie eine Winkeleinheit aus dem Einheiten-Popup-Menü des Berechnung-Editors aus. Hier lautet die Angabe Grad (D).

## Sinus [SIN]

Gibt den Sinus eines Winkels zurück.

### Syntax

```
SIN (Winkel)
```

### Datentyp des Arguments

Winkel

### Datentyp des Ergebnisses

Zahl

### Ausdruck

```
SIN (30D)
```

### Ergebnis

0.50

### Anmerkungen

Der Winkelwert des Arguments muss als Einheit (Grad, Radian, Gon) angegeben werden.

Wählen Sie eine Winkleinheit aus dem Einheiten-Popup-Menü des Berechnung-Editors aus. Hier lautet die Angabe Grad (D).

## Tangens [TAN]

Gibt den Tangens eines Winkels zurück.

### Syntax

```
TAN (Winkel)
```

### Datentypen der Argumente

Winkel

### Datentyp des Ergebnisses

Zahl

### Ausdruck

```
TAN (0.785 R)
```

### Ergebnis

1.00

### Anmerkungen

Der Winkelwert des Arguments muss als Einheit (Grad, Radiant, Gon) angegeben werden.

Wählen Sie eine Winkleinheit aus dem Einheiten-Popup-Menü des Berechnung-Editors aus. Hier lautet die Angabe Radiant (R).

# STATISTISCHE FUNKTIONEN

## Durchschnitt [AVG]

Berechnet den Durchschnitt der angegebenen Argumente.

### Syntax

```
AVG (Zahl1; [Zahl2], ...)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen oder Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Wie Datentyp des Arguments

### Beispiel

Stützen werden von einem geneigten Dach geschnitten, sodass jede Stütze eine maximale und eine minimale Höhe hat.

Verwenden Sie für jede Stütze AVG, um ihre durchschnittliche Höhe zu ermitteln.

### Ausdruck

```
AVG (Maximale Höhe der Stütze; Minimale Höhe der Stütze)
```

### Ergebnis

3,08 (für eine gegebene Stütze mit einer maximalen Höhe von 3,11 und einer minimalen Höhe von 3,04)

### Anmerkung

Beide Argumente haben den Datentyp Länge, sodass der Datentyp der Eigenschaft ebenfalls auf Länge eingestellt sein muss.

## Maximum [MAX]

Berechnet das Maximum der angegebenen Argumente.

### Syntax

```
MAX (Zahl1; [Zahl2]; ...)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen oder Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Wie Datentyp des Arguments

### Beispiel

Ein Träger schneidet einen anderen Träger und hat damit an den Seiten unterschiedliche Längen.

Verwenden Sie `MAX`, um die größere der beiden Längen zu ermitteln.

### Ausdruck

```
MAX (Länge Links, Länge Rechts)
```

### Ergebnis

3,86 (Für einen gegebenen Träger mit den Seitenlängen 3,86 und 3,56)

## Minimum [MIN]

Berechnet das Minimum der angegebenen Argumente.

### Syntax

```
MIN (Zahl1; [Zahl2]; ...)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Ganzzahl, Zahl, Länge, Fläche, Volumen oder Winkel (muss für alle Argumente identisch sein)

### Datentyp des Ergebnisses

Wie Datentyp des Arguments

### Beispiel

Ein Träger schneidet einen anderen Träger und hat damit an den Seiten unterschiedliche Längen. (3.86 und 3.56).

Verwenden Sie `MIN`, um die kürzere der beiden Längen zu ermitteln.

### Ausdruck

```
MIN (Länge Links, Länge Rechts)
```

### Ergebnis

3,56 (Für einen bestimmten Träger mit den Seitenlängen 3,86 und 3,56)

# TEXTBEARBEITUNGSFUNKTIONEN

## Zeichenfolgen vergleichen [COMPARE]

Vergleicht zwei Strings.

Gibt 0 zurück, wenn sie gleich sind

Gibt -1 zurück, wenn Text1 alphabetisch vor Text2 liegt

Gibt +1 zurück, wenn Text1 alphabetisch hinter Text2 liegt.

Bei der alphabetischen Reihenfolge wird zwischen Groß- und Kleinschreibung unterschieden: a kommt vor A.

### Syntax

```
COMPARE ("Text1"; "Text2")
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Zeichenfolge

### Datentyp des Ergebnisses

Ganzzahl

### Beispiel

Angenommen, die Brandschutz-Eigenschaft eines Elements kann einen Wert von A bis F haben.

Ein Brandschutz A, B oder C ist mit dem Standard konform.

Ein Brandschutz D, E oder F ist nicht mit dem Standard konform.

Verwenden Sie in diesem Ausdruck COMPARE, um die Brandschutz-Kategorie des Elements auf "C" zu prüfen.

Verwenden Sie darüber hinaus eine IF-Funktion, um einen der beiden Strings auf der Basis des Ergebnisses der COMPARE-Funktion zurückzugeben.

### Ausdruck

```
IF (COMPARE (Brandschutz, "C") < 1; "Standard-konform"; "Nicht-konform")
```

Wenn das Ergebnis 0 ist, so bedeutet dies, dass der Brandschutz C entspricht. Dies ist Standard-konform.

Wenn das Ergebnis -1 ist, so bedeutet dies, dass der Brandschutz alphabetisch vor C liegt: also A oder B. Dies ist ebenfalls Standard-konform.

Wenn das Ergebnis 1 ist, so bedeutet dies, dass der Brandschutz alphabetisch nach C liegt: also D, E oder F. Dies ist nicht Standard-konform.

### Ergebnis

"Standard-konform" für Werte von 0 oder -1, ansonsten "Nicht-konform"

## Zeichenfolgen verschmelzen [CONCAT]

Verbindet zwei oder mehr Zeichenfolgen (Strings) zu einer einzigen.

### Syntax

```
CONCAT ("Text1"; ["Text2"]; ...)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Zeichenfolge

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Erstellen einer Eigenschaft mit dem Namen "TÜR-Kontext-ID", bestehend aus der Hotlink Master-ID der Tür und ihrer zugehörigen Raumnummer. (Beides sind String-Eigenschaften.)

Die beiden Strings sollen durch einen Trennstrich getrennt werden.

### Ausdruck

```
CONCAT (Hotlink Master-ID;"-"; Zugehörige Raumnummer)
```

### Ergebnis

Raum 213-6

### Anmerkungen

Vergessen Sie nicht, dass eine Eigenschaft nur dann auf ein Element (einschließlich Hotlink-Elementen) angewendet werden kann, wenn die Eigenschaft für dieses Element verfügbar ist.

Die Auswertungsfunktion (aus dem Eigenschaften-Manager) funktioniert nicht mit Hotlink-Elementen. Der Ausdruck mit den Hotlink-Elementeigenschaften wird jedoch in der Interaktiven Auswertung, in den Elementeneinstellungen, in der Element-Informationspalette und in jedem anderen Interface korrekt berechnet, sofern die Eigenschaft gültig und verfügbar ist.

`CONCAT` funktioniert ähnlich wie `Zeichenfolgen verbinden [TEXTJOIN]`. Verwenden Sie `Zeichenfolgen verbinden [TEXTJOIN]`, wenn Sie eine automatische Begrenzung zwischen allen Text-Strings angeben und leere Strings ignorieren wollen.

[Ein weiteres Beispiel zu CONCAT finden Sie unter Beispiel 3: Erweitern der Eigenschaft mit beschreibendem Text-String.](#)



## Enthält [CONTAINS]

Prüft, ob eine angegebene Zeichenkette in einem Text enthalten ist.

Gibt Wahr zurück, wenn die Zeichenkette vorhanden ist.

Gibt Falsch zurück, wenn die Zeichenkette nicht vorhanden ist.

Groß-/Kleinschreibung wird standardmäßig berücksichtigt. (Andernfalls geben Sie für das Argument SchreibweiseBeachten Falsch ein.)

### Syntax

```
CONTAINS ("Suchtext"; "InText"; [SchreibweiseBeachten])
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Alle Texte: String

Groß-/Kleinschreibung berücksichtigen (optional) Wahr/Falsch

### Datentyp des Ergebnisses

Wahr/Falsch

### Beispiel

Basierend auf dem Text eines Herstellerkatalogs: Zeigen Sie die entsprechende Längeneinheit (entweder cm oder Zoll) für den Höhenwert eines Stuhls an.

Hier ist der Katalogtext (eine Zeichenfolge) ein vordefinierter Eigenschaftswert namens HeightInString.

Diese Zeichenkette wird in eine Zahl umgewandelt (mit STRTONUM).

Anschließend wird sie mit der entsprechenden Längeneinheit multipliziert (entweder cm oder Zoll, je nachdem, ob die Zeichenkette das Tag "cm" enthält: dies wird durch die Funktion CONTAINS bestimmt).

Das Ergebnis ist eine Eigenschaft vom Typ Länge.

### Ausdruck

```
STRTONUM (HeightInString) * IF (CONTAINS ("cm", HeightInString), 1 cm, 1  
Zl)
```

### Ergebnis

Bei einem gegebenen HeightInString "150 cm" wird als Ergebnis ein Längenwert von "150 cm" angezeigt.

## Zeichenfolge suchen [FIND]

Gibt die Position eines Text-Strings ("Suchtext") innerhalb eines anderen Text-Strings ("InText") zurück.

Die Suche beginnt standardmäßig mit dem ersten Zeichen. (Andernfalls beginnt die Suche mit dem Zeichen an der Position.)

Bei der Suche wird standardmäßig zwischen Groß- und Kleinbuchstaben unterschieden. (Andernfalls geben Sie für das Argument SchreibweiseBeachten Falsch ein.)

### Syntax

```
FIND ("Suchtext"; "InText"; [Position]; [SchreibweiseBeachten?])
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Alle Texte: String

Startposition (optional): Ganzzahl

Groß-/Kleinschreibung berücksichtigen? (Optional): Wahr/Falsch

### Datentyp des Ergebnisses

Ganzzahl

### Beispiel

Finden der Position des Trennstrich-Zeichens in der Element-ID (z. B. "DOO - 007").

### Ausdruck

```
FIND ("-"; Element-ID)
```

### Ergebnis

5 (für eine Element-ID "DOO - 007")

## Zeichenfolge von links [LEFT]

Gibt eine definierte Anzahl von Zeichen vom Beginn einer Zeichenfolge (String) an.

### Syntax

```
LEFT ("Text"; AnzahlVonZeichen)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Text: String

Anzahl der Zeichen: Ganzzahl

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Beginnend mit dem linken Ende des angegebenen Text-Strings werden die ersten neun Zeichen zurückgegeben.

### Ausdruck

```
LEFT ("The quick brown fox", 9)
```

### Ergebnis

The quick

## Länge der Zeichenfolge [LEN]

Gibt die Anzahl der Zeichen (einschließlich der Leerzeichen) in einer Zeichenfolge zurück.

### Syntax

```
LEN ("Text")
```

### Datentypen der Argumente

Zeichenfolge

### Datentyp des Ergebnisses

Ganzzahl

### Ausdruck

```
LEN ("The quick brown fox")
```

### Ergebnis

19

## Kleinbuchstaben [LOWER]

Wandelt alle Buchstaben einer Zeichenfolge in Kleinbuchstaben um.

### Syntax

```
LOWER ("Text")
```

### Datentypen der Argumente

Zeichenfolge

### Datentyp des Ergebnisses

Zeichenfolge

### Ausdruck

```
LOWER ("BACKSTEINWAND")
```

### Ergebnis

backsteinwand

## Zeichenfolgen von Position [MID]

Gibt eine definierte Anzahl von Zeichen, ausgehend von einer definierten Position innerhalb einer Zeichenfolge ein.

### Syntax

```
MID ("Text"; Position; AnzahlVonZeichen)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Text: String

Anfangsposition: Ganzzahl

Anzahl der Zeichen: Ganzzahl

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Beginnen Sie mit dem angegebenen Textstring an der 11. Position und geben Sie die nächsten fünf Zeichen zurück.

### Ausdruck

```
MID ("The quick brown fox", 11, 5)
```

### Ergebnis

brown

## Ersetzen in Zeichenfolge [REPLACE]

Ersetzt eine definierte Anzahl von Zeichen innerhalb einer Zeichenfolge durch eine andere Zeichenfolge, ausgehend von einer definierten Position.

### Syntax

```
REPLACE ("AlterText"; Position; AnzahlVonZeichen; "NeuerText")
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Alter und neuer Text: String

Anfangsposition: Ganzzahl

Anzahl der Zeichen: Ganzzahl

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Im String "2017" mit Position 3 beginnen und zwei Zeichen durch "18" ersetzen

### Ausdruck

```
REPLACE ("2017", 3, 2, "18")
```

### Ergebnis

2018

### Anmerkungen

Verwenden von `Zeichenfolgen ersetzen [SUBSTITUTE]`, um einen String in allen Instanzen oder an einer bestimmten Instanz durch einen anderen zu ersetzen.

Verwenden Sie `REPLACE`, um einen String ab einer gegebenen Startposition durch einen anderen zu ersetzen.

## Zeichenfolge wiederholen [REPT]

Wiederholt einen String eine angegebene Anzahl von Malen.

### Syntax

```
REPT ("Text"; AnzahlvonMalen)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Text: String

Anzahl von Malen: Ganzzahl

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Ein Sternchen und einen Bindestrich 3 Mal anzeigen.

### Ausdruck

```
REPT ("*-"; 3)
```

### Ergebnis

```
*-*-*-
```



## Zeichenfolgen von rechts [RIGHT]

Gibt eine angegebene Anzahl von Zeichen (einschließlich der Leerzeichen) vom Ende eines Text-Strings an.

### Syntax

```
RIGHT ("Text"; AnzahlVonZeichen)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Text: String

Anzahl der Zeichen: Ganzzahl

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Die letzten neun Zeichen vom (rechten) Ende eines Text-Strings zurückgeben.

### Ausdruck

```
RIGHT ("The quick brown fox", 9)
```

### Ergebnis

```
brown fox
```

## Zeichenfolge splitten [SPLIT]

Teilt eine Zeichenfolge an einer definierten Stelle (delimiter) in zwei Teile und gibt den definierten Teil zurück.

### Syntax

```
SPLIT("Text"; "Begrenzer"; Teilungsindex)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Text und Begrenzer: String

Teilungsindex: Ganzzahl

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Die Eigenschaft "Backstein-Abmessungen" einer Wand ist ein String: "92 mm x 57 mm x 194 mm".

Verwenden von SPLIT, um nur das zweite Element (die Breite) von diesem String zu erhalten: Splitten des Strings an den Begrenzungszeichen "x" und Zurückgeben des zweiten Teils.

### Ausdruck

```
SPLIT (Backstein-Abmessungen; "x"; 2)
```

### Ergebnis

57 mm

## Zeichenfolge splitten nach links [SPLITLEFT]

Teilt einen String in mehrere Teile an einem angegebenen Begrenzer.

Gibt den angegebenen Teil und alle Teile links davon zurück.

### Syntax

```
SPLITLEFT ("Text"; "Begrenzer"; Teilungsindex)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Text und Begrenzer: String

Teilungsindex: Ganzzahl

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Wir haben eine Archicad Element-ID wie z. B. TREAD-018 (Daten des Typs String):

Abrufen des String-Teils links von dem Begrenzungszeichen.

Das Begrenzungszeichen, das die Element-ID unterteilt, ist ein Trennstrich.

Die gewünschte Teilenummer (Index) ist 1 (der erste Teil).

### Ausdruck

```
SPLITLEFT (Element-ID; "-"; 1)
```

### Ergebnis

TREAD (für eine Element-ID "TREAD-018")

## Zeichenfolge splitten nach rechts [SPLITRIGHT]

Teilt einen String in mehrere Teile an einem angegebenen Begrenzer.

Gibt den angegebenen Teil und alle Teile rechts davon zurück.

### Syntax

```
SPLITRIGHT ("Text"; "Begrenzer"; Teilungsindex)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Text und Begrenzer: String

Teilungsindex: Ganzzahl

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Wir haben eine Archicad Element-ID wie z. B. TREAD-018:

Der numerische Teil der ID.

Das Begrenzungszeichen, das die Element-ID unterteilt, ist ein Trennstrich.

Die Teilenummer (Index) ist 2 (der zweite Teil).

### Ausdruck

```
SPLITRIGHT (Element-ID; "-"; 2)
```

### Ergebnis

018 (für eine Element-ID "TREAD-018")

## Zeichenfolgen ersetzen [SUBSTITUTE]

Ersetzt einen String durch einen anderen String an der angegebenen Instanz (oder alle Instanzen).

### Syntax

```
SUBSTITUTE ("Text"; "AlterText"; "NeuerText"; [InstanzIndex])
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Alle Texte: String

InstanzIndex (optional): Ganzzahl

Wenn InstanzIndex weggelassen wird, werden alle Instanzen von AlterText ersetzt.

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Die Eigenschaft "Backstein-Abmessungen" einer Wand ist ein String: "92 mm x 57 mm x 194 mm".

Verwenden von `SUBSTITUTE` zum Extrahieren nur der Nummern aus diesem String (z. B. zur Verwendung in einem anderen Ausdruck).

(Die Strings "mm" aus AlterText werden durch den leeren Inhalt von NeuerText ersetzt.)

### Ausdruck

```
SUBSTITUTE (Backstein-Abmessungen; "mm"; "")
```

Der InstanzIndex ist nicht angegeben. Somit werden alle Instanzen von AlterText ersetzt.

### Ergebnis

92 x 57 x 194

### Anmerkungen

Verwenden von `SUBSTITUTE`, um einen String in allen Instanzen oder an einer bestimmten Instanz durch einen anderen zu ersetzen.

Verwenden von Ersetzen in Zeichenfolge `[REPLACE]`, um einen bestimmten String mit einer angegebenen Startposition durch einen anderen zu ersetzen.

## Zeichenfolgen verbinden [TEXTJOIN]

Verbindet zwei oder mehr Strings zu einem String, getrennt durch ein definiertes Begrenzungszeichen. Wenn IgnoreIfEmpty Wahr ist, ignoriert das Ergebnis die leeren Strings.

### Syntax

```
TEXTJOIN ("Begrenzer"; IgnorierenWennLeer; "Text1"; "Text2"; ...)
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Begrenzer und Text: String

Ignorieren Wenn Leer: Wahr/Falsch

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Listet für jedes Element seine ID, den Raumnamen und den Umbau-Status auf.

Das Begrenzungszeichen besteht aus einem Komma und einem Leerzeichen.

### Ausdruck

```
TEXTJOIN (", "; WAHR; Element-ID; Zugehöriger Raumname; Umbau-Status)
```

### Ergebnis

SW-002, Küche, Abbruch

### Anmerkungen

`TEXTJOIN` funktioniert ähnlich wie Zeichenfolgen verschmelzen [`CONCAT`].

Im Vergleich zu `CONCAT` hat `TEXTJOIN` folgende zusätzlichen Funktionen:

- Automatisches Hinzufügen eines spezifischen Begrenzungszeichens zwischen allen Strings
- Option zum Ignorieren leerer Strings

## Anpassen [TRIM]

Entfernt doppelte Leerzeichen und die Leerzeichen vom Anfang und Ende einer Zeichenfolge.

### Syntax

```
TRIM ("Text")
```

### Datentypen der Argumente

Zeichenfolge

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Der Text-String umfasst nicht benötigte Leerzeichen am Anfang und am Ende sowie weitere überzählige Leerzeichen zwischen den Wörtern.

Verwenden von `TRIM` zum Entfernen dieser nicht benötigten Leerzeichen.

### Ausdruck

```
TRIM (" Q4 2017 Erträge Bericht ")
```

### Ergebnis

Q4 2017 Erträge Bericht

## Großbuchstaben [UPPER]

Wandelt alle Buchstaben einer Zeichenfolge in Großbuchstaben um.

### Syntax

```
UPPER ("Text")
```

### Datentypen der Argumente

Zeichenfolge

### Datentyp des Ergebnisses

Zeichenfolge

### Ausdruck

```
UPPER ("auftraggeber adresse")
```

### Ergebnis

AUFTRAGGEBER ADRESSE



# DATENKONVERTIERUNG

## Umwandeln in Zeichenfolge [STR]

Wandelt einen numerischen Wert ohne Einheit oder einen Wahr/Falsch-Wert in einen String um. Der resultierende String zeigt standardmäßig zwei Dezimalstellen.

### Syntax

```
STR (Wert; [AnzahlVonDezimalstellen])
```

**Anmerkung:** Das erforderliche Trennelement (Komma oder Semikolon) entspricht automatisch den Sprach-Einstellungen Ihres Computers.

### Datentypen der Argumente

Wert: Ganzzahl, Zahl, Wahr/Falsch

Anzahl der Dezimalstellen (optional): Ganzzahl

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Umwandeln der Element-Eigenschaft Ursprungsgeschoss-Nummer (ein Zahlenwert) in einen String. Sicherstellen, dass das Ergebnis eine Ganzzahl ohne Dezimalstellen ist.

### Ausdruck

```
STR Ursprungsgeschoss Nummer; 0)
```

### Ergebnis

2 (für einen Träger mit Ursprungsgeschoss 2)

### Anmerkungen

Der resultierende String wird standardmäßig mit zwei Dezimalstellen angezeigt. Wenn Sie keine Dezimalstellen anzeigen wollen (wie im Zahlenbeispiel Ursprungsgeschoss), fügen Sie 0 als Argument "Anzahl der Dezimalstellen" hinzu.

Zum Umwandeln eines Einheitenwerts (z. B. einer Länge) in einen String verwenden Sie die Funktion `Einheit umwandeln in Zeichenfolge [STRCALCUNIT]`.

*Ein weiteres Beispiel mit `STR` finden Sie unter [Beispiel 10: Auflisten der Gesamtkosten einer Position in Euro](#).*

## Einheit umwandeln in Zeichenfolge [STRCALCUNIT]

Wandelt einen *Wert mit Einheit* um in einen String, basierend auf den Einstellungen der Berechnungseinheiten in den Projekt-Präferenzen.

### Syntax

```
STRCALCUNIT (Wert)
```

### Datentypen der Argumente

Länge, Fläche, Volumen, Winkel

### Datentyp des Ergebnisses

Zeichenfolge

### Beispiel

Umwandeln der Dicke der Isolierschicht (eine Länge) in einen String

### Ausdruck

```
STRCALCUNIT (Dämmungsschichtdicke)
```

### Ergebnis

25

### Anmerkungen

Zum Umwandeln eines *Nicht-Einheitenwerts* (z. B. einer Ursprungsgeschoss-Nummer) in einen String verwenden Sie die Funktion `Umwandeln in Zeichenfolge [STR]`.

[Ein weiteres Beispiel zu STRCALCUNIT finden Sie unter \*Beispiel 3: Erweitern der Eigenschaft mit beschreibendem Text-String\*.](#)

## Zahl aus Zeichenfolge [STRTONUM]

Extrahiert numerische Zeichen von der ersten Position in einem String.

(Nur die numerischen Zeichen am Anfang des Strings werden berücksichtigt. Numerische Zeichen innerhalb des Strings werden nicht gefunden.)

### Syntax

```
STRTONUM ("Text")
```

### Datentypen der Argumente

Zeichenfolge

### Datentyp des Ergebnisses

Zahl

### Beispiel

Extrahieren der Zahl, die das erste Zeichen der Archicad IFC-ID des Elements (ein String-Parameter) bildet. (z.B. "3MtfJD8lf96mvm65fX2qo")

### Ausdruck

```
STRTONUM (Archicad IFC ID)
```

### Ergebnis

3 (für ein Element mit der ID "3MtfJD8lf96mvm65fX2qo")