

Formeln in Feldern

Im TecArt-System können Sie einfache Berechnungen über Angaben in den Feldern durchführen. Diese können Sie z.B. nutzen, um importierte Umsätze aus verschiedenen Jahren zusammen zurechnen oder auch die letzten Aktivitäten eines Kontaktes zu ermitteln und auszuwerten.

Die nachfolgende Beschreibung ermöglicht es Ihnen Formeln in die Felder einzufügen.

1. Auf welche Module sind die Rechenoperationen anwendbar?
2. Berechnete Felder und Grundeinstellungen
 - 2.1. Grundeinstellungen für ein Formelfeld
 - 2.2. Formel in den Objekten berechnen
3. Aufbau einer Variable
4. Aufbau von Formeln und Rechenoptionen
5. Beispiele für Formeln und deren Ergebnisse

1. Auf welche Module sind die Rechenoperationen anwendbar?

Die Rechenoperationen können in den Modulen [Kontakte](#), [Projekte](#), [Verträge](#), [Angebote](#) und [Tickets](#) angewendet werden. Zusätzlich haben Sie die Möglichkeit verschiedene Aktivitäten in die Berechnung einfließen zu lassen. Dabei können die Feldtypen verwendet werden, welche klare Ergebnisse wiedergeben können. Berechnen Sie aus zwei Währungsfeldern die Gesamtsumme in einem Feldtyp **Währung** oder lassen Sie sich aus zwei Textfeldern einen neuen **Text** erzeugen. Sie könnten auch mit einem Datumfeld und einer Währung rechnen, jedoch ist das Ergebnis fraglich.

Rechenoperationen benötigen viel Rechenkapazität. Vor allem, wenn Sie berechnete Felder in Listenansichten der Module anzeigen lassen und eine Neuberechnung über alle verwendeten Formeln anstoßen.

2. Berechnete Felder und Grundeinstellungen

2.1. Grundeinstellungen für ein Formelfeld


Erstellen Sie ein neues [benutzerdefiniertes Feld](#) und setzen Sie dieses bei Schreibgeschützt auf **Ja**. Die Formeln werden in das Standardwert-Feld eingetragen. Die Berechnung funktioniert nur mit einer sauberen Administration.

Formeln können nur innerhalb eines Modul mit den verknüpften Elementen des jeweiligen Objektes eingesetzt werden. Sie können also die Summe aller Verträge für diesen Kontakt berechnen oder den letzten Anruf ermitteln. Modulübergreifende Berechnungen und Formeln in der Form: **Zeige mir im Vertrag die Postleitzahl des verknüpften Kontaktes**, gehen hingegen nicht. Hierfür nutzen Sie den Feldtyp **Verknüpfung** in der Feldverwaltung.

2.2. Formel in den Objekten berechnen

Sobald Formeln in einem Modul vorhanden sind, wird in den Objekten ein weiterer Button angezeigt. Sofern ein Benutzer darüber hinaus auch das Recht besitzt Formeln in Übersichten zu berechnen, wird auch in der Listenansicht Ihres Moduls ein weiterer Button zur Berechnung der Formeln eingeblendet. Formeln werden also erst mal nicht grundlegend berechnet, sondern man muss die Berechnung anstoßen.

Variante 1: Button im Objekt

Button	Funktion
	Über diesen Button werden alle Formeln im ausgewählten Datensatz neu berechnet. Der Button wird nur angezeigt, sofern es hinterlegte Formeln gibt.

Variante 2: Nutzen Sie den Button  in der Übersicht, um die Berechnung für alle markierten oder gefilterten Objekte auszuführen.

Variante 3: Über Ihre [persönlichen Einstellungen](#) können Sie die Formelberechnung aktivieren, sobald Sie ein Objekt öffnen. Dabei werden sofort die enthaltenen Formeln erneut berechnet.

3. Aufbau einer Variable

Alle Variablen für die Formeln beginnen immer mit `<#` und enden mit `#>`. Dabei ist auf eine durchgehende Schreibweise der gesamten Variable zu achten. (Beispiel: `<#postcode#>`) Die Variablen müssen eindeutig zuzuordnen sein und den Anforderungen des oberen Punktes genügen.

Verwenden Sie eine Variable der selben Ebene, als Beispiel einen Kontakt, so verwenden Sie die Variable nur mit den klassischen Feldbezeichnungen, wie z.B. `<#postcode#>`. Wollen Sie Felder der verknüpften Aktivitäten verwenden, müssen Sie der Variable das gewünschte Modul mitgeben. `<#todo.stop#>` bezieht sich damit nicht mehr auf den Kontakt, in dem Sie die Formel eintragen, sondern auf das Stoppdatum der mit dem Kontakt verknüpften Aufgaben. Möchten Sie einen Wert eines ganz bestimmten Objektes für die Berechnung heranziehen, können Sie auch noch die ID des Objektes mitführen. So ergibt sich in einer Formel dann `<#todo.1332.stop#>`.

Wie bereits erwähnt, ist es nicht möglich sich über mehrere Felder zu einem anderen Feld zu hangeln, nach dem Muster einer Formel in Verträgen: Hole im Feld `<#cid#>` den Kontakt, gehe in diesen und nimm dort das Feld `<#postcode#>` und gib dieses im Vertrag aus.

Für Ihre Formeln benötigen Sie die Feldbezeichnungen Ihrer Felder. Diese finden Sie in der [Feldverwaltung](#) jeweils als Feldname bei der Bearbeitung jedes Feldes. Darüber hinaus für die verknüpften Objekte von übergeordneten Objekten die Angaben zu den Modulen.

- Kontakte = contact
- Projekte = project
- Verträge = contract
- Angebote = offer
- Anrufe = call
- Termine = appointment
- Aufgaben = todo

4. Aufbau von Formeln und Rechenoptionen

Eine Formel beginnt immer mit einem `=$=$`. Dabei gibt es folgende grundlegende Rechenoperationen. Numerischen Rechenoperationen können auch mit Strings vermischt werden.

SUM(): COUNT(): MIN(): MAX():

Zeichen	Funktion
+	Addition
-	Subtraktion
*	Multiplikation
/	Division
.	Punkt für die Bildung von Strings die Verbindung von Texten
'	Einfachen Anführungszeichen für feste Textbestandteile in einer Formel
(Klammer auf
)	Klammer zu
SUM()	Summe - Summiert die Werte aller Modulelemente, die mit dem aktuellen Modul verknüpft sind.
COUNT()	Zählen - Liefert die Anzahl aller Elemente eines Objekts.
MIN()	Minimum - Ermittelt den niedrigsten Wert aller Modulelemente, die mit dem aktuellem Modul verknüpft sind.
MAX()	Maximum - Ermittelt den maximalen Wert aller Modulelemente, die mit dem aktuellem Modul verknüpft sind.
round(,2)	Runden - Rundet das Ergebnis innerhalb des Klammersausdrucks auf 2 Stellen nach dem Komma. Dabei wird entsprechend der mathematischen Regeln auf- bzw. abgerundet.
floor()	Abrunden - Mit floor() wird diejenige ganze Zahl ausgegeben, welche kleiner oder gleich der vorgegebenen Zahl ist. Es wird ausschließlich abgerundet, auch wenn die Nachkommastelle größer als 5 ist.

5. Beispiele für Formeln und deren Ergebnisse

Eine Berechnung durch Formeln von mehreren bereits berechneten Feldern ist nur eingeschränkt möglich, da Sie den Formeln die Reihenfolge, in denen sie berechnet werden sollen, nicht mitgeben können. Wenn Sie sowohl die Einzelergebnisse als auch ein Gesamtergebnis berechnen möchten, müssen Sie für das Gesamtergebnis eine Gesamtformel aus den einzelnen Formeln bilden. Andernfalls müssen Sie die Berechnung so oft anstoßen, wie Sie Formelfelder untereinander verknüpft haben.

Um die nachfolgende Berechnung durchführen zu können, müssten Sie die folgende Teilschritte einleiten:

- $\langle \#user45\# \rangle = (\langle \#user30\# \rangle + \langle \#user31\# \rangle) / (\langle \#user30\# \rangle * \langle \#user31\# \rangle)$

Variante 1 mit Teilschritten

- $\langle \#user40\# \rangle = \langle \#user30\# \rangle + \langle \#user31\# \rangle$
- $\langle \#user41\# \rangle = \langle \#user30\# \rangle * \langle \#user31\# \rangle$
- $\langle \#user45\# \rangle = \langle \#user40\# \rangle / \langle \#user41\# \rangle$

In der Variante 1 werden die einzelnen Teilschritte berechnet und die Ausgabe erfolgt durch die Berechnung der einzelnen Teilschritte. Diese Variante wird nacheinander berechnet und benötigt bis zu zwei Aktualisierungen. Erste Berechnung User40 und User41, da diese eigenständige Ergebnisse für sich sind. Zweite Berechnung User45, da erst die Ergebnisse von User40 und User41 vorliegen müssen.

Variante 2 mit direkter Berechnung

- $\langle \#user40\# \rangle = \langle \#user30\# \rangle + \langle \#user31\# \rangle$
- $\langle \#user41\# \rangle = \langle \#user30\# \rangle * \langle \#user31\# \rangle$
- $\langle \#user45\# \rangle = (\langle \#user30\# \rangle + \langle \#user31\# \rangle) / (\langle \#user30\# \rangle * \langle \#user31\# \rangle)$

In der Variante 2 wird jede Formel für sich betrachtet und kann mit einer Aktualisierung berechnet werden, da keine der Formeln auf ein bereits berechnetes Feld zurückgreift.

Feldtyp	Formel	Beschreibung
Datum	$=\text{MAX}(\langle \#appointment.stop\# \rangle)$	Liefert das Datum des Termins mit der größten Fälligkeit im aktuell geöffneten Objekt.
Datum	$=\text{MAX}(\langle \#todo.stop\# \rangle)$	Liefert das höchste Fälligkeitsdatum einer Aufgabe im aktuell geöffneten Objekt.
Datum	$=\text{MAX}(\langle \#call.time\# \rangle)$	Liefert das letzte Anrufdatum im aktuell geöffneten Objekt.
Datum	$=\text{MAX}(\langle \#todo.start\# \rangle, \langle \#note.date\# \rangle, \langle \#call.time\# \rangle, \langle \#appointment.start\# \rangle)$	Liefert gleichzeitig den jeweils höchsten Wert der enthaltenen Aktivitäten des aktuellen Objektes und gibt den insgesamt höchst ermittelten Wert aus.
Text	$=\langle \#todo.done\# \rangle \cdot \%'$	Liefert alle Erledigungsstatus aller mit dem aktuellen Modul verknüpften Aufgaben in Prozent.
Text oder Nummer	$=\text{SUM}(\langle \#todo.achieved\# \rangle)$	Liefert die Summe der geleisteten Stunden aller mit dem Modul verknüpften Aufgaben.
Text oder Nummer	$=\text{COUNT}(\langle \#appointment\# \rangle)$	Zählt die Anzahl der mit dem Objekt verknüpften Termine.
Text oder Nummer	$=\text{COUNT}(\langle \#appointment\# \rangle) + \text{COUNT}(\langle \#call\# \rangle)$	Zählt die Anzahl der mit dem Objekt verknüpften Anrufe und Termine und addiert diese zu einer Gesamtzahl dieser beiden Aktivitäten.
Text oder Nummer	$=\text{MIN}(\langle \#todo.done\# \rangle)$	Liefert den prozentualen Wert der Aufgabe, welche am wenigsten fortgeschritten ist.
Text oder Nummer	$=\text{MAX}(\langle \#todo.done\# \rangle)$	Liefert den prozentualen Wert der Aufgabe, welche am meisten fortgeschritten ist.
Nummer oder Währung	$=\langle \#user21\# \rangle + \langle \#user22\# \rangle$	Diese Formel addiert zwei numerische Felder miteinander und liefert eine Summe.
Text	$=\langle \#user21\# \rangle \cdot \langle \#user22\# \rangle$	Diese Formel verknüpft das Textfeld User21 mit einem Leerzeichen und einem weiteren Textfeld User22 .
Nummer oder Währung	$= (\langle \#user21\# \rangle + \langle \#user22\# \rangle) * \langle \#user23\# \rangle$	Berechnet wird zuerst die Summe in den Klammern, welche dann mit dem Feld "User23" multipliziert wird.
Text oder Nummer oder Währung	$= (\langle \#user23\# \rangle * \langle \#user24\# \rangle / 100 + \langle \#user23\# \rangle) / \langle \#user25\# \rangle$	Berechnet die Summe aus einem Anfangswert User23 mit einem prozentualen Aufschlag User24 und gibt diese als Quotient wieder, weil diese durch User25 dividiert wurde.

Text	=floor((date('U')-(<#user29#>))/86400/365.25)	Diese Formel berechnet das Alter einer Person aufgrund eines eingetragenen Datums im Feld User29 . Der Wert wird als ganze Zahl ausgegeben. date('U') liefert dabei das aktuelle Datum inkl. Uhrzeit. Datumsfelder liegen als Timestamp in Sekunden seit dem 1.1.1970 vor. Demzufolge muss das Zwischenergebnis durch die Anzahl der Sekunden und durch Jahre (365.25) inkl. Schaltjahr gerechnet werden.
Text	=round((date('U')-(<#user29#>))/86400/365.25,2)	Diese Formel berechnet das Alter einer Person aufgrund eines eingetragenen Datums im Feld User29 . Der Wert wird als gerundete Zahl mit 2 Nachkommastellen ausgegeben.
Nummer oder Wahrung	=(<#user21#>+<#user22#>)*<#user23#>	Berechnet wird zuerst die Summe in den Klammern, welche dann mit dem Feld User23 multipliziert wird.
Text oder Nummer	=SUM(<#todo.achieved#>)	Diese Formel liefert die Summe der geleisteten Stunden aller mit dem Modul verknupften Aufgaben.
Text	=date('d.m.Y, G:i:s', MAX(<#call.time#>))	Diese Formel gibt Datum und Uhrzeit im angegeben Format 14.09.2019, 16:42:19 des letzten Anrufes aus.

Mehr Moglichkeiten der Formatierung finden Sie hier: <https://www.php.net/manual/de/function.date.php>.