



# Cresus Lohnbuchhaltung

## **31 - Der Formeleditor**

31.1 - Operatoren und Funktionen

31.2 - Totalisatoren

31.3 - Tests

31.4 - Umwandlungen

31.5 - Prozentrechnungen

31.6 - Datumsberechnungen

31.7 - Zeichenketten

31.8 - Sonderfunktionen


## 31 - Der Formeleditor

---

Für die Felder werden sehr oft mehr oder weniger komplizierte Formeln verwendet, um Beträge, Text oder Informationen jeder Art auszugeben.

Die mit den Feldern verbundenen Formeln werden in der Maske *Unternehmen > Felder definieren* festgelegt (§19.7.4 Rubriques de type Indemnités und §19.7.7 Création de rubriques personnalisées).

Der gleiche Formeleditor wird auch in den Feldern oder in Vorlagen verwendet.

Mit dem Symbol «Formel»  können Sie in eine Vorlage beliebige Werte, Zeichenketten oder Daten einfügen, die nicht in einem Standardfeld enthalten sind. In einer Vorlage kann das Ergebnis einer Formel nur angezeigt, nicht aber für eine spätere Verwendung gespeichert werden. Um eine bestehende Formel in einer Vorlage zu ändern, wählen Sie das Objekt aus und klicken Sie auf das Symbol «Formel bearbeiten» in der Symbolleiste oder doppelklicken Sie auf das Objekt.

Formel
✕

```
@Basis:Feriengeld %
+
@Eink:13. Monatslohn
```

**Operatoren und Funktionen**

Operator

Test

Umwandlung

%

Datum

Zeichenkett

+
-
*
/
MIN()

Einfügen

Addition oder Vereinigung von zwei Zeichenketten (Strings) .

Beispiel: 12+45

Beispiel: "Herr" + @Name

**Feld**

Kontext

Daten

Einkü

Abzüg

Basisl

Determ.

Koeff.

Wert

Numerisch

Alphanum.

Datum

Ja/Nein

Arbeitneh

Arbeitgeb.

	I...	N...	Präfix	Feldname
0,4	0	1200	@Eink:	13. Monatslohn
0,4	1	1160.	@Basis:	Feriengeld %
0,4	2		@Daten:	Aktuell bearbeiteter Monat
<input checked="" type="checkbox"/>	3		@Daten:	Verarbeitet die KA gemäss SECO/COVID-19
<input checked="" type="checkbox"/>	4		@Daten:	Arbeitsnehmer für BFS Meldungen ignorieren
0,4	5		@Basis:	KA-V1: Anrechenbarer Stundenverdienst (Monatslohn)
<input checked="" type="checkbox"/>	6		@A.Daten:	Verarbeitet die KA gemäss SECO/COVID-19
<input checked="" type="checkbox"/>	7		@Daten:	Art des Basislohns (Monat)
0,4	8		@Basis:	KA-G: Referenzstundenlohn

enthält:

Einfügen

OK

Abbrechen

C

Hilfe

Der oberste Teil der Maske ist für die Erstellung der Formel vorgesehen. Mit der Tastatur können Sie hier die Formel eingeben. Achten Sie jedoch darauf, die Feldnamen, die eher komplex sind, einschliesslich ihres Präfixes, korrekt einzugeben. Um die Eingabe der Felder zu erleichtern, verwenden Sie am besten die Listen. Zeichenketten müssen in Anführungszeichen eingegeben werden.

Die Liste *Operatoren und Funktionen* enthält alle dem Formelmodul bekannten Operatoren. Sie sind in sechs Untergruppen gegliedert:

- *Operator*: die vier Grundrechenarten und Totalisatoren
- *Test*: bedingte Anweisungen
- *Umwandlung*: Umwandlung von Formaten
- *%*: Prozentrechnungen
- *Datum*: Berechnungen mit Daten und Zeiten
- *Zeichenketten*: Verwendung von Zeichenketten

Wenn Sie eine Funktion in der Liste auswählen, erscheint rechts eine Beschreibung. Die Operatoren werden nachfolgend näher erläutert.

Die Liste *Feld* zeigt alle von Cresus Lohnbuchhaltung verwendeten Felder an. Es handelt sich um eine dynamische Liste: Tippen Sie die Bezeichnung eines Feldes in der Formel oder im Bearbeitungsfeld unten auf der Maske teilweise ein, zeigt Cresus die entsprechenden Felder an.

Um Ihre Suche zu verfeinern, können Sie die Optionen links aktivieren.

In der Liste werden zuerst die kürzlich verwendeten Felder angezeigt. Basen, Koeffizienten und Beträge eines Feldes werden, wenn möglich, zusammengefasst.

Um das ausgewählte Objekt in die Formel einzufügen, klicken Sie auf die Schaltfläche *Einfügen* oder doppelklicken Sie auf das Objekt.

Felder ohne Inhalt (leere Felder) werden in den meisten Fällen als Nullwerte behandelt. In gewissen Sonderfällen kann es nötig sein, die Funktion *STANDARD* zu verwenden. Zum Beispiel:  $3 * @Koeff:Stunden Krankheit KK$  ergibt null, wenn keine Stunden Krankheit KK bestehen.  $3 * STANDARD (@Koeff:Stunden Krankheit KK; 0)$  ergibt ebenfalls null, wenn keine Stunden Krankheit vorliegen.  $3 * STANDARD (@Koeff:Stunden Krankheit KK; 2)$  ergibt 6, wenn keine Stunden Krankheit KK bestehen

## 31.1 - Operatoren und Funktionen

---

Die vier Grundoperatoren (+, -, \*, /) dienen wie üblich zum Addieren, Subtrahieren, Multiplizieren und Dividieren von Zahlen. Multiplikationen und Divisionen werden vor Additionen und Subtraktionen ausgeführt (Punkt vor Strich).

Addition	Das Pluszeichen kann ausserdem verwendet werden, um zwei Zeilen zusammenzufügen. Dies ergibt «Guten Tag»
MIN() Sucht	den kleinsten Wert in einer Liste von Zahlen. Beispiel:
MIN(12;45;-3;28)	ergibt: -3
MAX()	Sucht den grössten Wert in einer Liste von Zahlen. Beispiel:
MAX(12;45;-3;28)	ergibt: 45
SUMME()	Berechnet die Summe einer Liste von Zahlen. Beispiel:
SUMME(12;45;-3;28)	ergibt: 82
MITTEL()	Berechnet den Mittelwert einer Liste von Zahlen. Beispiel:
MITTEL(12;45;-3;28)	ergibt: 20,5

## 31.2 - Totalisatoren

Die folgenden sechs Funktionen befinden sich ebenfalls in der Liste der Operatoren.

Mit diesen Funktionen erhalten Sie nicht den Wert eines Feldes für einen Angestellten im laufenden Monat, sondern summieren den Wert entweder über alle Angestellten, über das laufende Jahr oder über das laufende Quartal.

<p><i>TOTALANG()</i></p>	<p>Total für alle Angestellten  <i>TOTALANG(@Brutto:Ausbezahlter Lohn)</i> Die unter §30.7.9 Cumul sur les employés beschriebene Summierung über die Angestellten kann ebenfalls auf das Objekt Formel in der Vorlage angewandt werden.</p>
<p><i>TOTALJAHR()</i></p>	<p>Total für das ganze Jahr, beschränkt auf die Perioden mit einem berechneten Lohn.  <i>TOTALJAHR(@Abz:ALV-Beitrag)</i> ergibt das Total der ALV-Beiträge auf den ausbezahlten Löhnen.          Die unter §30.7.8 Cumul sur l'année beschriebene Summierung für die Periode kann ebenfalls auf das Objekt Formel in der Vorlage angewandt werden.</p>
<p><i>GANZJAHR()</i></p>	<p>Total für das ganze laufende Jahr, berücksichtigt auch die Monate ohne berechneten Lohn.  <i>GANZJAHR(STANDARD(@Abz:ALV-ALV; 200))</i> ergibt das Total der ALV-Beiträge plus 200.– pro leere Periode.</p>
<p><i>TOTQUARTAL()</i></p>	<p>Total für das Quartal, in dem sich der ausgewählte Monat befindet.  <i>TOTQUARTAL(@Eink:Rückerstattung Kilometerkosten)</i></p>
<p><i>VORHER()</i></p>	<p>Wert eines für die vorhergehende Periode berechneten Ausdrucks. <i>VORHER(@Brutto:Ausbezahlter Lohn)</i> gibt den im vorhergehenden Monat ausbezahlten Lohn aus. Zu Jahresbeginn ergibt <i>VORHER()</i> den Wert des letzten Lohnes des Vorjahres.</p>

<i>TOTALVORHER()</i>	<p>Summe aller vorhergehenden Perioden.          TOTALVORHER(@Abz:NBUV-Beitrag) gibt die bereits abgezogenen NBUV-Beiträge der vorhergehenden Monate aus.</p> <p>Diese Funktion akzeptiert einen zweiten optionalen Parameter zur Beschränkung der aufsummierten Perioden auf diejenigen mit dem gleichen Kriterium.          TOTALVORHER(@Koeff:Anzahl Tage Mutterschaftsurlaub ; @Basis:Datum der Niederkunft) ergibt die Anzahl Tage Mutterschaftsurlaub der vorhergehenden Perioden für das gleiche Datum der Niederkunft.</p>
----------------------	--

Bei der Berechnung eines Lohnes reichen diese Totalisierungen nicht über den laufenden Monat hinaus. Ein im Februar berechneter Lohn beispielsweise beachtet die Löhne vom März und von den folgenden Perioden nicht.

<i>KUMULIEREN()</i>	<p>Berechnet die Summe aller Einkünfte, die zu einer bestimmten Basis gehören. Beispiele:          KUMULIEREN(« AHV ») ergibt das Total der AHV-pflichtigen Beträge,          KUMULIEREN(« 2.1 ») ergibt das Total der unter Ziffer 2.1 des Lohnausweises ausgewiesenen Beträge</p>
---------------------	---

Folgende Begriffe können mit der Funktion KUMULIEREN() verwendet werden:



<i>NETTO</i>	im Nettolohn enthalten
<i>AHV</i>	AHV-beitragspflichtig
<i>ALV</i>	ALV-beitragspflichtig

<i>UVG</i>	UVG-beitragspflichtig
<i>UVGZ</i>	UVGZ-beitragspflichtig
<i>BVG</i>	für BVG-Prämie berücksichtigt
<i>HK</i>	unterliegt der (alten) Heilungskostenversicherung
<i>KTG</i>	unterliegt der Krankentaggeldversicherung
<i>GAV</i>	unterliegt dem GAV-Beitrag
<i>QS</i>	unterliegt der Quellensteuer
<i>FERIEN</i>	im Feriengeld berücksichtigt
<i>FEIER</i>	in der Feiertagsentschädigung berücksichtigt
<i>13E</i>	für die Berechnung des 13. Monatslohns berücksichtigt
<i>PAUSCHAL</i>	für die pauschalen AHV-Spesen berücksichtigt
<i>ZULAGEN</i>	wird zu den Familienzulagen hinzugefügt
<i>TG.</i>	Total der Taggelder
<i>DRITTE</i>	Total der Drittleistungen
<i>UKTG</i>	Total der Unfall- und Krankentaggelder






























































































*PTX*

proportionale Einkünfte für Quellensteuer (§32 Calcul des impôts à la source)

Sie können den *Code für Kumulierung* der persönlichen Basen ebenfalls verwenden (§19.7.6 Rubriques de type Bases).

Mit der Funktion *KUMULIEREN()* können auch die Positionen 1. bis 13.2.3 des Lohnausweises verarbeitet werden.

All diese Begriffe können ebenfalls im Filter  @Eink:\* und @Abz:\* verwendet werden (§30.7.10 Filtre)

Wenn Sie Ihre eigenen Basen erstellen (§19.7.6 Rubriques de type Bases)), können Sie den *Code für Kumulierung* angeben und diesen Code für die Funktion *KUMULIEREN()* verwenden.

Es können auch mehrere Kriterien beurteilt werden, indem sie mit dem Zeichen & (Funktion UND), Tastenkombination Shift + 6, oder dem «senkrechten Strich» | (Funktion ODER), Tastenkombination unter Windows Alt Gr + 7, voneinander getrennt werden.

<i>AHV&amp;UVG&amp;BVG</i>	AHV-beitragspflichtig und UVG-beitragspflichtig und BVG-beitragspflichtig
----------------------------	---

<i>AHV UVG BVG</i>	AHV-beitragspflichtig oder UVG-beitragspflichtig oder BVG-beitragspflichtig
--------------------	---

Sie können nicht gleichzeitig & (UND) und | (ODER) verwenden.

Mit der Verwendung des **Ausrufezeichens** (!) vor dem Code können Sie auch Summen von Einkünften erstellen, die **nicht** einer bestimmten Basis unterstellt sind.

<i>!AHV</i>	nicht AHV-beitragspflichtige Einkünfte
-------------	--

<i>BRUTTO&amp;!BVG</i>	dem Angestellten ausbezahlte, aber nicht BVG-beitragspflichtige Einkünfte
------------------------	---

Sie können die Funktionen auch ineinander verschachteln: Beispiel:

TOTALANG(TOTALJAHR(@Brutto: ALV-Lohn))

ergibt das Total des massgebenden ALV-Lohnes für alle Angestellten und für das ganze Jahr.

Das Verfahren kann mit Tests kombiniert werden (§31.3 Les tests), um ein Teiltotal für eine Angestelltenkategorie oder eine Periode von Daten zu erhalten. Um beispielsweise das Total eines Wertes für alle Frauen zu erhalten, geben Sie Folgendes ein:

TOTALANG(WENN(@Geschlecht des Angestellten=1; @Brutto:UVG-Lohn; 0))

Ist das Geschlecht des Angestellten 1 (also eine Frau), addieren Sie den UVG-Lohn, andernfalls addieren Sie null.

Ein weiteres Beispiel, um den Betrag eines Feldes für den Vormonat anzuzeigen:

TOTALJAHR(WENN(@Laufender Monat=@Ausgewählter Monat-1; @Eink:Lohn; 0))

Um diese Formel zu verstehen, muss man wissen, dass *@Ausgewählter Monat* dem Monat entspricht, in dem sich der Cursor in der Jahresübersicht befindet, also dem Monat, für den das Dokument ausgedruckt wird. *@Laufender Monat* hingegen ändert sich von 1 bis 12 (oder entsprechend der Anzahl Zahlperioden) während der Berechnung von *TOTALJAHR*. Die Formel in der Klammer wird somit n Mal berechnet. Indem der laufende Monat jeweils ersetzt wird, wenn er dem ausgewählten Monat minus eins entspricht, wird die gewünschte Einkunft hinzugerechnet. Sonst wird null hinzugefügt. Das Ergebnis ist somit der Wert der Einkunft für den Vormonat.

VORHER(@Eink:Lohn) ist ähnlich, sucht aber nach dem vorher ausbezahlten Betrag und ignoriert die Monate ohne Lohn.

Auf ähnliche Weise kann man den Wert für eine bestimmte Periode berechnen:

TOTALJAHR(WENN(@Laufender Monat=2; @Eink:Monatslohn + @Eink:Stundenlohn; 0))

gibt das Total der Februar-Löhne unabhängig von dem in der Jahresübersicht gerade ausgewählten Monat aus.

## 31.3 - Tests

Sie können auch Vergleiche zwischen zwei Elementen anstellen und je nach Ausgang des Vergleichs verschiedene Berechnungen ausführen.

Das Ergebnis eines Tests ist immer entweder 1 (wahr) oder 0 (falsch). Das Programm betrachtet jeden Wert, der grösser als 0 ist, als wahr und jeden Wert, der kleiner oder gleich 0 ist, als falsch.

*WENN(test; wennwahr; wennfalsch)*

Wenn das Ergebnis von *test* wahr ist, wird die Berechnung *wennwahr* ausgewertet, sonst die Berechnung *wennfalsch*:

WENN(@Brutto:Ausbezahlter Lohn > 0; « zu Ihren Gunsten »; « zu unseren Gunsten »)

Dies bedeutet: *Wenn der Nettolohn grösser als null ist, wird « zu Ihren Gunsten » geschrieben, sonst « zu unseren Gunsten ».*

=	Testet, ob zwei Zahlen <b>gleich</b> sind.
<>	Testet, ob zwei Zahlen <b>verschieden</b> sind.
<	Testet, ob die erste Zahl <b>kleiner als</b> die zweite ist.
<=	Testet, ob die erste Zahl <b>kleiner als oder gleich</b> wie die zweite ist.
>	Testet, ob die erste Zahl <b>grösser als</b> die zweite ist.
>=	Testet, ob die erste Zahl <b>grösser als oder gleich</b> wie die zweite ist.
UND()	Führt eine Addition (logisches <b>UND</b> ) zwischen mehreren Ausdrücken. Den Wert wahr ergibt, müssen alle Tests wahr sein. <i>UND(3 &gt; 2; 4 &lt; -1; 1 &gt;= 0)</i> ergibt 0 («falsch»), da 4 < -1 falsch ist.
ODER()	Führt eine Auswahl (logisches <b>ODER</b> ) unter mehreren Ausdrücken. Den Wert wahr ergibt, muss nur ein Test wahr sein. <i>ODER(3 &gt; 2; 4 &lt; -1; 1 &gt;= 0)</i> ergibt 1 («wahr»), da 3 > 2 und 1 >= 0 wahr ist.
NICHT()	kehrt das Ergebnis eines Tests um. <i>NICHT(4&lt;-1)</i> ergibt 1 (wahr).

<p><i>GESCHLECHT()</i></p>	<p>Ergibt je nach Geschlecht des Angestellten 0 oder 1: 0 = Mann, 1 = Frau Sie können auch @Daten:Geschlecht des Arbeitnehmers verwenden</p>
<p><i>MANN()</i></p>	<p>Berechnet den Ausdruck zwischen den Klammern unter Einbezug MANN(@Daten:AHV-Rentenalter) ergibt das AHV-Rentenalter für Männer.</p>
<p><i>FRAU()</i></p>	<p>Berechnet den Ausdruck zwischen den Klammern unter Einbezug FRAU(@Daten:AHV-Rentenalter) ergibt das AHV-Rentenalter für Frauen.</p>
<p><i>WENNUNTER() und WENNICHTUNTER()</i></p>	<p>Diese Tests befinden sich nicht in der Liste. Sie werden unter §...</p>



## 31.4 - Umwandlungen

<i>RUNDEN(zahl; vielfaches; aufab)</i>	<p>Rundet eine Zahl auf das nächste Vielfache auf bzw. ab. Lassen Sie den letzten Parameter weg, wird aufgerundet.</p> <p>RUNDEN(12.775) ergibt 12.80.</p> <p>RUNDEN(12.775; 0.05) ergibt 12.80</p> <p>RUNDEN(12.775; 0.05; 0) ergibt 12.75</p>
--	---

Hinweis: Das Format des angezeigten oder ausgedruckten Feldes führt ebenfalls zu einer Rundung (§30.7.6 Format des rubriques).

<i>INT(zahl)</i>	<p>Behält von einer Zahl nur den Teil vor dem Komma.</p> <p>INT(12.775) ergibt 12.</p>
------------------	--

<i>FRAC(zahl)</i>	<p>Behält von einer Zahl nur den Teil nach dem Komma.</p> <p>FRAC(12.775) ergibt 0.775.</p>
-------------------	---

<i>STANDARD(feld; wert)</i>	<p>Ersetzt ein nicht ausgefülltes Feld mit einem Standardwert (siehe Hinweis unter §31 L'éditeur de calculs).</p> <p>STANDARD(@Eink:Monatslohn;100) zeigt 100 an, wenn das Feld Monatslohn leer ist.</p>
-----------------------------	--

<i>LEER()</i>	<p>Ergibt eine leere Zahl (existiert nicht); dies kann z. B. in Verbindung mit der Funktion <i>STANDARD</i> nützlich sein (siehe Hinweis unter §31 L'éditeur de calculs).</p> <p>WENN(@Koeff:Gearbeitete Tage&gt;0 ; @Koeff:Gearbeitete Tage ; LEER() )</p> <p>zeigt nichts anstelle von 0 an, wenn die Zahl nicht grösser als 0 ist.</p>
---------------	---

ERGIBT den Wert 1 (wahr), wenn das Feld leer ist, und 0 (falsch), wenn das Gegenteil der Fall ist.  
 WENN(LEER(@Eink:Monatslohn)  
 ; « pro Stunde »  
 ; « pro Monat »  
 )  
 zeigt (druckt) «pro Stunde», wenn der Monatslohn leer ist, und «pro Monat», wenn das Gegenteil der Fall ist.

<p><i>STRING(wert; stellen)</i></p>	<p>Wandelt eine Zahl oder ein Datum in eine Zeichenkette um.          Der fakultative Parameter «stellen» bestimmt die Anzahl Kommastellen, die anzuzeigen (auszudrucken) sind.          « Betrag: » + STRING(@Eink:Monatslohn;2)          « Heute ist der » +STRING(HEUTE())</p>
-------------------------------------	---

<p><i>STRINGMONAT(monat; sprache)</i></p>	<p>Ergibt den Namen des Monats (1 bis 12). Der zweite Parameter ist fakultativ und gibt die Sprache an (0 = Französisch, 1 = Deutsch, 2 = Englisch, 3 = Italienisch). Standardmässig gibt die Funktion den Namen in der Sprache des Dokuments aus.          STRINGMONAT(2) ergibt «février».          STRINGMONAT(7; 1) ergibt «Juli».</p>
---	--

<p><i>STRINGWTAG(tag, sprache)</i></p>	<p>Ergibt den Wochentag (1 bis 7). Der zweite Parameter ist fakultativ und gibt die Sprache an (0 = Französisch, 1 = Deutsch, 2 = Englisch, 3 = Italienisch). Standardmässig gibt die Funktion den Namen in der Sprache des Dokuments aus.          STRINGWTAG(2) ergibt «mardi».          STRINGWTAG(1; 1) ergibt «Montag».</p>
--	--

*NEUEZEILE() oder n*

fügt einen Zeilenumbruch ein, sodass Sie eine Zeichenkette auf zwei Zeilen verteilen können.

@A.Daten:Name der Firma + NEUEZEILE() +

@A.Daten:Adresse der Firma

zeigt Folgendes an (oder druckt es aus):

«Espitec SA

Chemin du Fontenay»

@A.Daten:Name der Firma + « n « +

@A.Daten:Adresse der Firma

ist gleichwertig.

## 31.5 - Prozentrechnungen

---

*C%(betrag; prozentsatz)*

Berechnet den Prozentsatz eines Betrags.  
C%(20;7) ergibt 1.40

*A%(betrag; prozentsatz)*

Addiert einen Prozentsatz eines Betrags zum Betrag.  
RUNDEN(20; 7) ergibt 21.40  
% 20; -7) ergibt 18.60.

*S%(betrag; Prozentsatz)*

Berechnet den Wert eines Betrags, bevor zu diesem Betrag ein b  
wird.  
S%(21.4;7) ergibt 20,  
S%(18.6;-7) ergibt 20.

*M%(betrag; prozentsatz)*

Berechnet, um wie viel ein Betrag, zu dem ein bestimmter Prozen  
ist.  
M%(21,4;7) ergibt 1.40  
M%(18,6;-7) ergibt 1.40

*T%(betrag; betrag)*

Berechnet die Differenz in Prozent zwischen zwei Beträgen.  
T%(21.4;20) ergibt 7  
T%(18.6;20) ergibt -7

## 31.6 - Datumsberechnungen

---

Geben Sie bei einer Datumserfassung nur die letzten beiden Ziffern des Jahres ein, wird die Jahreszahl mit dem Jahrhundert 1900 oder 2000 ergänzt. Je nach Datumsart ist unterschiedlich festgelegt, welches Jahrhundert verwendet wird. Zum Beispiel:

- Bei einem Geburtsdatum werden 00 bis 19 in 2000 bis 2019 umgewandelt, 20 bis 99 in 1920 bis 1999.
- Bei einem Geburtsdatum werden 40 bis 99 in 1940 bis 1999 umgewandelt, 00 bis 39 in 2000 bis 2039.
- Handelt es sich um ein Datum für die Lohnberechnung, werden alle Jahre als 2000 und später betrachtet.

Vergewissern Sie sich deshalb immer, dass das vollständige Jahr korrekt ist.

Intern wird ein Datum durch eine ganze Zahl dargestellt, die der Anzahl vergangener Tage seit dem 1.1.1900 entspricht.

Auf diese Daten werden folgende Funktionen angewendet:

*TAG(datum)*

ergibt den Tag im Monat (1 bis 31) eines Datums.  
 TAG(DATUM(3;2;2014)) ergibt 3.

*DATUM(tag; monat; jahr)*

Ergibt die für das entsprechende Datum stehende Zahl (in Tagen seit dem 1.1.1900).  
 Behandelt die Formel nur ein Datum, wird das angegebene Datum zurückgegeben.  
 DATUM(25;12;18) ergibt 25.12.2018,  
 DATUM(25;12;18) – 15 ergibt 10.12.2018.  
 Bei einer Formel mit mehreren Daten wird die Zahl, die das Datum am nächsten liegt, zurückgegeben.  
 Zeitspannen zwischen den Daten berechnet werden können:  
 DATUM(25;12;2018) – HEUTE()  
 zeigt die Anzahl Tage zwischen heute und Weihnachten 2018 an.

<p><i>MONAT(datum)</i></p>	<p>ergibt die Nummer des Monats (1 bis 12) des betreffenden Datums.  MONAT(DATUM(3;2;2014)) ergibt 2.  Das Ergebnis wird in Textform angezeigt, wenn die Funktion nicht in einer Zelle verwendet wird.  « Es ist » +MONAT(HEUTE()) ergibt «Es ist März».  MONAT(DATUM(3;2;2014)) + 3 ergibt 5.  Um die Reihenfolge des Monats anzuzeigen, ist folgende Funktion zu verwenden:  ZAHL(MONAT(DATUM())).</p> <p>ergibt die einem Datum entsprechende Jahreszahl (1900 bis 2099).  Wird das Jahr zweistellig angegeben.</p>
<p><i>WTAG(datum)</i></p>	<p>ergibt den Wochentag (1 = Montag bis 7 = Sonntag) eines Datums.  WTAG(DATUM(28;3;14)) ergibt 5 (Freitag).</p>

<p>Wandelt ein Datum in eine Wochennummer um (0 bis 53). Die Komma-Ziffer gibt die Anzahl Tage seit Beginn der Woche an.</p>	
<p><i>ADDMONAT(datum; anzahl)</i></p>	<p>Fügt einen oder mehrere Monate zu einem Datum hinzu.  ADDMONAT(DATUM(15;5;18);2) ergibt den 15.07.18.  ADDMONAT(DATUM(31;12;17);2) ergibt den 29.02.2018  Der Parameter <i>Zahl</i> ist optional. Ohne entsprechende Angabe fügt die Funktion einen Monat hinzu.</p> <p><i>ENDEMONAT(datum)</i></p> <p>ergibt das Datum des letzten Tages eines Monats.  ENDEMONAT(DATUM(3;2;18)) ergibt den 29.02.2018.</p>

<p><i>UHRZEIT(stunde; minute; sekunde)</i></p>	<p>Berechnet die Anzahl Sekunden, die der in Stunden, Minuten und S entsprechen.</p> <p>UHRZEIT(13;40;35) ergibt 49 235.</p> <p>Die nicht angegebenen Parameter werden durch die jetzige Uhrzeit UHRZEIT(;0;0), entspricht der Uhrzeit am Anfang der Stunde. Wandelt eine Uhrzeit in Stunden um (0 bis 23). Die Kommastellen z Minuten an</p>
<p><i>MINUTE(uhrzeit)</i></p>	<p>ergibt die Anzahl Minuten (0 bis 59), die seit der vollen Stunde verg zeigen die Sekunden in 1/100 Minuten an.</p> <p>MINUTE(JETZT())</p> <p>MINUTE(UHRZEIT(13;40;35)) ergibt 40.58.</p>
<p><i>PERIODE(datum)</i></p>	<p>Wandelt eine Uhrzeit in Sekunden um (0 bis 59), die seit der vollen</p> <p>Beschränkt ein Datum auf die laufende Periode. Ein vor dieser Peri Wert des ersten Tages der Periode an, ein nach dieser Periode liegt des letzten Tages der Periode an.</p> <p>PERIODE(HEUTE())</p>
<p><i>ENDEPERIODE()</i></p>	<p>Erriht das Datum des ersten Tages der laufenden Periode. Mit dem</p> <p>Ergibt das Datum des letzten Tages der laufenden Periode. Mit dem eine andere Periode angegeben werden.</p> <p>ENDEPERIODE(11) ergibt das Datum des letzten Tages der elften</p>
<p><i>FEIERTAG(n)</i></p>	<p>Ergibt den n-ten Feiertag des laufenden Monats entsprechend der Feiertage (§19.14 Jours fériés) erfassten Liste. Hinweis: Der erste</p>





























































































--	--

## 31.7 - Zeichenketten

---

*GROSS*(*zeichenkette*) wandelt alle Zeichen der Zeichenkette in Grossbuchstaben um.  
*GROSS*(« Guten Tag ») ergibt «GUTEN TAG».

*KLEIN*(*zeichenkette*) wandelt alle Zeichen der Zeichenkette in Kleinbuchstaben um.  
*KLEIN*(« Guten Tag ») ergibt «guten tag».

*EIGENNAME*(*zeichenkette*) wandelt den ersten Buchstaben jedes ersten Wortes der Zeichenkette in einen Grossbuchstaben und die restlichen Buchstaben in Kleinbuchstaben um.  
*EIGENNAME*(« Guten tag hERR muster ») ergibt «Guten Tag Herr Muster».

*LINKS*(*zeichenkette; n*) ergibt die ersten *n* Zeichen links in der Zeichenkette.  
*LINKS*(« Guten Tag »; 5) ergibt «Guten».

*RECHTS*(*zeichenkette; n*) ergibt die ersten *n* Zeichen rechts in der Zeichenkette.  
*RECHTS*(« Guten Tag »; 3) ergibt «Tag».

*MITTE*(*zeichenkette; position; n*) extrahiert *n* Zeichen ab der *Position* (0 bis *n*-1).  
*MITTE*(« Guten Tag »;1;4) ergibt «uten».

*LÄNGE*(*zeichenkette*) ergibt die Anzahl Zeichen der Zeichenkette.  
*LÄNGE*(« Guten Tag ») ergibt 9.

*FINDEN*(*zeichenkette; kriterium*) sucht in der Zeichenkette nach dem betreffenden Kriterium.  
*FINDEN*(« Guten Tag »; « ten ») ergibt 2.  
 Befindet sich die Zeichenkette am Anfang, wird der Wert 0 ausgegeben.  
*FINDEN*(« Guten Tag »; « Gut ») ergibt 0.  
 Wird das Kriterium nicht gefunden, ergibt die Funktion den Wert -1.  
*FINDEN*(« Guten Tag »; « x ») ergibt -1.

ersetzt in einer Zeichenkette die Treffer des Parameters *Kriterium* durch den Parameter *ersetzende Zeichenkette*.

ERSETZEN(« Guten Tag »; « Tag »; « Abend ») ergibt «Guten Abend».

Es können mehrere Paare *Zeichenkette* > *ersetzende Zeichenkette* eingegeben werden. In diesem Fall wird jedes in der Zeichenkette gefundene *Kriterium* durch die entsprechende *ersetzende Zeichenkette* ersetzt. ERSETZEN(« Guten Tag Frau Bijou »; « jou »; « soi »; « Frau »; « Herr »)

ergibt «Guten Tag Herr Bisoi».

Der Parameter *Zeichenkette* kann selbstverständlich auch ein Feld sein.

ERSETZEN(@Daten:Name des Angestellten; »jou »; »soi »)

*ERSETZEN*(*zeichenkette*;  
*kriterium*; *ersetzende  
zeichenkette*)

<i>ZAHL</i> ( <i>zeichenkette</i> )	ersetzt eine Zeichenkette, die eine Zahl darstellt, durch eine Zahl. <i>ZAHL</i> (« 1'123.45 ») ergibt 1123.45.
-------------------------------------	--

extrahiert das n-te Element (gemäss Position) einer Zeichenkette, unter Berücksichtigung des definierten Separators.

WORT(« aa/bb/cc/dd »;2; »/ ») ergibt «bb».

Ein Separator kann nicht ein Wort oder eine Zeichenkette sein.

Werden mehrere Separatoren angegeben, wird jeder Separator einzeln betrachtet. Die Funktion gibt den ersten Treffer aus.

WORT(« aa+bb/cc/dd »;2; »/+& ») ergibt «bb».

Ist der zu betrachtende Separator ein Sonderzeichen, etwa ein Tabulator, ist der ASCII-Code mit der nachfolgend beschriebenen Funktion CODECAR() zu suchen.

WORT(« aa<TAB>bb<TAB>cc/dd »;2;CODECAR(9)) ergibt «bb».

*WORT*(*zeichenkette*; *position*;  
*separator*)

<p><i>ZEILE</i>(<i>zeichenkette</i>; <i>position</i>)</p>	<p>extrahiert die n-te Zeile (gemäss Parameter <i>Position</i>) eines mehrzeiligen Feldes.  <i>ZEILE</i>(@Daten:Adresse;2)</p>
<p><i>CODECAR</i>(<i>code_ascii</i>)</p>	<p>ergibt das dem ASCII-Code entsprechende Zeichen. Sie können eine Liste von Codes angeben.  <i>CODECAR</i>(65;66;9;67) ergibt «AB&lt;TAB&gt;C».</p>
<p><i>FELDNAME</i>()</p>	<p>gibt den Namen des Feldes aus. Nützlich, um den Feldnamen durch eine Berechnung ersetzen zu können (§19.2 Lieux de travail).  <i>FELDNAME</i>()+ » « +@Daten:Quellensteuercode        Es kann auch ein Parameterfeld angegeben werden. Die Funktion ergibt dann den dem Feldnamen entsprechenden Text.  <i>FELDNAME</i>(@Eink:Stundenlohn)        gibt den Wortlaut des Feldes aus, wie er vom Benutzer angepasst wurde (§19.2 Lieux de travail).</p>
<p><i>FINDENEINENUNTER</i>(<i>text</i>; <i>zeichen</i>)</p>	<p>ergibt die Position des ersten im Text gefundenen Zeichens der Liste. Zeichen ist eine <i>Zeichenkette</i>, welche die Liste der zu suchenden Zeichen ergibt.  <i>FINDENEINENUNTER</i>(« A B,C;D » ; « ;, . : »)        ergibt 3 = die Position des ersten Kommas. Der Leerschlag wird nicht gefunden, weil in der Liste der Separatoren kein Leerschlag aufgeführt ist.</p>









































































*TRIM(zeichenkette; position; kriterien)*

entfernt die Leerschläge, Tabulatoren oder Zeilenumbrüche in einer Zeichenkette.  
 TRIM( » Guten Tag ») ergibt «Guten Tag».  
 Der Parameter *Position* ist optional:  
 Wird er weggelassen, werden die Zeichen am Anfang und am Ende der Zeichenkette gelöscht.  
 Ist er 0, werden die Zeichen am Anfang der Zeichenkette gelöscht.  
 Ist er 1, werden die Zeichen am Ende der Zeichenkette gelöscht. .  
 Ist er 2, werden die Zeichen am Anfang und am Ende der Zeichenkette gelöscht.  
 TRIM( » Guten Tag « ; 0) ergibt «Guten Tag ».  
 TRIM( » Guten Tag « ; 1) ergibt « Guten Tag».  
 TRIM( » Guten Tag « ; 2) ergibt «Guten Tag».  
 Der optionale Parameter *Kriterien* ist eine Liste von Zeichen, die alle entfernt werden, wenn sie sich am Anfang oder am Ende des Textes befinden. Jedes Zeichen der *Kriterien* wird einzeln betrachtet. Sobald das Zeichen der Zeichenkette keinem der im Kriterium aufgeführten Zeichen entspricht, stoppt die Bearbeitung.  
 TRIM( » Guten Tag « ; 0; « Tag ») ergibt «Guten Tag».  
 TRIM( » Guten Tag »; 1; « Tag ») ergibt « Guten».  
 TRIM( » Guten Tag »; 2; « agT ») ergibt « Guten».  
 TRIM(« Guten Tag « ; 0; « gTa ») ergibt «Guten Tag ».  
 TRIM(« Guten Tag « ; 1; « Tga ») ergibt «Guten Tag ».  
 TRIM( » Guten Tag « ; 2; « Tag ») ergibt «Guten Tag » (das Zeichen Leerschlag ist in der Liste der zu löschenden Zeichen nicht enthalten.  
 TRIM(« Guten Tag » ; 2; « ga T ») ergibt «Guten».  
 TRIM( » agT Guten Tag « ; 2; « ga T ») ergibt «Guten».

## 31.8 - Sonderfunktionen

Die nachfolgenden Funktionen befinden sich nicht in der Liste der Operatoren der Maske. Sie können verwendet werden, indem sie direkt in die Formel eingegeben werden.

*WENNUNTER*(« *modus* »;  
*ausdruck*)

Berechnet den Ausdruck, indem nur die den Basen unterstellten Felder nach dem Modus verwendet werden (siehe Funktion *KUMULIEREN* §31.2 Les totalisateurs).

*WENNUNTER*(« BVG »

; @Eink:Dienstalterszulage + @Eink:Monatslohn)

berechnet die Summe der Werte, sofern sie der BVG-Basis unterstellt sind.

Ist nur einer der Werte dieser Basis unterstellt, wird der unterstellte Wert verwendet. Es müssen nicht beide Werte der Basis unterstellt sein.

*WENNUNTER*(« AHV »

; @Eink:Dienstalterszulage + @Eink:Monatslohn)

funktioniert auch, wenn eines der beiden Felder leer ist, weil bei einer Addition ein *leeres* Feld als null gewertet wird (siehe Hinweis §31 L'éditeur de calculs).

*WENNFELDUNTER*(« *modus* »;  
@*Feld*)

Gibt 1 oder 0 aus, je nachdem, ob das beim zweiten Parameter angegebene Feld gemäss dem Modus unterstellt ist oder nicht (siehe Funktion *KUMULIEREN* §31.2 Les totalisateurs).

*WENNFELDUNTER*(« BVG »;

@Eink:Dienstalterszulage)

ergibt 1, wenn die Dienstalterszulage BVG-beitragspflichtig ist.

berechnet den Ausdruck, indem nur die nicht den Basen unterstellten Felder nach dem Modus verwendet werden (siehe *KUMULIEREN* in §31.2 Les totalisateurs).

WENNNICHTUNTER(« BVG »  
; @Eink:Dienstalterszulage  
; @Eink:Monatslohn)

addiert die Werte, sofern sie der BVG-Basis nicht unterstellt sind

Ist nur einer der Werte dieser Basis unterstellt, wird der unterstellte Wert verwendet. Es müssen nicht beide Werte der Basis unterstellt sein.

WENNNICHTUNTER(« AHV »  
; @Eink:Dienstalterszulage  
; @Eink:Monatslohn)

funktioniert auch, wenn eines der beiden Felder leer ist, weil bei einer Addition ein leeres Feld als null gewertet wird (siehe Hinweis §31 L'éditeur de calculs).

WENNNICHTUNTER(« *modus*  
»; *ausdruck*)

ALTERSGRENZE(*grenze*;  
*modus*)

legt fest, ob das Alter des Angestellten unter einer Altersgrenze liegt.

ALTERSGRENZE(18) ergibt null, wenn der Angestellte mindestens 18 Jahre alt ist (im laufenden Jahr). Ergibt 1, wenn er 17 Jahre oder jünger ist.

Der Parameter *Modus* ist optional. Ist der Parameter 1, berücksichtigt die Funktion den Geburtsmonat und ergibt ab dem ersten Monat nach dem betreffenden Geburtstag den Wert null.

ALTERSGRENZE(65;1) ergibt ab dem ersten Monat nach dem 65. Geburtstag null.

PROPERIODE(*spalte*;  
*ausdruck*)

ergibt den Wert des Ausdrucks für die betreffende Periode.

Die Periode kann durch eine Zahl (Nr. der Spalte) oder ein Datum gegeben sein.

PROPERIODE(DATUM(15;3;2014);@Eink:Monatslohn) gibt den Monatslohn vom März 2014 aus.

<p><i>MARKIERUNGSPERIODE(spalte; parameter)</i></p>	<p>ergibt 1 (= wahr) oder 0 (= falsch) gemäss den Einstellungen, die für den Parameter unter <i>Unternehmen &gt; Perioden</i> für die betreffende <i>Spalte</i> (1 bis n) definiert wurden.</p> <p>Parameter = 1: Der 13. Monatslohn muss ausbezahlt werden.</p> <p>Parameter = 2: Die Ferien müssen ausbezahlt werden.</p> <p>Parameter = 3: Die Ferien schliessen den laufenden Monat mit ein.</p> <p>MARKIERUNGSPERIODE(6;1) gibt 1 aus, wenn der 13. Monatslohn im Juni ausbezahlt werden muss.</p>
<p><i>BETRAGSPERIODE(wert)</i></p>	<p>rechnet einen Monatsbetrag auf die laufende Periode um.</p> <p>Haben Sie z. B. einen Lohn vom 1. Februar bis und mit 15. März erstellt, Daten BETRAGSPERIODE(1000) 1500, da die Periode eineinhalb Monaten entspricht.</p>
<p><i>NAMEPERIODE(spalte)</i></p>	<p>ergibt den Namen der <i>Spalte</i> (1 bis n) in der Jahresübersicht.</p> <p>NAMEPERIODE(1) ergibt in der Regel «Januar».</p>
<p><i>FUERSPRACHE(sprache; ausdruck)</i></p>	<p>ergibt den Ausdruck in der angegebenen Sprache (0 = Französisch, 1 = Deutsch, 2 = Englisch, 3 = Italienisch).</p> <p>FUERSPRACHE(2;NAMEPERIODE(1)) gibt «January» aus.</p>



*VERSICHERUNG(feld; typ;  
modus)*

gibt die Werte je nach Versicherung des Angestellten aus.

Der erste Parameter muss das Feld mit dem Namen der Versicherung, die dem Angestellten zugeordnet ist (*@Daten:BVG-Versicherung, @Daten:KGT-Versicherung, @Daten:UVGZ-Versicherung, ...*).

Der zweite Parameter (Wert 1 bis 9) bestimmt die gewünschte Informationsart:

- 1: anwendbarer Satz (wenn BVG nach Altersgruppe)
- 2: monatliche Obergrenze
- 3: Wert für BVG von «unterstellt ab», für Krankenversicherungen der feste Abzugswert
- 4: «Abzugsbetrag» für das BVG
- 5: «Mindestbetrag» für den koordinierten Lohn (BVG)
- 8: Zustand der Option «Berechnung auf Jahresbasis» (BVG)

11: Satz des Arbeitgeberbeitrags

31: fester Wert für den Arbeitgeber

Alle diese Werte werden automatisch an die bearbeitete Periode und an das Alter des Angestellten angepasst.

*VERSICHERUNG(@Daten:KTG-Versicherung;1)*

gibt den Beitragssatz für die KTG-Versicherung des Angestellten aus.

Der dritte Parameter ist fakultativ und erlaubt es anzugeben, ob der ausgegebene Wert monatlich oder jährlich berechnet wird. Standardmässig ist der Modus monatlich. Fügen Sie den Parameter 1 hinzu, um den jährlichen Modus zu erzwingen.

*VERSICHERUNG(@Daten:KTG-Versicherung; 4; 1)*

gibt den Abzugsbetrag an, ggf. bezogen auf die Dauer der Periode.

gibt den für die Quellenbesteuerung anwendbaren Satz aus.

Der Parameter *Code* muss zwingend das Feld mit dem Quellensteuercode sein (*@Daten:Quellensteuercode*).

Der Parameter *Modus* bestimmt die gewünschte Informationsart. Dies kann sein:

1: Steuersatz gemäss dem massgebenden Betrag im 3. Parameter

2: ergibt 1, wenn die Berechnung jährlich erfolgt (0 für monatlich)

3: ergibt 1, wenn die jährliche Berechnung erst Ende Jahr erfolgen soll

4: ergibt 1, wenn die jährliche Berechnung nur bei ganzjähriger Anwesenheit des Angestellten möglich ist

5: ergibt 1, wenn die Schlussberechnung gemäss dem Code der letzten Periode erfolgen soll

6: ergibt die zu berücksichtigende monatliche Stundenzahl

7: ergibt die Adresse des unter *Unternehmen > Quellensteuer erfassten Steueramtes*

8: ergibt die Steuernummer des Unternehmens

9: ergibt den Bezugsprovisionssatz

0: ergibt 1, wenn die monatliche Berechnung rückwirkend durchzuführen ist

*STEUER(code; modus;  
jahresbasis)*

*JAEHRLICH(modus; basis; gelegentlich; rückstellung; abzug)*

führt eine Schätzung des massgebenden Jahreslohnes durch.

Der Parameter *Modus* kann sein:

0: Schätzung im Verlauf des Jahres

1: Ermittlung beim letzten Lohn

2: Extrapolation nur ab dem laufenden Monat

3: wie 1, aber beschränkt auf die Ein- und Austrittsdaten des Angestellten.

Der Parameter *Basis* muss die Summe der quellensteuerpflichtigen Einkünfte für die laufende Periode sein.

Der Parameter *gelegentlich* muss die Summe der ausnahmsweise quellensteuerpflichtigen Einkünfte sein. Diese Beträge werden deshalb nicht im Verhältnis zum Jahr berechnet.

Mit dem Parameter *Rückstellung* können die noch nicht bezahlten, aber bis Ende Jahr zu bezahlenden Beträge hinzugefügt werden (Feriengeld, 13. Monatslohn).

Der Parameter *Abzug* muss die Vorschüsse auf das Feriengeld und den 13. Monatslohn umfassen.

Die Software berücksichtigt auch die Daten der Perioden sowie die Ein- und Austrittsdaten des Angestellten.

*LISTEDEF(anzahl; text)*

Wird verwendet, um den frei wählbaren Inhalt eines Feldes vom Typ Liste zu definieren. Jedes Element der Liste setzt sich aus einem numerischen Wert und einem damit verbundenen Text zusammen. Ein Beispiel finden Sie unter §6.3 Les bases.

*VARTEMP(name; ausdruck)*

speichert das Ergebnis von *Ausdruck* in einer der 50 verfügbaren vorübergehenden Variablen. Der gespeicherte Wert kann mit *VARTEMP(name)* ohne zweiten Parameter übernommen werden.

*VARTEMP(« Test »; 250)* fügt den Wert 250 in die Variable «Test» ein.

*VARTEMP(« Test »)* gibt also 250 aus.

*WENN(VARTEMP(« Test »; KUMULIEREN(« AHV »)) > 0;*

*VARTEMP(« Test »);*

*LEER())*

In diesem Beispiel beginnt die Formel damit, das Total der AHV-pflichtigen Beträge in der vorübergehenden Variable «Test» zu speichern.

Anschliessend prüft sie, ob die Summe > 0 ist. Falls ja, gibt der Ausdruck den Inhalt der Variable «Test» (d. h. den kumulierten Betrag) aus, sonst bleibt er leer.

Das ist gleichwertig mit

*WENN(KUMULIEREN(« AHV ») > 0;*

*KUMULIEREN(« AHV »);*

*LEER())*

Das verhindert, dass zweimal kumuliert werden muss.

Die vorübergehende Variable (name) wird zu Beginn des Ausdrucks oder vor jeder Berechnung auf *LEER()* initialisiert, wenn sie anderswo als für einen Ausdruck verwendet wird.

*VARTEMP(index; ausdruck)*

speichert das Ergebnis von *ausdruck* in einer der 10 verfügbaren vorübergehenden Variablen (index = 0 .. 9). Der gespeicherte Wert kann mit *VARTEMP(index)* ohne zweiten Parameter übernommen werden.

WENN(VARTEMP(7; KUMULIEREN(« AHV »)) > 0;  
VARTEMP(7);

LEER())

Diese Formel speichert die Summe der Werte, die der AHV unterstellt sind, in der vorübergehenden Variablen 7 und zeigt sie an, wenn die Summe > 0 ist. Die vorübergehende Variable (index) wird bei Beginn jeder Berechnung oder bei einem Ausdruck auf LEER() initialisiert.

*REKURSIV(felder)*

Wird verwendet, um Crésus zu einer iterativen Berechnung mit einem von sich selbst abhängigen Wert zu zwingen.

Beispiel: Um den Arbeitgeberanteil der Sozialabgaben ausgehend von einem Brutto-Bruttolohn zu bestimmen, der somit diese Lasten einschliesst, muss man die Summe dieser Lasten berechnen und vom Brutto-Bruttolohn abziehen, um den Bruttolohn zu ermitteln, auf dem die Soziallasten zu berechnen sind, und anschliessend von vorne beginnen. Die Soziallasten variieren somit in Abhängigkeit von sich selbst.

REKURSIV(@BrutTO:Total Arbeitgeberbeiträge)

*DEBUG(ausdruck)*

zeigt die Formel und die Schritte der Beurteilung an. *Ausdruck* kann eine Formel oder ein Feld sein.

DEBUG(C%(@Brutto:AHV-Basis;@Koeff:AHV-Satz))

zeigt Folgendes an:

@Brutto:AHV-Basis[Januar] ==> 6500.

@Koeff:AHV-Satz[Januar] ==> 5.15

C%(6500. ; 5.15) ==> 334.75

<p><i>AUSGEBLENDET(ausdruck)</i></p>	<p>Blendet das gesamte Objekt aus, wenn das Ergebnis des Ausdrucks null ist.          AUSGEBLENDET(WENN(@Daten:Geschlecht des Angestellten = 0 ; « Herr »))  <i>Wenn es sich um eine Angestellte handelt, wird das Objekt ganz ausgeblendet, einschliesslich des Rahmens und/oder des Hintergrunds.</i></p>
<p><i>ZAEHLEN(zeichenkette; kriterium)</i></p>	<p>zählt die Anzahl Treffer des Kriteriums in der Zeichenkette.          ZAEHLEN(@Daten:Verschiedenes 1; « ! ») zählt die Anzahl Ausrufezeichen, die im Feld <i>Verschiedenes 1</i> vorhanden sind.</p>
<p><i>SWITCH(bedingung 0; 0; bedingung 1; 1; bedingung 2; 2; ...; bedingung n; n)</i></p>	<p>gibt das Ergebnis der Ausdrücke 0 bis n gemäss Zustand der letzten geprüften Bedingung 0 bis n aus. Ist Bedingung 0 wahr, ergibt das Resultat des Ausdrucks 0, wenn Bedingung 1 wahr ist, ergibt das Resultat des Ausdrucks 1 usw.          Das Ergebnis des Ausdrucks der letzten wahren geprüften Bedingung wird ausgegeben.          SWITCH(@Daten:ausgewählter Monat &gt;=1; « 1. Quartal »          ; @Daten:ausgewählter Monat &gt;=4; « 2. Quartal »          ;@Daten:ausgewählter Monat &gt;=7; « 3. Quartal »          ;@Daten:ausgewählter Monat &gt;=10; « 4. Quartal »)  <i>gibt je nach Monat, in dem sich der Cursor befindet, 1., 2., 3. oder 4. Quartal aus.</i></p>
<p><i>CASE(bedingung; 0; 1; .. ; n)</i></p>	<p>gibt das Ergebnis der Operationen 0 bis n anhand des Werts der Bedingung aus. Ist die Bedingung 0, wird das Ergebnis der Operation 0 ausgegeben, wenn die Bedingung 1 ist, wird das Ergebnis der Operation 1 ausgegeben, usw.</p>

*FUERJEDE(objekt; ausdruck)*

wiederholt den Ausdruck für alle *Objekttreffer*,  
Beispielsweise für jede der definierten UVG-  
Versicherungen oder für jeden Quellensteuercode.  
FUERJEDE(@Daten: BVG-Versicherung; ausdruck)  
Das Objekt muss das Feld sein, das sich auf die  
Versicherung, auf den Quellensteuercode oder auf den  
Arbeitsort bezieht.

*ARBEITSORT(ausdruck)*

Die für das Unternehmen angegebenen Arbeitsorte –  
früher «Zweigstellen» – werden mit einer  
Identifikationsnummer von 300 oder höher referenziert.  
*@Daten:Arbeitsort des Angestellten* zeigt diese  
Identifikationsnummer an oder druckt sie aus.  
ARBEITSORT(@A.Daten:Arbeitsorte)  
gibt den Namen des Arbeitsort des Angestellten aus.  
Um die Adresse und den Ort des Arbeitsorts  
auszudrucken:  
ARBEITSORT(@A.Daten:Adresse der Firma)  
+ « n »  
+ ARBEITSORT(@A.Daten:Ort der Firma)  
Dieser Ausdruck kann auch wie folgt formuliert werden:  
ARBEITSORT(@A.Daten:Adresse der Firma  
+ « n »  
+@A.Daten:Ort der Firma)  
Wenn das Dokument das gesamte Unternehmen  
betrifft, gibt die Funktion *ARBEITSORT()* die Daten  
des Hauptsitzes der Firma aus.

*PERIODEGRENZE(bedingung;  
ausdruck)*

erlaubt es, nur die Monate, in denen dieselbe  
Bedingung für den Angestellten gilt, zu bearbeiten.  
PERIODEGRENZE(@Daten: Eintrittsdatum des  
Arbeitnehmers  
; TOTALJAHR(@Brutto: Ausbezahlter Lohn))  
gibt den gesamten Lohnbetrag aus, der für die  
laufende Anstellungsperiode ausbezahlt wurde (bis  
zum laufenden Monat).

gibt die Buchungsdefinitionen eines Felds an (§19.11 Comptabilisation des rubriques).

Der ausgegebene Wert hängt vom *Parameter* ab:

*COMPTA(feld; 1)*: Sollkonto

*COMPTA(feld; 1.1)*: Sollkonto Arbeitgeber

*COMPTA(feld; 2)*: Habenkonto

*COMPTA(feld; 2.1)*: Habenkonto Arbeitgeber

*COMPTA(feld; 3)*: Buchungstext

*COMPTA(feld; 3.1)*: Buchungstext Arbeitgeberanteil

*COMPTA(feld; parameter)*

*COMPTA(feld; 4)*: Modus

*COMPTA(feld; 5)*: Analytik-Code

*COMPTA(feld; 5.1)*: Analytik-Code Arbeitgeberanteil

*COMPTA(feld; 6)*: MWST-Code

In einem Exportblock «für jede Einkunft» (§20.5

Exporter des données), verarbeitet die Funktion das aktuelle Feld der Schleife, wenn das Kriterium *Feld* leer ist:

*COMPTA(; 1.1)* gibt das Sollkonto Arbeitgeber für jedes in der Schleife verarbeitete Feld aus.



*SPECIALOP(parameter)*

*Spezielle Operationen, die beim Drucken oder beim Exportieren ausgeführt werden. SPECIALOP() verarbeitet alle Felder @Eink:\* oder @Abz:\* einer gedruckten Zeile oder eines «für jede Einkunft» exportierten Blocks (§20.5 Exporter des données). SPECIALOP() wird auf das aktuelle Feld angewandt, je nach Wert des Parameters.*

*SPECIALOP(90)*  
*gibt den Art des Koeffizienten des Felds aus:*

- 1: Typ leer*
- 0: Tage*
- 1: Stunden*
- 2: %*
- 3: ohne Typ (-)*

*SPECIALOP(91) gibt den Wert des Feldes aus.*  
*SPECIALOP(92) gibt die Basis des Feldes aus.*  
*SPECIALOP(93) gibt den Koeffizienten des Feldes aus.*  
*SPECIALOP(94) gibt den Koeffizienten Arbeitgeber des Feldes aus.*  
*SPECIALOP(95) gibt den Wert Arbeitgeber des Feldes aus.*

SPECIALOP() ist, auch für die Abzüge, ein positiver Wert. Um ein Pluszeichen vor einem negativen Abzug (eine Rückerstattung) zu drucken, ist eine Formel anzuwenden wie: *ERSETZEN(« + » + ZEICHENKETTE(0 – SPECIALOP(91); 2); « + -« ; « -« ; « +0.00 »; « »)*



Gibt die verschiedenen Angaben im Zusammenhang mit der Versicherung aus.

Alle ausgegebenen Elemente sind Texte.

*Daten:Versicherung* kann eines der folgenden Felder sein

*@Daten:UVG-Code*

*@Daten:UVGZ-Versicherung*

*@Daten:UVGZ-Versicherung (2)*

*@Daten:KTG-Versicherung*

*@Daten: KTG-Versicherung (2)*

*@Daten:HK-Versicherung*

*(Heilungskosten)*

*@Daten:BVG-Versicherung*

*@Daten: BVG-Versicherung (2)*

Der ausgegebene Wert hängt vom

**Parameter** ab. In den folgenden

Beispielen wird die

Unfallzusatzversicherung eines

Angestellten geprüft, dessen Code A

ist und der zur Abteilung Büropersonal

gehört.

0: gibt den vollständigen Namen mit

dem Code aus.

INFOVERSICHERUNG(@Daten:UVGZ-Versicherung (2); 0)

ergibt A2: Büropersonal, Kat. 2

1: gibt den Code alleine aus.

INFOVERSICHERUNG(@Daten:UVGZ-Versicherung (2); 1)

ergibt A2:

2: gibt den Namen und die Kategorie aus.

INFOVERSICHERUNG(@Daten:UVGZ-Versicherung (2); 2)

gibt Büropersonal, Kat. 2 aus

3: gibt den Code und den Namen alleine aus.

INFOVERSICHERUNG(@Daten:UVGZ-Versicherung (2); 3)

ergibt A2: Büropersonal

*INFOVERSICHERUNG(daten:versicherung; parameter)*

*TOTALANGESTCSV(kriterium; spalte 1; spalte 2; ...; spalte n)* und  
*TOTALJAHRCSV(kriterium; spalte 1; spalte 2; ...; spalte n)*

werden bei einem Export verwendet, um eine Tabelle mit den kumulierten Daten zu erstellen, die dem Kriterium entsprechen (§20.5.2 Les opérateurs spéciaux  
 TOTEMPLOYESCSV() et  
 TOTANNEESCSV()

*ADRESSE(modus; feld)*

Gibt das Element Modus des Felds PLZ/Strasse/Ort (§19.7.3 Rubriques de type Données) aus.  
 Der ausgegebene Wert hängt vom *Modus* ab.  
 0: PLZ Ort  
 1: Strasse  
 2: PLZ  
 3: Ort  
 4: Kanton  
 5: Land (ISO-Code)  
 6: Land (ausgeschrieben)  
 ADRESSE(0; @Daten:Adresse der Vertretung)  
 gibt PLZ und Ort der Vertretung aus.

*SETDATA(feld; ausdruck; parameter)*

wird verwendet, um einem Feld eine Zahl zuzuweisen. Kann in komplexen Ausdrücken verwendet werden.  
 Parameter ist fakultativ und gibt die betreffende Zahlungsperiode an. Wird er weggelassen, wird der Ausdruck dem Gesamtwert des Feldes zugewiesen.  
 SETDATA(@Eink:andere Pauschalkosten; 200; 2)  
 gibt bei den Pauschalkosten des Angestellten für den Februar 200.– ein.  
 TOTALANG(SETDATA(@Daten:Ferientage pro Jahr;25))  
 setzt als Wert für alle Angestellten 25 Tage ein.

Achtung: Bei der Verwendung von SETDATA() wird keine Überprüfung durchgeführt. Die bestehenden Löhne können geändert werden, sie werden jedoch nicht neu berechnet.

OP (ausdruck 1 |  
ausdruck 2 | ... |  
ausdruck n)

wird verwendet, um mehrere Berechnungen in einer Formel durchzuführen.

```
OP(VARTEMP(« Basis »
; STANDARD(@Eink:Monatslohn
; @Eink:Stundenlohn))
| VARTEMP(« Satz »; 12)
| VARTEMP(« Bonus »; C%(VARTEMP(« base »)
; VARTEMP(« Satz »)))
| SETDATA(@Eink:Bonus; VARTEMP(« Bonus »)))
```

INFOMESSAGE(modus;  
text)

wird in der Berechnung *Obligatorisch gemäss Berechnung* der Definition eines Feldes für die Anzeige einer Meldung in einer Hilfeblase oder nach der Eingabe eines Wertes bei der Datenerfassung verwendet.

Der Parameter *Modus* bestimmt, welches Symbol im Feld und bei der Meldung angezeigt wird:

«INFO»: *Die Meldung ist eine Information.*

«WARNING»: *Hierbei handelt es sich um eine Warnung, die aber nicht zwangsläufig befolgt werden muss, um die Datenerfassung zu beenden. Das Ausrufezeichen ist blassrot, das Eingabefeld ist weiss hinterlegt.*

«ERROR»: *Hierbei handelt es sich um eine wichtige Warnung, die aber nicht zwangsläufig befolgt werden muss, um die Datenerfassung zu beenden. Das Ausrufezeichen und das Eingabefeld sind blassrot.*

«KO»: *ist eine Warnung, die beachtet werden muss. Der eingegebene Wert entspricht nicht dem Kriterium. Die Eingabe kann nicht bestätigt werden. Das Ausrufezeichen ist leuchtend rot.*



erlaubt das Öffnen einer Textdatei und die Bearbeitung des Inhalts.

*ausdruck\_anfang* wird zu Beginn der Verarbeitung einmal geprüft

*ausdruck\_zeile* wird auf jede Zeile der Datei angewandt

*ausdruck\_ende* wird am Ende der Eingabe geprüft

Die Funktion wird nur einmal pro Datei überprüft, anschliessend gibt sie immer das geprüfte Ergebnis aus.

Werden mehrere Ergebnisse gewünscht, ist nach dem Dateinamen nach einem der folgenden Zeichen ein Kriterium einzufügen: |;!?\*/. Beispiel:

« %EXEPATH%ListeGemeinden.txt; » +

@Daten:PLZ der Angestellten

In diesem Fall wird *FURDATEI* für alle unterschiedlichen PLZ in der Cresus-Datei geprüft.

*ausdruck\_zeile* wird für jede Zeile der gelesenen Datei geprüft.

Der Inhalt der Zeile wird in *VARTEMP*(« #Line ») ausgegeben.

Auf die Variable können die üblichen Funktionen angewandt werden, etwa

*MOT*(*VARTEMP*(« #Line »); 5; « ; ») in einer CSV-Datei, um das 5. Element zu finden.

Das Ergebnis des Ausdrucks (immer eine Zeichenkette) wird zur Zeichenkette, die von der Funktion ausgegeben wird, hinzugefügt.

Beispiel:

*FURDATEI*

(« %EXEPATH%\NuPost.TXT »;

« Ortschaften: « ;

*WENN*(*WORT*(*VARTEMP*(« #Line »); 3; « t ») = « 1400 »;

*WORT*(*VARTEMP*(« #Line »); 5; « t ») + « ,»);

« plz = 1400 »

)

öffnet die Datei NuPost.txt

setzt den Text Ortschaften: an den Anfang der auszugebenden Zeichenkette

prüft jede Zeile der Datei, ob das 3. Wort der Zeile

1400 ist, fügt das 5. Wort der Zeile und ein Komma

*FURDATEI*(datei;  
*ausdruck\_anfang*;  
*ausdruck\_zeile*;  
*ausdruck\_ende*)

*DATEN(modus; ausdruck)*

gibt den von Cresus berechneten Wert oder den vom Benutzer eingegebenen Wert je nach *Modus* aus.

*Modus = 1*: gibt den von Cresus berechneten Wert aus

*Modus = 2*: gibt den vom Benutzer eingegebenen Wert aus

Der ausgegebene Wert ist leer, wenn er nicht vorhanden ist.

DATEN(1; @Eink:Monatslohn) gibt den für den Monat berechneten Monatslohn aus, auch wenn er bei der Lohnerfassung manuell geändert wurde.

*FUERSPRACHE(parameter; ausdruck)*

gibt den Ausdruck in der Sprache gemäss Parameter aus.

*Parameter* kann sein

0: Französisch

1: Deutsch

2: Englisch

3: Italienisch