

# Das Handbuch zu Calligra Sheets

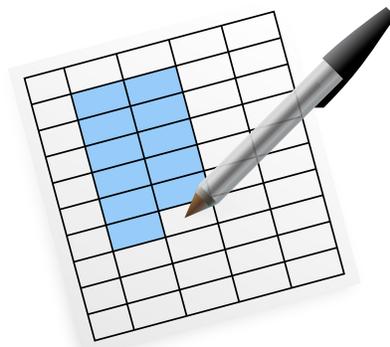
**Pamela Roberts**

**Anne-Marie Mahfouf**

**Gary Cramblitt**

**Deutsche Übersetzung: Hans-Frieder Hirsch**

**Deutsche Übersetzung: Johannes Schwall**



# Das Handbuch zu Calligra Sheets

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>16</b>
<b>2</b>	<b>Calligra Sheets-Grundlagen</b>	<b>17</b>
2.1	Tabellenkalkulation für Anfänger . . . . .	17
2.2	Zellen auswählen . . . . .	19
2.3	Dateneingabe . . . . .	20
2.3.1	Allgemeines Zellenformat . . . . .	20
2.4	Kopieren, Ausschneiden und Einfügen . . . . .	21
2.4.1	Kopieren und Einfügen von Zellenbereichen . . . . .	21
2.4.2	Andere Einfügungsarten . . . . .	22
2.5	Einfügen und löschen . . . . .	22
2.6	Einfache Summen . . . . .	23
2.6.1	Neuberechnung . . . . .	23
2.7	Daten sortieren . . . . .	23
2.8	Das Ergebnisfeld in der Statusleiste . . . . .	24
2.9	Speichern der Arbeit . . . . .	25
2.9.1	Vorlagen . . . . .	26
2.10	Drucken einer Tabelle . . . . .	26
<b>3</b>	<b>Tabellenformatierung</b>	<b>27</b>
3.1	Zellenformat . . . . .	27
3.1.1	Datenformate und Darstellungen . . . . .	28
3.1.2	Schriften und TextEinstellungen . . . . .	31
3.1.3	Textposition und -drehung . . . . .	32
3.1.4	Zellenumrandung . . . . .	33
3.1.5	Zellenhintergrund . . . . .	34
3.1.6	Zellenschutz . . . . .	34
3.2	Bedingte Zellenattribute . . . . .	35
3.3	Ändern der Zellengröße . . . . .	35
3.4	Zellen verbinden . . . . .	36
3.5	Zeilen und Spalten ausblenden . . . . .	36
3.6	Tabellen-Eigenschaften . . . . .	36

<b>4</b>	<b>Calligra Sheets für Fortgeschrittene</b>	<b>39</b>
4.1	Reihen	39
4.2	Formeln	40
4.2.1	Vorhandene Funktionen	40
4.2.2	Logische Vergleiche	40
4.2.3	Absolute Zellenverweise	41
4.3	Berechnungen mit „Einfügen Spezial“	41
4.4	Feldformeln	42
4.5	Zielwertsuche	42
4.6	Pivot-Tabellen	42
4.7	Arbeiten mit mehreren Tabellen	44
4.7.1	Daten zusammenfassen	44
4.8	Diagramme einfügen	45
4.9	Externe Daten einfügen	46
4.10	Zellen verknüpfen	47
4.11	Gültigkeitsprüfung	47
4.12	Schutz	47
4.12.1	Ein Dokument schützen	47
4.12.2	Tabellen schützen	49
4.12.3	Schutz von Zellen und Zellenbereichen	49
4.12.4	Zellenformeln ausblenden	50
4.12.5	Formel und Ergebnis ausblenden	51
4.13	Andere Funktionen	51
4.13.1	Benannte Zellen und Zellenbereiche	51
4.13.2	Zellenkommentare	52
<b>5</b>	<b>Kurzbefehle und Werkzeugleisten für Calligra Sheets einrichten</b>	<b>53</b>
5.1	Kurzbefehle	53
5.2	Werkzeugleisten	54
<b>6</b>	<b>Der Einrichtungsdialog von Calligra Sheets</b>	<b>55</b>
6.1	Oberfläche	55
6.2	Öffnen/Speichern	57
6.3	Module	58
6.4	Rechtschreibung	58
6.5	Autor	59

<b>7</b>	<b>Befehlsreferenz</b>	<b>60</b>
7.1	Das Menü Datei . . . . .	60
7.2	Das Menü Bearbeiten . . . . .	61
7.3	Das Menü Ansicht . . . . .	62
7.4	Das Menü Gehe zu . . . . .	62
7.5	Das Menü Einfügen . . . . .	63
7.6	Das Menü Format . . . . .	63
7.7	Das Menü Daten . . . . .	64
7.8	Das Menü Extras . . . . .	65
7.9	Das Menü Einstellungen . . . . .	66
7.10	Das Menü Hilfe . . . . .	66
7.11	Das Kontextmenü der rechten Maustaste . . . . .	67
7.12	Andere Tasten-Kurzbefehle . . . . .	69
<b>8</b>	<b>Funktionen</b>	<b>70</b>
8.1	Unterstützte Funktionen . . . . .	70
8.1.1	Bit-Operationen . . . . .	70
8.1.1.1	BITAND . . . . .	70
8.1.1.2	BITLSHIFT . . . . .	71
8.1.1.3	BITOR . . . . .	71
8.1.1.4	BITRSHIFT . . . . .	72
8.1.1.5	BITXOR . . . . .	72
8.1.2	Konvertierung . . . . .	72
8.1.2.1	ARABIC . . . . .	72
8.1.2.2	ASCIITOCHAR . . . . .	73
8.1.2.3	BOOL2INT . . . . .	73
8.1.2.4	BOOL2STRING . . . . .	74
8.1.2.5	CARX . . . . .	74
8.1.2.6	CARY . . . . .	75
8.1.2.7	CHARTOASCII . . . . .	75
8.1.2.8	DECSEX . . . . .	75
8.1.2.9	INT2BOOL . . . . .	76
8.1.2.10	NUM2STRING . . . . .	76
8.1.2.11	POLA . . . . .	77
8.1.2.12	POLR . . . . .	77
8.1.2.13	ROMAN . . . . .	78
8.1.2.14	SEXDEC . . . . .	78
8.1.2.15	STRING . . . . .	78
8.1.3	Datenbank . . . . .	79
8.1.3.1	DAVERAGE . . . . .	79
8.1.3.2	DCOUNT . . . . .	79
8.1.3.3	DCOUNTA . . . . .	79
8.1.3.4	DGET . . . . .	80
8.1.3.5	DMAX . . . . .	80
8.1.3.6	DMIN . . . . .	80
8.1.3.7	DPRODUCT . . . . .	81

## Das Handbuch zu Calligra Sheets

8.1.3.8	DSTDEV	81
8.1.3.9	DSTDEVP	82
8.1.3.10	DSUM	82
8.1.3.11	DVAR	82
8.1.3.12	DVARP	83
8.1.3.13	GETPIVOTDATA	83
8.1.4	Datum & Zeit	83
8.1.4.1	CURRENTDATE	83
8.1.4.2	CURRENTDATETIME	84
8.1.4.3	CURRENTTIME	84
8.1.4.4	DATE	84
8.1.4.5	DATE2UNIX	85
8.1.4.6	DATEDIF	85
8.1.4.7	DATEVALUE	85
8.1.4.8	DAY	86
8.1.4.9	DAYNAME	86
8.1.4.10	DAYOFYEAR	86
8.1.4.11	DAYS	87
8.1.4.12	DAYS360	87
8.1.4.13	DAYSINMONTH	88
8.1.4.14	DAYSINYEAR	88
8.1.4.15	EASTERSUNDAY	88
8.1.4.16	EDATE	89
8.1.4.17	EOMONTH	89
8.1.4.18	HOUR	90
8.1.4.19	HOURS	90
8.1.4.20	ISLEAPYEAR	90
8.1.4.21	ISOWEEKNUM	91
8.1.4.22	MINUTE	91
8.1.4.23	MINUTES	91
8.1.4.24	MONTH	92
8.1.4.25	MONTHNAME	92
8.1.4.26	MONTHS	92
8.1.4.27	NETWORKDAY	93
8.1.4.28	NOW	93
8.1.4.29	SECOND	94
8.1.4.30	SECONDS	94
8.1.4.31	TIME	94
8.1.4.32	TIMEVALUE	95
8.1.4.33	TODAY	95
8.1.4.34	UNIX2DATE	95
8.1.4.35	WEEKDAY	96
8.1.4.36	WEEKNUM	96
8.1.4.37	WEEKS	97
8.1.4.38	WEEKSINYEAR	97
8.1.4.39	WORKDAY	97
8.1.4.40	YEAR	98

## Das Handbuch zu Calligra Sheets

8.1.4.41	YEARFRAC	98
8.1.4.42	YEARS	98
8.1.5	Ingenieurwesen	99
8.1.5.1	BASE	99
8.1.5.2	BESSELI	99
8.1.5.3	BESSELJ	100
8.1.5.4	BESSELK	100
8.1.5.5	BESSELY	100
8.1.5.6	BIN2DEC	101
8.1.5.7	BIN2HEX	101
8.1.5.8	BIN2OCT	102
8.1.5.9	COMPLEX	102
8.1.5.10	CONVERT	102
8.1.5.11	DEC2BIN	103
8.1.5.12	DEC2HEX	104
8.1.5.13	DEC2OCT	104
8.1.5.14	DELTA	104
8.1.5.15	ERF	105
8.1.5.16	ERFC	105
8.1.5.17	GESTEP	105
8.1.5.18	HEX2BIN	106
8.1.5.19	HEX2DEC	106
8.1.5.20	HEX2OCT	107
8.1.5.21	IMABS	107
8.1.5.22	IMAGINARY	107
8.1.5.23	IMARGUMENT	108
8.1.5.24	IMCONJUGATE	108
8.1.5.25	IMCOS	108
8.1.5.26	IMCOSH	109
8.1.5.27	IMCOT	109
8.1.5.28	IMCSC	109
8.1.5.29	IMCSCH	110
8.1.5.30	IMDIV	110
8.1.5.31	IMEXP	110
8.1.5.32	IMLN	111
8.1.5.33	IMLOG10	111
8.1.5.34	IMLOG2	111
8.1.5.35	IMPOWER	112
8.1.5.36	IMPRODUCT	112
8.1.5.37	IMREAL	112
8.1.5.38	IMSEC	113
8.1.5.39	IMSECH	113
8.1.5.40	IMSIN	113
8.1.5.41	IMSINH	114
8.1.5.42	IMSQRT	114
8.1.5.43	IMSUB	114

## Das Handbuch zu Calligra Sheets

8.1.5.44	IMSUM	115
8.1.5.45	IMTAN	115
8.1.5.46	IMTANH	115
8.1.5.47	OCT2BIN	116
8.1.5.48	OCT2DEC	116
8.1.5.49	OCT2HEX	116
8.1.6	Finanzen	117
8.1.6.1	ACCRINT	117
8.1.6.2	ACCRINTM	117
8.1.6.3	AMORDEGRC	118
8.1.6.4	AMORLINC	118
8.1.6.5	COMPOUND	119
8.1.6.6	CONTINUOUS	119
8.1.6.7	COUPNUM	119
8.1.6.8	CUMIPMT	120
8.1.6.9	CUMPRINC	120
8.1.6.10	DB	121
8.1.6.11	DDB	121
8.1.6.12	DISC	122
8.1.6.13	DOLLARDE	122
8.1.6.14	DOLLARFR	122
8.1.6.15	DURATION	123
8.1.6.16	DURATION_ADD	123
8.1.6.17	EFFECT	124
8.1.6.18	EFFECTIVE	124
8.1.6.19	EURO	124
8.1.6.20	EUROCONVERT	125
8.1.6.21	FV	125
8.1.6.22	FV_ANNUITY	126
8.1.6.23	INTRATE	126
8.1.6.24	IPMT	126
8.1.6.25	IRR	127
8.1.6.26	ISPMT	127
8.1.6.27	LEVEL_COUPON	128
8.1.6.28	MDURATION	128
8.1.6.29	MIRR	129
8.1.6.30	NOMINAL	129
8.1.6.31	NPER	130
8.1.6.32	NPV	130
8.1.6.33	ODDLPRICE	131
8.1.6.34	ODDLYIELD	131
8.1.6.35	PMT	132
8.1.6.36	PPMT	132
8.1.6.37	PRICEMAT	133
8.1.6.38	PV	133
8.1.6.39	PV_ANNUITY	134

## Das Handbuch zu Calligra Sheets

8.1.6.40	RATE	134
8.1.6.41	RECEIVED	134
8.1.6.42	RRI	135
8.1.6.43	SLN	135
8.1.6.44	SYD	136
8.1.6.45	TBILLEQ	136
8.1.6.46	TBILLPRICE	137
8.1.6.47	TBILLYIELD	137
8.1.6.48	VDB	137
8.1.6.49	XIRR	138
8.1.6.50	XNPV	138
8.1.6.51	YIELDDISC	139
8.1.6.52	YIELDMAT	139
8.1.6.53	ZERO_COUPON	139
8.1.7	Information	140
8.1.7.1	ERRORTYPE	140
8.1.7.2	FILENAME	140
8.1.7.3	FORMULA	140
8.1.7.4	INFO	141
8.1.7.5	ISBLANK	141
8.1.7.6	ISDATE	141
8.1.7.7	ISERR	142
8.1.7.8	ISERROR	142
8.1.7.9	ISEVEN	142
8.1.7.10	ISFORMULA	143
8.1.7.11	ISLOGICAL	143
8.1.7.12	ISNA	143
8.1.7.13	ISNONTEXT	144
8.1.7.14	ISNOTTEXT	144
8.1.7.15	ISNUM	144
8.1.7.16	ISNUMBER	145
8.1.7.17	ISODD	145
8.1.7.18	ISREF	146
8.1.7.19	ISTEXT	146
8.1.7.20	ISTIME	146
8.1.7.21	N	147
8.1.7.22	NA	147
8.1.7.23	TYPE	147
8.1.8	Logik	148
8.1.8.1	AND	148
8.1.8.2	FALSE	148
8.1.8.3	IF	148
8.1.8.4	IFERROR	149
8.1.8.5	IFNA	149
8.1.8.6	NAND	149
8.1.8.7	NOR	150

## Das Handbuch zu Calligra Sheets

8.1.8.8	NOT	150
8.1.8.9	OR	150
8.1.8.10	TRUE	151
8.1.8.11	XOR	151
8.1.9	Nachschlagen & Referenz	152
8.1.9.1	ADDRESS	152
8.1.9.2	AREAS	152
8.1.9.3	CELL	153
8.1.9.4	CHOOSE	153
8.1.9.5	COLUMN	153
8.1.9.6	COLUMNS	154
8.1.9.7	HLOOKUP	154
8.1.9.8	INDEX	155
8.1.9.9	INDIRECT	155
8.1.9.10	LOOKUP	155
8.1.9.11	MATCH	156
8.1.9.12	MULTIPLE.OPERATIONS	156
8.1.9.13	OFFSET	157
8.1.9.14	ROW	157
8.1.9.15	ROWS	157
8.1.9.16	SHEET	158
8.1.9.17	SHEETS	158
8.1.9.18	VLOOKUP	158
8.1.10	Mathematik	159
8.1.10.1	ABS	159
8.1.10.2	CEIL	159
8.1.10.3	CEILING	159
8.1.10.4	COUNT	160
8.1.10.5	COUNTA	161
8.1.10.6	COUNTBLANK	161
8.1.10.7	COUNTIF	161
8.1.10.8	CUR	162
8.1.10.9	DIV	162
8.1.10.10	EPS	163
8.1.10.11	EVEN	163
8.1.10.12	EXP	164
8.1.10.13	FACT	164
8.1.10.14	FACTDOUBLE	164
8.1.10.15	FIB	165
8.1.10.16	FLOOR	165
8.1.10.17	GAMMA	166
8.1.10.18	GCD	166
8.1.10.19	G_PRODUCT	166
8.1.10.20	INT	167
8.1.10.21	INV	167
8.1.10.22	KPRODUCT	167

## Das Handbuch zu Calligra Sheets

8.1.10.23 LCM . . . . .	168
8.1.10.24 LN . . . . .	168
8.1.10.25 LOG . . . . .	169
8.1.10.26 LOG10 . . . . .	169
8.1.10.27 LOG2 . . . . .	170
8.1.10.28 LOGN . . . . .	170
8.1.10.29 MAX . . . . .	171
8.1.10.30 MAXA . . . . .	171
8.1.10.31 MDETERM . . . . .	172
8.1.10.32 MIN . . . . .	172
8.1.10.33 MINA . . . . .	173
8.1.10.34 MINVERSE . . . . .	173
8.1.10.35 MMULT . . . . .	174
8.1.10.36 MOD . . . . .	174
8.1.10.37 MROUND . . . . .	174
8.1.10.38 MULTINOMIAL . . . . .	175
8.1.10.39 MULTIPLY . . . . .	175
8.1.10.40 MUNIT . . . . .	176
8.1.10.41 ODD . . . . .	176
8.1.10.42 POW . . . . .	176
8.1.10.43 POWER . . . . .	177
8.1.10.44 PRODUCT . . . . .	177
8.1.10.45 QUOTIENT . . . . .	178
8.1.10.46 RAND . . . . .	178
8.1.10.47 RANDBERNOULLI . . . . .	178
8.1.10.48 RANDBETWEEN . . . . .	179
8.1.10.49 RANDBINOM . . . . .	179
8.1.10.50 RANDEXP . . . . .	179
8.1.10.51 RANDNEGBINOM . . . . .	180
8.1.10.52 RANDNORM . . . . .	180
8.1.10.53 RANDPOISSON . . . . .	180
8.1.10.54 ROOTN . . . . .	181
8.1.10.55 ROUND . . . . .	181
8.1.10.56 ROUNDDOWN . . . . .	182
8.1.10.57 ROUNDUP . . . . .	182
8.1.10.58 SERIESSUM . . . . .	183
8.1.10.59 SIGN . . . . .	183
8.1.10.60 SQRT . . . . .	184
8.1.10.61 SQRTPI . . . . .	184
8.1.10.62 SUBTOTAL . . . . .	184
8.1.10.63 SUM . . . . .	185
8.1.10.64 SUMA . . . . .	186
8.1.10.65 SUMIF . . . . .	186
8.1.10.66 SUMSQ . . . . .	187
8.1.10.67 TRANSPOSE . . . . .	187
8.1.10.68 TRUNC . . . . .	187

## Das Handbuch zu Calligra Sheets

8.1.11 Statistik . . . . .	188
8.1.11.1 AVEDEV . . . . .	188
8.1.11.2 AVERAGE . . . . .	188
8.1.11.3 AVERAGEA . . . . .	189
8.1.11.4 BETADIST . . . . .	189
8.1.11.5 BETAINV . . . . .	189
8.1.11.6 BINO . . . . .	190
8.1.11.7 CHIDIST . . . . .	190
8.1.11.8 COMBIN . . . . .	191
8.1.11.9 COMBINA . . . . .	191
8.1.11.10 CONFIDENCE . . . . .	191
8.1.11.11 CORREL . . . . .	192
8.1.11.12 COVAR . . . . .	192
8.1.11.13 DEVSQ . . . . .	193
8.1.11.14 EXPONDIST . . . . .	193
8.1.11.15 FDIST . . . . .	193
8.1.11.16 FINV . . . . .	194
8.1.11.17 FISHER . . . . .	194
8.1.11.18 FISHERINV . . . . .	194
8.1.11.19 FREQUENCY . . . . .	195
8.1.11.20 GAMMADIST . . . . .	195
8.1.11.21 GAMMAINV . . . . .	196
8.1.11.22 GAMMALN . . . . .	196
8.1.11.23 GAUSS . . . . .	196
8.1.11.24 GEOMEAN . . . . .	197
8.1.11.25 HARMEAN . . . . .	197
8.1.11.26 HYPGEOMDIST . . . . .	198
8.1.11.27 INTERCEPT . . . . .	198
8.1.11.28 INVBINO . . . . .	198
8.1.11.29 KURT . . . . .	199
8.1.11.30 KURTP . . . . .	199
8.1.11.31 LARGE . . . . .	200
8.1.11.32 LEGACYFDIST . . . . .	200
8.1.11.33 LOGINV . . . . .	200
8.1.11.34 LOGNORMDIST . . . . .	201
8.1.11.35 MEDIAN . . . . .	201
8.1.11.36 MODE . . . . .	202
8.1.11.37 NEGBINOMDIST . . . . .	202
8.1.11.38 NORMDIST . . . . .	202
8.1.11.39 NORMINV . . . . .	203
8.1.11.40 NORMSDIST . . . . .	203
8.1.11.41 NORMSINV . . . . .	203
8.1.11.42 PEARSON . . . . .	204
8.1.11.43 PERCENTILE . . . . .	204
8.1.11.44 PERMUT . . . . .	204
8.1.11.45 PERMUTATIONA . . . . .	205

## Das Handbuch zu Calligra Sheets

8.1.11.46	PHI	205
8.1.11.47	POISSON	205
8.1.11.48	RANK	206
8.1.11.49	RSQ	206
8.1.11.50	SKEW	207
8.1.11.51	SKEWP	207
8.1.11.52	SLOPE	207
8.1.11.53	SMALL	208
8.1.11.54	STANDARDIZE	208
8.1.11.55	STDEV	208
8.1.11.56	STDEVA	209
8.1.11.57	STDEVP	209
8.1.11.58	STDEVPA	210
8.1.11.59	STEYX	210
8.1.11.60	SUM2XMY	210
8.1.11.61	SUMPRODUCT	211
8.1.11.62	SUMX2MY2	211
8.1.11.63	SUMX2PY2	211
8.1.11.64	SUMXMY2	212
8.1.11.65	TDIST	212
8.1.11.66	TREND	212
8.1.11.67	TRIMMEAN	213
8.1.11.68	TTEST	213
8.1.11.69	VAR	213
8.1.11.70	VARA	214
8.1.11.71	VARIANCE	214
8.1.11.72	VARP	215
8.1.11.73	VARPA	215
8.1.11.74	WEIBULL	216
8.1.11.75	ZTEST	216
8.1.12	Text	217
8.1.12.1	ASC	217
8.1.12.2	BAHTTEXT	217
8.1.12.3	CHAR	217
8.1.12.4	CLEAN	218
8.1.12.5	CODE	218
8.1.12.6	COMPARE	218
8.1.12.7	CONCATENATE	219
8.1.12.8	DOLLAR	219
8.1.12.9	EXACT	220
8.1.12.10	FIND	220
8.1.12.11	FINDB	221
8.1.12.12	FIXED	221
8.1.12.13	JIS	222
8.1.12.14	LEFT	222
8.1.12.15	LEFTB	222

## Das Handbuch zu Calligra Sheets

8.1.12.16	LEN	223
8.1.12.17	LENB	223
8.1.12.18	LOWER	224
8.1.12.19	MID	224
8.1.12.20	MIDB	225
8.1.12.21	PROPER	225
8.1.12.22	REGEXP	225
8.1.12.23	REGEXPRE	226
8.1.12.24	REPLACE	226
8.1.12.25	REPLACEB	227
8.1.12.26	REPT	227
8.1.12.27	RIGHT	227
8.1.12.28	RIGHTB	228
8.1.12.29	ROT13	228
8.1.12.30	SEARCH	229
8.1.12.31	SEARCHB	229
8.1.12.32	SLEEK	230
8.1.12.33	SUBSTITUTE	230
8.1.12.34	T	231
8.1.12.35	TEXT	231
8.1.12.36	TOGGLE	232
8.1.12.37	TRIM	232
8.1.12.38	UNICHAR	232
8.1.12.39	UNICODE	233
8.1.12.40	UPPER	233
8.1.12.41	VALUE	233
8.1.13	Trigonometrie	234
8.1.13.1	ACOS	234
8.1.13.2	ACOSH	234
8.1.13.3	ACOT	235
8.1.13.4	ASIN	235
8.1.13.5	ASINH	235
8.1.13.6	ATAN	236
8.1.13.7	ATAN2	236
8.1.13.8	ATANH	237
8.1.13.9	COS	237
8.1.13.10	COSH	237
8.1.13.11	CSC	238
8.1.13.12	CSCH	238
8.1.13.13	DEGREES	238
8.1.13.14	PI	239
8.1.13.15	RADIANS	239
8.1.13.16	SEC	239
8.1.13.17	SECH	240
8.1.13.18	SIN	240
8.1.13.19	SINH	240
8.1.13.20	TAN	241
8.1.13.21	TANH	241
<b>9</b>	<b>Danksagungen und Lizenz</b>	<b>242</b>
<b>A</b>	<b>Installation</b>	<b>243</b>

### **Zusammenfassung**

Calligra Sheets ist ein mit allen Funktionen ausgestattetes Tabellenkalkulationsprogramm.

# Kapitel 1

## Einführung

Dieses Handbuch wird dem Gedenken an [Visicalc](#) gewidmet.

**WICHTIG**

Aktualisierte Versionen dieses Dokuments finden Sie unter <http://docs.kde.org>.

Calligra Sheets ist ein mit allen Funktionen ausgestattetes Tabellenkalkulationsprogramm. Es ist ein Teil der Calligra-Büroprogramme für KDE.

Andere Calligra-Anwendungen sind unter anderem Calligra Words (Textverarbeitung), Calligra Stage (Präsentationen).

Besuchen Sie <http://www.kde.org> für mehr Information über KDE im Allgemeinen oder <http://www.calligra.org> für mehr Informationen über Calligra.

## Kapitel 2

# Calligra Sheets-Grundlagen

Pamela Robert  
Deutsche Übersetzung: Hans-Frieder Hirsch  
Deutsche Übersetzung: Johannes Schwall

### ANMERKUNG

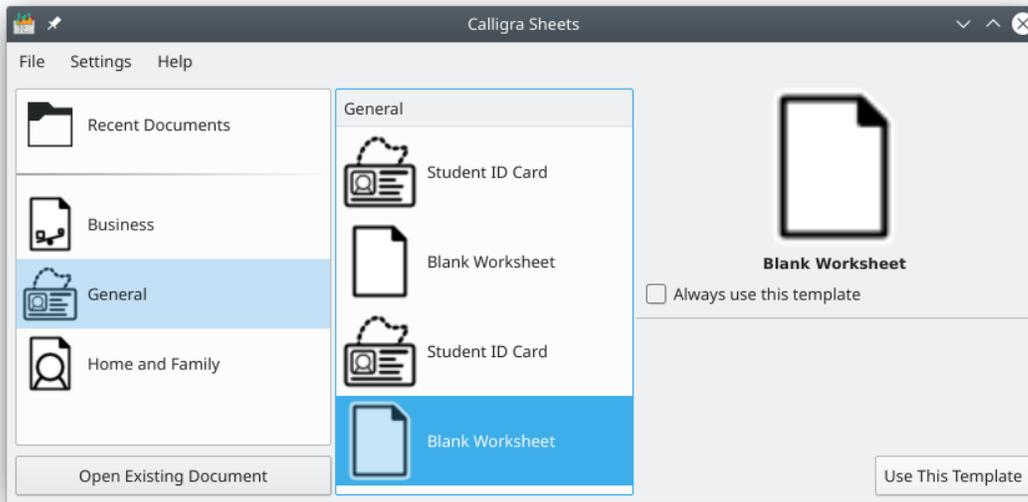
Wie alle anderen KDE-Programme kann Calligra Sheets in großem Umfang auf die individuellen Bedürfnisse der Benutzer eingestellt werden. Das kann problematisch sein, wenn Sie den Text in diesem Dokument mit Calligra Sheets auf Ihrem Bildschirm vergleichen. Um die mögliche Verwirrung zu verringern, ist es empfehlenswert, dass Sie beim ersten Start von Calligra Sheets auf allen Seiten des Calligra Sheets-Einstellungsdialogs die Voreinstellungen auswählen. Den Einstellungsdialog öffnen Sie mit **Einstellungen** → **Calligra Sheets einrichten ...**

## 2.1 Tabellenkalkulation für Anfänger

Dieser Abschnitt versucht beispielhaft zu erklären, was ein Tabellenkalkulationsprogramm wie Calligra Sheets tatsächlich tut und warum es in jeder Situation, in der Sie mit Zahlen umgehen müssen, ein so nützliches Werkzeug ist. Wenn Sie schon ein Tabellenkalkulationsprogramm benutzt haben können Sie im nächsten Abschnitt weiterlesen.

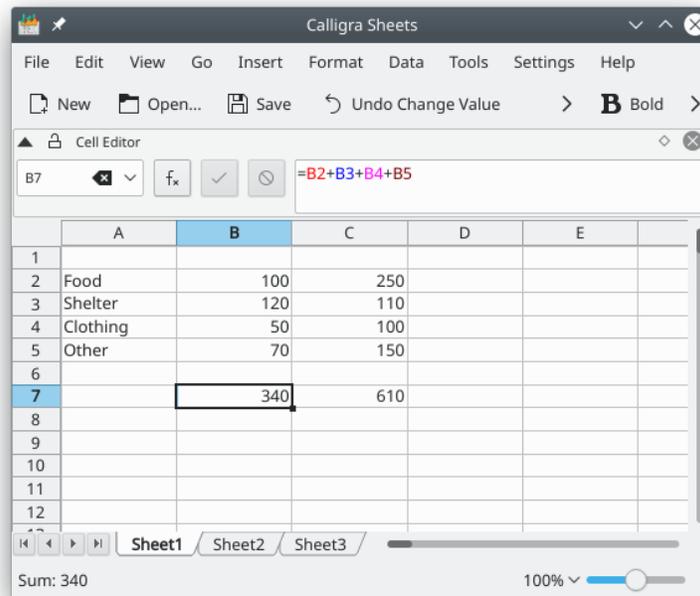
Zuerst müssen Sie Calligra Sheets starten, indem Sie mit der linken Maustaste auf das Calligra Sheets-Symbol auf Ihrer Arbeitsfläche oder in der Kontrollleiste klicken. Sie können aber auch im **K-Menü Büroprogramme** → **Calligra Sheets** auswählen.

## Das Handbuch zu Calligra Sheets



Wenn Calligra Sheets gestartet ist, haben Sie die Möglichkeit, ein kürzlich bearbeitetes Dokument zu öffnen, ein neues Dokument aus einer Vorlage zu erstellen oder ein vorhandenes Dokument zu öffnen. Klicken Sie auf die Kategorie **Allgemein** auf der linken Seite des Dialogs und wählen Sie die Vorlage **Leeres Arbeitsblatt**. Klicken Sie dann auf den Knopf **Diese Vorlage verwenden**.

Nach dem Start von Calligra Sheets sehen Sie eine Tabelle mit leeren rechteckigen Zellen, angeordnet in nummerierten Zeilen und mit Buchstaben benannten Spalten. Hier können Sie Daten, Formeln, Text oder Diagramme eingeben.



Geben Sie jetzt den Text und die Werte, die in den ersten fünf Zeilen des Bildschirmfotos oben zu sehen sind, in dieselben Zellen in Ihrer Kalkulationstabelle ein. Ignorieren Sie im Moment das, was in Zeile 7 steht. Um in eine Zelle irgend etwas einzutragen, wählen Sie zunächst eine Zelle aus, indem Sie mit der linken Maustaste in die Zelle klicken. Dann geben Sie in die Zelle beliebige Werte oder Texte ein. Danach drücken Sie die Taste **Eingabe** oder wählen Sie mit den Pfeiltasten eine andere Zelle aus.

Was bis jetzt eingetragen ist könnte ein kleiner Haushaltsplan für die nächsten zwei Monate sein, der auflistet wieviel Sie für Nahrungsmittel, Miete, Kleidung und irgendwelche andere Ausgaben ausgeben wollen. Wählen Sie jetzt die Zelle B7 (Spalte B, Zeile 7) aus und tippen Sie **=B2+B3+B4+B5** ein. Danach bestätigen Sie die Eingabe mit der Taste **Eingabe**. Weil die Eingabe mit dem Symbol **=**-beginnt, erkennt Calligra Sheets diese als eine Formel, die berechnet werden soll, in diesem Fall durch Addition der Werte in den 4 Zellen B2 bis B5. Was in der Zelle B7 zu sehen ist, ist das Ergebnis dieser Berechnung.

Sie könnten die gleiche Formel auch in die Zelle C7 eingeben. Allerdings müssten Sie dann **=C2+C3+C4+C5** eintippen. Es gibt aber einen einfacheren Weg: Kopieren Sie die Zelle B7 und fügen sie in C7 ein. Calligra Sheets wird dann die Zellverweise automatisch von B nach C umstellen, wenn das Einfügen beendet ist.

An dieser Stelle könnten Sie denken, dass Calligra Sheets nicht mehr als Sie mit Bleistift, Papier und Taschenrechner kann und damit haben Sie Recht. Denken Sie aber daran, dass das ein Beispiel einer sehr kleinen Kalkulationstabelle ist, in der nur wenige Zahlen berechnet werden. Jede große Wertetabelle kann aber mit einem Tabellenkalkulationsprogramm viel schneller und genauer berechnet werden als von Hand.

Mit einer Tabellenkalkulation können Sie auch das „Was wäre wenn?“-Spiel spielen. Denn jede Formel wird automatisch neu berechnet, wenn irgendeiner der Werte, auf die sie sich bezieht, verändert wird. Sie können sofort sehen was geschieht, wenn Sie einen der Werte verändern. In unserem Beispiel können Sie sehen welche Auswirkung es hat, wenn Sie die Ausgaben für Nahrungsmittel im Dezember verringern indem Sie einen neuen Wert in Zelle C2 eingeben. Hätten Sie eine Tabellenkalkulation, die den Treibhauseffekt genau abbildet, könnten Sie etwa die Auswirkung sehen, die eine Verringerung der Menge des in die Atmosphäre eingebrachten Methan um 50 % hätte.

## 2.2 Zellen auswählen

Sie können eine einzelne Zelle oder einen rechteckigen Zellenbereich in der Tabelle auswählen. Die ausgewählten Zellen werden mit einem dicken schwarzen Rand angezeigt.

EINE EINZELNE ZELLE KÖNNEN SIE FOLGENDERMASSEN AUSWÄHLEN:

- Klicken Sie mit der linken Maustaste auf die Zelle
- Geben Sie die Adresse der Zelle (zum Beispiel **B5**) in das Adressfeld oben links im Zelleneditor ein und drücken Sie die Taste **Eingabe**.
- Wählen Sie die Zelle mit **Gehe zu** → **Gehe zu Zelle ...**

Sie können den Cursor auch mit den **Pfeiltasten** verschieben. Die Taste **Eingabe** bewegt die aktuelle Auswahl einer Zelle nach unten, nach oben, nach links oder nach rechts, abhängig von der Einstellung auf der Seite **Oberfläche** im [Einrichtungsdialo](#)g von Calligra Sheets.

Wenn Sie die **Umschalt**-taste gedrückt halten und die **Pfeiltasten** drücken, bewegt sich die Auswahlmarkierung an den Anfang oder das Ende des Blocks der belegten Zellen.

Um einen zusammenhängenden Bereich von Zellen auszuwählen, ziehen Sie entweder mit gedrückter linker Maustaste den Mauszeiger über den gewünschten Bereich oder geben Sie die Adresse der Zelle links oben und der Zelle rechts unten, getrennt durch einen Doppelpunkt, in das Adressfeld oben links im Zelleneditor ein (Beispiel: **B7 : C14**) und drücken Sie die Taste **Eingabe**. Sie können die Zelladressen auch im gleichen Format in das Dialogfenster eingeben, das Sie **Gehe zu** → **Gehe zu Zelle ...** öffnen.

Sie können einen Bereich von Zellen auch auswählen, indem Sie eine Zelle in einer Ecke des gewünschten Bereichs auswählen und dann die **Umschalt**taste gedrückt halten, wenn Sie mit der linken Maustaste in die Zelle in der gegenüberliegenden Ecke klicken.

Um nicht zusammenhängende Zellen auszuwählen, klicken Sie in der erste Zelle, drücken dann die **Strg**-Taste und wählen die anderen Zellen aus.

Eine ganze Zeile oder Spalte von Zellen wählen Sie aus, indem Sie mit der linken Maustaste auf die Zahl links von der Zeile oder auf den Buchstaben über der Spalte klicken. Um angrenzende Zeilen oder Spalten auszuwählen ziehen Sie den Mauszeiger mit gedrückter linker Maustaste über die entsprechenden Zeilenzahlen oder Spaltenbuchstaben.

Um nicht zusammenhängende Zeilen oder Spalten auszuwählen, klicken Sie in Nummer der ersten Zeile oder auf den Buchstaben der ersten Spalte, drücken dann die **Strg**-Taste und wählen die anderen Zeilen oder Spalten aus aus.

## 2.3 Dateneingabe

Das Eingeben von Daten in eine Zelle ist recht einfach. Wählen Sie die Zelle aus, geben Sie die Daten ein und drücken Sie Taste **Eingabe** oder bewegen Sie den Cursor (die Auswahlmarkierung) mit einer der **Pfeiltasten** in eine andere Zelle. Abhängig davon, wie Sie die Daten eingeben, erkennt Calligra Sheets sie als Zahl, Datum, Zeit oder Text:

- Zahlen werden normalerweise so eingetragen: **123**, **-123**, **456**, **7** oder in wissenschaftlicher Darstellung so: **-1.2E-5**.
- Daten sollten im „System“-Format eingegeben werden, wie Sie es in den Systemeinstellungen in **Regionales** → **Land/Region & Sprache** → **Datum & Zeit** eingestellt haben. Wenn Sie beispielsweise das TT.MM.JJJJ-Format verwenden, sollten Sie **30.03.2012** für den 30. März 2012 eingeben. Führende Nullen können Sie bei der Eingabe der Tages- und Monatszahlen weglassen und für die Jahreszahl nur die letzte oder die zwei letzten Zahlen eingeben, wenn es ein Jahr im aktuellen Jahrhundert ist. Beispiel: **9.1.9** für den 9. Januar 2009.
- Uhrzeiten sollten ebenfalls im „System“-Format eingegeben werden. Wenn Sie eine 12-Stunden-Uhr benutzen, geben Sie Zeiten im Format HH:MIN am|pm oder HH:MIN:SS am|pm ein, wie z. B. **9:42 am** oder **10:30:52 pm**. **am** bedeutet dann vormittags, und **pm** nachmittags.
- Calligra Sheets behandelt jede Dateneingabe als „Text“ wenn es die Eingabe nicht als Zahl, Datum oder Zeit erkennt.

### ANMERKUNG

Als Standard richtet Calligra Sheets Zahlen, Daten und Zeiten innerhalb einer Zelle rechtsbündig und alles andere linksbündig aus. Das kann ein nützlicher Hinweis darauf sein, ob Sie ein Datum oder eine Zeit im korrekten Format eingegeben haben. Aber bedenken Sie, dass das Aussehen der Eingaben durch Ändern des [Zellen-Formats](#) geändert werden kann.

Das große Texteingabefeld des **Zelleneditors** in den Werkzeugoptionen stellt eine einfache Möglichkeit zum Eingeben und Ändern des Inhalts einer ausgewählten Zelle dar. Wenn Sie mit ihrer Eingabe zufrieden sind, drücken Sie **Eingabe** oder klicken Sie mit der linken Maustaste auf die grüne Häkchenmarkierung, oder klicken Sie auf das rote Stoppsymbol, um Ihre Änderungen zu verwerfen.

### 2.3.1 Allgemeines Zellenformat

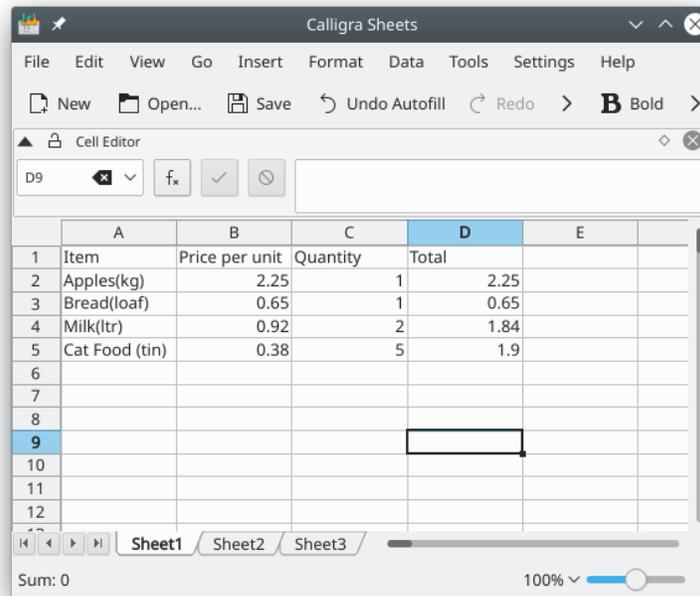
Calligra Sheets verwendet als Standard das Zellenformat „Allgemein“. Für Zellen in diesem Format bestimmt Calligra Sheets automatisch den Datentyp aus dem Zelleninhalt. Wenn Sie zum Beispiel Text in eine Zelle eingeben und später eine Zahl in dieselbe Zelle, betrachtet Calligra Sheets die neuen Daten automatisch als Zahl. Wenn Sie den Datentyp einer Zelle direkt festlegen wollen, stellen Sie das Format im Dialog [Zellenformat](#) ein. Sie können ein Zellenformat jederzeit wieder auf „Allgemein“ zurückstellen.

## 2.4 Kopieren, Ausschneiden und Einfügen

Auf den ersten Blick scheinen Calligra Sheetss **Ausschneiden**, **Kopieren** und **Einfügen** jenen Funktionen in anderen KDE-Anwendungen ähnlich zu sein. Wenn Sie eine Zelle oder Zellen ausgewählt haben, können Sie **Kopieren** oder **Ausschneiden** im Menü **Bearbeiten** oder im Auswahlmenü, das Sie bekommen, wenn Sie mit der rechten Maustaste auf eine ausgewählte Zelle klicken, auswählen. Sie können auch die Tastenkürzel **Strg+C** oder **Strg+X** verwenden. Danach bewegen Sie die Auswahl zu der gewünschten Zelle und wählen **Einfügen** oder **Strg+V**. Es gibt in Calligra Sheets jedoch einige mit diesen Funktionen verknüpfte Feinheiten, die weiter unten erläutert werden.

Wenn eine Zelle eine Formel enthält, dann wird die Formel anstatt des angezeigten Wertes kopiert. Wenn die Formel einen Verweis auf eine andere Zelle enthält, dann wird dieser Verweis durch **Ausschneiden** oder **Kopieren** und **Einfügen** so geändert, dass er auf die Zelle zeigt, die in der gleichen relativen Position zur Einfügezelle steht, wie die Zelle, auf die in der ausgeschnittenen oder kopierten Zelle verwiesen wird. Wenn z. B. Zelle A2 die Formel **=B3** enthält und nach C4 kopiert wird, enthält Zelle C4 **=D5**.

Das kann als eine ziemlich merkwürdige Art des Kopierens erscheinen, aber in 99 Prozent der Fälle ist es genau das, was gewünscht wird (wenn Sie es in einem Fall nicht wünschen, dann lesen den Abschnitt über [absolute Zellenverweise](#). Z. B. in der einfachen Einkaufsliste, die unten gezeigt wird, soll Zelle D2 **=B2\*C2** enthalten, D3 soll **=B3\*C3** enthalten, D4 soll **=B4\*C4** enthalten und so weiter. Anstatt dass Sie eine andere Formel in jede Zelle eintragen müssen, müssen Sie nur die erste Formel in D2 eintragen und sie in die Zellen weiter unten kopieren und lassen Calligra Sheets die Zellenverweise automatisch anpassen.



### 2.4.1 Kopieren und Einfügen von Zellenbereichen

Im oben genannten Beispiel kann D2 auf einmal in alle drei Zellen D3 bis D5 kopiert werden. Kopieren Sie dazu einmalig D2 und wählen dann den kompletten Zellenbereich D3:D5 aus, in den Sie dann den Inhalt von D2 einfügen.

Ein rechteckiger Bereich von Zellen kann in einem Schritt ausgeschnitten oder kopiert werden, indem man den Bereich auswählt, bevor man ihn ausschneidet oder kopiert. Wählen Sie dann

die obere linke Eckzelle des Bereichs aus, in den eingefügt werden soll, bevor Sie den Bereich einfügen.

Wenn Sie einen rechteckigen Bereich von Zellen ausschneiden oder kopieren, z. B. B2:C3 und Sie ihn in einen größeren Bereich, wie z. B. A10:D13 einfügen, wird der ursprüngliche Block von Zellen regelmäßig wiederholt, um den Zielbereich zu füllen.

Im Calligra Sheets können Sie auch den Inhalt von Zellen mit „Ziehen und Kopieren“ in andere Zellen unmittelbar unterhalb oder rechts von der Ausgangszelle kopieren. Dazu wählen Sie die zu kopierenden Zellen aus und bewegen den Mauszeiger über das kleine schwarze Quadrat an der unteren rechten Ecke der Ausgangszellen. Der Mauszeiger wird dann als einem Pfeil mit zwei Spitzen dargestellt. Halten Sie dann die linke Maustaste gedrückt, während Sie den ausgewählten Bereich durch Ziehen mit der Maus wie gewünscht vergrößern. Zellverweise in Formeln werden dabei an die geänderte Position angepasst, nur absolute Verweise bleiben erhalten.

## 2.4.2 Andere Einfügungsarten

Eine Zelle kann Text, einen Wert oder eine Formel enthalten und kann auch eine besondere Schriftart-, Rand- oder Hintergrund-**Formatierungsinformation** enthalten. Calligra Sheets verfügt über spezielle Versionen der Funktion „Einfügen“, mit denen Sie diese Formatierungen in unterschiedlicher Weise handhaben können.

Wenn Sie **Bearbeiten** → **Einfügen spezial ...** auswählen, wird der Dialog **Einfügen spezial** geöffnet. Wenn Sie im linken Teil des Dialogs den gewünschten Eintrag auswählen, können Sie **Text**, das **Format** der Zellen, einen **Kommentar** in der/den Zelle(n) oder **Alles ohne Umrandung** auswählen. Mit den Einträgen im rechten Teil des Dialogs können Sie **Rechenoperationen auf einen Zellbereich** anwenden.

**Einfügen mit Zusätzen ...** setzt die kopierte(n) Zelle(n) in das Blatt ein, indem die Zellen nach rechts oder unten verschoben werden, die sonst überschrieben werden würden. Sie kann auch verwendet werden, um komplette kopierte Zeilen oder Spalten in die Tabelle einzufügen.

## 2.5 Einfügen und löschen

Verwenden Sie die Taste **Entf** oder **Bearbeiten** → **Leeren** → **Inhalt**, um den Text, den Wert oder die Formel in den ausgewählten Zellen, Zeile(n) oder Spalte(n) zu entfernen, ohne etwas anderes, wie z. B. Formatierungen, zu beeinflussen.

Um alles in den ausgewählten Zellen, Zeilen oder Spalten, einschließlich der Kommentare und speziellen Formatierung, zu löschen, wählen Sie **Alle** aus dem Menü **Bearbeiten** → **Leeren** oder aus dem Kontextmenü, das Sie durch Klicken mit der linken Maustaste auf eine Zelle oder ausgewählten Bereich öffnen.

Um markierte Zeilen oder Spalten komplett aus der Tabelle zu entfernen, benutzen Sie **Zeilen löschen** oder **Spalten löschen** aus dem Kontextmenü der rechten Maustaste.

Wenn Sie eine oder mehrere Zellen auswählen und **Zellen entfernen ...** aus dem Kontextmenü der rechten Maustasten wählen, können Sie eingeben, ob andere Zellen von unten in der Tabelle nach oben oder von rechts nach links verschoben werden, um den Bereich der gelöschten Zellen auszufüllen.

Wenn Sie neue leere Zeilen oder Spalten in die Tabelle einfügen wollen, markieren Sie die Spalte oder Zeile, in der die neuen Zeilen oder Spalten eingefügt werden sollen und wählen Sie **Zeilen einfügen** oder **Spalten einfügen** aus dem Kontextmenü der rechten Maustaste.

Sie können neue Zellen in die Tabelle einfügen, indem Sie einen Bereich dafür auswählen. Dann wählen Sie **Zellen einfügen ...** aus dem Kontextmenü der rechten Maustaste. Geben Sie dann ein, ob die vorhandenen Zellen im ausgewählten Bereich nach rechts oder unten zum Einfügen der neuen Zellen verschoben werden sollen.

## 2.6 Einfache Summen

Wenn der erste Buchstabe in einer Zelle ein Gleichheitszeichen (=) ist, erkennt Calligra Sheets den Zelleninhalt als eine Formel, die berechnet werden soll. Das Resultat der Berechnung wird in der Zelle anstatt der Formel selbst angezeigt. Z. B. tragen Sie **=2+3** in eine Zelle ein, sollte sie 5 anzeigen.

Eine Formel kann auch Verweise auf andere Zellen enthalten, so kann mit **=B4+A3** die Summe der Werte in den Zellen B4 und A3 berechnet werden. Diese Berechnung wird immer aktualisiert, wenn eine der Zellen B4 oder A3 geändert wird.

Genauso wie + für Addition kann eine Formel die Zeichen - für Subtraktion, \* für Multiplikation und / für Division enthalten. Auch die runden Klammern ( und ) können genau wie in der Mathematik verwendet werden. Sie können also auch kompliziertere Formeln wie **=(B10+C3)\*5-F11)/2** eingeben.

Zellen, die eine Formel enthalten, werden mit einem blauen Dreieck in der linken unteren Ecke markiert, wenn die Einstellung **Formelmarkierung anzeigen** im Dialog **Format** → **Tabelle** → **Tabellen-Eigenschaften** eingeschaltet ist.

Calligra Sheets enthält auch viele eingebaute Funktionen wie statistische, trigonometrische und finanzielle Berechnungen. Ihre Anwendung wird genauer in einem späteren [Abschnitt](#) dieses Handbuchs beschrieben. Um einen ersten Eindruck zu erhalten, wählen Sie **Funktion ...** aus dem Menü **Einfügen** und im Dialog **Funktion** werden dann alle vorhandenen Funktionen angezeigt.

Häufig gebraucht wird die Funktion **SUM**. Sie berechnet die Summe aller Werte in einem angegebenen Bereich von Zellen. Zum Beispiel berechnet **=SUM(B4:C10)** die Summe aller Werte im Zellenbereich B4 bis C10.

Wenn Calligra Sheets mehrere **#VALUE!**-Symbolen anzeigt, wenn Sie eine Formel eingegeben haben, bedeutet das meistens, dass Calligra Sheets Ihre Eingabe nicht versteht. Wenn die Zeichen aber mit einem kleinen roten Pfeil enden, ist die Zelle nicht breit genug, um das gesamte Ergebnis anzuzeigen. In diesem Fall sollten Sie die Zellen verbreitern oder ihr [Format](#) ändern, damit das Ergebnis in die Zellen passt.

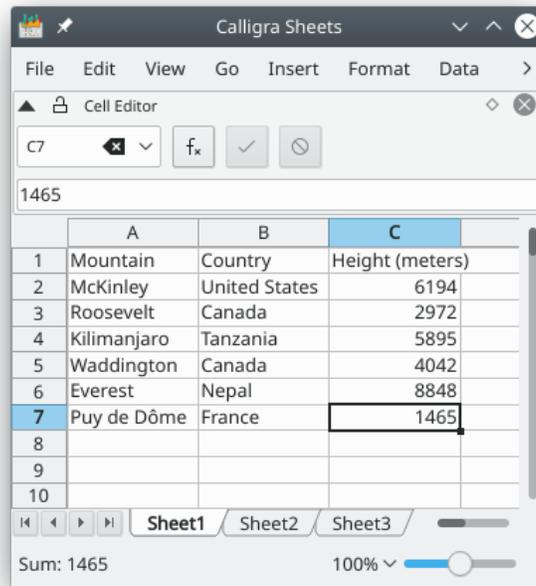
### 2.6.1 Neuberechnung

Wenn **Automatische Neuberechnung** im Dialog **Format** → **Tabelle** → **Tabelleneigenschaften** ausgewählt ist, berechnet Calligra Sheets die Werte der Zellen neu, wenn irgend ein Wert geändert wird, der in diese Berechnung eingeht.

Wenn **Automatische Neuberechnung** nicht für das aktuelle Dokument eingeschaltet ist, können eine jederzeit Neuberechnung durchführen, indem Sie **Tabelle neu berechnen** oder **Dokument neu berechnen** aus dem Menü **Extras** wählen oder die entsprechenden Kurzbefehle **Umschalt+F9** oder **F9** benutzen.

## 2.7 Daten sortieren

In dem unten gezeigten Beispiel bestehen die Daten aus den Namen von Bergen, den Ländern in denen sie stehen, und der Höhe über der Meeresspiegel. Calligra Sheets kann diese Daten auf verschiedene Weise sortieren.



Um Daten alphabetisch zu sortieren, wählen Sie die Zellen mit den Daten aus (in diesem Beispiel A2:C7) und wählen Sie **Sortieren ...** im Menü **Daten**. Damit öffnen Sie den Dialog **Sortieren**.

Die Sortierung erfolgt alphanumerisch und berücksichtigt in der Voreinstellung Groß- und Kleinschreibung. Zahlen kommen vor Großbuchstaben und diese vor Kleinbuchstaben, sodass Zellen mit den Einträgen **Cat**, **bar**, **77** und **Bat** in der folgenden Reihenfolge sortiert werden: *77 Bat Cat bar*.

Im Bereich **Richtung** diese Dialogs geben Sie an ob nach Zeilen oder Spalten sortiert werden soll. Aktivieren Sie **Erste Zeile enthält Spaltenüberschriften** oder **Erste Spalte enthält Zeilenüberschriften**, dann werden die Daten in der ersten Zeile oder Spalte bei der Sortierung nicht berücksichtigt.

Die Zeilen oder Spalten werden in der angegebenen Reihenfolge sortiert. Die kann mit den Knöpfen **Nach oben** und **Nach unten** geändert werden. Mit dem Beispiel aus dem oben gezeigten Bildschirmfoto werden bei einer Sortierung zuerst nach Spalte B und dann nach Spalte C die Daten nach Ländern und dann für jedes Land nach Höhe sortiert.

Schalten Sie die Einstellung **Groß-/Kleinschreibung beachten** ab, um die Schreibung beim Sortieren nicht zu berücksichtigen. Sie können die zwischen aufsteigender und absteigender Sortierung wechseln, indem Sie auf die Zellen in der Spalte **Sortier-Reihenfolge** klicken.

Auf der Erweiterung **Details >>** des Dialogs können Sie einstellen, ob statt alphanumerisch nach den Einträgen in einer benutzerdefinierten Liste sortiert werden soll wie beispielsweise Januar, Februar ... Die Formatierungen der Zellen werden zusammen mit dem Inhalt der Zellen bei der Sortierung verschoben, wenn Sie die Einstellung **Zellenformatierung verschieben (Umrandungen, Format, Textstil)** aktivieren.

## 2.8 Das Ergebnisfeld in der Statusleiste

Auf der linken Seite der Statuszeile wird das Ergebnis einer Berechnung mit den Werten in den ausgewählten Zellen angezeigt. Je nach Einstellung von **In der Statusleiste gezeigte Funktion** im Auswahlfeld der Seite **Oberfläche** im Einrichtungsdialog von Calligra Sheets wird als Ergebnis der Berechnung folgendes angezeigt:

### **Keine**

Es wird keine Berechnung durchgeführt.

### **Durchschnitt**

Der Durchschnitt der Werte in den ausgewählten Zellen wird angezeigt.

### **Anzahl**

Die Anzahl der Zellen mit numerischen Werten wird angezeigt.

### **CountA**

Die Anzahl der nicht leeren Zellen wird angezeigt.

### **Maximum**

Das Maximum der Werte in den ausgewählten Zellen wird angezeigt.

### **Minimum**

Das Minimum der Werte in den ausgewählten Zellen wird angezeigt.

### **Summe**

Die Summe der Werte in den ausgewählten Zellen wird angezeigt.

Die Berechnungsmethode können Sie ändern, indem Sie mit der rechten Maustaste auf das Ergebnisfeld in der Statusleiste klicken und einen Eintrag aus dem Kontextmenü auswählen.

## **2.9 Speichern der Arbeit**

Calligra Sheets speichert das gesamte Dokument, das mehr als eine Tabelle enthalten kann, in einer einzigen Datei.

Wenn Sie eine neues Dokument erstellt haben oder ein Dokument unter einem anderem Namen speichern wollen, verwenden Sie **Datei** → **Speichern unter ...**. Damit öffnen Sie den KDE-Standarddialog **Dokument speichern unter**. Wählen Sie einen Ordner für das Dokument aus und geben Sie einen aussagekräftigen Namen in das Textfeld **Name** ein. Calligra Sheets-Dokumente werden normalerweise automatisch mit der Dateinamenerweiterung `.ods` gespeichert. Deshalb müssen Sie die Erweiterung nicht zum Dateinamen hinzufügen. Vergewissern Sie sich aber, das im Feld **Filter: ODS-Tabelle** eingestellt ist.

Um Ihr Dokument unter dem gleichen Namen erneut zu speichern, wählen Sie **Datei** → **Speichern**.

Sie können ein Calligra Sheets-Dokument auch in einem anderen (fremden) Format speichern, indem das gewünschte Format im Auswahlfeld **Filter** einstellen.

Wenn Sie eine geänderte Version eines bereits vorhandenen Dokuments speichern, speichert Calligra Sheets die vorige Version als Sicherungs-(Backup-)Datei und fügt ein `~` am Ende des Dateinamens hinzu.

Calligra Sheets bietet einen begrenzten Schutz gegen den Verlust Ihrer Arbeit durch einen Rechnerabsturz, oder weil Sie Calligra Sheets beendet haben, ohne das aktuelle Dokument zu speichern. Die wird dadurch erreicht, dass die aktuelle Version, an der Sie arbeiten, automatisch in einem einstellbaren Zeitabstand unter einem anderen Dateinamen gesichert wird. Diese automatisch gesicherte Version wird gelöscht, wenn Sie Ihr Dokument speichern. Daher ist die automatisch gesicherte Version nur dann vorhanden, wenn sie aktueller als die von Ihnen gespeicherte Version ist. Wenn Sie ein Dokument öffnen, überprüft Calligra Sheets, ob eine automatisch gesicherte Version vorhanden ist. Dann haben Sie die Möglichkeit, diese automatisch gesicherte Version statt der älteren Version zu öffnen.

Automatisch gesicherte Dokumente werden mit einem Dateinamen in der Formn `.IhrDateiname.e.autosave` gespeichert (beachten Sie den Punkt am Beginn des Dateinamens), zum Beispiel die Datei `spread1.odswird` automatisch als `.spread1.ods.autosave` gespeichert. Die Einstellungen für *Automatische Sicherung* finden Sie im **Einrichtungsdialog**.

## 2.9.1 Vorlagen

Wenn Sie eine Menge ähnlicher Dokumente verwenden, sollten Sie zuerst eine Vorlage erstellen und diese dann als Grundlage für die einzelnen Dokumente verwenden.

Dazu erstellen Sie zuerst ein Dokument mit den gemeinsamen Elementen, dann speichern Sie die Vorlage mit **Datei** → **Vorlage aus Dokument erstellen ...**. Damit öffnen Sie den Dialog **Vorlage erstellen**. Tragen Sie hier einen Namen für Ihre neue Vorlage in das Textfeld **Name**: ein und drücken Sie auf **OK**. Wenn Sie das nächste Mal Calligra Sheets starten oder mit **Datei** → **Neu** mit einem neuen Dokument beginnen, können Sie im Startdialog Ihre Vorlage für das neue Dokument benutzen.

Im Dialog **Vorlage erstellen** können Sie auch ein anderes Bild wählen, das zusätzlich zum Namen der Vorlage im Startdialog angezeigt wird. Sie können ihre Vorlagen unter verschiedenen Gruppennamen abspeichern, die als verschiedene Seiten im Startdialog angezeigt werden.

Vorlagen werden als `.kst` Dateien unter `~/.kde/share/apps/tables/templates/` gespeichert.

## 2.10 Drucken einer Tabelle

Um eine Tabelle zu drucken, wählen Sie im Menü **Datei** → **Drucken ...**. Damit öffnen den Standarddialog **Drucken** von KDE. Hier können Sie unter anderem den Drucker und die Anzahl der Kopien sowie die zu druckenden Seiten eingeben.

In der Standardeinstellung druckt Calligra Sheets alle Daten in der aktuellen Tabelle, aber Sie können den zu druckenden Bereich ändern, indem Sie ihn zuerst markieren und dann **Druckbereich festlegen** aus dem Menü **Format** → **Druckbereich** wählen.

Calligra Sheets druckt alle erforderlichen Seiten für die gesamten Daten der aktuellen Tabelle. Sie können sofort erkennen, wie eine Tabelle in unterschiedliche Seiten für den Druck aufgeteilt wird, indem Sie die Einstellung **Ansicht** → **Seitenumrandungen** aktivieren. Die Ränder jeder Seite im Druckbild werden dann durch rote Linien in der Tabelle gekennzeichnet.

Für eine Vorschau der Druckansicht mit Kopf- und Fußzeilen (siehe unten), wählen Sie **Datei** → **Druckvorschau ...**.

Um das Aussehen der Druckdaten zu verbessern, können Sie die Schriftarten, Farben, Ränder und Größen der Zellen in der Tabelle ändern, lesen Sie dazu den Abschnitt [Tabellen-Formatierung](#).

Sie können auch den Dialog **Seitenlayout** benutzen, indem Sie **Format** → **Seitenlayout ...** auswählen. Hier können Sie die Ausrichtung der gedruckten Seite, die Papiergröße (diese sollte zu Ihrem Drucker passen) und die Größe der Seitenränder einstellen.

Auf der Karteikarte **Tabelle** finden Sie noch weitere Einstellungen. Im Bereich **Druckeinstellungen** können Sie festlegen, ob Gitterlinien, Kommentar- und Formelmarkierungen, Objekte und Diagramme gedruckt werden sollen. Außerdem können Sie im Abschnitt **Wiederholungen auf jeder Seite** Zeilen und Spalten auswählen, die auf jeder Seite gedruckt werden. Im Abschnitt **Skalierung** setzen Sie einen Vergrößerungsfaktor oder geben Seitenbegrenzungen für den Druck an.

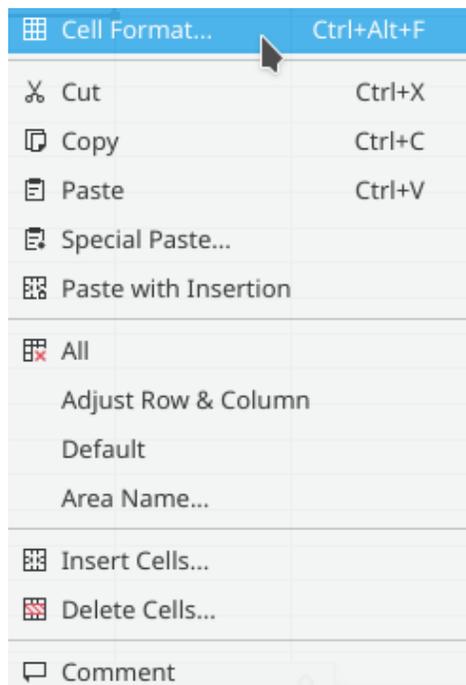
## Kapitel 3

# Tabellenformatierung

Pamela Robert  
Raphael Langerhorst  
Anne-Marie Mahfouf  
Deutsche Übersetzung: Hans-Frieder Hirsch  
Deutsche Übersetzung: Johannes Schwall

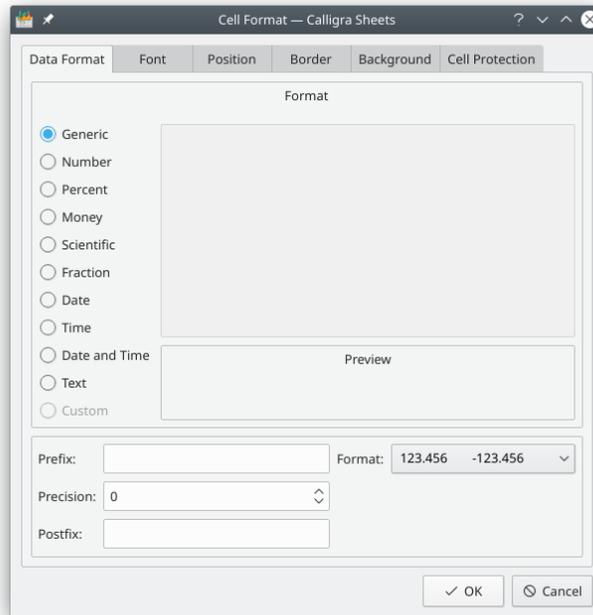
### 3.1 Zellenformat

Um Format und Erscheinungsbild ausgewählter Zellen, Zeilen oder Spalten zu ändern, verwenden Sie die Option **Zellenformat ...** aus dem Menü **Format** oder aus dem Kontextmenü, das Sie mit der rechten Maustaste öffnen.



Damit öffnen Sie den Dialog **Zellenformat** mit folgenden Seiten:

### 3.1.1 Datenformate und Darstellungen



Auf der Seite **Datenformat** des Dialogs **Zellenformat** können Sie einstellen, wie die Werte in den Zellen angezeigt werden.

Im oberen Teil dieser Seite können Sie das Format auswählen, das verwendet wird, um numerische Werte, Daten oder Zeiten anzuzeigen. Eine **Vorschau** zeigt Ihnen, wie das neue Format dargestellt wird.

Für eine Zeile oder Spalte können Sie das gleiche Datenformat einstellen, wenn Sie die Zeile oder Spalte auswählen und mit der rechten Maustaste den Dialog **Zellenformat** öffnen.

**ANMERKUNG**

Die Zahl der Nachkommastellen können Sie für Zahlen im Format **Allgemein**, **Zahl**, **Prozent**, **Währung** oder **Wissenschaftlich** mit dem Symbol **Mehr Dezimalstellen** in der Werkzeugleiste **Format**

vergrößern:

Die Zahl der Nachkommastellen können Sie für Zahlen im Format **Allgemein**, **Zahl**, **Prozent**, **Währung** oder **Wissenschaftlich** mit dem Symbol **Weniger Dezimalstellen** in der Werkzeugleiste **Format**

verringern:

#### Allgemein

Dies ist das Standardformat. Calligra Sheets erkennt automatisch den aktuellen Datentyp aus den Zellen. Als Standard zeigt Calligra Sheets Zahlen, Daten und Zeiten rechtsbündig und alles andere linksbündig innerhalb der Zellen an.

Wenn das Format **Allgemein** nicht zutreffend ist, wählen Sie eines der hier aufgeführten Formate:

#### Zahl

Zahlen werden mit den Einstellungen in den Systemeinstellungen in **Regionales** → **Land/Region & Sprache** → **Zahlen** dargestellt. Zahlen werden in der Voreinstellung rechtsbündig angezeigt.

### Prozent

Enthält die aktuelle Zelle eine Zahl und Sie ändern das Zellenformat von **Allgemein** zu **Prozent**, wird die Zahl in der Zelle mit 100 multipliziert.

Wenn Sie zum Beispiel ein 2 eingeben und dann als Zellenformat **Prozent** einstellen, ändert sich der Wert in der Zelle zu 200 %. Setzen Sie die Formatierung wieder auf **Allgemein** zurück, wird der Wert wieder auf 2 zurückgesetzt.



Zusätzlich können Sie das Symbol **Prozent** in der Werkzeugleiste **Format** benutzen.

### Währung

Mit dem Format **Währung** wird Ihre Eingabe als Währung mit den Einstellungen aus den Systemeinstellungen in **Regionales** → **Land/Region & Sprache** → **Währung** mit dem Währungssymbol und der ausgewählten Anzahl der Dezimalstellen dargestellt.

Zusätzlich können Sie das Symbol **Währungsformat** in der Werkzeugleiste **Format** benutzen, um als Zellenformat die aktuelle Währung einzustellen. 

### Wissenschaftlich

Das Format **Wissenschaftlich** stellt Ihre Zahlen in wissenschaftlicher Schreibweise dar. Die Eingabe von 0,0012 wird zum Beispiel als 1,2E-03 angezeigt. Wechseln Sie wieder zurück zum Format **Allgemein**, wird wieder 0,0012 angezeigt. Um Werte in Zellen in wissenschaftlicher Schreibweise darzustellen, müssen Sie diese Einstellung auswählen, da das Format **Allgemein** eine Eingabe in wissenschaftlicher Schreibweise nicht automatisch so anzeigt.

### Bruch

Das Format **Bruch** zeigt die eingegebene Zahl als Bruch an. Zum Beispiel 0,1 kann als 1/8, 2/16, 1/10 usw. dargestellt werden. Wählen Sie den Nenner des Bruchs im Auswahlfeld auf der rechten Seite des Dialogs. Kann die Zahl nicht als Bruch mit dem gewählten Nenner dargestellt werden, wird die beste Näherung angezeigt. Die Zahl 1,5 zum Beispiel wird mit dem Nenner **Sechzehntel 1/16** wird als richtig als „1 8/16“ angezeigt. Eine Zelle mit der Zahl 1,4 und dem Format **Bruch** mit dem Nenner **Sechzehntel 1/16** wird als „1 6/16“, der besten Näherung, dargestellt.

### Datum

Ein Datum müssen Sie in den Formaten, die in den Systemeinstellungen in **Regionales** → **Land/Region & Sprache** → **Datum & Zeit** eingestellt sind, eingeben. Hier gibt es das Langformat und das Kurzformat für Datum.

Eine beliebige natürliche Zahl N wird in das Datum 30. Dezember 1899 (Beginn der Zählung) plus N Tage übersetzt. Wenn zum Beispiel 100 als Zahl in einer Zelle eingeben ist und Sie das Format **Datum** einstellen, wird „9.4.1900“ in der Zelle angezeigt, das sind 100 Tage nach dem 30. Dezember 1899. Das Startdatum liegt zwei Tage vor dem ersten Tag des Jahrhunderts (ein Fehler in Lotus 123, der auch in Excel übernommen wurde, um kompatibel zu bleiben). Dieser Fehler spielt aber bei den üblichen Berechnung von Daten keine Rolle, wenn Sie zum Beispiel 10 Tage zum 1. November 2000 addieren, erhalten Sie das richtige Ergebnis 10. November 2000.

#### ANMERKUNG

Eine Zelle mit dem Format **Datum** können Sie wie Zellen mit Zahlen nach unten ziehen und die folgenden Zellen werden auch als Datum mit aufsteigenden Tagesdaten ausgefüllt.

### **Zeit**

Diese Format zeigt den Inhalt der Zelle als Zeit an. Zeiten müssen Sie im **Zeitformat**, das in den Systemeinstellungen in **Regionales** → **Land/Region & Sprache** → **Datum & Zeit** eingestellt ist, eingeben. Im Dialog **Zellenformat** können Sie die Darstellung von Zeiten aus einer Liste mit vorgegebenen Formaten wählen. Als Standard wird das Format aus dem Systemeinstellungen verwendet. Wenn der Wert einer Zelle nicht sinnvoll als Zeitformat dargestellt werden kann, wird 00:00 im allgemeinen Format aus den Systemeinstellungen angezeigt.

### **Datum und Zeit**

Diese Format zeigt den Inhalt der Zelle als Datum und Zeit an. Datum und Zeit müssen Sie im **Zeitformat**, das in den Systemeinstellungen in **Regionales** → **Land/Region & Sprache** → **Datum & Zeit** eingestellt ist, eingeben. Im Dialog **Zellenformat** können Sie die Darstellung von Datum und Zeit aus einer Liste mit vorgegebenen Formaten wählen. Als Standard wird das Format aus dem Systemeinstellungen verwendet. Wenn der Wert einer Zelle nicht sinnvoll als Datum und Zeit dargestellt werden kann, wird 00:00 im allgemeinen Format aus den Systemeinstellungen angezeigt.

### **Text**

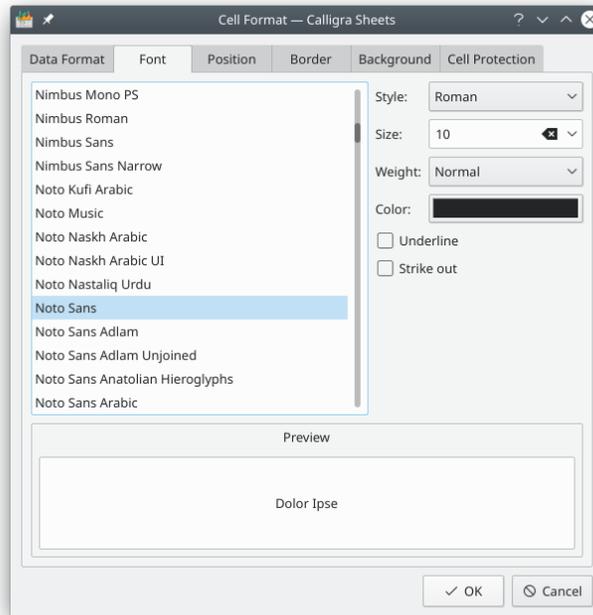
Diese Einstellung formatiert den Zelleninhalt als Text. Das kann nützlich sein, wenn eine Zahl als Text und nicht als Zahl behandelt werden soll, beispielsweise Postleitzahlen. Eine als Text formatierte Zahl wird linksbündig ausgerichtet. Wenn Zahlen als Text formatiert werden, können sie nicht in Berechnungen oder Formeln verwendet werden. Außerdem ändert sich dadurch die Ausrichtung innerhalb der Zellen.

### **Benutzerdefiniert**

Zurzeit noch nicht verfügbar, erst in der nächsten Version.

Im unteren Teil der Seite **Datenformat** können Sie ein **Präfix** wie ein \$-Symbol am Anfang jeder Zelle oder ein **Postfix** wie \$HK am Ende hinzufügen. Sie können auch einstellen, wieviele Stellen nach dem Dezimalkomma für numerische Werte angezeigt werden, und ob positive Werte mit einem führenden +-Zeichen angezeigt werden und ob negative Werte in Rot angezeigt werden sollen.

### 3.1.2 Schriften und Text Einstellungen



Auf der Seite **Schriftart** können Sie Schriftfamilie, **Stil**, **Größe**, **Stärke** und **Farbe** der Schrift für die gewählten Zellen einstellen, dazu noch Optionen für unterstrichenen und durchgestrichenen Text. Im unteren Teil der Seite sehen Sie eine **Vorschau** des ausgewählten Textformats.

Die Standardschrift setzen Sie im Menü **Format** → **Stilverwaltung** mit dem Stil Standard.

#### **Stil:**

Einstellung des Stils der Schrift für die ausgewählten Zellen. Wenn Sie mehrere Zellen mit unterschiedlichen Stilen ausgewählt haben, wird **Unterschiedlich (Keine Änderung)** angezeigt. Ohne Änderung behalten alle Zellen den eingestellten Stil. Ändern Sie den Stil zum Beispiel in **Unecht Kursiv (Roman)**, wird diese Einstellung auf alle ausgewählten Zellen angewendet.

#### **Größe:**

Einstellung der Schriftgröße für die ausgewählten Zellen. Wenn Sie mehrere Zellen mit unterschiedlichen Größen ausgewählt haben, wird **unterschiedlich**, also keine Größe angezeigt. Ohne Änderung behalten alle Zellen die eingestellte Schriftgröße. Ändern Sie die Größe zum Beispiel in **14**, wird diese Einstellung auf alle ausgewählten Zellen angewendet.

#### **Stärke:**

Einstellung der Schriftstärke für die ausgewählten Zellen. Wenn Sie mehrere Zellen mit unterschiedlichen Schriftstärken ausgewählt haben, wird **Unterschiedlich (Keine Änderung)** angezeigt. Ohne Änderung behalten alle Zellen die eingestellte Schriftstärke. Ändern Sie die Schriftstärke zum Beispiel in **Fett**, wird diese Einstellung auf alle ausgewählten Zellen angewendet.

#### **Farbe:**

Einstellung der Farbe für den Text in den ausgewählten Zellen. Klicken Sie auf das Farbfeld, um den KDE-Standarddialog **Farbe auswählen** zu öffnen, in dem Sie eine neue Farbe auswählen können.

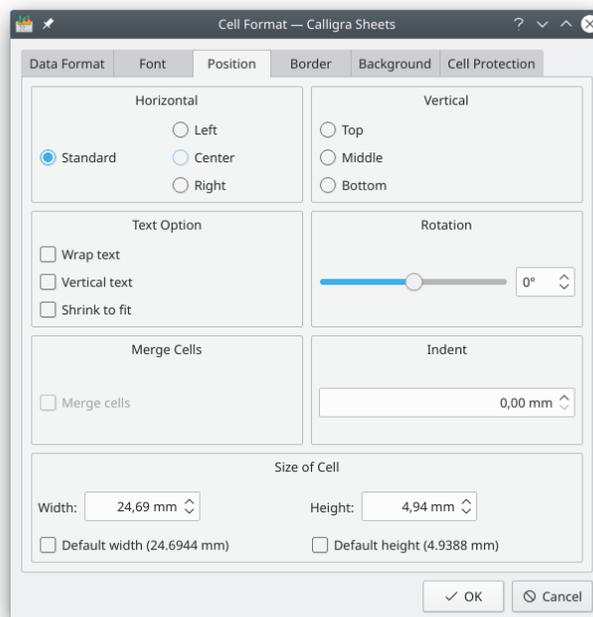
### Unterstrichen

Mit dieser Einstellung wird der Text in den ausgewählten Zellen unterstrichen dargestellt. In der Standardeinstellung ist dies nicht aktiviert.

### Durchgestrichen

Mit dieser Einstellung wird der Text in den ausgewählten Zellen durchgestrichen dargestellt. In der Standardeinstellung ist dies nicht aktiviert.

## 3.1.3 Textposition und -drehung



Auf der Seite **Position** können Sie die Position des Textes innerhalb einer Zelle festlegen, indem Sie entsprechende Einstellungen in den Bereichen **Waagerecht** und **Senkrecht** vornehmen oder indem Sie den Wert für **Einzuge** eingeben. Sie können auch einstellen, dass der Text senkrecht statt waagrecht oder sogar unter einem bestimmten Winkel angezeigt wird.

### Waagerecht

Mit dieser Einstellung wird der Inhalt einer Zelle waagrecht angezeigt. Mit der Voreinstellung **Standard** wird der Inhalt entsprechend dem ausgewählten Datentyp angezeigt. Der Inhalt wird mit der Einstellung **Links** linksbündig, mit **Zentriert** waagrecht mittig in der Zelle und mit **Rechts** rechtsbündig dargestellt.

### Senkrecht

Mit dieser Einstellung wird der Inhalt einer Zelle senkrecht angezeigt. Der Inhalt wird mit der Einstellung **Oben** oben, mit **Zentriert** senkrecht mittig in der Zelle und mit **Unten** unten dargestellt.

### Text-Einstellungen

Diese Einstellungen sind nur für Zellen mit 0 Grad Drehung möglich. Mit **Text umbrechen** werden die Zeilen so umgebrochen, dass der Text in die vorhandene Zellen passt. Ist diese Einstellung nicht ausgewählt, wird der Text ohne Zeilenumbruch angezeigt.

Mit der Einstellung **Senkrechter Text** wird der Text senkrecht ausgerichtet.

### Drehung

Ihr Text wird in dem hier eingestellten Winkel ausgerichtet. Positive Werte drehen den Text im Uhrzeigersinn, negative entgegen dem Uhrzeigersinn.

### Zellen verbinden

Diese Einstellung können Sie auch im Menü **Format** → **Zellen verbinden** aktivieren. Sie müssen dazu mindestens zwei benachbarte Zellen ausgewählt haben. Diese Zellen werden dann zu einer großen Zelle zusammengefasst.

Wenn sie verbundene Zellen ausgewählt haben und diese Einstellung abschalten, erhalten alle Zellen wieder die ursprüngliche Größe wie vor der Verbindung. Im Menü finden Sie diese Einstellung unter **Format** → **Zellen trennen**.

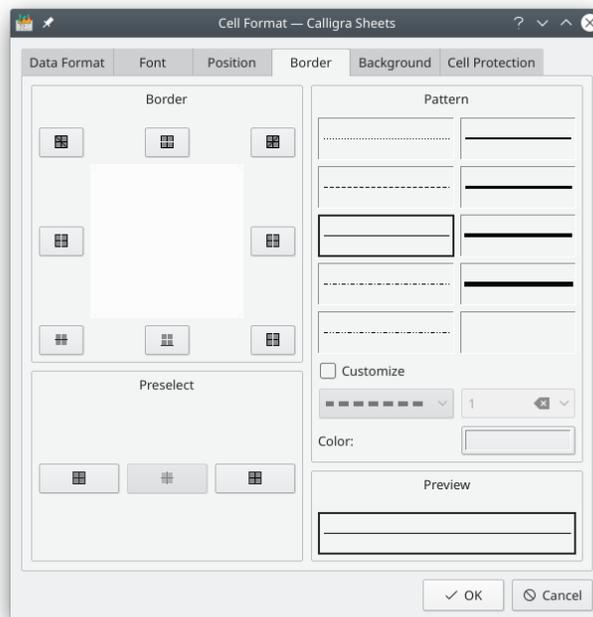
### Einzug

Legt den Wert den Einzugs des Inhalts der Zellen für die Funktion **Einzug vergrößern** oder **Einzug verkleinern** fest. Als Voreinstellung sind diese Aktionen nicht in der Werkzeugleiste.

### Zellengröße

Stellen Sie hier die Größe der Zellen ein, entweder mit benutzerdefinierten Werten für Breite und Höhe oder mit den Standardwerten.

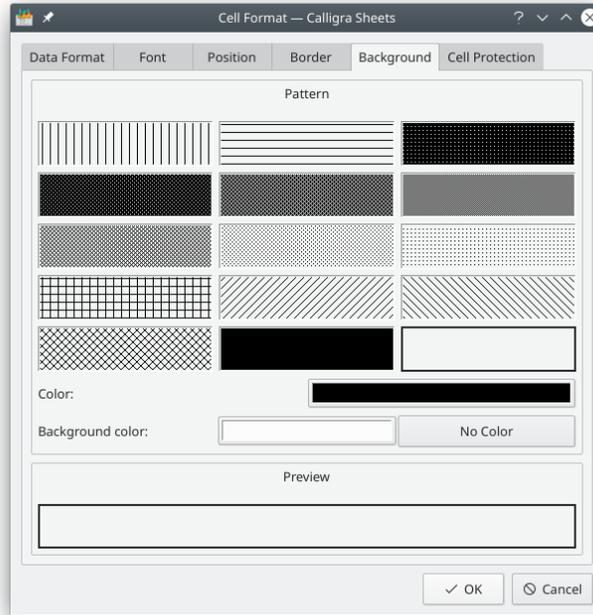
## 3.1.4 Zellenumrandung



Auf der Seite **Umrandung** können Sie das Aussehen der Zellenränder einstellen. Wenn Sie mehr als eine Zelle ausgewählt haben, können Sie unterschiedliche Arten von Rändern zwischen den Zellen und um den ausgewählten Bereich herum angeben.

Wählen Sie zuerst das Muster und die Farbe im Abschnitt **Muster** der Seite **Umrandung** aus und wenden Sie diese Einstellung dann auf die unterschiedlichen Teile des Randes an, indem Sie auf den passenden Knopf im Abschnitt **Umrandung** oder auf einen passenden Knopf der **Vorauswahl** klicken. Der linke Knopf im Abschnitt **Vorauswahl** löscht die vorher ausgewählten Ränder. Beachten Sie, dass Sie auch diagonale Durchkreuzungslinien zu den Zellen hinzufügen können.

### 3.1.5 Zellenhintergrund

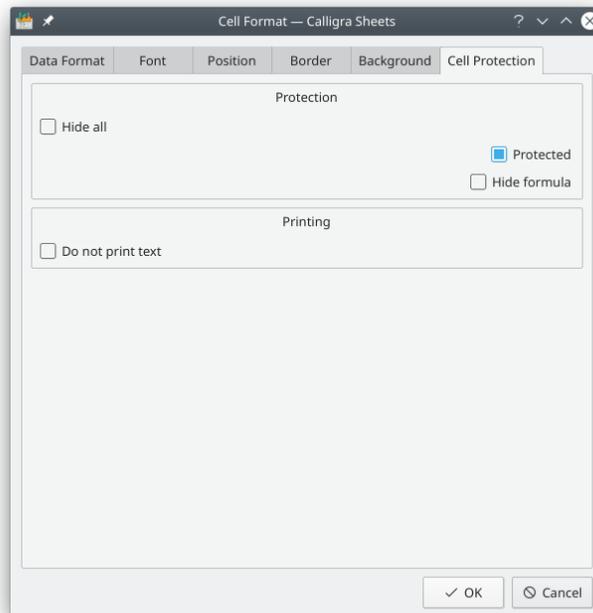


Das Muster und die Farbe des Zellenhintergrundes können Sie auf der Seite **Hintergrund** auswählen.

Wählen Sie das gewünschte **Muster**, dann die Farbe des **Musters** sowie die **Hintergrundfarbe** aus.

Unten auf dieser Seite können Sie eine **Vorschau** des eingestellten Zellenhintergrunds sehen.

### 3.1.6 Zellschutz



Den Schutz für Zellen stellen Sie auf der Seite **Zellenschutz** ein.

Alle Zellen sind in der Standardeinstellung geschützt, d. h. der Inhalt kann nicht geändert werden. Damit dieser Schutz wirksam wird, müssen Sie zusätzlich noch den Schutz der Tabelle mit **Extras** → **Tabelle schützen ...** und der Eingabe eines Passworts einschalten. Sie können zusätzlich auch die Formeln ausblenden, damit die Art der Berechnung nicht erkennbar ist, oder auch die Anzeige des gesamten Inhalts der Zelle mit **Alles ausblenden** abschalten. Auch dafür ist der Schutz der Tabelle erforderlich. Weitere Einzelheiten über diese Einstellungen finden Sie im Abschnitt [Schutz in Calligra Sheets](#).

#### **Alles ausblenden**

Damit wird die Anzeige des Inhalts der Zellen ausgeblendet, aber nur, wenn die Tabelle geschützt ist. Die Einstellung **Alles ausblenden** funktioniert nur dann, wenn die Tabelle geschützt ist. Ob die Zelle geschützt ist oder nicht, spielt dabei keine Rolle.

Wenn Sie **Alles ausblenden** ausgewählt haben, können Sie die Einstellungen **Geschützt** und **Formel ausblenden** nicht mehr verändern, genauso wie in einer geschützten Tabelle. Mit **Alles ausblenden** wird der Wert in der Zelle wie auch die Formeln nicht angezeigt und damit der Inhalt der Zelle geschützt.

#### **Geschützt**

Mit dieser Einstellung wird der Inhalt der Zelle geschützt. Das ist die Standardeinstellung. Sie müssen zusätzlich die gesamte Tabelle mit **Extras** → **Tabelle schützen ...** schützen, damit der Schutz für einzelne Zellen eingeschaltet wird. Der Inhalt einer geschützten Zellen kann nicht geändert werden.

#### **Formel ausblenden**

Wenn diese Einstellung aktiviert ist, wird die Zelle immer noch angezeigt, ihr Inhalt erscheint aber nicht in der **Formelleiste**. „Formel ausblenden“ funktioniert nur für Zellen mit Formeln. Die gesamte Tabelle muss dafür geschützt werden.

#### **Text nicht drucken**

Mit dieser Einstellung wird der Text in den ausgewählten Zellen nicht ausgedruckt. In der Standardeinstellung ist dies nicht aktiviert und der Text in Zellen wird immer gedruckt.

## **3.2 Bedingte Zellenattribute**

Sie können die Darstellung der Werte in einer Zelle abhängig von dem darin enthaltenen Wert ändern lassen. Das ist z. B. nützlich, wenn Sie Calligra Sheets verwenden, um die Übersicht über Ihre Haushaltungskosten zu behalten und alle Einzelposten größer als tausend Euro hervorheben wollen.

Dazu wählen Sie die zunächst die Zellen aus, dann **Bedingte Stile ...** im Menü **Format**. Damit öffnen Sie den Dialog **Bedingte Stile**. hier können Sie einstellen, dass sich der Schrifttyp und die Farbe einer Zelle ändern, wenn deren Werte eine oder mehrere Bedingungen erfüllen. Beachten Sie, dass die zweite und dritte Bedingung nur dann angewendet wird, wenn die vorhergehende Bedingung nicht zutrifft.

Wählen Sie **Leeren** → **Bedingte Stile** im Menü **Bearbeiten**, um die bedingten Zellenattribute von ausgewählten Zellen zu löschen.

## **3.3 Ändern der Zellengröße**

Auf der Seite **Position** im Dialog **Zellenformat** können Sie die Größe der ausgewählten Zellen ändern. Wenn Sie die Höhe einer einzelnen Zelle ändern, wird die Höhe für alle Zellen in der

zugehörigen Zeile ebenfalls geändert. Auch wenn Sie die Breite einer Zelle ändern, gilt das für die ganze Spalte.

Sie können auch Zeilen oder Spalten auswählen und dann **Zeilenhöhe ändern ...** oder **Spaltenbreite ändern ...** aus dem Kontextmenü der rechten Maustaste oder im Untermenü **Format** → **Zeile** oder **Format** → **Spalte** wählen.

Wenn Sie den Mauszeiger so über die Zeilenzahlen links von der Tabelle bewegen, dass die Spitze des Mauszeigers genau auf die Linie zwischen zwei Zeilen zeigt, ändert sich der Zeiger zu zwei waagerechten Linien mit zwei Pfeilen, die nach oben und nach unten zeigen. Wenn der Mauszeiger so aussieht, können Sie die linke Maustaste gedrückt halten und durch eine Auf- oder Abwärtsbewegung des Mauszeigers die Höhe der entsprechenden Zeile verändern. Entsprechend kann man die Spaltenbreiten verändern, indem man den Mauszeiger auf die senkrechte Linie zwischen zwei Spaltenbuchstaben bewegt und die Spaltengrenze hin- und herzieht.

Um die Zeilenhöhe oder die Spaltenbreite soweit zu verringern, dass noch der gesamte Inhalt angezeigt wird, wählen Sie eine ganze Zeile oder Spalte aus und klicken mit der rechten Maustaste auf die Zeilen- oder Spaltenbeschriftung. Wählen Sie im Kontextmenü **Zeile anpassen** oder **Spalte anpassen**. Die Zeile oder Spalte wird dadurch auf die erforderliche Mindestgröße gebracht. Sie können auch eine einzelne Zelle oder einen Zellenbereich auswählen und **Zeile und Spalte anpassen** im Kontextmenü der rechten Maustaste oder im Menü **Format** auswählen.

Sie können einer Anzahl von nebeneinander liegenden Zeilen oder Spalten die gleiche Breite oder Höhe geben, indem Sie diese auswählen und danach **Format** → **Zeile** → **Zeilen ausgleichen** oder **Format** → **Spalte** → **Spalten ausgleichen** wählen.

### 3.4 Zellen verbinden

Es ist häufig nützlich, eine Zelle zu haben, die über zwei oder mehr Spalten reicht oder mehr als eine Zeile hoch ist. Solche Zellen können erzeugt werden, indem man zwei oder mehr Zellen zu einer einzigen verbindet. Wählen Sie die zu verbindenden Zellen aus und wählen Sie dann **Format** → **Zellen verbinden**.

Um diesen Verbindung aufzuheben, wählen Sie zunächst die verbundenen Zellen aus und wählen dann **Zellen trennen** im Menü **Format** aus.

### 3.5 Zeilen und Spalten ausblenden

Eine fertige Tabelle sieht häufig besser aus, wenn Sie Zellen mit Zwischenberechnungen ausblenden, und wenn nur die Zellen mit wichtigen Daten und Ergebnissen angezeigt werden.

In Calligra Sheets können Sie ausgewählte Zeilen oder Spalten ausblenden, indem sie **Zeilen ausblenden** oder **Spalten ausblenden** im Menü **Format** → **Zeile** oder im **Format** → **Spalte** oder im Kontextmenü der rechten Maustaste wählen. Ausgeblendete Zeilen und Spalten werden nicht auf dem Bildschirm angezeigt und auch nicht ausgedruckt.

Das Ausblenden von Zellen bietet auch einen gewissen Schutz vor unbeabsichtigten Änderungen.

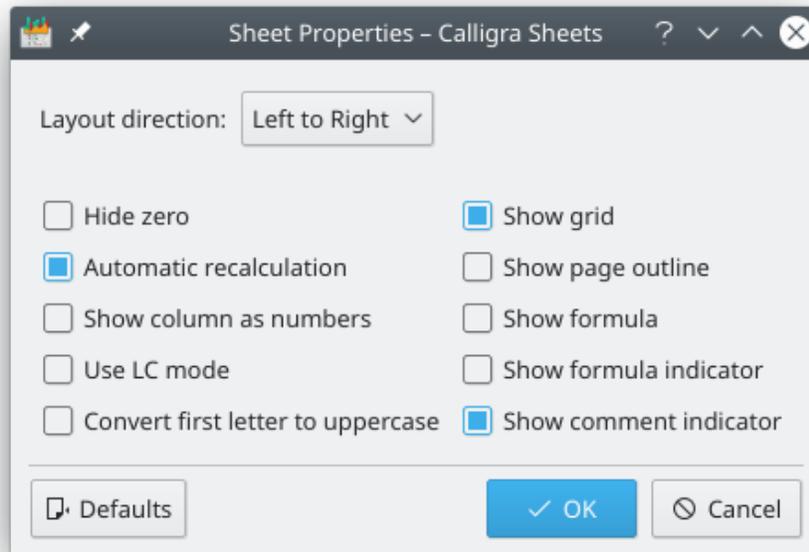
Um eine Zeile oder Spalte wieder anzuzeigen, wählen Sie **Zeile** → **Zeilen anzeigen ...** oder **Spalte** → **Spalten anzeigen ...** im Menü **Format**. Im Dialogfenster können Sie jede beliebige Zeilenzahl wählen, die angezeigt werden soll.

### 3.6 Tabellen-Eigenschaften

Die Eigenschaften der gewählten Tabelle können Sie entweder mit der Auswahl von **Tabelleigenschaften** aus dem Kontextmenü der rechten Maustaste oder im Menü **Format** → **Tabelle**

→ **Tabelleneigenschaften** bearbeiten. Den Dialog **Tabelleneigenschaften** können Sie aber nur öffnen, wenn die Tabelle und das Dokument nicht geschützt sind.

Hier können Sie alle Eigenschaften der aktuellen Tabelle einstellen. Mit **OK** werden die Änderungen übernommen, mit **Voreinstellungen** werden die Standardeinstellungen wieder hergestellt.



### Layout-Ausrichtung:

Einstellung der Richtung in der Tabelle. Als Standard liegt die erste Spalte der Tabelle links. Wenn Sie **Rechts nach Links** einstellen, befindet sich die erste Spalte der Tabelle rechts und alle anderen werden links eingefügt.

### Nullen ausblenden

Ist diese Einstellung aktiviert, wird jede Zelle mit einer Null als leere Zelle angezeigt.

### Automatische Neuberechnung

Ist diese Einstellung aktiviert, werden Formeln automatisch neu berechnet, wenn sich der Wert einer bezogenen Zelle ändert.

### Spalten als Nummern anzeigen

Ist diese Einstellung aktiviert, werden die Spaltenüberschriften als Nummern und nicht als Buchstaben angezeigt. Buchstaben sind die Standardeinstellung.

### LC-Modus verwenden

Ist diese Einstellung aktiviert, wird der Zellenverweis am linken Ende der Formelleiste im LC-Modus (z. B. L2C3) und nicht in seiner normalen Form angezeigt. Diese Einstellung wird selten verwendet.

### Ersten Buchstaben groß schreiben

Ist diese Einstellung aktiviert, wird der erste Buchstabe jedes Textes automatisch groß geschrieben.

### Gitter anzeigen

Ist diese Einstellung aktiviert, wird das Gitter (die Ränder der Zellen) angezeigt. Dies ist die Standardeinstellung. Um das Gitter auszublenden, deaktivieren Sie diese Einstellung.

**Seitenränder anzeigen**

Ist diese Einstellung aktiviert, werden die Seitenränder in Ihrer aktuellen Tabelle angezeigt. Standardmäßig sind sie ausgeblendet. Das Anzeigen der Seitenränder ist nützlich, wenn Sie Ihre Tabelle drucken wollen.

**Formel anzeigen**

Ist diese Einstellung aktiviert, werden Formeln anstelle der Ergebnisse in den Zellen angezeigt.

**Formelmarkierung anzeigen**

Ist diese Einstellung aktiviert, werden Zellen mit einer Formel mit einem kleinen blauen Dreieck in der linken oberen Ecke markiert. Dies hilft Ihnen, Zellen mit Formeln zu schützen.

**Kommentarmarkierung anzeigen**

Ist diese Einstellung aktiviert, werden Zellen mit einem Kommentar mit einem kleinen roten Dreieck in der rechten oberen Ecke markiert.

## Kapitel 4

# Calligra Sheets für Fortgeschrittene

Pamela Robert  
Anne-Marie Mahfouf  
Deutsche Übersetzung: Hans-Frieder Hirsch  
Deutsche Übersetzung: Johannes Schwall

### 4.1 Reihen

Wenn Sie eine Tabelle erstellen, müssen Sie häufig eine Reihe von Werten, wie z.B. 10, 11, 12, ... in einer Zeile oder in einer Spalte eingeben. Es gibt verschiedene Möglichkeiten, wie Sie dies in Calligra Sheets tun können.

Für eine einfache kurze Reihe wie 5, 6, 7, 8, ... ist die „Ziehen und Kopieren“-Methode die einfachste Möglichkeit. Tragen Sie den Anfangswert in die Startzelle und den folgenden Wert der Reihe in eine angrenzende Zelle ein. Dann wählen Sie beide Zellen aus und verschieben Sie den Mauszeiger so, dass er über dem kleinen Quadrat an der unteren rechten Ecke des markierten Bereiches steht. Der Cursor wird zu einem diagonalen Pfeil mit zwei Spitzen. Halten Sie dann die linke Maustaste gedrückt, während Sie die Zellen nach unten oder nach rechts ziehen.

Die Schrittgröße wird als die Differenz zwischen den zwei Anfangswerten errechnet, die Sie eingetragen haben. Beispiel: Wenn Sie **4** in Zelle A1 eingeben und **3, 5** in A2, dann beide Zellen auswählen und sie nach unten ziehen und kopieren, ist die Schrittgröße der Wert in A2 minus des Wertes in A1, in diesem Fall also -0,5. Damit erhalten Sie die Reihe 4, 3,5, 3, 2,5, 2, ...

Die „Ziehen und Kopieren“-Methode wird sogar mit Reihen fertig, in der der Schrittwert nicht ein konstanter Wert, sondern selbst eine Reihe ist. Wenn Sie also mit 1, 3, 4, 6 beginnen, wird die *Ziehen und Kopieren* Methode die Zahlen zur folgenden Reihe ergänzen: 1, 3, 4, 6, 7, 9, 10, 12, ... Der Schrittwert in diesem Beispiel ist die folgende Reihe: 2, 1, 2, 1, ...

Calligra Sheets kennt auch einige besondere „Reihen“, zum Beispiel die Wochentage. Geben Sie **Freitag** in eine Zelle ein (achten Sie auf den großen Anfangsbuchstaben) und Ziehen und Kopieren Sie dann nach unten. Wählen Sie **Extras** → **Benutzerdefinierte Listen ...** um zu sehen, welche besonderen Reihen verfügbar sind, und welche Sie vielleicht selbst erstellen wollen.

Wenn Sie eine Zelle auswählen und **Reihe ...** aus dem Menü **Einfügen** wählen, sehen Sie den Dialog **Reihe**. Dieses ist nützlich für das Erzeugen von Reihen, die zu lang sind, um mit der Ziehen-und-Kopieren-Methode erstellt zu werden, oder für das Erzeugen geometrischer Reihen wie 1, 1,5, 2,25, 3,375, ..., wo der Schrittwert, 1,5 in diesem Fall, als Multiplikator verwendet wird.

Wenn die gewünschte Art der Reihe für die vorhergehenden Methoden zu schwierig ist, verwenden Sie eine Formel und ziehen und kopieren Sie diese. Z. B., um eine Reihe mit den Werten 2, 4, 16, 256, ... zu erzeugen, geben Sie **2** in A1 und **= A1\*A1** in A2 ein und ziehen und kopieren die Zelle A2 nach unten.

## 4.2 Formeln

### 4.2.1 Vorhandene Funktionen

In Calligra Sheets gibt es eine Vielzahl von mathematischen und anderen **Funktionen**, die Sie als Formel in einer Zelle verwenden können. Eine Formel wird angezeigt oder eingegeben, indem Sie eine Zelle wählen, und dann **Funktion ...** aus dem Menü **Einfügen** wählen. Damit öffnen Sie den Dialog **Funktion**.

Wählen Sie aus der Liste links im Dialogfeld die Funktion, die Sie verwenden möchten. Auf der Seite **Hilfe** werden dann eine Beschreibung, der Rückgabebetyp, Syntax, Parameter und Beispiele für diese Funktion angezeigt. Zusätzlich finden Sie häufig auch Verweise auf ähnliche Funktionen. Klicken Sie auf den Knopf Pfeil runter, um eine Funktion in das Textfeld unten im Dialog einzufügen.

Jetzt wird die Seite **Parameter** angezeigt. Hier können Sie die Werte für die gewählte Funktion eingeben. Wenn Sie einen normalen Zahlen- oder Textwert für einen Parameter eintragen möchten, geben Sie ihn in das passende Textfeld auf der Seite **Parameter** ein. Um einen Verweis auf eine Zelle anstatt eines Wertes einzutragen, klicken Sie mit der linken Maustaste auf das entsprechende Textfeld auf der Seite **Parameter** und klicken Sie dann auf die Zielzelle in der Tabelle.

Anstatt die Seite **Parameter** zu verwenden, können Sie Zellenverweise wie **B6** auch direkt in das Textfeld unten im Dialog **Funktion** eintragen. Wenn ein Ausdruck mehrere Parameter hat, werden diese Parameter jeweils mit einem Semikolon (;) getrennt.

Drücken Sie auf **OK**, um die Funktion in die aktuelle Zelle einzufügen und den Dialog **Funktion** zu schließen.

Sie können natürlich auch ohne den Dialog **Funktion** die ganze Formel direkt in das Hauptbearbeitungsfeld im **Zelleneditor** der Werkzeugoptionen eingeben. Groß- oder Kleinschreibung spielt für Funktionsnamen keine Rolle. Denken Sie daran, dass alle Formeln mit dem Symbol = beginnen müssen.

### 4.2.2 Logische Vergleiche

Logische Funktionen wie IF(), AND() und OR() haben Parameter, die die logischen (booleschen) Werte „Wahr“ oder „Falsch“ annehmen können. Diese Art von Werten kann durch andere logische Funktionen wie ISEVEN() oder durch den Vergleich von Werten in Tabellenzellen mit den Vergleichsausdrücken produziert werden, die in der folgenden Tabelle aufgeführt werden.

Ausdruck	Beschreibung	Beispiel
==	ist gleich	<b>A2==B3</b> ist Wahr, wenn der Wert in A2 gleich dem Wert in B3 ist
!=	ist nicht gleich	<b>A2!=B3</b> ist Wahr, wenn der Wert in A2 nicht gleich dem Wert in B3 ist
<>	ist nicht gleich	Ist das gleiche wie <b>A2!=B3</b>
<	ist kleiner als	<b>A2&lt;B3</b> ist Wahr, wenn der Wert in A2 kleiner als der Wert in B3 ist
<=	ist kleiner oder gleich	<b>A2&lt;=B3</b> ist Wahr, wenn der Wert in A2 kleiner oder gleich dem Wert in B3 ist
>	ist größer als	<b>A2&gt;B3</b> ist Wahr, wenn der Wert in A2 größer als der Wert in B3 ist

>=	ist größer oder gleich	<b>A2&gt;=B3</b> ist Wahr, wenn der Wert in A2 größer oder gleich dem Wert in B3 ist
----	------------------------	--

Wenn Sie also **=IF (B3>B1 ; "GRÖßER" ; "")** in eine Zelle eingeben, zeigt die Zelle GRÖßER an, wenn der Wert in B3 größer als der in B1 ist, anderenfalls zeigt sie nichts an.

### 4.2.3 Absolute Zellenverweise

Wenn eine Formel einen Zellverweis enthält, wird dieser Verweis normalerweise geändert, wenn die Zelle in eine andere Zelle in der Tabelle kopiert wird. Um dieses Verhalten zu verhindern, setzt man ein \$-Symbol vor den Buchstaben der Spaltennummer, der Zeilennummer oder vor beide.

- Wenn A1 die Formel **=D5** enthält, wird der Inhalt der Zelle beim Kopieren nach B2 in **=E6** geändert (das normale Verhalten).
- Wenn A1 die Formel **=\$D5** enthält, wird der Inhalt der Zelle beim Kopieren nach B2 in **=D6** geändert (der Spaltenbuchstabe wurde nicht geändert).
- Wenn A1 die Formel **=D\$5** enthält, wird der Inhalt der Zelle beim Kopieren nach B2 in **=E5** geändert (die Zeilennummer wurde nicht geändert).
- Wenn A1 die Formel **=\$D\$5** enthält, wird der Inhalt der Zelle beim Kopieren nach B2 nicht verändert, der Inhalt ist weiterhin **=D5** (weder der Spaltenbuchstabe noch die Zeilennummer wurden geändert).

Wenn Sie einen Zellenverweis in einer Formel eingeben oder ändern, können Sie mit der Funktionstaste **F4** durch diese vier Möglichkeiten schalten. Durch wiederholtes Drücken von **F4** wird also aus dem Zellenverweis D5 **\$D\$5**, **\$D5**, **D\$5** und dann wieder **D5**.

**Benannte Zellenbereiche** können in einer ähnlichen Weise benutzt werden, um feste Zellenverweise in eine Formel einzugeben.

## 4.3 Berechnungen mit „Einfügen Spezial“

Vielleicht wollen Sie manchmal einen einzelnen Wert zu einer Anzahl von Zellen addieren oder von ihnen subtrahieren, oder Zellen mit einem Wert multiplizieren oder sie alle durch einen einzelnen Wert teilen. Die Funktion **Einfügen spezial ...** lässt Sie das schnell und einfach tun.

Zuerst tragen Sie den Wert, mit dem Sie eine Berechnung auf dem Zellenbereich durchführen wollen, in eine freie Zelle in ihrer Tabelle ein und **kopieren** ihn. Wählen Sie dann den Bereich der Zellen aus, die Sie ändern wollen und wählen Sie **Einfügen spezial ...** aus dem Menü **Bearbeiten** oder aus dem Kontextmenü der rechten Maustaste. Wählen Sie dann im Dialogfenster einen der Einträge **Addition**, **Subtraktion**, **Multiplikation** oder **Division** aus dem **Operation**-Abschnitt aus.

Sie können auch unterschiedliche Modifikationswerte auf unterschiedliche Zeilen oder Spalten des Zielbereichs anwenden, indem Sie einen Bereich, der die gewünschten Modifikationsfaktoren enthält, kopieren, bevor Sie den Zielbereich auswählen und **Einfügen spezial ...** ausführen. Z. B. wenn Sie **5** in Zelle A1 und **10** in B1 eingeben, beide Zellen auswählen und **kopieren**, danach die Zellen A10 bis D15 markieren und im Dialog**Einfügen spezial ...** **Addition** wählen, wird 5 zu den Zellen A10:A15 und 10C:C15 sowie 10 zu den Zellen B10:B15 und D10:D15 hinzugefügt.

Beachten Sie, dass ein Modifikationswert eine Formel sowie ein einfacher numerischer Wert sein kann. Wenn es eine Formel ist, dann passt Calligra Sheets die Zellenverweise genau wie beim normalen **Einfügen** an.

## 4.4 Feldformeln

Calligra Sheets erlaubt die Benutzung von Formeln, deren Ergebnis eine Matrix oder ein Bereich von Werten ist. Normalerweise wird nur der erste Wert in einer Zelle angezeigt. Wenn Sie die gesamte Matrix dargestellt werden soll, benutzen Sie den Kurzbefehl **Strg-Alt-Eingabe** bei der Bearbeitung einer Formel. Dann wird die Formel in eine Feldformel umgewandelt und die benachbarten Zellen werden, soweit erforderlich, belegt.

Zellen, die Teil einer Feldformel sind, sind für die Bearbeitung gesperrt.

## 4.5 Zielwertsuche

Calligra Sheets kann verwendet werden, um algebraische Ausdrücke wie  $x + x^2 = 4$  oder die Frage *Für welchen Wert von x ist die Formel x plus x zum Quadrat gleich 4 ?* zu lösen.

Für dieses Beispiel könnten Sie **=A2+A2\*A2** in A1 eingeben und dann entweder unterschiedliche Werte in A2 eingeben, bis das Resultat in A1 so nahe an 4 ist, wie Sie es wünschen, oder Sie nutzen vorzugsweise Calligra Sheetss **Zielwertsuche ...**, die automatisch den Wert in einer Zelle verändert, um zu versuchen, den Wert in einer anderen Zelle so nahe wie möglich an seinen Zielwert anzunähern.

Die Zielwertsuche starten Sie mit **Zielwertsuche ...** im Menü **Daten**. Damit öffnen Sie einen Dialog, in dem Sie die Adresse der Zielwertzelle (**A1** in diesem Fall) in das Feld **Setze Zelle:** eintragen. Den Zielwert selbst (**4**) tragen Sie dann in das Feld **Auf Wert:** ein und die Adresse der zu verändernden Zelle (**A2**) in das Feld **Durch Änderung von Zelle:**. Beachten Sie, dass Sie einen Ausgangswert in die Zelle eingetragen haben müssen, die angepasst werden soll, bevor Sie die **Zielwertsuche** starten.

Wenn Sie den Knopf **OK** drücken, wird die Berechnung gestartet. Wenn sie beendet ist und einen Zielwert gefunden hat, drücken Sie den Knopf **OK**, um das Ergebnis zu übernehmen. Oder drücken Sie auf **Abbrechen**, um den ursprünglichen Wert zu behalten.

## 4.6 Pivot-Tabellen

Mit Calligra Sheets können **Pivot-Tabellen** aus den Daten der aktuellen Tabelle konstruiert werden.

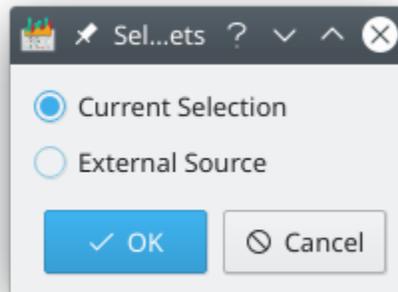
Diese Funktion wird mit **Pivot ...** aus dem Menü **Daten** aufgerufen. Es folgt ein Beispiel für die Erstellung einer Pivot-Tabelle.

Folgende Daten werden in diesem Beispiel verwendet.

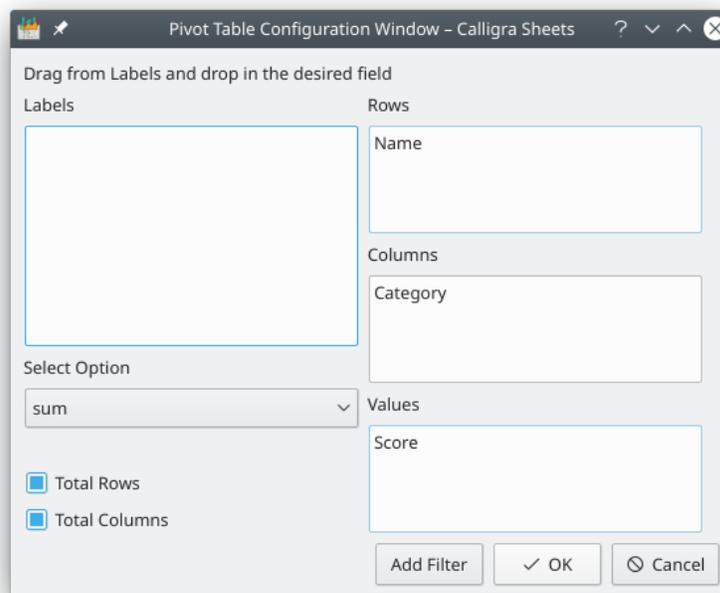
	A	B	C
1	Name	Category	Score
2	Jigar	Science	90
3	Smith	Math	80
4	John	Science	95
5	Smith	Science	60
6	Jigar	Math	81
7	John	Math	90

Um eine Pivot-Tabelle mit den gewünschten Anforderungen zu erstellen, wählen Sie **Daten** → **Pivot ...**

Im damit aufgerufenen Dialog kann die Quelle der Daten für die Pivot-Tabelle ausgewählt werden. Zur Auswahl stehen das aktuelle Arbeitsblatt oder externe Quellen wie eine Datenbank oder eine ODS-Datei.

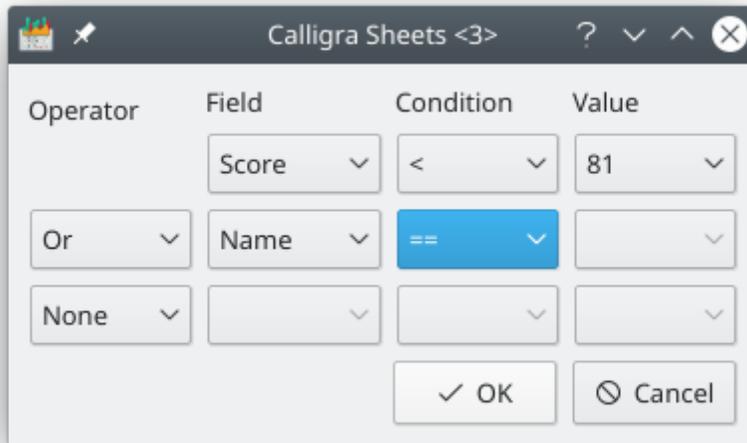


In diesem Dialog kann der Benutzer die Pivot-Tabelle anpassen. Die Überschriften der der Spalten in den Quelldaten werden als Beschriftungen verwendet, die als Arbeitsfelder dienen. Die Beschriftungen können in eine der drei Bereiche (**Reihen**, **Spalten** oder **Werte**) gezogen und abgelegt werden, um die Pivot-Tabelle zu erstellen. Ihre Auswahl können Sie mit **Ziehen/Ablegen zurücksetzen** wieder löschen.



In diesem Beispiel wird *Name* in **Zeilen**, *Kategorie* in **Spalten** und *Bewertung* in das Feld **Werte** gezogen.

Mit dem Knopf **Filter hinzufügen** können Sie einen Dialog öffnen, um die gewünschten Daten zu filtern. In diesem Dialog können Sie mehrere Filter auf der Basis der Spaltenüberschriften und deren Beziehungen zueinander (**Und** or **Oder**) definieren. Damit können Sie die Ausgabe in großem Umfang anpassen.



**Summe der Zeilen** und **Summe der Spalten**: Aktivieren Sie diese Ankreuzfelder, um automatisch Summen der zugehörigen Zeilen und Spalten in der Pivot-Tabelle anzuzeigen.

## 4.7 Arbeiten mit mehreren Tabellen

Wenn Sie ein neues leeres Dokument mit Calligra Sheets anlegen, enthält es bereits eine Anzahl von leeren Tabellen. Diese Anzahl der Tabellen wird durch die ausgewählte Vorlage festgelegt.

**Einfügen** → **Tabelle** fügt eine weitere Tabelle zum Dokument hinzu.

Sie können auch zwischen den Tabellen umschalten, indem Sie die Tasten **Strg+Bild ab** verwenden, um die folgende Tabelle anzuzeigen, und **Strg+Bild auf**, um die vorhergehende Tabelle anzuzeigen.

Tabellen haben standardmäßig die Namen *Tabelle1*, *Tabelle2*, ... Sie können einem Blatt dadurch einen anderen Namen geben, dass Sie auf den entsprechenden Karteireiter mit der rechten Maustaste klicken, **Tabelle umbenennen ...** wählen und einen neuen Namen für die Tabelle eingeben.

Um eine Tabelle aus dem Dokument zu löschen, verwenden Sie **Tabelle entfernen** im Kontextmenü, das Sie mit einem Klick der rechten Maustaste auf den Karteireiter der Tabelle, die Sie entfernen wollen, öffnen.

Andere Eintragungen im **Format** → **Tabelle**-Untermenü erlauben Ihnen, ein Blatt auf beinahe gleiche Art und Weise anzuzeigen oder auszublenden, wie Zeilen und Spalten angezeigt und ausgeblendet werden können.

Wenn Sie in einer Formel in einer Tabelle einen Verweis auf eine Zelle in einer anderen Tabelle eingeben wollen, muss der Zellenverweis mit dem Tabellennamen beginnen, gefolgt von einem Ausrufezeichen (!). Wenn Sie zum Beispiel **=Tabelle2!A2** in eine Zelle in Blatt 1 eingeben, erhält diese Zelle den Wert von A2 aus Tabelle2. Beachten Sie, dass bei Tabellennamen zwischen Groß- und Kleinschreibung unterschieden wird.

### 4.7.1 Daten zusammenfassen

Angenommen Sie haben ein Dokument aus mehreren Tabellen mit ähnlichen Daten, aber zum Beispiel für verschiedene Monate eines Jahres und wollen in einer zusätzlichen Tabelle die zusammengehörenden Werte aus den anderen Tabellen zusammenfassen (z. B. Summe oder Durchschnitt ermitteln).

Diese Aufgabe durch die Funktion **Zusammenfassen ...** aus dem Menü **Daten** erleichtert.

Dadurch wird der Dialog **Zusammenfassen** geöffnet.

Für jede der Quelltabellen tragen Sie einen Verweis auf den gewünschten Datenbereich im Feld **Verweis:** ein. Dann klicken Sie auf **Hinzufügen**, dadurch wird der Verweis in die Liste **Eingegebene Verweise** aufgenommen. Der Verweis sollte den Namen der Tabelle mit den Quelldaten enthalten, wie z. B. **Januar!A1:A10**, und kann automatisch eingegeben werden, indem Sie den gewünschten Bereich in der entsprechenden Tabelle auswählen.

Wenn Sie die Verweise auf alle Quelldaten-Tabellen eingegeben haben, wählen Sie in der Ziel-tabelle die Zelle aus, an der die linke obere Ecke der zusammengefassten Daten erscheinen soll. Dann wählen Sie aus dem Auswahlfeld **Funktion:** die gewünschte Funktion aus und drücken dann den Knopf **OK**.

Wenn Sie auf den Knopf **Details >>** im Dialog klicken und **Daten kopieren** auswählen, werden die Ergebnisse der Berechnung in die Zielzellen eingesetzt, anstatt der Formeln zur Berechnung der Werte.

## 4.8 Diagramme einfügen

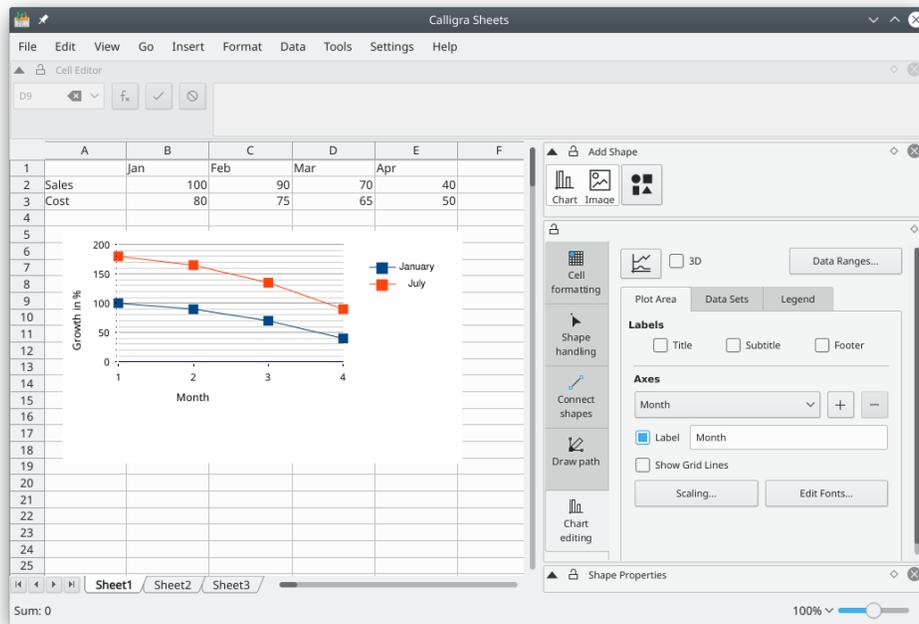
Sie können Diagramme in eine Tabelle einfügen, um Ihre Daten grafisch darzustellen.

Zuerst öffnen Sie **Objekt hinzufügen** im Menü **Einstellungen** → **Andockbare Dialoge**.

Wählen Sie zuerst den Bereich der Zellen aus, die die darzustellenden Daten enthalten und wählen dann **Diagramm** Im andockbaren Dialog **Objekt hinzufügen**. Ziehen Sie mit gedrückter linker Maustaste über die Tabelle, um den Bereich für die Anzeige des Diagramms festzulegen. Sie müssen dabei nicht so genau sein, da die Größe und Lage des Diagramms jederzeit leicht geändert werden kann. Wenn Sie die Maustaste loslassen, erscheint der Dialog **Diagramm-Einstellungen**.

Im Datenbereich ist bereits die ausgewählte Zellenbereich eingetragen. Sie können nun die erste Zeile oder Spalte als Beschriftung angeben und die Datensätze zeilenweise oder spaltenweise benutzen. Wenn Sie den Knopf **OK** betätigen, verschwindet der Dialog und Sie sehen das Diagramm einbettet in der Tabelle.

Wählen Sie nun **Diagrammbearbeitung** aus **Werkzeuge** und bearbeiten die Eigenschaften des Diagramms wie den Typ, Beschriftungen und Achsen in den **Diagrammbearbeitung**.



Um das eingebettete Diagramm zu verschieben, zu vergrößern oder zu verkleinern oder um es zu löschen, wechseln Sie zum **Werkzeug für Objekte** und klicken dann irgendwo in das Diagramm. Es wird jetzt mit einem grünen Rand und mit je einem kleinen gelben Quadrat an jeder Ecke und in der Mitte jeder Seite angezeigt.

Wenn Sie den Mauszeiger über eins der Quadrate bewegen, wird als Pfeil mit zwei Spitzen angezeigt. Sie können die Größe des Diagramms nun ändern, indem Sie eins dieser Quadrate mit der gedrückten linken Maustaste ziehen. Um das Diagramm zu löschen, klicken Sie mit der rechten Maustaste auf eins der Quadrate und wählen Sie **Löschen**.

Um das Diagramm zu verschieben, bewegen Sie den Cursor hinein. Der Cursor wird zu einem Kreuz, halten Sie jetzt die linke Maustaste gedrückt und Sie können das Diagramm dorthin bewegen, wo Sie es haben wollen.

Um das Aussehen des Diagramms wieder zu normalisieren, klicken Sie einfach irgendwo außerhalb des Diagramms in die Tabelle.

Um das Format des Diagramms selbst zu ändern, doppelklicken Sie mit der linken Maustaste innerhalb des Diagrammbereichs. Die **Diagrammbearbeitung** wird dann angezeigt und sie können dann damit die Eigenschaften des Diagramms ändern.

## 4.9 Externe Daten einfügen

Sie können Daten aus einer Textdatei oder aus der Zwischenablage in eine Tabelle einfügen, indem Sie zuerst die Zelle auswählen, in der Sie das obere linke Element der einzufügenden Daten einfügen wollen und dann **Aus Textdatei ...** oder **Aus Zwischenablage ...** aus dem Untermenü **Einfügen** → **Externe Daten** wählen.

In beiden Fällen wird Calligra Sheets annehmen, dass die Daten im Format CSV vorliegen und ein Dialogfeld öffnen, in dem Sie einstellen können, wie die Daten aus der Datei oder der Zwischenablage ausgelesen und in den Tabellenzellen angeordnet werden.

Wenn die Unterstützung dafür in Ihrem System vorhanden ist, kann Calligra Sheets auch Daten aus einer SQL-Datenbank in eine Tabelle einfügen. Das machen Sie, indem Sie **Einfügen** → **Externe Daten** → **Aus Datenbank** auswählen.

## 4.10 Zellen verknüpfen

Eine Tabellenzelle kann mit einer Aktion verknüpft werden, damit das linke Klicken auf der Zelle z. B. Ihren Webbrowser öffnet. Damit eine Zelle auf diese Weise funktioniert, wählen Sie **Einfügen** → **Verknüpfung ...**. Das öffnet das **Verweis einfügen**-Dialogfeld, in dem Sie zwischen vier Arten von Verknüpfungen wählen können:

- Eine mit **Internet** verknüpfte Zelle wird versuchen Ihren Standard-Webbrowser mit der URL zu öffnen, die Sie in dem **Internetadresse**-Textfeld des **Verweis einfügen**-Dialoges eingegeben haben. Das könnte zum Beispiel **http://www.calligra.org** sein.
- Wenn Sie auf eine Zelle klicken, die eine **Mail**-Verknüpfung enthält wird Ihr E-Mail-Programm geöffnet, das die Adresse verwendet, die Sie im **E-Mail**-Textfeld als „An:“ Adresse eingegeben haben. Zum Beispiel **anon@example.com**.
- Eine mit **Datei** verknüpfte Zelle enthält den Pfad zu einer Datei oder einem Ordner, wie er in dem Textfeld **Speicherort für Datei** eingegeben wurde und versucht die Datei oder den Ordner mit einer geeigneten Anwendung zu öffnen, wenn auf sie geklickt wird.
- Eine verknüpfte Zelle enthält einen Verweis auf eine Calligra Sheets-Zelle, die in das Feld **Zelle oder benannter Bereich** eingetragen wird. Wenn Sie mit der linken Maustaste auf eine solche Zelle klicken, wechselt Calligra Sheets zur Zielzelle.

Bei allen vier Arten von Verknüpfungszellen muss ein Text im Feld **Textanzeige** des Dialogs **Verknüpfung einfügen** eingegeben werden. Das ist der Text, der in der Tabelle in der Verknüpfungszelle erscheint

## 4.11 Gültigkeitsprüfung

Calligra Sheets kann automatisch die Gültigkeit der eingegebenen Daten mit einer Anzahl von Kriterien überprüfen und eine Warnmeldung ausgeben, wenn die Daten unzulässig sind.

Um diese Funktion einzuschalten, wählen Sie die Zellen aus, die überwacht werden sollen und wählen Sie **Daten** → **Gültigkeit ...**. Das öffnet den Dialog **Gültigkeit** von Calligra Sheets, der drei Karteikarten enthält.

Auf der Seite **Kriterien** wählen Sie im Auswahlfeld **Zulassen:** den Datentyp aus, der als gültig betrachtet werden soll. Dann wählen Sie den gültigen Wertebereich im Auswahlfeld **Daten:** und geben passende Werte in eine oder beide der Bearbeitungsfelder ein.

Wenn Sie damit fertig sind, wechseln Sie zur Karteikarte **Fehlerbenachrichtigung**. Hier können Sie die Art der Fehlerbenachrichtigung (**Anhalten**, **Warnung** oder **Information**) wählen, die erscheint, wenn ein unzulässiger Wert eingegeben wurde, und Sie können hier den Titel des Benachrichtigungsfensters und den Text der Benachrichtigung festlegen.

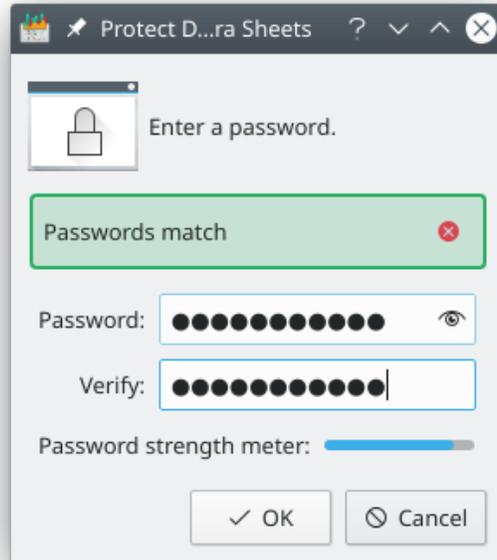
Beachten Sie, dass diese Funktion nur Daten überprüft, die Sie in die Zelle eingeben. Eine Möglichkeit zur Überprüfung der Werte von Formelzellen finden Sie im Abschnitt [bedingte Zellenattribute](#) in diesem Handbuch.

## 4.12 Schutz

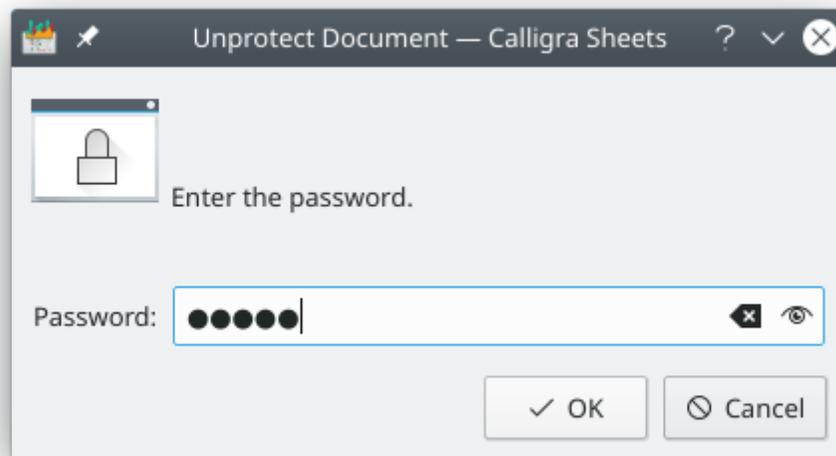
### 4.12.1 Ein Dokument schützen

Schutz des Dokuments bedeutet, das ein Benutzer keine Tabellen einfügen oder löschen kann. Die Zellen in einem geschützten Dokument sind nicht geschützt.

Wählen Sie **Extras** → **Dokument schützen ...**. Damit öffnen Sie einen Dialog zur Eingabe des Passwortes. Der **Passwortstärkeanzeiger** zeigt Ihnen, ob Ihr Passwort sicher genug ist. Je länger der angezeigte Balken, desto sicherer ist das Passwort.



Diese Passwort brauchen Sie, um den Schutz des Dokuments wieder aufzuheben.



Wenn ein Dokument geschützt ist, können Sie nicht:

- Eine Tabelle umbenennen
- Eine Tabelle einfügen
- Eine Tabelle löschen
- Eine Tabelle ausblenden

- Eine Tabelle einblenden
- Die Eigenschaften der Tabelle anzeigen
- Zellen verbinden oder trennen

#### 4.12.2 Tabellen schützen

Der Schutz einer Tabelle schaltet den Schutz für den Inhalt aller geschützten Zellen und Objekte ein. Einzelne Zellen oder Zellenbereiche in einer geschützten Tabelle können trotzdem ungeschützt sein, siehe das [nächste Kapitel](#).

Um eine Tabelle zu schützen, wählen Sie **Extras** → **Tabelle schützen ...**. Damit öffnen Sie einen Dialog zur Eingabe des Passwortes. Der **Passwortstärkeanzeiger** zeigt Ihnen, ob Ihr Passwort sicher genug ist. Je länger der angezeigte Balken, desto sicherer ist das Passwort.

Diese Passwort brauchen Sie, um den Schutz der Tabelle wieder aufzuheben.

Wenn eine Tabelle geschützt ist, können Sie nicht:

- Ein Objekt oder Diagramm einfügen
- Eine Zelle formatieren
- Eine Zeile oder Spalte einfügen
- Den Inhalt einer Zelle bearbeiten und ändern
- Den Inhalt einer Tabelle ändern

##### ANMERKUNG

Der Schutz einer Tabelle ist besonders sinnvoll, um das unbeabsichtigte Löschen von Formeln zu verhindern.

#### 4.12.3 Schutz von Zellen und Zellenbereichen

##### WARNUNG

Zellenschutz ist in der Voreinstellung für alle Zellen eingeschaltet und wird dann wirksam, wenn Sie die Tabelle schützen. Wenn Sie die Voreinstellung nicht ändern und die Tabelle schützen, sind alle Zellen geschützt.

Wollen Sie nur bestimmte Zellen schützen, muss der als Standard eingestellte Schutz für alle anderen Zellen ausgeschaltet werden. Wollen Sie den Benutzern zum Beispiel die Eingabe in die meisten Zellen erlauben, müssen Sie die Eigenschaft **Geschützt** für diese Zellen ausschalten und für die übrigen Zellen, die nicht geändert werden sollen, den Zellenschutz einschalten. Das erfordert drei Schritte: Schutz für alle Zellen aufheben, die zu schützenden Zellen auswählen und dafür den Schutz aktivieren und dann die gesamte Tabelle schützen.

Um den Schutz für alle Zellen aufzuheben:

- Wählen Sie die gesamte Tabelle mit dem Mauszeiger.
- Wählen Sie in der Menüleiste **Format** → **Zellenformat ...**
- Öffnen Sie dann im Dialog die Seite **Zellenschutz**.

- Wählen Sie **Alles ausblenden** und wählen Sie **Geschützt** ab, um den Schutz aller Zellen zu entfernen. Alle Zellen sind jetzt nicht mehr geschützt.

Um einen Zellenbereich oder einen Bereich nicht-zusammenhängender Zellen zu schützen:

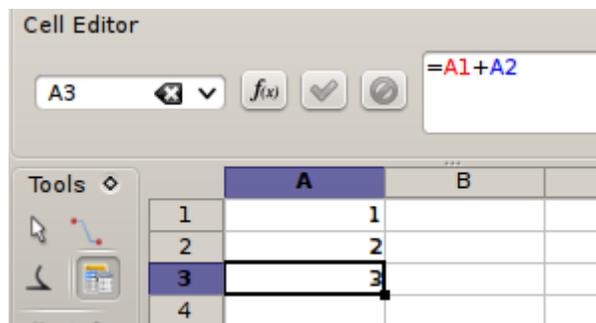
- Wählen einen Bereich von Zellen, die geschützt werden sollen, oder benutzen Sie die Taste **Strg**, um nicht-zusammenhängende Zellen auszuwählen.
- Wenn Sie alle gewünschten Zellen ausgewählt haben, wählen Sie **Format** → **Zellenformat ....**
- Öffnen Sie dann im Dialog die Seite **Zellenschutz**.
- Klicken Sie auf der Seite **Zellenschutz** auf **Geschützt** und dann auf **OK**.

Wenn der Schutz für alle gewünschten Zellen ausgewählt ist, muss die gesamte Tabelle geschützt werden, damit die gewählten Zellen tatsächlich geschützt sind:

- Wählen Sie **Extras** → **Tabelle schützen ....**
- Im Dialog geben Sie ein sicheres Passwort ein und bestätigen Sie es durch eine zweite Eingabe. Dann Klicken Sie auf **OK**.
- Geschützte Zellen in einer geschützten Tabellen können nicht ohne Aufhebung des Schutzes für die gesamte Tabelle bearbeitet werden, außerdem sind keine Änderungen in der Tabelle möglich. Sie können zum Beispiel keine Zeilen oder Spalten einfügen, die Breite von Spalten ändern oder eingebettete Diagramme erstellen.

#### 4.12.4 Zellenformeln ausblenden

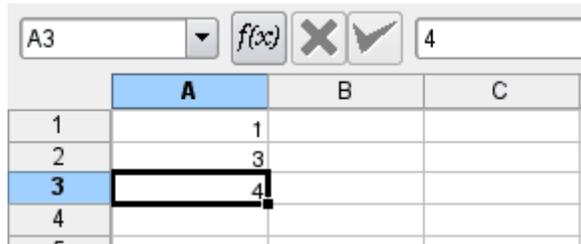
Sie können Ihre Formeln ausblenden, sodass Sie von anderen nicht gelesen werden können. In der Standardeinstellung ist jede Zelle geschützt, wird aber nicht ausgeblendet. Denken Sie aber daran, dass diese Einstellungen wirkungslos sind, solange die Tabelle selbst nicht geschützt ist.



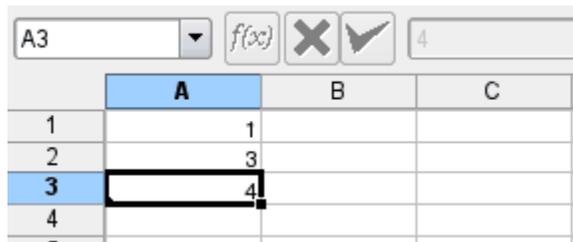
Um Formeln in Zellen auszublenden, wählen Sie die gewünschte Zelle oder nicht-zusammenhängende Zellen mit **Strg** und dann im Menü **Format** → **Zellenformat ....** Gehen Sie im Dialog Zellenformat zur Seite **Formel Ausblenden**. Wenn Sie dann die Tabelle schützen, wird das Ergebnis der Berechnung angezeigt, nicht aber die Formel selbst.

Jetzt müssen Sie noch die Tabelle schützen: Wählen Sie **Extras** → **Tabelle schützen ....**, um den Dialog **Tabellenschutz** anzuzeigen. Geben Sie hier ein sicheres Passwort zweimal ein, um zu verhindern, dass andere den Schutz der Tabelle aufheben können.

Wenn Sie **Formel Ausblenden** aktiviert haben, **Geschützt** aber nicht, wird die Formel nach Schutz der Tabelle ausgeblendet, aber Sie können den Inhalt der Zelle ändern.



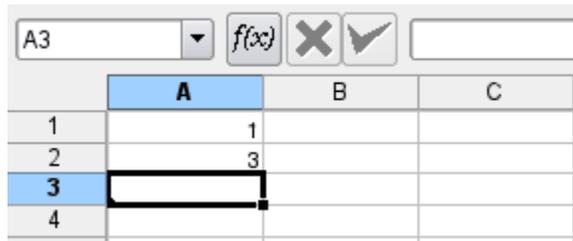
Wenn Sie **Formel Ausblenden** und **Geschützt** aktiviert haben, wird die Formel nach Schutz der Tabelle ausgeblendet und Sie können den Inhalt der Zelle nicht ändern.



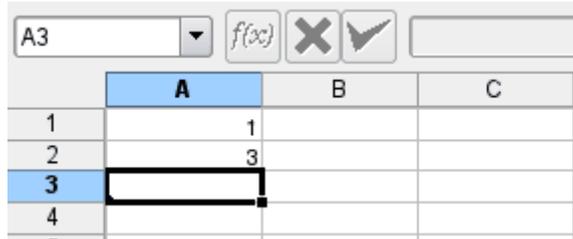
Denken Sie daran, das es sehr leicht ist, das Passwort für eine geschützte Tabelle zu knacken, für wirkliche Sicherheit brauchen Sie ein besseres Konzept.

#### 4.12.5 Formel und Ergebnis ausblenden

Sie können sowohl die Formel als auch den Inhalt einer Zelle ausblenden, indem Sie **Alles ausblenden** auf der Seite Zellschutz des Dialogs auswählen, den Sie mit **Format** → **Zellenformat** ... öffnen. Im Bildschirmfoto unten ist die Zelle selbst nicht geschützt (**Geschützt** ist nicht aktiviert), daher können Sie den Inhalt der Zelle ändern.



Dann ist die Zelle selbst geschützt und kann nicht überschrieben werden.



### 4.13 Andere Funktionen

#### 4.13.1 Benannte Zellen und Zellenbereiche

Sie können einer Zelle oder irgendeinem Bereich eines Blattes einen Namen wie **Toller Name** geben, indem Sie die Zelle oder den Bereich auswählen und dann **Bereichsname ...** im Kontextmenü der rechten Maustaste wählen. Das öffnet den Dialog **Bereichsname**, in dem Sie einen beliebigen Namen eintragen können.

Sie können eine Zelle oder einen Bereich auch benennen, indem Sie erst einen Bereich auswählen und dann den gewünschten Namen in das Adressfeld links neben dem Formeleingabefeld eingeben und damit die Zellenadresse überschreiben, die normalerweise dort steht.

Wenn Sie hier einen Namen eingeben, der bereits verwendet wird, ändert sich Calligra Sheets Auswahlmarkierung so, dass die entsprechenden Zellen markiert und angezeigt werden.

**Daten** → **Benannte Bereiche** zeigt Ihnen eine Liste der vorhandenen Namen. Sie können den Cursor in Calligra Sheets dann zu einem der benannten Bereiche bewegen oder Bereichsnamen bearbeiten und entfernen.

Benannte Zellen sind in Formeln besonders nützlich als eine Alternative zu [absoluten Zellenverweisen](#), da die Namen anstatt der normalen Zellenverweise verwendet werden können und nicht geändert werden, wenn die Zelle, die die Formel enthält, kopiert wird. Wenn ein Name auf diese Art verwendet wird, sollte er in einfachen Anführungszeichen eingeschlossen werden.

Z. B. wenn Sie Zelle A1 den Namen **Fred** gegeben haben, dann können Sie eine Formel wie = **'Fred'** + 2 in eine andere Zelle eingeben, die immer das Resultat des Hinzufügens von 2 zu dem Wert in A1 ergibt, egal wohin die Formelzelle kopiert wird.

Beachten Sie dass Zellen- und Bereichsnamen so behandelt werden als seien sie in Kleinbuchstaben geschrieben.

### 4.13.2 Zellenkommentare

Eine Zelle kann einen Textkommentar enthalten, der beim Arbeiten mit der Tabelle angesehen werden kann, der aber nicht gedruckt und normalerweise nicht angezeigt wird.

Um einen Kommentar einzugeben, wählen Sie die Zelle aus und wählen Sie **Kommentar ...** aus dem Kontextmenü der rechten Maustaste oder aus dem Menü **Einfügen** und geben Sie Ihren Kommentar im Dialog **Zellenkommentar** ein.

Um sich den Kommentar anzeigen zu lassen, bewegen Sie den Mauszeiger über die Zelle. Die Kommentar erscheint dann genau wie eine Kurzinfo.

Wenn Sie **Kommentarmarkierung anzeigen** im Dialog **Tabellen-Eigenschaften** auswählen, werden alle Zellen, die einen Kommentar enthalten, durch ein kleines rotes Dreieck in der rechten oberen Ecke gekennzeichnet.

Um diesen Dialog zu öffnen, klicken Sie mit der rechten Maustaste auf den Karteikartenreiter der Tabelle unten im Hauptfenster und wählen **Tabellen-Eigenschaften** aus dem Kontextmenü. Oder wählen Sie diesen Eintrag aus dem Menü **Format** → **Tabelle**.

Um einen Kommentar aus einer Zelle zu entfernen, wählen Sie **Kommentar entfernen** im Kontextmenü der rechten Maustaste oder wählen Sie **Bearbeiten** → **Löschen** → **Kommentar**.

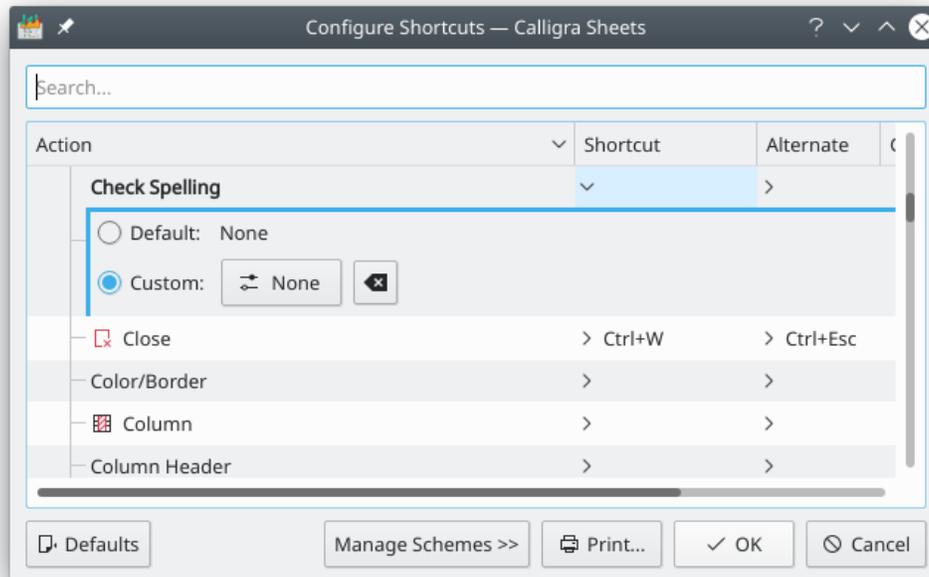
## Kapitel 5

# Kurzbefehle und Werkzeugleisten für Calligra Sheets einrichten

Pamela Robert  
Deutsche Übersetzung: Hans-Frieder Hirsch  
Deutsche Übersetzung: Johannes Schwall

### 5.1 Kurzbefehle

Um die Kurzbefehle für Calligra Sheets zu ändern, wählen Sie **Einstellungen** → **Kurzbefehle festlegen ...**. Damit öffnen Sie den folgenden Dialog:



Suchen Sie in der Liste nach Aktionen, für die Sie Kurzbefehle festlegen oder ändern möchten. Wählen Sie die gewünschte Aktion aus, indem Sie mit der linken Maustaste auf den Namen klicken. Durch die Eingabe des Namens der Aktion in der Suchleiste oben im Dialog können Sie

die gewünschte Aktion leichter finden. Dann ändern Sie den Kurzbefehl, indem Sie auf einen der Auswahlknöpfe **Keine**, **Standard** oder **Benutzerdefiniert** klicken.

Drücken Sie jetzt die gewünschte Tastenkombination für den Kurzbefehl, zum Beispiel **Strg+Umschalt+S**.

## 5.2 Werkzeuggesteilen

Calligra Sheets hat sechs Werkzeuggesteilen: **Datei**, **Bearbeiten**, **Navigation**, **Format**, **Schriftart** und **Farbe/Umrändung**. Jede dieser Werkzeuggesteilen kann angezeigt oder ausgeblendet werden, je nachdem was Sie im Menü **Einstellungen** festgelegt haben.

Wenn die Werkzeuggesteilen nicht gesperrt sind, können Sie wählen, ob eine Werkzeuggesteile **Oben**, **Links**, **Rechts** oder **Unten** im Calligra Sheets-Fenster angezeigt wird, indem Sie mit der rechten Maustaste auf die Werkzeuggesteile klicken. Damit öffnen Sie die **Werkzeuggesteilen-Einstellungen** und können einen Eintrag aus dem Menü **Ausrichtung** wählen. In den **Werkzeuggesteilen-Einstellungen** können Sie auch einstellen, ob in der Werkzeuggesteile nur Symbole, nur Text oder beides angezeigt werden sollen, sowie die Größe der Symbole festlegen.

Sie können eine Werkzeuggesteile auch bewegen, indem Sie den Mauszeiger auf der vertikalen Leiste am linken Ende einer Werkzeuggesteile platzieren und mit gedrückter linker Maustaste die Werkzeuggesteile in die gewünschte Position ziehen. Wenn Sie eine Werkzeuggesteile auf diese Weise in eine Position außerhalb der Begrenzungen des Calligra Sheets-Fensters ziehen, erhalten Sie eine unabhängige Werkzeuggesteile, die an kein Element des Calligra Sheets-Fensters mehr gebunden ist. Um eine unabhängige Leiste wieder in eine übliche Position zu bringen, klicken Sie mit der rechten Maustaste auf ihre Titelleiste um das **Werkzeuggesteilen-Einstellungen** aufzurufen. Dann wählen Sie eine der Optionen im Untermenü **Ausrichtung**.

Wenn Sie **Werkzeuggesteilen einrichten ...** aus dem Menü **Einstellungen** wählen, öffnet sich ein Dialog, in dem Sie Knöpfe zu den Werkzeuggesteilen Calligra Sheetshinzufügen oder von diesen entfernen können.

Im Dialog **Werkzeuggesteilen einrichten** wählen Sie zuerst eine Werkzeuggesteile im Auswahlfeld **Werkzeuggesteile**. In der Liste **Angezeigte Aktionen**: rechts im Dialog, sehen Sie die Knöpfe, die derzeit in der Werkzeuggesteile zu sehen sind. Sie können einen Knopf entfernen, indem Sie ihn im Fenster auswählen und dann auf den Pfeil-nach-links-Knopf klicken. Sie können ihn auch im Fenster bewegen, indem Sie auf die Pfeil-nach-oben- oder Pfeil-nach-unten-Knöpfe klicken. Einen neuen Knopf fügen Sie hinzu, indem Sie ihn in der Liste **Verfügbare Aktionen**: auswählen und dann auf den Pfeil-nach-rechts-Knopf klicken.

## Kapitel 6

# Der Einrichtungsdialog von Calligra Sheets

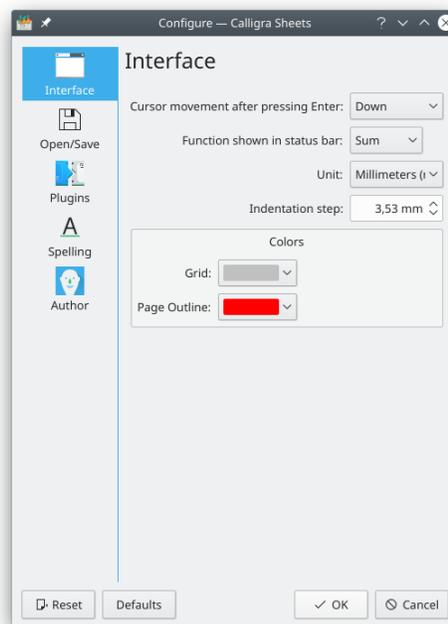
Pamela Robert

Deutsche Übersetzung: Hans-Frieder Hirsch

Deutsche Übersetzung: Johannes Schwall

Wenn Sie **Einstellungen** → **Calligra Sheets einrichten ...** wählen, wird ein Dialog mit mehreren Seiten geöffnet. Klicken Sie auf ein Symbol links im Dialog, um eine Seite auszuwählen. Auf diesen Seiten können Sie viele Einstellungen von Calligra Sheets ändern.

### 6.1 Oberfläche



**Cursor-Bewegung nach Drücken der Eingabetaste:**

Wenn Sie eine Zelle ausgewählt haben und danach die Taste **Eingabe** drücken, wird der Cursor **Nach unten**, **Nach oben**, nach **Rechts**, nach **Links** oder **Nach unten, erste Spalte** oder nicht verschoben, je nachdem, was Sie hier ausgewählt haben.

**In der Statusleiste gezeigte Funktion:**

Hier können Sie wählen, welche Berechnung standardmäßig vom **Ergebnisfeld in der Statuszeile** angezeigt wird.

**Einheit:**

Wählen Sie die Standardeinheit, die in Ihren Tabellen verwendet wird.

**Einzugsschritt:**

Legt den Wert den Einzugs des Inhalts der Zellen für die Funktion **Einzug vergrößern** oder **Einzug verkleinern** fest. Als Voreinstellung sind diese Aktionen nicht in der Werkzeugleiste.

**Alle Navigationstasten bei der Zellenbearbeitung verwenden**

Alle Navigationstasten, d. h. die Cursor-Tasten. die Tasten Bild ab/auf, die Tabulatortaste und die Rückschritttaste werden beim Editieren direkt in einer Zelle für die Bearbeitung benutzt. Ist diese Einstellung aktiviert, werden diese Tasten für die Navigation im Editor verwendet. Ist diese Einstellung abgeschaltet, werden die Tasten für die Zellennavigation verwendet.

In diesem Bereich des Einrichtungdialogs können Sie die Farbe des Gitterlinien in der Tabelle wählen. Wenn das Gitter nicht angezeigt werden soll, deaktivieren Sie **Gitter anzeigen** im Dialog **Format** → **Tabelle** → **Tabelleneigenschaften**.

Hier können Sie auch die Farbe der Linien wählen, die in der Tabelle die Seitenränder der gedruckten Seiten anzeigen, wenn **Seitenumrandungen** im Menü **Ansicht** ausgewählt ist.

Wählen Sie **Benutzerdefiniert** aus dem Farbfeld, um den Standarddialog von KDE zur **Farbe auswählen** zu öffnen.

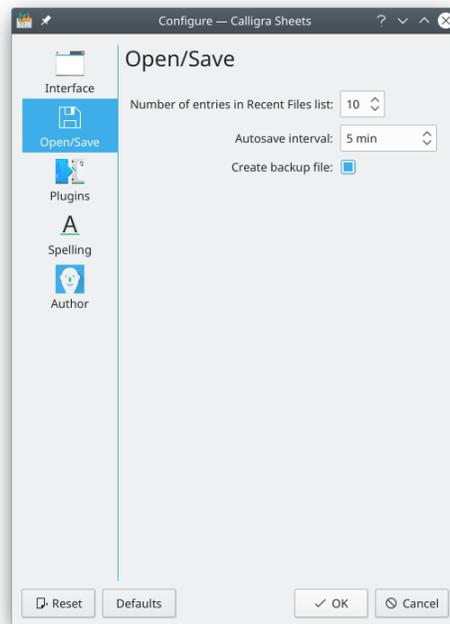
**Gitter**

Klicken Sie hier, um die Gitterfarbe, d. h. die Farbe der Ränder aller Zellen, auszuwählen.

**Seitenumrandung**

Wenn im Menü **Ansicht** → **Seitenränder anzeigen** ausgewählt ist, werden die Seitenränder angezeigt. Klicken Sie auf das Farbfeld, um für die Ränder eine andere Farbe als das voreingestellte Rot auszuwählen.

## 6.2 Öffnen/Speichern



### Anzahl der Einträge in der Liste der zuletzt geöffneten Dateien:

Bestimmt die maximale Anzahl der Dateinamen, die im Menü **Datei** → **Zuletzt geöffnete Dateien** angezeigt werden.

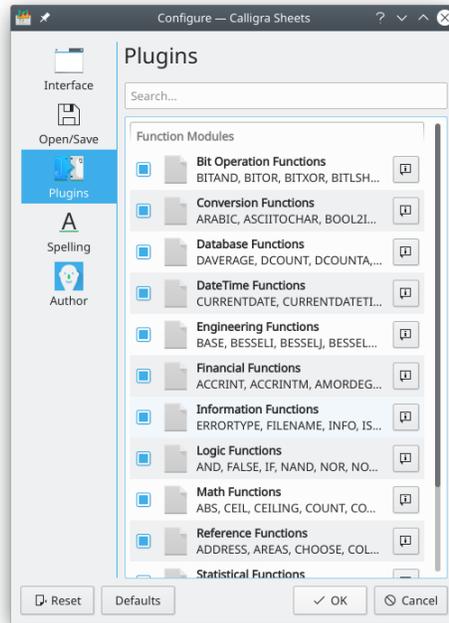
### Intervall für automatisches Speichern:

Hier können Sie den Zeitraum zwischen automatischen Sicherungen festlegen oder die automatische Sicherung ganz ausschalten. Dazu wählen Sie **Keine automatische Sicherung** angezeigt wird.

### Sicherungsdatei erstellen:

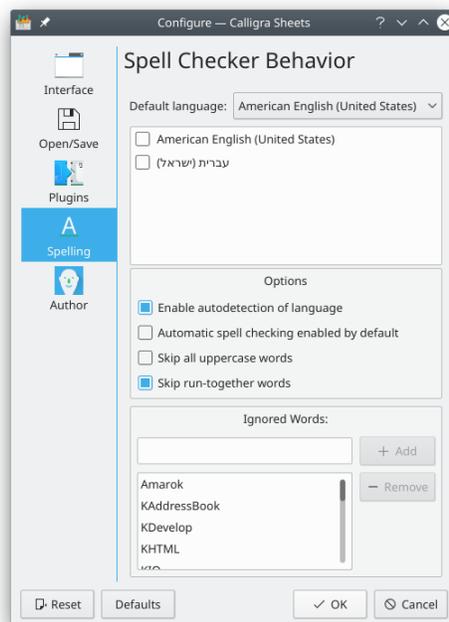
Aktivieren Sie diese Einstellung, wenn Sicherungskopien erstellt werden sollen. Standardmäßig ist diese Einstellung eingeschaltet.

## 6.3 Module



Kreuzen Sie ein Modul in der Liste an, um es zu aktivieren. Der Dialog **Über** wird angezeigt, wenn Sie auf den Info-Knopf rechts neben dem Element in der Liste klicken.

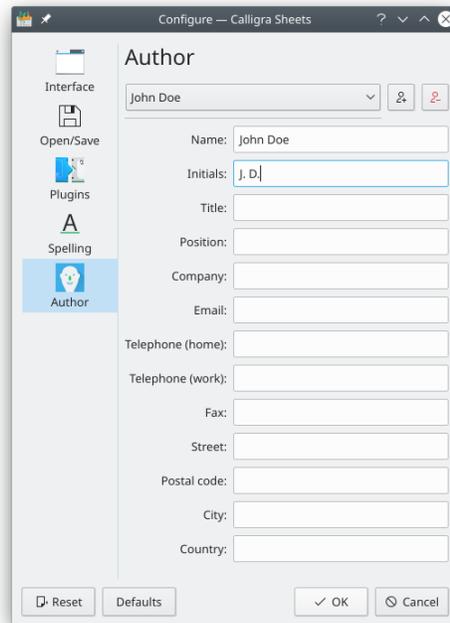
## 6.4 Rechtschreibung



Auf dieser Seite können Sie das Verhalten der Rechtschreibprüfung in Calligra Sheets einstellen.

Weitere Information dazu finden Sie in der Dokumentation zur [Rechtschreibprüfung](#).

## 6.5 Autor



Wählen Sie aus dem Auswahlfeld eine Autorprofil für Ihr Dokument.

Um neue Profile hinzuzufügen oder vorhandene zu löschen, benutzen Sie die Knöpfe rechts neben dem Auswahlfeld mit den Autorprofilen.

## Kapitel 7

# Befehlsreferenz

Pamela Robert  
Deutsche Übersetzung: Hans-Frieder Hirsch  
Deutsche Übersetzung: Johannes Schwall

### 7.1 Das Menü Datei

**Datei → Neu (Strg+N)**

Erstellt ein neues Dokument.

**Datei → Öffnen ... (Strg+O)**

Öffnet ein vorhandenes Dokument.

**Datei → Zuletzt geöffnete Dateien**

Öffnet ein vorhandenes Dokument, das Sie aus einer Liste kürzlich bearbeiteten Dokumenten auswählen.

**Datei → Speichern (Strg+S)**

Speichert das Dokument.

**Datei → Speichern unter ...**

Speichert das aktuelle Dokument unter einem neuen Namen und in einem anderen Format. Möchten Sie mit dem aktuellen Dokument weiterarbeiten und dabei den Namen und das Format beibehalten, benutzen Sie den Menüeintrag **Exportieren ...**

**Datei → Erneut laden**

Lädt das Dokument erneut.

**Datei → Importieren ...**

Importiert andere Dokumente.

Tabellen mit Daten werden häufig in Textdateien gespeichert. Die Werte stehen dann in diesen Dateien durch Komma, Leerzeichen, Tabulator oder andere Buchstaben getrennt in einer Zeile, z. B. *123, 456, 789, abcd, efgh*. Solche Dateien werden gewöhnlich „CSV“-Dateien (Comma Separated Values, durch Komma getrennte Werte) genannt, obwohl das trennende Zeichen möglicherweise kein Komma ist.

Wenn Sie mit Calligra Sheets eine Textdatei öffnen, wird angenommen, dass die Datei im CSV-Format ist. Ein Dialog wird angezeigt, in dem Sie den Begrenzer (trennendes Zeichen)

angeben können, der in der Datei benutzt wird. Es wird auch angezeigt, wie die Daten auf verschiedene Tabellenzellen verteilt werden.

Mit anderen Optionen in diesem Dialog können Sie das **Format** der Tabellenzellen festlegen, ob Anführungsstriche entfernt werden sollen, und ob die erste Zeile der Datei ignoriert werden sollte.

**Datei → Exportieren ...**

Speichert ein Dokument in eines der unterstützten Dateiformate. Das Dokument bleibt unverändert und wird nicht zur exportierten Datei.

**Datei → Versenden ...**

Versenden Sie die Datei als E-Mail-Anhang.

**Datei → Vorlage aus Dokument erstellen ...**

Erzeugt eine Calligra Sheets-[Vorlage](#) aus diesem Dokument.

**Datei → Drucken ... (Strg+P)**

Druckt das Dokument.

**Datei → Druckvorschau ...**

Zeigt eine Druckvorschau des Dokuments an.

**Datei → Dokument-Informationen**

Lassen Sie sich Informationen zum Dokument und zum Autor anzeigen oder ändern Sie diese.

**Datei → Schließen (Strg+W)**

Schließt das gegenwärtige Dokument, beendet Calligra Sheets aber nicht.

**Datei → Beenden (Strg+Q)**

Beendet Calligra Sheets.

## 7.2 Das Menü Bearbeiten

**Bearbeiten → Rückgängig (Strg+Z)**

Macht die letzte Aktion rückgängig.

**Bearbeiten → Wiederherstellen (Strg+Umschalt+Z)**

Stellt die letzte rückgängig gemachte Aktion wieder her.

**Bearbeiten → Ausschneiden (Strg+X)**

Verschiebt die ausgewählten Zelleninhalte in die Zwischenablage und entfernt sie aus den ursprünglichen Zellen. Wenn Sie danach **Einfügen** wählen, werden die Inhalte in gewählte Zellen wieder eingefügt.

**Bearbeiten → Kopieren (Strg+C)**

Kopiert die ausgewählten Zellen in die Zwischenablage.

**Bearbeiten → Einfügen (Strg+V)**

Fügt Daten (z. B. Zelleninhalte) aus der Zwischenablage in die ausgewählten Zellen ein.

**Bearbeiten → Einfügen spezial ...**

Spezielle Formen der Einfügen-Funktion. In den Abschnitten [Andere Einfügungsarten](#) und [Berechnungen mit Einfügen spezial](#) finden Sie dazu weitere Einzelheiten.

**Bearbeiten → Einfügen und Zellen verschieben**

Verschiebt die Zellinhalte im Einfügebereich nach rechts oder nach unten und fügt Elemente der Zwischenablage in die ausgewählte(n) Zelle(n) ein.

**Bearbeiten → Füllen**

Füllt den ausgewählten Bereich mit den Werten des ersten Element-Satzes. Alle vier Richtungen werden unterstützt. Beachten Sie, dass der Begriff „Element-Satz“ den ersten Satz von Werten meint, den Sie in der Füllrichtung sehen. Wenn die Füllrichtung Links ist, dann ist der erste Element-Satz die letzte Spalte der Auswahl.

**Bearbeiten → Suchen ... (Strg+F)**

Sucht die Zellen, die den angegebenen Text enthalten.

**Bearbeiten → Weitersuchen (F3)**

Sucht die nächste Zelle die den gesuchten Text enthält.

**Bearbeiten → Frühere suchen (Umschalt+F3)**

Sucht die vorhergehenden Zellen, die den gesuchten Text enthalten.

**Bearbeiten → Ersetzen ... (Strg+R)**

Suchen und ersetzen Sie gegebenen Text in Zellen.

**Bearbeiten → Leeren**

Löscht **Alle** oder nur **Inhalt, Kommentar, Bedingte Stile, Gültigkeit, Verknüpfungen** oder **Gültigkeit** in den ausgewählten Zellen.

**Bearbeiten → Löschen**

Löscht **Zellen, Spalten, Zeilen** oder eine **Tabelle**.

**Bearbeiten → Zellen Bearbeiten (F2)**

Bearbeiten Sie die ausgewählte Zelle in der Tabelle.

## 7.3 Das Menü Ansicht

**Ansicht → Neue Ansicht**

Öffnet ein neues Calligra Sheets-Fenster mit dem gleichen Dokument.

**Ansicht → Seitenumrandungen**

Schaltet die Anzeige der Seitenränder der Druckseiten durch rote Linien in der Tabelle ein oder aus.

**Ansicht → Zoom**

Vergrößern oder Verkleinern Sie die Darstellung der Tabelle auf dem Bildschirm im Bereich von 33 % bis 500 %.

## 7.4 Das Menü Gehe zu

In diesem Menü finden Sie Aktionen zur Navigation zwischen den Zellen des geöffneten Dokuments und können mit **Gehe zu** direkt zu einer Zelle gehen oder einen Bereich von Zellen auswählen.

## 7.5 Das Menü Einfügen

### **Einfügen** → **Kommentar**

Fügt Sie einen Kommentar zu den ausgewählten Zelle hinzu, löscht oder ändert ihn.

### **Einfügen** → **Funktion ...**

Fügt eine mathematische Funktion ein. Im Abschnitt [Formeln](#) finden Sie dazu ausführliche Informationen.

### **Einfügen** → **Reihe ...**

Fügt eine Reihe ein. Im Abschnitt [Reihen](#) finden Sie dazu ausführliche Informationen.

### **Einfügen** → **Verknüpfung ...**

Fügt eine Verknüpfung in die ausgewählte Zelle ein. Im Abschnitt [Verknüpfungen in Zellen](#) finden Sie dazu ausführliche Informationen.

### **Einfügen** → **Sonderzeichen ...**

Fügt ein Sonderzeichen in die ausgewählte Zelle ein.

### **Einfügen** → **Externe Daten**

Fügen Sie Daten **Aus Datenbank ...**, **Aus Textdatei ...** oder **Aus Zwischenablage ...** ein. Im Abschnitt [Externe Daten einfügen](#) finden Sie dazu ausführliche Informationen.

## 7.6 Das Menü Format

### **Format** → **Zellenformat ... (Alt+Strg+F)**

Formatiert ausgewählte Zellen. Im Abschnitt [Tabellenformatierung](#) finden Sie dazu ausführliche Informationen.

### **Format** → **Stilverwaltung**

Zellenstile erstellen, ändern oder löschen.

### **Format** → **Stil**

Wendet einen Stil auf ausgewählte Zellen an. Um Stile zu verwalten, verwenden Sie **Format** → **Stilverwaltung**

### **Format** → **Erstelle einen Stil basierend auf dieser Zelle ...**

Erstellt einen neuen Stil aus dem Format der ausgewählten Zelle. Um Stile zu verwalten, verwenden Sie **Format** → **Stilverwaltung**

### **Format** → **Automatische Formatierung ...**

Automatische Formatierung der ausgewählten Zellen. Wählen Sie einen Tabellenstil aus einem Dialog mit Vorschlägen.

### **Format** → **Zellen verbinden**

Verschmilzt die ausgewählten Zellen zu einer großen Zelle.

### **Format** → **Zellen trennen**

Trennt die ausgewählten verschmolzenen Zellen wieder in die entsprechenden ursprünglichen Zellen auf.

### **Format** → **Zeile und Spalte anpassen**

Stellt Sie die Zeilen- und Spaltengrößen so ein, dass die Inhalte der Zelle(n) vollständig angezeigt werden.

**Format → Zeile**

Bestimmen Sie die Höhe der Zeile neu, gleichen Sie sie aus, blenden Sie sie aus oder zeigen Sie sie wieder an.

**Format → Spalte**

Bestimmen Sie die Breite der Spalte neu, gleichen Sie sie aus, blenden Sie sie aus oder zeigen Sie sie wieder an.

**Format → Tabelle**

Eine Tabelle ausblenden, anzeigen oder erweiterte Tabelleneigenschaften einstellen.

**Format → Seitenlayout ...**

Formatieren Sie das Seitenlayout der zu druckenden Seiten.

**Format → Druckbereich**

Legen Sie den zu druckenden Bereich fest oder setzen Sie diesen Bereich wieder zurück.

## 7.7 Das Menü Daten

**Daten → Sortieren ...**

Sortiert die Daten in ausgewählten Zellen. Im Abschnitt [Daten sortieren](#) finden Sie dazu ausführliche Informationen.

**Daten → Text auf Spalten aufteilen ...**

Diese Option versucht Text in den ausgewählten Zellen als CSV-Daten zu interpretieren und platziert jedes Element in einer anderen Zelle der Zeile.

**Daten → Benannte Bereiche ... (Strg+Umschalt+G)**

Öffnet den Dialog **Benannte Bereiche**, um diese hinzuzufügen, zu bearbeiten oder zu löschen. Im Abschnitt [Benannte Zellen und Zellenbereiche](#) finden Sie dazu weitere Informationen.

**Daten → Zusammenfassen ...**

Fasst Daten zusammen. Im Abschnitt [Daten zusammenfassen](#) finden Sie dazu ausführliche Informationen.

**Daten → Zwischensummen ...**

Erstellt verschiedene Arten von Teilsummen in einer Datenbank.

**Daten → Gültigkeit ...**

Stellen Sie die Fehlerprüfungskriterien und Fehlermeldungen für ausgewählte Zellen ein oder ändern Sie diese. Im Abschnitt [Gültigkeitsprüfung](#) finden Sie dazu weitere Einzelheiten.

**Daten → Zielwertsuche ...**

Öffnet den Dialog Zielwertsuche. Im Abschnitt [Zielwertsuche](#) finden Sie dazu ausführliche Informationen.

**Daten → Pivot ...**

Öffnet den Einrichtungdialog Pivot-Tabelle. Im Abschnitt [Pivot-Tabelle](#) finden Sie dazu ausführliche Informationen.

## 7.8 Das Menü Extras

### Extras → Rechtschreibung ...

Überprüft die Rechtschreibung von Wörtern in der Tabelle.

### Extras → Benutzerdefinierte Listen ...

In Calligra Sheets definierte Wortreihen werden angezeigt und können im Dialog geändert werden. Diese Wortlisten können dann als [Serien](#) in das Dokument eingefügt werden.

### Extras → Tabelle schützen ...

Schützt die Tabelle mit einem Passwort. Dieses Passwort geben Sie in einem Dialog ein. Deaktivieren Sie diese Eigenschaft, müssen Sie wieder das Passwort eingeben, um den Schutz der Tabelle aufzuheben. In einer geschützten Tabelle sind alle darin enthaltenen Zellen geschützt. In einer geschützten Tabelle können Sie Zellen weder neu formatieren noch überschreiben.

### Extras → Dokument schützen ...

Schützt das gesamte Dokument mit einem Passwort. Dieses Passwort geben Sie in einem Dialog ein. Deaktivieren Sie diese Eigenschaft, müssen Sie wieder das Passwort eingeben, um den Schutz für das Dokument aufzuheben. Tabellen in einem geschützten Dokument können Sie weder umbenennen noch verschieben. In einem geschützten Dokument sind die einzelnen Tabellen jedoch nicht geschützt.

### Extras → Tabelle neu berechnen (Umschalt+F9)

Berechnet alle Formeln in der aktuellen Tabelle neu.

### Extras → Dokument neu berechnen (F9)

Berechnet alle Tabellen im Dokument neu.

### Extras → Skriptdatei ausführen ...

Führt das ausgewählte externe Skript in Calligra Sheets aus. Calligra Sheets unterstützt Skripte in den Programmiersprachen JavaScript, Python und Ruby. Standardbeispiele von Skripten finden Sie im Untermenü **Extras** → **Skripte**.

### Extras → Skripte

Mit diesen Menüeinträgen können Sie Skripte zum Import oder Export von Daten in verschiedenen Formaten ausführen, Ihre Tabellen speichern, in Protokolldateien schreiben, [Orca -Sprachausgabe](#) für Zugangshilfen, Python- und Ruby-Skripte auf Fehler untersuchen oder Funktionen hinzufügen, um Aktienwerte oder Wetterbedingungen. Sie können außerdem auch einige [R-Funktionen](#) verwenden, wenn [RPy-Module](#) installiert sind.

### Extras → Skriptverwaltung ...

Öffnet den Dialog **Skriptverwaltung**, in dem Sie Skripte ausführen, laden, entladen, installieren, deinstallieren und weitere Skripte herunterladen können.

### Extras → Funktionsoptimierung ...

Öffnet den Dialog **Funktionsoptimierung**, in dem Sie eine Zelle und **Maximieren** oder **Minimieren** als Optimierungsziel auswählen oder den **Wert** und Parameter für eine Gruppe von Zellen auswählen können.

## 7.9 Das Menü Einstellungen

### Einstellungen → Angezeigte Werkzeugleisten

Zeigt die Werkzeugleisten **Datei**, **Bearbeiten**, **Navigation**, **Schriftart**, **Format** und **Farbe/Umrandung** an oder blendet sie aus.

### Einstellungen → Statusleiste

Zeigt die Statusleiste an oder blendet sie aus. Die Statusleiste zeigt zusätzliche Informationen zu ausgewählten Elementen und aktuellen Berechnungen der ausgewählten Zellen an.

### Einstellungen → Tabellenleiste

Zeigt die Tabellenleiste an oder blendet sie aus. Alle Tabellen des aktuellen Dokuments können Sie über die Tabellenleiste erreichen.

### Einstellungen → Kurzbefehle festlegen ...

Legen Sie die Tastaturkurzbefehle fest, die von Calligra Sheets verwendet werden. Im Abschnitt [Kurzbefehle](#) finden Sie dazu ausführliche Informationen.

### Einstellungen → Werkzeugleisten einrichten ...

Richten Sie hier die Werkzeugleisten von Calligra Sheets ein. Im Abschnitt [Werkzeugleisten einrichten](#) finden Sie dazu weitere Informationen.

### Einstellungen → Designs

Auswahl des Farbdesigns für das Calligra Sheets-Fenster. Sie können eines der vordefinierten Farbschemata auswählen oder mit **Einrichtung ...** das [Modul Farben der Systemeinstellungen](#) öffnen.

### Einstellungen → Aktives Autorprofil

Stellt das Autorprofil für das aktuelle Dokument ein. Hier kann eines der im [Einrichtungsdialog von Calligra Sheets](#) definiertes Profil, **Standard-Autorprofil** definiert in den Systemeinstellungen oder das leere Profil **Anonym** zum Schutz der Privatsphäre ausgewählt werden.

### Einstellungen → Benachrichtigungen festlegen ...

Richtet das Benachrichtigungssystem für Calligra Sheets ein. In der aktuellen Version von Calligra Sheets gib es noch keine Aktionen, für die Benachrichtigungen eingestellt werden können.

### Einstellungen → Calligra Sheets einrichten ...

Allgemeine Einstellungen für Calligra Sheets. Im Abschnitt [Einrichtungsdialog von Calligra Sheets](#) finden Sie dazu weitere Einzelheiten.

## 7.10 Das Menü Hilfe

### Hilfe → Handbuch zu Calligra Sheets (F1)

Startet das Hilfe-System von KDE mit der Calligra Sheets-Hilfe (diesem Dokument).

### Hilfe → Was ist das? (Umschalt+F1)

Ändert den Mauscursor zu einer Kombination von Zeiger und Fragezeichen. Das Klicken auf ein Element innerhalb von Calligra Sheets öffnet ein Hilfefenster, das die Funktion des Elementes beschreibt (sofern es Hilfe für das bestimmte Element gibt).

**Hilfe → Probleme oder Wünsche berichten ...**

Öffnet den Dialog für Problembenachrichtungen, in dem Sie Fehler und Wünsche berichten können.

**Help → Sprache der Anwendung umschalten**

Öffnet einen Dialog zur Einstellung der **Hauptsprache** und **Ausweichsprache** für diese Anwendung.

**Hilfe → Über Calligra Sheets**

Zeigt Versions- und Autoreninformationen an.

**Hilfe → Über KDE**

Zeigt Versionsinformation und Grundsätzliches zu KDE an.

## 7.11 Das Kontextmenü der rechten Maustaste

Dieser Abschnitt beschreibt die Einträge in dem Menü, das durch Klicken mit der rechten Maustaste auf ausgewählten Zellen, Zeilen oder Spalten geöffnet wird.

**Zellenformat ... (Strg+Alt+F)**

Formatiert ausgewählte Zellen. Im Abschnitt [Tabellenformatierung](#) finden Sie dazu ausführliche Informationen.

**Ausschneiden (Strg+X)**

Verschiebt die ausgewählten Zellen in die Zwischenablage. Wenn Sie danach **Einfügen** wählen, werden die Zellen an der neuen Cursor-Position wieder eingefügt.

**Kopieren (Strg+C)**

Kopiert die ausgewählten Zellen in die Zwischenablage.

**Einfügen (Strg+V)**

Fügt Daten aus der Zwischenablage in die ausgewählten Zellen ein.

**Einfügen spezial ...**

Spezielle Formen der Einfügen-Funktion. In den Abschnitten [Andere Einfügungsarten](#) und [Berechnungen mit Einfügen spezial](#) finden Sie dazu weitere Einzelheiten.

**Einfügen und Zellen verschieben**

Fügt Daten aus der Zwischenablage in die ausgewählten Zellen ein, wobei die Zellen im ausgewählten Bereich vorher nach rechts oder nach unten verschoben werden, um Platz für die neuen Daten zu machen.

**Alle**

Löscht den Inhalt der ausgewählten Zellen.

**Zeile und Spalte anpassen**

Stellt die Zeilen- und Spaltengrößen so ein, dass die Zelleninhalte alle komplett angezeigt werden.

**Standard**

Stellt das Standardformat für die ausgewählten Zellen ein.

**Bereichsname ...**

Benennen Sie den ausgewählten Bereich. Im Abschnitt [Benannte Zellenbereiche](#) finden Sie dazu ausführliche Informationen.

**Zeilenhöhe ändern ...**

Ändert die Höhe der ausgewählten Zeile.

**Zeile anpassen**

Ändert die Höhe der ausgewählten Zeile, um Zelleninhalte vollständig anzuzeigen.

**Spaltenbreite ändern ...**

Ändert die Breite der ausgewählten Spalte.

**Spalte anpassen**

Ändert die Breite der ausgewählten Spalte, um Zelleninhalte vollständig anzuzeigen.

**Zellen einfügen ...**

Fügt neue Zellen an der ausgewählten Position ein, wobei die vorhandenen Zellen verschoben werden, um Platz für die neuen Zellen zu schaffen.

**Zellen löschen ...**

Löscht die ausgewählten Zellen und verschiebt andere Zellen in den ausgewählten Bereich, um den Raum, der durch die entfernten Zellen belegt war, wieder zu füllen.

**Zellen einfügen**

Fügen Sie neue Zeilen über den aktuell ausgewählten Zeilen ein.

**Zeilen löschen**

Löscht die ausgewählten Zeilen.

**Zeilen ausblenden**

Blendet die gewählten Zeilen aus.

**Zeilen anzeigen**

Zeigt die ausgewählte(n) Zeile(n) an. Um eine ausgeblendete Zeile anzuzeigen, müssen Sie mehrere Zeilen auswählen, die die ausgeblendete Zeile umfassen.

**Spalten einfügen**

Fügen Sie neue Spalten links von den ausgewählten Spalten ein.

**Spalten löschen**

Löscht die ausgewählten Spalten.

**Spalten ausblenden**

Blendet die gewählten Spalten aus.

**Spalten anzeigen**

Zeigt die ausgewählte(n) Spalte(n) an. Um eine ausgeblendete Spalte anzuzeigen, müssen Sie mehrere Spalten auswählen, die die ausgeblendete Spalte umfassen.

**Kommentar ...**

Fügen Sie einen Kommentar zur ausgewählten Zelle hinzu oder ändern Sie ihn.

**Auswahlliste ...**

Ermöglicht das Auswählen und Einfügen von Text aus einer anderen Zelle der aktuellen Spalte in die aktuell ausgewählte Zelle.

## 7.12 Andere Tasten-Kurzbefehle

Dieser Abschnitt beschreibt die Calligra Sheets-Tastatur-Kurzbefehle, die für Aktionen verwendet werden, die in keinem Menü vorkommen.

### **Strg+H**

Blendet die andockbaren Dialoge ein und aus.

### **Strg+Pfeiltasten**

Wenn die ausgewählte Zelle belegt ist, wird der Cursor auf den Anfang oder das Ende des belegten Blockes in der aktuellen Zeile oder Spalte bewegt. Wenn die ausgewählte Zelle nicht belegt ist, wird der Cursor auf den Anfang oder das Ende des nicht belegten Blockes in der aktuellen Zeile oder Spalte bewegt.

### **Strg+Umschalt+Pfeiltasten**

Wenn die ausgewählte Zelle belegt ist, werden alle Zellen bis zum Anfang oder Ende des belegten Blockes in der aktuellen Zeile oder Spalte ausgewählt. Wenn die ausgewählte Zelle nicht belegt ist, werden alle Zellen bis zum Anfang oder Ende des nicht belegten Blockes in der aktuellen Zeile oder Spalte ausgewählt.

### **Bild ab**

Bewegt den Cursor um 10 Zellen nach unten.

### **Bild auf**

Bewegt den Cursor um 10 Zellen nach oben.

### **Strg+Bild ab**

Zeigt die nächste Tabelle an.

### **Strg+Bild auf**

Zeigt die vorhergehende Tabelle an.

### **F4**

Ändert einen Zellenverweis zwischen normalen und [absoluten Verweisarten](#).

## Kapitel 8

# Funktionen

Calligra Sheets enthält eine große Zahl von eingebauten mathematischen und anderen Funktionen, die als Formel in einer Zelle verwendet werden können.

### 8.1 Unterstützte Funktionen

In diesem Kapitel finden Sie eine kurze Übersicht aller Funktionen in folgenden Gruppen:

- Bit-Operationen
- Konvertierung
- Datenbank
- Datum & Zeit
- Ingenieurwesen
- Finanzen
- Information
- Logik
- Nachschlagen & Referenz
- Mathematik
- Statistik
- Text
- Trigonometrie

#### 8.1.1 Bit-Operationen

##### 8.1.1.1 BITAND

Die Funktion BITAND() führt eine bitweise AND-Operation für die beiden ganzzahligen Parameter durch.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

BITAND(Wert;Wert)

##### Parameter

*Kommentar:* Erste Zahl, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Zweite Zahl, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

BITAND(12;10) ergibt 8 (weil dezimal 12 binär 1100 und dezimal 10 binär 1010 entspricht, die AND-Operation von 1100 und 1010 dann 1000 ergibt und dies wiederum der Ganzzahl 8 entspricht).

### Ähnliche Funktionen

[BITOR](#)  
[BITXOR](#)

#### 8.1.1.2 BITLSHIFT

Die Funktion BITLSHIFT() führt eine bitweise Linksverschiebung des ersten Parameters durch. Die Anzahl an Bits, um die verschoben werden soll, wird durch den zweiten Parameter angegeben. Beachten Sie, dass ein negativer Wert für die Linksverschiebung zu einer Rechtsverschiebung führt.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

### Syntax

BITLSHIFT(Wert; Verschiebung)

### Parameter

*Kommentar:* Erste Zahl, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Anzahl, um die nach links verschoben werden soll, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Ähnliche Funktionen

[BITLSHIFT](#)

#### 8.1.1.3 BITOR

Die Funktion BITOR() führt eine bitweise OR-Operation für die beiden ganzzahligen Parameter durch.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

### Syntax

BITOR(Wert;Wert)

### Parameter

*Kommentar:* Erste Zahl, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Zweite Zahl, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

BITOR(12;10) ergibt 14 (weil dezimal 12 binär 1100 und dezimal 10 binär 1010 entspricht, die OR-Operation von 1100 und 1010 dann 1110 ergibt und dies wiederum der Ganzzahl 14 entspricht).

### Ähnliche Funktionen

[BITAND](#)  
[BITXOR](#)

#### 8.1.1.4 BITRSHIFT

Die Funktion BITRSHIFT() führt eine bitweise Rechtsverschiebung des ersten Parameters durch. Die Anzahl an Bits, um die verschoben werden soll, wird durch den zweiten Parameter angegeben. Beachten Sie, dass ein negativer Wert für die Linksverschiebung zu einer Linksverschiebung führt.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

BITRSHIFT(Wert; Verschiebung)

##### Parameter

*Kommentar:* Erste Zahl, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Anzahl, um die nach rechts verschoben werden soll, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Ähnliche Funktionen

[BITLSHIFT](#)

#### 8.1.1.5 BITXOR

Die Funktion BITXOR() führt eine bitweise XOR-Operation für die beiden ganzzahligen Parameter durch.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

BITXOR(Wert; Wert)

##### Parameter

*Kommentar:* Erste Zahl, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Zweite Zahl, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

BITXOR(12;10) ergibt 6 (weil dezimal 12 binär 1100 und dezimal 10 binär 1010 entspricht, die XOR-Operation von 1100 und 1010 dann 0110 ergibt und dies wiederum der Ganzzahl 6 entspricht).

##### Ähnliche Funktionen

[BITAND](#)

[BITOR](#)

### 8.1.2 Konvertierung

#### 8.1.2.1 ARABIC

Die Funktion ARABIC() konvertiert eine römische Zahl in eine normale (arabische) Zahl.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

ARABIC(Zahl)

**Parameter**

*Kommentar:* Zahlzeichen, *Typ:* Text

**Beispiele**

ARABIC("IV") ergibt 4

**Beispiele**

ARABIC("XCIX") ergibt 99

**Ähnliche Funktionen**

[ROMAN](#)

**8.1.2.2 ASCIITOCHAR**

Die Funktion ASCIITOCHAR() gibt für jeden gegebenen ASCII-Code das entsprechende Zeichen zurück.

*Rückgabotyp:* Text

**Syntax**

ASCIITOCHAR(Wert)

**Parameter**

*Kommentar:* Die zu konvertierenden ASCII-Werte, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

**Beispiele**

ASCIITOCHAR(118) ergibt „v“

**Beispiele**

ASCIITOCHAR(75; 68; 69) ergibt „KDE“

**8.1.2.3 BOOL2INT**

Die Funktion BOOL2INT ergibt einen ganzzahligen Wert für einen gegebenen booleschen Wert. Mit dieser Funktion kann man einen booleschen Wert da verwenden, wo ein ganzzahliger Wert erwartet wird.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

**Syntax**

BOOL2INT(Wert)

**Parameter**

*Kommentar:* Zu konvertierender boolescher Wert, *Typ:* Wahrheitswert (WAHR oder FALSCH)

**Beispiele**

BOOL2INT(Wahr) ergibt 1

**Beispiele**

BOOL2INT(Falsch) ergibt 0

**Ähnliche Funktionen**

[INT2BOOL](#)

#### 8.1.2.4 BOOL2STRING

Die Funktion BOOL2STRING ergibt eine Zeichenfolge, die dem booleschen Wert entspricht. Diese Funktion dient dazu, dass man einen booleschen Wert in Funktionen verwenden kann, in denen eine Zeichenfolge erwartet wird.

*Rückgabebetyp:* Text

##### Syntax

BOOL2STRING(Wert)

##### Parameter

*Kommentar:* Zu konvertierender boolescher Wert, *Typ:* Wahrheitswert (WAHR oder FALSCH)

##### Beispiele

BOOL2STRING(Wahr) ergibt „Wahr“

##### Beispiele

BOOL2STRING(Falsch) ergibt „Falsch“

##### Beispiele

upper(BOOL2STRING(find("nan";"Banane"))) ergibt Wahr

#### 8.1.2.5 CARX

Die Funktion CARX() gibt die X Position zu der Position eines Punktes in einem polaren Koordinatensystem zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

CARX(Radius;Winkel)

##### Parameter

*Kommentar:* Radius, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

CARX(12;1,5707) ergibt 0,00115592

##### Beispiele

CARX(12;0) ergibt 12

##### Ähnliche Funktionen

CARY

POLA

POLR

### 8.1.2.6 CARY

Die Funktion CARY() gibt die Y Position zu der Position eines Punktes in einem polaren Koordinatensystem zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

CARY(Radius;Winkel)

#### Parameter

*Kommentar:* Radius, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

CARY(12;1,5707) ergibt 12

#### Beispiele

CARY(12;0) ergibt 0

#### Ähnliche Funktionen

[CARX](#)

[POLA](#)

[POLR](#)

### 8.1.2.7 CHARTOASCII

Die Funktion CHARTOASCII() gibt den ASCII-Code des gegebenen Zeichens zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

CHARTOASCII(Wert)

#### Parameter

*Kommentar:* Eine zu konvertierende Zeichenfolge bestehend aus einem Buchstaben, *Typ:* Text

#### Beispiele

CHARTOASCII("v") ergibt 118

#### Beispiele

CHARTOASCII(r) ist ein Fehler. Das Zeichen muss von Anführungszeichen umgeben sein.

### 8.1.2.8 DECSEX

Die Funktion DECSEX() konvertiert einen Double-Wert in einen Zeitwert.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

DECSEX(Double)

#### Parameter

*Kommentar:* Wert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

DECSEX(1,6668) ergibt 1:40

### Beispiele

DECSEX(7,8) ergibt 7:47

#### 8.1.2.9 INT2BOOL

Die Funktion INT2BOOL ergibt einen booleschen Wert für einen gegebenen ganzzahligen Wert. Mit dieser Funktion kann man einen ganzzahligen Wert da verwenden, wo ein boolescher Wert erwartet wird. Bei einer 1 wird Wahr zurückgegeben, bei jedem anderen Wert wird Falsch zurückgegeben.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

#### Syntax

INT2BOOL(Wert)

#### Parameter

*Kommentar:* Zu konvertierender ganzzahliger Wert, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

INT2BOOL(1) ergibt Wahr

#### Beispiele

INT2BOOL(0) ergibt Falsch

#### Beispiele

OR(INT2BOOL(1); Falsch) ergibt Wahr

#### Ähnliche Funktionen

[BOOL2INT](#)

#### 8.1.2.10 NUM2STRING

Die Funktion NUM2STRING() ergibt eine Zeichenfolge für eine gegebene Zahl. Beachten Sie, dass Calligra Sheets falls nötig automatisch Zahlen in Zeichenfolgen konvertieren kann. Von daher sollte diese Funktion kaum benötigt werden.

*Rückgabotyp:* Text

#### Syntax

NUM2STRING(Wert)

#### Parameter

*Kommentar:* In Zeichenfolge zu konvertierende Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

NUM2STRING(10) ergibt „10“

#### Beispiele

NUM2STRING(2,05) ergibt „2,05“

#### Beispiele

=find(„101“;NUM2STRING(A1)) (A1 = 2,010102) ergibt Wahr

#### Ähnliche Funktionen

[STRING](#)

### 8.1.2.11 POLA

Die Funktion POLA() gibt den Winkel (als Radiant) zu der Position eines Punktes in einem kartesischen Koordinatensystem zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

POLA(X;Y)

#### Parameter

*Kommentar:* Wert in X, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Wert in Y, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

POLA(12;12) ergibt 0,78539816

#### Beispiele

POLA(12;0) ergibt 0

#### Beispiele

POLA(0;12) ergibt 1,5707

#### Ähnliche Funktionen

[POLR](#)

[CARX](#)

[CARY](#)

### 8.1.2.12 POLR

Die Funktion POLR() gibt den Radius passend zu der Position eines Punktes in einem kartesischen Koordinatensystem zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

POLR(X;Y)

#### Parameter

*Kommentar:* Wert in X, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Wert in Y, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

POLR(12;12) ergibt 16,9705

#### Beispiele

POLR(12;0) ergibt 12

#### Ähnliche Funktionen

[POLA](#)

[CARX](#)

[CARY](#)

### 8.1.2.13 ROMAN

Die Funktion ROMAN() gibt eine Zahl im Römischen Format zurück. Nur positiv ganze Zahlen können umgewandelt werden. Das optionale Format-Argument legt den Vereinfachungsgrad fest. Standard ist 0.

*Rückgabotyp:* Text

#### Syntax

ROMAN(Zahl)

#### Parameter

*Kommentar:* Zahl, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Format, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

ROMAN(99) ergibt „XCIX“

#### Beispiele

ROMAN(-55) ergibt „Err“

#### Ähnliche Funktionen

[ARABIC](#)

### 8.1.2.14 SEXDEC

Die Funktion SEXDEC() gibt den Wert als Dezimalzahl zurück. Sie können auch ein Zeitwert angeben.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

SEXDEC(Zeitwert) oder SEXDEC(Stunden;Minuten;Sekunden)

#### Parameter

*Kommentar:* Stunden, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Minuten, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Sekunden, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

SEXDEC(1;5;7) ergibt 1,0852778

#### Beispiele

DECSEX("8:05") ergibt 8,08333333

### 8.1.2.15 STRING

Die Funktion STRING() ergibt eine Zeichenfolge für eine gegebene Zahl. Sie ist mit der NUM2STRING-Funktion identisch.

*Rückgabotyp:* Text

#### Syntax

#### Parameter

*Kommentar:* In Zeichenfolge zu konvertierende Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Ähnliche Funktionen

[NUM2STRING](#)

## 8.1.3 Datenbank

### 8.1.3.1 DAVERAGE

Berechnet den Durchschnitt in einer Spalte einer Datenbank, eingeschränkt durch ein Satz von Regeln für Werte die Zahlen sind.

*Rückgabetyyp:* FLOAT

#### Syntax

DAVERAGE(Datenbank; "Spaltenüberschrift"; Bedingungen)

#### Parameter

*Kommentar:* Ein Zellbereich, welcher die Datenbank darstellt., *Typ:* Eine Zeichenfolge

*Kommentar:* Eine Zeichenfolge, welche die Spalte in der Datenbank auswählt., *Typ:* Text

*Kommentar:* Ein Bereich, der die Bedingungen enthält., *Typ:* Eine Zeichenfolge

#### Beispiele

DAVERAGE(A1:C5; "Gehalt"; A9:A11)

### 8.1.3.2 DCOUNT

Zählt die Zellen, die numerische Werte in der Spalte einer Datenbank enthalten und die einen angegebenen Satz von Bedingungen erfüllen.

*Rückgabetyyp:* FLOAT

#### Syntax

DCOUNT(Datenbank; "Spaltenüberschrift"; Bedingungen)

#### Parameter

*Kommentar:* Ein Zellbereich, welcher die Datenbank darstellt., *Typ:* Eine Zeichenfolge

*Kommentar:* Eine Zeichenfolge, welche die Spalte in der Datenbank auswählt., *Typ:* Text

*Kommentar:* Ein Bereich, der die Bedingungen enthält., *Typ:* Eine Zeichenfolge

#### Beispiele

DCOUNT(A1:C5; "Gehalt"; A9:A11)

#### Ähnliche Funktionen

[DCOUNTA](#)

### 8.1.3.3 DCOUNTA

Zählt die Zellen, die numerische oder alphanumerische Werte in der Spalte einer Datenbank enthalten. Die zu zählenden Zellen werden durch einen Satz von Bedingungen eingeschränkt.

*Rückgabetyyp:* FLOAT

#### Syntax

DCOUNTA(Datenbank; "Spaltenüberschrift"; Bedingungen)

#### Parameter

*Kommentar:* Ein Zellbereich, welcher die Datenbank darstellt., *Typ:* Eine Zeichenfolge

*Kommentar:* Eine Zeichenfolge, welche die Spalte in der Datenbank auswählt., *Typ:* Text

*Kommentar:* Ein Bereich, der die Bedingungen enthält., *Typ:* Eine Zeichenfolge

### Beispiele

DCOUNTA(A1:C5; "Gehalt"; A9:A11)

### Ähnliche Funktionen

[DCOUNT](#)

#### 8.1.3.4 DGET

Gibt den Wert einer Datenbankspalte zurück, der einen Satz von Bedingungen erfüllt. Die Funktion gibt einen Fehler zurück, wenn kein Wert oder mehr als ein Wert die Bedingungen erfüllt.

*Rückgabety*p: FLOAT

#### Syntax

DGET(Datenbank; "Spaltenüberschrift"; Bedingungen)

#### Parameter

*Kommentar:* Ein Zellbereich, welcher die Datenbank darstellt., *Typ:* Eine Zeichenfolge

*Kommentar:* Eine Zeichenfolge, welche die Spalte in der Datenbank auswählt., *Typ:* Text

*Kommentar:* Ein Bereich, der die Bedingungen enthält., *Typ:* Eine Zeichenfolge

### Beispiele

DGET(A1:C5; "Gehalt"; A9:A11)

#### 8.1.3.5 DMAX

Gibt die größte Zahl der Spalte einer Datenbank zurück, eingeschränkt durch einen Satz von Bedingungen.

*Rückgabety*p: FLOAT

#### Syntax

DMAX(Datenbank; "Spaltenüberschrift"; Bedingungen)

#### Parameter

*Kommentar:* Ein Zellbereich, welcher die Datenbank darstellt., *Typ:* Eine Zeichenfolge

*Kommentar:* Eine Zeichenfolge, welche die Spalte in der Datenbank auswählt., *Typ:* Text

*Kommentar:* Ein Bereich, der die Bedingungen enthält., *Typ:* Eine Zeichenfolge

### Beispiele

DMAX(A1:C5; "Gehalt"; A9:A11)

### Ähnliche Funktionen

[DMIN](#)

#### 8.1.3.6 DMIN

Gibt den kleinsten Wert der Spalte einer Datenbank zurück, eingeschränkt durch einen Satz von Bedingungen.

*Rückgabety*p: FLOAT

#### Syntax

DMIN(Datenbank; "Spaltenüberschrift"; Bedingungen)

**Parameter**

*Kommentar:* Ein Zellbereich, welcher die Datenbank darstellt., *Typ:* Eine Zeichenfolge  
*Kommentar:* Eine Zeichenfolge, welche die Spalte in der Datenbank auswählt., *Typ:* Text  
*Kommentar:* Ein Bereich, der die Bedingungen enthält., *Typ:* Eine Zeichenfolge

**Beispiele**

DMIN(A1:C5; "Gehalt"; A9:A11)

**Ähnliche Funktionen**

[DMAX](#)

**8.1.3.7 DPRODUCT**

Gibt das Produkt aller numerischen Werte der Spalte einer Datenbank zurück, welche einen Satz von Bedingungen erfüllen.

*Rückgabetyyp:* FLOAT

**Syntax**

DPRODUCT(Datenbank; "Spaltenüberschrift"; Bedingungen)

**Parameter**

*Kommentar:* Ein Zellbereich, welcher die Datenbank darstellt., *Typ:* Eine Zeichenfolge  
*Kommentar:* Eine Zeichenfolge, welche die Spalte in der Datenbank auswählt., *Typ:* Text  
*Kommentar:* Ein Bereich, der die Bedingungen enthält., *Typ:* Eine Zeichenfolge

**Beispiele**

DPRODUCT(A1:C5; "Gehalt"; A9:A11)

**8.1.3.8 DSTDEV**

Gibt die geschätzte Standardabweichung einer Grundgesamtheit zurück. Basierend auf einer Stichprobe aus allen numerischen Werten der Spalte einer Datenbank, welche einen Satz von Bedingungen erfüllt.

*Rückgabetyyp:* FLOAT

**Syntax**

DSTDEV(Datenbank; "Spaltenüberschrift"; Bedingungen)

**Parameter**

*Kommentar:* Ein Zellbereich, welcher die Datenbank darstellt., *Typ:* Eine Zeichenfolge  
*Kommentar:* Eine Zeichenfolge, welche die Spalte in der Datenbank auswählt., *Typ:* Text  
*Kommentar:* Ein Bereich, der die Bedingungen enthält., *Typ:* Eine Zeichenfolge

**Beispiele**

DSTDEV(A1:C5; "Gehalt"; A9:A11)

**Ähnliche Funktionen**

[DSTDEVP](#)

### 8.1.3.9 DSTDEVP

Gibt die Standardabweichung einer Grundgesamtheit zurück. Basierend auf allen numerischen Werten einer Datenbankspalte, die einen Satz von Bedingungen erfüllen.

*Rückgabetyp:* FLOAT

#### Syntax

DSTDEVP(Datenbank; "Spaltenüberschrift"; Bedingungen)

#### Parameter

*Kommentar:* Ein Zellbereich, welcher die Datenbank darstellt., *Typ:* Eine Zeichenfolge

*Kommentar:* Eine Zeichenfolge, welche die Spalte in der Datenbank auswählt., *Typ:* Text

*Kommentar:* Ein Bereich, der die Bedingungen enthält., *Typ:* Eine Zeichenfolge

#### Beispiele

DSTDEVP(A1:C5; "Gehalt"; A9:A11)

#### Ähnliche Funktionen

[DSTDEV](#)

### 8.1.3.10 DSUM

Summiert alle Zahlen einer Spalte einer Datenbank, eingeschränkt durch einen Satz von Bedingungen.

*Rückgabetyp:* FLOAT

#### Syntax

DSUM(Datenbank; "Spaltenüberschrift"; Bedingungen)

#### Parameter

*Kommentar:* Ein Zellbereich, welcher die Datenbank darstellt., *Typ:* Eine Zeichenfolge

*Kommentar:* Eine Zeichenfolge, welche die Spalte in der Datenbank auswählt., *Typ:* Text

*Kommentar:* Ein Bereich, der die Bedingungen enthält., *Typ:* Eine Zeichenfolge

#### Beispiele

DSUM(A1:C5; "Gehalt"; A9:A11)

### 8.1.3.11 DVAR

Gibt die geschätzte Streuung einer Grundgesamtheit zurück. Basierend auf einer Stichprobe bestehend aus allen numerischen Werten der Spalte einer Datenbank, eingeschränkt durch einen Satz von Bedingungen.

*Rückgabetyp:* FLOAT

#### Syntax

DVAR(Datenbank; "Spaltenüberschrift"; Bedingungen)

#### Parameter

*Kommentar:* Ein Zellbereich, welcher die Datenbank darstellt., *Typ:* Eine Zeichenfolge

*Kommentar:* Eine Zeichenfolge, welche die Spalte in der Datenbank auswählt., *Typ:* Text

*Kommentar:* Ein Bereich, der die Bedingungen enthält., *Typ:* Eine Zeichenfolge

#### Beispiele

DVAR(A1:C5; "Gehalt"; A9:A11)

#### Ähnliche Funktionen

[DVARP](#)

### 8.1.3.12 DVARP

Gibt die Streuung einer Grundgesamtheit zurück. Basieren auf allen numerischen Werten der Spalte einer Datenbank, eingeschränkt durch einen Satz von Bedingungen.

*Rückgabetyp:* FLOAT

#### Syntax

DVARP(Datenbank; "Spaltenüberschrift"; Bedingungen)

#### Parameter

*Kommentar:* Ein Zellbereich, welcher die Datenbank darstellt., *Typ:* Eine Zeichenfolge

*Kommentar:* Eine Zeichenfolge, welche die Spalte in der Datenbank auswählt., *Typ:* Text

*Kommentar:* Ein Bereich, der die Bedingungen enthält., *Typ:* Eine Zeichenfolge

#### Beispiele

DVARP(A1:C5; "Gehalt"; A9:A11)

#### Ähnliche Funktionen

[DVAR](#)

### 8.1.3.13 GETPIVOTDATA

Holt die zusammengefassten Daten von einer Kreuztabelle.

*Rückgabetyp:* FLOAT

#### Syntax

GETPIVOTDATA(Datenbank; "Verkäufe")

#### Parameter

*Kommentar:* Bereich der die Kreuztabelle enthält, *Typ:* Eine Zeichenfolge

*Kommentar:* Der Name des Feldes von welchem Sie die zusammengefassten Daten möchten, *Typ:* Text

## 8.1.4 Datum & Zeit

### 8.1.4.1 CURRENTDATE

Die Funktion CURRENTDATE() gibt das aktuelle Datum formatiert mit lokalen Parametern zurück. Sie ist äquivalent mit der Funktion TODAY.

*Rückgabetyp:* Datum

#### Syntax

CURRENTDATE()

#### Parameter

#### Beispiele

CURRENTDATE() ergibt (bzw. ergab) „Samstag, 15. Juli 2002“

#### Ähnliche Funktionen

[CURRENTTIME](#)

[TODAY](#)

#### 8.1.4.2 CURRENTDATETIME

Die Funktion CURRENTDATETIME() gibt das aktuelle Datum und die Zeit formatiert mit lokalen Parametern zurück.

*Rückgabetyt:* Datum

##### Syntax

CURRENTDATETIME()

##### Parameter

##### Beispiele

CURRENTDATETIME() ergibt „Samstag, 15. Juli 2002 19:12:01“

#### 8.1.4.3 CURRENTTIME

Die Funktion CURRENTTIME() gibt die aktuelle Zeit formatiert mit lokalen Parametern zurück.

*Rückgabetyt:* Datum

##### Syntax

CURRENTTIME()

##### Parameter

##### Beispiele

CURRENTTIME() ergibt „19:12:01“

#### 8.1.4.4 DATE

Die Funktion DATE() gibt das Datum entsprechend den Landeseinstellungen formatiert zurück.

*Rückgabetyt:* Text

##### Syntax

DATE(Jahr;Monat;Tag)

##### Parameter

*Kommentar:* Jahr, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Monat, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Tag, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

DATE(2000;5;5) ergibt Freitag, 5. Mai 2000

#### 8.1.4.5 DATE2UNIX

Die Funktion DATE2UNIX() konvertiert einen Datums- und Zeitwert in die Unix-Zeit.

Die Unix-Zeit ist die Anzahl an Sekunden nach dem 1. Januar 1970, 00:00 Uhr UTC.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

DATE2UNIX(Datum)

##### Parameter

*Kommentar:* Datum, *Typ:* Text

##### Beispiele

DATE2UNIX(„01.01.2000“) ergibt 946,684,800

#### 8.1.4.6 DATEDIF

Die Funktion DATEDIF() ergibt die Differenz zwischen zwei Datumsangaben.

Das Intervall muss eines der folgenden sein: „m“: Monate; „d“: Tage; „y“: komplette Jahre; „ym“: Monate ohne Jahre; „yd“: Tage ohne Jahre; „md“: Tage ohne Monate und Jahre

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

DATEDIF(Erstes Datum; Zweites Datum; Intervall)

##### Parameter

*Kommentar:* Erstes Datum, *Typ:* Text

*Kommentar:* Zweites Datum, *Typ:* Text

*Kommentar:* Intervall, *Typ:* Text

##### Beispiele

DATEDIF(A1;A2;„d“) mit A1 „1. Januar 1995“ und A2 „15. Juni 1999“ ergibt als Anzahl an Tagen 1626.

##### Beispiele

DATEDIF(A1;A2;„d“) mit A1 „1. Januar 1995“ und A2 „15. Juni 1999“ ergibt eine Anzahl von 53 Wochen

#### 8.1.4.7 DATEVALUE

Die Funktion DATEVALUE gibt eine Zahl zurück, welche den Tag repräsentiert. Das heißt, die Anzahl der Tage seit dem 31. Dezember 1899.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

DATEVALUE(Datum)

##### Parameter

*Kommentar:* Datum, *Typ:* Text

##### Beispiele

DATEVALUE(„2/22/2002“) ergibt 37309

#### Ähnliche Funktionen

[TIMEVALUE](#)

#### 8.1.4.8 DAY

Die Funktion DAY gibt den Tag eines Datum zurück. Wird kein Parameter angegeben, gibt die Funktion den aktuellen Tag zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

DAY(Datum)

##### Parameter

*Kommentar:* Datum, *Typ:* Text

##### Beispiele

DAY("2/22/2002") ergibt 22

##### Beispiele

DAY(2323,1285) ergibt 11

##### Ähnliche Funktionen

[MONTH](#)

[YEAR](#)

#### 8.1.4.9 DAYNAME

Die Funktion DAYNAME() gibt den Namen eines Wochentags (1..7) zurück. In einigen Ländern ist der erste Tag der Woche Montag, in anderen ist der erste Tag der Woche Sonntag.

*Rückgabotyp:* Text

##### Syntax

DAYNAME(Wochentag)

##### Parameter

*Kommentar:* Nummer des Wochentags (1..7), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

DAYNAME(1) ergibt Montag (wenn die Woche nach den Landeseinstellungen mit Montag beginnt)

##### Ähnliche Funktionen

[WEEKDAY](#)

#### 8.1.4.10 DAYOFYEAR

Die Funktion DAYOFYEAR() gibt die Nummer des Tages im Jahr zurück (1-365).

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

DAYOFYEAR(Jahr;Monat;Tag)

#### Parameter

*Kommentar:* Jahr, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Monat, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Tag, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

DAYOFYEAR(2000;12;1) ergibt 336

#### Beispiele

DAYOFYEAR(2000;2;29) ergibt 60

#### 8.1.4.11 DAYS

Die Funktion DAYS() ergibt die Differenz zwischen zwei Daten in Tagen.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

DAYS(datum1; datum2)

#### Parameter

*Kommentar:* Erster (früherer) Datumswert, *Typ:* Text

*Kommentar:* Zweiter Datumswert, *Typ:* Text

#### Beispiele

DAYS("22.02.2002"; "26.02.2002") ergibt 4

#### 8.1.4.12 DAYS360

Die Funktion DAYS360() gibt die Anzahl der Tage von Datum1 bis Datum2 zurück. Dabei wird ein 360-Tage-Kalender verwendet, in dem alle Monate mit 30 Tagen gerechnet werden. Wenn die Methode „Falsch“ ist (Standard) dann wird die amerikanische Berechnungsmethode verwendet, ist Methode „Wahr“ wird die europäische Berechnungsmethode verwendet.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

DAYS360(Datum1; Datum2; Methode)

#### Parameter

*Kommentar:* Datum1, *Typ:* Text

*Kommentar:* Datum2, *Typ:* Text

*Kommentar:* Methode, *Typ:* Wahrheitswert (WAHR oder FALSCH)

#### Beispiele

DAYS360("2/22/2002"; "4/21/2002"; Falsch) ergibt 59

#### Ähnliche Funktionen

[DAYS](#)

[MONTHS](#)

[WEEKS](#)

[YEARS](#)

#### 8.1.4.13 DAYSINMONTH

Die Funktion DAYSINMONTH() gibt die Anzahl der Tage in dem angegebenen Monat eines Jahres zurück.

*Rückgabetyt:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

DAYSINMONTH(Jahr;Monat)

##### Parameter

*Kommentar:* Jahr, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Monat, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

DAYSINMONTH(2000;2) ergibt 29

#### 8.1.4.14 DAYSINYEAR

Die Funktion DAYSINYEAR() gibt die Anzahl der Tage im angegebenen Jahr zurück.

*Rückgabetyt:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

DAYSINYEAR(jahr)

##### Parameter

*Kommentar:* Jahr, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

DAYSINYEAR(2000) ergibt 366

#### 8.1.4.15 EASTERSUNDAY

Die Funktion EASTERSUNDAY() gibt das Datum zurück, welches Ostersonntag in dem angegebenen Jahr entspricht.

*Rückgabetyt:* Datum

##### Syntax

EASTERSUNDAY(Jahr)

##### Parameter

*Kommentar:* Jahr, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

EASTERSUNDAY(2003) ergibt „20. April 2003“

#### 8.1.4.16 EDATE

Die Funktion EDATE gibt das Datum zurück, welches entsteht wenn man von einem gegebenen Datum eine bestimmte Anzahl Monate addiert oder subtrahiert.

*Rückgabebetyp:* Datum

##### Syntax

EDATE(Datum; Monate)

##### Parameter

*Kommentar:* Datum, *Typ:* Text

*Kommentar:* Monate, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

EDATE("2/22/2002"; 3) ergibt "5/22/2002"

##### Beispiele

EDATE("3/31/2002"; -1) ergibt "2/28/2002"

##### Ähnliche Funktionen

[DATE](#)

[EOMONTH](#)

#### 8.1.4.17 EOMONTH

Die Funktion EOMONTH gibt den letzten Tag in einem Monat zurück. Dieser wird festgelegt durch ein Datum und einer Anzahl von Monaten welche zu diesem Datum addiert oder subtrahiert werden.

*Rückgabebetyp:* Datum

##### Syntax

EOMONTH(Datum; Monate)

##### Parameter

*Kommentar:* Datum, *Typ:* Text

*Kommentar:* Monate, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

EOMONTH("2/22/2002"; 3) ergibt "5/31/2002"

##### Beispiele

EOMONTH("3/12/2002"; -1) ergibt "2/28/2002"

##### Beispiele

EOMONTH("3/12/2002"; 0) ergibt "3/31/2002"

##### Ähnliche Funktionen

[EDATE](#)

[MONTH](#)

#### 8.1.4.18 HOUR

Die Funktion HOUR gibt die Stunde der angegebenen Zeit zurück. Wird kein Parameter angegeben, liefert die Funktion die aktuelle Stunde zurück.

*Rückgabetyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

HOUR(Zeit)

##### Parameter

*Kommentar:* Zeit, *Typ:* Text

##### Beispiele

HOUR("22:10:12") ergibt 22

##### Beispiele

HOUR(0,1285) ergibt 3

##### Ähnliche Funktionen

[MINUTE](#)

[SECOND](#)

#### 8.1.4.19 HOURS

Die Funktion HOURS() gibt die Anzahl der Stunden eines Zeitausdrucks zurück.

*Rückgabetyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

HOURS(zeit)

##### Parameter

*Kommentar:* Zeit, *Typ:* Text

##### Beispiele

HOURS("10:5:2") ergibt 10

#### 8.1.4.20 ISLEAPYEAR

Die Funktion ISLEAPYEAR() gibt den Wert „Wahr“ zurück, wenn das angegebene Jahr ein Schaltjahr ist.

*Rückgabetyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

ISLEAPYEAR(jahr)

##### Parameter

*Kommentar:* Jahr, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

ISLEAPYEAR(2000) ergibt Wahr

#### 8.1.4.21 ISOWEEKNUM

Die Funktion ISOWEEKNUM() gibt die Wochenzahl zurück, in welche das angegebene Datum fällt. Die Funktion richtet sich nach dem ISO8601 Standard. Das bedeutet, dass die Woche immer mit dem Montag beginnt, und die erste Woche im Jahr die Woche ist, welche den ersten Donnerstag enthält.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

ISOWEEKNUM(Datum)

##### Parameter

*Kommentar:* Datum, *Typ:* Text

##### Beispiele

ISOWEEKNUM(A1) ergibt 51 wenn A1 „21st of Dec“ enthält.

##### Ähnliche Funktionen

[WEEKNUM](#)

#### 8.1.4.22 MINUTE

Die Funktion MINUTE gibt die Minuten einer Zeit zurück. Wird kein Parameter angegeben gibt die Funktion die aktuelle Minute zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

MINUTE(Zeit)

##### Parameter

*Kommentar:* Zeit, *Typ:* Text

##### Beispiele

MINUTE("22:10:12") ergibt 10

##### Beispiele

MINUTE(0,1234) ergibt 57

##### Ähnliche Funktionen

[HOUR](#)  
[SECOND](#)

#### 8.1.4.23 MINUTES

Die Funktion MINUTES() gibt die Anzahl der Minuten in einem Zeitausdruck zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

MINUTES(zeit)

##### Parameter

*Kommentar:* Zeit, *Typ:* Text

##### Beispiele

MINUTES("10:5:2") ergibt 5

#### 8.1.4.24 MONTH

Die Funktion MONTH gibt den Monat eines Datums zurück. Wird kein Parameter angegeben gibt die Funktion den aktuellen Monat zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

MONTH(Datum)

##### Parameter

*Kommentar:* Datum, *Typ:* Text

##### Beispiele

MONTH("2/22/2002") ergibt 2

##### Beispiele

MONTH(2323,1285) ergibt 5

##### Ähnliche Funktionen

DAY  
YEAR

#### 8.1.4.25 MONTHNAME

Die Funktion MONTHNAME() gibt den Namen des Monats zurück (1..12).

*Rückgabotyp:* Text

##### Syntax

MONTHNAME(Zahl)

##### Parameter

*Kommentar:* Nummer des Monats (1..12), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

MONTHNAME(5) ergibt Mai

#### 8.1.4.26 MONTHS

Die Funktion MONTHS() ergibt die Differenz in Monaten zwischen zwei Daten. Der dritte Parameter gibt den Berechnungsmodus an: Wenn der Modus 0 ist, gibt die Funktion die maximal mögliche Anzahl an Monaten zwischen den gegebenen Tagen zurück. Wenn der Modus 1 ist, wird die Anzahl an kompletten Kalendermonaten zurückgegeben.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

MONTHS(datum1; datum2; modus)

##### Parameter

*Kommentar:* Erster (früherer) Datumswert, *Typ:* Text

*Kommentar:* Zweiter Datumswert, *Typ:* Text

*Kommentar:* Berechnungsmodus, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

MONTHS("18.01.2002"; "26.02.2002"; 0) ergibt 1, weil 1 Monat und 8 Tage zwischen den beiden Daten liegen.

### Beispiele

MONTHS("18.01.2002"; "26.02.2002"; 1) ergibt 0, weil kein ganzer Kalendermonat zwischen den beiden Daten liegt.

#### 8.1.4.27 NETWORKDAY

Die Funktion NETWORKDAY() ergibt die Anzahl an Arbeitstagen zwischen dem Start- und dem Enddatum.

Urlaub muss wie folgt angegeben werden: Anzahl hinzuzufügender Tage, ein einzelnes Datum oder ein Datumsbereich.

*Rückgabetyyp:* Ganze Zahl (z. B. 1, 132, 2344)

### Syntax

NETWORKDAY(Startdatum; Enddatum; Urlaub)

### Parameter

*Kommentar:* Startdatum, *Typ:* Text

*Kommentar:* Enddatum, *Typ:* Text

*Kommentar:* Urlaub, *Typ:* Text

### Beispiele

NETWORKDAY("01.01.2001";"08.01.2001") ergibt 5 Arbeitstage

### Beispiele

NETWORKDAY("01.01.2001";"08.01.2001";2) ergibt 3 Arbeitstage

#### 8.1.4.28 NOW

Die Funktion NOW() liefert das aktuelle Datum mit lokalen Parameter formatiert zurück. Sie ist identisch mit der Funktion CURRENTDATETIME und wird aus Kompatibilitätsgründen mit anderen Anwendungen angeboten.

*Rückgabetyyp:* Datum

### Syntax

NOW()

### Parameter

### Beispiele

NOW() ergibt z. B. „Samstag, 13. April 2002 19:12:01“

### Ähnliche Funktionen

[CURRENTTIME](#)

[TODAY](#)

#### 8.1.4.29 SECOND

Die Funktion SECOND gibt die Sekunden einer Zeit zurück. Wird kein Parameter angegeben gibt die Funktion die aktuellen Sekunden zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

SECOND(Zeit)

##### Parameter

*Kommentar:* Zeit, *Typ:* Text

##### Beispiele

SECOND("22:10:12") ergibt 12

##### Beispiele

SECOND(0,1234) ergibt 42

##### Ähnliche Funktionen

[HOUR](#)  
[MINUTE](#)

#### 8.1.4.30 SECONDS

Die Funktion SECONDS() gibt die Anzahl der Sekunden in einem angegebenen Zeitausdruck zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

SECONDS(zeit)

##### Parameter

*Kommentar:* Zeit, *Typ:* Text

##### Beispiele

SECONDS("10:5:2") ergibt 2

#### 8.1.4.31 TIME

Die Funktion TIME() gibt die Zeit entsprechend den Landeseinstellungen formatiert zurück.

*Rückgabotyp:* Text

##### Syntax

TIME(Stunden;Minuten;Sekunden)

##### Parameter

*Kommentar:* Stunden, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Minuten, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Sekunden, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

TIME(10;2;2) ergibt 10:02:02

### Beispiele

TIME(10;70;0) ergibt 11:10:00

### Beispiele

TIME(10;-40;0) ergibt 9:20:00

#### 8.1.4.32 TIMEVALUE

Die Funktion TIMEVALUE() gibt eine Zahl (zwischen 0 und 1) zurück, welche die Zeit des Tages repräsentiert.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

TIMEVALUE(Zeit)

#### Parameter

*Kommentar:* Zeit, *Typ:* Text

#### Beispiele

TIMEVALUE("10:05:02") ergibt 0,42

#### Ähnliche Funktionen

[DATEVALUE](#)

#### 8.1.4.33 TODAY

Die Funktion TODAY() liefert das aktuelle Datum zurück.

*Rückgabotyp:* Datum

#### Syntax

TODAY()

#### Parameter

#### Beispiele

TODAY() ergibt „Samstag, 13. April 2002“

#### Ähnliche Funktionen

[CURRENTTIME](#)

[NOW](#)

#### 8.1.4.34 UNIX2DATE

Die Funktion UNIX2DATE() konvertiert die Unix-Zeit in einen Datums- und Zeitwert.

Die Unix-Zeit ist die Anzahl an Sekunden nach dem 1. Januar 1970, 00:00 Uhr UTC.

*Rückgabotyp:* Datum

#### Syntax

UNIX2DATE(Unix-Zeit)

#### Parameter

*Kommentar:* Unix-Zeit, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

UNIX2DATE(0) ergibt 01.01.1970

#### 8.1.4.35 WEEKDAY

Die WEEKDAY() Funktion gibt den Wochentag für das angegebene Datum zurück. Wenn die Berechnungsmethode 1 ist (Standard), gibt die Funktion 1 für Sonntag, 2 für Montag, ... usw. zurück. Bei Berechnungsmethode 2 ist Montag 1, Dienstag 2, ... usw. Und bei Berechnungsmethode 3 ist Montag 0, Dienstag 1, ... usw.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

WEEKDAY(Datum; Berechnungsmethode)

##### Parameter

*Kommentar:* Datum, *Typ:* Text

*Kommentar:* Berechnungsmethode (optional), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

WEEKDAY("2002-02-22"; 2) ergibt 5 für Freitag

##### Ähnliche Funktionen

[DAYNAME](#)

#### 8.1.4.36 WEEKNUM

Die Funktion WEEKNUM() gibt die Nicht-ISO-Wochenzahl zurück, in welche das angegebene Datum fällt.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

WEEKNUM(Datum; Berechnungsmethode)

##### Parameter

*Kommentar:* Datum, *Typ:* Text

*Kommentar:* Berechnungsmethode (optional), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

WEEKNUM(A1; 1) ergibt 11, wenn A1 der „9. März 2008“ ist. Die Wochenzahl wird davon ausgehend berechnet, dass die Woche an einem Sonntag (1) beginnt. Dies ist Standard, wenn keine Methode übergeben wird.

##### Beispiele

WEEKNUM(A1; 2) ergibt 10, wenn A1 der „9. März 2008“ ist. Die Wochenzahl wird davon ausgehend berechnet, dass die Woche an einem Montag (2) beginnt.

##### Ähnliche Funktionen

[ISOWEEKNUM](#)

#### 8.1.4.37 WEEKS

Die Funktion WEEKS() ergibt die Differenz in Wochen zwischen zwei Daten. Der dritte Parameter gibt den Berechnungsmodus an: Wenn der Modus 0 ist, gibt die Funktion die maximal mögliche Anzahl an Wochen zwischen den gegebenen Daten zurück. Wenn der Modus 1 ist, wird die Anzahl an kompletten Kalenderwochen zurückgegeben.

*Rückgabety*p: Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

WEEKS(datum1; datum2; modus)

##### Parameter

*Kommentar*: Erster (früherer) Datumswert, *Typ*: Text

*Kommentar*: Zweiter Datumswert, *Typ*: Text

*Kommentar*: Berechnungsmodus, *Typ*: Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

WEEKS("18.02.2002"; "26.02.2002"; 0) ergibt 1, weil 1 Woche und 1 Tag zwischen den beiden Daten liegen.

##### Beispiele

WEEKS("19.02.2002"; "26.02.2002"; 0) ergibt 0, weil keine komplette Woche zwischen den beiden Daten liegt. Eine Woche beginnt, abhängig von den Landeseinstellungen, am Montag oder am Sonntag.

#### 8.1.4.38 WEEKSINYEAR

Die Funktion WEEKSINYEAR() gibt die Anzahl der Wochen eines angegebenen Jahres zurück.

*Rückgabety*p: Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

WEEKSINYEAR(jahr)

##### Parameter

*Kommentar*: Jahr, *Typ*: Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

WEEKSINYEAR(2000) ergibt 52

#### 8.1.4.39 WORKDAY

Die Funktion WORKDAY() ergibt den ersten Arbeitstag ausgehend vom Startdatum.

Urlaub muss wie folgt angegeben werden: Anzahl hinzuzufügender Tage, ein einzelnes Datum oder ein Datumsbereich.

*Rückgabety*p: Datum

##### Syntax

WORKDAY(Startdatum, Tage, Urlaub)

##### Parameter

*Kommentar*: Startdatum, *Typ*: Text

*Kommentar*: Arbeitstage, *Typ*: Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar*: Urlaub, *Typ*: Text

### Beispiele

Mit B9 „01.01.2001“, D3 „03.01.2001“ und D4 „04.01.2001“ ergibt WORKDAY(B9;2;D3:D4) als Arbeitstag „05.01.2001“

#### 8.1.4.40 YEAR

Die Funktion YEAR gibt das Jahr eines Datums zurück. Wird kein Datum angegeben wird das aktuelle Jahr zurückgegeben.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

YEAR(Datum)

#### Parameter

*Kommentar:* Datum, *Typ:* Text

#### Beispiele

YEAR("2/22/2002") ergibt 2002

#### Beispiele

YEAR(2323,1285) gibt 1906

#### Ähnliche Funktionen

[DAY](#)

[MONTH](#)

#### 8.1.4.41 YEARFRAC

Die Funktion YEARFRAC() ergibt die Anzahl an vollen Tagen zwischen dem Start- und dem Enddatum, abhängig von der Basis.

Die Basis muss eine der folgenden sein: 0 = 30/360 US, 1 = Tatsächlich/tatsächlich, 2 = Tatsächlich/360, 3 = Tatsächlich/365, 4 = 30/360 europäisch

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

YEARFRAC(Startdatum; Enddatum; Basis)

#### Parameter

*Kommentar:* Erstes Datum, *Typ:* Text

*Kommentar:* Zweites Datum, *Typ:* Text

*Kommentar:* Intervall, *Typ:* Text

#### 8.1.4.42 YEARS

Die Funktion YEARS() ergibt die Differenz in Jahren zwischen zwei Daten. Der dritte Parameter gibt den Berechnungsmodus an: Wenn der Modus 0 ist, gibt die Funktion die maximal mögliche Anzahl an Jahren zwischen den gegebenen Tagen zurück. Wenn der Modus 1 ist, wird die Anzahl der kompletten Kalenderjahre zurückgegeben. Diese beginnen am 1. Januar und enden am 31. Dezember.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

### Syntax

YEARS(datum1; datum2; modus)

### Parameter

*Kommentar:* Erster (früherer) Datumswert, *Typ:* Text

*Kommentar:* Zweiter Datumswert, *Typ:* Text

*Kommentar:* Berechnungsmodus, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

YEARS("19.02.2002"; "26.02.2002"; 0) ergibt 1, weil 1 Jahr und 7 Tage zwischen den beiden Daten liegen.

### Beispiele

YEARS("19.02.2002"; "26.02.2002"; 1) ergibt 0, weil kein komplettes Kalenderjahr zwischen den beiden Daten liegt.

## 8.1.5 Ingenieurwesen

### 8.1.5.1 BASE

Die Funktion BASE() konvertiert eine Zahl zur Basis 10 in eine Zeichenfolge, die eine Zahl zu einer Zielbasis zwischen 2 und 36 darstellt.

*Rückgabotyp:* Text

### Syntax

BASE(Zahl;Basis;Genauigkeit)

### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* MinLänge, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

BASE(128;8) ergibt „200“

### 8.1.5.2 BESSELI

Die Funktion BESSELI() gibt die modifizierte „Bessel“ Funktion  $\ln(x)$  zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

BESSELI(X;N)

### Parameter

*Kommentar:* Wo die Funktion ausgewertet wird, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Reihenfolge der Funktion, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

BESSELI(0,7;3) ergibt 0,007367374

### Ähnliche Funktionen

[BESSELJ](#)

[BESSELK](#)

[BESSELY](#)

### 8.1.5.3 BESSELJ

Die Funktion BESSELJ() gibt die „Bessel“ Funktion zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

BESSELJ(X;N)

#### Parameter

*Kommentar:* Wo die Funktion ausgewertet wird, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Reihenfolge der Funktion, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

BESSELJ(0,89;3) ergibt 0,013974004

#### Ähnliche Funktionen

[BESSELI](#)  
[BESSELK](#)  
[BESSELY](#)

### 8.1.5.4 BESSELK

Die Funktion BESSELK() gibt die modifizierte „Bessel“ Funktion zurück, welche der „Bessel“ Funktion entspricht, die für Imaginäre Zahlen verwendet wird.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

BESSELK(X;N)

#### Parameter

*Kommentar:* Wo die Funktion ausgewertet wird, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Reihenfolge der Funktion, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

BESSELK(3;9) ergibt 397,95880

#### Ähnliche Funktionen

[BESSELI](#)  
[BESSELJ](#)  
[BESSELY](#)

### 8.1.5.5 BESSELY

Die Funktion BESSELY() gibt die „Bessel“ Funktion zurück, welche auch die „Weber“ oder die „Neumann“ Funktion genannt wird.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

BESSELY(X;N)

**Parameter**

*Kommentar:* Wo die Funktion ausgewertet wird, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Reihenfolge der Funktion, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

**Beispiele**

BESSELY(4;2) ergibt 0,215903595

**Ähnliche Funktionen**

BESSELI  
BESSELJ  
BESSELK

**8.1.5.6 BIN2DEC**

Die Funktion BIN2DEC() gibt den Wert als dezimal-formatierte Zahl zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

**Syntax**

BIN2DEC(Wert)

**Parameter**

*Kommentar:* Der zu konvertierende Wert, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

**Beispiele**

BIN2DEC("1010") ergibt 10

**Beispiele**

BIN2DEC("11111") ergibt 31

**8.1.5.7 BIN2HEX**

Die Funktion BIN2HEX() gibt den Wert als hexadezimal-formatierte Zahl zurück.

*Rückgabotyp:* Text

**Syntax**

BIN2HEX(Wert)

**Parameter**

*Kommentar:* Der zu konvertierende Wert, *Typ:* Text

*Kommentar:* Die minimale Länge der Ausgabe, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

**Beispiele**

BIN2HEX("1010") ergibt "a"

**Beispiele**

BIN2HEX("11111") ergibt "1f"

### 8.1.5.8 BIN2OCT

Die Funktion BIN2OCT() gibt den Wert als oktal-formatierte Zahl zurück.

*Rückgabetyt:* Text

#### Syntax

BIN2OCT(Wert)

#### Parameter

*Kommentar:* Der zu konvertierende Wert, *Typ:* Text

*Kommentar:* Die minimale Länge der Ausgabe, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

BIN2OCT("1010") ergibt "12"

#### Beispiele

BIN2OCT("11111") ergibt 37

### 8.1.5.9 COMPLEX

Die Funktion COMPLEX(Realteil; Imaginärteil) gibt eine komplexe Zahl in der Form  $x+yi$  zurück.

*Rückgabetyt:* Text

#### Syntax

COMPLEX(Realteil;Imaginärteil)

#### Parameter

*Kommentar:* Realer Koeffizient, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Imaginärer Koeffizient, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

COMPLEX(1,2;3,4) ergibt „1,2+3,4i“

#### Beispiele

COMPLEX(0;-1) ergibt „-i“

### 8.1.5.10 CONVERT

Die Funktion CONVERT() gibt die Konvertierung von einem Maßsystem in ein anderes zurück.

Unterstützte Gewichtseinheiten: g (Gramm), sg (Stück), lbm (Pfund), u (Atommasse), ozm (Unze), stone (entspricht 14 Pfund), ton (Tonne), grain (1 Unze = 480 grains), pweight (pennyweight), hweight (Zentner).

Unterstützte Längeneinheiten: m (Meter), in (Zoll), ft (Fuß), mi (Meile), Nmi (Nautische Meile), ang (Ångström), parsec, lightyear (Lichtjahr).

Unterstützte Druckeinheiten: Pa (Pascal), atm (Atmosphären), mmHg (Millimeter-Quecksilbersäule), psi (Pfund pro Quadratzoll), Torr.

Unterstützte Kräfteinheiten: N (Newton), dyn, pound (Pfund).

Unterstützte Energieeinheiten: J (Joule), e (erg), c (Thermodynamische Kalorie), cal (I.T.-Kalorie), eV (Elektrovolt), HPh (Pferdestärke pro Stunde), Wh (Wattstunde), flb (Fuß-Pfund), BTU (British thermal unit).

Unterstützte Leistungseinheiten: W (Watt), HP (Horsepower), PS (Pferdestärke).

Unterstützte Magnetismus-Einheiten: T (Tesla), ga (Gauß).

Unterstützte Temperatur-Einheiten: C (Celsius), F (Fahrenheit), K (Kelvin).

Unterstützte Volumeneinheiten: l (Liter), tsp (Teelöffel), tbs (Esslöffel), oz (Flüssigunze), cup (Tasse), pt (Pint), qt (Quart), gal (Gallone), barrel, m3 (Kubikmeter), mi3 (Kubikmeile), Nmi3 (Nautische Kubikmeile), in3 (Kubikzoll), ft3 (Kubikfuß), yd3 (Kubik-Yard), GRT oder regton (Bruttoregistertonne)

Unterstützte Flächeneinheiten: m2 (Quadratmeter), mi2 (Quadratmeile), Nmi2 (Quadrat-Nautische-Meile), in2 (Quadratzoll), ft2 (Quadratfuß), yd2 (Quadrat-Yard), acre, ha (Hektar).

Unterstützte Geschwindigkeitseinheiten: m/s (Meter pro Sekunde), m/h (Meter pro Stunde), mph (Meilen pro Stunde), kn (Knoten).

Für metrische Einheiten kann jeder der folgenden Präfixe verwendet werden: E (exa, 1E+18), P (peta, 1E+15), T (tera, 1E+12), G (giga, 1E+09), M (mega, 1E+06), k (kilo, 1E+03), h (hecto, 1E+02), e (deka, 1E+01), d (deci, 1E-01), c (centi, 1E-02), m (milli, 1E-03), u (micro, 1E-06), n (nano, 1E-09), p (pico, 1E-12), f (femto, 1E-15), a (atto, 1E-18).

*Rückgabetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

CONVERT(Zahl; Ausgangseinheit; Zieleinheit)

### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Ausgangseinheit, *Typ:* Text

*Kommentar:* Zieleinheit, *Typ:* Text

### Beispiele

CONVERT(32;"C";"F") ergibt 89,6

### Beispiele

CONVERT(3;"lbm";"kg") ergibt 1,3608

### Beispiele

CONVERT(7,9;"cal";"J") ergibt 33,0757

## 8.1.5.11 DEC2BIN

Die Funktion DEC2BIN() gibt den Wert als binär-formatierte Zahl zurück.

*Rückgabetyp:* Text

### Syntax

DEC2BIN(Wert)

### Parameter

*Kommentar:* Der zu konvertierende Wert, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Die minimale Länge der Ausgabe, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

DEC2BIN(12) ergibt „1100“

### Beispiele

DEC2BIN(55) ergibt „110111“

#### 8.1.5.12 DEC2HEX

Die Funktion DEC2HEX() gibt den Wert als hexadezimal-formatierte Zahl zurück.

*Rückgabebetyp:* Text

##### Syntax

DEC2HEX(Wert)

##### Parameter

*Kommentar:* Der zu konvertierende Wert, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Die minimale Länge der Ausgabe, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

DEC2HEX(12) ergibt „c“

##### Beispiele

DEC2HEX(55) ergibt „37“

#### 8.1.5.13 DEC2OCT

Die Funktion DEC2OCT() gibt den Wert als oktal-formatierte Zahl zurück.

*Rückgabebetyp:* Text

##### Syntax

DEC2OCT(Wert)

##### Parameter

*Kommentar:* Der zu konvertierende Wert, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Die minimale Länge der Ausgabe, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

DEC2OCT(12) ergibt „14“

##### Beispiele

DEC2OCT(55) ergibt „67“

#### 8.1.5.14 DELTA

Die Funktion DELTA() ergibt 1, wenn x gleich y ist. Sonst gibt sie 0 zurück. y ist standardmässig 0.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

DELTA(x; y)

##### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

DELTA(1,2; 3,4) ergibt 0

##### Beispiele

DELTA(3; 3) ergibt 1

##### Beispiele

DELTA(1; Wahr) ergibt 1

#### 8.1.5.15 ERF

Die Funktion ERF() ergibt die Fehlerfunktion. Wird ein Argument übergeben, gibt ERF() die Fehlerfunktion zwischen 0 und dem Argument zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

ERF(Unteres Limit; Oberes Limit)

##### Parameter

*Kommentar:* Unteres Limit, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Oberes Limit, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

ERF(0.4) ergibt 0,42839236

##### Ähnliche Funktionen

[ERFC](#)

#### 8.1.5.16 ERFC

Die Funktion ERFC() gibt die komplementäre Fehlerfunktion zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

ERFC(Unteres Limit; Oberes Limit)

##### Parameter

*Kommentar:* Unteres Limit, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Oberes Limit, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

ERFC(0;4) ergibt 0,57160764

##### Ähnliche Funktionen

[ERF](#)

#### 8.1.5.17 GESTEP

Die Funktion GESTEP() gibt 1 zurück, wenn x größer oder gleich y ist. Sonst gibt sie 0 zurück. y ist standardmäßig 0.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

GESTEP(x; y)

##### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

GESTEP(1,2; 3,4) ergibt 0

**Beispiele**

GESTEP(3; 3) ergibt 1

**Beispiele**

GESTEP(0,4; WAHR) ergibt 0

**Beispiele**

GESTEP(4; 3) ergibt 1

**8.1.5.18 HEX2BIN**

Die Funktion HEX2BIN() gibt den Wert als binär-formatierte Zahl zurück.

*Rückgabotyp:* Text

**Syntax**

HEX2BIN(Wert)

**Parameter**

*Kommentar:* Der zu konvertierende Wert, *Typ:* Text

**Beispiele**

HEX2BIN("a") ergibt "1010"

**Beispiele**

HEX2BIN("37") ergibt "110111"

**8.1.5.19 HEX2DEC**

Die Funktion HEX2DEC() gibt den Wert als dezimal-formatierte Zahl zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

**Syntax**

HEX2DEC(Wert)

**Parameter**

*Kommentar:* Der zu konvertierende Wert, *Typ:* Text

**Beispiele**

HEX2DEC("a") ergibt 10

**Beispiele**

HEX2DEC("37") ergibt 55

#### 8.1.5.20 HEX2OCT

Die Funktion HEX2OCT() gibt den Wert als oktal-formatierte Zahl zurück.

*Rückgabebetyp:* Text

##### Syntax

HEX2OCT(Wert)

##### Parameter

*Kommentar:* Der zu konvertierende Wert, *Typ:* Text

##### Beispiele

HEX2OCT("a") ergibt "12"

##### Beispiele

HEX2OCT("37") ergibt 67

#### 8.1.5.21 IMABS

Die Funktion IMABS(komplexe Zahl) gibt die Norm einer komplexen Zahl der Form  $x+yi$  zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

IMABS(komplexe Zahl)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

##### Beispiele

IMABS("1,2+5i") ergibt 5,1419

##### Beispiele

IMABS("-i") ergibt 1

##### Beispiele

IMABS("12") ergibt 12

#### 8.1.5.22 IMAGINARY

Die Funktion IMAGINARY(Zeichenfolge) gibt den imaginären Koeffizienten einer komplexen Zahl zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

IMAGINARY(Zeichenfolge)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

##### Beispiele

IMAGINARY("1,2+3,4i") ergibt 3,4

##### Beispiele

IMAGINARY("1,2") ergibt 0

### 8.1.5.23 IMARGUMENT

Die Funktion IMARGUMENT(komplexe Zahl) gibt das Argument einer komplexen Zahl der Form  $x+yi$  zurück.

*Rückgabotyp:* Text

#### Syntax

IMARGUMENT(komplexe Zahl)

#### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

#### Beispiele

IMARGUMENT("1,2+5i") ergibt 0,6072

#### Beispiele

IMARGUMENT("-i") ergibt -1,57079633

#### Beispiele

IMARGUMENT("12") ergibt "#Div/0"

### 8.1.5.24 IMCONJUGATE

Die Funktion IMCONJUGATE(komplexe Zahl) ergibt die zu einer komplexen Zahl der Form  $x+yi$  konjugiert komplexe Zahl.

*Rückgabotyp:* Text

#### Syntax

IMCONJUGATE(komplexe Zahl)

#### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

#### Beispiele

IMCONJUGATE("1,2+5i") ergibt "1,2-5i"

#### Beispiele

IMCONJUGATE("-i") ergibt "i"

#### Beispiele

IMCONJUGATE("12") ergibt "12"

### 8.1.5.25 IMCOS

Die Funktion IMCOS(Zeichenfolge) gibt den Kosinus einer komplexen Zahl zurück.

*Rückgabotyp:* Text

#### Syntax

IMCOS(Zeichenfolge)

**Parameter**

*Kommentar:* Komplexe Zahl, *Typ:* Text

**Beispiele**

IMCOS("1+i") ergibt "0,83373-0,988898i"

**Beispiele**

IMCOS("12i") ergibt 81 377,4

**8.1.5.26 IMCOSH**

Die Funktion IMCOSH(Zeichenfolge) gibt den Kosinus Hyperbolicus einer komplexen Zahl zurück.

*Rückgabetyt:* Text

**Syntax**

IMCOSH(Zeichenfolge)

**Parameter**

*Kommentar:* Komplexe Zahl, *Typ:* Text

**Beispiele**

IMCOSH("1+i") ergibt "0,83373+0,988898i"

**Beispiele**

IMCOSH("12i") ergibt 0,84358

**8.1.5.27 IMCOT**

Die Funktion IMCOT(Zeichenfolge) gibt den Kotangens einer komplexen Zahl zurück.

*Rückgabetyt:* Text

**Syntax**

IMCOT(Zeichenfolge)

**Parameter**

*Kommentar:* Komplexe Zahl, *Typ:* Text

**Beispiele**

IMCOT("1+i") ergibt "0.21762-0.86801i"

**8.1.5.28 IMCSC**

Die Funktion IMCSC(Zeichenfolge) gibt den Kosekans einer komplexen Zahl zurück.

*Rückgabetyt:* Text

**Syntax**

IMCSC(Zeichenfolge)

**Parameter**

*Kommentar:* Komplexe Zahl, *Typ:* Text

**Beispiele**

IMCSC("1+i") ergibt "0.62151-0.30393i"

### 8.1.5.29 IMCSCH

Die Funktion IMCSCH(Zeichenfolge) gibt den Kosekans Hyperbolicus einer komplexen Zahl zurück.

*Rückgabety*p: Text

#### Syntax

IMCSCH(Zeichenfolge)

#### Parameter

*Kommentar*: Komplexe Zahl, *Typ*: Text

#### Beispiele

IMCSCH("1+i") ergibt "0.30393-i0.62151"

### 8.1.5.30 IMDIV

Die Funktion IMDIV() gibt den Quotienten mehrerer komplexer Zahlen der Form  $x+yi$  zurück. Die erste Zahl wird also der Reihe nach durch die anderen Zahlen geteilt.

*Rückgabety*p: Text

#### Syntax

IMDIV(Wert;Wert;...)

#### Parameter

*Kommentar*: Komplexe Zahl, *Typ*: Eine Zeichenfolge

#### Beispiele

IMDIV(1,2;"3,4+5i") ergibt "0,111597-0,164114i"

#### Beispiele

IMDIV("12+i";"12-i") ergibt "0,986207+0,16551i"

### 8.1.5.31 IMEXP

Die Funktion IMEXP(Zeichenfolge) gibt den Exponenten einer komplexen Zahl zurück.

*Rückgabety*p: Text

#### Syntax

IMEXP(Zeichenfolge)

#### Parameter

*Kommentar*: Komplexe Zahl, *Typ*: Text

#### Beispiele

IMEXP("2-i") ergibt "3,99232-6,21768i"

#### Beispiele

IMEXP("12i") ergibt 0,843854-0,536573i

#### 8.1.5.32 IMLN

Die Funktion IMLN(Zeichenfolge) gibt den natürlichen Logarithmus einer komplexen Zahl zurück.

*Rückgabetyyp:* Text

##### Syntax

IMLN(Zeichenfolge)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

##### Beispiele

IMLN("3-i") ergibt "1,15129-0,321751i"

##### Beispiele

IMLN("12") ergibt 2,48491

#### 8.1.5.33 IMLOG10

Die Funktion IMLOG10(Zeichenfolge) gibt den dekadischen Logarithmus einer komplexen Zahl zurück.

*Rückgabetyyp:* Text

##### Syntax

IMLOG10(Zeichenfolge)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

##### Beispiele

IMLOG10("3+4i") ergibt "0,69897+0,402719i"

#### 8.1.5.34 IMLOG2

Die Funktion IMLOG2(Zeichenfolge) gibt den binären Logarithmus einer komplexen Zahl zurück.

*Rückgabetyyp:* Text

##### Syntax

IMLOG2(Zeichenfolge)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

##### Beispiele

IMLOG2("3+4i") ergibt "2,321928+1,337804i"

#### 8.1.5.35 IMPOWER

Die Funktion IMPOWER(Zeichenfolge) gibt eine komplexe Zahl potenziert mit einem Wert zurück.

*Rückgabetyp:* Text

##### Syntax

IMPOWER(Zeichenfolge)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

*Kommentar:* Potenz, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

IMPOWER("4-i";2) ergibt "15-8i"

##### Beispiele

IMPOWER("1,2";2) ergibt 1,44

#### 8.1.5.36 IMPRODUCT

Die Funktion IMPRODUCT() gibt das Produkt mehrerer komplexer Zahlen der Form  $x+yi$  zurück.

*Rückgabetyp:* Text

##### Syntax

IMPRODUCT(Wert;Wert;...)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Eine Zeichenfolge

##### Beispiele

IMPRODUCT(1,2;"3,4+5i") ergibt "4,08+6i"

##### Beispiele

IMPRODUCT(1,2;"1i") ergibt "+1,2i"

#### 8.1.5.37 IMREAL

Die Funktion IMREAL(Zeichenfolge) gibt den reellen Koeffizienten einer komplexen Zahl zurück.

*Rückgabetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

IMREAL(Zeichenfolge)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

**Beispiele**

IMREAL("1,2+3,4i") ergibt 1,2

**Beispiele**

IMREAL("1,2i") ergibt 0

**8.1.5.38 IMSEC**

Die Funktion IMSEC(Zeichenfolge) gibt den Sekans einer komplexen Zahl zurück.

*Rückgabetyt:* Text

**Syntax**

IMSEC(Zeichenfolge)

**Parameter**

*Kommentar:* Komplexe Zahl, *Typ:* Text

**Beispiele**

IMSEC("1+i") ergibt "0.49833+i0.59108"

**8.1.5.39 IMSECH**

Die Funktion IMSECH(Zeichenfolge) gibt den Sekans Hyperbolicus einer komplexen Zahl zurück.

*Rückgabetyt:* Text

**Syntax**

IMSECH(Zeichenfolge)

**Parameter**

*Kommentar:* Komplexe Zahl, *Typ:* Text

**Beispiele**

IMSECH("1+i") ergibt "0.49833-i0.59108"

**8.1.5.40 IMSIN**

Die Funktion IMSIN(Zeichenfolge) gibt den Sinus einer komplexen Zahl zurück.

*Rückgabetyt:* Text

**Syntax**

IMSIN(Zeichenfolge)

**Parameter**

*Kommentar:* Komplexe Zahl, *Typ:* Text

**Beispiele**

IMSIN("1+i") ergibt "1,29846+0,634964i"

**Beispiele**

IMSIN("1,2") ergibt -0,536573

#### 8.1.5.41 IMSINH

Die Funktion IMSINH(Zeichenfolge) gibt den Sinus Hyperbolicus einer komplexen Zahl zurück.

*Rückgabetyyp:* Text

##### Syntax

IMSINH(Zeichenfolge)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

##### Beispiele

IMSINH("1+i") ergibt "0,63496+1,29846i"

##### Beispiele

IMSINH("1,2") ergibt 1,50946

#### 8.1.5.42 IMSQRT

Die Funktion IMSQRT(Zeichenfolge) gibt die Quadratwurzel einer komplexen Zahl zurück.

*Rückgabetyyp:* Text

##### Syntax

IMSQRT(Zeichenfolge)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

##### Beispiele

IMSQRT("1+i") ergibt "1,09868+0,45509i"

##### Beispiele

IMSQRT("1,2i") ergibt 0,774597+0,774597i

#### 8.1.5.43 IMSUB

Die Funktion IMSUB() gibt die Differenz mehrerer komplexer Zahlen der Form  $x+yi$  zurück.

*Rückgabetyyp:* Text

##### Syntax

IMSUB(Wert;Wert;...)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Eine Zeichenfolge

##### Beispiele

IMSUB(1,2;"3,4+5i") ergibt "-2,2-5i"

##### Beispiele

IMSUB(1,2;"1i") ergibt "1,2-i"

#### 8.1.5.44 IMSUM

Die Funktion IMSUM() gibt die Summe mehrerer komplexer Zahlen der Form  $x+yi$  zurück.

*Rückgabetyyp:* Text

##### Syntax

IMSUM(Wert;Wert;...)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Eine Zeichenfolge

##### Beispiele

IMSUM(1,2;"3,4+5i") ergibt "4,6+5i"

##### Beispiele

IMSUM(1,2;"1i") ergibt "1,2+i"

#### 8.1.5.45 IMTAN

Die Funktion IMTAN(Zeichenfolge) gibt den Tangens einer komplexen Zahl zurück.

*Rückgabetyyp:* Text

##### Syntax

IMTAN(Zeichenfolge)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

##### Beispiele

IMTAN("1+i") ergibt "0,27175+1,08392i"

##### Beispiele

IMTAN("1,2") ergibt 2,57215

#### 8.1.5.46 IMTANH

Die Funktion IMTANH(Zeichenfolge) gibt den Tangens Hyperbolicus einer komplexen Zahl zurück.

*Rückgabetyyp:* Text

##### Syntax

IMTANH(Zeichenfolge)

##### Parameter

*Kommentar:* Komplexe Zahl, *Typ:* Text

##### Beispiele

IMTANH("1+i") ergibt "1,08392+0,27175i"

##### Beispiele

IMTANH("1,2") ergibt 0,83365

#### 8.1.5.47 OCT2BIN

Die Funktion OCT2BIN() gibt den Wert als binär-formatierte Zahl zurück.

*Rückgabebetyp:* Text

##### Syntax

OCT2BIN(Wert)

##### Parameter

*Kommentar:* Der zu konvertierende Wert, *Typ:* Text

*Kommentar:* Die minimale Länge der Ausgabe, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

OCT2BIN("12") ergibt "1010"

##### Beispiele

OCT2BIN("55") ergibt "101101"

#### 8.1.5.48 OCT2DEC

Die Funktion OCT2DEC() gibt den Wert als dezimal-formatierte Zahl zurück.

*Rückgabebetyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

OCT2DEC(Wert)

##### Parameter

*Kommentar:* Der zu konvertierende Wert, *Typ:* Text

##### Beispiele

OCT2DEC("12") ergibt 10

##### Beispiele

OCT2DEC("55") ergibt 45

#### 8.1.5.49 OCT2HEX

Die Funktion OCT2HEX() gibt den Wert als hexadezimal-formatierte Zahl zurück.

*Rückgabebetyp:* Text

##### Syntax

OCT2HEX(Wert)

##### Parameter

*Kommentar:* Der zu konvertierende Wert, *Typ:* Text

*Kommentar:* Die minimale Länge der Ausgabe, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

OCT2HEX("12") ergibt "A"

##### Beispiele

OCT2HEX("55") ergibt "2D"

## 8.1.6 Finanzen

### 8.1.6.1 ACCRINT

Die Funktion ACCRINT gibt die aufgewachsenen Zinsen für ein Wertpapier für welches regelmäßig Zinsen ausbezahlt werden. Erlaubte Werte für die Häufigkeit sind 1 für Jährlich, 2 für Halbjährlich und 4 für Vierteljährlich. Basis ist die Art der Tageszählung Sie verwenden möchten: 0 für US 30/360 (Standard), 1 für tatsächliche Tage, 2 für tatsächliche Tage/360, 3 für tatsächliche Tage/365 oder 4 für Europäisch 30/365.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ACCRINT(Ausgabe; erster Zinstermin; Abrechnung; Zins; Nennwert; Häufigkeit; Basis)

#### Parameter

*Kommentar:* Ausgabetag, *Typ:* Datum

*Kommentar:* Erster Zinstermin, *Typ:* Datum

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Jährlicher Zins eines Wertpapiers, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Nennwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Anzahl der Zahlungen pro Jahr, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis der Tageszählung, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

ACCRINT("2/28/2001"; "8/31/2001"; "5/1/2001"; 0,1; 1000; 2; 0) ergibt 16,944

#### Ähnliche Funktionen

[ACCRINTM](#)

### 8.1.6.2 ACCRINTM

Die Funktion ACCRINTM gibt die aufgewachsenen Zinsen für ein Wertpapier zurück, welches am Fälligkeitsdatum ausbezahlt wird. Basis ist die Art der Tageszählung Sie verwenden möchten: 0 für US 30/360 (Standard), 1 für tatsächliche Tage, 2 für tatsächliche Tage/360, 3 für tatsächliche Tage/365 oder 4 für Europäisch 30/365.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ACCRINTM(Ausgabe; Abrechnung; Zins; Nennwert; Basis)

#### Parameter

*Kommentar:* Ausgabetag, *Typ:* Datum

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Jährlicher Zins eines Wertpapiers, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Nennwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis der Tageszählung, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

ACCRINTM("2/28/2001"; "8/31/2001"; 0,1; 100) ergibt 5,0278

#### Ähnliche Funktionen

[ACCRINT](#)

### 8.1.6.3 AMORDEGRC

Die Funktion AMORDEGRC berechnet den Abschreibungswert für das französische Rechnungswesen bei degressiver Abschreibung.

*Rückgabetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

AMORDEGRC( Kosten; Anschaffungsdatum; Enddatum erste Periode; Restwert; Periode; Rate; Basis)

#### Parameter

*Kommentar:* Kosten, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Pv, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fv, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

AMORDEGRC( 1000; "01.02.2006"; "31.12.2006"; 10; 0; 0,1; 1 ) ergibt 228

#### Ähnliche Funktionen

[AMORLINC](#)

[DB](#)

[DDB](#)

[YEARFRAC](#)

### 8.1.6.4 AMORLINC

Die Funktion AMORLINC berechnet den Abschreibungswert für das französische Rechnungswesen bei linearer Abschreibung.

*Rückgabetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

AMORLINC( Kosten; Anschaffungsdatum; Enddatum erste Periode; Restwert; Periode; Rate; Basis)

#### Parameter

*Kommentar:* P, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Pv, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fv, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

AMORLINC( 1000; "01.02.2004"; "31.12.2004"; 10; 0; 0,1; 1 ) ergibt 91,256831

#### Ähnliche Funktionen

[AMORDEGRC](#)

[DB](#)

[DDB](#)

[YEARFRAC](#)

#### 8.1.6.5 COMPOUND

Die Funktion COMPOUND() gibt den Wert einer Anlage zurück. Der Wert wird dabei mit der Startkapital, dem nominalen Zinssatz, der Häufigkeit der Zinsberechnung und der Laufzeit berechnet. Beispiel: 5000 EUR bei 12 % Zins, vierteljährlich verzinst über 5 Jahr ergibt COMPOUND(5000;0,12;4;5) oder 9030,56 EUR.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

COMPOUND(Startkapital;Zins;Perioden;Perioden pro Jahr)

##### Parameter

*Kommentar:* Startkapital, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Zins, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Zinsperioden pro Jahr, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Jahre, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

COMPOUND(5000;0,12;4;5) ergibt 9030,56

#### 8.1.6.6 CONTINUOUS

Die Funktion CONTINUOUS() gibt den Endwert eines fortlaufend verzinsten Betrags zurück. Dabei erwartet sie das Startkapital, den nominalen Zinssatz und die Laufzeit in Jahren. Beispiel: 1000 EUR mit 10% Zins bei einem Jahr wird zu continuous(1000;0,1;1) oder 1105,17 EUR.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

CONTINUOUS(Startkapital;Zins;Jahre)

##### Parameter

*Kommentar:* Startkapital, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Zins, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Jahre, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

CONTINUOUS(1000;0,1;1) ergibt 1105,17

#### 8.1.6.7 COUPNUM

Die Funktion COUPNUM gibt die Anzahl der Gewinnanteilsscheinen zurück, die zwischen dem Abrechnungsdatum und dem Fälligkeitsdatum ausgezahlt werden. Basis ist die Art der Tageszählung Sie verwenden möchten: 0 für US 30/360 (Standard), 1 für tatsächliche Tage, 2 für tatsächliche Tage/360, 3 für tatsächliche Tage/365 oder 4 für Europäisch 30/365.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

COUPNUM(Abrechnung; Fälligkeit; Häufigkeit; Basis)

### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Berechnungshäufigkeit, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis der Tageszählung, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

COUPNUM("2/28/2001"; "8/31/2001"; 2; 0) ergibt 1

### 8.1.6.8 CUMIPMT

Berechnet die kumulative Zinszahlung.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

CUMIPMT(Zins, Perioden, Wert, Start, Ende, Typ)

### Parameter

*Kommentar:* Rate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Perioden, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Wert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Start, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Ende, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Typ, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

CUMIPMT( 0,06/12; 5\*12; 100000; 5; 12; 0 ) ergibt -3562,187023

### Ähnliche Funktionen

[IPMT](#)

[CUMPRINC](#)

### 8.1.6.9 CUMPRINC

Berechnet die kumulative Kapitalzahlung.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

CUMPRINC(Zins, Perioden, Wert, Start, Ende, Typ)

### Parameter

*Kommentar:* Rate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Perioden, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Wert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Start, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Ende, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Typ, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

CUMPRINC( 0,06/12; 5\*12; 100000; 5; 12; 0 ) ergibt -11904,054201

### Ähnliche Funktionen

[PPMT](#)

[CUMIPMT](#)

#### 8.1.6.10 DB

Die Funktion DB() berechnet die Abschreibung einer Anlage in einer gegebenen Periode mit der festen degressiven Abschreibungsmethode. „Monat“ ist optional, wenn er weggelassen wird, wird 12 als Wert angenommen.

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

DB(Kosten; Restwert; Lebensdauer; Periode [;Monat])

##### Parameter

*Kommentar*: Kosten, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Restwert, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Lebensdauer, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Periode, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Monat, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

DB(8000;400;6;3) ergibt 1158,40

##### Beispiele

DB(8000;400;6;3;2) ergibt 1783,41

##### Ähnliche Funktionen

[DDB](#)

[SLN](#)

#### 8.1.6.11 DDB

Die Funktion DDB() berechnet die Abschreibung von einem Vermögen für eine angegebene Periode. Dabei wird die arithmetische Degressionsmethode verwendet. Der Faktor ist optional, wenn er fehlt wird angenommen das er 2 ist. Alle Parameter müssen größer als null sein.

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

DDB(Kosten; Restwert; Lebensdauer; Periode [;Faktor])

##### Parameter

*Kommentar*: Kosten, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Restwert, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Lebensdauer, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Periode, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Faktor, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

DDB(75000;1;60;12;2) ergibt 1721,81

##### Ähnliche Funktionen

[SLN](#)

#### 8.1.6.12 DISC

Die Funktion DISC gibt den Disagio für ein Wertpapier zurück. Basis ist die Art der Tageszählung, die Sie verwenden möchten: 0 für US 30/360 (Standard), 1 für tatsächliche Tage, 2 für tatsächliche Tage/360, 3 für tatsächliche Tage/365 oder 4 für Europäisch 30/365.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

DISC(Abrechnung; Fälligkeit; Nennwert; Auszahlung [; Basis ] )

##### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Kurs pro \$100 Nennwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Auszahlung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis der Tageszählung, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

DISC("2/28/2001"; "8/31/2001"; 12; 14) ergibt 0,2841

##### Ähnliche Funktionen

[YEARFRAC](#)

#### 8.1.6.13 DOLLARDE

Die Funktion DOLLARDE() gibt eine Notierung, die als Dezimalbruch ausgedrückt wurde, als Dezimalzahl zurück. Die Zahl ist dabei die als Dezimalbruch ausgedrückte Zahl und der Teiler eine ganze Zahl, die als Nenner des Dezimalbruchs verwendet wird.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

DOLLARDE(Zahl; Teiler)

##### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Teiler, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

DOLLARDE(1,02; 16) – steht für 1 und 2/16 – ergibt 1,125

##### Ähnliche Funktionen

[DOLLARFR](#)

[TRUNC](#)

#### 8.1.6.14 DOLLARFR

Die Funktion DOLLARFR() gibt eine Notierung, die als Dezimalzahl ausgedrückt wurde, als Dezimalbruch zurück. Die Zahl ist dabei die als Dezimalzahl ausgedrückte Zahl und der Teiler eine ganze Zahl, die als Nenner des Dezimalbruchs verwendet wird.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

DOLLARFR(Zahl; Teiler)

### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Teiler, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

DOLLARFR(1,125; 16) ergibt 1,02. (1 + 2/16)

### Ähnliche Funktionen

DOLLARDE

TRUNC

#### 8.1.6.15 DURATION

Gibt die Zahl der Perioden zurück, die benötigt werden, damit eine Anlage einen gewünschten Wert erreicht.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

DURATION(Zins; PV; FV)

### Parameter

*Kommentar:* Rate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Barwert (PV), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Endwert (FV), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

DURATION(0,1; 1000; 2000) ergibt 7,27

### Ähnliche Funktionen

FV

PV

#### 8.1.6.16 DURATION\_ADD

Gibt die Macauley-Dauer eines Wertpapiers in Jahren zurück.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

DURATION\_ADD(Abrechnung; Fälligkeit; Kupon; Rendite; Häufigkeit; Basis)

### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Kupon, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Rendite, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Berechnungshäufigkeit, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

DURATION\_ADD("01.01.1998"; "01.01.2006"; 0,08; 0,09; 2; 1 ) ergibt 5,9937749555

### Ähnliche Funktionen

MDURATION

#### 8.1.6.17 EFFECT

Die Funktion EFFECT() berechnet die effektive Rendite für einen nominalen Zins (Jahresrate oder effektiver Jahreszins). Beispiel: 8 % Zins jeden Monat ergibt eine effektive Rendite von EFFECT(0,08;12) oder 8,3 %.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

EFFECT(Nominalzins;Perioden)

##### Parameter

*Kommentar:* Nominaler Zinssatz, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Perioden, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

EFFECT(0,08;12) ergibt 0,083

##### Ähnliche Funktionen

[EFFECTIVE](#)  
[NOMINAL](#)

#### 8.1.6.18 EFFECTIVE

Die Funktion EFFECTIVE() berechnet die effektive Rendite für einen nominalen Zins (Jahresrate oder effektiver Jahreszins). Sie ist identisch mit der EFFECT-Funktion.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

EFFECTIVE(Nominalzins;Perioden)

##### Parameter

*Kommentar:* Nominaler Zinssatz, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Perioden, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Ähnliche Funktionen

[EFFECT](#)

#### 8.1.6.19 EURO

Die Funktion EURO() konvertiert einen Euro in die angegebene nationale Währung der Europäischen Währungsunion. Die Währung kann eine der folgenden sein: ATS (Österreich), BEF (Belgien), DEM (Deutschland), ESP (Spanien), EUR (Euro), FIM (Finnland), FRF (Frankreich), GRD (Griechenland), IEP (Irland), ITL (Italien), LUF (Luxemburg), NLG (Niederlande) oder PTE (Portugal).

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

EURO(Währung)

##### Parameter

*Kommentar:* Währung, *Typ:* Text

##### Beispiele

EURO("DEM") ergibt 1,95583

##### Ähnliche Funktionen

[EUROCONVERT](#)

### 8.1.6.20 EUROCONVERT

Die Funktion EUROCONVERT() konvertiert eine nationale Währung in eine andere nationale Währung der Europäischen Währungsunion. Die Umrechnung erfolgt über den Euro als Zwischenschritt. Die Währung kann eine der folgenden sein: ATS (Österreich), BEF (Belgien), DEM (Deutschland), ESP (Spanien), EUR (Euro), FIM (Finnland), FRF (Frankreich), GRD (Griechenland), IEP (Irland), ITL (Italien), LUF (Luxemburg), NLG (Niederlande) oder PTE (Portugal).

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

EUROCONVERT(Anzahl; Ausgangswährung, Zielwährung)

#### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Ausgangswährung, *Typ:* Text

*Kommentar:* Zielwährung, *Typ:* Text

#### Beispiele

EUROCONVERT(1; "EUR"; "DEM") ergibt 1,95583

#### Ähnliche Funktionen

[EURO](#)

### 8.1.6.21 FV

Die Funktion FV() gibt den Endwert einer Anlage zurück, wenn die Rendite und die Laufzeit angegeben werden. Wenn Sie 1000 EUR auf einem Bankkonto mit 8 % Verzinsung haben, besitzen sie nach zwei Jahren FV(1000;0,08;2) oder 1166,40 EUR.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

FV(Barwert;Rendite;Perioden)

#### Parameter

*Kommentar:* Barwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Rate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Perioden, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

FV(1000;0,08;2) ergibt 1166,40

#### Ähnliche Funktionen

[PV](#)

[NPER](#)

[PMT](#)

[RATE](#)

#### 8.1.6.22 FV\_ANNUIITY

Die Funktion FV\_ANNUIITY() gibt den Endwert eines regelmäßigen Geldflusses zurück. Sie erwartet die Höhe der Zahlungen, den Zinssatz und die Anzahl der Perioden. Beispiel: Sie erhalten 500 EUR pro Jahr für 20 Jahre und investieren dieses Geld auf einem Konto mit 8 % Zins. Nach 20 Jahren haben Sie dann insgesamt FV\_annuity(500;0,08;20) oder 22880,98 EUR. Die Funktion geht davon aus, dass die Zahlungen immer am Ende der Periode getätigt werden.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

FV\_ANNUIITY(Betrag; Zins; Perioden)

##### Parameter

*Kommentar:* Zahlung pro Periode, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Zins, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Perioden, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

FV\_ANNUIITY(1000;0,05;5) ergibt 5525,63

#### 8.1.6.23 INTRATE

Die Funktion INTRATE gibt den Zins für ein voll angelegtes Wertpapier zurück. Basis ist die Art der Tageszählung Sie verwenden möchten: 0 für US 30/360 (Standard), 1 für tatsächliche Tage, 2 für tatsächliche Tage/360, 3 für tatsächliche Tage/365 oder 4 für Europäisch 30/365.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

INTRATE(Abrechnung; Fälligkeit; Anlage; Auszahlung; Basis)

##### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Anlage, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Auszahlung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis der Tageszählung, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

INTRATE("2/28/2001"; "8/31/2001"; 1000000; 2000000; 1) ergibt 1,98

#### 8.1.6.24 IPMT

IPMT berechnet den Betrag einer Zinszahlung einer Investition für die Periode.

Zins ist der periodische Zinssatz.

Periode ist die amortisierte Periode. 1 für die erste und NPER für die letzte Periode.

NPER ist die Gesamtzahl der Perioden in welcher die Annuität bezahlt wird.

PV ist der Barwert in einer Sequenz von Zahlungen.

FV (optional) ist der gewünschte (End-) Wert. Standard: 0.

Typ (optional) definiert die Frist. 1 für eine Zahlung am Anfang der Periode und 0 (Standard) für eine Zahlung am Ende der Periode.

Das Beispiel zeigt den Zinssatz, der im letzten Jahr eines Kredits mit einer Laufzeit von drei Jahren bezahlt werden muss. Der Zinssatz beträgt 10 Prozent.

*Rückgabety*p: Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

IPMT(Zins; Periode; NPer; PV; FV; Typ)

### Parameter

*Kommentar*: Rate, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Periode, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Anzahl der Perioden, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Gegenwärtige Werte, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Endwert (optional), *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Typ (optional), *Typ*: Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

IPMT(0,1;3;3;8000) ergibt -292,45

### Ähnliche Funktionen

[PPMT](#)

[PV](#)

[PMT](#)

#### 8.1.6.25 IRR

Die Funktion IRR berechnet den internen Zinsfuß für eine Serie von Rückflüssen.

*Rückgabety*p: Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

IRR( Werte [; Schätzung = 0.1 ] )

### Parameter

*Kommentar*: Werte, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Schätzung, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

### Ähnliche Funktionen

[XIRR](#)

#### 8.1.6.26 ISPMT

Berechnet den Zins, welcher in einer bestimmten Periode einer Anlage bezahlt wurde.

Zins ist der periodische Zinssatz.

Periode ist die amortisierte Periode. 1 für die erste und NPer für die letzte Periode.

NPer ist die Gesamtzahl der Perioden währenddessen Annuität bezahlt wird.

PV ist der Barwert in einer Sequenz von Zahlungen.

*Rückgabety*p: Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

ISPMT(Zins; Periode; NPer; PV)

### Parameter

*Kommentar:* Rate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Periode, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Anzahl der Perioden, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Barwerte (PV), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

ISPMT(0,1; 1; 3; 8000000) ergibt -533333

### Ähnliche Funktionen

PV

FV

NPER

PMT

RATE

#### 8.1.6.27 LEVEL\_COUPON

Die Funktion LEVEL\_COUPON() berechnet den Wert einer Kuponanleihe. Beispiel: Der Zinssatz beträgt 10%. Eine 1000 EUR Anleihe mit halbjährlichen Kupons bei einem Zinssatz von 13 %, der sich 4 Jahre lang entwickelt, hat einen Wert von LEVEL\_COUPON(1000;.13;2;4;0,1) oder 1096,95 EUR.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

LEVEL\_COUPON(Nennwert;Kuponrate;Kupons pro Jahr;Jahre;Marktzinssatz)

### Parameter

*Kommentar:* Nennwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Kuponrate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Kupons pro Jahr, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Jahre, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Marktzinssatz, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

LEVEL\_COUPON(1000;0,13;2;4;0,1) ergibt 1096,95

#### 8.1.6.28 MDURATION

Die Funktion MDURATION() berechnet die modifizierte Macauley-Dauer eines Wertpapiers in Jahren.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

MDURATION( Abrechnung; Fälligkeit; Kupon; Rendite; Häufigkeit; [ Basis=0 ])

### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fälligkeitsdatum, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Kupon, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Rendite, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Berechnungshäufigkeit, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

MDURATION("01.02.2004"; "31.05.2004"; 0,08; 0,09; 2; 0) ergibt 0,316321106

### Ähnliche Funktionen

[DURATION](#)

#### 8.1.6.29 MIRR

Die Funktion MIRR() berechnet den internen Zinsfuß (IRR) einer Serie von periodischen Anlagen.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

MIRR(Werte; Anlage; Reinvestition)

### Parameter

*Kommentar:* Werte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Anlage, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Reinvestition, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

MIRR({100;200;-50;300;-200}, 5 %, 6 %) ergibt 34,2823387842 %

### Ähnliche Funktionen

[IRR](#)

#### 8.1.6.30 NOMINAL

Die Funktion NOMINAL() berechnet den Nominalzins eines Effektivzinssatzes, der in gegebenen Zeitabständen berechnet wird. Beispiel: Um 8 % Zins auf einem Konto zu erhalten, das monatlich verzinst wird, müssen sie NOMINAL(0,08;12) oder 7,72 % Nominalzins bekommen.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

NOMINAL(Effektiver Zins;Perioden)

### Parameter

*Kommentar:* Effektiver Zins, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Perioden, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

NOMINAL(0,08;12) ergibt 0,0772

### Ähnliche Funktionen

[EFFECT](#)

### 8.1.6.31 NPER

Gibt die Anzahl der Perioden einer Anlage zurück.

*Rückgabetyyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

NPER(Zins;Zahlung;PV;FV;Typ)

#### Parameter

*Kommentar:* Rate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Zahlung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Barwert (PV), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Endwert (FV – optional), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Typ (optional), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

NPER(0,1; -100; 1000) ergibt 11

#### Beispiele

NPER(0,06; 0; -10000; 20000 ;0) ergibt 11,906

#### Ähnliche Funktionen

FV  
RATE  
PMT  
PV

### 8.1.6.32 NPV

Der Kapitalwert (NPV) für eine Serie von periodischen Rückflüssen.

Berechnet den Kapitalwert (NPV) für eine Serie von periodischen Rückflüssen mit dem Abzinsfaktor „Zins“. Die Werte sollten positiv sein, wenn sie als Einnahme und negativ, wenn sie als Ausgabe verbucht werden.

*Rückgabetyyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

NPV(Zins; Werte)

#### Parameter

*Kommentar:* Rate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Werte (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

NPV(100%;4;5;7) = 4,125

#### Ähnliche Funktionen

FV  
IRR  
NPER  
PMT  
PV

### 8.1.6.33 ODDLPRICE

Die Funktion ODDLPRICE berechnet den Wert des Wertpapiers pro 100 Währungseinheiten des Nennwerts. Das Wertpapier hat dabei einen irregulären letzten Zinstermin.

*Rückgabetyyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ODDLPRICE( Abrechnung; Fälligkeit; Letzter Zinstermin; Zins; Jährliche Rendite; Auszahlung; Häufigkeit [; Basis = 0 ] )

#### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Letzte, *Typ:* Datum

*Kommentar:* Rate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Jährliche Rendite, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Auszahlung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Berechnungshäufigkeit, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

ODDLPRICE(DATE(1990;6;1);DATE(1995;12;31);DATE(1990;1;1);3 %;5 %;100;2) ergibt 90,991042345

### 8.1.6.34 ODDLYIELD

Die Funktion ODDLYIELD berechnet die Rendite eines Wertpapiers mit einem irregulären letzten Zinstermin.

*Rückgabetyyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ODDLYIELD( Abrechnung; Fälligkeit; Letzter Zinstermin; Zins; Kurs; Auszahlung; Häufigkeit [; Basis = 0 ] )

#### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Letzte, *Typ:* Datum

*Kommentar:* Rate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Kurs, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Auszahlung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Berechnungshäufigkeit, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

ODDLYIELD(DATE(1990;6;1);DATE(1995;12;31);DATE(1990;1;1);3 %;91;100;2) ergibt 4,997775351

#### Ähnliche Funktionen

[ODDLPRICE](#)

### 8.1.6.35 PMT

PMT ergibt den Zahlungsbetrag für einen Kredit mit konstantem Zinssatz und konstanten Zahlungen (jede Zahlung hat den gleichen Betrag).

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

PMT(Zins; NPer ; PV [; FV = 0 [; Typ = 0 ] ] )

#### Parameter

*Kommentar:* Rate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Anzahl der Perioden (NPer), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Barwert (PV), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Endwert (FV – optional), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Typ (optional), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

PMT(0,1; 4; 10000) ergibt -3154,71

#### Ähnliche Funktionen

[NPER](#)

[IPMT](#)

[PPMT](#)

[PV](#)

### 8.1.6.36 PPMT

PPMT berechnet den Betrag der Kapitalrückzahlung einer Investition für die Periode.

Zins ist der periodische Zinssatz.

Periode ist die amortisierte Periode. 1 für die erste und NPER für die letzte Periode.

NPER ist die Gesamtzahl der Perioden in welcher die Annuität bezahlt wird.

PV ist der Barwert in einer Sequenz von Zahlungen.

FV (optional) ist der gewünschte (End-) Wert. Standard: 0.

Typ (optional) definiert die Frist. 1 für eine Zahlung am Anfang der Periode und 0 (Standard) für eine Zahlung am Ende der Periode.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

PPMT(Zins; Periode; NPer; PV [; FV = 9 [; Typ = 0 ] ] )

#### Parameter

*Kommentar:* Rate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Periode, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Anzahl der Perioden, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Barwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Endwert (optional), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Typ (optional), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

PPMT(0,0875;1;36;5000;8000;1) ergibt -18,48

#### Ähnliche Funktionen

[IPMT](#)

[PMT](#)

[PV](#)

### 8.1.6.37 PRICEMAT

Die Funktion PRICEMAT berechnet den Kurs pro 100 Währungseinheiten des Nennwerts eines Wertpapiers, das Zinsen am Fälligkeitsdatum auszahlt.

Basis-Berechnungsmodus

0 US-Methode, 12 Monate, jeder Monat hat 30 Tage

1 Tatsächliche Anzahl an Tagen im Jahr, tatsächliche Anzahl an Tagen pro Monat

2 360 Tage in einem Jahr, tatsächliche Anzahl an Tagen pro Monat

4 365 Tage in einem Jahr, tatsächliche Anzahl an Tagen pro Monat

5 Europäische Methode, 12 Monate, jeder Monat hat 30 Tage

*Rückgabety*p: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

PRICEMAT(Abrechnung; Fälligkeit; Ausgabe; Zins; Rendite [; basis = 0 ] )

#### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Ausgabetag, *Typ:* Datum

*Kommentar:* Disagio, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Rendite, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

PRICEMAT(DATE(1990;6;1);DATE(1995;12;31);DATE(1990;1;1);6 %;5 %) ergibt  
103,819218241

### 8.1.6.38 PV

Die Funktion PV() gibt den Barwert einer Anlage zurück. Das ist der heutige Wert einer Geldsumme in der Zukunft bei einem bestimmten Zinssatz bzw. einer bestimmten Inflationsrate. Wenn Sie z. B. 1166,40 EUR für einen neuen Computer benötigen, welchen Sie in zwei Jahren kaufen möchten, benötigen Sie bei einem Zinssatz von 8 % jetzt PV(1166,4;0,08;2) oder 1000 EUR.

*Rückgabety*p: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

PV(Endwert;Zins;Perioden)

#### Parameter

*Kommentar:* Endwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Zins, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Perioden, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

PV(1166,4; 0,08; 2) ergibt 1000

### 8.1.6.39 PV\_ANNUIITY

Die Funktion PV\_ANNUIITY() gibt den Barwert einer jährlichen Zahlung oder eines Geldstromes. Beispiel: Ein „eine Million“-Lottozettel welcher Ihnen 50000 EUR pro Jahr für 20 Jahre auszahlt, ist bei einem Zinssatz von 5 % aktuell PV\_ANNUIITY(50000;0,05;20) oder 623,111 EUR wert. Diese Funktion geht davon aus, dass die Zahlungen am Ende jeder Periode getätigt werden.

*Rückgabetyyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

PV\_ANNUIITY(Betrag; Zins; Perioden)

#### Parameter

*Kommentar:* Zahlung pro Periode, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Zins, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Perioden, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

PV\_ANNUIITY(1000;0,05;5) ergibt 4329,48

### 8.1.6.40 RATE

Die Funktion RATE() berechnet die konstante Zinsrate der Periode einer Investition.

*Rückgabetyyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

RATE(nper;pmt;pv;fv;type;guess)

#### Parameter

*Kommentar:* Zahlungsperiode, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Regelmäßige Zahlungen, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Barwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Endwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Typ, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Schätzung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

RATE(4\*12;-200;8000) ergibt 0,007701472

### 8.1.6.41 RECEIVED

Die Funktion RECEIVED gibt den Betrag zurück, der am Fälligkeitsdatum für eine Wertpapier ausbezahlt wird. Basis ist die Art der Tageszählung Sie verwenden möchten: 0 für US 30/360 (Standard), 1 für tatsächliche Tage, 2 für tatsächliche Tage/360, 3 für tatsächliche Tage/365 oder 4 für Europäisch 30/365. Das Abrechnungsdatum muss vor dem Fälligkeitsdatum liegen.

*Rückgabetyyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

RECEIVED(Abrechnung; Fälligkeit; Anlage; Disagio; Basis)

### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Anlage, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Disagio, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

RECEIVED("2/28/2001"; "8/31/2001"; 1000; 0,05; 0) ergibt 1.025,787

### 8.1.6.42 RRI

Die Funktion RRI berechnet die aus einem Profit einer Investition resultierende Zinsrate.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

RRI( P; Pv; Fv)

### Parameter

*Kommentar:* P, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Pv, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fv, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

RRI(1;100;200) ergibt 1

### Ähnliche Funktionen

[FV](#)  
[NPER](#)  
[PMT](#)  
[PV](#)  
[RATE](#)

### 8.1.6.43 SLN

Die Funktion SLN() berechnet die lineare Abschreibung eines Anlagegegenstandes über eine Abschreibungsperiode. „Kosten“ ist der Betrag, der für die Anlage bezahlt wurde. „Restwert“ ist der Wert der Anlage am Ende des Abschreibungszeitraumes. „Lebensdauer“ ist die Anzahl der Perioden, in denen die Anlage abgeschrieben wird. SLN() verteilt die Kosten gleich hoch über die Lebensdauer einer Anlage.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

SLN(Kosten; Restwert; Lebensdauer)

### Parameter

*Kommentar:* Kosten, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Restwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Lebensdauer, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

SLN(10000;700;10) ergibt 930

### Ähnliche Funktionen

[SYD](#)  
[DDB](#)

#### 8.1.6.44 SYD

Die Funktion SYD() berechnet die „sum-of-years digits“-Abschreibung für eine Anlage abhängig von ihren Kosten, dem Restwert, der erwarteten Lebensdauer und der gewählten Abschreibungsperiode. Diese Methode beschleunigt die Rate der Abschreibung, sodass in den früheren Abschreibungsperioden höhere Abschreibungskosten auftreten als in den späteren Perioden. Die abschreibbaren Kosten sind die Anschaffungskosten minus Restwert. Die Nutzungsdauer ist die Anzahl der Perioden (typischerweise Jahre), über die die Anlage abgeschrieben wird.

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SYD(Kosten; Restwert; Lebensdauer; Periode)

##### Parameter

*Kommentar*: Kosten, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Restwert, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Lebensdauer, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Periode, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

SYD(5000; 200; 5; 2) ergibt 1280

##### Ähnliche Funktionen

SLN  
DDB

#### 8.1.6.45 TBILLEQ

Die Funktion TBILLEQ gibt die passende Obligation zu einem Schatzschein zurück. Das Fälligkeitsdatum muss nach dem Abrechnungsdatum, aber innerhalb von 365 Tagen liegen.

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

TBILLEQ(Abrechnung; Fälligkeit; Disagio)

##### Parameter

*Kommentar*: Abrechnungsdatum, *Typ*: Datum

*Kommentar*: Fälligkeitsdatum, *Typ*: Datum

*Kommentar*: Disagio, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

TBILLEQ("2/28/2001"; "8/31/2001"; 0,1) ergibt 0,1068

##### Ähnliche Funktionen

TBILLPRICE  
TBILLYIELD

#### 8.1.6.46 TBILLPRICE

Die Funktion TBILLPRICE gibt den Preis pro \$100 Wert für einen Schatzschein zurück. Das Fälligkeitsdatum muss nach dem Abrechnungsdatum, aber innerhalb von 365 Tagen liegen. Der Disagio muss positiv sein.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

TBILLPRICE(Abrechnung; Fälligkeit; Disagio)

##### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Disagio, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

TBILLPRICE("2/28/2001"; "8/31/2001"; 0,05) ergibt 97,4444

##### Ähnliche Funktionen

[TBILLEQ](#)

[TBILLYIELD](#)

#### 8.1.6.47 TBILLYIELD

Die Funktion TBILLYIELD gibt die Rendite für einen Schatzschein zurück. Das Fälligkeitsdatum muss nach dem Abrechnungsdatum, aber innerhalb von 365 Tagen liegen. Der Preis muss positiv sein.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

TBILLYIELD(Abrechnung; Fälligkeit; Kurs)

##### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Kurs pro \$100 Nennwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

TBILLYIELD("2/28/2001"; "8/31/2001"; 600) ergibt -1,63

##### Ähnliche Funktionen

[TBILLEQ](#)

[TBILLPRICE](#)

#### 8.1.6.48 VDB

VDB berechnet den Abschreibungsbetrag einer Anlage aus einem Anschaffungswert, einer erwarteten Nutzungsdauer, einem Restwert für den angegebenen Zeitraum mit der degressiven Abschreibungsmethode.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

VDB(Kosten; Restwert; Lebensdauer; Startperiode; Endperiode; [; Abschreibungsfaktor = 2  
[; wechseln=falsch]] )

### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Kurs, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Auszahlung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

VDB(10000;600;10;0;0,875;1,5) ergibt 1312,5

#### 8.1.6.49 XIRR

Die Funktion XIRR berechnet den internen Zinsfuß für eine nicht-periodische Serie von Rückflüssen.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

XIRR( Werte; Daten[; Schätzung = 0.1 ] )

### Parameter

*Kommentar:* Werte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Daten, *Typ:* Datum

*Kommentar:* Schätzung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

XIRR(B1:B4;C1:C4) ergibt 0.2115964, wenn B1:B4 „-20000“, „4000“, „12000“, „8000“ und C1:C4 „=DATE(2000;1;1)“, „=DATE(2000;6;1)“, „=DATE(2000;12;30)“, „=DATE(2001;3;1)“ enthält.

### Ähnliche Funktionen

[IRR](#)

#### 8.1.6.50 XNPV

Die Funktion XNPV berechnet den Kapitalwert für eine Serie von Rückflüssen.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

XNPV(Zins; Werte; Daten)

### Parameter

*Kommentar:* Rate, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Werte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Daten, *Typ:* Datum

### Beispiele

XNPV(5 %;B1:B4;C1:C4) ergibt 2907,83187, wenn B1:B4 „-20000“, „4000“, „12000“, „8000“ und C1:C4 „=DATE(2000;1;1)“, „=DATE(2000;6;1)“, „=DATE(2000;12;30)“, „=DATE(2001;3;1)“ enthält.

### Ähnliche Funktionen

[NPV](#)

### 8.1.6.51 YIELDDISC

Die Funktion YIELDDISC berechnet die Rendite eines Wertpapiers pro 100 Währungseinheiten des Nennwerts.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

YIELDDISC(Abrechnung; Fälligkeit; Kurs, Auszahlung, Basis)

#### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Kurs, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Auszahlung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

YIELDDISC(DATE(1990;6;1);DATE(1990;12;31);941,66667;1000) ergibt 0,106194684

### 8.1.6.52 YIELDMAT

Die Funktion YIELDMAT berechnet die Rendite eines Wertpapiers, dessen Zinsen am Fälligkeits-termin gezahlt werden.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

YIELDMAT( Abrechnung; Fälligkeit; Ausgabe; Zins; Kurs; Basis )

#### Parameter

*Kommentar:* Abrechnungsdatum, *Typ:* Datum

*Kommentar:* Fälligkeitsdatum, *Typ:* Datum

*Kommentar:* Ausgabetag, *Typ:* Datum

*Kommentar:* Disagio, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Kurs, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

YIELDMAT(DATE(1990;6;1);DATE(1995;12;31);DATE(1990;1;1);6 %;103,819218241) ergibt 0,050000000.

#### Ähnliche Funktionen

[YIELDDISC](#)

### 8.1.6.53 ZERO\_COUPON

Die Funktion ZERO\_COUPON() berechnet den Wert einer Nullkuponanleihe. Beispiel: Sie haben 10% Zins. Ein 1000 EUR Anleihe, die sich 20 Jahre lang entwickelt, hat einen Wert von ZERO\_COUPON(1000;0,1;20) oder 148,64 EUR.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ZERO\_COUPON(Nennwert;Zins;Jahre)

### Parameter

*Kommentar:* Nennwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Zins, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Jahre, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

ZERO\_COUPON(1000;0,1;20) ergibt 148,64

## 8.1.7 Information

### 8.1.7.1 ERRORTYPE

Die Funktion ERRORTYPE() konvertiert einen Fehler in eine Zahl. Wenn der Wert kein Fehler ist, wird ein Fehler ausgegeben. Anderenfalls wird ein numerischer Fehlerwert ausgegeben. Die Fehlerwerte entsprechen denen von Microsoft Excel.

*Rückgabetyp:* Ganze Zahl (z. B. 1, 132, 2344)

### Syntax

ERRORTYPE(Wert)

### Parameter

*Kommentar:* Fehler, *Typ:* Ein beliebiger Wert

### Beispiele

ERRORTYPE(NA()) ergibt 7

### Beispiele

ERRORTYPE(0) gibt einen Fehler zurück

### 8.1.7.2 FILENAME

Ergibt den aktuellen Dateinamen. Wenn das aktuelle Dokument noch nicht gespeichert wurde, wird eine leere Zeichenfolge zurückgegeben.

*Rückgabetyp:* Text

### Syntax

FILENAME()

### Parameter

### 8.1.7.3 FORMULA

Die Funktion FORMULA() ergibt die Formel einer Zelle als Zeichenfolge.

*Rückgabetyp:* Text

### Syntax

FORMULA(x)

### Parameter

*Kommentar:* Verweis, *Typ:* Verweis

### Beispiele

FORMULA(A1) gibt "=SUM(1+2)" zurück wenn die Zelle A1 diese Formel enthält.

#### 8.1.7.4 INFO

Die Funktion INFO() gibt Informationen über die aktuelle Arbeitsumgebung zurück. Der Parameter „Typ“ gibt an, welche Art von Informationen zurückgegeben werden soll. Er kann einer der folgenden Werte sein: „directory“ gibt den Pfad des aktuellen Ordners zurück, „numfile“ ergibt die Anzahl aktuell geöffneter Dokumente, „release“ ergibt die Version von Calligra Tables als Text, „recalc“ ergibt den aktuellen Neuberechnungsmodus: „Automatisch“ oder „Manuell“, „system“ ergibt den Namen des Betriebssystems und „osversion“ ergibt den Namen und die Version des Betriebssystems.

*Rückgabotyp:* Text

##### Syntax

INFO(Typ)

##### Parameter

*Kommentar:* Typ der Information, *Typ:* Text

#### 8.1.7.5 ISBLANK

Die Funktion ISBLANK() ergibt Wahr, wenn der Parameter leer ist. Ansonsten ergibt sie Falsch.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

ISBLANK(x)

##### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

##### Beispiele

ISBLANK(A1) ergibt Wahr, wenn A1 leer ist

##### Beispiele

ISBLANK(A1) ergibt Falsch, wenn A1 einen Wert enthält

#### 8.1.7.6 ISDATE

Die Funktion ISDATE() ergibt Wahr, wenn der Parameter ein Datumswert ist. Sonst ergibt sie Falsch.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

ISDATE(x)

##### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

##### Beispiele

ISDATE("2.2.2000") ergibt Wahr

##### Beispiele

ISDATE("Hallo") ergibt Falsch

#### 8.1.7.7 ISERR

Die Funktion ISERR() gibt Wahr zurück, wenn der Parameter ein anderer Fehler als „N/A“ ist. Ansonsten gibt sie Falsch zurück. Wenn „N/A“ ebenfalls behandelt werden soll, müssen Sie die Funktion ISERROR() benutzen.

*Rückgabebetyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

ISERR(x)

##### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

##### Ähnliche Funktionen

[ISERROR](#)  
[ISNA](#)

#### 8.1.7.8 ISERROR

Die Funktion ISERROR() ergibt Wahr, wenn der Parameter ein Fehler einer beliebigen Art ist. Sonst ergibt sie Falsch.

*Rückgabebetyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

ISERROR(x)

##### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

##### Ähnliche Funktionen

[ISERR](#)  
[ISNA](#)

#### 8.1.7.9 ISEVEN

Die Funktion ISEVEN() ergibt Wahr, wenn die angegebene Zahl gerade ist. Sonst ergibt sie Falsch.

*Rückgabebetyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

ISEVEN(x)

##### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

##### Beispiele

ISEVEN(12) ergibt Wahr

##### Beispiele

ISEVEN(-7) ergibt Falsch

#### 8.1.7.10 ISFORMULA

Die Funktion ISFORMULA() ergibt Wahr, wenn der Zellenverweis eine Formel enthält. Ansonsten ergibt sie Falsch.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

ISFORMULA(x)

##### Parameter

*Kommentar:* Verweis, *Typ:* Verweis

#### 8.1.7.11 ISLOGICAL

Die Funktion ISLOGICAL() ergibt Wahr, wenn der Parameter ein boolescher Wert ist. Ansonsten ergibt die Funktion Falsch.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

ISLOGICAL(x)

##### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

##### Beispiele

ISLOGICAL(A1 >A2) ergibt Wahr

##### Beispiele

ISLOGICAL(12) ergibt Falsch

#### 8.1.7.12 ISNA

Die Funktion ISNA() ergibt Wahr, wenn der Parameter ein „N/A“-Fehler ist. Sonst ergibt sie Falsch.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

ISNA(x)

##### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

##### Ähnliche Funktionen

[ISERR](#)

[ISERROR](#)

### 8.1.7.13 ISNONTEXT

Die Funktion ISNONTEXT() ergibt Wahr wenn der Parameter keine Zeichenfolge ist. Ansonsten ergibt sie Falsch. Sie ist identisch mit der ISNOTTEXT-Funktion.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

#### Syntax

ISNONTEXT(x)

#### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

#### Beispiele

ISNONTEXT(12) ergibt Wahr

#### Beispiele

ISNONTEXT("Hallo") ergibt Falsch

#### Ähnliche Funktionen

[ISNOTTEXT](#)

### 8.1.7.14 ISNOTTEXT

Die Funktion ISNOTTEXT() ergibt Wahr wenn der Parameter keine Zeichenfolge ist. Ansonsten ergibt sie Falsch. Sie ist identisch mit der ISNONTEXT-Funktion.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

#### Syntax

ISNOTTEXT(x)

#### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

#### Beispiele

ISNOTTEXT(12) ergibt Wahr

#### Beispiele

ISNOTTEXT("Hallo") ergibt Falsch

#### Ähnliche Funktionen

[ISNONTEXT](#)

### 8.1.7.15 ISNUM

Die Funktion ISNUM() gibt Wahr zurück, wenn der Parameter ein numerischer Wert ist. Ansonsten gibt sie Falsch zurück. Sie ist identisch mit der ISNUMBER-Funktion.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

#### Syntax

ISNUM(x)

**Parameter**

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

**Beispiele**

ISNUM(12) ergibt Wahr

**Beispiele**

ISNUM(Hallo) ergibt Falsch

**Ähnliche Funktionen**

[ISNUMBER](#)

**8.1.7.16 ISNUMBER**

Die Funktion ISNUMBER() gibt Wahr zurück, wenn der Parameter ein numerischer Wert ist. Ansonsten gibt sie Falsch zurück. Sie ist identisch mit der ISNUM-Funktion.

*Rückgabetyt:* Wahrheitswert (WAHR oder FALSCH)

**Syntax**

ISNUMBER(x)

**Parameter**

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

**Beispiele**

ISNUMBER(12) ergibt Wahr

**Beispiele**

ISNUMBER(Hallo) ergibt Falsch

**Ähnliche Funktionen**

[ISNUM](#)

**8.1.7.17 ISODD**

Die Funktion ISODD() ergibt Wahr, wenn die angegebene Zahl ungerade ist. Sonst ergibt sie Falsch.

*Rückgabetyt:* Wahrheitswert (WAHR oder FALSCH)

**Syntax**

ISODD(x)

**Parameter**

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

**Beispiele**

ISODD(12) ergibt Falsch

**Beispiele**

ISODD(-7) ergibt Wahr

#### 8.1.7.18 ISREF

Die Funktion ISREF() ergibt Wahr, wenn der Parameter ein Zellenverweis ist. Ansonsten ergibt sie Falsch.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

ISREF(x)

##### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

##### Beispiele

ISREF(A12) ergibt Wahr

##### Beispiele

ISREF("Hallo") ergibt Falsch

#### 8.1.7.19 ISTEEXT

Die Funktion ISTEEXT() ergibt Wahr, wenn der Parameter eine Zeichenfolge ist. Ansonsten ergibt sie Falsch.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

ISTEXT(x)

##### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

##### Beispiele

ISTEXT(12) ergibt Falsch

##### Beispiele

ISTEXT("Hallo") ergibt Wahr

#### 8.1.7.20 ISTITUTE

Die Funktion ISTITUTE() ergibt Wahr, wenn der Parameter ein Zeitwert ist. Sonst ergibt sie Falsch.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

ISTIME(x)

##### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

##### Beispiele

ISTIME("12:05") ergibt Wahr

##### Beispiele

ISTIME("Hallo") ergibt Falsch

#### 8.1.7.21 N

Die Funktion N() konvertiert einen Wert zu einer Zahl. Wenn der Wert eine Zahl ist oder auf eine Zahl verweist, gibt diese Funktion die Zahl wieder. Wenn der Wert Wahr, also wahr ist, ergibt die Funktion 1. Wenn der Wert ein Datum ist, gibt diese Funktion die Seriennummer des Datums zurück. Bei jedem anderen Wert gibt die Funktion 0 zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

N(Wert)

##### Parameter

*Kommentar:* Wert, *Typ:* Ein beliebiger Wert

##### Beispiele

N(3,14) ergibt 3,14

##### Beispiele

N("7") ergibt 0 (weil "7" Text ist)

#### 8.1.7.22 NA

Die Funktion NA() ergibt den konstanten Fehlerwert „N/A“.

*Rückgabotyp:* Fehler

##### Syntax

NA()

##### Parameter

##### Ähnliche Funktionen

[ISNA](#)  
[ISERR](#)  
[ISERROR](#)

#### 8.1.7.23 TYPE

Die Funktion TYPE() ergibt 1, wenn der Wert eine Zahl ist. Sie ergibt 2, wenn der Wert ein Text ist, sie ergibt 4, wenn er ein boolescher Wert ist, sie ergibt 16, wenn er ein Fehlerwert ist, und sie ergibt 64, wenn der Wert ein Feld ist. Wenn die Zelle, auf die der Wert zeigt, eine Formel enthält, ergibt die Funktion deren Rückgabotyp.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

TYPE(x)

##### Parameter

*Kommentar:* Irgendein Wert, *Typ:* Ein beliebiger Wert

##### Beispiele

TYPE(A1) ergibt 2, wenn A1 „Text“ enthält.

##### Beispiele

TYPE(-7) ergibt 1

##### Beispiele

TYPE(A2) ergibt 1, wenn A2 „=CURRENTDATE()“ enthält

## 8.1.8 Logik

### 8.1.8.1 AND

Die Funktion AND() ergibt Wahr, wenn alle angegebenen Parameter Wahr sind. Ansonsten ergibt sie Falsch (sofern nicht einer der Werte fehlerhaft ist – dann wird ein Fehler zurückgegeben)

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

#### Syntax

AND(Wert;Wert;...)

#### Parameter

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

#### Beispiele

AND(Wahr;Wahr;Wahr) ergibt Wahr

#### Beispiele

AND(Wahr;Falsch) ergibt Falsch

### 8.1.8.2 FALSE

Die Funktion Falsch() ergibt den booleschen Wert Falsch.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

#### Syntax

Falsch()

#### Parameter

#### Beispiele

Falsch() ergibt Falsch

### 8.1.8.3 IF

Die IF() Funktion ist eine Bedingungs-Funktion. Die Funktion ergibt den zweiten Parameter, wenn die Bedingung Wahr ist, ansonsten ergibt sie den dritten Parameter.

*Rückgabotyp:* Ein beliebiger Wert

#### Syntax

IF(Bedingung;Wenn\_Wahr;Wenn\_Falsch)

#### Parameter

*Kommentar:* Bedingung, *Typ:* Wahrheitswert (WAHR oder FALSCH)

*Kommentar:* If Wahr, *Typ:* Ein beliebiger Wert

*Kommentar:* If Falsch, *Typ:* Ein beliebiger Wert

#### Beispiele

A1=4;A2=6;IF(A1 >A2;5;3) ergibt 3

#### 8.1.8.4 IFERROR

Gibt X zurück, wenn es kein Fehlerwert ist. Ansonsten wird der alternative Wert zurückgegeben.

*Rückgabotyp:* Ein beliebiger Wert

##### Syntax

IFERROR(BeliebigesX;BeliebigeAlternative)

##### Parameter

*Kommentar:* Beliebiges X , *Typ:* Ein beliebiger Wert

*Kommentar:* Beliebige Alternative, *Typ:* Ein beliebiger Wert

##### Beispiele

IFERROR(A1;A2) gibt den Inhalt von A1 zurück, wenn dieser Inhalt kein Fehlerwert ist. Ansonsten wird der Inhalt von A2 zurückgegeben.

#### 8.1.8.5 IFNA

Gibt X zurück, wenn es kein „N/A“-Wert (Nicht verfügbar) ist. Ansonsten wird der alternative Wert zurückgegeben.

*Rückgabotyp:* Ein beliebiger Wert

##### Syntax

IFNA(BeliebigesX;BeliebigeAlternative)

##### Parameter

*Kommentar:* Beliebiges X , *Typ:* Ein beliebiger Wert

*Kommentar:* Beliebige Alternative, *Typ:* Ein beliebiger Wert

##### Beispiele

IFNA(A1;A2) gibt den Inhalt von A1 zurück, wenn dieser Inhalt kein „N/A“-Wert ist. Ansonsten wird der Inhalt von A2 zurückgegeben.

#### 8.1.8.6 NAND

Die Funktion NAND() ergibt Wahr, wenn mindestens ein Parameter Falsch ist. Sind alle Parameter Wahr, ergibt sie Falsch.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

##### Syntax

NAND(Wert;Wert;...)

##### Parameter

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

##### Beispiele

NAND(Wahr;Falsch;Falsch) ergibt Wahr

##### Beispiele

NAND(Wahr;Wahr) ergibt Falsch

### 8.1.8.7 NOR

Die Funktion NOR() ergibt Wahr, wenn alle Parameter boolesch sind und den Wert Falsch haben. Ansonsten ergibt sie Falsch.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

#### Syntax

NOR(Wert;Wert;...)

#### Parameter

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

#### Beispiele

NOR(Wahr;Falsch;Falsch) ergibt Falsch

#### Beispiele

NOR(Falsch;Falsch) ergibt Wahr

### 8.1.8.8 NOT

Die Funktion NOT() ergibt Wahr, wenn der Parameter den Wert Falsch hat und ergibt Falsch, wenn der Parameter den Wert Wahr hat.

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

#### Syntax

NOT(Bool)

#### Parameter

*Kommentar:* Boolescher Wert, *Typ:* Wahrheitswert (WAHR oder FALSCH)

#### Beispiele

NOT(Falsch) ergibt Wahr

#### Beispiele

NOT(Wahr) ergibt Falsch

### 8.1.8.9 OR

Die Funktion OR() ergibt Wahr, wenn mindestens ein Parameter Wahr ist. Sind alle Parameter Falsch, ergibt sie Falsch (sofern nicht einer der Werte fehlerhaft ist – dann wird ein Fehler zurückgegeben)

*Rückgabotyp:* Wahrheitswert (WAHR oder FALSCH)

#### Syntax

OR(Wert;Wert;...)

**Parameter**

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)  
*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)  
*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)  
*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)  
*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

**Beispiele**

OR(Falsch;Falsch;Falsch) ergibt Falsch

**Beispiele**

OR(Wahr;Falsch) ergibt Wahr

**8.1.8.10 TRUE**

Die Funktion Wahr() ergibt den booleschen Wert Wahr.

*Rückgabetyyp:* Wahrheitswert (WAHR oder FALSCH)

**Syntax**

Wahr()

**Parameter**

**Beispiele**

Wahr() ergibt Wahr

**8.1.8.11 XOR**

Die Funktion XOR() ergibt Falsch, wenn die Anzahl der Wahr-Werte gerade ist. Anderenfalls ergibt sie Wahr. Wenn eines der Argumente fehlerhaft ist, wird ein Fehler zurückgegeben.

*Rückgabetyyp:* Wahrheitswert (WAHR oder FALSCH)

**Syntax**

XOR(Wert;Wert;...)

**Parameter**

*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)  
*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)  
*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)  
*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)  
*Kommentar:* Boolesche Werte, *Typ:* Eine Folge von Wahrheitswerten (WAHR oder FALSCH)

**Beispiele**

XOR(Falsch;Falsch;Falsch) ergibt Wahr

**Beispiele**

XOR(Wahr;Falsch) ergibt Wahr

## 8.1.9 Nachschlagen & Referenz

### 8.1.9.1 ADDRESS

Die Funktion ADDRESS erzeugt eine Zelladresse. Der Parameter Zeile ist die Zeilennummer und Spalte die Spaltennummer.

Absolut gibt den Typ der Referenz an: 1 oder weggelassen = Absolut, 2 = Absolute Zeile, relative Spalte, 3 = Relative Zeile, absolute Spalte und 4 = Relativ.

A1 Stil bezeichnet den Stil der Adresse, welche zurückgegeben wird. Wenn A1 auf Wahr gesetzt ist (Standard) dann ist die zurückgegebene Adresse im A1 Stil. Wenn A1 auf Falsch gesetzt ist, dann ist die zurückgegebene Adresse im R1C1 Stil.

Tabellenname ist der Text, welcher den Namen der Tabelle bezeichnet.

*Rückgabebetyp:* Text

#### Syntax

ADDRESS(Zeile; Spalte; Absolut; Stil; Tabellenname)

#### Parameter

*Kommentar:* Zeilennummer, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Spaltennummer, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Absolute Zahl (optional), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* A1-Stil (optional), *Typ:* Wahrheitswert (WAHR oder FALSCH)

*Kommentar:* Tabellenname, *Typ:* Text

#### Beispiele

ADDRESS(6; 4) ergibt \$D\$6

#### Beispiele

ADDRESS(6; 4; 2) ergibt D\$6

#### Beispiele

ADDRESS(6; 4; 2; Falsch; "Tabelle1") ergibt Tabelle1!R6C[4]

#### Beispiele

ADDRESS(6; 4; 1; Falsch; "Tabelle1") ergibt Tabelle1!R6C4

#### Beispiele

ADDRESS(6; 4; 4; Wahr; "Tabelle1") ergibt Tabelle1!D6

### 8.1.9.2 AREAS

Gibt die Anzahl der Bereiche in dem angegebenen Zellenverweis zurück. Ein Bereich kann eine einzelne Zelle oder ein Satz von Zellen sein.

*Rückgabebetyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

AREAS(Zellenverweis)

#### Parameter

*Kommentar:* Verweis, *Typ:* Eine Zeichenfolge

#### Beispiele

AREAS(A1) ergibt 1

#### Beispiele

AREAS((A1; A2:A4)) ergibt 2

### 8.1.9.3 CELL

Gibt Informationen über die Position, Formatierung oder Inhalt eines Zellenverweises zurück.

*Rückgabotyp:* Ein beliebiger Wert

#### Syntax

CELL(Typ; Zellenverweis)

#### Parameter

*Kommentar:* Typ, *Typ:* Text

*Kommentar:* Verweis, *Typ:* Verweis

#### Beispiele

CELL("COL", C7) ergibt 3

#### Beispiele

CELL("ROW", C7) ergibt 7

#### Beispiele

CELL("ADDRESS", C7) ergibt \$C\$7

### 8.1.9.4 CHOOSE

Gibt den Parameter zurück, der durch den Index angegeben wird.

*Rückgabotyp:* Ein beliebiger Wert

#### Syntax

CHOOSE(Index; Parameter1; Parameter2;...)

#### Parameter

*Kommentar:* Index, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Argumente, *Typ:*

#### Beispiele

CHOOSE(1; "erster"; "zweiter") ergibt "erster"

#### Beispiele

CHOOSE(2; 3; 2; 4) ergibt 2

### 8.1.9.5 COLUMN

Die Funktion COLUMN gibt die Spalte eines angegebenen Zellenverweises zurück. Wenn kein Parameter angegeben wird, gibt die Funktion die Spalte der aktuellen Zelle zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

COLUMN(Zellenverweis)

#### Parameter

*Kommentar:* Verweis, *Typ:* Text

### Beispiele

COLUMN(A1) ergibt 1

### Beispiele

COLUMN(D2) ergibt 4

### Ähnliche Funktionen

COLUMNS  
ROW

#### 8.1.9.6 COLUMNS

Die Funktion COLUMNS gibt die Anzahl der Spalten in einem Zellenverweis zurück.

*Rückgabetyp:* Ganze Zahl (z. B. 1, 132, 2344)

### Syntax

COLUMNS(Zellenverweis)

### Parameter

*Kommentar:* Verweis, *Typ:* Text

### Beispiele

COLUMNS(A1:C3) ergibt 3

### Beispiele

COLUMNS(D2) ergibt 1

### Ähnliche Funktionen

COLUMN  
ROWS

#### 8.1.9.7 HLOOKUP

Sucht nach einem passenden Wert in der ersten Zeile der angegebenen Tabelle und gibt den Wert aus der angegebenen Zeile und der Spalte mit dem gefundenen Wert zurück.

Sucht nach dem „Suchwert“ in der ersten Zeile der „Datenquelle“. Wird ein passender Wert gefunden, dann wird der Wert in der angegebenen „Zeile“ und in der Spalte, in der der „Suchwert“ gefunden wurde, zurückgegeben. Ist „Sortiert“ wahr (Standard), dann wird angenommen, dass die erste Zeile sortiert ist. Die Suche ist beendet, wenn der „Suchwert“ kleiner als der aktuelle Vergleichswert ist.

*Rückgabetyp:* Zeichenfolge/Zahl

### Syntax

HLOOKUP(Suchwert; Datenquelle; Zeile; Sortiert)

### Parameter

*Kommentar:* Suchwert, *Typ:* Zeichenfolge/Zahl

*Kommentar:* Datenquelle, *Typ:* Feld/Bereich

*Kommentar:* Zeile, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Sortiert (optional), *Typ:* Wahrheitswert (WAHR oder FALSCH)

### 8.1.9.8 INDEX

Wenn ein Bereich angegeben ist, wird der in der entsprechenden Zeile/Spalte gespeicherte Wert zurückgegeben. Wenn eine Zelle angegeben ist, die ein Feld enthält, wird ein Element des Feldes zurückgegeben.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

INDEX(Zelle, Zeile, Spalte)

#### Parameter

*Kommentar:* Verweis, *Typ:* Text

*Kommentar:* Zeile, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Spalte, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

INDEX(A1:C3,2,2) ergibt den Inhalt von B2.

#### Beispiele

INDEX(A1,2,2) ergibt, falls A1 das Ergebnis einer Feldberechnung ist, sein (2,2) Element.

### 8.1.9.9 INDIRECT

Gibt den Inhalt einer Zelle welche als Verweis angegeben wird zurück. Der zweite Parameter ist optional.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

INDIRECT(Zellenverweis, A1 Stil)

#### Parameter

*Kommentar:* Verweis, *Typ:* Text

*Kommentar:* A1-Stil (optional), *Typ:* Wahrheitswert (WAHR oder FALSCH)

#### Beispiele

INDIRECT(A1), A1 enthält „B1“, und B1 1 = > ergibt 1

#### Beispiele

INDIRECT(„A1“), ergibt den Inhalt von A1

### 8.1.9.10 LOOKUP

Die Funktion LOOKUP sucht den ersten Parameter im Suchvektor. Sie ergibt einen Wert im Ergebnisvektor mit demselben Index wie der entsprechende Wert im Suchvektor. Wenn der Wert nicht im Suchvektor ist, wird der nächstkleinere verwendet. Wenn kein Wert im Suchvektor passt, wird ein Fehler ausgegeben. Der Suchvektor muss aufsteigend sortiert sein und Such- und Ergebnisvektor müssen dieselbe Größe haben. Numerische Werte, Zeichenfolgen und boolesche Werte werden akzeptiert.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

LOOKUP(Wert; Suchvektor; Ergebnisvektor)

### Parameter

*Kommentar:* Suchwert, *Typ:* Zeichenfolge/Zahl  
*Kommentar:* Suchvektor, *Typ:* Zeichenfolge/Zahl  
*Kommentar:* Ergebnisvektor, *Typ:* Zeichenfolge/Zahl

### Beispiele

LOOKUP(1,232; A1:A6; B1:B6) für A1 = 1, A2 = 2 ergibt den Wert von B1.

#### 8.1.9.11 MATCH

Sucht nach einem Suchwert in einem Suchbereich und gibt dessen Position beginnend mit 1 zurück. Die Art der Übereinstimmung kann entweder -1, 0 oder 1 sein und bestimmt die Art der Suche. Ist die Art der Übereinstimmung 0, dann wird der Index des ersten Werts gleich dem Suchwert zurückgegeben. Bei 1 als Art der Übereinstimmung oder keiner Angabe wird der Index des ersten Wertes kleiner oder gleich dem Suchwert zurückgegeben und die Werte im Suchbereich müssen aufsteigend sortiert sein. Ist die Art der Übereinstimmung -1, dann wird der Index des ersten Werts größer oder gleich dem Suchwert zurückgegeben und die Werte im Suchbereich müssen absteigend sortiert sein.

*Rückgabebetyp:* Ganze Zahl (z. B. 1, 132, 2344)

### Syntax

MATCH(Suchwert; Suchbereich; Art der Übereinstimmung)

### Parameter

*Kommentar:* Suchwert, *Typ:* Zeichenfolge/Zahl  
*Kommentar:* Suchbereich, *Typ:* Verweis/Feld  
*Kommentar:* Art der Übereinstimmung (optional), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### 8.1.9.12 MULTIPLE.OPERATIONS

MULTIPLE.OPERATIONS führt den Formelausdruck in der Formelzelle und alle darauf verweisenden Formelausdrücke. Dabei werden alle Verweise auf Zeilenzelle mit Verweisen auf Zeilenersatz und entsprechend alle Verweise auf Spaltenzelle mit Verweisen auf Spaltenersatz ersetzt. Diese Funktion kann dazu benutzt werden, Tabellen mit Ausdrücken zu erstellen, die von zwei Eingabeparametern abhängen.

*Rückgabebetyp:* Zeichenfolge/Zahl

### Syntax

MULTIPLE.OPERATIONS(Formelzelle; Zeilenzelle; Zeilenersatz; Spaltenzelle; Spaltenersatz)

### Parameter

*Kommentar:* Formelzelle, *Typ:* Verweis  
*Kommentar:* Zeilenzelle, *Typ:* Verweis  
*Kommentar:* Zeilenersatz, *Typ:* Verweis  
*Kommentar:* Spaltenzelle (optional), *Typ:* Verweis  
*Kommentar:* Spaltenersatz (optional), *Typ:* Verweis

### 8.1.9.13 OFFSET

Ändert die Position und Dimension eines Verweises.

*Rückgabetyp:* Verweis

#### Syntax

OFFSET(Referenz Zellenverweis; Ganzzahl Reihenversatz; Ganzzahl Spaltenversatz; Ganzzahl neueHöhe; Ganzzahl neueBreite)

#### Parameter

*Kommentar:* Verweis oder Bereich, *Typ:* Verweis

*Kommentar:* Anzahl der zu versetzenden Reihen, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Anzahl der zu versetzenden Spalten, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Höhe des Versatzbereichs (optional), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Breite des Versatzbereichs (optional), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### 8.1.9.14 ROW

Die Funktion ROW gibt die Zeile des angegebenen Zellenverweises zurück. Ohne Parameter gibt die Funktion die aktuelle Zeile zurück.

*Rückgabetyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

ROW(Zellenverweis)

#### Parameter

*Kommentar:* Verweis, *Typ:* Text

#### Beispiele

ROW(A1) ergibt 1

#### Beispiele

ROW(D2) ergibt 2

#### Ähnliche Funktionen

[ROWS](#)

[COLUMN](#)

### 8.1.9.15 ROWS

Die Funktion ROWS gibt die Anzahl der Zeilen in einem Zellenverweis zurück.

*Rückgabetyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

ROWS(Zellenverweis)

#### Parameter

*Kommentar:* Verweis, *Typ:* Text

#### Beispiele

ROWS(A1:C3) ergibt 3

### Beispiele

ROWS(D2) ergibt 1

### Ähnliche Funktionen

ROW  
COLUMNS

#### 8.1.9.16 SHEET

Gibt die Tabellennummer in dem angegebenen Zellenverweis oder eine Zeichenfolge mit dem Tabellennamen zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

SHEET(Zellenverweis)

#### Parameter

*Kommentar:* Verweis, *Typ:* Verweis

#### Beispiele

SHEET(Sheet1!C7) ergibt 1

#### Beispiele

SHEET(Sheet2!C7) ergibt 2

#### 8.1.9.17 SHEETS

Gibt die Anzahl der Tabellen in einem Verweis oder im aktuellen Dokument zurück.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

SHEETS(Zellenverweis)

#### Parameter

*Kommentar:* Verweis, *Typ:* Verweis

#### 8.1.9.18 VLOOKUP

Sucht nach einem passenden Wert in der ersten Spalte der angegebenen Tabelle und gibt den Wert aus der angegebenen Spalte und der Zeile mit dem gefundenen Wert zurück.

Sucht nach dem „Suchwert“ in der ersten Spalte der „Datenquelle“. Wird ein passender Wert gefunden, dann wird der Wert in der angegebenen „Spalte“ und in der Zeile, in der der „Suchwert“ gefunden wurde, zurückgegeben. Ist „Sortiert“ wahr (Standard), dann wird angenommen, dass die erste Spalte sortiert ist. Die Suche ist beendet, wenn der „Suchwert“ kleiner als der aktuelle Vergleichswert ist.

*Rückgabotyp:* Zeichenfolge/Zahl

#### Syntax

VLOOKUP(Suchwert; Datenquelle; Spalte; Sortiert)

#### Parameter

*Kommentar:* Suchwert, *Typ:* Zeichenfolge/Zahl

*Kommentar:* Datenquelle, *Typ:* Feld/Bereich

*Kommentar:* Spalte, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Sortiert (optional), *Typ:* Wahrheitswert (WAHR oder FALSCH)

## 8.1.10 Mathematik

### 8.1.10.1 ABS

Die Funktion ABS() ergibt den absoluten Wert der Fließkommazahl x.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ABS(x)

#### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

ABS(12,5) ergibt 12,5

#### Beispiele

ABS(-12,5) ergibt 12,5

### 8.1.10.2 CEIL

Die Funktion CEIL() ergibt den auf die nächste ganze Zahl nach oben gerundeten Wert als Fließkommawert.

*Rückgabebetyp:* Eine ganze Zahl wie 0, -5, 14

#### Syntax

CEIL(x)

#### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

CEIL(12,5) ergibt 13

#### Beispiele

CEIL(-12,5) ergibt -12

#### Ähnliche Funktionen

CEILING  
FLOOR  
ROUND  
ROUNDUP

### 8.1.10.3 CEILING

Die Funktion CEILING() rundet auf (weg von Null) zum nächsten Vielfachen der Signifikanz größer als der Eingabewert. Der Standardwert der Signifikanz ist 1, wenn x positiv ist. Wenn der Wert negativ ist, beträgt der Standardwert der Signifikanz -1, das bedeutet Aufrunden zur nächsten ganzen Zahl. Wenn ein Modus angegeben und ungleich Null ist, wird der Betrag von x weg von Null zu einem Vielfachen der Signifikanz gerundet. Ansonsten wird zu positiv Unendlich gerundet.

*Rückgabebetyp:* Eine ganze Zahl wie 0, -5, 14

### Syntax

CEILING(x)

### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)  
*Kommentar:* Signifikanz (optional), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)  
*Kommentar:* Modus (optional), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

CEILING(12,5) ergibt 13

### Beispiele

CEILING(6,43; 4) ergibt 8

### Beispiele

CEILING(-6,43; -4; 1) ergibt -8

### Beispiele

CEILING(-6,43; -4; 0) ergibt -4

### Ähnliche Funktionen

CEIL  
FLOOR  
ROUND  
ROUNDUP

#### 8.1.10.4 COUNT

Die Funktion COUNT() ergibt die Anzahl der übergebenen ganzzahligen und Fließkommawerte. Sie können einen Bereich zählen: COUNT(A1:B5). Sie können auch eine Liste von Werten zählen: COUNT(12;5;12,5).

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

COUNT(Wert;Wert;...)

### Parameter

*Kommentar:* Werte, *Typ:* FLOAT

### Beispiele

COUNT(-5;"KSpread";2) ergibt 2

### Beispiele

COUNT(5) ergibt 1

### Ähnliche Funktionen

COUNTA  
COUNTIF  
SUM

#### 8.1.10.5 COUNTA

Diese Funktion gibt die Anzahl aller nicht leeren Argumente zurück. Sie können auch einem Zellbereich zählen. COUNTA(A1:B5) oder benutzen Sie eine Liste von Werten wie COUNTA(12;5;12,5).

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

COUNTA(Wert;Wert;Wert ...)

##### Parameter

*Kommentar:* Werte, *Typ:* FLOAT

##### Beispiele

COUNTA(-5;"KSpread";2) ergibt 3

##### Beispiele

COUNTA(5) ergibt 1

##### Ähnliche Funktionen

[COUNT](#)  
[COUNTIF](#)

#### 8.1.10.6 COUNTBLANK

Diese Funktion gibt die Anzahl von allen leeren Zellen in dem angegebenen Bereich zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

COUNTBLANK(Zellbereich)

##### Parameter

*Kommentar:* Zellbereich, *Typ:* Bereich

##### Beispiele

COUNTBLANK(A1:B5)

##### Ähnliche Funktionen

[COUNT](#)  
[COUNTA](#)  
[COUNTIF](#)

#### 8.1.10.7 COUNTIF

Die Funktion COUNTIF() ergibt die Anzahl der Zellen im gegebenen Bereich, die das gegebene Kriterium erfüllen.

*Rückgabebetyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

COUNTIF(Bereich;Kriterium)

**Parameter**

*Kommentar:* Bereich, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)  
*Kommentar:* Kriterium, *Typ:* Text

**Beispiele**

COUNTIF(A2:A3;"14") ergibt 1, wenn A2 = -4 und A3 = 14 sind

**Ähnliche Funktionen**

COUNT  
SUMIF

**8.1.10.8 CUR**

Die Funktion CUR() ergibt die nicht negative kubische Wurzel von x.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

CUR(x)

**Parameter**

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Beispiele**

CUR(27) ergibt 3

**Ähnliche Funktionen**

SQRT

**8.1.10.9 DIV**

Die Funktion DIV() teilt den ersten Wert der Reihe nach durch die anderen Werte.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

DIV(Wert;Wert;...)

**Parameter**

*Kommentar:* Werte, *Typ:* FLOAT

**Beispiele**

DIV(20;2;2) ergibt 5

**Beispiele**

DIV(25;2,5) ergibt 10

**Ähnliche Funktionen**

MULTIPLY  
MOD

#### 8.1.10.10 EPS

Die Funktion EPS() ergibt das Rechner-epsilon. Das ist die Differenz zwischen 1 und der nächstgrößeren Fließkommazahl. Da Computer eine begrenzte Anzahl an Stellen verwenden, sind Rundungsfehler in allen Berechnungen vorhanden, aber normalerweise unbedeutend klein.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

EPS()

##### Parameter

##### Beispiele

Auf den meisten Systemen ergibt diese Funktion  $2^{-52}=2,2204460492503131e-16$

##### Beispiele

$0,5*EPS()$  ergibt die „unit round“; dieser Wert ist interessant, weil es der größte Wert ist, bei dem durch Rundungsfehler  $(1+x)-1=0$  ist.

##### Beispiele

EPS() ist so klein, dass Calligra Sheets  $1+eps()$  als 1 darstellt

##### Beispiele

Wählen Sie eine Zahl  $x$  zwischen 0 und EPS().  $1+x$  rundet  $x$  unter Verwendung der Gleichung  $(1+x)-1$  entweder zu 0 oder zu EPS().

#### 8.1.10.11 EVEN

Die Funktion EVEN() gibt die angegebene Zahl, aufgerundet zur nächsten geraden Zahl zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

EVEN(Wert)

##### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

EVEN(1,2) ergibt 2

##### Beispiele

EVEN(2) ergibt 2

##### Ähnliche Funktionen

[ODD](#)

#### 8.1.10.12 EXP

Die Funktion EXP() ergibt e „hoch“ x, wobei „e“ die Basis des natürlichen Logarithmus ist und „x“ der angegebene Parameter.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

EXP(x)

##### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

EXP(9) ergibt 8 103,08392758

##### Beispiele

EXP(-9) ergibt 0,00012341

#### Ähnliche Funktionen

[LN](#)

#### 8.1.10.13 FACT

Die Funktion fact() berechnet die Fakultät des Parameters. Der mathematische Ausdruck dazu ist x!.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

FACT(Zahl)

##### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

FACT(10) ergibt 3628800

##### Beispiele

FACT(0) ergibt 1

#### 8.1.10.14 FACTDOUBLE

Die Funktion FACTDOUBLE() berechnet die doppelte Fakultät des Parameters. Der mathematische Ausdruck dazu ist x!!.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

FACTDOUBLE(Zahl)

##### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

FACTDOUBLE(6) ergibt 48

##### Beispiele

FACTDOUBLE(7) ergibt 105

#### 8.1.10.15 FIB

Die Funktion FIB berechnet den n-ten Term einer Fibonacci-Folge (1, 1, 2, 3, 5, 8, 13, 21...). In dieser Folge ist nach den ersten beiden Zahlen jede Zahl die Summe der beiden direkt vorangegangenen Zahlen. FIB(0) ist als 0 definiert.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

FIB(n)

##### Parameter

*Kommentar:* n-ter Term, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

FIB(9) ergibt 34

##### Beispiele

FIB(26) ergibt 121393

#### 8.1.10.16 FLOOR

Eine Zahl x zum nächsten Vielfachen des zweiten Parameters, Signifikanz, abrunden.

Die Funktion FLOOR() rundet ab (gegen Null) zum nächsten Vielfachen der Signifikanz kleiner als der Eingabewert. Der Standardwert der Signifikanz ist 1, wenn x positiv ist. Wenn der Wert negativ ist, beträgt der Standardwert der Signifikanz -1, das bedeutet Aufrunden zur nächsten ganzen Zahl. Wenn ein Modus angegeben und ungleich Null ist, wird der Betrag von x gegen Null zu einem Vielfachen der Signifikanz gerundet und das Vorzeichen hinzugefügt. Ansonsten wird zu negativ Unendlich gerundet. Ist einer der beiden Parameter x oder die Signifikanz Null, ist auch das Ergebnis Null.

*Rückgabebetyp:* Eine ganze Zahl wie 0, -5, 14

##### Syntax

FLOOR(x)

##### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Signifikanz (optional), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Modus (optional), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

FLOOR(12,5) ergibt 12

##### Beispiele

FLOOR(-12,5) ergibt -13

##### Beispiele

FLOOR(5; 2) ergibt 4

##### Beispiele

FLOOR(5; 2,2) ergibt 4,4

##### Ähnliche Funktionen

CEIL

CEILING

ROUND

ROUNDDOWN

### 8.1.10.17 GAMMA

Die Funktion GAMMA() gibt den Wert der Gamma-Funktion zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

GAMMA(Wert)

#### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

GAMMA(1) ergibt 1

#### Ähnliche Funktionen

[FACT](#)

### 8.1.10.18 GCD

Die Funktion GCD() ergibt den größten gemeinsamen Teiler von zwei oder mehr ganzzahligen Werten.

*Rückgabebetyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

GCD(Wert;Wert)

#### Parameter

*Kommentar:* Erste Zahl, *Typ:* Eine Folge ganzer Zahlen (wie 1, 132, 2344)

*Kommentar:* Zweite Zahl, *Typ:* Eine Folge ganzer Zahlen (wie 1, 132, 2344)

*Kommentar:* Dritte Zahl, *Typ:* Eine Folge ganzer Zahlen (wie 1, 132, 2344)

#### Beispiele

GCD(6;4) ergibt 2

#### Beispiele

GCD(10;20) ergibt 10

#### Beispiele

GCD(20;15;10) ergibt 5

#### Ähnliche Funktionen

[LCM](#)

### 8.1.10.19 G\_PRODUCT

Die G\_PRODUCT()-Funktion ist identisch mit der KPRODUCT-Funktion. Sie ist für die Gnumeric-Kompatibilität nötig.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

G\_PRODUCT(Wert;Wert;...)

#### Parameter

*Kommentar:* Werte, *Typ:* FLOAT

#### Ähnliche Funktionen

[KPRODUCT](#)

#### 8.1.10.20 INT

Die Funktion INT() ergibt den ganzzahligen Teil des angegebenen Parameters.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

INT(x)

##### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

INT(12,55) ergibt 12

##### Beispiele

INT(15) ergibt 15

##### Ähnliche Funktionen

[FLOOR](#)  
[QUOTIENT](#)

#### 8.1.10.21 INV

Diese Funktion multipliziert jeden Wert mit -1

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

INV(Wert)

##### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

INV(-5) ergibt 5

##### Beispiele

INV(5) ergibt -5

##### Beispiele

INV(0) ergibt 0

#### 8.1.10.22 KPRODUCT

Die Funktion KPRODUCT() berechnet das Produkt von allen als Parameter übergebenen Werten. Sie können das Produkt eines Zellbereichs: KPRODUCT(A1:B5) oder der Liste von Parametern wie KPRODUCT(12;5;12,5) berechnen lassen. Wenn keine numerischen Werte gefunden werden ergibt die Funktion 1.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

KPRODUCT(Wert;Wert;...)

**Parameter**

*Kommentar:* Werte, *Typ:* FLOAT

**Beispiele**

KPRODUCT(3;5;7) ergibt 105

**Beispiele**

KPRODUCT(12,5;2) ergibt 25

**Ähnliche Funktionen**

[G\\_PRODUCT](#)  
[MULTIPLY](#)  
[PRODUCT](#)

**8.1.10.23 LCM**

Die Funktion LCM() ergibt das kleinste gemeinsame Vielfache von zwei oder mehr Fließkommawerten.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

LCM(Wert;Wert)

**Parameter**

*Kommentar:* Erste Zahl, *Typ:* FLOAT

*Kommentar:* Zweite Zahl, *Typ:* FLOAT

**Beispiele**

LCM(6;4) ergibt 12

**Beispiele**

LCM(1,5;2,25) ergibt 4,5

**Beispiele**

LCM(2;3;4) ergibt 12

**Ähnliche Funktionen**

[GCD](#)

**8.1.10.24 LN**

Die Funktion LN() ergibt den natürlichen Logarithmus von x.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

LN(x)

**Parameter**

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Beispiele**

LN(0,8) ergibt -0,22314355

### Beispiele

LN(0) ergibt -unendlich

### Ähnliche Funktionen

LOG  
LOG10  
LOG2

#### 8.1.10.25 LOG

Die Funktion LOG() ergibt den Logarithmus zur Basis 10 von x.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

LOG(x)

### Parameter

*Kommentar:* Ein Fließkommawert größer als 0, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

LOG(0,8) ergibt -0,09691001

### Beispiele

LOG(0) ergibt einen Fehler.

### Ähnliche Funktionen

LN  
LOGN  
LOG10  
LOG2

#### 8.1.10.26 LOG10

Die Funktion LOG10() ergibt den Logarithmus zur Basis 10 des Arguments.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

LOG10(x)

### Parameter

*Kommentar:* Ein positiver Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

LOG10(10) ergibt 1

### Beispiele

LOG10(0) ergibt einen Fehler.

### Ähnliche Funktionen

LN  
LOGN  
LOG  
LOG2

### 8.1.10.27 LOG2

Die Funktion LOG2() ergibt den Logarithmus zur Basis 2 von x.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

LOG2(x)

#### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

LOG2(0,8) ergibt -0,32192809

#### Beispiele

LOG2(0) ergibt -unendlich

#### Ähnliche Funktionen

LN  
LOGN  
LOG  
LOG10

### 8.1.10.28 LOGN

Die Funktion LOGN() ergibt den Logarithmus zur Basis n von x.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

LOGN(Wert;Basis)

#### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Basis, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

LOGN(12; 10) ergibt 1,07918125

#### Beispiele

LOGN(12;2) ergibt 3,5849625

#### Ähnliche Funktionen

LOG  
LN  
LOG10  
LOG2

### 8.1.10.29 MAX

Die Funktion MAX() ergibt den größten Wert, der als Parameter angegeben wurde. Zeichenfolge und boolesche Werte werden ignoriert.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

MAX(Wert;Wert;...)

#### Parameter

*Kommentar:* Werte, *Typ:* FLOAT

#### Beispiele

MAX(12;5; 7) ergibt 12

#### Beispiele

MAX(12,5; 2) ergibt 12,5

#### Beispiele

MAXA(0,5; 0,4; Wahr; 0,2) ergibt 0,5

#### Ähnliche Funktionen

COUNT  
COUNTA  
MAXA  
MIN  
MINA

### 8.1.10.30 MAXA

Die Funktion MAXA() gibt den größten als Parameter übergebenen Wert zurück ohne boolesche Werte zu ignorieren. Der boolesche Wert Wahr entspricht 1, der Wert Falsch entspricht 0. Zeichenfolgen werden ignoriert.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

MAXA(Wert;Wert;...)

#### Parameter

*Kommentar:* Werte, *Typ:* FLOAT

#### Beispiele

MAXA(12;5; 7) ergibt 12

#### Beispiele

MAXA(12,5; 2) ergibt 12,5

#### Beispiele

MAXA(0,5; 0,4; Wahr; 0,2) ergibt 1

#### Ähnliche Funktionen

COUNT  
COUNTA  
MAX  
MIN  
MINA

### 8.1.10.31 MDETERM

Die Funktion MDETERM gibt die Determinante einer gegebenen Matrix aus. Die Matrix muss vom Typ n x n sein.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

MDETERM(Matrix)

#### Parameter

*Kommentar:* Bereich, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

#### Beispiele

MDETERM(A1:C3)

#### Ähnliche Funktionen

[MMULT](#)

### 8.1.10.32 MIN

Die Funktion MIN() ergibt den kleinsten Wert, welcher als Parameter angegeben wurde. Zeichenfolgen und boolesche Werte werden ignoriert.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

MIN(Wert;Wert;...)

#### Parameter

*Kommentar:* Werte, *Typ:* FLOAT

#### Beispiele

MIN(12;5; 7) ergibt 5

#### Beispiele

MIN(12,5; 2) ergibt 2

#### Beispiele

MIN(0,4, 2; Falsch, 0,7) ergibt 0,4

#### Ähnliche Funktionen

[COUNT](#)  
[COUNTA](#)  
[MAX](#)  
[MAXA](#)  
[MINA](#)

### 8.1.10.33 MINA

Die Funktion MINA() gibt den kleinsten als Parameter übergebenen Wert zurück ohne boolesche Werte zu ignorieren. Der boolesche Wert Wahr entspricht 1, der Wert Falsch entspricht 0. Zeichenfolgen werden ignoriert.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

MINA(Wert;Wert;...)

#### Parameter

*Kommentar:* Werte, *Typ:* FLOAT

#### Beispiele

MINA(12;5;7) ergibt 5

#### Beispiele

MINA(12,5; 2) ergibt 2

#### Beispiele

MINA(0,4, 2; Falsch, 0,7) ergibt 0

#### Ähnliche Funktionen

COUNT  
COUNTA  
MAX  
MAXA  
MIN

### 8.1.10.34 MINVERSE

Berechnet die Inverse der Matrix.

Die Matrix multipliziert mit ihrer Inversen ergibt die Einheitsmatrix derselben Dimension.

Die Determinante invertierbarer Matrizen ist nicht 0.

*Rückgabebetyp:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

#### Syntax

MINVERSE(Matrix)

#### Parameter

*Kommentar:* Matrix, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

#### Beispiele

MINVERSE(A1:C3)

#### Ähnliche Funktionen

MDETERM  
MUNIT

#### 8.1.10.35 MMULT

Die Funktion MMULT multipliziert zwei Matrizen. Die Spaltenanzahl der ersten Matrix muss der Zeilenanzahl der zweiten entsprechen. Das Ergebnis ist eine Matrix.

*Rückgabebetyp:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

##### Syntax

MMULT(Matrix1;Matrix2)

##### Parameter

*Kommentar:* Erste Matrix, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Zweite Matrix, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

##### Beispiele

MMULT(A1:C3)

##### Ähnliche Funktionen

[MDETERM](#)

#### 8.1.10.36 MOD

Die Funktion MOD() ergibt den Rest einer Division des ersten Parameters mit dem zweiten Parameter. Wenn der zweite Parameter null ist, wird #DIV/0 zurückgegeben.

*Rückgabebetyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

MOD(Wert;Wert)

##### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Fließkommawert, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

MOD(12;5) ergibt 2

##### Beispiele

MOD(5;5) ergibt 0

##### Ähnliche Funktionen

[DIV](#)

#### 8.1.10.37 MROUND

Die Funktion MROUND() ergibt den Wert auf ein angegebenes Vielfaches gerundet zurück. Der Wert und das Vielfache müssen dasselbe Vorzeichen haben.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

MROUND(Wert; Vielfaches)

### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Vielfaches, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

MROUND(1,252; 0,5) ergibt 1,5

### Beispiele

MROUND(-1,252; -0,5) ergibt -1,5

### Ähnliche Funktionen

[ROUND](#)

#### 8.1.10.38 MULTINOMIAL

Die Funktion MULTINOMIAL() berechnet das „Multinom“ für die Zahlen in der Parameterliste. Die folgende Formel wird für MULTINOMIAL(a,b,c) verwendet:

$(a+b+c)! / a!b!c!$

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

MULTINOMIAL(Wert;Wert;...)

### Parameter

*Kommentar:* Werte, *Typ:* FLOAT

### Beispiele

MULTINOMIAL(3;4;5) ergibt 27720

#### 8.1.10.39 MULTIPLY

Die Funktion MULTIPLY() multipliziert alle als Parameter angegebenen Werte miteinander. Sie können einen Wertebereich MULTIPLY(A1:B5), aber auch eine Wertliste MULTIPLY(12;5;12,5) angeben. Sie ist mit der PRODUCT-Funktion identisch.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

MULTIPLY(Wert;Wert;...)

### Parameter

*Kommentar:* Werte, *Typ:* FLOAT

### Beispiele

MULTIPLY(12;5;7) ergibt 420

### Beispiele

MULTIPLY(12,5;2) ergibt 25

### Ähnliche Funktionen

[DIV](#)  
[PRODUCT](#)  
[KPRODUCT](#)

#### 8.1.10.40 MUNIT

Erstellt die Einheitsmatrix der angegebenen Dimension.

*Rückgabebetyp:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

##### Syntax

MUNIT(Dimension)

##### Parameter

*Kommentar:* Dimension, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

MUNIT(3) erstellt eine 3×3 Einheitsmatrix.

##### Ähnliche Funktionen

[MINVERSE](#)

#### 8.1.10.41 ODD

Die Funktion ODD() rundet eine Zahl auf die nächste ungerade ganze Zahl auf (oder ab für negative Zahlen). ODD(0) ist definitionsgemäß 1.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

ODD(Wert)

##### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

ODD(1,2) ergibt 3

##### Beispiele

ODD(2) ergibt 3

##### Beispiele

ODD(-2) ergibt -3

##### Ähnliche Funktionen

[EVEN](#)

#### 8.1.10.42 POW

Die Funktion POW(x;y) ergibt dem Wert x potenziert mit y. (x „hoch“ y). Sie ist identisch mit der POWER-Funktion.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

POW(Wert;Wert)

**Parameter**

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Beispiele**

POW(1,2;3,4) ergibt 1,8572

**Beispiele**

POW(2;3) ergibt 8

**Ähnliche Funktionen**

[POWER](#)

**8.1.10.43 POWER**

Die Funktion POWER(x;y) ergibt dem Wert x potenziert mit y. (x „hoch“ y).

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

POWER(Wert;Wert)

**Parameter**

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Beispiele**

POWER(1,2;3,4) ergibt 1,8572

**Beispiele**

POWER(2;3) ergibt 8

**Ähnliche Funktionen**

[POW](#)

**8.1.10.44 PRODUCT**

Die Funktion PRODUCT() berechnet das Produkt von allen als Parameter übergebenen Werten. Sie können das Produkt eines Zellbereichs: PRODUCT(A1:B5) oder der Liste von Parametern wie PRODUCT(12;5;12,5) berechnen lassen. Wenn keine numerischen Werte gefunden werden ergibt die Funktion 0.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

PRODUCT(Wert;Wert;...)

**Parameter**

*Kommentar:* Werte, *Typ:* FLOAT

**Beispiele**

PRODUCT(3;5;7) ergibt 105

**Beispiele**

PRODUCT(12,5;2) ergibt 25

**Ähnliche Funktionen**

[MULTIPLY](#)

[KPRODUCT](#)

#### 8.1.10.45 QUOTIENT

Die Funktion QUOTIENT() ergibt den ganzzahligen Teil des angegebenen Quotienten Zähler / Nenner.

*Rückgabebetyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

QUOTIENT(Zähler;Nenner)

##### Parameter

*Kommentar:* Zähler, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Nenner, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

QUOTIENT(21;4) ergibt 5

##### Ähnliche Funktionen

[INT](#)

#### 8.1.10.46 RAND

Die Funktion RAND() gibt eine pseudo-zufällige Zahl zwischen 0 und 1 zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

RAND()

##### Parameter

##### Beispiele

RAND() ergibt z. B. 0,78309922...

##### Ähnliche Funktionen

[RANDBETWEEN](#)

[RANDEXP](#)

#### 8.1.10.47 RANDBERNOULLI

Die Funktion RANDBERNOULLI() gibt eine Bernoulli-verteilte pseudo-zufällige Zahl zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

RANDBERNOULLI(x)

##### Parameter

*Kommentar:* Ein Fließkommawert (zwischen 0 und 1), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

RANDBERNOULLI(0,45)

##### Ähnliche Funktionen

[RAND](#)

#### 8.1.10.48 RANDBETWEEN

Die Funktion RANDBETWEEN() gibt eine pseudo-zufällige Zahl zwischen dem oberen und dem unteren Wert zurück. Ist der untere Wert größer als der obere, gibt diese Funktion „Err“ zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

RANDBETWEEN(Unterer;Oberer)

##### Parameter

*Kommentar:* Unterer Wert, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Oberer Wert, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

RANDBETWEEN(12;78) ergibt z. B. 61,0811...

##### Ähnliche Funktionen

[RAND](#)

#### 8.1.10.49 RANDBINOM

Die Funktion RANDBINOM() gibt eine binomisch verteilte pseudo-zufällige zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

RANDBINOM(x)

##### Parameter

*Kommentar:* Ein Fließkommawert (zwischen 0 und 1), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Versuche (größer 0), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

RANDBINOM(4)

##### Ähnliche Funktionen

[RAND](#)

[RANDNEGBINOM](#)

#### 8.1.10.50 RANDEXP

Die Funktion RANDEXP() gibt eine exponentiell verteilte pseudo-zufällige Zahl zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

RANDEXP(x)

##### Parameter

*Kommentar:* Ein Fließkommawert (größer als 0), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

RANDEXP(0,88)

##### Ähnliche Funktionen

[RAND](#)

#### 8.1.10.51 RANDNEGBINOM

Die Funktion RANDNEGBINOM() gibt eine negativ binomisch verteilte pseudo-zufällige Zahl zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

RANDNEGBINOM(x)

##### Parameter

*Kommentar:* Ein Fließkommawert (zwischen 0 und 1), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fehler (größer 0), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

RANDNEGBINOM(4)

##### Ähnliche Funktionen

[RAND](#)

[RANDBINOM](#)

#### 8.1.10.52 RANDNORM

Die Funktion RANDNORM() gibt eine normal-(Gauß-)verteilte pseudo-zufällige Zahl zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

RANDNORM(mü; sigma)

##### Parameter

*Kommentar:* Mittelwert der Normalverteilung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Standardabweichung der Normalverteilung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

RANDNORM(0; 1)

##### Ähnliche Funktionen

[RAND](#)

#### 8.1.10.53 RANDPOISSON

Die Funktion RANDPOISSON() gibt eine Poisson-verteilte pseudo-zufällige Zahl zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

RANDPOISSON(x)

##### Parameter

*Kommentar:* Ein Fließkommawert (größer als 0), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

RANDPOISSON(4)

### Ähnliche Funktionen

[RAND](#)

#### 8.1.10.54 ROOTN

Die Funktion ROOTN() ergibt die nicht negative n-te Wurzel von x.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

ROOTN(x;n)

### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Wert, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

ROOTN(9;2) ergibt 3

### Ähnliche Funktionen

[SQRT](#)

#### 8.1.10.55 ROUND

Die Funktion ROUND(Wert;[Ziffern]) gibt die gerundete Zahl zurück. „Ziffern“ ist dabei die Anzahl der Nachkommaziffern, auf welche Sie die Zahl runden möchten. Falls Sie „Ziffern“ weglassen oder auf 0 setzen, wird die angegebene Zahl auf den nächsten ganzzahligen Wert gerundet. Wenn „Ziffern“ kleiner als 0 ist, wird die Zahl auf einen entsprechende ganzzahligen Wert gerundet.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

ROUND(Wert;[Ziffern])

### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Ziffern, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

ROUND(1,252;2) ergibt 1,25

### Beispiele

ROUND(-1,252;2) ergibt -1,25

### Beispiele

ROUND(1,258;2) ergibt 1,26

### Beispiele

ROUND(-12,25;-1) ergibt -10

### Beispiele

ROUND(-1,252;0) ergibt -1

### Ähnliche Funktionen

MROUND  
ROUNDDOWN  
ROUNDUP

#### 8.1.10.56 ROUNDDOWN

Die Funktion ROUNDDOWN(Wert;[Ziffern]) gibt eine Zahl mit abgerundetem Betrag zurück. „Ziffern“ ist dabei die Anzahl der Nachkommaziffern, auf welche Sie die Zahl runden möchten. Falls Sie „Ziffern“ weglassen oder auf 0 setzen, wird die angegebene Zahl auf den nächsten ganzzahligen Wert abgerundet.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

ROUNDDOWN(Wert; [Ziffern])

### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Ziffern, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

ROUNDDOWN(1,252) ergibt 1

### Beispiele

ROUNDDOWN(1,252;2) ergibt 1,25

### Beispiele

ROUNDDOWN(-1,252;2) ergibt -1,25

### Beispiele

ROUNDDOWN(-1,252) ergibt -1

### Ähnliche Funktionen

ROUND  
ROUNDUP

#### 8.1.10.57 ROUNDUP

Die Funktion ROUNDUP(Wert;[Ziffern]) gibt eine Zahl mit aufgerundetem Betrag zurück. „Ziffern“ ist dabei die Anzahl der Nachkommaziffern, auf welche Sie die Zahl runden möchten. Falls Sie „Ziffern“ weglassen oder auf 0 setzen, wird die angegebene Zahl auf den nächsten ganzzahligen Wert aufgerundet.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

ROUNDUP(Wert;[Ziffern])

### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Ziffern, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

**Beispiele**

ROUNDUP(1,252) ergibt 2

**Beispiele**

ROUNDUP(1,252;2) ergibt 1,26

**Beispiele**

ROUNDUP(-1,252;2) ergibt -1,26

**Beispiele**

ROUNDUP(-1,252) ergibt -2

**Ähnliche Funktionen**

[ROUND](#)

[ROUNDDOWN](#)

**8.1.10.58 SERIESSUM**

Die Funktion SERIESSUM() gibt die Summe einer Potenzreihe zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

SERIESSUM( X; N; M; Koeffizienten)

**Parameter**

*Kommentar:* X ist die unabhängige Variable der Potenzreihe, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* N ist die Anfangspotenz, in die Sie x erheben möchten, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* M ist das Inkrement, um das Sie N in jedem Glied der Reihe vergrößern möchten, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Koeffizienten ist eine Gruppe von Koeffizienten, mit denen die aufeinander folgenden Potenzen der Variablen X multipliziert werden, *Typ:* FLOAT

**Beispiele**

SERIESSUM(2;0;2;{1;2}) ergibt 9

**8.1.10.59 SIGN**

Diese Funktion ergibt -1, wenn die Zahl negativ ist. Sie ergibt 0, wenn die Zahl null ist. Sie ergibt 1, wenn die Zahl positiv ist.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

**Syntax**

SIGN(Wert)

**Parameter**

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Beispiele**

SIGN(5) ergibt 1

**Beispiele**

SIGN(0) ergibt 0

**Beispiele**

SIGN(-5) ergibt -1

#### 8.1.10.60 SQRT

Die Funktion SQRT() ergibt die nicht negative Quadratwurzel des Arguments. Wenn das Argument negativ ist, wird ein Fehler zurückgegeben.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SQRT(x)

##### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

SQRT(9) ergibt 3

##### Beispiele

SQRT(-9) ergibt einen Fehler

##### Ähnliche Funktionen

[IMSQRT](#)

#### 8.1.10.61 SQRTPI

Die Funktion SQRTPI() ergibt die nicht negative Quadratwurzel von  $x * \pi$ . Wenn das Argument negativ ist, wird ein Fehler zurückgegeben.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SQRTPI(x)

##### Parameter

*Kommentar:* Ein Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

SQRTPI(2) ergibt 2,506628

#### 8.1.10.62 SUBTOTAL

Die Funktion SUBTOTAL() gibt eine Zwischensumme von einer angegebenen Liste von Argumenten. Dabei werden andere Zwischensummen ignoriert. Funktion kann eine der folgenden Zahlen sein: 1 – Average, 2 – Count, 3 – CountA, 4 – Max, 5 – Min, 6 – Product, 7 – StDev, 8 – StDevP, 9 – Sum, 10 – Var, 11 – VarP.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SUBTOTAL(Funktion; Wert)

##### Parameter

*Kommentar:* Funktion, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Werte, *Typ:* FLOAT

### Beispiele

Wenn A1:A5 7, 24, 23, 56 und 9 enthält:

### Beispiele

SUBTOTAL(1; A1:A5) ergibt 23,8

### Beispiele

SUBTOTAL(4; A1:A5) ergibt 56

### Beispiele

SUBTOTAL(9; A1:A5) ergibt 119

### Beispiele

SUBTOTAL(11; A1:A5) ergibt 307,76

### Ähnliche Funktionen

AVERAGE  
COUNT  
COUNTA  
MAX  
MIN  
PRODUCT  
STDEV  
STDEVP  
SUM  
VAR  
VARP

### 8.1.10.63 SUM

Die Funktion SUM() berechnet die Summe von allen Werten, welche als Parameter angegeben wurden. Sie können die Summe eines Zellbereichs SUM(A1:B5) oder einer Werteliste SUM(12;5;12,5) berechnen lassen.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

SUM(Wert;Wert;...)

### Parameter

*Kommentar:* Werte, Typ: FLOAT

### Beispiele

SUM(12;5;7) ergibt 24

### Beispiele

SUM(12,5;2) ergibt 14,5

### Ähnliche Funktionen

SUMA  
SUMSQ  
SUMIF

#### 8.1.10.64 SUMA

Die Funktion SUMA() berechnet die Summe von allen als Parameter übergebenen Werten. Sie können die Summe von einem Zellbereich SUMA(A1:B5) oder eine Liste von Parametern wie SUMA(12;5;12,5) berechnen lassen. Wenn ein Parameter Text enthält, oder den booleschen Wert Falsch wird er als 0 gezählt. Wenn ein Parameter den booleschen Wert Wahr enthält wird dieser als 1 gezählt.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SUM(Wert;Wert;...)

##### Parameter

*Kommentar:* Werte, *Typ:* FLOAT

##### Beispiele

SUMA(12;5; 7) ergibt 24

##### Beispiele

SUMA(12,5; 2; Wahr) ergibt 15,5

##### Ähnliche Funktionen

SUM  
SUMSQ

#### 8.1.10.65 SUMIF

Die Funktion SUMIF() berechnet die Summe aller als Parameter übergebenen Werte, die die Bedingung erfüllen. Die Angabe des Summenbereichs ist optional. Wird er nicht übergeben, werden die Werte in Bereich summiert. Die Länge von Bereich sollte kleiner oder gleich der Länge von Summenbereich sein.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SUMIF(Bereich;Bedingung;Summenbereich)

##### Parameter

*Kommentar:* Prüfbereich, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Kriterium, *Typ:* Text

*Kommentar:* Bereich, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

##### Beispiele

SUMIF(A1:A4;">1") summiert alle Werte im Bereich A1:A4, die die Bedingung >1 erfüllen

##### Beispiele

SUMIF(A1:A4;"=0";B1:B4) summiert alle Werte im Bereich B1:B4 wenn der entsprechende Wert in A1:A4 die Bedingung =0 erfüllt.

##### Ähnliche Funktionen

SUM  
COUNTIF

#### 8.1.10.66 SUMSQ

Die Funktion SUMSQ() berechnet die Summe der Quadrate der übergebenen Werte („hoch 2“). Sie können die Summe eines Zellbereichs SUMSQ(A1:B5) oder einer Werteliste SUMSQ(12;5;12,5) berechnen lassen.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SUMSQ(Wert;Wert;...)

##### Parameter

*Kommentar:* Werte, *Typ:* FLOAT

##### Beispiele

SUMSQ(12;5;7) ergibt 218

##### Beispiele

SUMSQ(12;5;2) ergibt 173

##### Ähnliche Funktionen

[SUM](#)

#### 8.1.10.67 TRANSPOSE

Ergibt die Transponierte einer Matrix. Das bedeutet, dass Zeilen und Spalten der Matrix ausgetauscht sind.

*Rückgabotyp:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

##### Syntax

TRANSPOSE(Matrix)

##### Parameter

*Kommentar:* Matrix, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

##### Beispiele

TRANSPOSE(A1:C3)

#### 8.1.10.68 TRUNC

Die Funktion TRUNC() beschneidet einen Wert auf eine angegebene Genauigkeit. Wenn der Parameter Genauigkeit fehlt, wird 0 angenommen.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

TRUNC(Wert; Genauigkeit)

##### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Genauigkeit, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

TRUNC(1,2) ergibt 1

## Beispiele

TRUNC(213,232; 2) ergibt 213,23

## Ähnliche Funktionen

ROUND  
ROUNDDOWN  
ROUNDUP

## 8.1.11 Statistik

### 8.1.11.1 AVEDEV

Die Funktion AVEDEV() berechnet den Durchschnitt der absoluten Abweichungen eines Wertbereiches AVEDEV(A1:B5) oder einer Liste von Werten AVEDEV(12;5;12,5) von seinem Mittelwert.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

AVEDEV(Wert;Wert;...)

#### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

## Beispiele

AVEDEV(11,4;17,3;21,3;25,9;40,1) ergibt 7,84

## Beispiele

AVEDEV(A1:A5) ...

### 8.1.11.2 AVERAGE

Die Funktion AVERAGE() berechnet den Durchschnitt aller als Parameter angegebenen Werte. Sie können einen Wertbereich AVERAGE(A1:B5), aber auch eine Liste von Werten AVERAGE(12;5;12,5) berechnen.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

AVERAGE(Wert;Wert;...)

#### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

## Beispiele

AVERAGE(12;5;7) ergibt 8

## Beispiele

AVERAGE(12,5;2) ergibt 7,25

### 8.1.11.3 AVERAGEA

Die Funktion AVERAGEA() berechnet den Durchschnitt der angegebenen Parameter. Sowohl Zahlen als auch Text und boolesche Werte werden in der Berechnung verwendet. Wenn die Zelle Text oder „Falsch“ enthält, wird sie als null (0) gewertet. Wenn ein boolesches „Wahr“ in der Zelle steht, wird es als 1 gewertet. Leere Zellen werden nicht in die Berechnung mit einbezogen.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

AVERAGEA(Wert;Wert;...)

#### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Zeichenfolgen, *Typ:* Text

#### Beispiele

AVERAGEA(11,4;17,3;"eintext";25,9;40,1) ergibt 18,94

### 8.1.11.4 BETADIST

Die Funktion BETADIST() gibt die kumulierte Beta-Wahrscheinlichkeitsdichte zurück.

Der dritte und vierte Parameter sind optional. Sie geben die untere und die obere Schranke an. Wenn sie weggelassen werden, werden Standardwerte von 0,0 bzw. 1,0 verwendet.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

BETADIST(Zahl;Alpha;Beta;Start;Ende;[kumulativ=WAHR])

#### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Alpha-Parameter, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Beta-Parameter, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Start, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Ende, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Kumulativ, *Typ:* Wahrheitswert (WAHR oder FALSCH)

#### Beispiele

BETADIST(0,2859;0,2606;0,8105) ergibt 0,675444

#### Beispiele

BETADIST(0,2859;0,2606;0,8105;0,2;0,9) ergibt 0,537856

### 8.1.11.5 BETAINV

Die Funktion BETAINV() ergibt die Umkehrfunktion von BETADIST(x;Alpha;Beta;a;b;WAHR()).

Die Start- und Endparameter sind optional. Sie geben die untere und die obere Schranke an. Wenn sie weggelassen werden, werden Standardwerte von 0,0 bzw. 1,0 verwendet.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

BETAINV(Zahl;Alpha;Beta [; Start=0 [; Ende=1]])

### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Alpha-Parameter, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Beta-Parameter, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Start, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Ende, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

BETADIST(BETAINV(0,1;3;4);3;4) ergibt 0,1

### Beispiele

BETADIST(BETAINV(0,3;3;4);3;4) ergibt 0,3

### 8.1.11.6 BINO

Die Funktion BINO() gibt die binomische Verteilung zurück.

Der erste Parameter ist die Zahl der Versuche, der zweite Parameter ist die Zahl der Erfolge und der dritte Parameter ist die Wahrscheinlichkeit des Erfolgs. Die Zahl der Versuche sollte größer sein als die Zahl der Erfolge und die Wahrscheinlichkeit sollte kleiner oder 1 sein.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

BINO(Versuche;Erfolge;Wahrsch\_des\_Erfolgs)

### Parameter

*Kommentar:* Anzahl der Versuche, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Anzahl erfolgreicher Versuche, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Wahrscheinlichkeit des Erfolges, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

BINO(12;9;0,8) ergibt 0,236223201

### 8.1.11.7 CHIDIST

Die Funktion CHIDIST() berechnet aus einem angegebenen Chi-Quadrat den Wahrscheinlichkeitswert, dass eine Hypothese bestätigt wird.

CHIDIST vergleicht den für eine Zufallsstichprobe zu gebenden Chiquadratwert, der aus folgender Summe errechnet wird:  $(\text{beobachteter Wert} - \text{erwarteter Wert})^2 / \text{erwarteter Wert}$  für alle Werte mit der theoretischen Chiquadratverteilung und berechnet daraus die Wahrscheinlichkeit des Fehlers, um die Hypothese zu prüfen.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

CHIDIST(Zahl;Freiheitsgrade)

### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Freiheitsgrade, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

CHIDIST(13,27;5) ergibt 0,021

#### 8.1.11.8 COMBIN

Die Funktion COMBIN() berechnet die Zahl der möglichen Kombinationen. Der erste Parameter ist die Gesamtzahl der Elemente. Der zweite Parameter ist die Anzahl der zu wählenden Elemente. Diese beiden Parameter müssen positiv sein, sonst gibt die Funktion einen Fehler zurück. Der erste Parameter muss größer als der zweite sein, sonst gibt die Funktion auch einen Fehler zurück.

*Rückgabebetyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

COMBIN(gesamt;gewählt)

##### Parameter

*Kommentar:* Anzahl der Elemente, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Anzahl der auszuwählenden Elemente, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

COMBIN(12;5) ergibt 792

##### Beispiele

COMBIN(5;5) ergibt 1

#### 8.1.11.9 COMBINA

Die Funktion COMBINA() berechnet die Zahl der möglichen Kombinationen. Der erste Parameter ist die Gesamtzahl der Elemente. Der zweite Parameter ist die Anzahl der zu wählenden Elemente. Diese beiden Parameter müssen positiv sein, sonst gibt die Funktion einen Fehler zurück. Der erste Parameter muss größer als der zweite sein, sonst gibt die Funktion auch einen Fehler zurück.

*Rückgabebetyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

COMBIN(gesamt;gewählt)

##### Parameter

*Kommentar:* Anzahl der Elemente, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Anzahl der auszuwählenden Elemente, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

COMBIN(12;5) ergibt 792

##### Beispiele

COMBIN(5;5) ergibt 1

#### 8.1.11.10 CONFIDENCE

Die Funktion CONFIDENCE() gibt den Vertrauensbereich eines Populationsmittel zurück.

Der Alpha-Parameter muss zwischen 0 und 1 liegen (exklusive 0 und 1 selbst), stdev muss positiv und Größe muss größer gleich 1 sein.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

CONFIDENCE(alpha;stddev;Größe)

### Parameter

*Kommentar:* Ebene des Konfidenzintervalls, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Standardabweichung für die gesamte Population, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Größe der ganzen Population, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

CONFIDENCE(0,05;1,5;100) ergibt 0,294059

## 8.1.11.11 CORREL

Die Funktion CORREL() berechnet den Korrelationskoeffizienten von zwei Zellbereichen.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

CORREL(Bereich1; Bereich2)

### Parameter

*Kommentar:* Zellbereich von Werten, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Zweiter Zellbereich von Werten, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

CORREL(A1:A3; B1:B3)

### Ähnliche Funktionen

[PEARSON](#)

## 8.1.11.12 COVAR

Die Funktion COVAR() berechnet die Kovarianz von zwei Zellbereichen.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

COVAR(Bereich1; Bereich2)

### Parameter

*Kommentar:* Zellbereich von Werten, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Zweiter Zellbereich von Werten, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

COVAR(A1:A3; B1:B3)

### 8.1.11.13 DEVSQ

Die Funktion DEVSQ() berechnet die Summe von quadratischen Abweichungen.

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

DEVSQ(Wert; Wert;...)

#### Parameter

*Kommentar*: Fließkommawerte, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Fließkommawerte, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Fließkommawerte, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Fließkommawerte, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Fließkommawerte, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

DEVSQ(A1:A5)

#### Beispiele

DEVSQ(21; 33; 54; 23) ergibt 684,75

### 8.1.11.14 EXPONDIST

Die Funktion EXPONDIST() gibt die exponentielle Verteilung zurück.

Der lambda Parameter muss positiv sein.

Kumulativ = 0 berechnet die Dichtefunktion, Kumulativ = 1 berechnet die Verteilung.

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

EXPONDIST(Zahl;lambda;kumulativ)

#### Parameter

*Kommentar*: Zahl, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Lambda-Parameter, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: 0 = Dichte, 1 = Verteilung, *Typ*: Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

EXPONDIST(3;0,5;0) ergibt 0,111565

#### Beispiele

EXPONDIST(3;0,5;1) ergibt 0,776870

### 8.1.11.15 FDIST

Die Funktion FDIST() gibt die f-Verteilung zurück.

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

FDIST(Zahl;Freiheitsgrade\_1;Freiheitsgrade\_2)

**Parameter**

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Freiheitsgrade 1, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Freiheitsgrade 2, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

**Beispiele**

FDIST(0,8;8;12) ergibt 0,61

**8.1.11.16 FINV**

Die Funktion FINV() ergibt die eindeutige, nicht negative Zahl X, sodass gilt  $FDIST(x;r1;r2) = p$ .

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

FINV(Zahl; r1; r2)

**Parameter**

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Zahl r1, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Zahl r2, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

**Beispiele**

FDIST(FINV(0,1;3;4);3;4) ergibt 0,1

**8.1.11.17 FISHER**

Die Funktion FISHER() ergibt die Fisher-Transformation für x und erzeugt eine der Normalverteilung ähnliche Funktion.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

FISHER(Zahl)

**Parameter**

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Beispiele**

FISHER(0,2859) ergibt 0,294096

**Beispiele**

FISHER(0,8105) ergibt 1,128485

**8.1.11.18 FISHERINV**

Die Funktion FISHERINV() ergibt die Inverse der Fisher-Transformation für x und erzeugt eine der Normalverteilung ähnliche Funktion.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

FISHERINV(Zahl)

**Parameter**

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Beispiele**

FISHERINV(0,2859) ergibt 0,278357

**Beispiele**

FISHERINV(0,8105) ergibt 0,669866

**8.1.11.19 FREQUENCY**

Zählt die Anzahl an Werten für jedes, durch die Grenzwerte im zweiten Parameter definierte, Intervall.

Die Werte des zweiten Parameters bestimmen die obere Grenze der Intervalle. Die Intervalle schließen die obere Grenze mit ein. Der ausgegebene Wert ist ein Spaltenvektor und hat ein Element mehr als der zweite Parameter. Das letzte Element repräsentiert die Anzahl aller Elemente, die größer sind als der letzte Wert im zweiten Parameter. Wenn der zweite Parameter nicht angegeben wird, werden alle Werte des ersten Parameters gezählt.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

FREQUENCY(Daten; Klassen)

**Parameter**

*Kommentar:* Fließkommawerte, die gezählt werden sollen., *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, die die oberen Grenzen der Intervalle angeben., *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

**8.1.11.20 GAMMADIST**

Die Funktion GAMMADIST() gibt die Gamma-Verteilung zurück.

Wenn der letzte Parameter (kumuliert) 0 ist, wird die Dichtefunktion berechnet. Wenn der Parameter 1 ist, wird die Verteilung berechnet.

Die ersten drei Parameter müssen positiv sein.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

GAMMADIST(Zahl; Alpha; Beta; kumuliert)

**Parameter**

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Alpha-Parameter, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Beta-Parameter, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Kumuliert-Schalter, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

**Beispiele**

GAMMADIST(0,758;0,1;0,35;1) ergibt 0,995450

**Beispiele**

GAMMADIST(0,758;0,1;0,35;0) ergibt 0,017179

#### 8.1.11.21 GAMMAINV

Die Funktion GAMMAINV() ergibt die eindeutige Zahl  $x \geq 0$ , sodass gilt  $\text{GAMMAINV}(x; \text{Alpha}; \text{Beta}; \text{WAHR}()) = p$ .

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

GAMMAINV(Zahl; Alpha; Beta)

##### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Alpha-Parameter, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Beta-Parameter, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

GAMMADIST(GAMMAINV(0,1;3;4);3;4) ergibt 0,1

##### Beispiele

GAMMADIST(GAMMAINV(0,3;3;4);3;4) ergibt 0,3

#### 8.1.11.22 GAMMALN

Die Funktion GAMMALN() gibt den natürlichen Logarithmus der Gamma-Funktion  $G(x)$  zurück. Der Parameter „Zahl“ muss positiv sein.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

GAMMALN(Zahl)

##### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

GAMMALN(2) ergibt 0

#### 8.1.11.23 GAUSS

Die Funktion GAUSS() berechnet die Integralwerte für die kumulative Standardnormalverteilung.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

GAUSS(Wert)

##### Parameter

*Kommentar:* Die Zahl, für die der Integralwert der Standardnormalverteilung berechnet werden soll., *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

GAUSS(0,25) ergibt 0,098706

#### 8.1.11.24 GEOMEAN

Die Funktion GEOMEAN() gibt das geometrische Mittel der angegebenen Argumente zurück. Dies entspricht der N-ten Wurzel des Produkts der Werte.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

GEOMEAN(Wert; Wert;...)

##### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

GEOMEAN(A1:A5)

##### Beispiele

GEOMEAN(21; 33; 54; 23) ergibt 30,45886

##### Ähnliche Funktionen

[HARMEAN](#)

#### 8.1.11.25 HARMEAN

Die Funktion HARMEAN() berechnet das harmonische Mittel von den N Datenpunkten (N dividiert durch die Summe des Inversen der Datenpunkte).

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

HARMEAN(Wert; Wert;...)

##### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

HARMEAN(A1:A5)

##### Beispiele

HARMEAN(21; 33; 54; 23) ergibt 28,588

##### Ähnliche Funktionen

[GEOMEAN](#)

#### 8.1.11.26 HYPGEOMDIST

Die Funktion HYPGEOMDIST() gibt die Hypergeometrische Verteilung zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

HYPGEOMDIST(x; n; M; N)

##### Parameter

*Kommentar:* Anzahl der Erfolge in der Probe, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Anzahl der Versuche, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Anzahl der Erfolge insgesamt, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Populationsgröße, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

HYPGEOMDIST(2; 5; 6; 20) ergibt 0,3522

#### 8.1.11.27 INTERCEPT

Die Funktion INTERCEPT() berechnet den Schnittpunkt der Regressionsgeraden mit der Y-Achse.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

INTERCEPT(y;x)

##### Parameter

*Kommentar:* Y-Werte (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* X-Werte (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### 8.1.11.28 INVBINO

Die Funktion INVBINO() ergibt die negative binomische Verteilung. Der erste Parameter ist die Zahl der Versuche, der zweite Parameter die Zahl der Misserfolge und die dritte Zahl ist die Wahrscheinlichkeit des Misserfolgs. Die Zahl der Versuche sollte größer sein als die Zahl der Misserfolge und die Wahrscheinlichkeit sollte kleiner oder gleich 1 sein.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

INVBINO(Versuche;Misserfolge;Wahrsch\_des\_Misserfolgs)

##### Parameter

*Kommentar:* Anzahl der Versuche, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Zahl der Misserfolge, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Wahrscheinlichkeit des Misserfolgs, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

INVBINO(12;3;0,2) ergibt 0,236223201

### 8.1.11.29 KURT

Die Funktion KURT() berechnet eine erwartungstreue Schätzung der Kurtosis eines Datensatzes. Sie müssen mindestens 4 Werte angeben, sonst wird ein Fehler zurückgegeben.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

KURT(Wert; Wert;...)

#### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

KURT(A1:A5)

#### Beispiele

KURT(21; 33; 54; 23) ergibt 1,344239

#### Ähnliche Funktionen

[KURTP](#)

### 8.1.11.30 KURTP

Die Funktion KURTP() berechnet eine Population-Kurtosis eines Datensatzes. Sie müssen mindestens 4 Werte angeben, sonst wird ein Fehler zurückgegeben.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

KURTP(Wert; Wert;...)

#### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

KURTP(A1:A5)

#### Beispiele

KURTP(21; 33; 54; 23) ergibt -1,021

#### Ähnliche Funktionen

[KURT](#)

### 8.1.11.31 LARGE

Die Funktion LARGE() gibt die „k“ größte Zahl aus dem Datensatz zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

LARGE(Bereich; k)

#### Parameter

*Kommentar:* Zellbereich von Werten, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Position (von dem größten), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

A1: 3, A2: 1, A3: 5 => LARGE(A1:A3; 2) ergibt 3

### 8.1.11.32 LEGACYFDIST

Die Funktion LEGACYFDIST() gibt die f-Verteilung zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

LEGACYFDIST(Zahl;Freiheitsgrade\_1;Freiheitsgrade\_2)

#### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Freiheitsgrade 1, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Freiheitsgrade 2, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

LEGACYFDIST(0,8;8;12) ergibt 0,61

### 8.1.11.33 LOGINV

Die Funktion LOGINV() berechnet das Inverse der lognormal kumulativen Verteilung.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

LOGINV(p; Mitte; Standardverteilung)

#### Parameter

*Kommentar:* Wahrscheinlichkeit, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Mittelwert der logarithmischen Standardverteilung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Standardabweichung der logarithmischen Standardverteilung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

LOGINV(0,1;0;1) ergibt 0,2776

#### 8.1.11.34 LOGNORMDIST

Die Funktion LOGNORMDIST() gibt die kumulative lognormale Verteilung zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

LOGNORMDIST(Number;MV;STD)

##### Parameter

*Kommentar:* Wahrscheinlichkeitswert, für den die logarithmische Standardverteilung berechnet werden soll, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Mittelwert der logarithmischen Standardverteilung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Standardabweichung der logarithmischen Standardverteilung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

LOGNORMDIST(0,1;0;1) ergibt 0,01

#### 8.1.11.35 MEDIAN

Die Funktion MEDIAN() berechnet den Median von allen Werten, die als Parameter angegeben wurden. Sie können den Median eines Zellbereichs MEDIAN(A1:B5) oder einer Werteliste MEDIAN(12; 5; 12,5) berechnen lassen. Leere Zellen werden wie Zellen, die eine 0 enthalten, behandelt. Zellen, die Text enthalten, werden ignoriert.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

MEDIAN(Wert;Wert;...)

##### Parameter

*Kommentar:* Fließkommawert oder Wertbereich, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte oder Wertbereich, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte oder Wertbereich, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte oder Wertbereich, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte oder Wertbereich, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

##### Beispiele

MEDIAN(12; 5; 5,5) ergibt 5,5

##### Beispiele

MEDIAN(12; 7; 8;2) ergibt 7,5

### 8.1.11.36 MODE

Die Funktion MODE() gibt die am häufigsten vorkommende Zahl in einem Satz von Daten zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

MODE(Zahl1; Zahl2; ...)

#### Parameter

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

MODE(12; 14; 12; 15) ergibt 12

### 8.1.11.37 NEGBINOMDIST

Die Funktion NEGBINOMDIST() gibt die negative binomische Verteilung zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

NEGBINOMDIST(Fehler; Erfolge; Wahrscheinlichkeit des Erfolges)

#### Parameter

*Kommentar:* Zahl der Misserfolge, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Anzahl erfolgreicher Versuche, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Wahrscheinlichkeit des Erfolges, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

NEGBINOMDIST(2;5;0,55) ergibt 0,152872629

### 8.1.11.38 NORMDIST

Die Funktion NORMDIST() gibt die normale kumulative Verteilung zurück.

Die Zahl für die Verteilung auf welcher die Standardnormalverteilung berechnet werden soll.

MV ist die lineare Mitte der Verteilung.

STD ist die Standardabweichung der Verteilung.

K = 0 berechnet die dichte Funktion, K = 1 berechnet die Verteilung.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

NORMDIST(Zahl;MV;STD;K)

#### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Lineare Mitte der Verteilung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Standardabweichung der Verteilung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* 0 = Dichte, 1 = Verteilung, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

NORMDIST(0,859;0,6;0,258;0) ergibt 0,934236

### Beispiele

NORMDIST(0,859;0,6;0,258;1) ergibt 0,842281

#### 8.1.11.39 NORMINV

Die Funktion NORMINV() ergibt die Inverse der kumulativen Normalverteilung. Die Zahl muss zwischen 0 und 1 liegen (exklusive 0 und 1 selbst) und STD muss positiv sein.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

NORMINV(Zahl;MV;STD)

#### Parameter

*Kommentar:* Wahrscheinlichkeitswert, für den die logarithmische Standardverteilung berechnet werden soll, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Mittelwert der Normalverteilung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Standardabweichung der Normalverteilung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

NORMINV(0,9;63;5) ergibt 69,41

#### 8.1.11.40 NORMSDIST

Die Funktion NORMSDIST() gibt die standard normal Verteilung zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

NORMSDIST(Zahl)

#### Parameter

*Kommentar:* Wert für welche die Standardnormalverteilung berechnet werden soll, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

NORMSDIST(1) ergibt 0,84

#### 8.1.11.41 NORMSINV

Die Funktion NORMSINV() ergibt die Inverse der kumulativen Standardnormalverteilung. Die Zahl muss zwischen 0 und 1 liegen (jedoch exklusive 0 und 1 selbst).

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

NORMSINV(Zahl)

#### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

NORMSINV(0,908789) ergibt 1,3333

#### 8.1.11.42 PEARSON

Die Funktion PEARSON() berechnet den Korrelationskoeffizienten von zwei Zellbereichen. Sie ist identisch mit der Funktion CORREL.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

PEARSON(Bereich1; Bereich2)

##### Parameter

*Kommentar:* Zellbereich von Werten, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Zweiter Zellbereich von Werten, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

PEARSON(A1:A3; B1:B3)

##### Ähnliche Funktionen

[CORREL](#)

#### 8.1.11.43 PERCENTILE

Die Funktion PERCENTILE() gibt das Perzentil oder die Überschreitungswahrscheinlichkeit von Datenwerten in Daten zurück. Ein Perzentil oder eine Überschreitungswahrscheinlichkeit ist der skalierte Wert einer Datenreihe vom kleinsten Wert (alpha=0) zum größten Wert (alpha=1) der Datenreihe. Alpha = 25 % ist das Quartil oder der Viertelwert. Alpha = 50 % ist der Median oder Mittelwert. Leere Zeilen werden als Null verarbeitet und Zellen mit Text werden ignoriert.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

PERCENTILE(daten;alpha)

##### Parameter

*Kommentar:* Wertebereich, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Der Prozentwert zwischen 0 und 1 einschließlich dieser Werte., *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Ähnliche Funktionen

[MEDIAN](#)

#### 8.1.11.44 PERMUT

Die PERMUT() Funktion ergibt die Anzahl der Permutationen. Der erste Parameter ist die Anzahl der Elemente und der zweite Parameter ist die Anzahl in der Permutation verwendeten Elemente.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

PERMUT(gesamt;verwendet)

##### Parameter

*Kommentar:* Anzahl der Elemente, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Anzahl der Elemente zum Permutieren, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

PERMUT(8;5) ergibt 6720

### Beispiele

PERMUT(1;1) ergibt 1

#### 8.1.11.45 PERMUTATIONA

Die PERMUTATIONA() Funktion ergibt die Anzahl der geordneten Permutationen, wenn Wiederholungen erlaubt sind. Der erste Parameter ist die Anzahl der Elemente und der zweite Parameter ist die Anzahl in der Permutation auszuwählenden Elemente. Beide Parameter müssen positiv sein.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

### Syntax

PERMUTATIONA(gesamt;gewählt)

### Parameter

*Kommentar:* Anzahl der Elemente, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Anzahl der auszuwählenden Elemente, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

PERMUTATIONA(2,3) ergibt 8

### Beispiele

PERMUTATIONA(0,0) ergibt 1

#### 8.1.11.46 PHI

Die Funktion PHI() berechnet die Verteilungsfunktion für die Standardnormalverteilung.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

PHI(Wert)

### Parameter

*Kommentar:* Die Zahl, für die die Standardnormalverteilung berechnet werden soll., *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

PHI(0,25) ergibt 0,386668

#### 8.1.11.47 POISSON

Die Funktion POISSON() gibt die Poisson-Verteilung zurück.

Der „Lambda“ und der „Zahl“ Parameter müssen positiv sein.

Kumulativ = 0 berechnet die Dichtefunktion, Kumulativ = 1 berechnet die Verteilung.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

POISSON(Zahl;lambda;kumulativ)

### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Lambda-Parameter (Mittelwert), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* 0 = Dichte, 1 = Verteilung, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

POISSON(60;50;0) ergibt 0,020105

### Beispiele

POISSON(60;50;1) ergibt 0,927840

## 8.1.11.48 RANK

Die Funktion RANK() ergibt den Rang, den eine Zahl innerhalb einer Liste von Zahlen einnimmt.

Die Reihenfolge gibt an, wie der Rang der Zahlen bestimmt werden soll:

Wenn die Reihenfolge 0 oder nicht angegeben ist, werden die Daten in absteigender Reihenfolge ermittelt.

Wenn die Reihenfolge ungleich 0 ist, werden die Daten in aufsteigender Reihenfolge ermittelt.

*Rückgabetyyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

RSQ(Wert; Daten; Reihenfolge)

### Parameter

*Kommentar:* Wert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Daten (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Reihenfolge, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

RANK (2;{1;2;3}) ergibt 2

## 8.1.11.49 RSQ

Die Funktion RSQ() ergibt das Quadrat des Pearsonschen Produktmomentkorrelationskoeffizienten entsprechend Datenpunkten in bekannten X- und Y-Werten.

Wenn „BekanntesY“ und „BekanntesX“ leer sind oder eine unterschiedliche Anzahl von Datenpunkten haben, wird #N/A zurückgegeben.

*Rückgabetyyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

RSQ(BekanntesY; BekanntesX)

### Parameter

*Kommentar:* BekanntesY (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* BekanntesX (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### 8.1.11.50 SKEW

Die Funktion SKEW() gibt die voraussichtliche Schräge einer Verteilung zurück

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SKEW(Zahl1; Zahl2; ...)

##### Parameter

*Kommentar*: Fließkommawert, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Fließkommawert, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Fließkommawert, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Fließkommawert, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

SKEW(11,4; 17,3; 21,3; 25,9; 40,1) ergibt 0,9768

##### Ähnliche Funktionen

[SKEWP](#)

#### 8.1.11.51 SKEWP

Die Funktion SKEWP() gibt die Grundschräge einer Verteilung zurück

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SKEWP(Zahl1; Zahl2; ...)

##### Parameter

*Kommentar*: Fließkommawert, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Fließkommawert, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Fließkommawert, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: Fließkommawert, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

SKEWP(11,4; 17,3; 21,3; 25,9; 40,1) ergibt 0,6552

##### Ähnliche Funktionen

[SKEW](#)

#### 8.1.11.52 SLOPE

Die Funktion SLOPE() berechnet die Steigung der Regressionsgeraden.

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SLOPE(y;x)

##### Parameter

*Kommentar*: Y-Werte (Feld), *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar*: X-Werte (Feld), *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### 8.1.11.53 SMALL

Die Funktion SMALL() gibt den „k“ kleinsten Wert von dem Datensatz zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SMALL(Bereich; k)

##### Parameter

*Kommentar:* Zellbereich von Werten, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Position (von dem kleinsten Wert), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

A1: 3, A2: 1, A3: 5 => SMALL(A1:A3; 1) ergibt 1

#### 8.1.11.54 STANDARDIZE

Die Funktion STANDARDIZE() berechnet einen normalisierten Wert.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

STANDARDIZE(x; Mittel, Standardabweichung)

##### Parameter

*Kommentar:* Zahl die normalisiert werden soll, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Der Mittelwert der Verteilung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Standardabweichung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

STANDARDIZE(4; 3; 7) ergibt 0,1429

#### 8.1.11.55 STDEV

Die Funktion STDEV() gibt die geschätzte Standardabweichung basierend auf einer Probe zurück. Die Standardabweichung ist ein Maß wie weit die Werte vom Mittelwert gestreut sind.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

STDEV(Wert;Wert;...)

##### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

##### Beispiele

STDEV(6;7;8) ergibt 1

#### Ähnliche Funktionen

[STDEVP](#)

#### 8.1.11.56 STDEVA

Die Funktion STDEVA() gibt die geschätzte Standardabweichung basierend auf einer Probe zurück. Die Standardabweichung ist ein Maß wie weit Werte vom Mittelwert gestreut sind. Wenn eine referenzierte Zelle Text oder den booleschen Wert Falsch enthält, wird er als 0 gezählt. Enthält eine Zelle den booleschen Wert Wahr, wird sie als 1 gezählt.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

STDEVA(Wert;Wert;...)

##### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

##### Beispiele

STDEVA(6; 7; A1; 8) ergibt 1, wenn A1 leer ist

##### Beispiele

STDEVA(6; 7; A1; 8) ergibt 3,109, wenn A1 Wahr ist

##### Ähnliche Funktionen

[STDEV](#)

[STDEVP](#)

#### 8.1.11.57 STDEVP

Die Funktion STDEVP() berechnet die Abweichung basierend auf der gesamten Population.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

STDEVP(Wert;Wert;...)

##### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

##### Beispiele

STDEVP(6;7;8) ergibt 0,816497...

##### Ähnliche Funktionen

[STDEV](#)

#### 8.1.11.58 STDEVPA

Die Funktion STDEVPA() berechnet die Standardabweichung basierend auf der ganzen Population. Wenn eine referenzierte Zelle Text oder den booleschen Wert Falsch enthält, wird sie als 0 gezählt. Enthält die Zelle den booleschen Wert Wahr wird sie als 1 gezählt.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

STDEVPA(Wert;Wert;...)

##### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

##### Beispiele

STDEVPA(6; 7; A1; 8) ergibt 0,816497..., wenn A1 leer ist

##### Beispiele

STDEVPA(6; 7; A1; 8) ergibt 2,69..., wenn A1 Wahr ist.

##### Beispiele

STDEVPA(6; 7; A1; 8) ergibt 3,11..., wenn A1 Falsch ist

##### Ähnliche Funktionen

[STDEV](#)

[STDEVP](#)

#### 8.1.11.59 STEYX

Die Funktion STEYX() berechnet den Standardfehler des geschätzten Y-Wertes für jeden X-Wert der Regression.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SLOPE(y;x)

##### Parameter

*Kommentar:* Y-Werte (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* X-Werte (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### 8.1.11.60 SUM2XMY

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SLOPE(y;x)

##### Parameter

#### 8.1.11.61 SUMPRODUCT

Die Funktion SUMPRODUCT() gibt die Summe der Produkte (SUM(X\*Y)) der angegebenen Werte zurück. Die Anzahl der Werte in beiden Feldern sollte gleich sein, sonst gibt die Funktion „Err“ zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SUMPRODUCT(Feld1;Feld2)

##### Parameter

*Kommentar:* Wert (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Wert (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

SUMPRODUCT(A1:A2;B1:B2) mit A1=2 A2=5 B1=3 und B2=5 ergibt 31

#### 8.1.11.62 SUMX2MY2

Die Funktion SUMX2MY2() gibt die Differenz des Quadrats (SUM(X<sup>2</sup>-Y<sup>2</sup>)) der angegebenen Werte zurück. Die Anzahl der Werte beider Felder muss gleich sein, sonst gibt die Funktion „Err“ zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SUMX2MY2(Feld1;Feld2)

##### Parameter

*Kommentar:* Wert (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Wert (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

SUMX2MY2(A1:A2;B1:B2) mit A1=2 A2=5 B1=3 und B2=5 ergibt -5

#### 8.1.11.63 SUMX2PY2

Die Funktion SUMX2PY2() gibt die Summe der Quadrate (SUM(X<sup>2</sup>+Y<sup>2</sup>)) der angegebenen Werte zurück. Die Anzahl der Elemente jedes Feldes sollte gleich sein, sonst gibt die Funktion „Err“ zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SUMX2PY2(Feld1;Feld2)

##### Parameter

*Kommentar:* Wert (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Wert (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

SUMX2PY2(A1:A2;B1:B2) mit A1=2 A2=5 B1=3 und B2=5 ergibt 63

#### 8.1.11.64 SUMXMY2

Die Funktion SUMXMY2() gibt das Quadrat der Differenz ( $\text{SUM}((X-Y)^2)$ ) der angegebenen Werte zurück. Die Anzahl der Werte in den beiden Feldern muss gleich sein, da die Funktion sonst einen Fehler zurückgibt.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

SUMXMY2(Feld1;Feld2)

##### Parameter

*Kommentar:* Wert (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Wert (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Beispiele

SUMXMY2(A1:A2;B1:B2) mit A1=2, A2=5, B1=3 und B2=5 ergibt 1

#### 8.1.11.65 TDIST

Die Funktion TDIST() gibt die t-Verteilung zurück.

Modus = 1 gibt den einseitigen Test zurück, Modus = 2 gibt den zweiseitigen Test zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

TDIST(Zahl;Freiheitsgrade;Modus)

##### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Freiheitsgrade für die t-Verteilung, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Modus (1 oder 2), *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

TDIST(12;5;1) ergibt 0,000035

#### 8.1.11.66 TREND

Die Funktion TREND() berechnet eine Folge von Werten basierend auf einer linearen Regression von bekannten Wertepaaren.

Vorgabe: COUNT(BekanntesY) = COUNT(BekanntesX).

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

TREND(BekanntesY[;BekanntesX[;NeuesX[;Konstante = WAHR]])

##### Parameter

*Kommentar:* BekanntesY, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* BekanntesX, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* NeuesX, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Konstante, *Typ:* Wahrheitswert (WAHR oder FALSCH)

#### 8.1.11.67 TRIMMEAN

Die Funktion TRIMMEAN() berechnet den Mittelwert eines Datensatzes unter Auslassung der durch den Grenzbruch festgelegten Randwerte.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

TRIMMEAN(Datensatz; Grenzbruch)

##### Parameter

*Kommentar:* Datensatz, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Grenzbruch, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### 8.1.11.68 TTEST

Die Funktion TTEST() berechnet die Wahrscheinlichkeit eines t-Tests.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

TTEST(x; y; Typ; Modus)

##### Parameter

*Kommentar:* x (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* y (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Typ, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Modus, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### 8.1.11.69 VAR

Die Funktion VAR() berechnet die geschätzte Varianz, basierend auf einer Probe.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

##### Syntax

VAR(Wert;Wert;...)

##### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

##### Beispiele

VAR(12;5;7) ergibt 13

##### Beispiele

VAR(15;80;3) ergibt 1716,333...

##### Beispiele

VAR(6;7;8) ergibt 1

##### Ähnliche Funktionen

VARIANCE

VARA

VARP

VARPA

### 8.1.11.70 VARA

Die Funktion VARA() berechnet die Varianz basierend auf einer Probe.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

VARA(Wert;Wert;...)

#### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

#### Beispiele

VARA(12;5;7) ergibt 13

#### Beispiele

VARA(15;80;3) ergibt 1716,333...

#### Beispiele

VARA(6;7;8) ergibt 1

#### Ähnliche Funktionen

[VAR](#)

[VARP](#)

[VARPA](#)

### 8.1.11.71 VARIANCE

Die Funktion VARIANCE() berechnet die geschätzte Varianz, basierend auf einer Probe. Sie ist mit der Funktion VAR identisch.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

VARIANCE(Wert;Wert;...)

#### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

#### Beispiele

VARIANCE(12;5;7) ergibt 13

#### Beispiele

VARIANCE(15;80;3) ergibt 1716,333...

#### Beispiele

VARIANCE(6;7;8) ergibt 1

## Ähnliche Funktionen

VAR  
VARA  
VARP  
VARPA

### 8.1.11.72 VARP

Die Funktion VARP() berechnet die Varianz basierend auf der gesamten Population.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

VARP(Wert;Wert;...)

#### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

#### Beispiele

VARP(12;5;7) ergibt 8,666...

#### Beispiele

VARP(15;80;3) ergibt 1144,22...

#### Beispiele

VARP(6;7;8) ergibt 0,6666667...

## Ähnliche Funktionen

VAR  
VARA  
VARPA

### 8.1.11.73 VARPA

Die Funktion VARPA() berechnet die Varianz basierend auf der gesamten Population. Text und boolesche Werte, die Falsch ergeben, werden als 0 gewertet, boolesche Werte, die Wahr ergeben, werden als 1 gewertet.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

VARPA(Wert;Wert;...)

#### Parameter

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

*Kommentar:* Fließkommawerte, *Typ:* Eine Folge von Fließkommawerten (wie 1,3; 0,343; 253)

### Beispiele

VARPA(12;5;7) ergibt 8,666...

### Beispiele

VARPA(15;80;3) ergibt 1144,22...

### Beispiele

VARPA(6;7;8) ergibt 0,6666667...

### Ähnliche Funktionen

VAR  
VARA  
VARP

#### 8.1.11.74 WEIBULL

Die Funktion WEIBULL() gibt die Weibull-Verteilung zurück.

Die Parameter Alpha und Beta müssen positiv sein, der erste Parameter Zahl darf nicht negativ sein.

Kumulativ = 0 berechnet die Dichtefunktion, Kumulativ = 1 berechnet die Verteilung.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

WEIBULL(Zahl;Alpha;Beta;kumulativ)

### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Alpha-Parameter, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Beta-Parameter, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* 0 = Dichte, 1 = Verteilung, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

WEIBULL(2;1;1;0) ergibt 0,135335

### Beispiele

WEIBULL(2;1;1;1) ergibt 0,864665

#### 8.1.11.75 ZTEST

Die Funktion ZTEST() berechnet die zweiseitige Wahrscheinlichkeit eines z-Tests mit Normalverteilung.

Führt einen Test der Null-Hypothese durch, dass x ein Beispiel einer normal verteilten Zufallsprobe mit der dem angegebenen Mittelwert und der angegebenen Standardabweichung. Ein Rückgabewert von 1 bedeutet, das die Null-Hypothese zurückgewiesen wurde, d. h. x ist keine zufällige Probe einer Normalverteilung. Wenn die Standardabweichung nicht angegeben ist, wird sie mit der Funktion STDEV aus der Probe x ermittelt.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Syntax

ZTEST(x; Mittelwert; Standardabweichung)

### Parameter

*Kommentar:* x (Feld), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Mittelwert, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Standardabweichung, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)



#### 8.1.12.4 CLEAN

Die Funktion CLEAN() entfernt alle nicht druckbaren Zeichen aus einer Zeichenfolge.

*Rückgabotyp:* Text

##### Syntax

CLEAN(Text)

##### Parameter

*Kommentar:* Quell-Zeichenfolge, *Typ:* Text

##### Beispiele

CLEAN(AsciiToChar(7) + "HALLO") ergibt "HALLO"

#### 8.1.12.5 CODE

Die Funktion CODE() gibt eine Zahl für das erste Zeichen in einer Zeichenfolge zurück. Die Zuordnung wird anhand des sogenannten ASCII-Zeichensatzes vorgenommen.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

CODE(Text)

##### Parameter

*Kommentar:* Text, *Typ:* Text

##### Beispiele

CODE("KDE") ergibt 75

##### Ähnliche Funktionen

[CHAR](#)

#### 8.1.12.6 COMPARE

Die Funktion COMPARE() gibt 0 zurück, wenn die beiden Zeichenfolgen übereinstimmen. Sie ergibt -1, wenn die erste bei alphabetischer Sortierung vor der zweiten kommt. Ansonsten ergibt sie 1.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

##### Syntax

COMPARE(Zeichenfolge1; Zeichenfolge2; Wahr | Falsch)

##### Parameter

*Kommentar:* Erste Zeichenfolge, *Typ:* Text

*Kommentar:* Zeichenfolge, mit der verglichen wird, *Typ:* Text

*Kommentar:* Vergleiche mit Beachtung der Groß-/Kleinschreibung (wahr/falsch), *Typ:* Wahrheitswert (WAHR oder FALSCH)

##### Beispiele

COMPARE("Calligra"; "Calligra"; true) ergibt 0

### Beispiele

COMPARE("calligra"; "Calligra"; true) ergibt 1

### Beispiele

COMPARE("kspread"; "Calligra"; false) ergibt 1

### Ähnliche Funktionen

[EXACT](#)

#### 8.1.12.7 CONCATENATE

Die Funktion CONCATENATE() ergibt eine Zeichenfolge, welche durch Aneinander hängen aller als Parameter übergebenen Zeichenfolgen entsteht.

*Rückgabotyp:* Text

### Syntax

CONCATENATE(Wert;Wert;...)

### Parameter

*Kommentar:* Zeichenfolgen, *Typ:* Eine Zeichenfolge

### Beispiele

CONCATENATE("Sheets";"Calligra";"KDE") ergibt "SheetsCalligraKDE"

#### 8.1.12.8 DOLLAR

Die Funktion DOLLAR() konvertiert eine Zahl unter Verwendung des Währungsformates in eine Zeichenfolge. Die Nachkommastellen werden entsprechend gerundet. Obwohl der Name der Funktion DOLLAR ist, wird in die aktuell eingestellte Landeswährung umgewandelt.

*Rückgabotyp:* Text

### Syntax

DOLLAR(Zahl;Dezimalstellen)

### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Dezimalstellen, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

DOLLAR(1403,77) ergibt „EUR 1 403,77“

### Beispiele

DOLLAR(-0,123;4) ergibt „EUR -0,1230“

### 8.1.12.9 EXACT

Die Funktion EXACT() gibt Wahr zurück, wenn die beiden Zeichenfolgen übereinstimmen, ansonsten Falsch.

*Rückgabetyp:* Wahrheitswert (WAHR oder FALSCH)

#### Syntax

EXACT(Zeichenfolge1;Zeichenfolge2)

#### Parameter

*Kommentar:* Zeichenfolge, *Typ:* Text

*Kommentar:* Zeichenfolge, *Typ:* Text

#### Beispiele

EXACT("Calligra";"Calligra") ergibt Wahr

#### Beispiele

EXACT("KSpread";"Calligra") ergibt False

#### Ähnliche Funktionen

[COMPARE](#)

### 8.1.12.10 FIND

Die Funktion FIND() sucht eine Zeichenfolge (find\_text) in einer anderen Zeichenfolge (within\_text) und gibt die Position des ersten Zeichens von „find\_text“ in „within\_text“ an.

Der Parameter „start\_num“ gibt das Zeichen an, bei dem die Suche gestartet werden soll. Das erste Zeichen hat die Position 1. Wenn „start\_num“ nicht gegeben ist, wird ein Wert von 1 dafür angenommen.

Sie können auch die Funktion SEARCH() verwenden, aber im Gegensatz zu SEARCH() beachtet FIND() Groß/Kleinschreibung und erlaubt keine Platzhalter.

*Rückgabetyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

FIND(find\_text;within\_text;start\_num)

#### Parameter

*Kommentar:* Der zu suchende Text, *Typ:* Text

*Kommentar:* Der Text, in dem gesucht werden soll, *Typ:* Text

*Kommentar:* Gibt den Index an, an dem die Suche beginnt, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

FIND("Cal";"Calligra") ergibt 1

#### Beispiele

FIND("i";"Calligra") ergibt 5

#### Beispiele

FIND("a";"Sheets in Calligra";4) ergibt 12

#### Ähnliche Funktionen

[FINDB](#)

[SEARCH](#)

[REPLACE](#)

[SEARCHB](#)

[REPLACEB](#)

#### 8.1.12.11 FINDB

Die Funktion FINDB() sucht eine Zeichenfolge (find\_text) in einer anderen Zeichenfolge (within\_text) und gibt die Position des ersten Zeichens von „find\_text“ in „within\_text“ an unter Verwendung der Byteposition.

Der Parameter „BytePosition“ gibt das Zeichen an, bei dem die Suche gestartet werden soll. Das erste Zeichen hat die Position 2. Wenn „BytePosition“ nicht gegeben ist, wird ein Wert von 2 dafür angenommen.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

FINDB(find\_text;within\_text;BytePosition Start)

#### Parameter

*Kommentar:* Der zu suchende Text, *Typ:* Text

*Kommentar:* Der Text, in dem gesucht werden soll, *Typ:* Text

*Kommentar:* Gibt die Byteposition an, an der die Suche beginnt, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Ähnliche Funktionen

[FIND](#)  
[SEARCH](#)  
[REPLACE](#)  
[SEARCHB](#)  
[REPLACEB](#)

#### 8.1.12.12 FIXED

Die Funktion FIXED() rundet eine Zahl auf die angegebene Anzahl von Dezimalstellen, formatiert die Zahl entsprechend der Ländereinstellung für Dezimalzahlen und gibt das Ergebnis als Text aus. Wenn „Dezimalstellen“ negativ ist, wird entsprechend links vom Dezimalpunkt gerundet. Wenn Sie „Dezimalstellen“ weglassen, wird 2 als Wert angenommen. Wenn der optionale Parameter Kein\_Tausendertrennzeichen Wahr ist, werden keine Tausender-Trennzeichen angezeigt.

*Rückgabotyp:* Text

#### Syntax

FIXED(Zahl;Dezimalstellen;Kein\_Tausendertrennzeichen)

#### Parameter

*Kommentar:* Zahl, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Dezimalstellen, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Kein\_Tausendertrennzeichen, *Typ:* Wahrheitswert (WAHR oder FALSCH)

#### Beispiele

FIXED(1234,567;1) ergibt „1 234,6“

#### Beispiele

FIXED(1234,567;1;Wahr) ergibt „1234,6“

#### Beispiele

FIXED(44,332) ergibt „44,33“

### 8.1.12.13 JIS

Die Funktion JIS() gibt die, dem Argument halber Breite entsprechenden, Zeichen voller Breite zurück.

*Rückgabetyp:* Text

#### Syntax

JIS(Text)

#### Parameter

*Kommentar:* Zeichen halber Breite, *Typ:* Text

#### Ähnliche Funktionen

ASC

### 8.1.12.14 LEFT

Die Funktion LEFT() ergibt eine Zeichenfolge, die der unter „Länge“ festgelegte Anzahl von Zeichen vom Anfang des angegebenen Textes entspricht. Wenn der Wert „Länge“ die Gesamtlänge der Zeichenfolge übersteigt, wird die ganze Zeichenfolge zurückgegeben. Negative Werte für „Länge“ führen zu einem Fehler.

*Rückgabetyp:* Text

#### Syntax

LEFT(Text;Länge)

#### Parameter

*Kommentar:* Quell-Zeichenfolge, *Typ:* Text

*Kommentar:* Anzahl an Zeichen, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

LEFT("Hallo";2) ergibt "Ha"

#### Beispiele

LEFT("KSpread";10) ergibt "KSpread"

#### Beispiele

LEFT("KSpread") ergibt "K"

#### Ähnliche Funktionen

RIGHT

MID

RIGHTB

MIDB

### 8.1.12.15 LEFTB

Die Funktion LEFTB() ergibt eine Zeichenfolge, die der unter „Länge“ festgelegte Anzahl von Zeichen vom Anfang des angegebenen Textes unter Verwendung der Byteposition entspricht. Wenn der Wert „Länge“ die Gesamtlänge der Zeichenfolge übersteigt, wird die ganze Zeichenfolge zurückgegeben. Negative Werte für „Länge“ führen zu einem Fehler.

*Rückgabetyp:* Text

### Syntax

LEFTB(Text;ByteLänge)

### Parameter

*Kommentar:* Quell-Zeichenfolge, *Typ:* Text

*Kommentar:* Bytelänge, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Ähnliche Funktionen

[RIGHT](#)

[MID](#)

[RIGHTB](#)

[MIDB](#)

#### 8.1.12.16 LEN

Die Funktion LEN() ergibt die Länge der übergebenen Zeichenfolge.

*Rückgabebetyp:* Ganze Zahl (z. B. 1, 132, 2344)

### Syntax

LEN(Text)

### Parameter

*Kommentar:* Zeichenfolge, *Typ:* Text

### Beispiele

LEN("Hallo") ergibt 5

### Beispiele

LEN("KSpread") ergibt 7

### Ähnliche Funktionen

[LENB](#)

#### 8.1.12.17 LENB

Die Funktion LENB() ergibt die Länge der übergebenen Zeichenfolge unter Verwendung der Byteposition.

*Rückgabebetyp:* Ganze Zahl (z. B. 1, 132, 2344)

### Syntax

LENB(Text)

### Parameter

*Kommentar:* Zeichenfolge, *Typ:* Text

#### 8.1.12.18 LOWER

Die Funktion LOWER() konvertiert eine Zeichenfolge in Kleinbuchstaben.

*Rückgabebetyp:* Text

##### Syntax

LOWER(text)

##### Parameter

*Kommentar:* Quell-Zeichenfolge, *Typ:* Text

##### Beispiele

LOWER("hallo") ergibt "hallo"

##### Beispiele

LOWER("HALLO") ergibt "hallo"

##### Ähnliche Funktionen

UPPER  
TOGGLE

#### 8.1.12.19 MID

Die Funktion MID() ergibt eine Teil-Zeichenfolge, bestehend aus einer bestimmten Anzahl Zeichen ab der Indexposition

*Rückgabebetyp:* Text

##### Syntax

MID(Text;Position;Länge)

##### Parameter

*Kommentar:* Quell-Zeichenfolge, *Typ:* Text

*Kommentar:* Position, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Länge, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

MID("Calligra";2;3) ergibt "all"

##### Beispiele

MID("Calligra";2) ergibt "alligra"

##### Ähnliche Funktionen

LEFT  
RIGHT  
LEFTB  
RIGHTB  
MIDB

#### 8.1.12.20 MIDB

Die Funktion MIDB() ergibt eine Teil-Zeichenfolge, bestehend aus einer bestimmten Anzahl Zeichen ab der Indexposition unter Verwendung der Byteposition,

*Rückgabetyp:* Text

##### Syntax

MIDB(Text;Byteposition Start;Bytelänge)

##### Parameter

*Kommentar:* Quell-Zeichenfolge, *Typ:* Text

*Kommentar:* Byteposition, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Bytelänge, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Ähnliche Funktionen

LEFT

RIGHT

LEFTB

RIGHTB

MID

#### 8.1.12.21 PROPER

Die Funktion PROPER() wandelt den ersten Buchstaben jedes Wortes in einen Großbuchstaben und die restlichen Buchstaben in Kleinbuchstaben um.

*Rückgabetyp:* Text

##### Syntax

PROPER(Zeichenfolge)

##### Parameter

*Kommentar:* Zeichenfolge, *Typ:* Text

##### Beispiele

PROPER("das ist ein titel") ergibt "Das Ist Ein Titel"

#### 8.1.12.22 REGEXP

Liefert den Teil einer Zeichenfolge, die einen regulären Ausdruck erfüllt. Erfüllt die Zeichenfolge den regulären Ausdruck nicht, wird der angegebene Standardwert zurückgeliefert.

Ist ein Rückverweis angegeben, wird der Wert des Rückverweises zurückgeliefert.

Ist kein Standardwert angegeben, wird eine leere Zeichenfolge angenommen. Wird kein Rückverweis angegeben, wird 0 angenommen (sodass der Bereich der Zeichenfolge, der den regulären Ausdruck erfüllt, vollständig zurückgeliefert wird).

*Rückgabetyp:* Text

##### Syntax

REGEXP(Zeichenfolge, regulärer Ausdruck, Standardwert, Rückverweis)

### Parameter

*Kommentar:* Suchtext, *Typ:* Text  
*Kommentar:* Regulärer Ausdruck, *Typ:* Text  
*Kommentar:* Standardwert (optional), *Typ:* Text  
*Kommentar:* Rückverweis (optional), *Typ:* Zahl

### Beispiele

REGEXP("Anzahl ist 15."; "[0-9]+") = "15"

### Beispiele

REGEXP("15, 20, 26, 41"; "([0-9]+), \*[0-9]+\$"; ";"; 1) = "26"

### 8.1.12.23 REGEXPRE

Ersetzt alle Vorkommen eines regulären Ausdrucks mit dem Ersetzungstext.

*Rückgabetyt:* Text

### Syntax

REGEXPRE(Text, regulärer Ausdruck, Ersetzungstext)

### Parameter

*Kommentar:* Suchtext, *Typ:* Text  
*Kommentar:* Regulärer Ausdruck, *Typ:* Text  
*Kommentar:* Ersetzung, *Typ:* Text

### Beispiele

REGEXPRE(„14 und 15 und 16“; „[0-9]+“; „Zahl“) liefert „Zahl und Zahl und Zahl“

### 8.1.12.24 REPLACE

Die Funktion REPLACE() ersetzt einen Teil einer Zeichenfolge durch eine andere Zeichenfolge.

*Rückgabetyt:* Text

### Syntax

REPLACE(text;position;laenge;neuer\_text)

### Parameter

*Kommentar:* Text, in dem einige Zeichen ersetzt werden sollen, *Typ:* Text  
*Kommentar:* Position des ersten zu ersetzenden Zeichens, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)  
*Kommentar:* Anzahl der zu ersetzenden Zeichen, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)  
*Kommentar:* Text, der die Zeichen im alten Text ersetzen wird, *Typ:* Text

### Beispiele

REPLACE("abcdefghijk";6;5;"-") ergibt "abcde-k"

### Beispiele

REPLACE("2002";3;2;"03") ergibt "2003"

### Ähnliche Funktionen

FIND  
MID  
FINDB  
MIDB

#### 8.1.12.25 REPLACEB

Die Funktion REPLACEB() ersetzt einen Teil einer Zeichenfolge durch eine andere Zeichenfolge unter Verwendung der Byteposition.

*Rückgabebetyp:* Text

##### Syntax

REPLACEB(text;BytePosition;ByteLänge Län;neuer\_text)

##### Parameter

*Kommentar:* Text, in dem einige Zeichen unter Verwendung der Byteposition ersetzt werden sollen, *Typ:* Text

*Kommentar:* Byteposition des ersten zu ersetzenden Zeichens, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Bytelänge der zu ersetzenden Zeichen, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

*Kommentar:* Text, der die Zeichen im alten Text ersetzen wird, *Typ:* Text

##### Ähnliche Funktionen

[FINDB](#)

[MIDB](#)

[FIND](#)

[MID](#)

#### 8.1.12.26 REPT

Die Funktion REPT() wiederholt den ersten Parameter so häufig, wie durch den zweiten Parameter festgelegt. Der zweite Parameter darf nicht negativ sein. Die Funktion ergibt eine leere Zeichenfolge, wenn der zweite Parameter auf 0 gesetzt ist (oder auf 0 abgerundet wird).

*Rückgabebetyp:* Text

##### Syntax

REPT(Text;Anzahl)

##### Parameter

*Kommentar:* Quell-Zeichenfolge, *Typ:* Text

*Kommentar:* Anzahl der Wiederholungen, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

REPT("KSpread";3) ergibt "KSpreadKSpreadKSpread"

##### Beispiele

REPT("KSpread";0) ergibt ""

#### 8.1.12.27 RIGHT

Die Funktion RIGHT() ergibt eine Teil-Zeichenfolge, welche aus der angegebenen Anzahl Zeichen von rechts aus der Quellzeichenfolge gebildet wird. Falls „Länge“ die Gesamtlänge der Zeichenfolge überschreitet, wird die ganze Zeichenfolge zurückgegeben.

*Rückgabebetyp:* Text

##### Syntax

RIGHT(Text;Länge)

**Parameter**

*Kommentar:* Quell-Zeichenfolge, *Typ:* Text

*Kommentar:* Anzahl an Zeichen, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

**Beispiele**

RIGHT("Hallo";2) ergibt "lo"

**Beispiele**

RIGHT("KSpread";10) ergibt "KSpread"

**Beispiele**

RIGHT("KSpread") ergibt "d"

**Ähnliche Funktionen**

LEFT  
MID  
LEFTB  
MIDB

**8.1.12.28 RIGHTB**

Die Funktion RIGHTB() ergibt eine Teil-Zeichenfolge, welche aus der angegebenen Anzahl Zeichen von rechts aus der Quellzeichenfolge unter Verwendung der Byteposition gebildet wird. Falls „Länge“ die Gesamtlänge der Zeichenfolge überschreitet, wird die ganze Zeichenfolge zurückgegeben.

*Rückgabotyp:* Text

**Syntax**

RIGHTB(Text;ByteLänge)

**Parameter**

*Kommentar:* Quell-Zeichenfolge, *Typ:* Text

*Kommentar:* Bytelänge, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

**Ähnliche Funktionen**

LEFT  
MID  
LEFTB  
MIDB

**8.1.12.29 ROT13**

Die Funktion ROT13() verschlüsselt Text dadurch, dass jeder Buchstabe durch den Buchstaben ersetzt wird, der 13 Zeichen später im Alphabet steht. Wenn die dreizehnte Position hinter dem Buchstaben Z ist, wird wieder bei A begonnen (Rotation).

Durch Anwenden der Verschlüsselungsfunktion auf den verschlüsselten Text kann der Text wieder entschlüsselt werden.

*Rückgabotyp:* Text

**Syntax**

ROT13(Text)

#### Parameter

*Kommentar:* Text, *Typ:* Text

#### Beispiele

ROT13("KSpread") ergibt "XFcernq"

#### Beispiele

ROT13("XFcernq") ergibt "KSpread"

### 8.1.12.30 SEARCH

Die Funktion SEARCH() findet eine Zeichenfolge „find\_text“ in einer anderen Zeichenfolge „within\_text“ und gibt die Position des ersten Zeichens von „find\_text“ in „within\_text“ zurück.

Sie können die Platzhalter Fragezeichen (?) und Stern (\*) verwenden. Ein Fragezeichen steht für ein einzelnes Zeichen und ein Stern steht für eine beliebige Folge von Zeichen.

Der Parameter start\_num gibt den Buchstaben an, bei dem die Suche beginnen soll. Der erste Buchstabe steht an Position 1. Wenn start\_num nicht angegeben ist, wird ein Wert von 1 angenommen. Die Funktion SEARCH() unterscheidet nicht zwischen Großbuchstaben und Kleinbuchstaben.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

SEARCH(find\_text;within\_text;start\_num)

#### Parameter

*Kommentar:* Der zu suchende Text, *Typ:* Text

*Kommentar:* Der Text, in dem gesucht werden soll, *Typ:* Text

*Kommentar:* Startposition für die Suche, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

#### Beispiele

SEARCH("e";"Kommentare";6) ergibt 10

#### Beispiele

SEARCH("schwelle";"Gewinnschwelle") ergibt 7

#### Ähnliche Funktionen

[FIND](#)

[FINDB](#)

[SEARCHB](#)

### 8.1.12.31 SEARCHB

Die Funktion SEARCHB() findet eine Zeichenfolge „find\_text“ in einer anderen Zeichenfolge „within\_text“ und gibt die Position des ersten Zeichens von „find\_text“ in „within\_text“ unter Verwendung der Byteposition zurück.

Sie können die Platzhalter Fragezeichen (?) und Stern (\*) verwenden. Ein Fragezeichen steht für ein einzelnes Zeichen und ein Stern steht für eine beliebige Folge von Zeichen.

Der Parameter BytePosition gibt den Buchstaben an, bei dem die Suche beginnen soll. Der erste Buchstabe steht an Position 2. Wenn BytePosition nicht angegeben ist, wird ein Wert von 2 angenommen. Die Funktion SEARCHB() unterscheidet nicht zwischen Großbuchstaben und Kleinbuchstaben.

*Rückgabotyp:* Ganze Zahl (z. B. 1, 132, 2344)

### Syntax

SEARCHB(find\_text;within\_text;BytePosition Start)

### Parameter

*Kommentar:* Der zu suchende Text, *Typ:* Text

*Kommentar:* Der Text, in dem gesucht werden soll, *Typ:* Text

*Kommentar:* Angegebene Byteposition für der Start der Suche, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Ähnliche Funktionen

[FINDB](#)  
[FIND](#)  
[SEARCH](#)

#### 8.1.12.32 SLEEK

Die Funktion SLEEK() entfernt alle Leerzeichen aus einer Zeichenfolge.

*Rückgabetyp:* Text

### Syntax

SLEEK(text)

### Parameter

*Kommentar:* Quell-Zeichenfolge, *Typ:* Text

### Beispiele

SLEEK("Dies ist ein Text ") ergibt "DiesisteinText"

### Ähnliche Funktionen

[TRIM](#)

#### 8.1.12.33 SUBSTITUTE

Die Funktion SUBSTITUTE() ersetzt in einer Zeichenfolge alter\_text durch neuer\_text. Wenn Anzahl angegeben ist, wird nur die entsprechende Anzahl an Ersetzungen durchgeführt. Sonst wird jedes Vorkommen von alter\_text durch neuer\_text ersetzt. Verwenden Sie SUBSTITUTE(), wenn Sie bestimmten Text in einer Zeichenfolge ersetzen möchten. Verwenden Sie REPLACE(), wenn Sie Text, der an einer bestimmten Position steht, ersetzen möchten.

*Rückgabetyp:* Text

### Syntax

SUBSTITUTE(text; alter\_text; neuer\_text; Anzahl)

### Parameter

*Kommentar:* Text, in dem ersetzt werden soll, *Typ:* Text

*Kommentar:* Zu ersetzender Text, *Typ:* Text

*Kommentar:* Ersatztext, *Typ:* Text

*Kommentar:* Anzahl der Ersetzungen, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

### Beispiele

SUBSTITUTE("Kosten-Daten";"Kosten";"Verkaufs") ergibt "Verkaufs-Daten"

### Beispiele

SUBSTITUTE("Quartal 1, 2001";"1";"3";1) ergibt "Quartal 3, 2001"

### Beispiele

SUBSTITUTE("Quartal 1, 2001";"1";"3";4) ergibt "Quartal 3, 2003"

### Ähnliche Funktionen

REPLACE  
REPLACEB  
FIND  
FINDB

#### 8.1.12.34 T

Die Funktion T() ergibt den Text, auf den durch „Wert“ verwiesen wird. Wenn „Wert“ ein Text ist oder auf einen Text verweist, dann wird der entsprechende Text zurückgegeben. Wenn „Wert“ nicht auf einen Text verweist, wird ein leerer Text zurückgegeben.

*Rückgabotyp:* Text

### Syntax

T(Wert)

### Parameter

*Kommentar:* Wert, *Typ:* Ein beliebiger Wert

### Beispiele

T("Calligra") ergibt "Calligra"

### Beispiele

T(1,2) ergibt "" (leerer Text)

#### 8.1.12.35 TEXT

Die Funktion TEXT() konvertiert einen Wert in einen Text.

*Rückgabotyp:* Text

### Syntax

TEXT(Wert)

### Parameter

*Kommentar:* Wert, *Typ:* Ein beliebiger Wert

### Beispiele

TEXT(1234,56) ergibt „1234,56“

### Beispiele

TEXT("KSpread") ergibt "KSpread"

#### 8.1.12.36 TOGGLE

Die Funktion TOGGLE() verwandelt Großbuchstaben in Kleinbuchstaben und Kleinbuchstaben in Großbuchstaben.

*Rückgabetyp:* Text

##### Syntax

TOGGLE(Text)

##### Parameter

*Kommentar:* Quell-Zeichenfolge, *Typ:* Text

##### Beispiele

TOGGLE("hallo") ergibt "HALLO"

##### Beispiele

TOGGLE("HALLO") ergibt "hallo"

##### Beispiele

TOGGLE("HaLIo") ergibt "hAIlo"

##### Ähnliche Funktionen

UPPER

LOWER

#### 8.1.12.37 TRIM

Die Funktion TRIM() gibt die Zeichenfolge mit einzelnen Leerzeichen zwischen den Worten zurück.

*Rückgabetyp:* Text

##### Syntax

TRIM(text)

##### Parameter

*Kommentar:* Zeichenfolge, *Typ:* Text

##### Beispiele

TRIM(" Hallo KSpread ") ergibt "Hallo KSpread"

#### 8.1.12.38 UNICHAR

Die Funktion UNICHAR() gibt den durch eine Unicode-Zahl spezifizierten Buchstaben zurück.

*Rückgabetyp:* Text

##### Syntax

UNICHAR(Code)

##### Parameter

*Kommentar:* Zeichen-Code, *Typ:* Ganze Zahl (z. B. 1, 132, 2344)

##### Beispiele

UNICHAR(65) ergibt "A"

##### Ähnliche Funktionen

UNICODE

CHAR

### 8.1.12.39 UNICODE

Die Funktion UNICODE() gibt die Unicode-Zahl für das erste Zeichen in einer Zeichenfolge zurück.

*Rückgabebetyp:* Ganze Zahl (z. B. 1, 132, 2344)

#### Syntax

UNICODE(Text)

#### Parameter

*Kommentar:* Text, *Typ:* Text

#### Beispiele

UNICODE("KDE") ergibt 75

#### Ähnliche Funktionen

[UNICHAR  
CODE](#)

### 8.1.12.40 UPPER

Die Funktion UPPER() konvertiert eine Zeichenfolge in Großbuchstaben.

*Rückgabebetyp:* Text

#### Syntax

UPPER(text)

#### Parameter

*Kommentar:* Quell-Zeichenfolge, *Typ:* Text

#### Beispiele

UPPER("hallo") ergibt "HALLO"

#### Beispiele

UPPER("HALLO") ergibt "HALLO"

#### Ähnliche Funktionen

[LOWER  
TOGGLE](#)

### 8.1.12.41 VALUE

Konvertiert eine Zeichenfolge, die einen Wert darstellt, in den Wert.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

VALUE(Text)

#### Parameter

*Kommentar:* Text, *Typ:* Text

#### Beispiele

VALUE("14,03") ergibt 14,03

## 8.1.13 Trigonometrie

### 8.1.13.1 ACOS

Die Funktion ACOS() gibt den Arcuskosinus im Bogenmaß zurück. Der Wert ist mathematisch zwischen 0 und PI definiert (inklusive).

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ACOS(Fließkommawert)

#### Parameter

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

ACOS(0,8) ergibt 0,6435011

#### Beispiele

ACOS(0) ergibt 1,57079633

#### Ähnliche Funktionen

[COS](#)

### 8.1.13.2 ACOSH

Die Funktion ACOSH() berechnet den inversen Kosinus Hyperbolicus von x. Das ist der Wert, dessen Kosinus Hyperbolicus gleich x ist. Wenn x kleiner als 1,0 ist, gibt ACOSH() einen ungültigen Wert (NaN) zurück und errno wird entsprechend gesetzt.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ACOSH(Fließkommawert)

#### Parameter

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

ACOSH(5) ergibt 2,29243167

#### Beispiele

ACOSH(0) ist ungültig

#### Ähnliche Funktionen

[COSH](#)

### 8.1.13.3 ACOT

Die Funktion ACOT() liefert den inversen Kotangens einer Zahl.

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ACOT(Fließkommawert)

#### Parameter

*Kommentar*: Winkel (rad);, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

ACOT(0) ergibt 1,57079633

### 8.1.13.4 ASIN

Die Funktion ASIN() gibt den Arcussinus im Bogenmaß zurück. Der Wert ist mathematisch zwischen  $-\pi/2$  und  $\pi/2$  definiert (inklusive).

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ASIN(Fließkommawert)

#### Parameter

*Kommentar*: Winkel (rad);, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

ASIN(0,8) ergibt 0,92729522

#### Beispiele

ASIN(0) ergibt 0

#### Ähnliche Funktionen

[SIN](#)

### 8.1.13.5 ASINH

Die Funktion ASINH() berechnet den inversen Sinus Hyperbolicus von x. Das ist der Wert, dessen Sinus Hyperbolicus gleich x ist.

*Rückgabetyt*: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ASINH(Fließkommawert)

#### Parameter

*Kommentar*: Winkel (rad);, *Typ*: Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

ASINH(0,8) ergibt 0,73266826

#### Beispiele

ASINH(0) ergibt 0

#### Ähnliche Funktionen

[SINH](#)

### 8.1.13.6 ATAN

Die Funktion ATAN() gibt den Arcustangens im Bogenmaß zurück. Der Wert ist mathematisch zwischen  $-\pi/2$  und  $\pi/2$  definiert (inklusive).

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ATAN(Fließkommawert)

#### Parameter

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

ATAN(0,8) ergibt 0,67474094

#### Beispiele

ATAN(0) ergibt 0

#### Ähnliche Funktionen

[TAN](#)  
[ATAN2](#)

### 8.1.13.7 ATAN2

Diese Funktion berechnet den Arcustangens der beiden Variablen x und y. Es ist ähnlich dazu, den Arcustangens von  $y/x$  auszurechnen, nur dass die Vorzeichen der beiden Argumente dazu benutzt werden, den Quadranten des Ergebnisses zu bestimmen.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ATAN2(Wert;Wert)

#### Parameter

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

ATAN2(0,5;1,0) ergibt 1,107149

#### Beispiele

ATAN2(-0,5;2,0) ergibt 1,815775

#### Ähnliche Funktionen

[ATAN](#)

### 8.1.13.8 ATANH

Die Funktion ATANH() berechnet den inversen Tangens Hyperbolicus von x. Das ist der Wert, dessen Tangens Hyperbolicus gleich x ist. Wenn der Betrag von x größer als 1,0 ist, gibt ATANH() einen ungültigen Wert (NaN) zurück.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

ATANH(Fließkommawert)

#### Parameter

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

ATANH(0,8) ergibt 1,09861229

#### Beispiele

ATANH(0) ergibt 0

#### Ähnliche Funktionen

[TANH](#)

### 8.1.13.9 COS

Die Funktion COS() gibt den Kosinus von x zurück, wobei x im Bogenmaß gegeben ist.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

COS(Fließkommawert)

#### Parameter

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

COS(0) ergibt 1,0

#### Beispiele

COS(PI/2) ergibt 0,0

#### Ähnliche Funktionen

[SIN](#)  
[ACOS](#)

### 8.1.13.10 COSH

Die Funktion COSH() gibt den Kosinus Hyperbolicus von x zurück. Er ist mathematisch als  $(\exp(x) + (\exp(-x))) / 2$  definiert.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

COSH(Fließkommawert)

**Parameter**

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Beispiele**

COSH(0,8) ergibt 1,33743495

**Beispiele**

COSH(0) ergibt 1

**Ähnliche Funktionen**

[ACOSH](#)

**8.1.13.11 CSC**

Die Funktion CSC() gibt den Kosecans von x zurück, wobei x im Bogenmaß gegeben ist.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

CSC(Fließkommawert)

**Parameter**

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Beispiele**

CSC(PI()/2) ergibt 1

**8.1.13.12 CSCH**

Die Funktion CSCH() gibt den Kosecans Hyperbolicus von x zurück, wobei x im Bogenmaß gegeben ist.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

CSCH(Fließkommawert)

**Parameter**

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Beispiele**

CSCH(PI()/2) ergibt 0.434537208...

**8.1.13.13 DEGREES**

Diese Funktion transformiert einen Winkel vom Bogenmaß ins Gradmaß.

*Rückgabetyt:* Ein Fließkommawert (wie 1,3; 0,343; 253)

**Syntax**

DEGREE(Fließkommawert)

**Parameter**

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

### Beispiele

DEGREES(0,78) ergibt 44,69

### Beispiele

DEGREES(1) ergibt 57,29

### Ähnliche Funktionen

[RADIANS](#)

#### 8.1.13.14 PI

Die Funktion PI() gibt den Wert von PI zurück.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

PI()

#### Parameter

#### Beispiele

PI() ergibt 3,141592654...

#### 8.1.13.15 RADIANS

Diese Funktion transformiert einen Winkel vom Gradmaß ins Bogenmaß.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

RADIANS(Fließkommawert)

#### Parameter

*Kommentar:* Winkel (grad), *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

RADIANS(75) ergibt 1,308

#### Beispiele

RADIANS(90) ergibt 1,5707

### Ähnliche Funktionen

[DEGREES](#)

#### 8.1.13.16 SEC

Die Funktion SECH() gibt den Secans von x zurück, wobei x im Bogenmaß gegeben ist.

*Rückgabebetyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

SEC(Fließkommawert)

#### Parameter

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

SEC(0) ergibt 1

### 8.1.13.17 SECH

Die Funktion SECH() gibt den Secans Hyperbolicus von x zurück, wobei x im Bogenmaß gegeben ist.

*Rückgabetyyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

SECH(Fließkommawert)

#### Parameter

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

SECH(0) ergibt 1

### 8.1.13.18 SIN

Die Funktion SIN() gibt den Sinus von x zurück, wobei x im Bogenmaß gegeben ist.

*Rückgabetyyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

SIN(Fließkommawert)

#### Parameter

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

SIN(0) ergibt 0

#### Beispiele

SIN(PI/2) ergibt 1

#### Ähnliche Funktionen

[COS](#)  
[ASIN](#)

### 8.1.13.19 SINH

Die Funktion SINH() gibt den Sinus Hyperbolicus von x zurück. Er ist mathematisch als  $(\exp(x) - \exp(-x)) / 2$  definiert.

*Rückgabetyyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

SINH(Fließkommawert)

#### Parameter

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

SINH(0,8) ergibt 0,88810598

#### Beispiele

SINH(0) ergibt 0

#### Ähnliche Funktionen

[ASINH](#)

### 8.1.13.20 TAN

Die Funktion TAN() gibt den Tangens von  $x$  zurück, wobei  $x$  im Bogenmaß gegeben ist.

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

TAN(Fließkommawert)

#### Parameter

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

TAN(0,7) ergibt 0,84228838.

#### Beispiele

TAN(0) ergibt 0

#### Ähnliche Funktionen

[ATAN](#)

### 8.1.13.21 TANH

Die Funktion TANH() gibt den Tangens Hyperbolicus von  $x$  zurück. Er ist definiert als  $\sinh(x) / \cosh(x)$

*Rückgabotyp:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Syntax

TANH(Fließkommawert)

#### Parameter

*Kommentar:* Winkel (rad);, *Typ:* Ein Fließkommawert (wie 1,3; 0,343; 253)

#### Beispiele

TANH(0,8) ergibt 0,66403677

#### Beispiele

TANH(0) ergibt 0

#### Ähnliche Funktionen

[ATANH](#)

## Kapitel 9

# Danksagungen und Lizenz

Calligra Sheets

Programm Copyright 1998-2019 Das Calligra Sheets-Team:

- Torben Weis [weis@kde.org](mailto:weis@kde.org)
- Laurent Montel [lmontel@mandrakesoft.com](mailto:lmontel@mandrakesoft.com)
- David Faure [faure@kde.org](mailto:faure@kde.org)
- John Dailey [dailey@vt.edu](mailto:dailey@vt.edu)
- Philipp Müller [philipp.mueller@gmx.de](mailto:philipp.mueller@gmx.de)
- Ariya Hidayat [ariya@kde.org](mailto:ariya@kde.org)
- Norbert Andres [nandres@web.de](mailto:nandres@web.de)
- Shaheed Haque [srhaque@iee.org](mailto:srhaque@iee.org)
- Werner Trobin [trobin@kde.org](mailto:trobin@kde.org)
- Nikolas Zimmermann [wildfox@kde.org](mailto:wildfox@kde.org)
- Helge Deller [deller@kde.org](mailto:deller@kde.org)
- Percy Leonhart [percy@eris23.org](mailto:percy@eris23.org)
- Eva Brucherseifer [eva@kde.org](mailto:eva@kde.org)
- Phillip Ezolt [phillipezolt@hotmail.com](mailto:phillipezolt@hotmail.com)
- Enno Bartels [ebartels@nwn.de](mailto:ebartels@nwn.de)
- Graham Short [grahshrt@netscape.net](mailto:grahshrt@netscape.net)

Dokumentation Copyright 2002 Pamela Roberts [pamroberts@blueyonder.co.uk](mailto:pamroberts@blueyonder.co.uk)

Kleinere Aktualisierungen zur Dokumentation von KOffice 1.3 von Philip Rodrigues [phil@kde.org](mailto:phil@kde.org).

Aktualisierung der Bildschirmfotos für Calligra 3.1 durch Carl Schwan [carl@carlschwan.eu](mailto:carl@carlschwan.eu)

Deutsche Übersetzung von Holger Schröder [holger-kde@holgis.net](mailto:holger-kde@holgis.net).

Diese Dokumentation ist unter den Bedingungen der [GNU Free Documentation License](#) veröffentlicht.

Dieses Programm ist unter den Bedingungen der [GNU General Public License](#) veröffentlicht.

## Anhang A

# Installation

Calligra Sheets ist ein Teil des Calligra-Pakets für KDE und benutzt verschiedene Calligra-Bibliotheken. Calligra selbst ist ein Teil von KDE und benötigt die allgemeinen KDE-Bibliotheken.

Für Informationen zum Herunterladen und Installieren von KDE, Calligra und Calligra Sheets besuchen Sie bitte <http://www.kde.org> und <http://www.calligra.org>.