

LIBREOFFICE BASIC-MAKRO



Inhaltsverzeichnis

Makroverwaltung.....	3
Anlegen einer eigenen Bibliothek.....	3
Die Basic-IDE.....	5
Makro per Symbol und/ oder Tastatur starten.....	7
Codeeingabe in Modulblatt.....	7
Vorbereitung zur Ausführung des Makros.....	8
Option 1: Neue Symbolleiste und Makrozuweisung.....	8
Option 2: Neue Tastenkombi und Makrozuweisung.....	13
Makros: Löschen Zelleninhalte.....	14
Warum verschiedene Makros?.....	14
Makro Zahlen und Texte löschen.....	15
Makro: Nur Werte (Zahlen) löschen.....	16
Makro: Nur Datum und Zeit löschen.....	16
Makro: Nur Text löschen.....	17
Makro: Nur Notizen löschen.....	17
Makro: Nur Notizen löschen.....	18

Makroverwaltung

Anlegen einer eigenen Bibliothek

Wenn Makros nur in einem Dokument laufen sollen, muss man keine Bibliothek an legen. Wenn ein Makro in allen Dokumenten lauffähig sein soll, empfiehlt es sich eine neue Makrobibliothek anzulegen.

1. Dazu in der Menüleiste Extras\Makros → „Makros verwalten“ → „LibreOffice Basic...“, im sich öffnenden Dialog auf „Verwalten“ klicken.

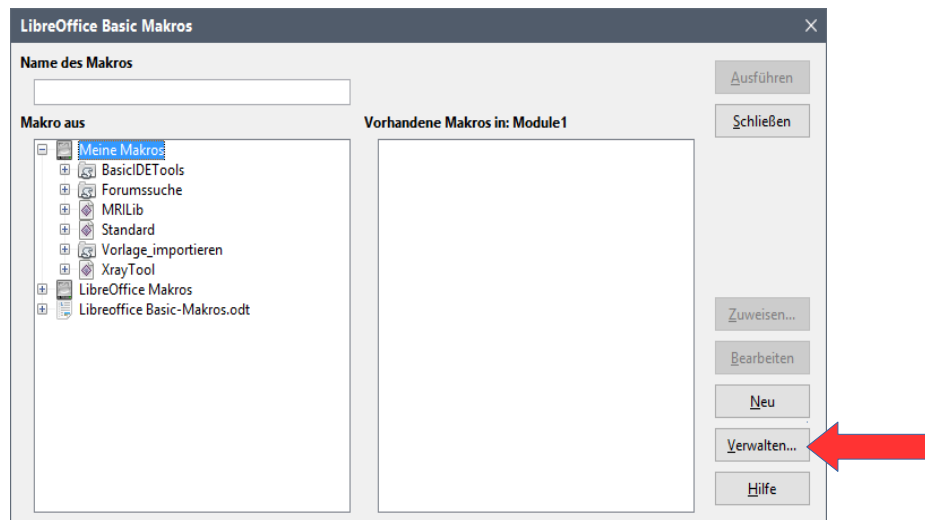


Abbildung 1: Makro

2. Im Fenster „Makros verwalten“ wir nun der Tab (Reiter) „Bibliothek“ ausgewählt. Nun unter der Option „Ort“: „Meine Makros & Dialoge“ auswählen, dann auf „Neu...“

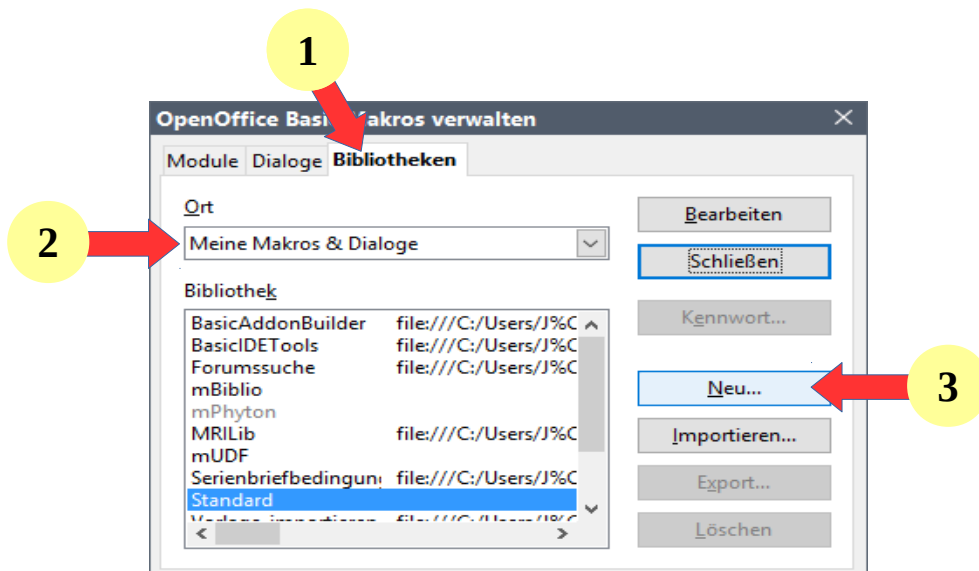


Abbildung 2: Makro verwalten

3. Es öffnet sich eine Inputbox in der ein sinnvoller Name für die neue Bibliothek vergeben wird. Bitte darauf achten das der Name keine Leerzeichen und keine Umlaute enthält, sonst wird dieser von LO nicht akzeptiert!
- Wie im nächsten Bild zusehen ist, habe ich eine Bibliothek namens „Cora“ erstellt.

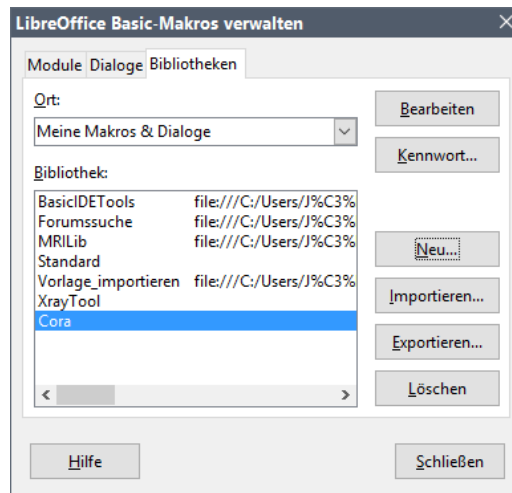


Abbildung 3: Makros verwalten - Neue Bibliothek

4. Und schon sind wir in der Abteilung Programmierung unterwegs. Das Fenster bleibt geöffnet und der Tab „Module“ wird ausgewählt.

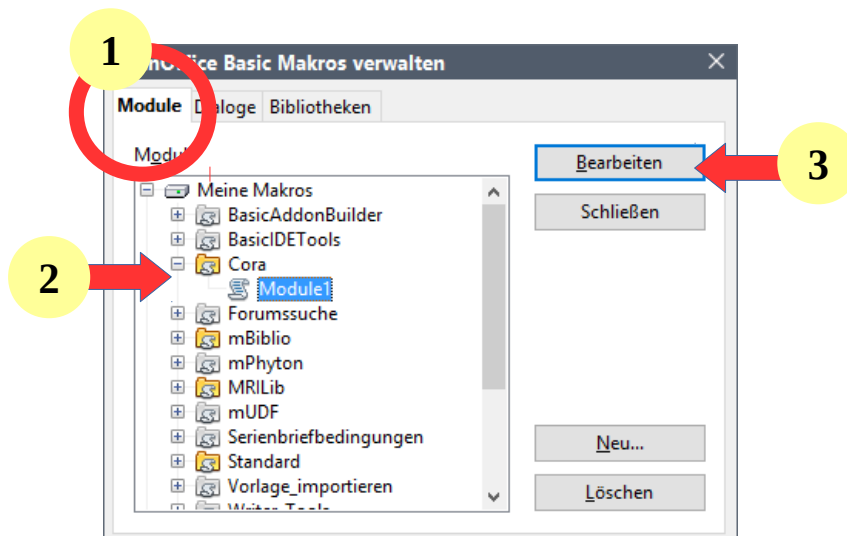


Abbildung 4: Basic-Makros verwalten

Die Basic-IDE

1. Jetzt wird die Basic *INTEGRATED DEVELOPMENT ENVIRONMENT* (IDE) geöffnet.
Damit dies schneller geht, ALT+F11 drücken.

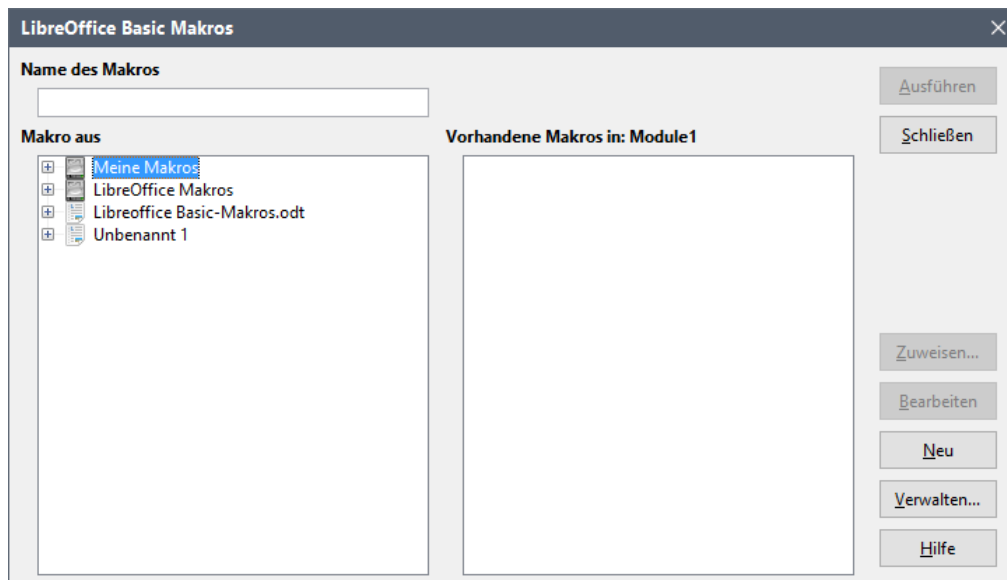


Abbildung 1: Basic Makros

2. Jetzt unter der Rubrik „Makro aus“ → „Meine Makros“ aufklappen, die neue Bibliothek aufklappen und auf „Modul1“ klicken. Dann auf bearbeiten.

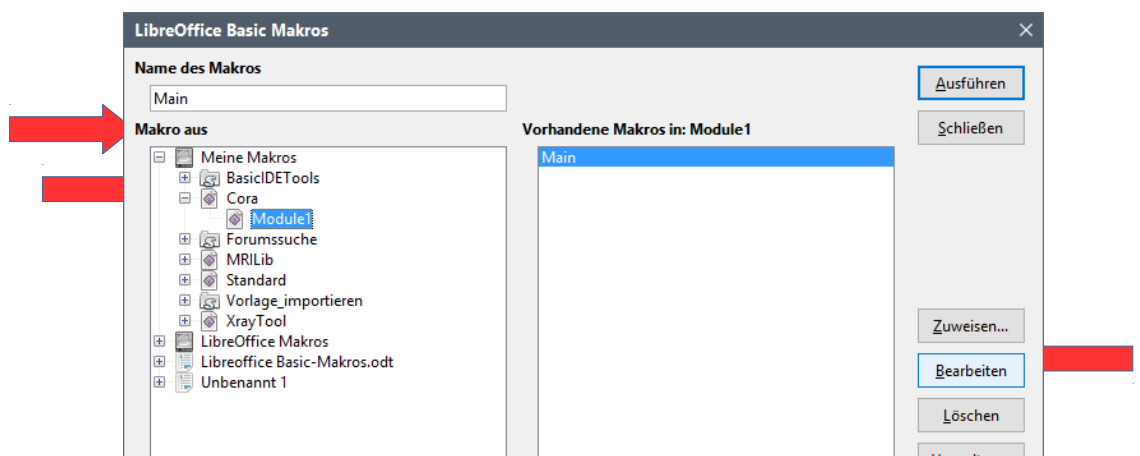


Abbildung 2: Basix Makros bearbeiten

3. Die integrierte Entwicklungsumgebung für Basic ist im Grunde sehr einfach zu verstehen.

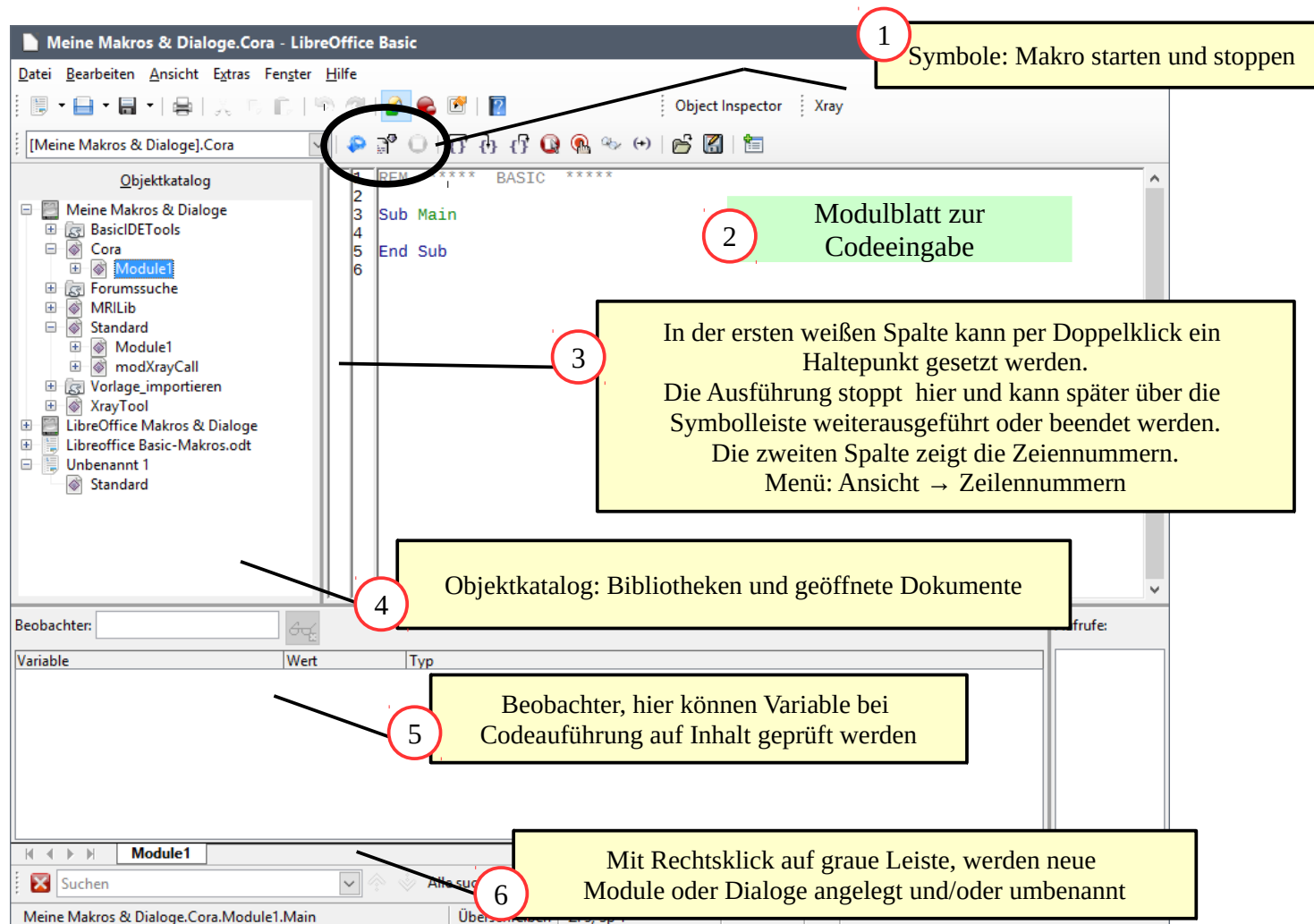


Abbildung 3: Die Basic-IDE

Makro per Symbol und/ oder Tastatur starten

Codeeingabe in Modulblatt

1. Im Objektkatalog die zuvor erstellte Bibliothek aufklappen und „Modul1“ anklicken.
2. Cursor in Modulblatt stellen und alles in Modulblatt mit STRG+a markieren, dann löschen (ENTF).
3. Am Ende des Dokuments steht der Code. Diesen markieren und mit STR+c kopieren, dann auf das Modulblatt in der IDE wechseln und in das leere Blatt einfügen.
4. Wie in Abbildung 1 unter Punkt 6 beschrieben kann das Modulblatt umbenannt werden, z.B. in: modDelete
5. Die IDE schließen. Hierbei wird der Code automatisch in LO gespeichert.

Vorbereitung zur Ausführung des Makros

Option 1: Neue Symbolleiste und Makrozuweisung

Auf der CALC-Oberfläche (nicht in der IDE) wird eine neue Symbolleiste mit neuen Symbolen erstellt. Den einzelnen Symbolen werden dann die jeweiligen Makro-SUB-Routinen zugewiesen:

1. Menü: „Extras\Anpassen...“ → Tab „Symbolleisten“ wählen.
2. „NEU“ anklicken. Folgender Dialog öffnet sich, hier wird der Name für die neue Leiste vergeben und sehr wichtig ist, dass in der Combobox unter „Speichern in:“ **LibreOffice CALC** ausgewählt ist. Damit ist gewährleistet, dass die Symbolleiste nicht nur im aktuellen CALC-Dokument, sondern in jedem CALC-Dokument verfügbar ist.

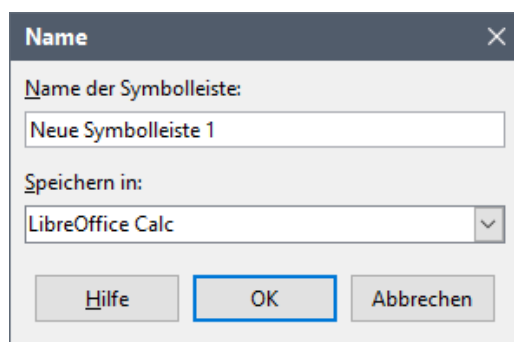


Abbildung 1:
Namensvergabe und Speicherort

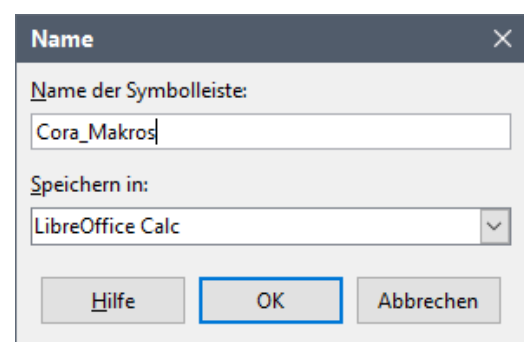


Abbildung 2:
zum Beispiel Name = Cora_Makros

3. Mit OK Rückkehr zum Anpassen-Dialog.

4. In der Comboxbox, links neben dem Button „Neu...“ befindet sich nun die neue Symbolleiste „Cora_Makros“ und wird jetzt ausgewählt.

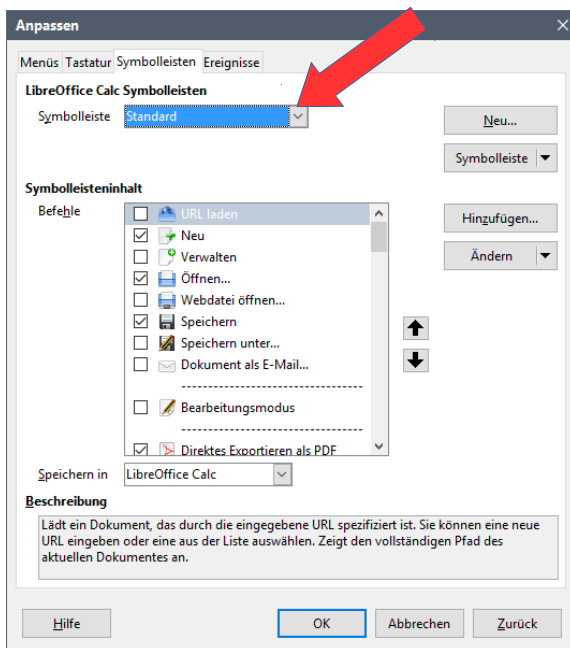


Abbildung 4:
Anpassen Symbolleiste

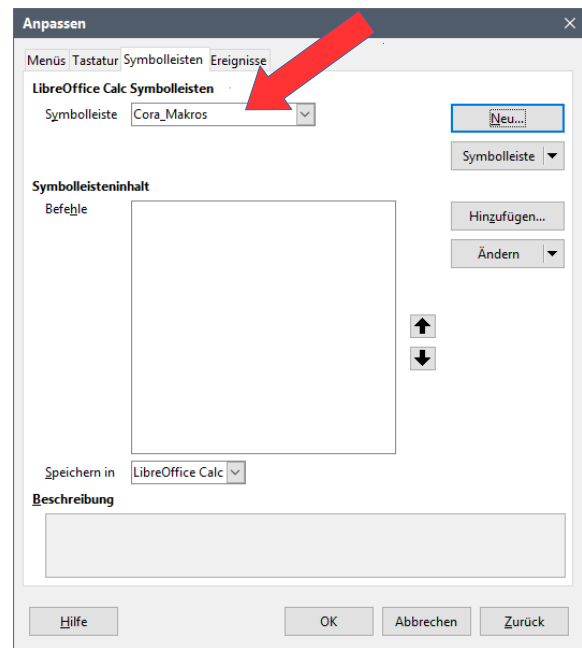


Abbildung 3:
Neue Symbolleiste auswählen

5. In Abbildung 3 ist nun der IST-Zustand der neuen Leiste. Es werden nun unter der Option „Symboleisteninhalt → Befehle neue Symbole hinzugefügt“, dazu auf den Button „Hinzufügen...“ klicken.

6. Im Dialog „Befehle hinzufügen“ wird unter „Bereich“ die Option „LibreOffice Makros“ gesucht und aufgeklappt. Dort unter „Meine Makros“ die neue „Bibliothek“ aufgeklappt.

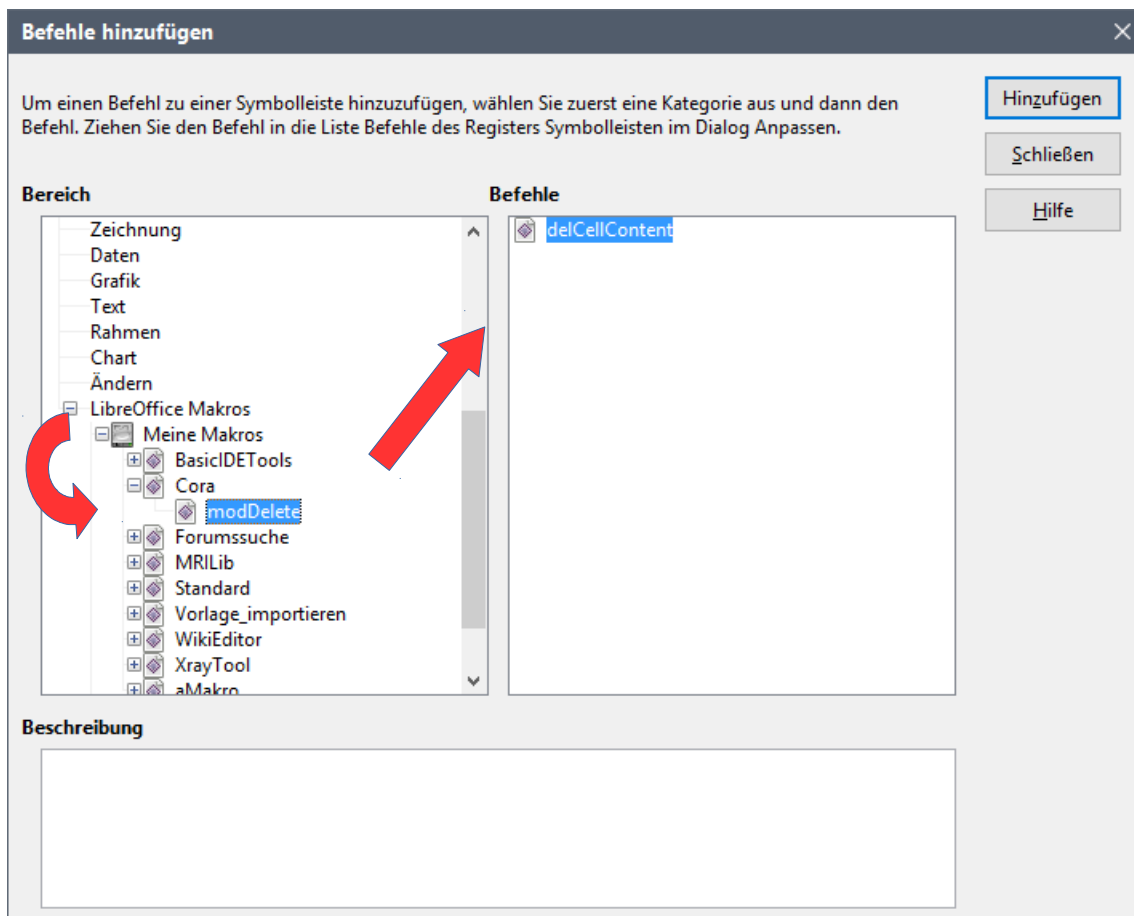


Abbildung 5:
Makro zuweisen

7. In der Abbildung sieht man die Bibliothek „Cora“ und darunter den Namen des Modulblattes „modDelete“. Dieses auswählen und es erscheint im rechten Fenster unter „Befehle“ der Name „delCellContent“, dies ist der Name der SUB-Routine/ des Makros. Wären mehrere Makros im Modulblatt „modDelete“ angelegt, dann würden hier entsprechend mehr Routinen untereinander aufgeführt und müssten separat angewählt werden. Nun, hier ist nur eine Routine vorhanden, dieses hat LO bereits makiert, falls nicht bitte markieren und anschließend auf den „Hinzufügen“ Button klicken, danach auf „Schließen“.

8. Unter „Ändern“ kann dem „Makro-Symbol“ nun ein eigenes Symbol hinzugefügt werden. Es gibt eine ganze Reihe vorgefertigter Symbole (Abb. 6) die hierfür verwendet werden können. Falls keines dabei ist was einem gefällt, kann auch ein selbst gezeichnetes oder aus dem Netz heruntergeladenes Symbol verwendet werden. Siehe Hinweis in Abb. 7 hier wird empfohlen Bilder mit einer Größe von 16x16 Pixeln. zu verwenden.

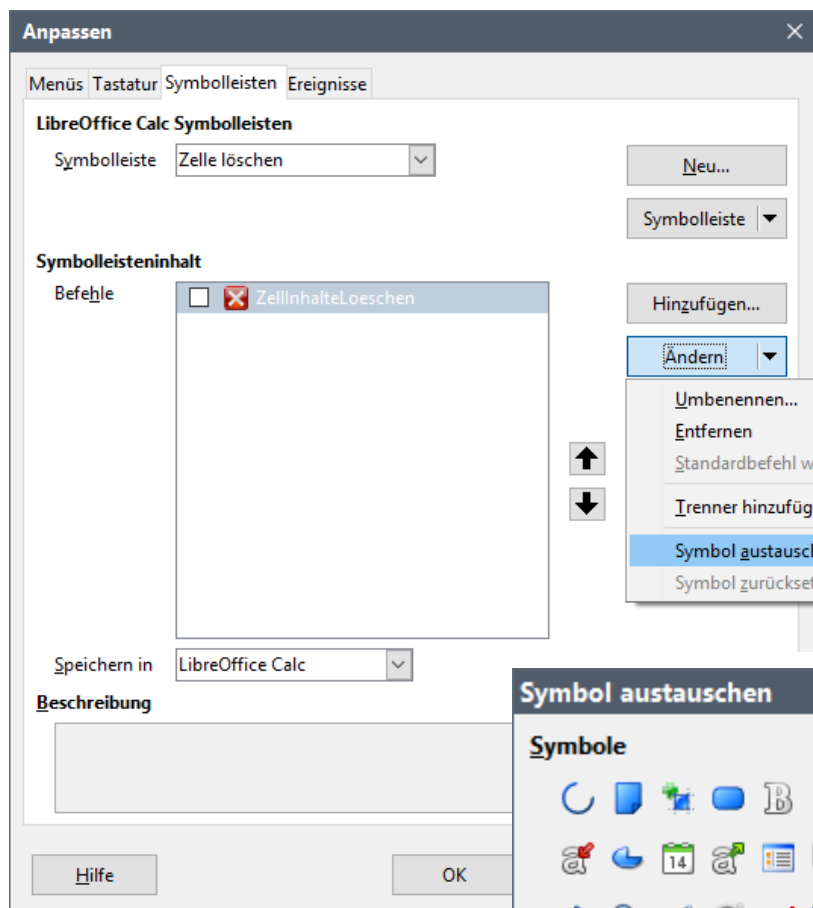


Abbildung 6: Symbol und/ oder Text in

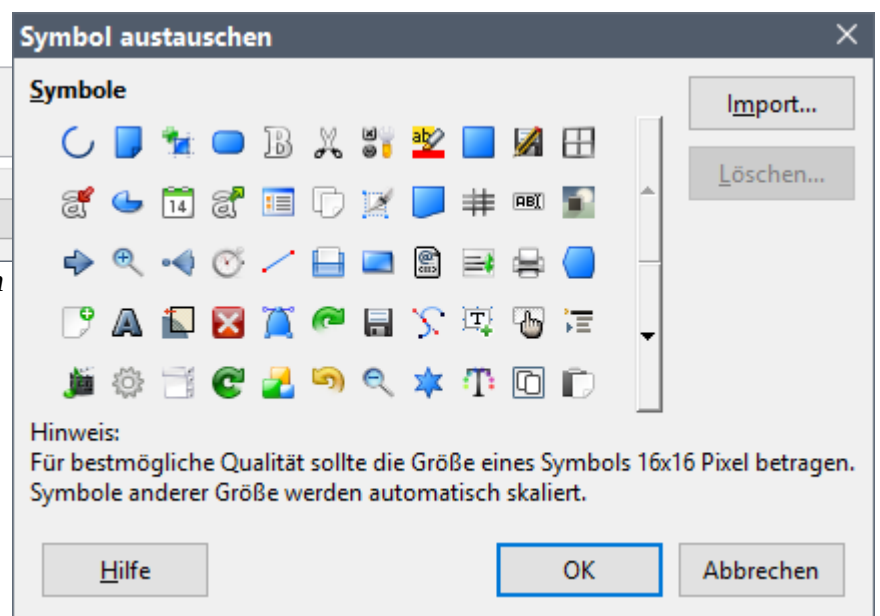


Abbildung 7: diverse Symbole

9. Eine weitere Option zur Darstellungsform findet man mit einem Klick auf Button „Symbolleiste“

Hier sind drei Darstellungsformen auswählbar (Abb. 8).

- Nur Symbole
- Nur Text
- Symbole & Text

10. Die Darstellungsform kann jederzeit geändert werden (Symbol und Text).

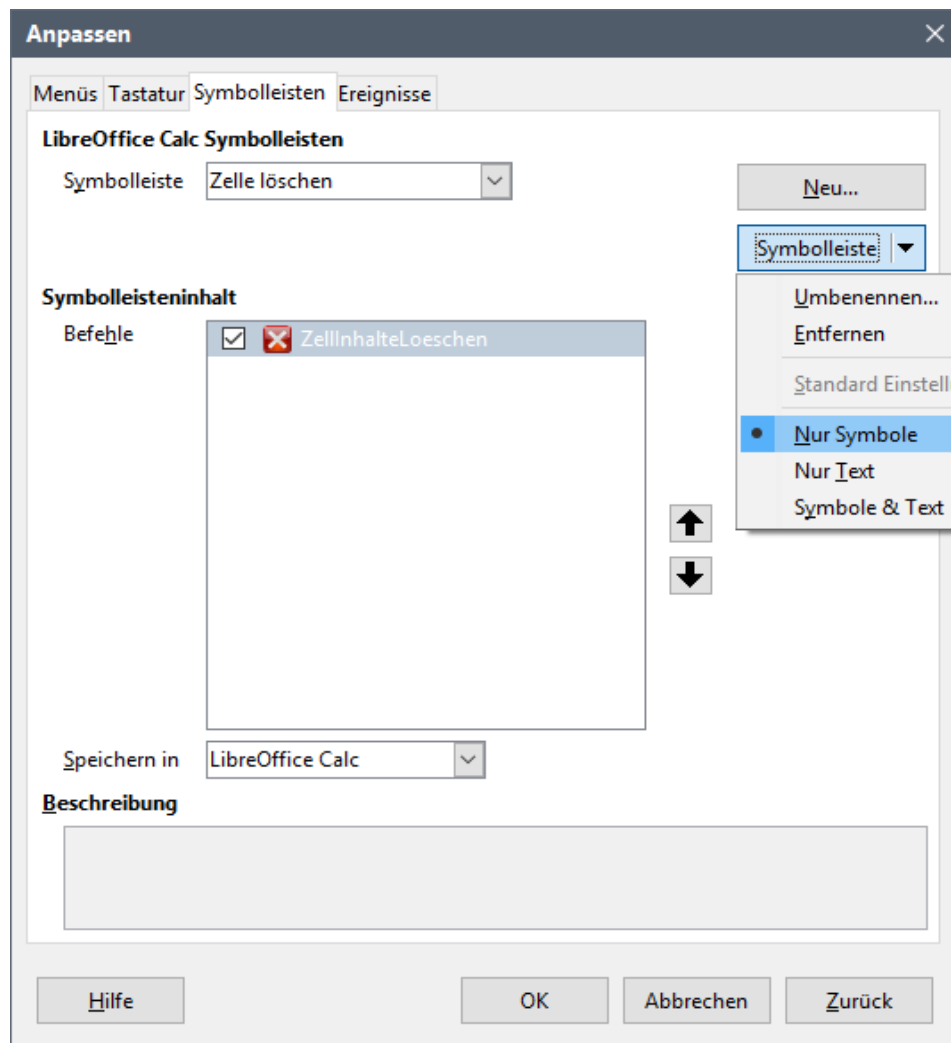


Abbildung 8: Symbol anpassen

11. Mit OK bestätigen.
Die neue Symbolleiste kann nun über Menü: Ansicht → Symbolleiste aktiviert und deaktiviert werden.
12. Im CALC-Dokument die gewünschten Zellen markieren und auf das Symbol klicken.
FERTIG!

Option 2: Neue Tastenkombi und Makrozuweisung

Im Prinzip funktioniert die Zuweisung einer Tastenkombination genauso, wie die Zuweisung bei einem Symbol.

1. Menü: Extras → Anpassen...
2. Tab: Tastatur auswählen
3. Unter Funktion\ Bereich: LibreOffice Makros → user → Cora → modCellContent auswählen.
4. Jetzt unter „Tastenkombinationen“ die gewünschte Tastenkombination auswählen.
5. Klick auf den Button „Ändern“
6. Die Aktion mit „OK“ bestätigen.
7. Das Makro funktioniert jetzt mit der ausgewählten Tastenkombination.
Hier im Bild ist Alt+Entf ausgewählt.

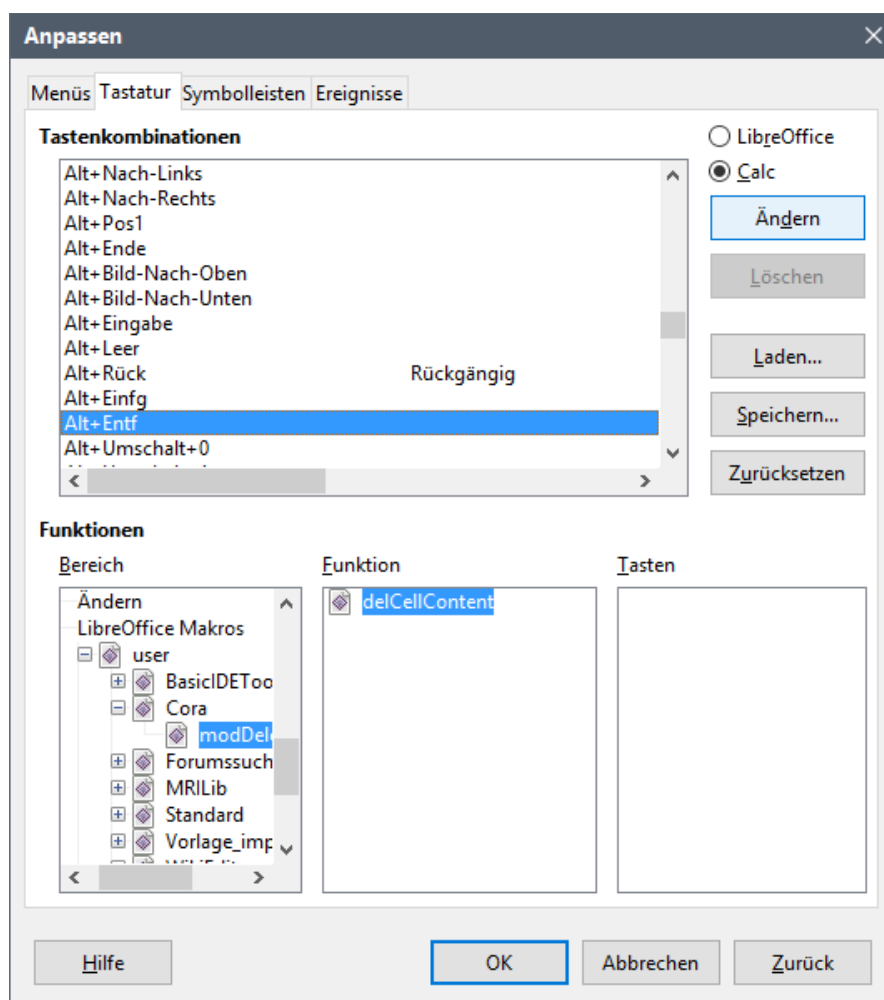


Abbildung 9: Tastatur anpassen

Makros: Löschen Zelleninhalte

Warum verschiedene Makros?

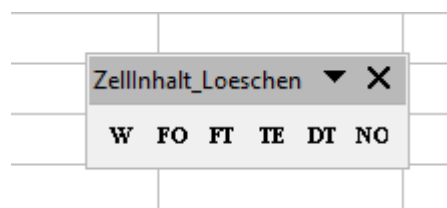
Eine neue Symbolleiste mit nur einem Symbol sieht etwas verloren aus.

Hier ein paar simple Symbole für verschiedene Löschaktionen.

Die Symbole haben eine Größe von 16x16 Pixeln und können im Symboleditor verwendet werden. Zum Import der Symbole in den Symboleditor, müssen diese hier einzeln kopiert werden. Dann Paint öffnen die Bildgröße auf 16x16 Pixel stellen, im Anschluß Bild für Bild einfügen und speichern. Danach können diese in den Symboleditor importiert werden. Siehe dazu Seite 10 Abb.7

FT	FT für Formatierungen
FO	FO für Formel
W	W für Werte
TE	TE für Texte
DT	DT für DateTime
NO	NO für Notizen

Das Ergebnis sieht dann z.B. so aus:



Es muss dann jedem einzelnen Symbol ein separates Makro zugewiesen werden.

Nachfolgend ersteinmal das Makro zum Löschen von Zahlen innerhalb einer Bereichsauswahl.

Makro Zahlen und Texte löschen

REM Nachfolgend die verfügbaren CELLFLAGS zur Löschungen der verschiedenen Zellinhalte.
 REM Um diese einfacher handhaben zu können, werden diese im Code in ein ARRAY gespeichert
 REM Siehe weitere Kommentare im Code

REM

REM Konstante	Inhalt	Integerwert
REM com.sun.star.sheet.CellFlags.VALUE	numerischer Wert	1
REM com.sun.star.sheet.CellFlags.DATETIME	Datum / Uhrzeit	2
REM com.sun.star.sheet.CellFlags.STRING	Text	4
REM com.sun.star.sheet.CellFlags.ANNOTATION	Notizen	8
REM com.sun.star.sheet.CellFlags.FORMULA	Funktionen	16
REM com.sun.star.sheet.CellFlags.HARDATTR	Harte Formatierung	32
REM com.sun.star.sheet.CellFlags.STYLES	Formatvorlagen	64
REM com.sun.star.sheet.CellFlags.OBJECTS	Zeichenobjekte	128

REM Der Code kann auch auf eine Mehrbereichsauswahl angewendet werden, z.B. auf
 REM markierten von Zellbereich(A1:B10) und Zellbereich(F20:F30)

Sub delCellContent

REM --- Variable deklarieren -----

Dim mConst(8) as Variant ' Array zur Aufnahme der Konstanten
 Dim oDoc as Object ' aktuelles Dokument
 Dim nConst as Long ' zur Übergabe der addierten Cellflags

REM Konstanten in Array laden

mConst(0) = com.sun.star.sheet.CellFlags.VALUE
 mConst(1) = com.sun.star.sheet.CellFlags.DATETIME
 mConst(2) = com.sun.star.sheet.CellFlags.STRING
 mConst(3) = com.sun.star.sheet.CellFlags.ANNOTATION
 mConst(4) = com.sun.star.sheet.CellFlags.FORMULA
 mConst(5) = com.sun.star.sheet.CellFlags.HARDATTR
 mConst(6) = com.sun.star.sheet.CellFlags.STYLES
 mConst(7) = com.sun.star.sheet.CellFlags.OBJECTS
 REM 255 = Alles löschen: Formate, Formeln, Texte, etc.
 mConst(8) = 255

' Referenz: aktuelles Dokument
 oDoc = ThisComponent

' aktuelle Zell-Selektion ermitteln
 oSel = oDoc.getCurrentSelection()

' Konstanten, welche im ARRAY gespeichert sind können hintereinander addiert werden.
 ' BEISPIEL: nConst = mConst(0) + mConst(2) + mConst(6)
 ' BEISPIEL: nConst = mConst(0) + mConst(2) löscht Werte und Texte
 nConst = mConst(0) + mConst(2)

' löschen der addierten Inhalte über com.sun.star.sheet.XSheetOperation
 oSel.clearContents(nConst)

End Sub

Makro: Nur Werte (Zahlen) löschen

Hellblaue Texte sind Kommentare und werden bei Codeausführung ignoriert!

Die dunkelblauen Überschriften nicht mit kopieren.

REM Der Code kann auch auf eine Mehrbereichsauswahl angewendet werden, z.B. auf

REM markierten von Zellbereich(A1:B10) und Zellbereich(F20:F30)

Sub delCellValue

REM --- Variable deklarieren -----

Dim oDoc as Object ' aktuelles Dokument

Dim nConst as Long ' zur Übergabe der addierten Cellflags

' Referenz: aktuelles Dokument

oDoc = ThisComponent

' aktuelle Zell-Selektion ermitteln

oSel = oDoc.getCurrentSelection()

' Konstante in Variable schreiben

nConst = com.sun.star.sheet.CellFlags.VALUE

' löschen der addierten Inhalte über com.sun.star.sheet.XSheetOperation

oSel.clearContents(nConst)

End Sub

Makro: Nur Datum und Zeit löschen

REM Der Code kann auch auf eine Mehrbereichsauswahl angewendet werden, z.B. auf

REM markierten von Zellbereich(A1:B10) und Zellbereich(F20:F30)

REM HINWEIS:

REM DateTime löscht keine reinen Datumsinhalte, sondern nur kombinierte Datum und Zeit Inhalte

REM 17.05.2016 wird nicht gelöscht

REM 17.05.2016 18:05:45 wird gelöscht

Sub delCellDateTime

REM --- Variable deklarieren -----

Dim oDoc as Object ' aktuelles Dokument

Dim nConst as Long ' zur Übergabe der addierten Cellflags

' Referenz: aktuelles Dokument

oDoc = ThisComponent

' aktuelle Zell-Selektion ermitteln

oSel = oDoc.getCurrentSelection()

' Konstante in Variable schreiben

nConst = com.sun.star.sheet.CellFlags.DATETIME

' löschen der addierten Inhalte über com.sun.star.sheet.XSheetOperation

oSel.clearContents(nConst)

End Sub

Makro: Nur Text löschen

REM Der Code kann auch auf eine Mehrbereichsauswahl angewendet werden, z.B. auf
REM markierten von Zellbereich(A1:B10) und Zellbereich(F20:F30)

REM Löscht nur Text

Sub delCellText

REM --- Variable deklarieren -----

Dim oDoc as Object ' aktuelles Dokument

Dim nConst as Long ' zur Übergabe der addierten Cellflags

' Referenz: aktuelles Dokument

oDoc = ThisComponent

' aktuelle Zell-Selektion ermitteln

oSel = oDoc.getCurrentSelection()

' Konstante in Variable schreiben

nConst = com.sun.star.sheet.CellFlags.STRING

' löschen der addierten Inhalte über com.sun.star.sheet.XSheetOperation

oSel.clearContents(nConst)

End Sub

Makro: Nur Notizen löschen

REM Der Code kann auch auf eine Mehrbereichsauswahl angewendet werden, z.B. auf
REM markierten von Zellbereich(A1:B10) und Zellbereich(F20:F30)

REM Löscht nur Notizen

Sub delCellAnnotation

REM --- Variable deklarieren -----

Dim oDoc as Object ' aktuelles Dokument

Dim nConst as Long ' zur Übergabe der addierten Cellflags

' Referenz: aktuelles Dokument

oDoc = ThisComponent

' aktuelle Zell-Selektion ermitteln

oSel = oDoc.getCurrentSelection()

' Konstante in Variable schreiben

nConst = com.sun.star.sheet.CellFlags.ANNOTATION

' löschen der addierten Inhalte über com.sun.star.sheet.XSheetOperation

oSel.clearContents(nConst)

End Sub

Makro: Nur Notizen löschen

REM Der Code kann auch auf eine Mehrbereichsauswahl angewendet werden, z.B. auf
REM markierten von Zellbereich(A1:B10) und Zellbereich(F20:F30)

REM Löscht nur Formeln

Sub delCellFormula

REM --- Variable deklarieren -----

Dim oDoc as Object ' aktuelles Dokument

Dim nConst as Long ' zur Übergabe der addierten Cellflags

' Referenz: aktuelles Dokument

oDoc = ThisComponent

' aktuelle Zell-Selektion ermitteln

oSel = oDoc.getCurrentSelection()

' Konstante in Variable schreiben

nConst = com.sun.star.sheet.CellFlags.FORMULA

' löschen der addierten Inhalte über com.sun.star.sheet.XSheetOperation

oSel.clearContents(nConst)

End Sub