



Für alle
Excel-
Versionen

Inge Baumeister, Dieter Klein

einfach gelernt!

VBA mit Excel

Der leichte Einstieg: Vom ersten Makro zur eigenen Eingabemaske

- *Mit vielen Beispielen und Lösungen, auch zum Download*
- *Für Einsteiger ohne Programmierkenntnisse*

VBA mit Excel - Der leichte Einstieg

Vom ersten Makro zur eigenen Eingabemaske

Inge Baumeister, Dieter Klein

Verlag:
BILDNER Verlag GmbH
Bahnhofstraße 8
94032 Passau

<http://www.bildner-verlag.de>
info@bildner-verlag.de

Tel.: +49 851-6700
Fax: +49 851-6624

ISBN: 978-3-8328-5363-1

Covergestaltung:
Christian Dadlhuber

Autoren:
Inge Baumeister, Dieter Klein

Bildquelle Cover: © sepy - Fotolia.com
Kapitelbild: © Brad Pict - stock.adobe.com

Herausgeber:
Christian Bildner

© 2018 BILDNER Verlag GmbH Passau

Die Informationen in diesem Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

Fast alle Hard- und Softwarebezeichnungen und Markennamen der jeweiligen Firmen, die in diesem Buch erwähnt werden, können auch ohne besondere Kennzeichnung warenzeichen-, marken- oder patent-rechtlichem Schutz unterliegen.

Die Namen, Adressen und sonstigen Daten der verwendeten Beispiele sind frei erfunden, Ähnlichkeiten mit Personen oder Produkten sind rein zufällig und keinesfalls beabsichtigt.

Das Werk einschließlich aller Teile ist urheberrechtlich geschützt. Es gelten die Lizenzbestimmungen der BILDNER Verlag GmbH Passau.

Vorwort

Keine Lust auf stets wiederkehrende Routineaufgaben mit Excel oder die Eingabe ewig gleicher, endlos verschachtelter Formeln? Wollen oder müssen Sie Daten aus Fragebögen, Messprotokollen oder sonstigen Listen in eine Excel-Tabelle eintragen und sind es leid, sich am Bildschirm umständlich zwischen Zeilen und Spalten zu bewegen?

Dann sollten Sie sich mit Makros, VBA, benutzerdefinierten Funktionen und selbst gestalteten Formularen befassen. Makros sind eigentlich nur gespeicherte Befehlsfolgen, die jederzeit und beliebig oft auf „Knopfdruck“ bzw. Mausklick ausgeführt werden. Sie werden entweder mit dem integrierten Makrorecorder von Excel aufgezeichnet oder in der Programmiersprache VBA (Visual Basic for Applications) geschrieben. Aufgezeichnete Makros können auch nachträglich mit VBA bearbeitet und z. B. um weitere Anweisungen ergänzt werden.

Mit diesem Buch wenden wir uns an Anwender, die bereits über Grundlagenkenntnisse im Umgang mit Microsoft Excel verfügen, Programmierkenntnisse werden hingegen nicht vorausgesetzt. Daher beginnen wir auch mit dem Aufzeichnen einfacher Makros mit Hilfe des Makrorecorders. Allerdings werden Sie schnell merken, dass solche Makros wenig flexibel und zudem oft mit unnötigem Programmcode überfrachtet sind. Zur gezielten Anpassung an individuelle Belange kommt VBA ins Spiel.

Neben der bekannten Arbeitsblattansicht verfügt Excel „backstage“ über ein gesondertes Fenster zur Eingabe von Makroanweisungen, den VBA-Editor, der Sie außerdem mit verschiedenen Eingabehilfen unterstützt. Das Buch führt Sie Schritt für Schritt und mit vielen nachvollziehbaren Beispielen in die Grundlagen der VBA-Programmierung ein und zeigt Ihnen anschließend in einem weiteren Kapitel, wie Sie in VBA mit Excel-Objekten, z. B. Tabellenblätter, Zellen und Zellbereiche umgehen. Sie erfahren außerdem, wie Sie als Ergänzung zu den vorhandenen Excel-Berechnungsfunktionen (MITTELWERT, WENN, usw.) Ihre eigenen Funktionen in VBA schreiben, im Arbeitsblatt nutzen und bei Bedarf auch anderen Anwendern zur Verfügung stellen.

Der zweite Teil des Buches ist den benutzerdefinierten Formularen, den sogenannten UserForms gewidmet. Anhand zahlreicher Beispiele lernen Sie, wie Sie mit Steuerelementen, z. B. Schaltflächen und Auswahlfeldern, Formulare als komfortable Benutzeroberfläche auch für weniger geübte Excel-Anwender gestalten. In Form eines Workshops erstellen Sie zum Beispiel eine Eingabemaske, die nicht nur zur Eingabe in eine Excel-Tabelle dient, einschließlich aller notwendigen Kontrollfunktionen, sondern auch zur Anzeige der Tabellendaten genutzt werden kann.

Im abschließenden Kapitel zeigen wir Ihnen anhand einiger konkreter Beispiele, wie UserForms die alltägliche Arbeit mit Excel erleichtern können. Einige Steuerelemente lassen sich übrigens auch in Arbeitsblättern einsetzen, auch hierzu finden Sie im Buch einige Tipps.

Und noch ein Hinweis zum Buch

In zahlreichen Kursen und Seminaren haben wir die Erfahrung gemacht, dass der Einstieg in die Programmierung mit greifbaren Ergebnissen einfach mehr Spaß macht. Daher befasst sich dieses Buch auch relativ ausführlich mit benutzerdefinierten Formularen, den UserForms. Sie lernen in Form von Workshops deren vielfältige Einsatzmöglichkeiten im Alltag kennen und erhalten gleichzeitig in zahlreichen Beispielen die Möglichkeit, Ihre neu erworbenen Kenntnisse der VBA-Programmierung praktisch anzuwenden und spielerisch zu vertiefen.

Download der Beispieldateien

Sämtliche verwendeten Beispiele erhalten Sie auf unserer Homepage kostenlos zum Download. Rufen Sie dazu die folgende Seite auf:

www.bildner-verlag.de/00324

Viel Spaß und Erfolg mit dem Buch wünschen Ihnen
BILDNER Verlag und Autoren

Inhalt

1	Vorbereitungen.....	13
1.1	Zum grundlegenden Verständnis von Makros und VBA	14
	Wozu Makros und VBA?	14
	Wichtige Begriffe.....	14
1.2	Das Register Entwicklertools anzeigen	15
1.3	Arbeitsmappen mit Makros speichern	17
	Der Dateityp Excel-Arbeitsmappe mit Makros (.xlsm).....	17
	Die persönliche Makroarbeitsmappe	18
1.4	Sicherheitseinstellungen	18
	Sicherheitseinstellungen kontrollieren.....	18
	Arbeitsmappe mit Makros öffnen.....	19
	Makros digital signieren	20
2	Makros aufzeichnen und ausführen.....	21
2.1	Hinweise zum Umgang mit aufgezeichneten Makros	22
2.2	Ein einfaches Makro aufzeichnen.....	22
	Die Aufzeichnung starten.....	22
	Makro ausführen.....	24
	Mögliche Probleme bei der Makro-Ausführung.....	25
2.3	Zellbezüge in Makros	26
2.4	Makroausführung starten.....	28
	Tastenkombination zuweisen	28
	Makro über die Symbolleiste für den Schnellzugriff starten	29
	Makros im Menüband einfügen.....	30
	Makro über eine Befehlsschaltfläche im Tabellenblatt starten	32
2.5	Beispiel: Diagramm mit Makros steuern	34

3	Grundlagen der VBA-Programmierung	37
3.1	Der VBA-Editor	38
	VBA-Editor öffnen	38
	So finden Sie sich im VBA-Editor zurecht	39
	Prozedurcode anzeigen	40
	Objekteigenschaften im Eigenschaftenfenster	41
3.2	Module und Prozeduren	42
	Ein neues Modul einfügen	42
	Eine neue Prozedur erzeugen	43
	Prozedur ausführen	45
3.3	Anweisungen eingeben	46
	Einstellungen im VBA-Editor	46
	Hinweise zur Texteingabe	49
	Eingabe von Parametern bzw. Argumenten	50
	Objekte, Methoden und Eigenschaften	52
	Kommentare	52
	Die VBA-Hilfe	53
3.4	Variablen, Konstanten und Operatoren	54
	Variablen verwenden	54
	Konstanten festlegen	56
	Hinweise für die Vergabe von Namen	57
	Operatoren und Ausdrücke	57
3.5	Einfache Dialoge	58
	Eine Meldung ausgeben	59
	Benutzereingaben mit InputBox	60
3.6	Abfragen und Schleifen	61
	Abfragen oder Verzweigungen	61
	Wiederholungsschleifen	64
	Bedingungsschleifen	66
	Die With-Anweisung	68
3.7	Fehlersuche und Fehlerbehandlung	69
	Einzelschritte ausführen	69
	Variablen überwachen	70
	Ausführung an Haltepunkten unterbrechen	72
	Anweisungszeilen auskommentieren	73
	Fehlerbehandlung	73

3.8	Routinen und Funktionen einsetzen.....	76
	Wozu Routinen benötigt werden.....	76
	Funktion erstellen und aufrufen.....	76
	Prozeduren als Routine.....	78
3.9	Mit Datenfeldern arbeiten	79
	Eindimensionale Datenfelder.....	79
	Die Verwendung von LBound und UBound.....	80
	Dynamische Datenfelder	81
	Zweidimensionale Datenfelder	82

4 Arbeiten mit Excel-Objekten 85

4.1	Die Excel-Objekte	86
	Die Objekthierarchie.....	86
	Eigenschaften, Methoden und Ereignisse von Objekten.....	87
	Der Objektkatalog	88
	Objektvariablen.....	90
4.2	Arbeitsmappen und Arbeitsblätter.....	91
	Arbeitsblätter	91
	Zugriff auf Arbeitsmappen.....	92
	Ereignisprozeduren für Arbeitsmappen und Arbeitsblätter erstellen.....	93
	Andere Ereignisse	95
4.3	Das Range Objekt (Zellen und Zellbereiche).....	96
	Zell- und Bereichsadresse als Text	96
	Adressierung über Cells.....	97
	Arbeiten mit der aktiven Zelle bzw. dem markierten Zellbereich	98
	Markierung mit der Offset Methode verschieben	99
	Position der aktuellen Zelle ermitteln	100
	Umfang eines Zellbereichs ermitteln	101
	Beispiel: Jede zweite Zeile mit einer Füllfarbe formatieren.....	103
	Zellbereiche ausschneiden, kopieren und einfügen	105
4.4	Berechnungen mit Formeln und Funktionen	106
	Bezüge in der A1-Schreibweise	106
	Bezüge als Zeilen- und Spaltenindex (R1C1)	107
	Formeln in einen Zellbereich schreiben	108
	Tabellenfunktionen verwenden	109
4.5	Letzte benutzte Zeile einer Tabelle bestimmen	111
	Verwendeten Zellbereich ermitteln.....	112
	Die Rückwärtssuche ab der letzten Tabellenzeile und Spalte	113

4.6	Diagramme steuern	114
	Datenreihen anzeigen, Fortsetzung aus Kapitel 2.5	114
	Diagrammobjekte.....	115
	Diagrammelemente hinzufügen.....	117
4.7	Benutzerdefinierte Funktionen im Arbeitsblatt einsetzen	118
	Funktion erstellen	118
	Funktionsbeschreibung hinzufügen	119
	Funktion in der persönlichen Makroarbeitsmappe speichern.....	121
	Arbeitsmappe als Add-In speichern.....	123
	Kleine Beispiele für benutzerdefinierte Funktionen	126

5 **Steuerelemente in Tabellenblättern** **129**

5.1	Wozu Steuerelemente?	130
5.2	Grundlagen Formularsteuerelemente	130
	Wo finden Sie die Formularsteuerelemente?	130
	Formularsteuerelement einfügen.....	131
	Formularsteuerelement bearbeiten.....	132
5.3	Beispiele für Formularsteuerelemente	133
	Makro einer Befehlsschaltfläche zuweisen.....	133
	Kombinationsfeld und Listenfeld zur Auswahl nutzen.....	133
	Kontrollkästchen	135
	Drehfeld und Bildlaufleiste.....	136
	Weitere Steuerelementeigenschaften.....	136
5.4	ActiveX-Steuerelemente	138
	Einige Eigenschaften des ActiveX-Kombinationsfeldes	138
	Ein ActiveX-Kombinationsfeld per VBA füllen	140

6 **Grundlagen zu UserForms** **143**

6.1	Vorbereitungen im VBA-Editor	145
6.2	Ein Formular anlegen	146
	UserForm einfügen.....	146
	UserForm-Eigenschaften.....	148
	Präfixe für Objektnamen	150

6.3	Die Elemente der Werkzeugsammlung	151
	Beschriftungsfeld (Label)	152
	Textfeld (TextBox)	153
	Kontrollkästchen (CheckBox).....	154
	Optionsfeld (OptionButton).....	154
	Rahmen (Frame).....	155
	Kombinationsfeld (ComboBox)	155
	Listenfeld (ListBox)	156
	Befehlsschaltfläche (CommandButton).....	157
	Umschaltfläche (ToggleButton).....	157
	Bildlaufleiste (ScrollBar)	157
	Drehfeld (SpinButton)	158
	Anzeige (Image)	158
	Register (TabStrip)	159
	Multiseite (Page).....	159
6.4	Formularprozeduren	160
	Formularcode anzeigen.....	160
	Formularereignisse.....	161
	Adressierung des Formulars	161
6.5	Eigenschaften von Steuerelementen	162
	Beschriftungsfeld (Label) verwenden	162
	Textfeld (TextBox) verwenden	163
	Optionsfeld (OptionButton) verwenden	163
	Rahmen (Frame) verwenden	164
	Kontrollkästchen (CheckBox) verwenden.....	164
	Kombinationsfeld (ComboBox) verwenden.....	165
	Synchronisation zwischen Kombinationsfeld und Listenfeld	169
	Listenfeld (ListBox) verwenden.....	170
	Befehlsschaltfläche (CommandButton) verwenden	175
	Umschaltfläche (ToggleButton) verwenden.....	176
	Bildlaufleiste (ScrollBar) verwenden	177
	Drehfeld (SpinButton) verwenden	177
	Anzeige (Image) verwenden.....	178
	Multiseiten (Page) verwenden	179
6.6	Allgemeine Tipps zu UserForms	182
	Aufruf eines Formulars.....	182
	Verlassen des Formulars.....	183
	Aktivierreihenfolge der Steuerelemente.....	184
	Die Accelerator-Eigenschaft.....	185
	Gruppieren von Steuerelementen.....	186
	Linienelemente und Einrahmungen.....	188

7 **Formulare als Dialogelemente einsetzen 191**

7.1	Die individuelle MessageBox.....	192
7.2	Schaltflächen - die individuelle Schaltzentrale.....	193
7.3	Nutzername und Passwort abfragen	195
	Formular zur Benutzeranmeldung erstellen.....	195
	Passworteingabe und -überprüfung.....	196
7.4	Eingabewerte aus Textfeldern verwenden	197
	Summe berechnen und anzeigen	197
	Ein Formular für Berechnungen auf Tabellenebene	198
	Ergebnis in das Arbeitsblatt einfügen.....	201
7.5	Speichern von Zahlen aus Text- und Kombinationsfeldern	203
	Zahlen aus Textfeldern speichern.....	203
	Werte aus Kombinationsfeldern speichern.....	204
	Übersicht Konvertierungsfunktionen.....	206
7.6	Optionsfelder können entscheiden	207
	Einfache Optionsfelder	207
	Optionsfelder mit verdeckter Vorgabe.....	209
7.7	Datenanzeige in der Eingabemaske	209

8 **Workshop: Die individuelle Benutzeroberfläche 213**

8.1	Ein Formular als Eingabemaske für Tabellen erstellen.....	214
	Übersicht und Layout.....	214
	Textfelder einfügen, Elemente ausrichten.....	216
	Beschriftungsfelder hinzufügen.....	218
	Optionsfelder in Gruppen einfügen.....	218
	Kontrollkästchen einfügen	219
	Kombinationsfelder einfügen	220
	Listenfeld einfügen	221
	Bildlaufleiste einfügen	221
	Befehlsschaltflächen einfügen.....	222
	Foto oder Logo einbinden.....	222
	Ergänzungen am Layout	223
8.2	Die Eingabemaske für Tabellendaten aktivieren	224
	Vorbereitungen zum Start der Eingabemaske.....	224
	Eingabemaske starten.....	227
	Allgemeine Tipps und Hinweise zur Eingabe von Prozeduren	229

	Voreinstellungen für Eingabefelder vorgeben.....	230
	Die Reihenfolge bei der Eingabe (Tab-Taste) festlegen	234
	Alle Einträge in der Eingabemaske zurücksetzen (Reset).....	236
	Das Alter automatisch berechnen	237
	Pflichtfelder vor Datenübernahme prüfen	238
8.3	Exkurs: Plausibilitätsprüfung (Validierung)	239
	Prüffunktionen.....	240
	Individuelle Prüfprozeduren	242
	Ereignisgesteuerte Überprüfung	244
	Zeichenfolgen (Muster) überprüfen	247
	Eingaben einschränken	250
8.4	Daten aus der Eingabemaske in die Tabelle übertragen.....	252
	Den ersten Datensatz aus der Eingabemaske übernehmen.....	252
	Die Tabelle wächst.....	255
8.5	Daten aus der Tabelle in der Eingabemaske anzeigen	260
	Namen im Listenfeld der Eingabemaske anzeigen	261
	Ausgewählte Tabellenzeilen in der Eingabemaske anzeigen.....	262
	Daten in der Tabelle korrigieren bzw. überschreiben.....	267
	Eingabemaske testen	269

9 Beispielformulare für besondere Aufgaben 271

9.1	Formelassistent.....	272
9.2	Kopierassistent	275
9.3	Formular als VBA-Wissenstest	278
9.4	Grammatik-Trainer	280
9.5	Formular mit Multiple-Choice-Fragen	282
9.6	Farbmischpult	286
9.7	Dateiauswahl über ein Formular	288

Anhang:

Nützliche Tastenkombinationen im VBA-Editor	291
Stichwortverzeichnis.....	293

1 Vorbereitungen

In diesem Kapitel lernen Sie...

- Allgemeine Begriffe
- Im Menüband das Register Entwicklertools anzeigen
- Arbeitsmappen mit Makros speichern und öffnen
- Sicherheitseinstellungen beim Umgang mit Makros

Das sollten Sie bereits wissen

- Dateien speichern und öffnen
- Grundkenntnisse Excel

1.1 Zum grundlegenden Verständnis von Makros und VBA

Wozu Makros und VBA?

Obwohl Microsoft Excel ein äußerst leistungsfähiges Programm mit zahlreichen Funktionen ist, werden häufig für spezielle Probleme Lösungen benötigt, die sich mit den Standardfunktionen von Excel nur mit einigem Aufwand oder überhaupt nicht realisieren lassen. Dazu zählen wiederkehrende Routinearbeiten wie z. B. Formatierung von Zellen oder Berechnungen, aber auch komfortable Dateneingabemöglichkeiten mittels eigens dafür erstellter Dialogfenster. Außerdem lassen sich mit Makros und VBA Arbeitsmappen und Tabellen zurechtbasteln, die auch für ungeübte Nutzer einfach und vor allem sicher zu bedienen sind, z. B. zur Dateneingabe.

Ein weiterer Einsatzschwerpunkt ist die Erstellung eigener Funktionen für häufig benötigte Formeln. So lassen sich beispielsweise bei mehrfach abzufragenden Bedingungen statt verschachtelter WENN-Funktionen alle Anweisungen komfortabel und wesentlich übersichtlicher als Programmanweisung schreiben und bei Bedarf auch in mehreren Arbeitsmappen nutzen.

Wichtige Begriffe

Makros

Bereits seit frühen Versionen von Excel besteht die Möglichkeit, Befehlsabläufe mit Hilfe des integrierten Makrorecorders aufzuzeichnen. Die aufgezeichnete Befehlsfolge wird als Makro bezeichnet, unter einem Namen gespeichert und jedes Mal ausgeführt, wenn Sie das Makro aufrufen. Zu den wichtigsten Einsatzmöglichkeiten von Makros gehört die Ausführung von Routinetätigkeiten, beispielsweise Aufbereitung und Auswertung von Tabellen mit gleichbleibendem Aufbau, aber wechselnden Daten.

Mit dem Makrorecorder aufgezeichnete Makros besitzen Vor- und Nachteile: Sie erfordern keinerlei Programmierkenntnisse, sind allerdings wenig flexibel und reagieren nicht auf Bedingungen.

Mit der Aufzeichnung eines Makros wird eigentlich ein Computerprogramm in der Programmiersprache VBA erstellt. Ein Makro kann jederzeit nachträglich bearbeitet und beispielsweise um weitere Anweisungen ergänzt werden, VBA-Kenntnisse vorausgesetzt.

Die Programmiersprache VBA

Microsoft Office verfügt unter der Bezeichnung VBA (Visual Basic for Applications) über eine integrierte Programmiersprache mit beträchtlichem Sprachumfang. Diese ist in allen Office-Anwendungen, also z. B. auch in Word, Access oder PowerPoint verfügbar, wird aber hauptsächlich für Access und Excel genutzt. Zudem basieren die einzelnen Anwendungen auf unterschiedlichen Objekten, sodass sich mit Ausnahme grundlegender Sprachelemente VBA in Word durchaus von Excel-VBA unterscheidet.

Anstelle der Aufzeichnung mit dem Makrorecorder kann ein Makro auch komplett als Folge von Anweisungen geschrieben werden, diese werden in der Regel als Prozeduren bezeichnet. Allerdings ist VBA eine sehr komplexe Sprache und wie bei allen Programmiersprachen, gelten auch für die Programmierung mit VBA feste Regeln für den Aufbau der Befehle, die sogenannte Sprachsyntax. Im Gegenzug stehen Ihnen dafür mit VBA nahezu alle Möglichkeiten der Programmierung offen.

Programmanweisungen werden in der Regel in einer gesonderten Entwicklungsumgebung mit entsprechenden Eingabehilfen und Befehlen geschrieben. Microsoft Office verfügt zu diesem Zweck über den integrierten VBA-Editor, der in einem eigenen Fenster geöffnet wird.

Gibt es einen Unterschied zwischen Makros und Prozeduren?

Nein, sowohl bei aufgezeichneten Makros als auch bei selbst geschriebenen Prozeduren handelt es sich um VBA-Anweisungen, die im VBA-Editor jederzeit eingesehen und geändert werden können.

Tipp für Einsteiger und Gelegenheitsprogrammierer

Wenn Sie beispielsweise nicht genau wissen, wie in VBA ein bestimmtes Excel-Objekt angesprochen oder die gewünschte Aktion bezeichnet wird, dann zeichnen Sie ein einfaches Makro auf, in dem Sie genau diese Aktion ausführen. Diese Zeile kopieren Sie dann einfach über die Zwischenablage in Ihre Prozedur.

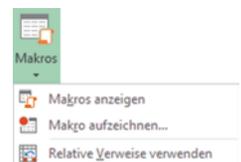
Was versteht man unter objektorientierter Programmierung?

Objektorientierte Programmiersprachen, dazu zählt auch VBA, behandeln alle Dinge der realen Welt als Objekte. Jedes Objekt verfügt über bestimmte Eigenschaften, z. B. Farbe, viele Objekte können auch Aktionen ausführen, diese werden als Methoden bezeichnet. Manche Objekte verfügen auch noch über sogenannte Ereignisse. Typische Excel-Objekte sind die Anwendung Excel selbst, eine Arbeitsmappe, ein Tabellenblatt, Zellbereiche oder Diagramme. Ereignisse sind dagegen z. B. das Öffnen oder Schließen einer Arbeitsmappe, zu dem Methoden zählen Aktionen wie Kopieren, Einfügen oder Löschen.

Bis auf wenige Ausnahmen, z. B. Formatierungen, entspricht jede Programmzeile einer Anweisung.

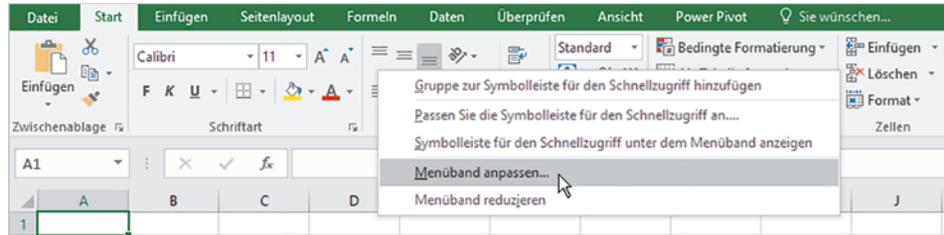
1.2 Das Register Entwicklertools anzeigen

Zum Umgang mit Makros und VBA benötigen Sie einige zusätzliche Befehle und Schaltflächen im Menüband. Zwar enthält das Register *Ansicht* ▶ *Makros* eine Schaltfläche mit Befehlen zum Aufzeichnen und Ausführen von Makros, wesentlich komfortabler arbeiten Sie allerdings mit dem Register *Entwicklertools*. Wenn Sie außerdem auch noch Steuerelemente in Tabellenblättern verwenden möchten, wie in diesem Buch in Kapitel 5 beschrieben, dann ist das Register *Entwicklertools* unverzichtbar.



Leider ist das Register *Entwicklertools* nicht standardmäßig sichtbar. Zum Anzeigen klicken Sie mit der rechten Maustaste an eine beliebige Stelle im Menüband und auf *Menüband anpassen....* Der etwas umständlichere Weg führt über das Register *Datei*, den Befehl *Optionen* und die Auswahl *Menüband anpassen*.

Bild 1.1 Menüband anpassen



Das Dialogfenster *Excel-Optionen* mit der Auswahl *Menüband anpassen* öffnet sich. Aktivieren Sie hier in der rechten Liste unterhalb von *Menüband anpassen*, *Hauptregisterkarten* das Kontrollkästchen der Registerkarte *Entwicklertools* und übernehmen Sie die Einstellung mit *OK*.

Bild 1.2 Entwicklertools anzeigen

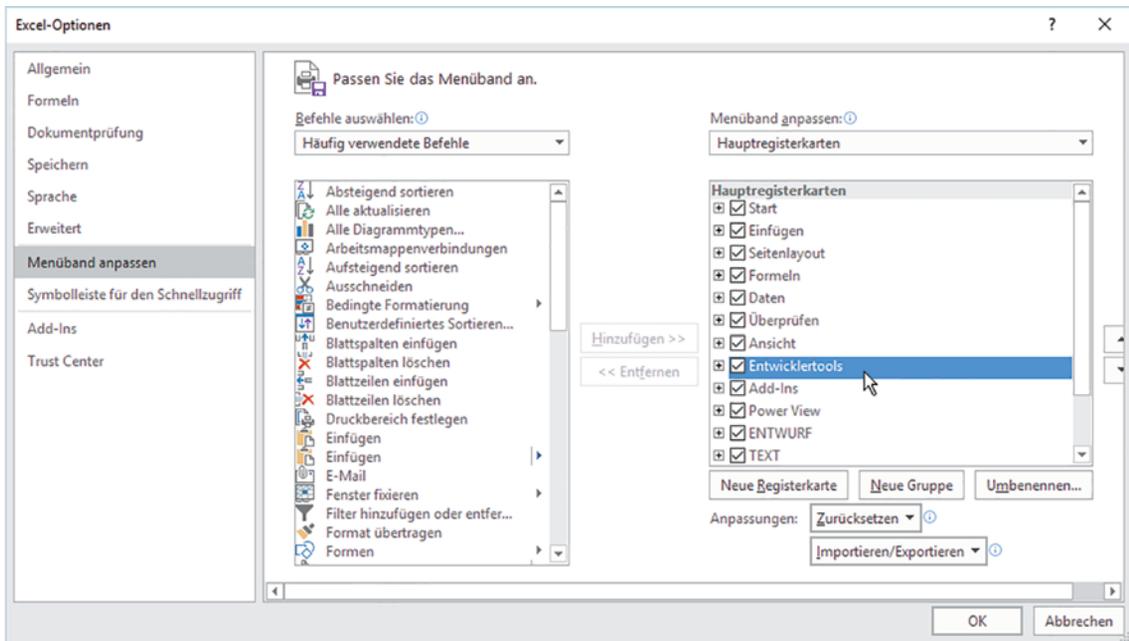


Bild 1.3 Das Register Entwicklertools

Die Registerkarte erscheint im Menüband rechts vom letzten Standardregister.



1.3 Arbeitsmappen mit Makros speichern

Der Dateityp Excel-Arbeitsmappe mit Makros (.xlsm)

Im Gegensatz zu früheren Versionen verwendet Excel seit der Version 2007 einen eigenen Dateityp zum Speichern von Arbeitsmappen, die Makros oder VBA-Prozeduren enthalten. Beim Speichern einer solchen Arbeitsmappe muss daher unbedingt der Dateityp *Excel-Arbeitsmappe mit Makros* (.xlsm) ausgewählt werden. Dazu verwenden Sie eine der beiden folgenden Möglichkeiten:

- ▶ Klicken Sie entweder im Fenster *Speichern unter* im Feld *Dateityp* auf den Dropdown-Pfeil und wählen Sie *Excel-Arbeitsmappe mit Makros* (Bild unten).
- ▶ Oder klicken Sie im Register *Datei* auf *Exportieren* und hier auf *Dateityp ändern*. Klicken Sie dann auf *Arbeitsmappe mit Makros*.

Dies gilt nicht, wenn die Arbeitsmappe im älteren Dateiformat Excel-97-2003-Arbeitsmappe (.xls) gespeichert wurde.

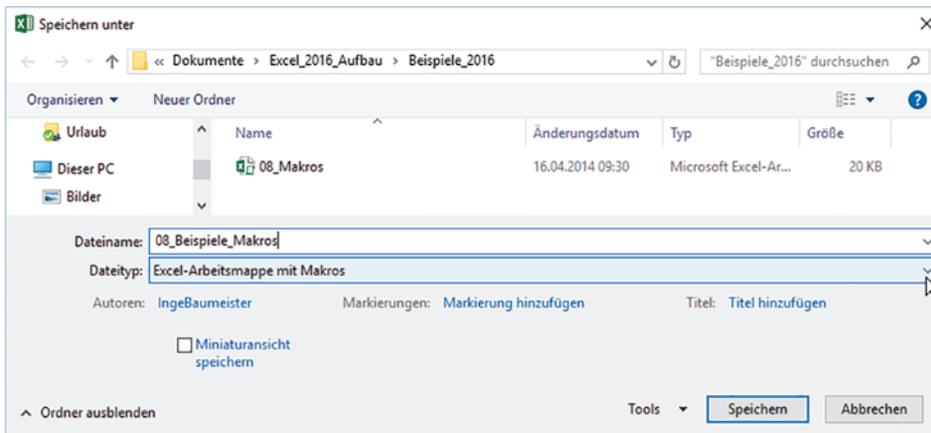
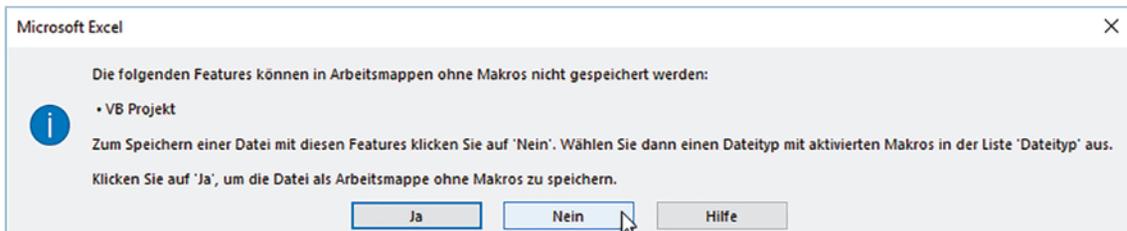


Bild 1.4 Speichern - Dateityp auswählen

Hinweis: Sollten Sie für eine Arbeitsmappe mit vorhandenen Makros versehentlich den Dateityp Excel-Arbeitsmappe (.xlsx) gewählt haben, so erhalten Sie beim Speichern eine Warnung, dass Ihre Makros nicht gespeichert werden. Klicken Sie auf *Nein* und wählen Sie den passenden Dateityp aus. Achtung: Wenn Sie dagegen auf *Ja* klicken, werden alle vorhandenen Makros aus der Mappe entfernt.

Bild 1.5 Klicken Sie auf *Nein* und wählen Sie den passenden Dateityp



Die persönliche Makroarbeitsmappe

In der Regel werden Makros zusammen mit derjenigen Arbeitsmappe gespeichert, in der sie erstellt wurden und stehen somit nur hier zur Verfügung.

Ausnahme: Wenn Sie eine versehentlich erstellte Makroarbeitsmappe mit nicht benötigten Makros löschen möchten. Dies ist problemlos möglich.

Falls ein Makro in allen Excel-Arbeitsmappen verfügbar sein soll, kann es in einer gesonderten Datei, der Persönlichen Makroarbeitsmappe gespeichert werden. Diese befindet sich unter dem Namen *PERSONAL.xlsb* zusammen mit den übrigen Benutzereinstellungen im Ordner C:\Users\Benutzername\AppData\Roaming\Microsoft\Excel\XLStart (Windows 10), der genaue Speicherort ist abhängig vom Betriebssystem. Ein Zugriff auf die Makroarbeitsmappe über den Datei-Explorer ist in der Regel nicht nötig, da Sie alle, in dieser Mappe enthaltenen Makros im VBA-Editor anzeigen, bearbeiten und bei Bedarf auch löschen können. Zudem ist der Ordner *AppData* standardmäßig ausgeblendet,

Beachten Sie außerdem: Wenn Sie als Speicherort die persönliche Makroarbeitsmappe gewählt haben, dann müssen Sie auch alle Änderungen an dieser Mappe speichern. Beim Beenden von Excel erscheint ein entsprechender Hinweis, siehe Bild unten. Enthält in einem solchen Fall die Arbeitsmappe selbst keine weiteren Makros, dann muss sie auch nicht mit dem Dateityp *Excel-Arbeitsmappe mit Makros* gespeichert werden.

Bild 1.6 Änderungen der persönlichen Makroarbeitsmappe speichern



1.4 Sicherheitseinstellungen

Ein Office-Dokument kann Schadsoftware in Form von Makros enthalten

Die Programmiersprache VBA ist in alle Microsoft Office-Anwendungen, also auch Word, PowerPoint oder Access integriert und stellt eine äußerst leistungsfähige Sprache dar, die auch auf wichtige Funktionen Ihres Systems, beispielsweise die Dateiverwaltung zugreifen kann. Makros können daher durchaus auch eine Bedrohung für die Sicherheit Ihres Computers darstellen, schädliche Makros sind unter der Bezeichnung Makroviren bekannt.

Sicherheitseinstellungen kontrollieren

Im Gegensatz zu anderer Schadsoftware werden Makroviren zusammen mit Office-Dokumenten gespeichert und verbreitet. Beim Öffnen eines infizierten Dokuments wird auch das Makrovirus aktiviert. Makroviren werden von gängigen Antivirenprogrammen nicht immer erkannt, daher verfügt Excel über eigene Sicherheitseinstellungen zum Umgang mit Makros.

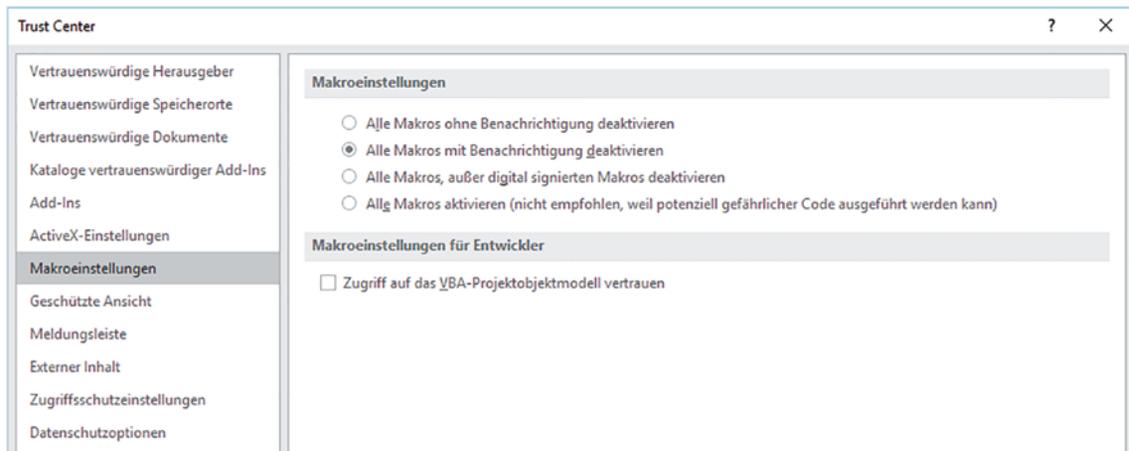
Zur Kontrolle und ggf. zum Ändern der Sicherheitseinstellungen klicken Sie im Register *Entwicklertools*, Gruppe *Code*, auf die Schaltfläche *Makrosicherh.* und öffnen damit das *Trust Center* mit den Makroeinstellungen.

Als Alternative öffnen Sie das *Trust Center* über das Register *Datei* und die Excel *Optionen*. Klicken Sie hier auf *Trust Center* und auf die Schaltfläche *Einstellungen für das Trust Center*.



Damit Makros bzw. VBA-Prozeduren später ausgeführt werden können, dürfen sie nicht komplett und ohne Hinweis deaktiviert werden, wählen Sie daher unter *Makroeinstellungen* die Option *Alle Makros mit Benachrichtigung deaktivieren*. Dies ist auch die empfohlene Standardeinstellung.

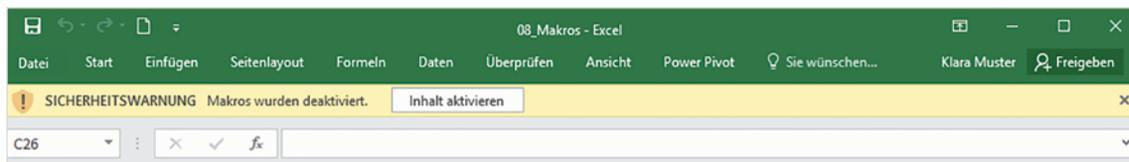
Bild 1.7 Trust Center - Makroeinstellungen



Arbeitsmappe mit Makros öffnen

Mit der Einstellung *Alle Makros mit Benachrichtigung deaktivieren* erhalten Sie beim Öffnen einer Arbeitsmappe, die Makros enthält, eine Sicherheitswarnung und können entscheiden, ob Sie die Makros aktivieren möchten. Klicken Sie dazu auf die Schaltfläche *Inhalt aktivieren*. Wenn Sie beim Öffnen die Inhalte bzw. Makros einer Arbeitsmappe aktiviert haben, dann wird diese Mappe künftig von Excel als vertrauenswürdige Dokument eingestuft und die Inhalte werden beim nächsten Öffnen automatisch aktiviert. Ausnahme: wenn Sie die Mappe zwischenzeitlich an einen anderen Ort verschoben oder umbenannt haben, dann müssen Sie beim nächsten Öffnen den Inhalt erneut aktivieren.

Bild 1.8 Makros beim Öffnen aktivieren

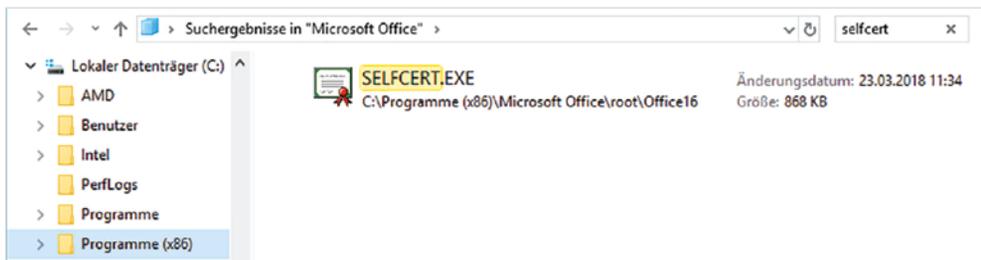


Makros digital signieren

Wenn Sie möchten, können Sie Makros mit Hilfe eines Zertifikats digital signieren. Ein solches digitales Zertifikat kann von einer kommerziellen Zertifizierungsstelle, in größeren Firmen auch vom internen Sicherheitsadministrator bezogen werden. Zu Testzwecken oder wenn nur Sie selbst mit Makros arbeiten, können Sie auch ein selbst signiertes Zertifikat erstellen. Dieses lässt sich allerdings nicht übertragen und besitzt somit nur auf dem Gerät Gültigkeit, auf dem es erstellt wurde. So gehen Sie vor:

- 1 Öffnen Sie im Datei-Explorer den Ordner C:\Programme (x86)\Microsoft Office\root\ und starten Sie hier mit Doppelklick die Anwendung SELFCERT.EXE. Achtung: der genaue Speicherort kann auf Ihrem Gerät etwas abweichen, in diesem Fall durchsuchen Sie den Ordner Microsoft Office.

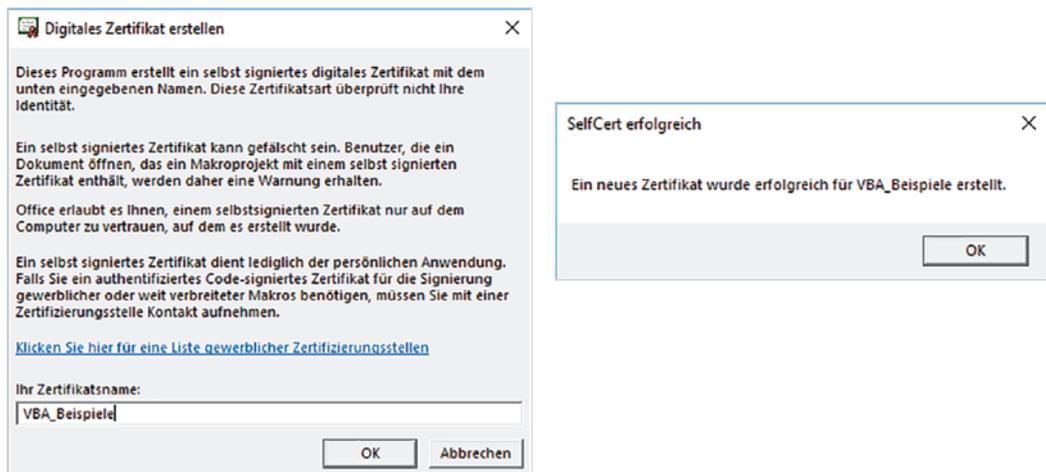
Bild 1.9 Datei suchen



Tipp: Über einen Link gelangen Sie zu einer Liste gewerblicher Zertifizierungsstellen.

- 2 Das Fenster *Digitales Zertifikat erstellen* öffnet sich. Geben Sie im Feld *Ihr Zertifikatsname* einen aussagefähigen Namen für Ihr Zertifikat ein und klicken Sie auf *OK*. Schließen Sie die Meldung, dass mit SelfCert erfolgreich ein neues Zertifikat erstellt wurde, mit Klick auf *OK*.

Bild 1.10 Geben Sie einen Namen ein



Das Zertifikat kann nun zum Signieren von Makros verwendet werden. Dazu klicken Sie im VBA-Editor (siehe Kapitel 3) auf das Menü *Extras* ▶ *Digitale Signatur...* und wählen mit Klick auf die Schaltfläche *Wählen...* die gewünschte Signatur aus.

2

Makros aufzeichnen und ausführen

In diesem Kapitel lernen Sie...

- Einfache Makros mit dem Makrorecorder aufzeichnen
- Speicherort von Makros wählen
- Zellbezüge in Makros
- Makros ausführen
- Die Diagrammanzeige per Makro steuern

Das sollten Sie bereits wissen

- Umgang mit Excel-Arbeitsmappen und Tabellenblättern
- Zellbezüge und Formeln

2.1 Hinweise zum Umgang mit aufgezeichneten Makros

Wie bereits erwähnt, erlaubt der Makrorecorder von Excel das Erstellen von Makros ohne Programmierkenntnisse. Er zeichnet alle Anweisungen und Eingaben in Form von VBA-Befehlen auf, die Sie später bei Bedarf im VBA-Editor ansehen und bearbeiten können.

Die Begriffe Makros und Prozedur meinen im Grunde immer dasselbe, nämlich ein VBA-Programm das bestimmte Aufgaben ausführt. Allerdings wird zur besseren Unterscheidung der Begriff Makros häufig nur für, mit dem Makrorecorder aufgezeichnete, Makros verwendet.

Leider zeichnet der Makrorecorder alle Arbeitsschritte und Eingaben, also auch misslungene Versuche auf. Überlegen Sie daher vor der Aufzeichnung, welche Arbeitsschritte in welcher Reihenfolge erforderlich sind und testen Sie die Schritte eventuell vorher. Auch ein kurzes Notieren der richtigen Reihenfolge kann hilfreich sein.

Bevor wir mit dem ersten Makro beginnen, einige Hinweise zu Makrorecorder. Dieser ist auch für fortgeschrittene Anwender in VBA-Kenntnissen nicht ganz überflüssig. Besonders dann, wenn einzelne VBA-Anweisungen nicht bekannt sind, ist es manchmal schneller, einzelne Befehle aufzuzeichnen und diese dann im VBA-Editor zu bearbeiten oder in die eigene Prozedur einzufügen.

Allerdings ist der aufgezeichnete VBA-Code meist wesentlich umständlicher als selbst geschriebene Prozeduren und enthält häufig auch überflüssige Anweisungen.

Zuletzt noch ein Tipp: Durch ein Makro ausgeführte Schritte können nicht mehr rückgängig gemacht werden. Speichern Sie also beim Testen von Makros die Arbeitsmappe, bevor Sie ein Makro ausführen. Dadurch vermeiden Sie, dass zwischenzeitlich vorgenommene Änderungen verlorengehen. Außerdem kann ein fehlerhaftes Makro Excel zum Absturz bringen.

2.2 Ein einfaches Makro aufzeichnen

Die Aufzeichnung starten

Als Beispiel wollen wir ein Makro aufzeichnen, das den Text „Hallo“ in eine zuvor markierte Zelle schreibt. So gehen Sie vor:

- 1 Markieren Sie eine Zelle, beispielsweise A1 und klicken Sie im Register *Entwicklertools*, Gruppe *Code*, auf *Makro aufzeichnen*.

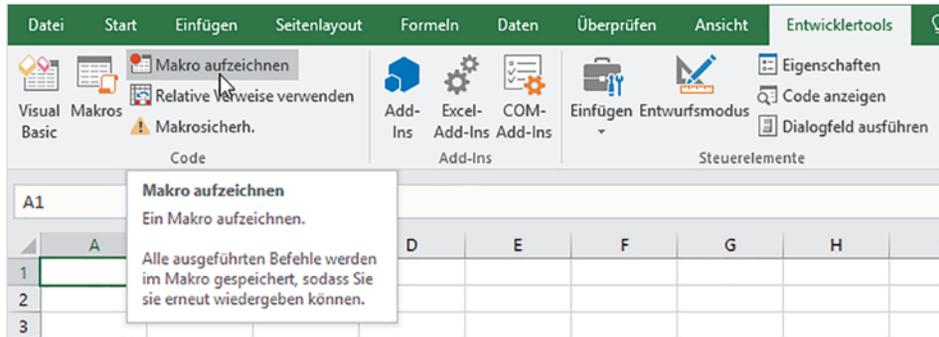


Bild 2.1 Klicken Sie auf Makro aufzeichnen

- Das Dialogfenster *Makro aufzeichnen* wird geöffnet: Geben Sie einen Namen an, unter dem das Makro gespeichert und später aufgerufen werden soll.

Beachten Sie die Regeln für Makronamen: Ein Makroname muss mit einem Buchstaben beginnen, darf maximal 255 Zeichen lang sein und keine Leerzeichen und mit Ausnahme des Unterstrichs (_) auch keine Sonderzeichen enthalten, also auch keinen Bindestrich.

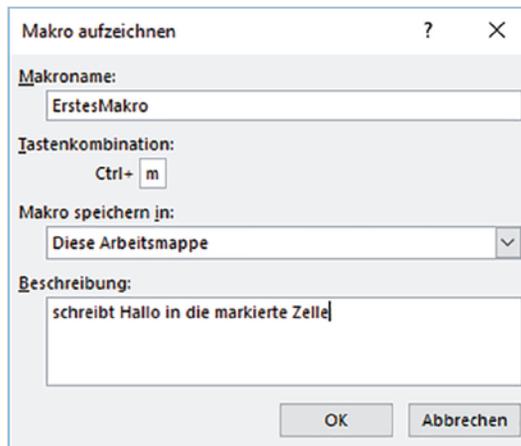


Bild 2.2 Makroaufzeichnung starten

3 Tastenkombination zum Starten des Makros

Falls Sie später das Makro über eine Tastenkombination starten möchten, so geben Sie die gewünschte Taste in Verbindung mit der Strg (Ctrl)-Taste an. Sie können ein Makro aber auch auf andere Weise starten oder dem Makro nachträglich eine Tastenkombination zuweisen.

Achtung: Die Tastenkombination unterscheidet zwischen Groß- und Kleinbuchstaben. Zudem sollten Sie eine Tastenkombination wählen, die nicht bereits anderweitig belegt ist, z. B. m, M, j, J.

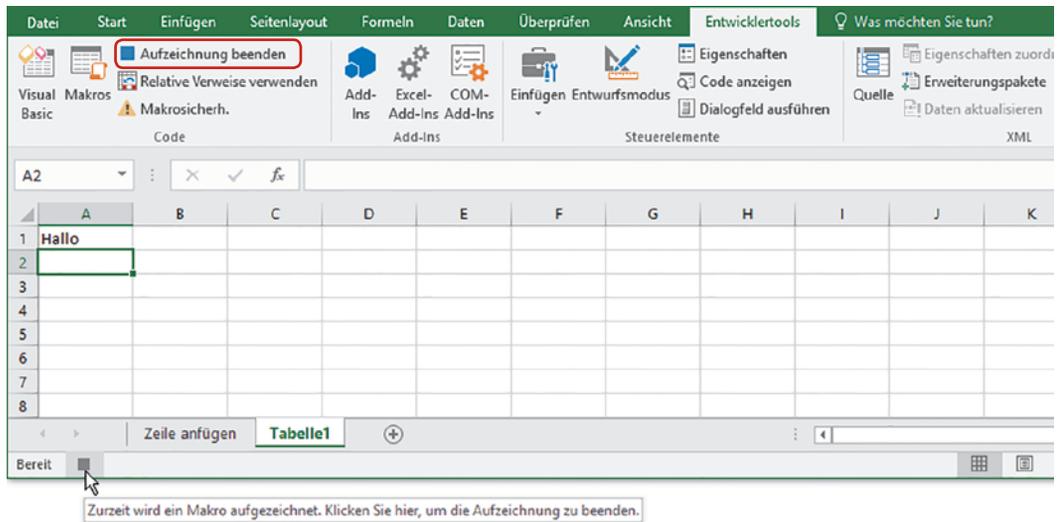
4 Speicherort festlegen

Unter *Makro speichern in* wählen Sie den Speicherort des Makros. Hier haben Sie die Wahl zwischen der Persönlichen Makroarbeitsmappe und der aktuellen Ar-

Diese Beschreibung erscheint später im VBA-Editor als Kommentar, siehe Kap. 3.

- 5 Im Feld *Beschreibung* können Sie optional eine kurze Beschreibung der Funktionsweise des Makros eingeben. Dies ist im Hinblick auf die spätere Nachvollziehbarkeit unbedingt zu empfehlen.
- 6 Mit der Schaltfläche *OK* starten Sie abschließend die Makroaufzeichnung.
- 7 Ab jetzt werden alle Ihre Befehle und Eingaben aufgezeichnet. Da in unserem Beispiel die Zelle A1 bereits vor der Makroaufzeichnung markiert wurde, tippen Sie das Wort „Hallo“ in diese Zelle und drücken anschließend die Eingabe-Taste. Nun ist die Zelle A2 unterhalb markiert.
- 8 Zum Schluss müssen Sie die Makroaufzeichnung beenden: Klicken Sie dazu im Register *Entwicklertools*, Gruppe *Code*, auf *Aufzeichnung beenden*. Als Alternative können Sie auch die Schaltfläche in der Statusleiste verwenden, siehe Bild unten.

Bild 2.3 Aufzeichnung beenden



Makro ausführen

Um das soeben aufgezeichnete Makro zu testen, löschen Sie zuvor das Wort „Hallo“ aus A1 und markieren dann eine beliebige andere Zelle des Tabellenblatts.

- ▶ Haben Sie dem Makro eine Tastenkombination zugewiesen, so verwenden Sie zum Starten des Makros diese Tastenkombination.
- ▶ Andernfalls klicken Sie im Register *Entwicklertools* ▶ *Code* auf die Schaltfläche *Makros*. Markieren Sie im Fenster *Makro* das Makro, das Sie ausführen möchten und klicken Sie auf die Schaltfläche *Ausführen*. Als Alternative öffnen Sie das Fenster *Makro* mit den Tasten Alt+F8.

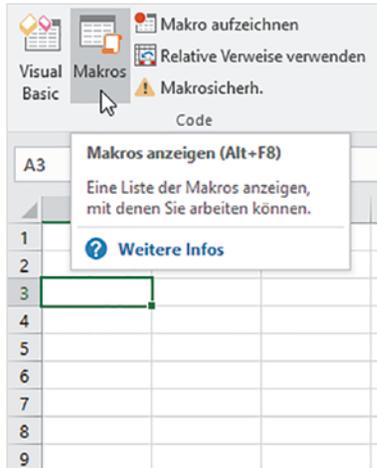


Bild 2.4 Makros anzeigen

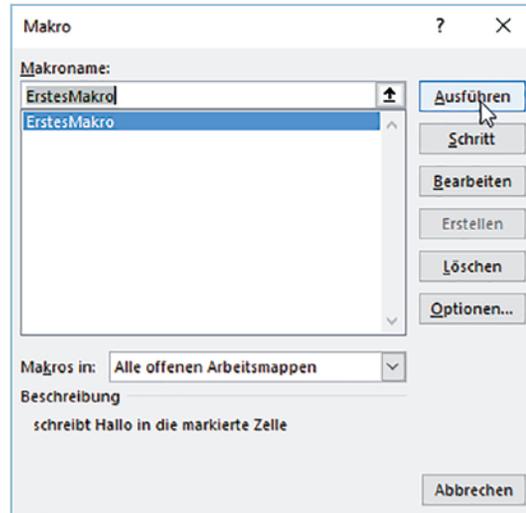


Bild 2.5 Ausführung starten

Das zuvor aufgezeichnete Makro schreibt nun das Wort „Hallo“ in die aktuell markierte Zelle und markiert anschließend die Zelle A2.

Markieren Sie anschließend eine andere, beliebige Zelle und führen Sie das Makro nochmals aus. Haben Sie bemerkt, dass beim Ausführen dieses Makros zwar der Text in die aktuell markierte Zelle geschrieben, anschließend aber immer dieselbe Zelle, nämlich A2 markiert wird? Die Ursache liegt darin, dass bei der Makroaufzeichnung standardmäßig absolute Zellbezüge verwendet werden. Weiter unten erfahren Sie, wie Sie ein Makro mit relativen Zellbezügen aufzeichnen.

Das Makro verwendet absolute Zellbezüge!

Mögliche Probleme bei der Makro-Ausführung

Makros wurden deaktiviert

Möglicherweise erscheint beim Starten des Makros eine Meldung, die Sie darauf aufmerksam macht, dass Makros aufgrund der Sicherheitseinstellungen deaktiviert wurden, gleichzeitig ist der Entwurfsmodus aktiviert. In diesem Fall müssen Sie die Mappe schließen und erneut öffnen und dann die Inhalte aktivieren (siehe Kapitel 1.4).

Laufzeitfehler - Aufzeichnung wurde nicht beendet

Als weitere mögliche Fehlerquelle haben Sie vielleicht vergessen, die Aufzeichnung zu beenden, bevor Sie das Makro zum ersten Mal ausgeführt wurde. Dann erscheint entweder sofort oder beim nächsten Aufruf die Fehlermeldung *Laufzeitfehler*.



Bild 2.6 Laufzeitfehler

Sie haben bei Laufzeitfehlern folgende Möglichkeiten:

- ▶ *Debuggen* bedeutet, die Makroausführung wird unterbrochen und der VBA-Programmcode im VBA-Editor angezeigt, wobei die fehlerhafte Anweisung gelb hervorgehoben ist.
- ▶ Mit Klick auf die Schaltfläche *Beenden* wird die Makroausführung abgebrochen und Sie können anschließend das Makro neu aufzeichnen.

Kontrollieren Sie anschließend in jedem Fall, ob die Makroaufzeichnung noch läuft und beenden Sie diese, falls ja.

Makro löschen oder ersetzen

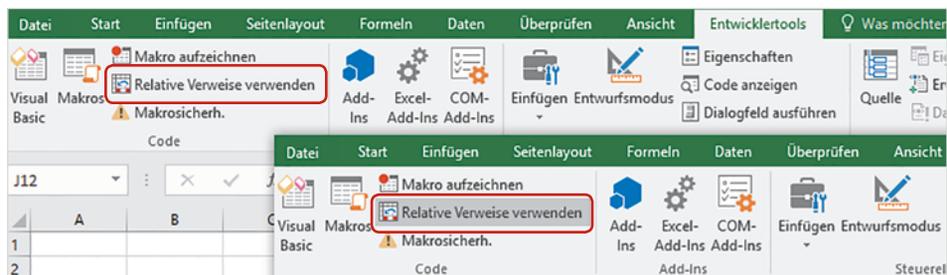
Zum Löschen eines fehlerhaften oder nicht mehr benötigten Makros, öffnen Sie über die Schaltfläche *Makros* das Dialogfenster *Makro*. Markieren Sie das Makro, das Sie löschen möchten und klicken Sie auf die Schaltfläche *Löschen*. Sie können aber auch ein fehlerhaftes Makro einfach neu aufzeichnen. Dazu geben Sie bei der Neuaufzeichnung als Namen einfach den Namen des zu ersetzenden Makros ein. Bestätigen Sie die nachfolgende Meldung, ob Sie das Makro ersetzen möchten, mit *OK*.

2.3 Zellbezüge in Makros

Wie Sie bei Ihrem ersten Makro gesehen haben, unterscheidet Excel nicht nur in Formeln, sondern auch bei der Makroaufzeichnung zwischen relativen und absoluten (festen) Zellbezügen. Im ersten Beispiel haben Sie ein Makro mit festen Zellbezügen aufgezeichnet. Feste Zellbezüge sind immer dann erforderlich, wenn eine Eingabe immer an derselben Position erfolgen soll. Soll dagegen die Eingabe beispielsweise am Ende einer Liste erfolgen, unabhängig davon, wie viele Zeilen die Liste umfasst, dann benötigen Sie bei der Aufzeichnung relative Zellbezüge.

Über die Schaltfläche *Relative Verweise verwenden* im Register *Entwicklertools* können Sie vor und während der Aufzeichnung zwischen festen und relativen Zellbezügen wechseln: Bei deaktivierter Schaltfläche werden alle Aktionen wie Markieren und Eingeben mit absoluten Zellbezügen aufgezeichnet. Ist hingegen die Schaltfläche aktiviert, erfolgt die Aufzeichnung mit relativen Zellbezügen.

Bild 2.7 Absolute und relative Zellbezüge



Beispiel: Makro mit unterschiedlichen Verweisen aufzeichnen

Als zweites Beispiel soll ein Makro aufgezeichnet werden, das für jeden neuen Rechnungsposten in der oberen Tabelle (*Neuer Rechnungsposten*) den Gesamtpreis in E3 berechnet, anschließend die Werte der Zellen A3 bis E3 ausschneidet und am Ende der unteren Liste *Vorhandene Rechnungsposten* anfügt.

Im ersten Schritt legen Sie die beiden Tabellen an, wie unten abgebildet. Nehmen Sie auch gleich alle gewünschten Formatierungen vor, insbesondere Zahlenformate.

	A	B	C	D	E	F	G
1	Neuer Rechnungsposten						
2	Artikel-Nr.	Bezeichnung	Kilopreis	Menge	Gesamtpreis		
3	1009	Bananen	5,00	2,00			
4							
5	Vorhandene Rechnungsposten						
6	Artikel-Nr.	Bezeichnung	Kilopreis	Menge	Gesamtpreis		
7	1008	Kartoffeln	3,50	10,00	35,00		
8	1006	Orangen	2,80	6,00	16,80		
9	1123	Äpfel	3,20	20,00	64,00		
10							
11							
12							
13							

Bild 2.8 Die beiden Tabellen Neuer Rechnungsposten und Rechnungsposten

So zeichnen Sie das Makro auf:

- 1 Markieren Sie im Tabellenblatt eine beliebige Zelle, nicht aber E3, da sonst das Markieren dieser Zelle nicht mit aufgezeichnet wird. Starten Sie dann die Makroaufzeichnung und speichern Sie das Makro unter dem Namen *ListeAnfügen*.
- 2 Für den ersten Schritt, die Berechnung der Formel, benötigen Sie absolute Zellbezüge. Achten Sie also darauf, dass die Schaltfläche *Relative Verweise verwenden* nicht aktiviert ist, markieren Sie die Zelle E3 und geben Sie die Formel $=D3*C3$ ein. Schließen Sie die Formeleingabe mit der Enter-Taste oder Klick auf das Symbol *Eingeben* in der Bearbeitungsleiste ab.
- 3 Markieren Sie anschließend den Bereich A3 bis E3 und schneiden Sie mit den Tasten Strg+X den Zellbereich aus.
- 4 Im nächsten Schritt müssen Sie zunächst das Ende der Liste ermitteln. Markieren Sie dazu die Zelle A6 in der ersten Zeile der Liste. Diese Zelladresse bildet den festen Bezugspunkt der Liste, daher benötigen Sie hier ebenfalls einen absoluten Zellbezug.
- 5 Ab jetzt benötigen Sie relative Bezüge, aktivieren Sie also die Schaltfläche *Relative Verweise verwenden*. Um in Spalte A in die letzte Zeile der Tabelle zu gelangen, verwenden Sie die Tastenkombination Strg+Pfeiltaste nach unten. Dann markieren Sie mit der Pfeiltaste die Zelle in der darunterliegenden Zeile und betätigen zum Einfügen aus der Zwischenablage die Tasten Strg+V.
- 6 Im letzten Schritt der Makroaufzeichnung deaktivieren Sie die relative Aufzeichnung wieder und markieren diejenige Zelle, in der Sie mit der Bearbeitung fortfahren möchten, beispielsweise die Zelle A3.



Ab hier müssen relative Verweise aufgezeichnet werden!

Bild 2.9 Nach dem Ende der Aufzeichnung

7 Beenden Sie die Makroaufzeichnung!

	A	B	C	D	E	F	G
1	Neuer Rechnungsposten						
2	Artikel-Nr.	Bezeichnung	Kilopreis	Menge	Gesamtpreis		
3							
4							
5	Vorhandene Rechnungsposten						
6	Artikel-Nr.	Bezeichnung	Kilopreis	Menge	Gesamtpreis		
7	1008	Kartoffeln	3,50	10,00	35,00		
8	1006	Orangen	2,80	6,00	16,80		
9	1123	Äpfel	3,20	20,00	64,00		
10	1009	Bananen	5,00	2,00	10,00		
11							
12							
13							

Testen Sie dann das Makro, indem Sie in Zeile 3 einen weiteren Artikel eingeben und die Ausführung starten. Falls nicht alles auf Anhieb klappt, zeichnen Sie das Makro unter demselben Namen erneut auf.

Tipp: Bei längeren Makros kann es für Ungeübte nützlich sein, vor der Aufzeichnung die einzelnen Schritte testweise durchzugehen und eventuell auf einem Blatt Papier zu notieren.

2.4 Makroausführung starten

Wie Sie ein Makro über das Dialogfenster *Makro* starten, haben Sie bereits kennengelernt. Es gibt aber noch eine Reihe weiterer Möglichkeiten, mit denen Sie ein Makro schneller ausführen und vor allem den Start für ungeübte Excel-Nutzer vereinfachen.

Hinweis: Die folgenden Möglichkeiten gelten nicht nur aufgezeichnete Makros, sondern auch für Prozeduren, die Sie mit VBA schreiben, siehe Kapitel 3 und 4.

Tastenkombination zuweisen

Falls Sie das Makro mit einer Tastenkombination starten möchten, können Sie diese dem Makro bereits vor der Aufzeichnung zuweisen. Über eine Tastenkombination kann ein Makro zwar sehr schnell aufgerufen werden, allerdings bringt diese Methode auch einige Nachteile mit sich.

- ▶ Benutzer, die das Makro verwenden möchten, müssen sich die entsprechenden Tasten merken, dies dürfte für ungeübte Benutzer problematisch sein.
- ▶ Achten Sie bei der Wahl der Tasten darauf, dass die Tastenkombination nicht bereits anderweitig belegt ist. Wichtige Tastenkombinationen wie beispielsweise Strg+C (Kopieren) werden sonst überschrieben.

Vorsicht bei bestehenden Tastenkombinationen

Falls Sie einem Makro nachträglich eine Tastenkombination zuweisen möchten, dann gehen Sie so vor:

- 1 Öffnen Sie das Dialogfenster *Makro*, markieren Sie das Makro, dem Sie eine Tastenkombination zuweisen wollen und klicken Sie auf die Schaltfläche *Optionen*.
- 2 Geben Sie die gewünschte Tastenkombination an, ggf. können Sie hier auch die Beschreibung ändern, und bestätigen Sie mit der Schaltfläche *OK*.

Achtung: Excel unterscheidet hier zwischen Groß- und Kleinbuchstaben!

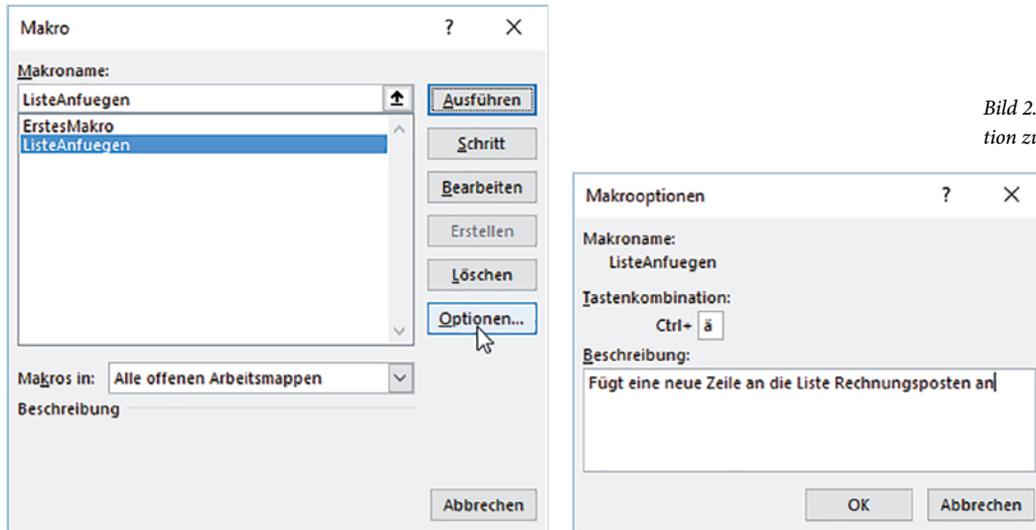


Bild 2.10 Tastenkombination zuweisen

Makro über die Symbolleiste für den Schnellzugriff starten

Als zweite Möglichkeit können Sie das Makro in Form eines Symbols der *Symbolleiste für den Schnellzugriff* hinzufügen und später per Mausklick starten. Klicken Sie dazu am Ende der Schnellstartleiste auf das Symbol *Symbolleiste für den Schnellzugriff anpassen* und hier auf *Weitere Befehle...*



- 1 Das Fenster *Excel-Optionen* mit Anpassungsmöglichkeiten für den Schnellzugriff öffnet sich. Klicken Sie unter *Befehle auswählen* auf den Dropdown-Pfeil und auf *Makros* ①.
- 2 Unterhalb erscheint eine Liste Ihrer aufgezeichneten Makros und in der rechten Liste sehen Sie die vorhandenen Symbole der Schnellzugriffsleiste: Mit einem Doppelklick fügen Sie das markierte Makro ② der Liste hinzu oder benutzen Sie die Schaltfläche *Hinzufügen* ③. Mit der Schaltfläche *Entfernen* können nicht mehr benötigte Makros wieder aus der Schnellzugriffsleiste entfernt werden.
- 3 Legen Sie außerdem fest, ob das Makro bzw. das dazugehörige Symbol in allen Dokumenten sichtbar sein soll (Standardeinstellung *Für alle Dokumente*) oder nur in der aktuellen Arbeitsmappe. Da unser Makro zusammen mit der Arbeits-