

LERNEN EINFACH GEMACHT



2. Auflage

Excel Makros

für **dummies**[®]



Über 100 Makros
direkt nutzen und Zeit
sparen

Verstehen, wie Makros
aufgebaut sind und wie man
sie anpasst

Makros anderer
Arbeitsmappen effizient
einsetzen

Michael Alexander

Excel Makros für Dummies

Schummelseite

Dank der Excel-Tastenkombinationen können Sie bestimmte Aktionen direkt mit der Tastatur auslösen. Sie können so schneller arbeiten, da Sie seltener zur Maus greifen müssen, um einen bestimmten Befehl auszuwählen. Wenn Sie sich angewöhnen, diese Tastenkombinationen zu verwenden, ist Ihre Arbeit auch im Visual-Basic-Editor effizienter.

STANDARDTASTENKOMBINATIONEN FÜR DEN VISUAL-BASIC-EDITOR

Mit den folgenden Tastenkombinationen können Sie schnell im Visual-Basic-Editor navigieren.

Tastenkombination	Aktion
Alt + F11	zwischen Excel-Fenster und Visual-Basic-Editor wechseln
⇧ + F10	Kontextmenü des aktiven Fensters anzeigen (entspricht einem Rechtsklick mit der Maus)
Strg + R	Projekt-Explorer öffnen
F4	Eigenschaftfenster öffnen
F2	Objekt-Browser öffnen
F1	VBA-Hilfe öffnen
F7	geöffnetes Modul-Fenster aktivieren

TASTENKOMBINATIONEN FÜR DAS BEARBEITEN VON CODE IM CODEFENSTER

Früher oder später werden Sie zur gleichen Zeit an mehreren Excel-Makros arbeiten. Mit der Maus von einem zum anderen und innerhalb einer Makroprozedur zu navigieren, kann etwas umständlich sein. Die folgenden Tastenkombinationen ermöglichen es, schnell zu einer Prozedur zu springen, sich in den Modulen zu bewegen und sogar die Stelle zu finden, an der eine bestimmte Variable deklariert wurde.

Tastenkombination	Aktion
Strg + ↓	nächste Prozedur auswählen
Strg + ↑	vorige Prozedur auswählen
Strg + Bild ↕	Bildschirminhalt um eine Seite nach unten bewegen
Strg + Bild ↕	Bildschirminhalt um eine Seite nach oben bewegen
↕ + F2	zur ausgewählten Funktion oder Variable gehen
Strg + ↕ + F2	zur letzten Position zurückgehen
Strg + Pos1	zum Anfang des Moduls gehen
Strg + Ende	zum Ende des Moduls gehen
Strg + →	Einfügemarke wortweise nach rechts bewegen
Strg + ←	Einfügemarke wortweise nach links bewegen
Ende	Einfügemarke an das Ende der Zeile bewegen
Pos1	Einfügemarke an den Anfang der Zeile bewegen
⇐→	aktuelle Zeile einrücken
↕ + ⇐→	Einrückung der aktuellen Zeile entfernen
Strg + J	Eigenschaften und Methoden des aktuellen Objekts anzeigen

TASTENKOMBINATIONEN FÜR DAS DEBUGGEN VON CODE IM VISUAL-BASIC-EDITOR

Das Debuggen Ihres Codes ist ein wichtiger Arbeitsschritt bei der Programmierung von Excel-Makros. Sie können auf die Debugging-Features zwar auch über das Menü des Visual-Basic-Editors zugreifen, jedoch kommen Sie mit den folgenden Tastenkombinationen schneller zum Ziel.

Tastenkombination	Aktion
F5	aktuelle Prozedur ausführen oder deren Ausführung fortsetzen, falls sie unterbrochen wurde
Strg + Untbr	Ausführung der aktuell laufenden Prozedur unterbrechen
F8	Debug-Modus aktivieren und Code zeilenweise ausführen
Strg + F8	Code bis zur aktuellen Cursorposition ausführen
↵ + F8	im Debug-Modus die aktuelle Zeile überspringen
F9	Haltepunkt für die aktuelle Zeile einbeziehungsweise ausschalten
Strg + ↵ + F9	alle Haltepunkte entfernen
Alt + D + L	aktuelles Visual-Basic-Projekt kompilieren



Michael Alexander

Excel Makros für **dummies**[®]

2. Auflage

Übersetzung aus dem Amerikanischen von
Rainer G. Haselier

WILEY-VCH
WILEY-VCH GmbH

Excel Makros für Dummies

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

2. Auflage 2021

© 2021 Wiley-VCH GmbH, Weinheim

Original English language edition Excel Macros for Dummies 2. edition © 2017 by Wiley Publishing, Inc. All rights reserved including the right of reproduction in whole or in part in any form. This translation published by arrangement with John Wiley and Sons, Inc.

Copyright der englischsprachigen Originalausgabe Excel Macros for Dummies 2. Auflage © 2017 by Wiley Publishing, Inc. Alle Rechte vorbehalten inklusive des Rechtes auf Reproduktion im Ganzen oder in Teilen und in jeglicher Form. Diese Übersetzung wird mit Genehmigung von John Wiley and Sons, Inc. publiziert.

Wiley, the Wiley logo, Für Dummies, the Dummies Man logo, and related trademarks and trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries. Used by permission.

Wiley, die Bezeichnung »Für Dummies«, das Dummies-Mann-Logo und darauf bezogene Gestaltungen sind Marken oder eingetragene Marken von John Wiley & Sons, Inc., USA, Deutschland und in anderen Ländern.

Das vorliegende Werk wurde sorgfältig erarbeitet. Dennoch übernehmen Autoren und Verlag für die

Richtigkeit von Angaben, Hinweisen und Ratschlägen
sowie eventuelle Druckfehler keine Haftung.

Coverfoto: © amazing studios / stock.adobe.com

Korrektur: Matthias Delbrück

Print ISBN: 978-3-527-71841-2

ePub ISBN: 978-3-527-83315-3

Inhaltsverzeichnis

Cover

Titelblatt

Impressum

Einleitung

[Über dieses Buch](#)

[Törichte Annahmen über den Leser](#)

[Icons in diesem Buch](#)

[Die Beispieldateien zu diesem Buch](#)

[Wie es weitergeht](#)

Teil I: Heiliges Makro, Batman!

Kapitel 1: Makrogrundlagen

[Warum sollten Sie Makros verwenden?](#)

[Grundlagen der Makroaufzeichnung](#)

[Absolute und relative Makroaufzeichnung im Vergleich](#)

[Weitere Konzepte für die Makroaufzeichnung](#)

[Makros im Einsatz - Beispiele](#)

Kapitel 2: Machen Sie sich mit dem Visual-Basic-Editor vertraut

[Im Visual-Basic-Editor arbeiten](#)

[Mit dem Projekt-Explorer arbeiten](#)

[Mit dem Codefenster arbeiten](#)

[Die VBA-Entwicklungsumgebung anpassen](#)

Kapitel 3: Die Anatomie von Makros

[Ein kurzer Überblick über das Excel-Objektmodell](#)

[Ein kurzer Blick auf Variablen](#)

[Ereignisprozeduren verstehen](#)

[Fehlerbehandlung kurz und bündig](#)

Teil II: Aufgaben in Arbeitsmappen mit wenig Arbeit erledigen

Kapitel 4: Makros für Arbeitsmappen

- Eine neue Arbeitsmappe aus dem Nichts erstellen
- Eine Arbeitsmappe speichern, wenn eine bestimmte Zelle geändert wird
- Eine Arbeitsmappe vor dem Schließen speichern
- Ein Tabellenblatt vor dem Schließen schützen
- Beim Öffnen einer Arbeitsmappe den Blattschutz aufheben
- Beim Öffnen einer Arbeitsmappe ein bestimmtes Tabellenblatt anzeigen
- Eine vom Benutzer ausgewählte Arbeitsmappe öffnen
- Feststellen, ob eine Arbeitsmappe bereits geöffnet ist
- Überprüfen, ob eine Arbeitsmappe in einem Ordner vorhanden ist
- Alle Arbeitsmappen auf einmal schließen
- Alle Arbeitsmappen in einem Ordner drucken
- Arbeitsmappe erst dann schließen, wenn eine bestimmte Zelle Inhalte besitzt
- Backup der aktuellen Arbeitsmappe mit aktuellem Datum erstellen

Kapitel 5: Makros für Tabellenblätter

- Ein neues Tabellenblatt einfügen und benennen
- Alle Tabellenblätter bis auf das aktive löschen
- Alle Tabellenblätter bis auf das aktive ausblenden
- Alle Tabellenblätter einer Arbeitsmappe einblenden
- Tabelleblätter verschieben
- Tabelleblätter nach Namen sortieren
- Tabelleblätter nach Farben gruppieren
- Ein Tabellenblatt in eine neue Arbeitsmappe kopieren
- Eine neue Arbeitsmappe für jedes Tabellenblatt erstellen
- Bestimmte Tabellenblätter drucken
- Alle Tabellenblätter schützen
- Den Blattschutz aller Tabellenblätter aufheben

[Ein Inhaltsverzeichnis Ihrer Arbeitsmappe erstellen](#)
[Mit einem Doppelklick in einem Tabellenblatt rein- und rauszoomen](#)
[Die aktive Zeile und Spalte hervorheben](#)

Teil III: Datenbearbeitung mit einem Klick

Kapitel 6: Makros für Zellen und Zellbereiche

[Einen Bereich auswählen und formatieren](#)
[Benannte Bereiche erstellen und auswählen](#)
[Alle Zellen eines Bereichs in einer Schleife durchlaufen](#)
[In einen Bereich leere Zeilen einfügen](#)
[Alle Zeilen und Spalten einblenden](#)
[Leere Zeilen löschen](#)
[Leere Spalten löschen](#)
[Den Scrollbereich einschränken](#)
[Alle Formeln in einer Arbeitsmappe auswählen und formatieren](#)
[Die erste leere Zeile oder Spalte finden und auswählen](#)

Kapitel 7: Makros für das Bearbeiten von Zelldaten

[Einen Zellbereich kopieren und einfügen](#)
[Alle Formeln eines Zellbereichs in Werte konvertieren](#)
[Den Befehl »Text in Spalten« auf alle Spalten anwenden](#)
[Nachstehende Minuszeichen konvertieren](#)
[Führende und nachstehende Leerzeichen aller Zellen in einem Bereich entfernen](#)
[US-Postleitzahlen auf die ersten fünf Stellen kürzen](#)
[Zellen mit führenden Nullen versehen](#)
[Leere Zellen durch einen Wert ersetzen](#)
[Vor oder nach dem Zellinhalt Text einfügen](#)
[Nicht druckbare Zeichen entfernen](#)
[Duplikate in einem Datenbereich hervorheben](#)
[Alle Zeilen bis auf Zeilen mit Duplikaten ausblenden](#)

Festlegen, in welchen Spalten AutoFilter eingesetzt werden darf

Gefilterte Zeilen in eine neue Arbeitsmappe kopieren

In der Statusleiste die Spalten anzeigen, nach denen gefiltert wird

Teil IV: Berichte, E-Mails und externe Datenquellen mit Makros steuern

Kapitel 8: Die Erstellung von Berichten automatisieren

Alle Pivot-Tabellen einer Arbeitsmappe aktualisieren

Eine Liste mit Informationen zu den Pivot-Tabellen einer Arbeitsmappe erstellen

Die Titel aller Datenfelder der Pivot-Tabelle anpassen

Für alle Datenelemente Summe verwenden

Alle Datenelemente als Zahlen formatieren

Alle Datenfelder alphabetisch sortieren

Datenelemente benutzerdefiniert sortieren

Einschränkungen für Pivot-Tabellen festlegen

Einschränkungen für PivotTable-Felder festlegen

Drilldown-Tabellenblätter automatisch entfernen

Eine Pivot-Tabelle für jedes Element des Berichtsfilters drucken

Für jedes Element des Berichtsfilters eine neue Arbeitsmappe erstellen

Die Größe aller Diagramme eines Tabellenblatts ändern

Diagramm platzieren

Verknüpfung zwischen Diagramm und Daten lösen

Alle Diagramme eines Tabellenblatts drucken

Kapitel 9: E-Mails mit Excel versenden

Die aktive Arbeitsmappe als E-Mail-Anhang senden

Einen bestimmten Zellbereich als E-Mail-Anhang senden

Ein bestimmtes Tabellenblatt als E-Mail-Anhang senden

Mail mit einem Link zu Ihrer Arbeitsmappe senden

[E-Mail an alle Adressen in einer Liste mit Kontakten senden](#)

[Alle E-Mail-Anhänge in einem Ordner speichern](#)

[Bestimmte E-Mail-Anhänge in einem Ordner speichern](#)

Kapitel 10: Externe Daten mit Makros bearbeiten

[Externe Datenverbindungen verwenden](#)

[Mit Makros dynamische Verbindungen erstellen](#)

[Alle Verbindungen in einer Arbeitsmappe in einer Schleife bearbeiten](#)

[ADO und VBA verwenden, um externe Daten abzurufen](#)

[ADO in einen Makro verwenden](#)

[Mit Textdateien arbeiten](#)

Teil V: Der Top-Ten-Teil

Kapitel 11: Zehn nützliche Tipps für den Visual-Basic-Editor

[Kommentarblöcke verwenden](#)

[Mehrere Codezeilen auf einen Rutsch kopieren](#)

[Zwischen Modulen und Prozeduren springen](#)

[Beamern Sie sich zu Ihren Funktionen](#)

[In der richtigen Prozedur bleiben](#)

[Den Code schrittweise ausführen](#)

[Zu einer bestimmten Zeile in Ihrem Code springen](#)

[Die Ausführung Ihres Codes an einer bestimmten Stelle unterbrechen](#)

[Den Anfang und das Ende von Variablenwerten anzeigen](#)

[Die automatische Syntaxüberprüfung ausschalten](#)

Kapitel 12: Zehn Orte, wo Sie Hilfe bei der Makro-Programmierung erhalten

[Lassen Sie Excel die Makros für Sie schreiben](#)

[Verwenden Sie die VBA-Hilfdateien](#)

[Stibitzen Sie Code im Internet](#)

[Userforen optimal nutzen](#)

[Besuchen Sie Experten-Blogs](#)
[Suchen Sie auf YouTube nach Schulungsvideos](#)
[Nehmen Sie an Online-Kursen teil](#)
[Vom Office Dev Center lernen](#)
[Analysieren Sie andere Excel-Dateien in Ihrem Unternehmen](#)
[Fragen Sie Ihre Excel-Experten vor Ort](#)

Kapitel 13: Zehn Methoden zur Beschleunigung von Makros

[Automatische Neuberechnung abschalten](#)
[Die Bildschirmaktualisierung deaktivieren](#)
[Aktualisierung der Statusleiste abschalten](#)
[Excel anweisen, Ereignisse zu ignorieren](#)
[Seitenwechsel ausblenden](#)
[Die Aktualisierung von Pivot-Tabellen unterdrücken](#)
[Kopieren und Einfügen vermeiden](#)
[Die With-Anweisung verwenden](#)
[Die Methode Select vermeiden](#)
[Zugriffe auf das Tabellenblatt optimieren](#)

Stichwortverzeichnis

End User License Agreement

Illustrationsverzeichnis

Kapitel 1

[Abbildung 1.1: Das Dialogfeld »Makro aufzeichnen«](#)
[Abbildung 1.2: Dieses Tabellenblatt enthält zwei nebeneinander steh...](#)
[Abbildung 1.3: Das Tabellenblatt mit der neuen Zeile »Anzahl«](#)
[Abbildung 1.4: Das Excel-Dialogfeld »Makro«](#)
[Abbildung 1.5: Aufzeichnung eines Makros mit relativen Verweisen](#)

[Abbildung 1.6: Fügen Sie hier Speicherorte ein, die als vertrauensw...](#)

[Abbildung 1.7: Sie finden die Formularsteuerelemente auf der Registerkarte »Entwi...](#)

[Abbildung 1.8: Der neu hinzugefügten Schaltfläche ein Makro zuweisen](#)

[Abbildung 1.9: Ein Makro in die Symbolleiste für den Schnellzugriff einfügen](#)

[Abbildung 1.10: Verwenden Sie Makros, um Schaltflächen zu erstellen, mit denen di...](#)

[Abbildung 1.11: In diesem Bericht können die Benutzer die Perspektive des Diagram...](#)

[Abbildung 1.12: Die Makros hinter diesen Schaltflächen ordnen die Datenfelder der...](#)

[Abbildung 1.13: Vorher aufgezeichnete Ansichten zur Verfügung zu stellen, führt n...](#)

[Abbildung 1.14: Sie können Ihren Anwendern die Entscheidung darüber überlassen, w...](#)

Kapitel 2

[Abbildung 2.1: Die wichtigsten Elemente der Entwicklungsumgebung Visual-Basic-Edi...](#)

[Abbildung 2.2: Der Projekt-Explorer zeigt zwei Projekte an. Beide Projekte sind e...](#)

[Abbildung 2.3: Code-Module sind im Knoten »Module« des Projekt-Explorers sichtbar...](#)

[Abbildung 2.4: Die Registerkarte »Editor« des Dialogfelds »Optionen«](#)

[Abbildung 2.5: So legen Sie das Aussehen von VBE auf der Registerkarte »Editorfor...](#)

[Abbildung 2.6: Die Registerkarte »Allgemein« des Dialogfelds »Optionen«](#)

[Abbildung 2.7: Die Registerkarte »Verankern« des Dialogfelds »Optionen«](#)

Kapitel 3

[Abbildung 3.1: Das integrierte Code-Modul eines Tabellenblatts öffnen](#)

[Abbildung 3.2: Das Standardereignis »SelectionChange«](#)

[Abbildung 3.3: Wählen Sie das passende Ereignis aus.](#)

[Abbildung 3.4: Das integrierte Modul für »eine Arbeitsmappe öffnen«](#)

[Abbildung 3.5: Das Standardereignis »Open« für Arbeitsmappen](#)

[Abbildung 3.6: Öffnen Sie die Drop-down-Liste »Ereignisse« und wählen Sie das pas...](#)

Kapitel 4

[Abbildung 4.1: Ändert sich eine beliebige Zelle im Bereich C5:C16, wird die Arbei...](#)

[Abbildung 4.2: Tippen Sie den Code in das Tabellenblattereignis Change ein oder k...](#)

[Abbildung 4.3: Wenn Sie versuchen, die Arbeitsmappe zu schließen, w...](#)

[Abbildung 4.4: Geben Sie den Code in die Prozedur für das Arbeitsmappenereignis B...](#)

[Abbildung 4.5: Geben Sie den Code in die Prozedur für das Arbeitsmappenereignis B...](#)

[Abbildung 4.6: Geben Sie den Code in die Prozedur für das Arbeitsmappenereignis O...](#)

[Abbildung 4.7: Sie möchten erreichen, dass beim Öffnen der Arbeitsmappe das Tabel...](#)

[Abbildung 4.8: Geben Sie den Code in die Prozedur für das Arbeitsmappenereignis O...](#)

[Abbildung 4.9: Das von einem Makro aktivierte Dialogfeld zum Öffnen einer Datei](#)

[Abbildung 4.10: Sie können diese nervige Meldung vermeiden, die an...](#)

[Abbildung 4.11: Eine deutliche und gut lesbare Meldung](#)

[Abbildung 4.12: Sie können das Schließen einer Arbeitsmappe verhindern, solange e...](#)

[Abbildung 4.13: Geben Sie den Code in die Prozedur für das Arbeitsmappenereignis](#)

Kapitel 5

[Abbildung 5.1: Entfernen Sie im Makro die Anweisung Application.DisplayAlerts = F...](#)

[Abbildung 5.2: Ohne Makro sind Sie auf das Excel-Dialogfeld »Einblenden« angewies...](#)

[Abbildung 5.3: Es ist häufig nützlich, wenn die Tabellenblätter alphabetisch sort...](#)

[Abbildung 5.4: Sie können im Blattregister ein Tabellenblatt mit der rechten Maus...](#)

[Abbildung 5.5: Der Befehl »Blatt schützen« ist deaktiviert, wenn Sie mehrere Tabe...](#)

[Abbildung 5.6: Der Befehl »Blattschutz aufheben« ist deaktiviert, wenn Sie mehrer...](#)

[Abbildung 5.7: Mit einem Inhaltsverzeichnis lässt sich sehr schnell auch in einer...](#)

[Abbildung 5.8: Tippen Sie den Code in das Tabellenblattereignis BeforeDoubleClick](#)

[Abbildung 5.9: Dank der hervorgehobenen Zeile und Spalte ist es einfacher, die an...](#)

[Abbildung 5.10: Geben Sie den Code in das Tabellenblattereignis BeforeDoubleClick...](#)

Kapitel 6

[Abbildung 6.1: Klicken Sie auf den Befehl »Namen definieren«, um den ausgewählten...](#)

[Abbildung 6.2: Hier legen Sie den Namen für den Bereich fest.](#)

[Abbildung 6.3: Das Dialogfeld »Namens-Manager« zeigt alle benannten Bereiche an.](#)

[Abbildung 6.4: Tippen Sie den Code in die Prozedur für das Arbeitsm...](#)

[Abbildung 6.5: Das Dialogfeld »Inhalte auswählen«](#)

[Abbildung 6.6: Aktivieren Sie im Dialogfeld »Inhalte auswählen« das Kontrollkästc...](#)

[Abbildung 6.7: Sie können ein Makro verwenden, um dynamisch die erste freie Zelle...](#)

Kapitel 7

[Abbildung 7.1: Importierte Zahlen sind manchmal als Text formatiert.](#)

[Abbildung 7.2: Wählen Sie den Befehl »Text in Spalten«.](#)

[Abbildung 7.3: Klicken Sie auf »Fertig stellen«, um falsch formatierte Zahlen zu ...](#)

[Abbildung 7.4: Dieses Makro sucht Duplikate in einem Bereich und heben sie hervor...](#)

[Abbildung 7.5: Nur Zeilen, die Duplikate enthalten, sind noch zu sehen.](#)

[Abbildung 7.6: Der Standard-AutoFilter fügt in jede Spalte des Datenbereichs eine...](#)

[Abbildung 7.7: Mit ein wenig VBA-Code erreichen Sie, dass nur die Spalten »Vertri...](#)

[Abbildung 7.8: Dieses Makro zeigt alle gefilterten Spalten in der Statusleiste an...](#)

[Abbildung 7.9: Tippen Sie den Code in das Tabellenblattereignis Calculate ein ode...](#)

Kapitel 8

[Abbildung 8.1: Verzeichnis aller PivotTable-Berichte](#)

[Abbildung 8.2: Excel zeigt eine Liste der Funktionen an, die Sie für die Zusammen...](#)

[Abbildung 8.3: Tippen Sie den Code in das Tabellenblattereignis BeforeDoubleClick](#)

[Abbildung 8.4: Geben Sie den Code in die Prozedur für das Arbeitsmappenereignis B...](#)

[Abbildung 8.5: Im Auswahlbereich werden alle Diagrammobjekte sowie deren Namen an...](#)

Kapitel 10

[Abbildung 10.1: Wählen Sie die Datenbankquelle aus, die die Daten enthält, die Si...](#)

[Abbildung 10.2: Wählen Sie das Access-Objekt aus, das Sie importieren wollen.](#)

[Abbildung 10.3: Legen Sie fest, wo und wie Sie die Access-Daten sehen wollen.](#)

[Abbildung 10.4: Daten, die aus Access importiert wurden.](#)

[Abbildung 10.5: Solange eine Verbindung mit Ihrer Datenbank verfügbar ist, können...](#)

[Abbildung 10.6: Im Excel 2019-Aufgabenbereich ABFRAGEN UND VERBINDUNGEN werden au...](#)

[Abbildung 10.7: Legen Sie auf der Registerkarte DEFINITION den Befehlstyp auf »SQ...](#)

[Abbildung 10.8: Legen Sie die Zelle fest, in der das Auswahlkriterium eingegeben ...](#)

[Abbildung 10.9: Merken Sie sich den Namen der Verbindung, in diesem Fall *Facility*...](#)

[Abbildung 10.10: Sie haben nun ein einfach zu verwendendes Verfahren, um die exte...](#)

[Abbildung 10.11: Wählen Sie die aktuelle Version der Microsoft ActiveX Data Objec...](#)

Kapitel 11

[Abbildung 11.1: Ein Hochkomma am Anfang einer Zeile wandelt diese Zeile in einen ...](#)

[Abbildung 11.2: Verwenden Sie die Symbolleiste BEARBEITEN, wenn Sie einen Block a...](#)

[Abbildung 11.3: Wenn Sie beim Ziehen die !\[\]\(8bbc1f1299a246c196d33c27b686a2d7_img.jpg\)-Taste gedrückt halten, erstellen Sie ...](#)

[Abbildung 11.4: Drücken Sie !\[\]\(0d12326490dcf2fb15b925ccac8d3f27_img.jpg\) + !\[\]\(28b2b9cbed27eda80300eb805b7f8a0d_img.jpg\), wenn der Cursor auf dem Namen einer Variablen od...](#)

[Abbildung 11.5: Verwenden Sie die Schaltfläche »Prozeduransicht«, um den Bildlauf...](#)

[Abbildung 11.6: Drücken Sie auf Ihrer Tastatur !\[\]\(70fe8e9535b25b843243cca42ea05eeb_img.jpg\), um das Makro schrittweise ausfü...](#)

[Abbildung 11.7: Ziehen Sie den Pfeil, während Ihr Code schrittweise ausgeführt wi...](#)

[Abbildung 11.8: Ein Haltepunkt wird durch einen roten Punkt in der Markierungslei...](#)

[Abbildung 11.9: Anzeige der ersten und der letzten Zeichen einer String-Variablen...](#)

[Abbildung 11.10: Eine unvollendete Codezeile führt zu diesem irritierenden Warnhi...](#)

[Abbildung 11.11: So unterdrücken Sie während der Programmierung die Syntax-Warnhi...](#)

Einleitung

Im weitesten Sinne ist ein Excel-Makro eine Reihe von Anweisungen, die irgendeine Funktion von Excel automatisieren. Sie können Excel somit effizienter einsetzen und machen weniger Fehler. Sie können beispielsweise ein Makro programmieren, um einen monatlichen Umsatzbericht zu drucken. Nachdem Sie das Makro entwickelt haben, können Sie es ausführen und die ganze ansonsten zeitaufwändige Aktion in einem Rutsch durchführen.

Makros werden in VBA geschrieben. VBA ist die Abkürzung für *Visual Basic for Applications* (Visual Basic für Anwendungen). Visual Basic for Applications ist eine von Microsoft entwickelte Programmiersprache und ein Werkzeug, um Programme zu entwickeln, mit denen Sie Excel steuern können.

Die Begrifflichkeit der Excel-Programmierung kann ein wenig verwirrend sein. So ist beispielsweise VBA sowohl eine Programmiersprache als auch eine Makrosprache. Wie nennen Sie etwas, das in VBA geschrieben wurde und in Excel ausgeführt wird? Ist es ein Makro oder ist es ein Programm? Das Excel-Hilfesystem verwendet für VBA-Prozeduren den Begriff Makro; daher wird in diesem Buch die gleiche Terminologie verwendet.

Im Verlauf dieses Buchs werden Sie häufig auf den Begriff *automatisieren* stoßen. Automatisierung bedeutet, dass eine Reihe von Schritten automatisch durchgeführt wird. Wenn Sie beispielsweise ein Makro schreiben, das die Hintergrundfarbe von Zellen ändert, das Tabellenblatt druckt und die Einfärbung wieder entfernt, dann haben Sie diese drei Schritte automatisiert.

Sie wissen bestimmt, dass Excel-Anwender das Programm für Tausende von unterschiedlichen Aufgaben einsetzen. Hier nur ein paar Beispiele:

- ✓ Verwaltung von Listen oder Tabellen, beispielsweise für Kundendaten oder Transaktionen
- ✓ Budgetplanung und Umsatzprognosen
- ✓ Analyse wissenschaftlicher Daten
- ✓ Erstellung von Rechnungen und anderen Formularen
- ✓ Daten in Form von Diagrammen darstellen

Diese Liste könnte noch stundenlang fortgesetzt werden. Der entscheidende Punkt ist, dass Excel für unglaublich viele verschiedene Aufgaben verwendet wird, und alle, die dieses Buch lesen, haben im Hinblick auf Excel unterschiedliche Anforderungen und Erwartungen. Etwas, was wohl die meisten Leser bewegt, ist, dass jeder irgendeine Funktion von Excel automatisieren will. Und genau darum geht es bei Makros und in diesem Buch.

Über dieses Buch

Diesem Buch über Excel-Makros liegt die Erkenntnis zugrunde, dass die Programmierung in VBA Zeit und Übung erfordert – Zeit, die Sie im Moment vielleicht gar nicht haben. Tatsächlich können sich viele Datenanalytiker nicht den Luxus leisten, sich ein paar Wochen Zeit zu nehmen, um zu VBA-Experten zu werden. Das Buch trägt diesem Sachverhalt Rechnung. Sie erhalten daher nicht einen allgemeinen Überblick zu VBA, sondern finden im Buch praxisnahe Beispielmakros zu den Einsatzgebieten, die am häufigsten vorkommen.

Jeder Abschnitt des Buchs beschreibt eine Aufgabe, die häufig vorkommt, und liefert dann ein Excel-Makro, mit dem diese Aufgabe erledigt werden kann. Außerdem erhalten Sie detaillierte Informationen dazu, wie das Makro funktioniert und wie Sie es einsetzen können.

Jeder Abschnitt enthält die folgenden Informationen:

- ✓ das Problem,
- ✓ die Makrolösung,
- ✓ wie das Makro funktioniert.

Nachdem Sie einen Abschnitt gelesen haben, sind Sie in der Lage,

- ✓ das beschriebene Makro sofort zu implementieren,
- ✓ die Funktionsweise des Makros zu verstehen,
- ✓ das Makro in anderen Arbeitsmappen zu verwenden,
- ✓ die beschriebene Technik in anderen Makros einzusetzen.

Die Makros in diesem Buch wurden mit der Zielsetzung erstellt, Ihnen einen schnellen und einfachen Einstieg in VBA zu ermöglichen. Jedes Makro behandelt eine regelmäßig vorkommende Aufgabe, bei der sich eine Automatisierung lohnt. Die Idee ist, dass Sie die Verwendung von Makros erlernen, indem Sie Makros benutzen. Dieses Buch wurde so geschrieben, dass Sie die einzelnen Makros implementieren können und sozusagen ganz nebenbei lernen, was das Makro wie macht.

Törichte Annahmen über die Leser

Ich setze nicht viel voraus, aber die folgenden Vermutungen habe ich schon angestellt, als ich mir Sie, meine Leser, vorgestellt habe:

- ✓ Sie haben Microsoft Excel 2007 oder eine neuere Version installiert.
- ✓ Sie haben grundlegende Kenntnisse über die Datenanalyse mit Excel, zum Beispiel über das Arbeiten mit Tabellen, die Zusammenfassung von Daten, das Erstellen von Formeln, das Verweisen auf Zellen und das Filtern und Sortieren.
- ✓ Sie können auf das Internet zugreifen, um die Beispieldateien herunterzuladen, die Sie auf <https://www.wiley-vch.de/de/dummies/downloads> finden.

Icons in diesem Buch

Was wäre ein »... für Dummies«-Buch ohne die vertrauten Icons, die Sie auf wichtige Informationen hinweisen und Sie auf Ihrem Weg unterstützen. In diesem Abschnitt stelle ich kurz die in diesem Buch verwendeten Icons vor.



Dieses Tipp-Icon weist Sie auf hilfreiche Empfehlungen und andere nützliche Informationen hin.



Beim Symbol »Nicht vergessen« finden Sie allgemeine, interessante und nützliche Hinweise, die Sie sich für den späteren Einsatz merken sollten.



Das Icon »Warnung« macht Sie auf drohende Gefahren aufmerksam. Sie finden hier Informationen, auf die Sie achten sollten, und es rät Ihnen, hier besonders behutsam vorzugehen.

Die Beispieldateien zu diesem Buch

Im Web finden Sie Beispieldateien zu diesem Buch.

Für jedes Makro in diesem Buch gibt es eine Beispieldatei, die zeigt, wie das Makro funktioniert und in der Sie sich den Makro-Code ansehen können. Sie können die Beispieldateien auch verwenden, um den Code in Ihre eigenen Arbeitsmappen einzufügen: Sie ersparen sich so Tipparbeit. Laden Sie die Beispieldateien von folgender Webseite herunter:

<https://www.wiley-vch.de/de/dummies/downloads>

Suchen Sie auf dieser Seite nach *Excel Makros für Dummies* und klicken Sie den gefundenen Eintrag an, um die ZIP-Datei mit allen Beispielen herunterzuladen.

Sie finden bei jedem Beispielmakro ausführliche Informationen dazu, wo Sie den kopierten Code einfügen müssen. In den meisten Fällen öffnen Sie die Beispieldatei mit dem betreffenden Makro, gehen zum Visual-Basic-Editor (indem Sie **Alt**+**F11** drücken) und kopieren den Code. Gehen Sie dann zu Ihrer eigenen Arbeitsmappe, öffnen Sie den Visual-Basic-Editor und

fügen Sie den Code an der gewünschten Stelle ein.



Beachten Sie, dass Sie bei einigen Makros Änderungen vornehmen müssen, damit sie zu Ihrer Umgebung passen. So gibt es beispielsweise in [Kapitel 4](#) ein Makro, das alle Arbeitsmappen im Verzeichnis `C:\Temp\` druckt. Bevor Sie dieses Makro verwenden, müssen Sie den Ordner angeben, in dem sich auf Ihrem Computer die zu druckenden Arbeitsmappen befinden.



Falls ein bestimmtes Makro bei Ihnen nicht funktioniert, müssen Sie in den meisten Fällen nicht viel mehr als eine bestimmte Komponente ändern. Achten Sie insbesondere auf Adressen von Zellbereichen, Namen von Verzeichnissen und andere »hart-kodierte« Namen.

Wie es weitergeht

Falls das Thema Excel-Makros für Sie vollkommenes Neuland ist, beginnen Sie am besten mit den [Kapiteln 1](#) bis [3](#). Sie finden dort das Grundlagenwissen, auf dem die Makros in diesem Buch basieren. Mit diesen kurzen und prägnanten Informationen werden Sie verstehen, wie Makros und VBA funktionieren. Außerdem wird dort das Fundament geschaffen, das Sie brauchen, um die Makros dieses Buchs zu implementieren.

Falls Sie bereits Erfahrungen mit Excel-Makros besitzen und direkt in die Makrobeispiele eintauchen wollen, schauen Sie sich die [Kapitel 4](#) bis [9](#) an und suchen Sie sich die Makros heraus, die für Sie besonders interessant sind. Jedes Makrobeispiel steht für sich allein. Jeder Abschnitt enthält also alles, was Sie

brauchen, um den jeweiligen Code zu verstehen und um ihn in Ihren eigenen Arbeitsmappen zu verwenden.

In [Teil II](#) finden Sie Makros, die regelmäßig wiederkehrende Arbeitsmappen- und Tabellenblattaufgaben automatisieren und die Ihnen helfen, Zeit zu sparen und die Aufgaben effizienter zu erledigen.

Durchforsten Sie [Teil III](#), wenn Sie Makros suchen, mit denen Sie in Zellbereichen navigieren, Zellen formatieren und die Daten Ihrer Arbeitsmappen bearbeiten können.

Wenn Sie auf der Suche nach Makros für die Automatisierung von redundanten Schritten beim Einsatz von PivotTable-Berichten und Diagrammen sind wenn Sie Makros benötigen, mit denen Sie E-Mails mit Excel-Dateien als Anhängen versenden können, oder wenn Sie Verbindungen mit externen Datenquellen herstellen wollen, dann blättern Sie durch [Teil IV](#).

Vergessen Sie nicht, sich auch die Kapitel in [Teil V](#) anzusehen, dem immer wieder beliebten »... für Dummies«-Top-Ten-Teil. Sie finden dort nützliche und hilfreiche Tipps und Hinweise, mit denen Sie noch mehr aus Ihren Makro-Fähigkeiten herausholen können.

Hier noch ein paar abschließende Bemerkungen, die die Verwendung der Beispielmakros vereinfachen:

- ✓ **Alle Excel-Dateien, die Makros enthalten, müssen die Dateinamenserweiterung .xlsm besitzen.** In [Kapitel 1](#) finden Sie einen eigenen Abschnitt mit wichtigen Informationen zu den Dateierweiterungen für Arbeitsmappen mit Makros.
- ✓ **Excel führt ein Makro erst dann aus, wenn es aktiviert wird.** Wenn Sie die Makros implementieren,

müssen Sie und Ihre Kunden die Sicherheitsvorkehrungen von Excel beachten. Im Abschnitt »Makrosicherheit ab Excel 2010« in [Kapitel 1](#) finden Sie weiterführende Informationen dazu.

- ✓ **Sie können Makro-Aktionen nicht rückgängig machen.** Während Sie in Excel arbeiten, können Sie oft die Schritte, die Sie gerade durchgeführt haben, wieder rückgängig machen, da Excel die letzten 100 Aktionen in einem Protokoll festhält. Wenn Sie ein Makro ausführen, werden diese Aktionen nicht protokolliert und sie können daher auch nicht rückgängig gemacht werden.
- ✓ **Sie müssen die Makros so anpassen, dass sie zu Ihren Arbeitsmappen passen.** Viele der Makrobeispiele verwenden Zelladressen oder Namen von Tabellenblättern, die möglicherweise so in Ihren Arbeitsmappen nicht vorkommen. Passen Sie daher alle verwendeten Namen von Tabellenblättern (wie *Tabelle1*) und die Zellbereiche wie in `Range("A1")` so an, dass sie zu den Zellbereichen und den Tabellenblättern passen, die Sie in Ihren Arbeitsmappen verwenden.

Teil I

Heiliges Makro, Batman!



IN DIESEM TEIL ...

- ✓ Legen Sie das Fundament für Ihr Makro-Können und lernen Sie die grundlegenden Konzepte der Makroaufzeichnung kennen
- ✓ Erlangen Sie solides Grundlagenwissen zur Verwendung und Verteilung von Makros in Excel
- ✓ Erforschen Sie die Excel-Programmierung und tauchen Sie in die Welt des Visual-Basic-Editors ein
- ✓ Verwenden Sie das Excel-Objektmodell, um Ihre eigenen Makros komplett selbst zu schreiben
- ✓ Verstehen Sie, welche Rollen Variablen, Ereignisse und die Fehlerbehandlung bei der Makroprogrammierung spielen

Kapitel 1

Makrogrundlagen

IN DIESEM KAPITEL

Warum Sie Makros verwenden sollten

Makros aufzeichnen

Makrosicherheit verstehen

Beispiele für Makros in der Praxis

Ein Makro ist im Grunde eine Reihe von Anweisungen oder Code, mit dem Sie Excel anweisen, bestimmte Aktionen auszuführen. Sie können Excel-Makros schreiben/programmieren oder aufzeichnen. Das Schlüsselwort für dieses Kapitel lautet *Aufzeichnung*.

Das Aufzeichnen eines Makros ähnelt dem Speichern einer Telefonnummer in Ihrem Handy. Auf Ihrem Handy wählen Sie die Nummer zuerst von Hand, und anschließend speichern Sie sie. Danach können Sie die gespeicherte Nummer mit einem einzigen Tastendruck erneut wählen. In Excel starten Sie die Makroaufzeichnung und führen dann die gewünschten Aktionen aus. Während der Aufzeichnung übersetzt Excel im Hintergrund Ihre Tastenanschläge und Mausklicks in Code, und zwar in einer Programmiersprache mit dem Namen *Visual Basic for Applications* (VBA, Visual Basic für Anwendungen). Nachdem ein Makro aufgezeichnet ist, können Sie es so oft Sie wollen starten und so die aufgezeichneten Aktionen immer wieder neu ausführen lassen.

In diesem Kapitel erkunden Sie Makros und Sie werden dabei sehen, wie Sie mit Makros sich wiederholende Aufgaben automatisieren und sich so Ihr Leben (zumindest Ihr Leben mit Excel) einfacher machen können.

Warum sollten Sie Makros verwenden?

Der erste Schritt bei der Verwendung von Makros ist die Erkenntnis, dass Sie ein Problem haben. Genau genommen haben Sie mehrere Probleme:

- ✓ **Problem 1: Sich wiederholende Aufgaben** An jedem Monatsanfang müssen Sie wieder und wieder diesen einen Bericht erstellen. Sie müssen Daten importieren. Sie müssen diese Pivot-Tabellen aktualisieren. Sie müssen diese Spalten löschen und so weiter. Wäre es nicht fantastisch, wenn Sie einfach ein Makro starten könnten, das die redundanten Schritte beim Erstellen Ihres Dashboards automatisch durchführt?
- ✓ **Problem 2: Sie machen Fehler** Wenn man mit Excel in den Nahkampf geht, macht man zwangsläufig Fehler. Wenn Sie öfters Formeln verwenden, Tabellen sortieren oder Daten von Hand verschieben, besteht immer die Gefahr einer Katastrophe. Kommen dann noch enge Abgabetermine und permanent neue Änderungswünsche hinzu, steigt die Wahrscheinlichkeit von Fehlern noch weiter. Warum zeichnen Sie nicht einfach ein Makro auf, kontrollieren noch schnell, dass es auch korrekt funktioniert, und vergessen dann einfach die einzelnen Schritte der Aufgabe? Das Makro führt bei jedem Start alle