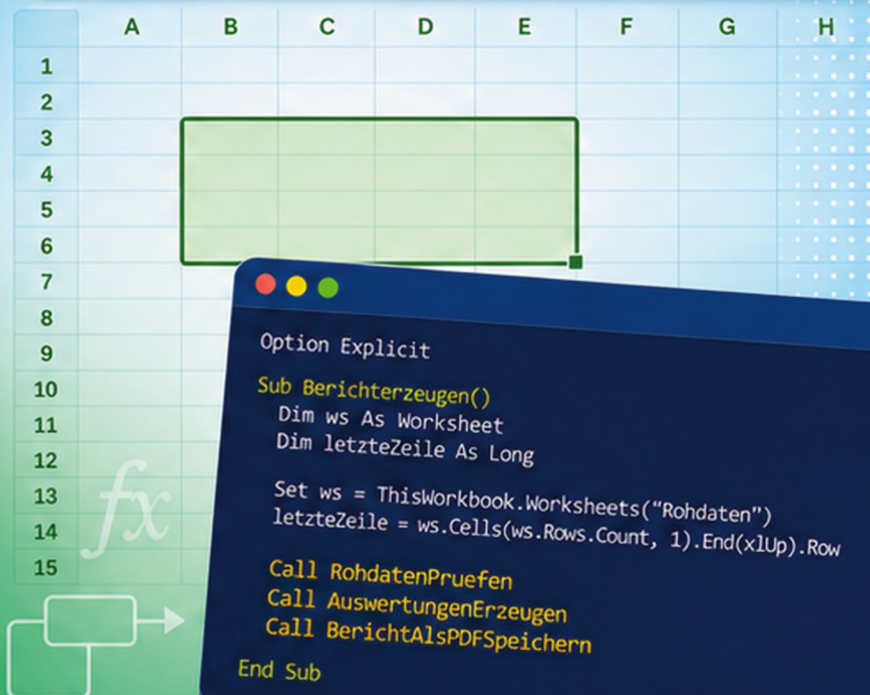


PRAXISBUCH

Excel VBA für Anfänger



Programmieren lernen mit Makros,
Zellen, Tabellen, Schleifen und eigenen
Automatisierungen



- ✓ Makros verstehen
- ✓ Excel automatisieren
- ✓ Berichte und PDF-Ausgabe erstellen

Excel VBA für Anfänger

**Programmieren lernen mit Makros, Zellen,
Tabellen, Schleifen und eigenen
Automatisierungen**

Hartmut Kruger

Deutsche Ausgabe

Impressum

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über dnb.dnb.de abrufbar.

Die automatisierte Analyse des Werkes, um daraus Informationen insbesondere über Muster, Trends und Korrelationen gemäß §44b UrhG („Text und Data Mining“) zu gewinnen, ist untersagt.

© 2026 Hartmut Kruger

Herstellung und Verlag:

BoD · [Books on Demand GmbH](#), Überseering 33, 22297 Hamburg

ISBN: 978-3-6963-1193-3

.

Wer wiederkehrende Arbeit erkennt, hat den ersten Schritt zur Automatisierung bereits getan.

VBA beginnt dort, wo aus manuellen Handgriffen ein klarer Ablauf wird.

Inhaltsverzeichnis

Impressum	2
Inhaltsverzeichnis	4
Vorwort.....	6
Hinweise zum Arbeiten mit diesem Buch	7
Begleitdateien zum Buch	8
Aufbau des Buches.....	9
Zielgruppe und Voraussetzungen.....	10
Schreibweisen und Code-Konventionen	11
Warum VBA in Excel nützlich ist	13
Entwicklertools und VBA-Editor einrichten	18
Das erste Makro aufzeichnen	24
Der Aufbau eines VBA-Programms	30
Variablen und Datentypen	38
Texte, Zahlen und Datumswerte	44
Entscheidungen mit If und Select Case.....	48
Schleifen	54
Fehler finden und verstehen	59
Das Excel-Objektmodell	66
Zellen und Bereiche bearbeiten.....	71
Tabellenblätter verwalten.....	75
Arbeitsmappen und Dateien	79
Dynamische Bereiche und letzte Zeile	83
Sortieren, Filtern und Suchen	87
Daten prüfen und bereinigen	92

Berichte automatisch erzeugen	98
PDF-Ausgabe und Druck.....	103
Mehrere Dateien verarbeiten.....	107
Meldungen und Eingaben.....	113
Schaltflächen und Startseiten.....	118
Erste UserForms.....	125
Makrosicherheit und Dateiformate.....	133
Besserer und übersichtlicher Code	138
Typische Anfängerfehler.....	143
Projekt: Kunden- und Umsatzbericht automatisieren	148
Projektcode erklären und erweitern.....	159
Anhang A – VBA-Kurzreferenz.....	165
Anhang B – Übungsdateien und Beispieldaten.....	169
Anhang C – Checklisten für VBA-Projekte	172
Anhang D – Glossar wichtiger VBA-Begriffe	174
Schlusswort – Vom ersten Makro zur eigenen Automatisierung.....	176
Über den Autor	177

Vorwort

Excel ist für viele Anwenderinnen und Anwender ein tägliches Arbeitswerkzeug. Listen werden geprüft, Daten bereinigt, Berichte erstellt und Ergebnisse kontrolliert. Viele dieser Tätigkeiten wiederholen sich regelmäßig.

Dieses Buch führt Schritt für Schritt in Excel VBA ein. Es richtet sich an Menschen, die Excel bereits praktisch nutzen und nun verstehen möchten, wie wiederkehrende Arbeitsschritte mit Makros und VBA automatisiert werden können.

Die Beispiele bleiben bewusst nah an typischen Excel-Aufgaben: Zellen bearbeiten, Tabellenblätter steuern, Daten prüfen, Auswertungen erzeugen, Berichte vorbereiten und Ergebnisse als PDF speichern.

Hinweise zum Arbeiten mit diesem Buch

Arbeiten Sie beim Ausprobieren der Makros immer mit Kopien. Makros können Inhalte verändern, überschreiben oder Dateien speichern.

Makrodateien werden als `.xlsm` gespeichert. Eine normale `.xlsx`-Datei speichert keinen VBA-Code.

Viele Beispiele verwenden bewusst einfache Daten und klare Blattnamen. Dadurch bleibt sichtbar, welcher Code welchen Arbeitsschritt ausführt.

Begleitdateien zum Buch

Zu diesem Buch gehören Begleitdateien, mit denen Sie die Beispiele praktisch nachvollziehen können. Die Dateien enthalten makrofähige Excel-Arbeitsmappen, Beispieldaten, Importdateien und ergänzende Hinweise.

Die Begleitdateien finden Sie unter folgender Adresse:

<https://www.hartmut-kruger.de/de/vba/>

Sie können die Adresse entweder im Browser eingeben oder den folgenden QR-Code mit einem Smartphone oder Tablet scannen.

QR-Code: Downloadseite für die Begleitdateien zum Buch



Wichtig: Die Beispieldateien enthalten Makros. Öffnen und aktivieren Sie Makros nur bewusst und arbeiten Sie beim Ausprobieren möglichst mit Kopien der Dateien.

Aufbau des Buches

Das Buch beginnt mit den Grundlagen: Entwicklertools, VBA-Editor, Makrorekorder und Aufbau einer Sub-Prozedur.

Danach folgen Variablen, Datentypen, Bedingungen, Schleifen und Fehlersuche. Anschließend wird gezeigt, wie Excel-Objekte gezielt angesprochen werden.

Der Praxisteil führt zu Datenprüfung, Berichten, PDF-Ausgabe, Dateiimport, Schaltflächen, UserForms und einem Abschlussprojekt.

Zielgruppe und Voraussetzungen

Das Buch richtet sich an Excel-Anwender, Sachbearbeiter, Controller, Analysten, Projektmitarbeiter, Selbständige und alle, die wiederkehrende Excel-Arbeitsschritte automatisieren möchten.

Programmierkenntnisse werden nicht vorausgesetzt. Sinnvoll sind grundlegende Excel-Kenntnisse: Tabellenblätter, Zellen, Formeln, einfache Listen und Speichern von Arbeitsmappen.

Die Beispiele orientieren sich an modernen Excel-Versionen unter Windows. Einzelne Dialoge können je nach Version abweichen.

Schreibweisen und Code-Konventionen

Makro- und Variablennamen werden ohne Leerzeichen und ohne Umlaute geschrieben, zum Beispiel `letzteZeile`, `wsDaten` oder `BerichtErzeugen`.

Sichtbare Texte in Excel dürfen Umlaute enthalten. Der Code selbst bleibt möglichst robust und leicht lesbar.

Option `Explicit` wird als hilfreicher Standard empfohlen, weil Variablen dadurch ausdrücklich deklariert werden müssen.

Teil 1

Einstieg in Excel VBA

Dieser Teil umfasst Kapitel 1 bis 4 und führt den Lernweg schrittweise fort.

Kapitel 1

Warum VBA in Excel nützlich ist

In diesem Kapitel lernen Sie:

- wann VBA in Excel sinnvoll eingesetzt wird
- wie sich ein Makro von VBA-Code unterscheidet
- warum Automatisierung immer mit einem klaren Ablauf beginnt
- welchen roten Faden dieses Buch verwendet

Excel ist in vielen Büros das Werkzeug, mit dem Daten gesammelt, geprüft, sortiert, gefiltert und ausgewertet werden. Oft entstehen dabei wiederkehrende Arbeitsschritte: Eine Liste wird geöffnet, bestimmte Spalten werden kontrolliert, ungültige Werte werden markiert, Summen werden berechnet, ein Bericht wird formatiert und am Ende wird eine Datei gespeichert oder als PDF ausgegeben.

Solche Aufgaben lassen sich manuell erledigen. Das funktioniert, solange die Liste klein ist, der Ablauf selten vorkommt und Fehler leicht zu erkennen sind. Sobald dieselbe Arbeit aber regelmäßig wiederholt wird, entsteht ein typischer Automatisierungsfall. Genau hier beginnt der praktische Nutzen von VBA.

VBA steht für Visual Basic for Applications. Es ist die Programmiersprache, mit der Excel gezielt gesteuert werden kann. Mit VBA können Sie Zellen beschreiben, Tabellenblätter anlegen, Arbeitsmappen öffnen, Daten prüfen, Berichte erzeugen und Dialoge anzeigen. Das Ziel ist nicht, Excel zu ersetzen. Das Ziel ist, Excel dort zu unterstützen, wo wiederkehrende Abläufe zuverlässig und nachvollziehbar ausgeführt werden sollen.

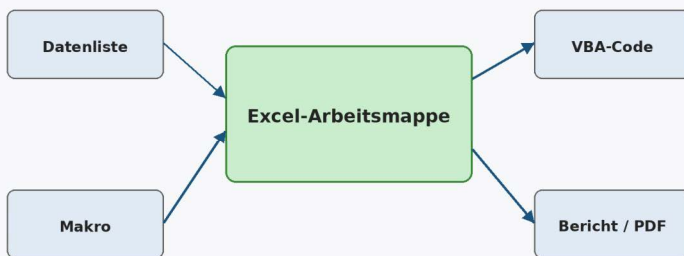
Vom wiederkehrenden Arbeitsschritt zum Makro

Ein Makro ist zunächst nichts anderes als ein gespeicherter Ablauf. Dieser Ablauf kann sehr einfach sein: Eine Überschrift wird formatiert, eine Spalte wird angepasst oder eine Meldung wird angezeigt. Ein Makro kann aber auch mehrere Schritte zusammenfassen: Daten

prüfen, Fehler markieren, eine Auswertung erzeugen und anschließend einen Bericht speichern.

Der wichtigste Gedanke lautet: Bevor ein Makro geschrieben wird, muss der Ablauf verstanden werden. VBA automatisiert keine unklare Arbeit. Es macht nur das schneller und zuverlässiger, was vorher als Schrittfolge beschrieben werden kann.

Abbildung 1-01: Excel als Arbeitsumgebung für VBA



Arbeitsgrafik V0.2 – finale Screenshots nach Erstellung der Begleitdateien ersetzen

Abb. 1-01: Excel als Arbeitsumgebung für VBA

Die Abbildung zeigt die Grundidee dieses Buches. Excel bleibt die Arbeitsumgebung. Datenlisten, Makros, VBA-Code und Berichte gehören zusammen. Der Code arbeitet nicht losgelöst von Excel, sondern steuert Arbeitsmappen, Tabellenblätter und Zellbereiche.

Der Unterschied zwischen Makrorekorder und VBA

Excel besitzt einen Makrorekorder. Er zeichnet manuelle Arbeitsschritte auf und erzeugt daraus VBA-Code. Das ist ein sehr guter Einstieg, weil sofort sichtbar wird, dass eine Handlung in Excel in Code übersetzt werden kann.

Der Makrorekorder ist aber kein Ersatz für Verständnis. Aufgezeichneter Code enthält häufig Befehle wie Select, Selection oder ActiveCell. Diese

Begriffe sind für den Anfang hilfreich, weil sie zeigen, was gerade ausgewählt wurde. Für stabile Makros sind direkte Objektbezüge aber meist besser. Ein gutes Makro sollte möglichst eindeutig wissen, in welcher Arbeitsmappe, auf welchem Tabellenblatt und in welchem Zellbereich es arbeitet.

Wann VBA sinnvoll ist

VBA ist besonders sinnvoll, wenn Aufgaben wiederkehrend, regelbasiert und gut beschreibbar sind. Typische Beispiele sind Datenprüfungen, Formatierungen, Berichtserstellung, PDF-Ausgabe oder das Zusammenführen mehrerer Dateien.

VBA ist weniger geeignet, wenn eine Aufgabe nur einmal vorkommt oder wenn die Regeln noch unklar sind. In solchen Fällen ist es oft schneller, zuerst den manuellen Ablauf zu klären. Danach kann entschieden werden, ob sich eine Automatisierung lohnt.

Typische VBA-Aufgaben in diesem Buch

Dieses Buch verwendet einen durchgehenden Praxisfaden. Am Anfang stehen kleine Makros. Danach werden Variablen, Bedingungen und Schleifen eingeführt. Anschließend wird gezeigt, wie Excel-Objekte angesprochen werden: Arbeitsmappen, Tabellenblätter, Zellen und Zellbereiche.

Später werden diese Grundlagen in praktische Aufgaben übertragen: Rohdaten werden geprüft, Fehler werden in einer Fehlerliste gesammelt, Auswertungen werden erstellt, ein Bericht wird aufgebaut und als PDF gespeichert. Am Ende entsteht ein kleines Abschlussprojekt für einen Kunden- und Umsatzbericht.



Der wiederkehrende Ablauf wird einmal verstanden und danach als Makro ausgeführt.

Arbeitsgrafik V0.1 – finale Screenshots nach Erstellung der Begleitdateien ersetzen

Abb. 1-02: Vom manuellen Ablauf zum Makro

Was VBA nicht automatisch löst

VBA kann Arbeitsschritte automatisieren, aber es kann keine fachlichen Entscheidungen ersetzen. Wenn eine Datenliste unklare Spalten, widersprüchliche Werte oder wechselnde Strukturen enthält, muss zuerst geklärt werden, was richtig ist. Erst danach kann ein Makro zuverlässig arbeiten.

Außerdem sollten Makros immer mit Kopien getestet werden. Ein Makro kann Zellen überschreiben, Blätter löschen oder Dateien speichern. Das ist gewollt, wenn der Code korrekt ist. Beim Lernen sollte deshalb nie direkt mit produktiven Originaldaten gearbeitet werden.

Praxisbeispiel

Stellen Sie sich eine monatliche Umsatzliste vor. Jeden Monat müssen fehlende Kundennummern gesucht, ungültige Regionen erkannt und offene oder überfällige Aufträge markiert werden. Danach wird ein Bericht erstellt. Manuell ist dieser Ablauf möglich, aber fehleranfällig. Mit VBA kann die Prüfung immer nach denselben Regeln erfolgen.