



Inhaltsverzeichnis

Danksagungen	xxii
Einleitung	xxiii
Ist JavaScript eine Bedrohung für VBA?	xxiii
Themen und Inhalte dieses Buchs	xxiv
Die Lernkurve verkürzen	xxiv
Die Stärken von Excel VBA	xxiv
Technische Dinge, die für die Anwendungserstellung benötigt werden ..	xxv
Ist dies ein Buch, um Excel zu erlernen?	xxv
Excel-Versionen	xxviii
Unterschiede für Mac-Anwender	xxviii
Besondere Elemente und typografische Konventionen	xxviii
Die Begleitdateien	xxix
Support und Feedback	xxix
Wir wollen von Ihnen hören	xxix
Errata, Updates und Support	xxix

Kapitel 1

Die Leistungsfähigkeit von Excel mit VBA nutzen	1
Hindernisse beim Einstieg	1
Der Makrorekorder funktioniert nicht	1
Niemand im Excel-Team ist auf den Makrorekorder fokussiert	2
Visual Basic ist nicht wie BASIC	2
Gute Nachrichten: Die Lernkurve ist nicht sehr steil	3
Großartige Nachrichten: Für Excel mit VBA lohnt sich der Aufwand	3
Kennen Sie Ihre Werkzeuge: die Registerkarte Entwicklertools	4
Verstehen, in welchen Dateitypen Makros erlaubt sind	5
Makrosicherheit	6
Hinzufügen eines vertrauenswürdigen Speicherorts	7
Verwenden von Makroeinstellungen, um Makros in Arbeitsmappen zu aktivieren, die außerhalb vertrauenswürdiger Speicherorte gespeichert sind	8
Verwenden der Option »Alle Makros mit Benachrichtigung deaktivieren«	9

Aufzeichnen, Speichern und Ausführen eines Makros im Überblick	9
Ausfüllen des Dialogfelds »Makro aufzeichnen«	10
Ausführen eines Makros	11
Erstellen einer Makroschaltfläche im Menüband	12
Erstellen einer Makroschaltfläche in der Symbolleiste für den Schnellzugriff	13
Zuweisen eines Makros an ein Formularsteuerelement, ein Textfeld oder eine Form	14
Den Visual Basic-Editor verstehen	16
VB-Editor-Einstellungen	16
Der Projekt-Explorer	17
Das Eigenschaftenfenster	17
Die Schwachpunkte des Makrorekorders verstehen	18
Aufzeichnen des Makros	20
Untersuchen des Codes im Programmierfenster	20
Ausführung des Makros an einem anderen Tag führt zu ungewünschten Ergebnissen	23
Mögliche Lösung: Beim Aufzeichnen relative Verweise verwenden	24
Verwenden Sie bei der Makroaufzeichnung niemals AutoSumme oder die Schnellanalyse	28
Vier Tipps für die Verwendung des Makrorekorders	30
Nächste Schritte	31

Kapitel 2

Dies klingt wie BASIC, warum sieht es nicht vertraut aus?	33
Die Elemente der »VBA-Sprechweise«	34
VBA ist nicht wirklich schwierig	38
VBA-Hilfedateien: Verwenden Sie F1, um alles zu finden	38
Verwenden der Hilfethemen	39
Untersuchen des aufgezeichneten Makrocodes: VB-Editor und die Hilfe verwenden	40
Optionale Parameter	41
Definierte Konstanten	42
Eigenschaften können Objekte zurückgeben	45
Verwenden der Debugging-Werkzeuge, um aufgezeichneten Code zu untersuchen	46
Schrittweise Codeausführung	46
Weitere Debugging-Optionen: Haltepunkte	48
Codeausführung an anderer Stelle fortsetzen	48
Nicht jede Codezeile einzeln ausführen lassen	49
Werte abfragen, während Sie schrittweise durch den Code gehen	49
Überwachung für ein Objekt verwenden	52

Objektkatalog: Die ultimative Referenz	53
Sieben Tipps für das Aufräumen aufgezeichneten Codes	54
Tipp 1: Wählen Sie nichts aus	54
Tipp 2: Verwenden Sie Cells(2,5), weil es bequemer ist als Range("E2") ...	55
Tipp 3: Verwenden Sie zuverlässigere Wege, um die letzte Zeile zu finden	56
Tipp 4: Verwenden Sie Variablen, um hartverdrahtete Zeilen und Formeln zu vermeiden	57
Tipp 5: Verwenden Sie R1C1-Formeln, die Ihnen das Leben vereinfachen	58
Tipp 6: Kopieren und Einfügen in einer Anweisung	58
Tipp 7: Verwenden Sie »With...End With«, um mehrere Aktionen durchzuführen	58
Nächste Schritte	62

Kapitel 3

Auf Bereiche verweisen	63
Das Range-Objekt	64
Syntax für die Angabe eines Bereichs	64
Auf benannte Bereiche verweisen	65
Kurzschreibweise für Bereichsadressen	65
Bezüge auf Bereiche in anderen Arbeitsblättern	65
Einen Bereichsbezug relativ zu einem anderen Bereich verwenden	66
Die Cells-Eigenschaft verwenden, um einen Bereich auszuwählen	67
Die Eigenschaft Offset verwenden, um auf einen Bereich zu verweisen	68
Die Eigenschaft Resize verwenden, um die Größe eines Bereichs zu ändern ...	70
Die Eigenschaften Columns und Rows verwenden, um einen Bereich anzugeben	72
Die Methode Union verwenden, um mehrere Bereiche zusammenzufassen ...	72
Die Methode Intersect verwenden, um aus überlappenden Bereichen einen neuen Bereich zu erstellen	73
Die Funktion IsEmpty verwenden, um zu prüfen, ob eine Zelle leer ist	73
Die Eigenschaft CurrentRegion verwenden, um einen Datenbereich auszuwählen	74
Die Auflistung Areas verwenden, um einen nicht zusammenhängenden Bereich zurückzugeben	78
Verweisen auf Tabellen	78
Nächste Schritte	79

Kapitel 4

Schleifen und Verzweigungen	81
For...Next-Schleifen	82
Variablen in der For-Anweisung verwenden	84
Variationen der For...Next-Schleife	85
Eine Schleife vorzeitig beenden, wenn eine Bedingung erfüllt ist	86
Eine Schleife innerhalb einer anderen Schleife verschachteln	87
Do-Schleifen	88
In Do-Schleifen die While- oder Until-Klausel verwenden	91
While...Wend-Schleifen	93
Die VBA-Schleife: For...Each	93
Objektvariablen	93
Verzweigungen: If...Then...Else und Select Case verwenden	95
Einfache Verzweigungen: If...Then...Else	95
Bedingungen verwenden	96
If...Then...End If verwenden	96
Entweder-oder-Entscheidungen: If...Then...Else...End If	96
If...Elseif...End If für mehrere Bedingungen verwenden	97
Select Case...End Select für mehrere Bedingungen verwenden	97
Komplexe Ausdrücke in Case-Anweisungen	98
If-Anweisungen verschachteln	99
Nächste Schritte	101

Kapitel 5

Formeln und die R1C1-Bezugsart	103
R1C1-Begzugsart aktivieren und deaktivieren	104
Das Wunder der Excel-Formeln erleben	105
Eine Formel einmal eingeben und 1000-mal kopieren	105
Das Geheimnis: Es ist nicht so erstaunlich	106
Die R1C1-Bezugsart verstehen	108
R1C1-Notation mit relativen Verweisen verwenden	108
R1C1-Notation mit absoluten Verweisen	110
R1C1-Notation mit gemischten Verweisen	110
Im R1C1-Format auf komplette Spalten oder Zeilen verweisen	111
Viele A1-Formeln durch eine R1C1-Formel ersetzen	111
Eine interessante Erkenntnis	113
Welche Spaltennummer gehören zu den Spaltenbuchstaben?	113
R1C1-Formeln mit Arrayformeln verwenden	114
Nächste Schritte	115

Kapitel 6

In VBA Namen erstellen und bearbeiten	117
Globale versus lokale Namen	117
Namen hinzufügen	118
Namen löschen	120
Kommentare hinzufügen	120
Arten von Namen	121
Formeln	121
Zeichenfolgen (Strings)	122
Zahlen	124
Tabellen	124
Arrays in Namen verwenden	125
Reservierte Namen	126
Namen ausblenden	127
Prüfen, ob ein Name existiert	127
Nächste Schritte	130

Kapitel 7

Ereignisprogrammierung	131
Ebenen von Ereignissen	131
Ereignisse verwenden	132
Ereignisparameter	133
Ereignismakros deaktivieren	133
Arbeitsmappenereignisse	134
Arbeitsblattereignisse auf Arbeitsmappenebene	136
Arbeitsblattereignisse	138
Diagrammereignisse	141
Eingebettete Diagramme	141
Ereignisse für eingebettete Diagramme und Diagrammblätter	142
Ereignisse auf der Anwendungsebene	143
Nächste Schritte	150

Kapitel 8

Arrays	151
Ein Array deklarieren	151
Ein mehrdimensionales Array deklarieren	152

Ein Array füllen	153
Daten aus einem Array abrufen	155
Arrays zur Beschleunigung des Codes verwenden	156
Dynamische Arrays verwenden	157
Ein Array übergeben	159
Nächste Schritte	160

Kapitel 9

Klassen und Auflistungen erstellen	161
Ein Klassenmodul einfügen	161
Anwendungsereignisse und Ereignisse eingebetteter Diagramme abfangen ..	162
Anwendungsereignisse	162
Ereignisse eingebetteter Diagramme	164
Ein benutzerdefiniertes Objekt erstellen	166
Ein benutzerdefiniertes Objekt verwenden	168
Auflistungen verwenden	169
Eine Auflistung erstellen	170
Eine Auflistung in einem Standardmodul erstellen	170
Eine Auflistung in einem Klassenmodul erstellen	173
Dictionaries verwenden	175
Benutzerdefinierte Typen für die Erstellung von benutzerdefinierten Eigenschaften verwenden	180
Nächste Schritte	184

Kapitel 10

UserForms: Eine Einführung	185
Eingabefelder (InputBox)	185
Meldungsfelder (MsgBox)	186
Eine UserForm erstellen	187
Eine UserForm aufrufen und ausblenden	188
UserForms programmieren	189
UserForm-Ereignisse	189
Steuerelemente programmieren	191

Einfache Formularsteuerelemente verwenden	192
Beschriftungsfelder, Textfelder und Befehlsschaltflächen verwenden	193
Entscheiden, ob ein Listenfeld oder ein Kombinationsfeld verwendet werden soll	195
Die Eigenschaft MultiSelect eines Listenfelds verwenden	197
Optionsfelder in eine UserForm einfügen	199
Grafiken in die UserForm einfügen	201
Ein Drehfeld auf einer UserForm verwenden	202
Das Multiseiten-Steuerelement verwenden, um Formulare zu kombinieren	204
Eingabe in Feldern überprüfen	207
Unerlaubtes Schließen von UserForms verhindern	208
Einen Dateinamen abfragen	209
Nächste Schritte	210

Kapitel 11

Data Mining mit dem Spezialfilter	211
Eine Schleife durch AutoFilter ersetzen	211
AutoFilter-Techniken verwenden	214
Nur sichtbare Zellen auswählen	218
Spezialfilter – In VBA einfacher als in Excel	220
In der Excel-Oberfläche einen Spezialfilter erstellen	221
Mit Spezialfilter eine Liste eindeutiger Werte extrahieren	222
Eine eindeutige Liste mit Werten in der Benutzeroberfläche extrahieren	222
Eine eindeutige Liste mit Werten mit VBA-Code extrahieren	224
Eindeutige Kombinationen aus zwei oder mehr Feldern erhalten	228
Spezialfilter mit Kriterienbereichen verwenden	229
Mehrere Kriterien durch logisches ODER verknüpfen	231
Zwei Kriterien durch ein logisches UND verknüpfen	232
Andere leicht komplexe Kriterienbereiche	232
Die komplexesten Kriterien: Die Liste der Werte durch eine Bedingung ersetzen, die das Ergebnis einer Formel ist	232
Eine Bedingung als Ergebnis einer Formel einrichten	234
Spezialfilter an gleicher Stelle filtern lassen	241
Das echte Arbeitsferd: xlFilterCopy mit allen Datensätzen statt nur mit den eindeutigen	243
Alle Spalten kopieren	243
Eine Teilmenge der Spalten kopieren und deren Reihenfolge ändern	244
Excel in der Praxis: Ein paar Dropdownmenüs im AutoFilter abschalten ..	251
Nächste Schritte	252

PivotTables mit VBA erstellen	253
Wie sich PivotTables bei den verschiedenen Excel-Versionen entwickelt haben	254
Eine PivotTable mit Excel VBA erstellen	254
Den PivotTable-Datencache definieren	255
Die PivotTable erstellen und konfigurieren	256
Felder in den Datenbereich einfügen	257
Warum Sie keinen Teil eines PivotTable-Berichts verschieben oder ändern können	260
Die Größe einer fertigen PivotTable ermitteln, um die PivotTable in Werte zu konvertieren	260
Fortgeschrittene PivotTable-Features verwenden	263
Mehrere Wertefelder verwenden	264
Die Anzahl der Datensätze zählen	264
Die Tagesdaten in Monaten, Quartalen oder Jahren gruppieren	265
Die Berechnung ändern, um Prozentwerte anzuzeigen	267
Leerzellen im Wertebereich eliminieren	269
Die Sortierreihenfolge mit AutoSort steuern	270
Den Bericht für jedes Produkt vervielfältigen	270
Ein Dataset filtern	274
Zwei oder mehr Elemente in einem Pivot-Feld filtern	274
Die erweiterten Filter verwenden	275
Den Suchfilter verwenden	280
Datenschnitte zur Filterung einer PivotTable einrichten	283
Eine Zeitachse einrichten, um eine Excel 2019-PivotTable zu filtern	288
Das Datenmodell in Excel 2019 verwenden	291
Beide Tabellen zum Datenmodell hinzufügen	291
Eine Beziehung zwischen den beiden Tabellen erstellen	292
Den Pivot-Cache definieren und die PivotTable erstellen	293
Felder aus dem Datenmodell in die PivotTable einfügen	293
Numerische Felder in den Wertebereich einfügen	293
Alles zusammenfügen	295
Andere PivotTable-Features verwenden	297
Berechnete Datenfelder	297
Berechnete Elemente	298
ShowDetail verwenden, um ein Recordset zu filtern	298
Das Layout mit der Registerkarte »Entwurf« ändern	298
Einstellungen für das Berichtslayout	299
Teilergebnisse für mehrere Zeilenfelder ausblenden	300
Nächste Schritte	301

Kapitel 13

Power-Tools und Tricks mit Excel VBA	303
Dateioperationen	303
Dateien in einem Verzeichnis auflisten	303
CSV-Datei importieren und löschen	306
Eine Textdatei in den Speicher einlesen und parsen	307
Arbeitsmappen kombinieren und aufteilen	308
Arbeitsblätter einer Arbeitsmappe in jeweils eigenen Arbeitsmappen speichern	308
Arbeitsmappen kombinieren	309
Daten ohne Filter auf verschiedene Arbeitsblätter kopieren	311
Daten in eine XML-Datei exportieren	312
Mit Zellkommentaren arbeiten	314
Größe der Kommentare ändern	314
Ein Diagramm in einen Kommentar einfügen	315
Änderungen nachverfolgen	317
Techniken von und für VBA-Profis	318
Ein Klassenmodul für den Excel-Status erstellen	318
Drilldown für eine PivotTable ausführen	321
Eine OLAP-PivotTable anhand einer Liste von Elementen filtern	322
Eine benutzerdefinierte Sortierreihenfolge erstellen	325
Fortschrittsanzeige in einer Zelle erstellen	326
Geschütztes Kennwort-Eingabedialogfeld verwenden	327
Groß-/Kleinschreibung ändern	330
Zellen mit SpecialCells auswählen	331
Format einer Tabelle zurücksetzen	332
Mit VBA Extensibility Code in neue Arbeitsmappen einfügen	333
Nächste Schritte	335

Kapitel 14

Beispiele für benutzerdefinierte Funktionen	337
Benutzerdefinierte Funktionen erstellen	337
Eine einfache benutzerdefinierte Funktion erstellen	338
Benutzerdefinierte Funktionen bereitstellen	340
Nützliche benutzerdefinierte Excel-Funktionen	340
Den Dateinamen der aktiven Arbeitsmappe in eine Zelle einfügen	341
Den kompletten Pfad der aktiven Arbeitsmappe in eine Zelle einfügen	341
Prüfen, ob eine Arbeitsmappe geöffnet ist	341
Prüfen, ob ein Arbeitsblatt in einer geöffneten Arbeitsmappe existiert	342
Anzahl der Arbeitsmappen in einem Ordner zählen	343
Windows-Anmeldennamen ermitteln	344

Datum und Uhrzeit des letzten Speicherns	346
Datum und Uhrzeit als dauerhafte Werte	346
Eine E-Mail-Adresse überprüfen	347
Zellen mit einer bestimmten Füllfarbe addieren	349
Eindeutige Werte zählen	350
Duplikate aus einem Bereich entfernen	351
Die erste Zelle mit einer Länge ungleich 0 in einem Zellbereich finden ..	353
Mehrere Zeichen suchen und ersetzen	354
Zahlen aus gemischtem Text extrahieren	356
Wochennummern in Datum konvertieren	356
Ein Element aus einer Zeichenfolge mit Trennzeichen extrahieren	357
Sortieren und verketten	358
Zahlen und Buchstaben sortieren	360
Nach einer Zeichenfolge in Text suchen	362
Den Inhalt einer Zelle rückwärts ausgeben	363
Die Adressen von doppelten Maximalwerten zurückgeben	363
Die Adresse eines Links ermitteln	365
Den Spaltenbuchstaben einer Zelladresse ermitteln	365
Statische Zufallszahlen verwenden	366
Select...Case auf einem Arbeitsblatt verwenden	366
Nächste Schritte	367

Kapitel 15

Diagramme erstellen	369
Ein Diagramm mit AddChart2 erstellen	370
Diagrammformatvorlagen verstehen	371
Ein Diagramm formatieren	374
Ein bestimmtes Diagramm ansprechen	375
Einen Diagrammtitel festlegen	376
Diagrammfarbe zuweisen	376
Ein Diagramm filtern	378
SetElement verwenden, um Änderungen im Menü »Diagramm- elemente« (Plussymbol) zu emulieren	379
Die Methode »Format« verwenden, um Formatierungsoptionen im Detail einzustellen	384
Die Füllung eines Objekts ändern	385
LinienEinstellungen formatieren	387
Ein Verbunddiagramm erstellen	388
Flächenkartogramme erstellen	392
Wasserfalldiagramme erstellen	393
Ein Diagramm als Grafik exportieren	395
Abwärtskompatibilität berücksichtigen	395
Nächste Schritte	396

Kapitel 16

Datenvisualisierungen und bedingte Formatierung	397
VBA-Methoden und Eigenschaften für Datenvisualisierungen	399
Einen Bereich mit Datenbalken versehen	400
Einen Bereich mit Farbskalen versehen	405
Einen Bereich mit einem Symbolsatz versehen	406
Einen Symbolsatz festlegen	407
Wertebereiche der einzelnen Symbole festlegen	408
Visualisierungstricks verwenden	409
Einen Symbolsatz für eine Teilmenge eines Zellbereichs erstellen	409
In einem Zellbereich zwei Datenbalken mit verschiedenen Farben verwenden	411
Andere Methoden zur bedingten Formatierung verwenden	414
Zellen formatieren, die über oder unter dem Durchschnitt liegen	414
Die zehn größten oder fünf kleinsten Elemente formatieren	414
Eindeutige oder doppelte Zellen formatieren	415
Zellen basierend auf ihrem Wert formatieren	417
Zellen formatieren, die Text enthalten	418
Zellen formatieren, die Datumswerte enthalten	418
Zellen formatieren, die leer sind oder Fehler enthalten	418
Eine Formel verwenden, um festzulegen, welche Zellen formatiert werden sollen	419
Die neue Eigenschaft »NumberFormat« verwenden	420
Nächste Schritte	422

Kapitel 17

Dashboards mit Sparklines in Excel 2019	423
Sparklines erstellen	424
Sparklines skalieren	426
Sparklines formatieren	430
Designfarben verwenden	430
RGB-Farben verwenden	434
Sparkline-Elemente formatieren	435
Gewinn-Verlust-Diagramme formatieren	438
Ein Dashboard erstellen	440
Beobachtungen über Sparklines	441
In einem Dashboard Hunderte einzelne Sparklines erstellen	441
Nächste Schritte	446

Kapitel 18

Aus dem Web lesen und in das Web schreiben	447
Daten aus dem Web abrufen	447
Mehrere Abfragen mit VBA erstellen	449
Ergebnisse in den abgerufenen Daten finden	450
Alles zusammenbauen	452
Beispiele für das Scraping von Websites mittels Webabfragen	453
»Application.OnTime« verwenden, um in bestimmten Intervallen	
Daten zu analysieren	454
Der Modus »Bereit« und geplante Prozeduren	455
Ein Zeitfenster für ein Update angeben	455
Ein zuvor eingeplantes Makro löschen	455
Das Beenden von Excel löscht alle eingeplanten Makros	456
Festlegen, dass ein Makro in x Minuten in der Zukunft ausgeführt wird	456
Eine gesprochene Erinnerung einplanen	457
Ein Makro alle zwei Minuten ausführen	459
Daten als Webseite veröffentlichen	459
Mit VBA benutzerdefinierte Webseiten erstellen	461
Excel als Content-Management-System verwenden	462
Bonus: FTP aus Excel heraus	465
Nächste Schritte	466

Kapitel 19

Textdateien verarbeiten	467
Textdateien importieren	467
Textdateien mit weniger als 1.048.576 Zeilen importieren	467
Öffnen einer Datei mit Trennzeichen	471
Textdateien mit mehr als 1.048.576 Zeilen einlesen	474
Mit Power Query große Dateien in das Datenmodell laden	479
Textdateien schreiben	480
Nächste Schritte	480

Kapitel 20

Word automatisieren	481
Mit früher Bindung auf ein Word-Objekt zugreifen	481
Mit später Bindung auf ein Word-Objekt zugreifen	484
Mit dem Schlüsselwort »New« die Word-Anwendung referenzieren	485
Mit der Funktion »CreateObject« eine neue Instanz eines Objekts erzeugen	486

Mit der Funktion »GetObject« eine vorhandene Instanz von Word referenzieren	486
Konstanten verwenden	487
Das Fenster »Überwachungsausdrücke« verwenden, um den echten Wert einer Konstanten zu ermitteln	487
Den Objektkatalog verwenden, um den echten Wert einer Konstanten zu ermitteln	488
Words Objekte verstehen	489
Das Document-Objekt	490
Das Selection-Objekt	492
Das Range-Objekt	493
Textmarken	497
Word-Formularfelder steuern	499
Nächste Schritte	502

Kapitel 21

Access als Backend verwenden, um den Mehrbenutzerzugriff auf die Daten zu verbessern	503
ADO versus DAO	504
Die Werkzeuge von ADO	506
Einen Datensatz in eine Datenbank einfügen	508
Datensätze aus einer Datenbank abrufen	510
Einen vorhandenen Datensatz aktualisieren	513
Datensätze mit ADO löschen	515
Datensätze mit ADO zusammenfassen	515
Weitere Möglichkeiten mit ADO	517
Die Existenz von Tabellen überprüfen	517
Die Existenz eines Felds in einer Tabelle überprüfen	518
Auf die Schnelle eine Tabelle hinzufügen	519
Auf die Schnelle ein neues Feld in eine Tabelle einfügen	520
SQL Server-Beispiele	521
Nächste Schritte	522

Kapitel 22

Fortgeschrittene UserForm-Techniken	523
Die Symbolleiste »UserForm« während des Entwurfs von Steuerelementen auf UserForms verwenden	523

Weitere Steuerelemente für UserForms	524
Kontrollkästchen-Steuerelemente (CheckBox)	524
Register-Steuerelement (TabStrip)	526
RefEdit-Steuerelemente	528
Umschaltfeld-Steuerelement (ToggleButton)	530
Eine Bildlaufleiste als Schieberegler verwenden, um Werte auszuwählen	531
Steuerelemente und Auflistungen	533
Ungebundene UserForms	535
Hyperlinks in UserForms verwenden	536
Zur Laufzeit Steuerelemente hinzufügen	537
Größe des Formulars zur Laufzeit anpassen	539
Steuerelemente zur Laufzeit einfügen	539
Größe der Steuerelemente zur Laufzeit festlegen	540
Andere Steuerelemente hinzufügen	540
Ein Bild zur Laufzeit einfügen	540
Die Einzelteile zusammenfügen	541
Eine UserForm mit Hilfe-Features versehen	544
Zugriffstasten anzeigen	544
Ein Steuerelement mit einer QuickInfo versehen	545
Die Aktivierreihenfolge festlegen	545
Das aktive Steuerelement einfärben	546
Transparente Formulare erstellen	549
Nächste Schritte	550

Kapitel 23

Die Windows Application Programming Interface, API	551
Eine API-Deklaration verstehen	552
Eine API-Deklaration verwenden	553
32-Bit- und 64-Bit-kompatible API-Deklarationen erstellen	554
Beispiele für API-Funktionen	555
Den Computernamen ermitteln	555
Prüfen, ob eine Excel-Datei im Netzwerk geöffnet ist	556
Informationen über die Bildschirmauflösung abrufen	557
Das Dialogfeld »Info« anpassen	558
In einer UserForm die Schließen-Schaltfläche (X) deaktivieren	559
Eine Stoppuhr programmieren	560
Sounds abspielen	561
Nächste Schritte	561

Kapitel 24

Fehlerbehandlung	563
Was passiert, wenn ein Fehler auftritt?	563
Ein irreführender Debug-Fehler in UserForm-Code	565
Elementare Fehlerbehandlung mit »OnError GoTo«	567
Allgemeine Fehlerbehandlungsroutinen	569
Fehler behandeln, indem sie ignoriert werden	569
Excel-Warnungen unterdrücken	571
Fehler zu Ihrem Vorteil nutzen	572
Schulen Sie Ihre Kunden	573
Fehler, die im Debug-Modus nicht auftreten	573
Fehler während der Entwicklung versus Fehler Monate später	574
Laufzeitfehler 9: Index außerhalb des gültigen Bereichs	574
Laufzeitfehler 1004: Methode 'Range' für das Objekt '_Global' ist fehlgeschlagen	575
Das Übel von geschütztem Code	576
Weitere Probleme mit Kennwörtern	578
Fehler, die durch Unterschiede in den Excel-Versionen verursacht werden	579
Nächste Schritte	579

Kapitel 25

Das Menüband anpassen, um Makros auszuführen	581
Wohin mit dem Code? Der Ordner und die Datei customui	582
Eine Registerkarte und eine Gruppe erstellen	584
Ein Steuerelement in das Menüband einfügen	584
Auf die Dateistruktur zugreifen	590
Die Datei .rels verstehen	591
Die Excel-Datei wieder umbenennen und die Arbeitsmappe öffnen	592
Grafiken auf Schaltflächen verwenden	592
Microsoft Office-Icons im Menüband verwenden	593
Benutzerdefinierte Grafiken in das Menüband einfügen	594
Problembehandlung bei Fehlermeldungen	595
Das Attribut »Attributname« im Element »benutzerdefiniertes Menüband« ist im DTD/Schema nicht definiert	596
Ungültiges qualifiziertes Zeichen	596

Inhalt des Elements »TagName in customui« ist gemäß dem Inhaltsmodell des übergeordneten Elements »TagName in customui« nicht gültig	597
Problem bei einigen Inhalten gefunden	597
Falsche Anzahl von Argumenten oder ungültige Eigenschaftszuweisung	598
Ungültiges Dateiformat oder falsche Dateierweiterung	599
Es passiert nichts	599
Andere Möglichkeiten, Makros zu starten	599
Makro mit einer Tastenkombination starten	600
Makro mit einer Befehlsschaltfläche starten	601
Makro mit einer Form verknüpfen	601
Makro mit einem ActiveX-Steuerelement verknüpfen	602
Ein Makro über einen Link starten	603
Nächste Schritte	604

Kapitel 26

Add-Ins erstellen	605
Merkmale von Standard-Add-Ins	605
Eine Excel-Arbeitsmappe in ein Add-In konvertieren	606
Mit »Speichern unter« eine Datei in ein Add-In konvertieren	607
Mit dem VB-Editor eine Datei in ein Add-In konvertieren	608
Installation des Add-Ins beim Kunden	609
Standard-Add-Ins sind nicht sicher	611
Add-Ins schließen	612
Add-Ins entfernen	612
Eine ausgeblendete Arbeitsmappe als Alternative zu einem Add-In verwenden	612
Nächste Schritte	614

Kapitel 27

Eine Einführung in die Erstellung von Office-Add-Ins	615
Ihr erstes Office-Add-In erstellen: Hello World	616
Dem Office-Add-In Interaktivität hinzufügen	621
Eine grundlegende Einführung in HTML	624
Tags verwenden	624
Schaltflächen hinzufügen	625
CSS-Dateien verwenden	626

XML verwenden, um ein Office-Add-In zu definieren	626
Office-Add-Ins mit JavaScript interaktiv machen	627
Der Aufbau einer Funktion	627
Geschweifte Klammern und Leerzeichen	628
Semikola und Zeilenwechsel	628
Kommentare	628
Variablen	629
Zeichenfolgen (Strings)	629
Arrays	630
for-Schleifen in JavaScript	631
if-Anweisungen in JavaScript	632
Select...Case-Anweisungen in JavaScript	632
For each...next-Anweisungen in JavaScript	634
Mathematische, logische und Zuweisungsoperatoren	635
Mathematische Funktionen in JavaScript	636
Ausgabe in den Inhaltsbereich oder Aufgabenbereich	638
JavaScript-Besonderheiten für Office-Add-Ins	638
Ein Office-Add-In initialisieren	638
Daten aus einem Arbeitsblatt lesen und in ein Arbeitsblatt schreiben	639
Nächste Schritte	639

Kapitel 28

Neuerungen und Änderungen in Excel 2019	641
Office 365-Abonnement versus Excel 2019 für den einmaligen Kauf	641
Wenn es sich im Frontend geändert hat, dann hat es sich in VBA geändert ...	641
Das Menüband	642
Single Document Interface (SDI)	642
Moderne Arrayformeln	643
Schnellanalyse	644
Diagramme	644
PivotTables	644
Datenschnitte	645
Piktogramme	645
3D-Modelle	645
SmartArt	645
Neue Objekte und Methoden kennenlernen	646
Kompatibilitätsmodus	646
Die Eigenschaft »Version« verwenden	647
Die Eigenschaft »Excel8CompatibilityMode« verwenden	648
Nächste Schritte	649
Index	651