

Excel Seminare

Release 1.1-2023

Joe Brandes

03.02.2024

1	Excel Einführung	3
2	Excel Basics	5
2.1	Excel Versionen	6
2.2	Mappentypen	8
2.3	Tabellenblätter	8
2.4	Zellen	10
2.5	Automatisches Ausfüllen	11
2.6	Tipps & Tricks	13
2.7	Formatierungen	14
2.8	Bedingte Formatierung	15
2.9	Als Tabelle formatieren	16
2.10	Zellbezüge	17
2.11	Große Tabellen	18
2.12	Kommentar / Notizen	21
2.13	Diagramme 101	22
3	Excel Funktionen	25
3.1	Standardfunktionen	27
3.2	SUMME()	28
3.3	WENN()	29
3.4	SVERWEIS()	30
3.5	TEILERGEBNIS()	31
3.6	Datenbankfunktionen	32
3.7	KGRÖSSTE (engl. LARGE)	32
3.8	SORTIEREN(), EINDEUTIG()	34
3.9	FILTER() - Excel 365	35
4	Pivot Technik	37
4.1	PivotTable erstellen	38
4.2	PivotChart erstellen	39
4.3	Pivot Gruppierungen	40
4.4	Datenschnitt / Zeitachse	40
4.5	Pivot Daten Export	41
4.6	Pivot Measure	42
4.7	Pivot - Beziehungen	43

4.8	Power Pivot	44
5	Diagramme 102	47
5.1	Diagrammtypen	48
5.2	Verbunddiagramm	48
5.3	Diagrammvorlagen	49
5.4	Daten zu Diagramm hinzufügen	51
5.5	Datenvisualisierung	51
5.6	Animation mit PowerPoint	52
6	Arbeitsmappenvorlagen	55
7	Was-wäre-wenn-Analysen	57
7.1	Szenarien	58
7.2	Datentabellen	58
7.3	Zielwertsuche	59
8	Makros mit Excel	61
8.1	Vorlagen - Rekap	62
8.2	Vorlagen - mit Makros	63
8.3	PERSONAL.XLSB	63
8.4	Makros - Sicherheit	64
8.5	Makros - Aufzeichnen	65
8.6	Absolut vs Relativ	67
9	Visual Basic for Applications	69
9.1	VBA-Editor	69
9.2	Direktfenster	70
9.2.1	Berechnungen	71
9.2.2	Debug Ausgaben	72
9.3	Eigenschaften	72
9.4	Variablen	73
9.5	Konstanten	75
9.6	Kontrollstrukturen	75
9.6.1	Verzweigungen	76
9.6.2	Schleifen	76
9.6.3	Einfache Codes Kontrollstrukturen	76
9.7	Objektorientierung	78
9.8	Workbook_Open()	79
9.9	Tabellen ansprechen	80
9.10	VBA Beispiele	82
9.10.1	VBA Beispiel 01	82
9.10.2	VBA Beispiel 02	82
9.11	UserForms	83
9.11.1	Herdt-Taschenrechner	84
9.11.2	Eingabemasken mit Excel	84
9.12	Eigene Funktion mit VBA	85
9.13	Textdateien einlesen	85
10	Was mit Medien	87
10.1	Seminar	88
10.2	Tabellenformate	89

10.2.1	Excel-Dateiformate	89
10.2.2	CSV	90
10.2.3	JSON	91
10.2.4	XML	93
10.3	Import von Daten	94
10.3.1	Import-Quellen	95
10.3.2	CSV-Import LibreOffice	97
10.3.3	CSV-Import Excel	97
10.3.4	CSV-Import Excel WebApp	99
10.3.5	CLI Tools	100
10.4	Grafische Darstellungen	101
10.5	Daten verändern / auswerten	103
10.6	Export von Daten	104
10.7	Arbeitsauftrag	105
11	Data Literacy	107
11.1	Praxistag	108
12	Linksammlung	111
13	Bücher	113
13.1	Excel - Formeln und Funktionen	114
13.2	Excel Projektmanagement	115
13.3	Business Intelligence mit Excel	116
13.4	M - Daten abfragen und verarbeiten	117
13.5	Excel programmieren	118
13.6	VBA mit Excel	119
13.7	VBA Referenz	120
13.8	Datenvisualisierung mit Excel	121
14	Bildinfos	123
14.1	Unsplash	124
15	Dokumenteninfos	125
	Stichwortverzeichnis	133

Wer sofort mit den Excel Seminarunterlagen loslegen will: weiter zu Kapitel *Excel Einführung*

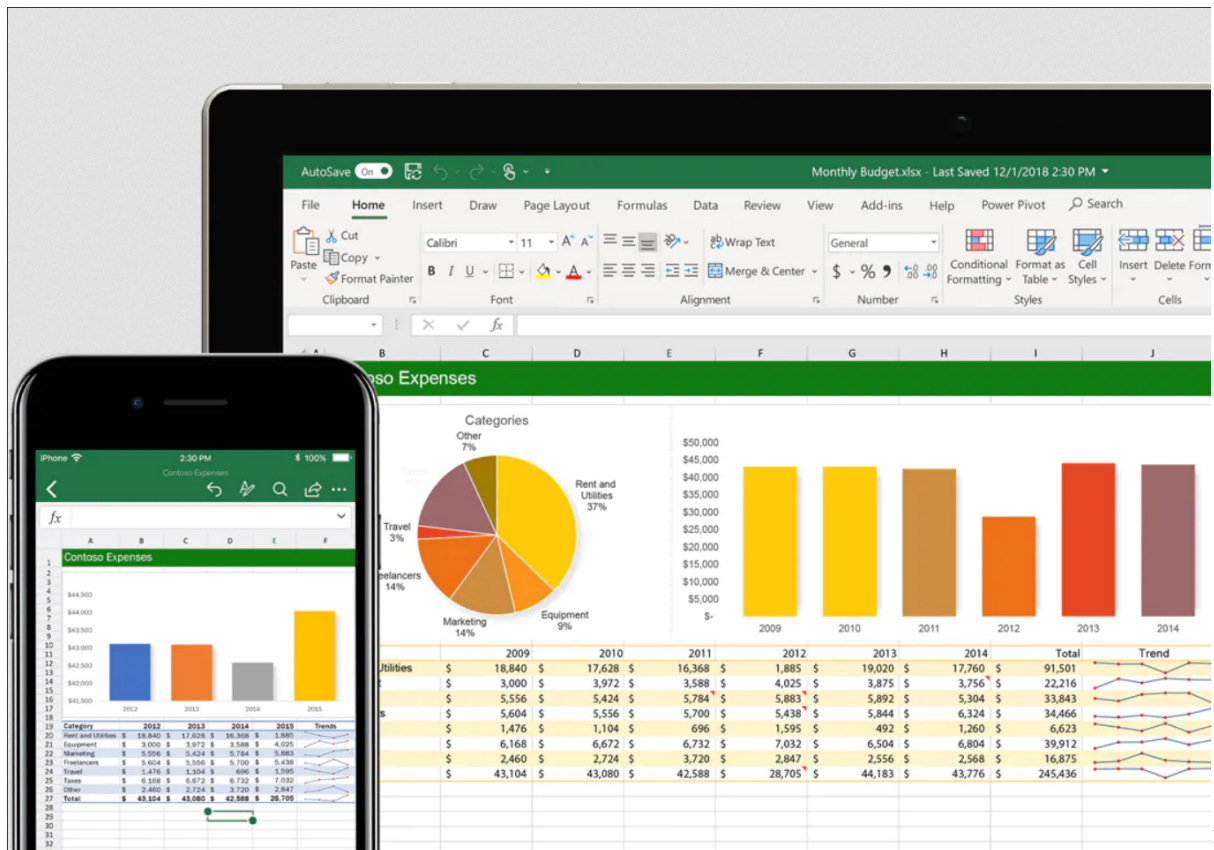


Abb. 1: Excel Screenshot (Microsoft Website); Quelle: microsoft.com/excel

Mit dieser Unterlage will ich meine bisherigen Online- Veröffentlichungen zum Thema **Excel** aufbereiten.

Die Zielgruppe sind Teilnehmer (Trainees) für diverse Seminare mit Themenbereich **Excel**. Diese Ausarbeitung soll keines der hier verlinkten Bücher (siehe *Bücher*) ersetzen, sondern einen *Leitfaden* für meine Seminare bereitstellen.

Also bitte einfach entspannt den praktischen Übungen im Seminar folgen und die entsprechenden Topics in Ruhe nachschlagen.

Ich habe das schon für diverse andere meiner Seminare durchgeführt - das ausführlichste Beispiel hier ist sicherlich die Ausarbeitung zu **TYPO3**². Von der dortigen CMS-Technik stammen auch meine Interessen und Vertiefungen hinsichtlich **RestructuredText** Dokumentationen.

Die hier vorliegende Ausarbeitung ist so konfiguriert, dass die *Quellcodes (RST-Dateien)* für Interessierte einsehbar sind.

Hinweis: Die Unterlagen befinden sich in stetiger Überarbeitung. Bitte die Versionierungen und Datum/Uhrzeit-Stempel der Ausarbeitung beachten/verfolgen.

Die Ausarbeitungen sind kurz gehalten und ergeben oft nur in Verbindungen mit den gehalten Seminaren und dortigen Übungen und Ausführungen nachhaltig Sinn.

¹ <http://www.microsoft.com/excel>

² <http://typo3.joe-brandes.de>

Für alles Weitere verweise ich auf die verlinkte Literatur.

Themenabschnitte

Hier eine Übersicht über die behandelten Themenabschnitte rund um Excel:

1. **Excel 101**

Einführung in Excel, Excel Basics und Funktionen ...

2. **Excel 102**

Pivot, Dokumenttypen, Diagramme, ...

3. **Excel Automatisierung**

Makros und VBA

4. **Extras**

Linksammlung, Bücher, Dokumenteninfos, Bildinfos ...

KAPITEL 1

Excel Einführung

Und los geht es mit unseren Themen rund um die Tabellenkalkulation **Excel!** Und wie immer geht es mir auch um meinen/unseren **Roten Faden**.



Abb. 1.1: Roter Faden - Bild: amirali-mirhashemian-hTeYcjviZ-s-unsplash.jpg

Diesen Begriff hört man in meinen Seminaren häufiger ;-). Gemeint ist hier: Das grundsätzliche Verständnis der fraglichen IT-Techniken. Am Besten gleich so, dass man auch nach einer gewissen Zeit ohne Beschäftigung mit diesen Techniken sehr schnell wieder in Fahrt kommt.

Unter einem roten Faden versteht man ein Grundmotiv, einen leitenden Gedanken, einen Weg oder auch eine Richtlinie. „Etwas zieht sich wie ein roter Faden durch etwas“ bedeutet beispielsweise, dass man darin eine durchgehende Struktur oder ein Ziel erkennen kann.

Quelle: [Wikipedia - Roter Faden](#)³

Oder in anderen Worten:

Wichtig: Man sollte sich auch bei Excel immer möglichst an die **Best Practises** halten. Das soll beispielsweise heißen:

- Zellen sauber bearbeiten und beenden
 - Workflow beim Erstellen von Tabellen beachten
 - Nutzung von cleveren Tastenkombinationen
-

Im ersten Abschnitt wollen wir uns diesen **Excel Basics** widmen.

³ https://de.wikipedia.org/wiki/Faden#Roter_Faden

Besonders in einführenden Veranstaltungen wollen wir uns auf die Praxis mit den folgenden **Excel-Grundlagen-Themen** konzentrieren.



Abb. 2.1: Basics - Bild: lauren-richmond-490uCO8h7ZA-unsplash.jpg

Im Mittelpunkt soll immer *Learning by Doing* stehen - also die praktische Anwendung und Erprobung der Exceltechniken.

Und los geht es mit einem Einstieg und mit den folgenden Themen.

2.1 Excel Versionen

Für die meisten Beschäftigungen mit **Excel** oder auch allgemein Tabellenkalkulationen kann man sich an Grundtechniken und Grundfunktionen halten, die sich seit der Einführung von Excel mitte der 1980er Jahre oder sogar bei Vorgängern wie **Multiplan** nutzen lassen.

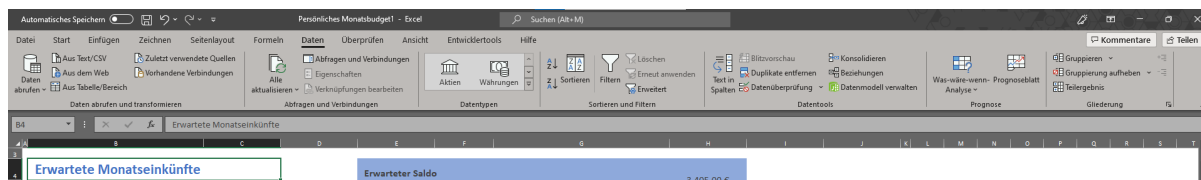


Abb. 2.2: Excel 365 (Jahrgang 2022)

Infolink zu Wikipedia Beitrag zu Excel: [Wikipedia Microsoft Excel⁴](https://de.wikipedia.org/wiki/Microsoft_Excel)

Hier nur die *grobe* Übersicht zu Excel-Versionen (zeitlich und technisch absteigend):

Excel als SAAS (Software as a Service):

- **Excel 365** verfügbar seit ca. 2011 als Teil von Microsoft 365 (ehemals Office 365)
Abo-Service mit diversen Diensten und Bezahl- und Lizenz-Modellen
Microsoft entwickelt *neue* Excel-Techniken ausschließlich für die Softwareumsetzungen im Rahmen der 365-Familie.

Die (noch erhältlichen) modernen Kaufversionen:

- **Excel 2021** ab 5. Oktober 2021
und immer noch Version 16.0 (!!)
verbesserte Verweisfunktion XVERWEIS, Dark Mode
- Excel 2019 ab 24. September 2018
aktuell letzte eigenständige Office-Kaufversion
Version: 16.0 (also Versionsgleich mit **Excel 2016!**)
- **Excel 2016** ab 22. September 2015
Designanpassung zu Windows 10, Power Query, neue Diagramme (s.u.)
- Versionen **Excel 2010** (Empfehlung nach Einführung 2007) und 2013
- Excel 2007 ab 30. Januar 2007
neue Oberfläche (Menübänder/Ribbons), größere Arbeitsblätter
und neue Dateitypen `xlsx`, `xltx`, `xlsm`

Versionen mit alten Dateitypen:

- Excel 2003 ab 17. November 2003 (für Windows)
letzte Version mit klassischen Dateitypen `xls`, `xlt`

⁴ https://de.wikipedia.org/wiki/Microsoft_Excel

- Excel 95 ab 30. August 1995 (für Windows 95)
ab hier jetzt die Versionierungen mit den Jahreszahlen
- Excel Version 2.0 ab November 1987 (für Windows 2.0)
Versionsnummern bis 5.0

Mit den Office-Suites **2013/2016** haben dann einige Office-Programme ein paar *frische* Tools und Techniken bekommen, um diese für Käufer attraktiv zu machen.

Hinweis: Alle weiteren modernen Technikangebote bleiben dann sehr oft der Office 365 Suite und damit Excel 365 vorbehalten.

Microsoft unterstützt diverse Online-/Youtube Kanäle, die immer wieder die *neuen* Excel-Techniken bewerben.

Hier ein paar Stichworte und Beispiele für Neuerungen bei Excel Versionen 2013 / 2016: (Anm.: Kaufanreize)

- Schnellanalyse
- Blitzvorschau
- Datenschnitt universell einsetzbar
- Empfohlene Pivot-Tabellen
- Tabellen-Datenmodell mit Beziehungen
- Power Pivot integriert
- Externe Datenquellen
- Zeitachse in Pivot-Auswertungen
- Prognose
- Überblick Diagramme
 - Histogramm
 - Pareto
 - Hierarchie (Treemap und Sunburst)
 - Wasserfall
 - Kastengrafik (Box and Whiskers)
 - Blasendiagramm
- 3D-Karte

2.2 Mappentypen

Wir starten mit einer Übersicht über Arbeitsmappentypen (Dateitypen) für Excel (ohne Unterstützung für aktive Inhalte / Makros).

Tab. 2.1: Excel Dateiformate (ohne. Makrotypen)

Versionen	Dateiformate
Excel 97-2003	xls - das klassische Excel-Sheet (Arbeitsmappen) xlt - die passenden Vorlagen (engl.: Templates)
Excel 2007/2010/...	xlsx - XML basiertes Dokumentformat (ohne Makros) xltx - die passenden Vorlagen

Ich erlaube mir die folgenden Anmerkungen:

Gefahr: Die Legacy Formate xls und xlt sollten nicht mehr genutzt werden, da sie proprietäre und nicht offene Formate darstellen. Wir sollten bitte nur noch die neuen (und sichereren) XML-basierten Typen nutzen.

Im Grunde handelt es sich bei einer **xlsx** Arbeitsmappe um ein *Zip-Archiv* mit einer internen Ordner-/Dokumentstruktur.

Tipp: Die **xlsx** Arbeitsmappen lassen sich mit geeigneten Packertools (z.B. 7-Zip) einfach entpacken und analysieren. So kann man beispielsweise die enthaltenen Bildmaterialien extrahieren.

2.3 Tabellenblätter

Die Tabellenblätter sind mittels **Zellen in Spalten und Zeilen** organisiert.

Zellbezüge werden mit Buchstaben für Spalten und Zahlen für Zeilen bezeichnet.

Das Bewegen/Orientieren ist sowohl mit Maus als auch Tastatur möglich. Von mir werden immer wieder Hinweise auf die effiziente Nutzung von Tastatureingaben und Tastenkombinationen geliefert.

Besonders effizient ist die Nutzung der Tastatur, wenn man an Tabellengrenzen springen möchte: **Strg** und **Cursortasten** (Pfeiltasten). So kann man mal fix mit zwei Tastenkombinationen an den unteren rechten Rand eines Arbeitsblattes springen.

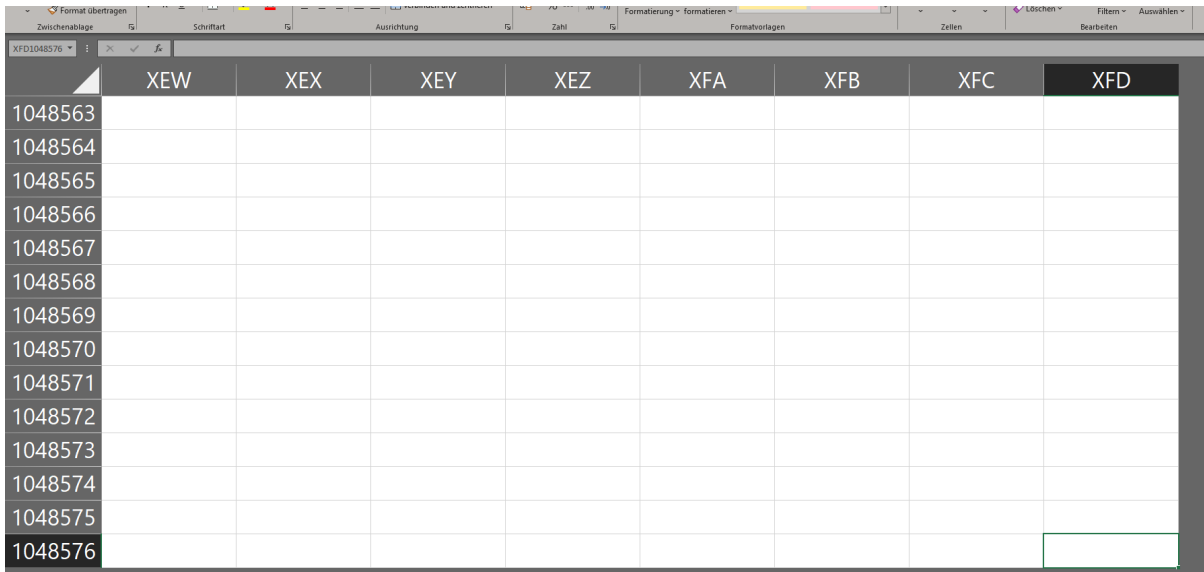


Abb. 2.3: Tabellen Dimensionen (16.384 Spalten x 1.048.576 Zeilen)

Das Markieren mit der Tastatur wird mittels Umschalt en-Taste und Cursor erreicht.

Das lässt sich clever für das effizientes Markieren großer Tabellenbereiche kombinieren.

Hier die Daten für Tabellenblätter in Excel-Tabellen:

Tab. 2.2: Excel Tabellenblatt Dimensionen

Version	Spalten	Zeilen
xls (bis 2007)	256	65.536
xlsx (ab 2010)	16.384	1.048.576

Mit dem Namensfeld (oben links) ist eine sehr schnelle Navigation zu einer Zelle möglich.

Hinweis: Die echten Inhalte der Zellen sieht man über den Inhalt der Bearbeitungsleiste (rechts vom fx-Symbol)

Übungen zu Spalten/Zeilen:

- **löschen** - Spalte (Zeile) markieren und dann Strg + Minus
- **einfügen** - genauer: davor einfügen
Spalte (Zeile) markieren und dann Strg + Plus
- **ausblenden/einblenden**
Am einfachsten über Kontextmenü (Rechte Maustaste) auf Spalten-/Zeilenkopf
Einblenden ebenfalls über Kontextmenü; ggf. mehrere Spalten/Zeilen markieren.

Mehrere Tabellenblätter ergeben eine **Arbeitsmappe**. Für das Speichern von Arbeitsmappen stehen verschiedene Formate zur Verfügung.

2.4 Zellen

Zellen können **drei Inhalte** aufnehmen:

1. **Texte**
2. **Zahlen**
3. **Formeln** - beginnen immer mit einem Gleichheitszeichen „=“

Empfehlung zur Eingabe von Zellinhalten:

- Zahlen werden immer sauber ohne Größenangaben und Währungen eingegeben.
- Eingaben immer sauber mit Return beenden.

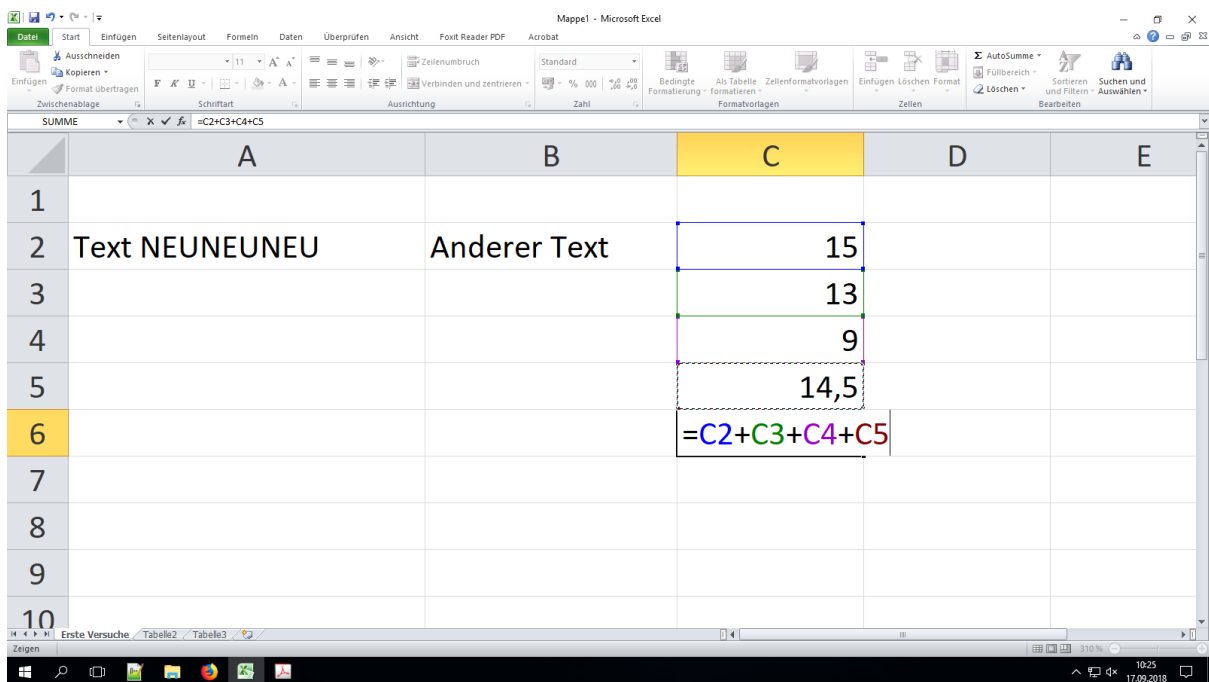


Abb. 2.4: Text - Zahl - Formel

Spezialformat *Zahlzelle* für Datum und Uhrzeit:

Datum und Uhrzeit sind spezielle Zahlenformate (siehe Rechte Maus - Zellen formatieren. . .) mit Bezug auf den 01.01.1900 bei Microsoft Systemen.

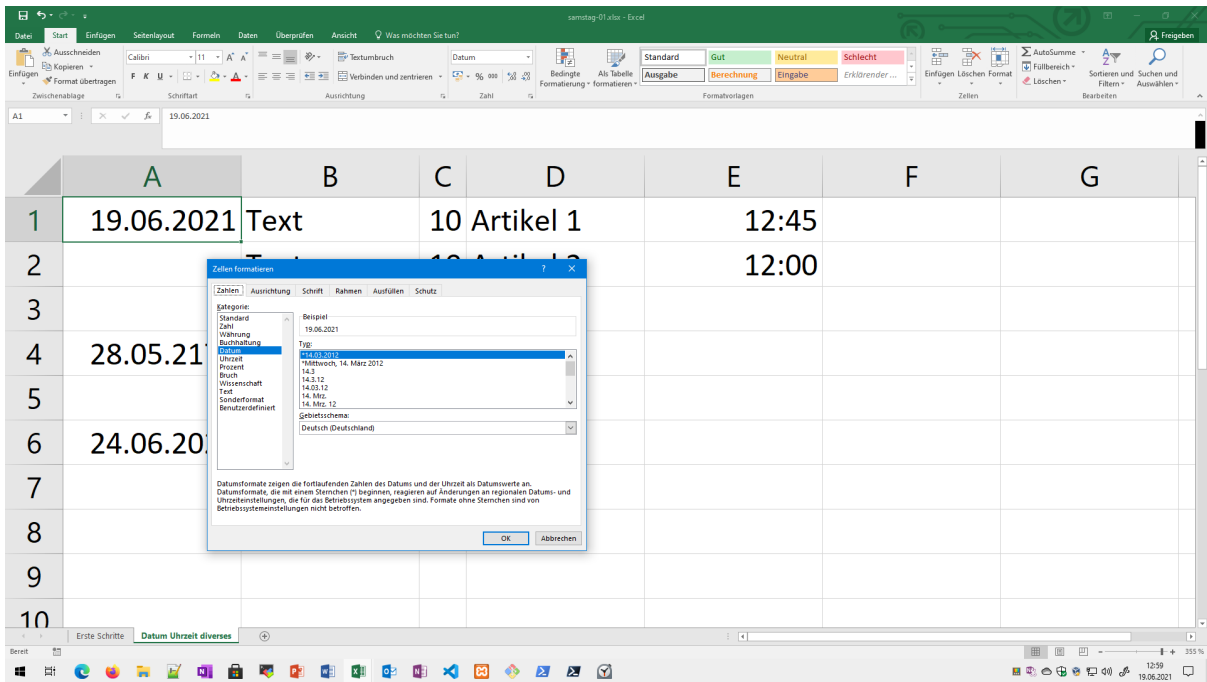


Abb. 2.5: Zahlenformat Datum und Uhrzeit

Es sind keine negativen Uhrzeitformate möglich (Hinweis auf ##### in Zellen).

Anm./Hilfe bei MacOS-Excel-Mappen: über Optionen - Erweitert - 1904 Datumswerte verwenden (Rubrik: Beim Berechnen dieser Arbeitsmappe)

Erste einfache Formeln mit Einsatz von „Arithmetischen Operatoren“ ;-)- also:

- + (Plus)
- - (Minus)
- / (Division/Geteilt)
- * (Multiplikation/Mal)

Man findet diese **Arithmetischen Operatoren** (oft) auf dem Zehnerblock einer Tastatur (in Richtung entgegen Uhr).

2.5 Automatisches Ausfüllen

Vorab ein wichtiger Hintergrund zum Verständnis und später für die Nutzung der Techniken mit Formeln.

Technisch stellen die folgenden Techniken das *Kopieren* mit der Maus dar!

Praxis: Ziehen mit kleinem Fadenkreuz an Zelle rechts unten!

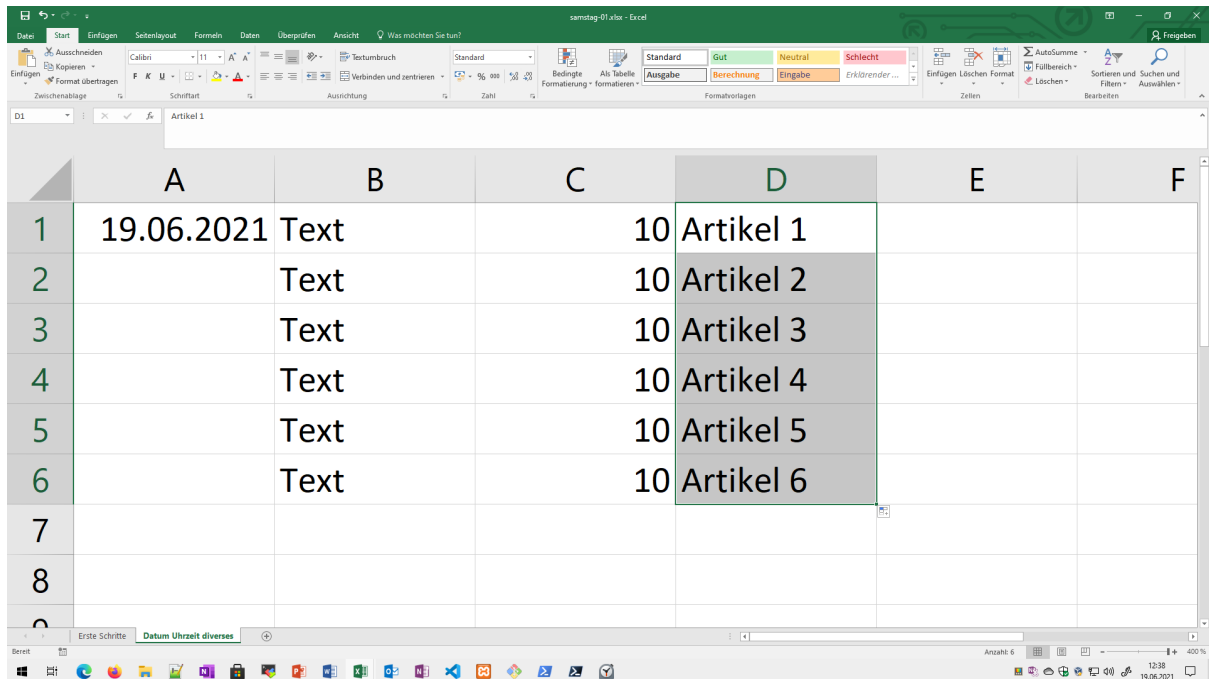


Abb. 2.6: Automatisches Ausfüllen mit Anfasser (kleines Fadenkreuz unten rechts)

Übungen:

- mit Monaten
- Wochentagen und
- Zahlenformaten

Tip: Für Zahlen Startwert und Schrittweite eingeben - dann **beide Zellen markieren** und ziehen.

Das Ziehen ist mit der linken und **rechten** Maustaste möglich! Bei der Rechten Maustaste erhält man noch ein Auswahlm Menü/Kontextmenü.

Benutzerdefinierte Listen

Eine besondere Funktion erhält man im Zusammenhang mit Benutzerdefinierten Listen:

- Excel 2007: Excel-Optionen - Kategorie Häufig verwendet - Benutzerdefinierte Listen
- ab Excel 2010 / 2013: Excel-Optionen - Kategorie Erweitert - nach unten scrollen - Benutzerdefinierte Listen

2.6 Tipps & Tricks

Wir wollen ein paar Eingabetricks für Zellen nutzen!

Text für Zellen erzwingen

Zelleingaben mit ' beginnen: Zellen werden automatisch 1-zu-1 als Text übernommen (z.B. '007)

	A	B	C	D	E	F
1	Texte	15	15		3	
2		33	33		4	
3		10 Liter	10 Liter		5	
4		3,5	3,5		= (E1+E2)*E3	
5		#WERT!	51,5			
6					Eingabe mit '	
7	Manuell:	mit Strg+.	=HEUTE()		007	7
8	08.04.2019	08.04.2019	08.04.2019		08/04	08. Apr

Abb. 2.7: Zellinhalte in ersten Übungen

Tipp: so lassen sich auch „Formeln“ in Zellen eintragen

Manuelle Umbrüche in Zellen

Manuelle Zeilenumbrüche in Zellen mit Alt + Return

	A	B	C	D	E	F	G
1	Text	15	16		7		
2		7	7		007		
3		18	50				
4		40	73				
5							
6		15,4	Text in Zeile 1 Text in Zeile 2 Mit Alt + Return in 3				
7		15,567					
8							

Abb. 2.8: Eingabetricks für Excel Zellen

Neue Spalten/Zeilen

Neue Spalten/Zeilen mit Markieren der Gewünschten und dann mit Strg + Plus bzw. Minus einfügen/löschen

Aktuelles Datum

Aktuelles Datum mit Strg + . (Strg + Punkt)

Das Datum ist somit fix! Bitt hier den Unterschied zur Datumsfunktion HEUTE() beachten.

2.7 Formatierungen

Beispiele/Übungen:

- Buchhaltung vs. Währung (Ausrichtung in Zelle beachten)
- Dezimalstellen
- Datum & Uhrzeit

Datum / Uhrzeit

Problem / Analyse für die Nutzung von Datum und Uhrzeit in Excel:

- Datum-/Uhrzeitzellen sind speziell formatierte Zahlzellen!
- Bei Uhrzeiten kann es keine negativen Zahlenwerte als Uhrzeitformat geben!
- Zahleninhalte mit Bezug auf den 01.01.1900 bei Microsoft Systemen
- keine negativen Uhrzeitformate möglich (Hinweis auf ##### in Zellen)

Hinweis: Bei Apple/MacOS-Excel-Mappen: über Optionen - Erweitert - 1904 Datumswerte verwenden (Rubrik: Beim Berechnen dieser Arbeitsmappe)

Benutzerdefinierte Zahlenformate

am Beispiel „Liter“:

0,00 "Liter" formatiert die Zahl 15 in die formatierte Darstellung 15,00 Liter.

Oder erweitert gerne auch mit Tausender-Trennzeichen: : #.##0,00 "Liter"

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	Umsätze	Artikel 1	Artikel 2	Artikel 3	Gesamt:	
2	Januar	88,00 Liter	15,00 Liter	87,00 Liter	190,00 Liter	
3	Februar	32,00 Liter	11,00 Liter	32,00 Liter	75,00 Liter	
4	März	87,00 Liter	22,00 Liter	87,00 Liter	196,00 Liter	
5	April	68,00 Liter	68,00 Liter	3.333,00 Liter	3.469,00 Liter	
6	Gesamt:	275,00 Liter	116,00 Liter	3.539,00 Liter	3.930,00 Liter	
7						
8	Proz. Umsatz:	6,				
9						
10	Minimum:					

A dialog box titled 'Unbenannt - Editor' is open, showing the custom number format: #.##0,00 "Liter".

Abb. 2.9: Benutzerdefinierte Zahlenformate

Alternativ kann man dann einfach die Zeichenkette (engl. String - Text in Anführungszeichen) mit gewünschtem Text austauschen (z.B. kWh).

Einmal definierte benutzerdefinierte Zahlenformate merkt sich Excel in der Kategorie „Benutzerdefiniert“ bei den Zahlenformaten.

Format übertragen

Über Kontextmenü und/oder Symbol „Pinsel“ die Formate aus Vorlage-Zelle auf andere Zellen übertragen.

Tipp: Mit Doppelklick behält man den Kopierpinsel bis man ESC drückt!

2.8 Bedingte Formatierung

Mit Excel 2007/2010 hatte man zwei Regeln und Hervorhebungen für bedingte Formatierungen erhalten:

- Datenbalken
- Farbskalen
- Symbolsätze

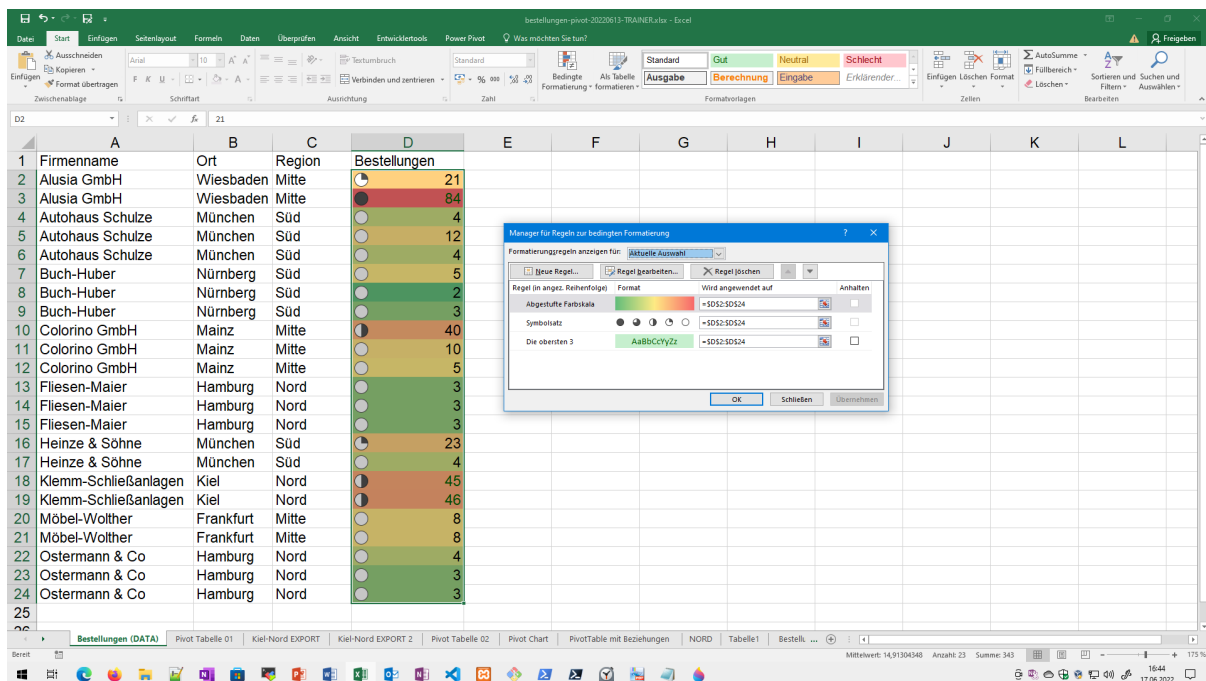


Abb. 2.10: Bedingte Formatierungen

Alle Formate lassen sich kombinieren. Die Regeln löschen und verwalten kann man mittels des Menübands Start - Formatvorlagen - Bedingte Formate...

Hinweis: Alle vordefinierten Regeln (z.B. „Top 10 Werte“ oder „Obere 10%“) lassen sich auf die eigenen Bedürfnisse anpassen. Auch die grafischen Formatierungen können auf die eigenen gewünschten

Intervalle konfiguriert werden.

Und wie immer gilt: Weniger ist Mehr! Soll heißen das obere Beispiel ist nur eine Darstellung für das Mögliche - Sinnvoll ist das Ganze hier nicht mehr.

2.9 Als Tabelle formatieren

Diese mit 2007/2010er eingeführte Technologie, die oft als als einfache „Design-Technik“ zum Einfärben von Tabellenbereichen unterschätzt wird, kann noch so viel mehr.

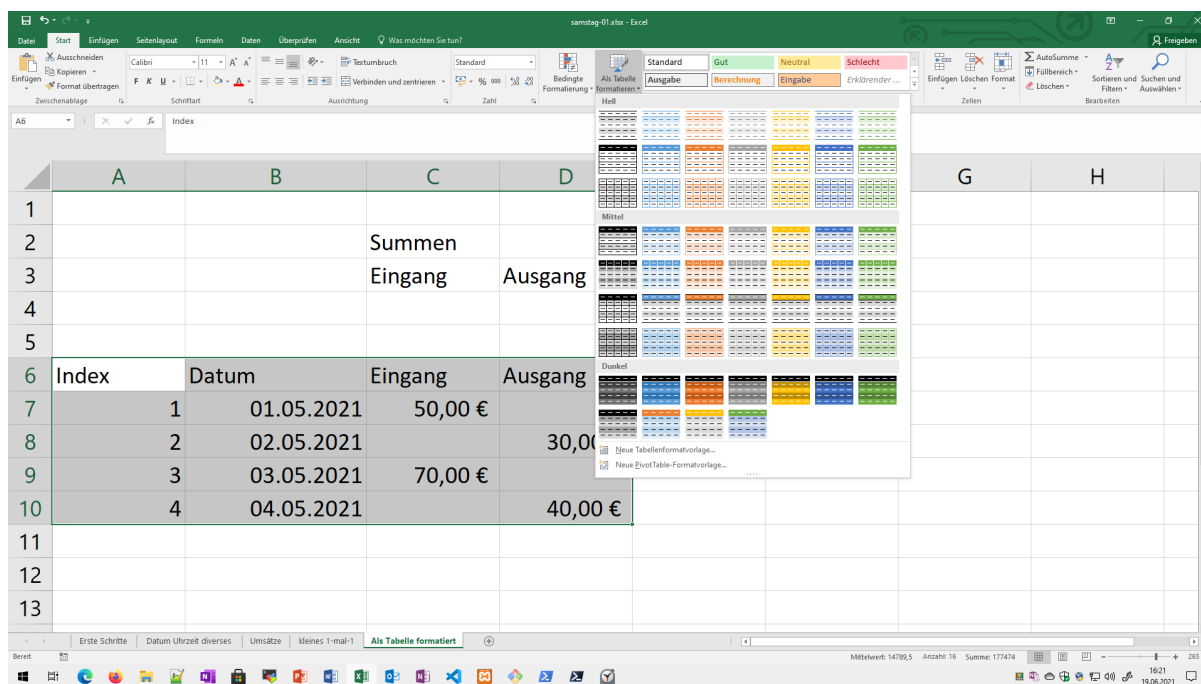


Abb. 2.11: Als Tabelle formatierter Bereich

Unter der Haube wird eine komplett neue intelligente „Datenbank“-Technik mit einfachen und sicheren Möglichkeiten zum Sortieren und Filtern von Informationen bereit gestellt.

Außerdem lassen sich diese Tabellenbereiche sinnvoll benennen und damit „sprechende“ Formeln erzeugen: $=\text{SUMME}(\text{KassenbuchSep}[\text{Eingang}])$

Die folgende Vorgehensweise sollte für die Nutzung herangezogen werden.

Tipps / Best Practise Tabellen

- Tabellen sauber benennen
- Filter-Pfeile ausblenden mittels (z.B.) Ribbon Daten - Filter
- **Indexspalte** nutzen eine Spalte mit Reihenfolge der Eingabe einpflegen (z.B. „Laufende Nr“ oder „Id“), damit sich eine Reihenfolge bewusst erzeugen lassen kann

Die Tabellenbereiche lassen sich auch wieder mittels „In Bereich konvertieren“ in Standard-Tabellenbereiche zurückwandeln.

Von besonderer Wichtigkeit ist eine **Indexspalte**!

Tipp: Eine Spalte mit Reihenfolge der Eingabe einpflegen (z.B. „Laufende Nr.“), damit sich eine Reihenfolge bewusst erzeugen und wiederherstellen lassen kann!

Sortieren und Filtern

Übung:

Nutzen der neuen Tabellenformate (s.a. Menüband Daten) über die entsprechenden Filter- und Sortierschalter.

Anm.: bei Filterungen reagieren die Standardsummen (SUMME) nicht auf die gefilterten Spaltenbereiche. Es werden einfach nur Daten ausgeblendet! Wir brauchen hier später spezielle Summenfunktionen (s. DBSUMME).

Hinweis: Nochmals: nach Änderung mit Sortieren lassen sich Datensätze nicht einfach in *Originalfolgen* bringen, was Datenbanktechnik entspricht und in Ordnung ist, da eine DB diese Info nicht interessiert. Dafür gibt es Indizes.

2.10 Zellbezüge

Zellbezüge (relativ und absolut)

Durch Relative Zellbezüge wird das Kopieren (Ziehen) vieler Formeln einfach und effizient.

Dieses Kopieren gerät aber in Probleme, wenn beteiligte Zellbezüge immer von einer gleichen Zellposition bezogen werden sollen. Hier helfen dann Absolute Zellbezüge weiter.

	A	B	C	D	E
1	Umsätze	Artikel 1	Artikel 2	Artikel 3	
2	Januar	25,00 Liter	1,00 Liter	25,00 Liter	
3	Februar	55,00 Liter	1,00 Liter	33,00 Liter	
4	März	7,00 Liter	7,50 Liter	7,00 Liter	
5	April	88,00 Liter	1,00 Liter	88,00 Liter	
6	Gesamt:	175,00 Liter	10,50 Liter	153,00 Liter	338,50 Liter
7					
8	proz. Umsatz.	=B6/\$E\$6	3,10%	45,20%	
9					
10	Anzahl:	12			
11	Anzahl2:	12			

Abb. 2.12: Zellbezüge - relativ und absolut

Absolute Zellbezüge erhält man durch Hinzufügung von **\$-Symbolen**. So werden Zellbezüge beim Kopieren *fixiert*.

Tipp: Mit Cursor an Zellbezug und Druck auf Funktionstaste F4 kann man alle \$-Kombinationen durchlaufen.

Notebook-User beachten bitte die Belegung der Funktionstasten an ihren Geräten! Oft muss noch eine zusätzlich Fn Taste gedrückt werden.

Übung: gemischte relative/absolute Zellbezüge bei „kleinem 1-Mal-1“

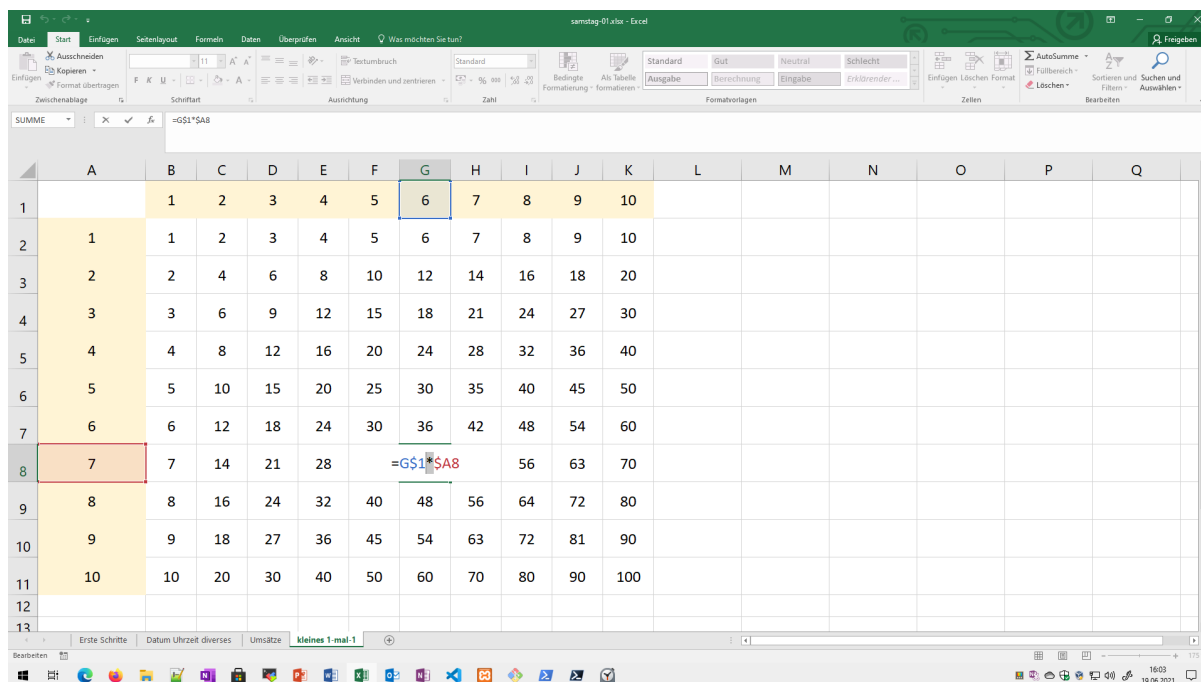


Abb. 2.13: Gemischte relative und absolute Bezüge

Beispielhafte erste Formel (Zelle B2) zum Kopieren/Vervielfältigen: =B\$1 * \$A2

2.11 Große Tabellen

Wir pflegen in unseren Excel-Seminaren seit vielen Jahren das Thema **Große Tabellen**.

Allerdings könnte man das Thema auch wie folgt beschreiben:

„Wie gehe ich effizient mit den Excel-Tabellen um und welche Helferlein stellt mir Excel beiseite?!“

Im Einzelnen werden wir die folgenden Techniken kennen lernen:

- In Tabellenblatt navigieren
- In Arbeitsbereich zoomen
- Tabellenfenster teilen
- Zeilen und/oder Spalten fixieren
- Ausdruck optimieren

Mit verschiedenen Übungen wollen wir uns die einzelnen Exceltechniken erarbeiten.

In Tabellenblatt navigieren

Hier seien die Tastenkombinationen zum effizienten bewegen und markieren in Erinnerung gebracht:

- Strg + Cursor - an Tabellengrenzen springen
- Umschalten + Cursor - Markieren

In Arbeitsbereich zoomen

Über die Oberflächen-Zoom- und Ansichtstools den Ausschnitt für die Arbeit optimieren. Insbesondere:

- mit Strg + Maus-Rollrad manuell zoomen
- nach Markieren eines Bereichs: Menüband Ansicht - Gruppe Zoom - Auswahl vergrößern

Tabellenfenster teilen

Mittels Menüband Ansicht - Gruppe Fenster - Teilen lässt sich ein Arbeitsblatt mit geteilten Fensterbereichen nach eigenen Wünschen mit Inhalten darstellen.

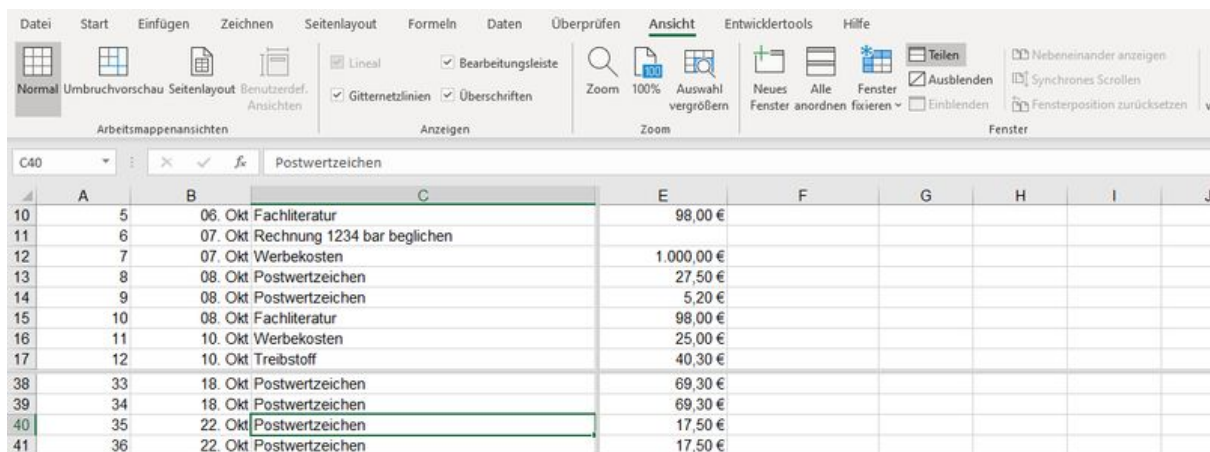


Abb. 2.14: Fenstern Teilen

Zeilen und/oder Spalten fixieren

Bei langen Tabellen kann man hier die gewünschten Kopfbereiche „festhalten / fixieren“, damit man beim Scrollen immer die Tabellenkopfbeschriftungen erhält.

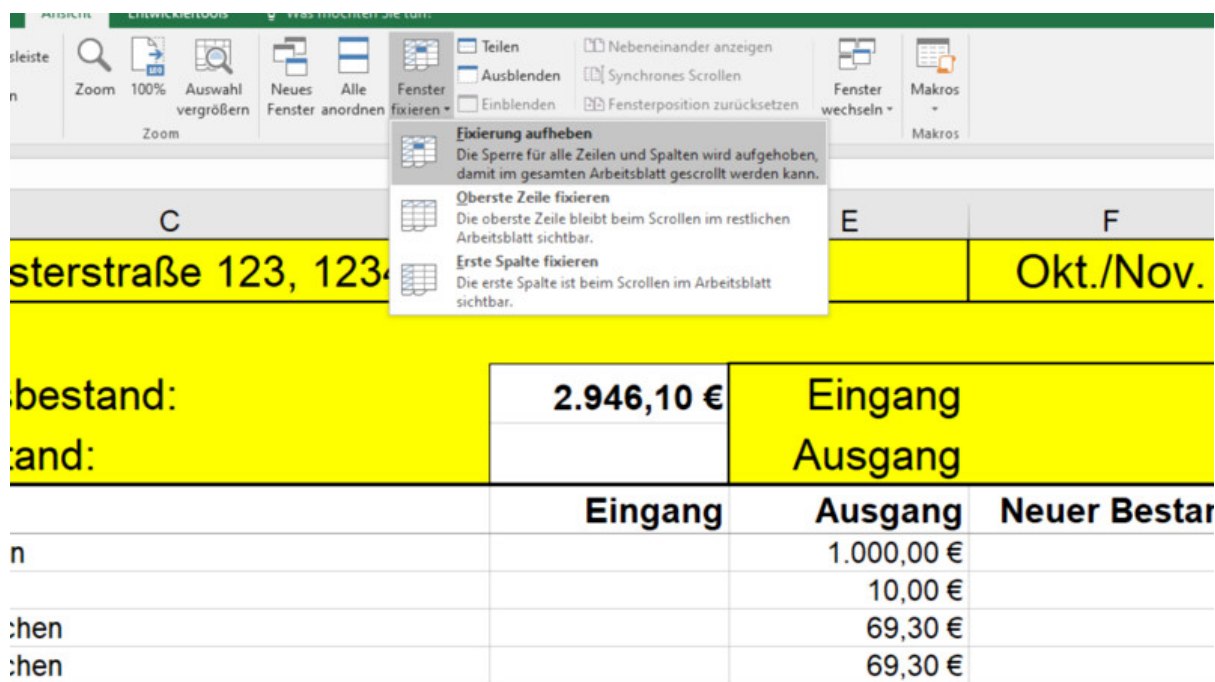


Abb. 2.15: Fenster fixieren

Einstellung mittels Menüband Ansicht - Gruppe Fenster - Fenster fixieren. Für die Standardwahl „Fenster fixieren“ bitte vorher Zelle markieren an der links/oberhalb fixiert wird.

Tipp: Die „als Tabelle definierten“ Bereiche zeigen dieses Verhalten automatisch und tragen ihre Spaltenköpfe in die Standardspalten des Tabellenblatts ein.

Ausdruck optimieren

Bei größeren Tabellenblättern sollte man sich die Ausdrucksgestaltung über die Druckoptionen von Excel genauer anschauen.

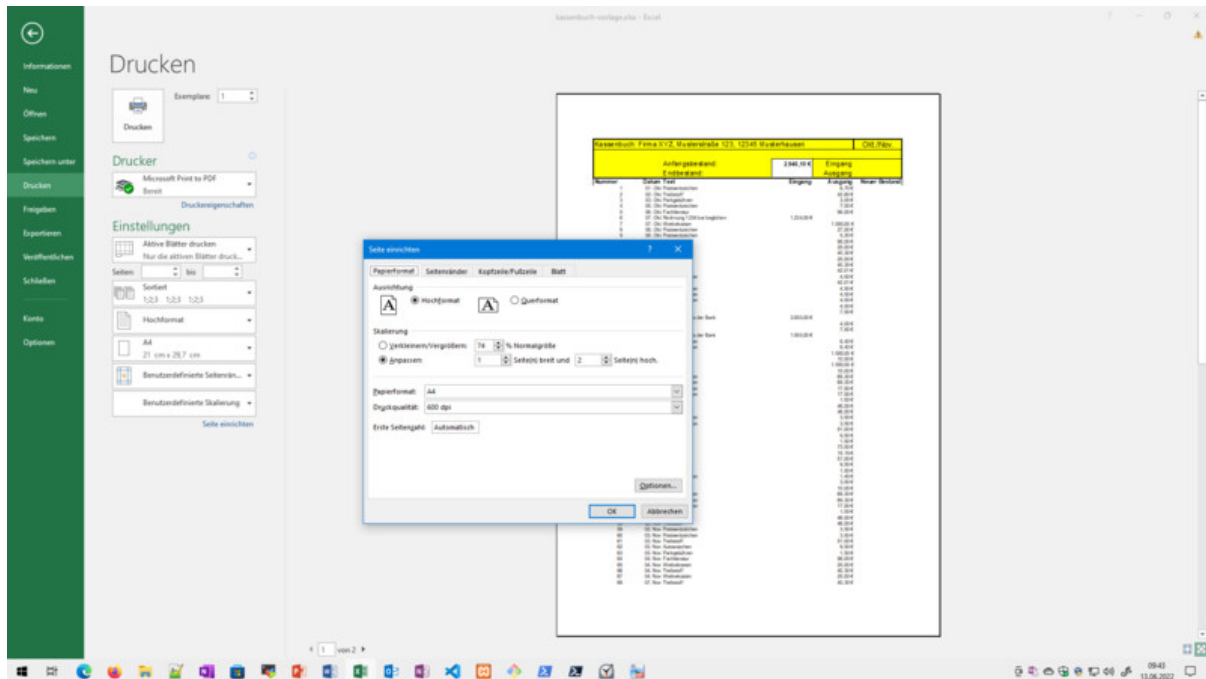


Abb. 2.16: Ausdruck optimieren

Bei der Skalierung sollte man - falls Ausdrücke gewünscht sind - auf die resultierenden Schriftgrößen achten. Excel skaliert *gnadenlos* proportional, sodass eine Schriftgröße 10 bei Skalierung 45% eben nur noch Schriftgröße 4,5 hat!

2.12 Kommentar / Notizen

Im Excel 365 und den Vorgängern mit mehr und mehr Cloud-Unterstützung (siehe OneDrive, Azure Cloud) bekamen die klassischen Notizen moderne Kommentar-Technik.

So lassen sich ganze *Gesprächsfäden (Threads)* mit KollegInnen in einer Arbeitsmappe unterbringen.

Bei der Nutzung in einer Zelle muss man sich für eine der beiden Varianten entscheiden!

Mittels **Rechter Maus (Kontextmenü) - Kommentar / Notiz** lassen sich **Kommentare oder Notizen** einfügen und später auch bearbeiten bzw. löschen, ein-/ausblenden.

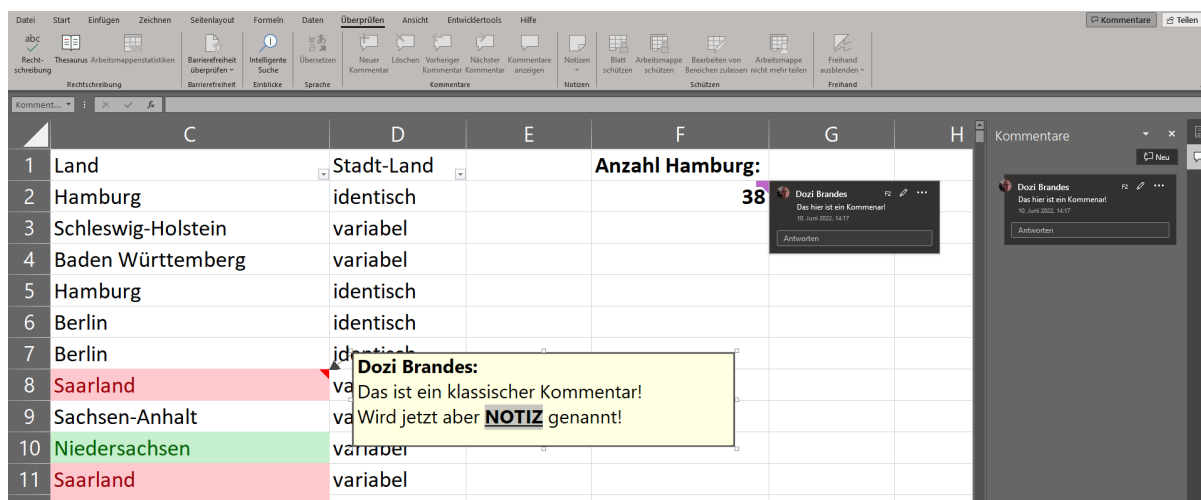


Abb. 2.17: Kommentare und Notizen bei aktuellem Excel

Kommentare in Menübändern: Ribbon Überprüfen - Gruppe Kommentare / Notizen

In Druckoptionen lassen sich die Ausdrucksgestaltungen für die Kommentare (bzw. Notizen) einstellen.

2.13 Diagramme 101

Hier soll es erst einmal um einfache Standarddiagramme und deren Erstellung gehen. Und am schnellsten geht das über eine *One-Click* Erzeugung eines eigenständigen Diagrammblatts.

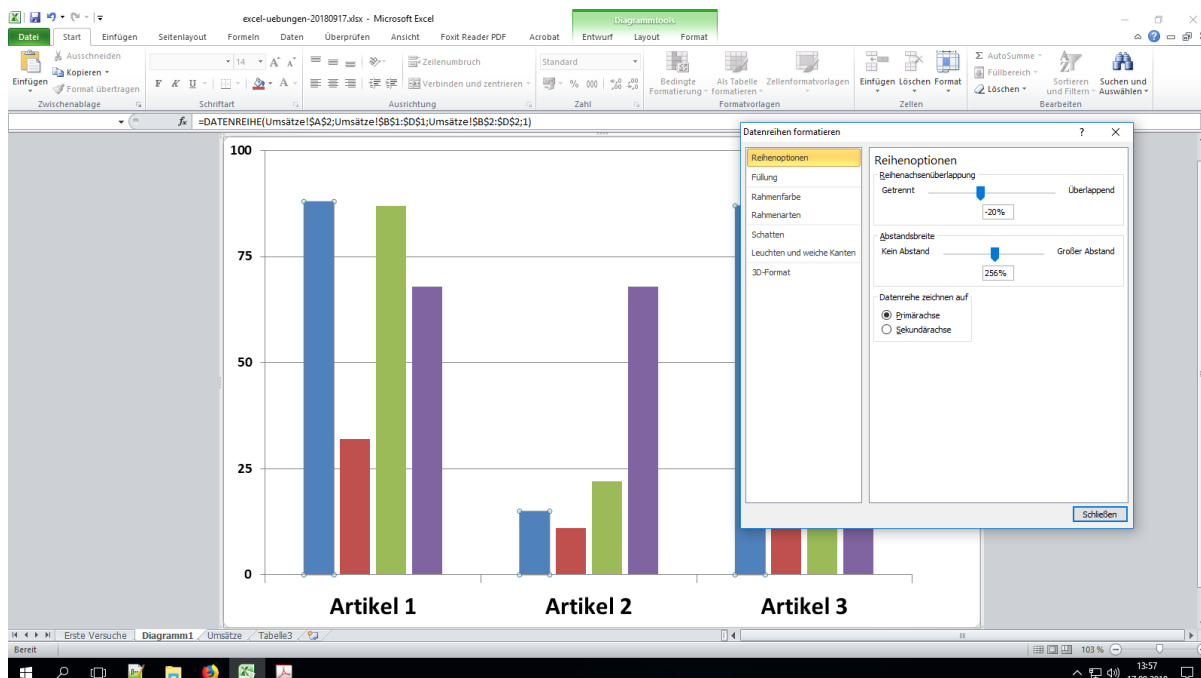


Abb. 2.18: Schnelldiagramm mittels F11

Am Einfachsten markiert man also die gewünschten Datenreihen inklusive der beschreibenden Spalten- und Zeilenköpfe, die man per Diagramm auswerten möchte und drückt F11.

Tipp: Schnelldiagramm mittels Funktionstaste F11!

Diagramm-Assistenten

Mittels Menübandaktionen über Menüband Einfügen - Diagramme kann man für markierten Bereiche das gewünschte Diagramm vorauswählen und vorkonfigurieren.

Übungen

In den Seminare werden Übungen zu diversen einfachen Diagrammen und die erste Nutzung der bereitgestellten Menübänder für Diagramme durchgeführt.

KAPITEL 3

Excel Funktionen

Hier wollen wir uns auf die in Excel integrierten Funktionen konzentrieren. Und an dieser Stelle ist tatsächlich manchmal die genutzte Excel-Version ein Unterscheidungsmerkmal, das Microsoft über die letzten Jahre immer wieder mal *neue* Funktionen in einzelnen Versionen präsentiert hat. Insbesondere die Excel 365 Entwicklungen stehen bei Microsoft im Fokus.

Excel-Funktionen (nach Kategorie)

Excel für Microsoft 365, Excel für Microsoft 365 für Mac, Excel für das Web, Excel 2021, Excel 2021 für Mac, Excel 2019, Excel 2019 für Mac, Excel 2016, Excel 2016 für Mac, Excel 2013, Excel 2010, Excel 2007, Excel für Mac 2011, Excel Starter 2010, [Weniger](#)

Arbeitsblattfunktionen werden nach ihrer Funktion kategorisiert. Klicken Sie auf eine Kategorie, um ihre Funktionen zu durchsuchen. Oder drücken Sie STRG+F, um eine Funktion zu suchen, indem Sie die ersten Buchstaben oder einen beschreibenden Begriff eingeben. Detaillierte Informationen zu einer Funktion erhalten Sie, indem Sie auf ihren Namen in der ersten Spalte klicken.

>
Unsere zehn am häufigsten verwendeten Funktionen
>
Kompatibilitätsfunktionen
>
Cubefunktionen
>
Datenbankfunktionen
>
Datums- und Uhrzeitfunktionen
>
Technische Funktionen
>
Finanzmathematische Funktionen
>
Informationsfunktionen
>
Logische Funktionen
>
Nachschlage- und Verweisfunktionen
>
Mathematische und trigonometrische Funktionen
>
Statistische Funktionen
>
Textfunktionen
>
Benutzerdefinierte Funktionen, die mit Add-Ins installiert werden
>
Webfunktionen
>

Versionsmarkierungen geben die Version von Excel an, in der eine Funktion eingeführt wurde. Diese Funktionen sind in früheren Versionen nicht verfügbar.

Abb. 3.1: Excel Funktionen (Microsoft Website); Quelle: microsoft.com

Microsoft Online Portal zu Excel Funktionen: [Link Excel Funktionen nach Kategorien](#)⁵

Insbesondere falls man Online Anleitungen und Umsetzungen mit Excel-Funktionen gefunden hat, sollte man schnell mal recherchieren, ob diese Funktionen auch bei der eigenen Excel-Technik verfügbar sind.

Microsoft | Support Microsoft 365 Office Windows Surface Xbox Sonderangebote [Microsoft 365 kaufen](#)

Microsoft 365-Support Produkte ▾ Geräte ▾ Neuerungen Microsoft 365 installieren Konto und Abrechnung ▾ Vorlagen

FILTER-Funktion

Excel für Microsoft 365, Excel für Microsoft 365 für Mac, Excel für das Web, Excel 2021, Excel 2021 für Mac, Excel 2019, Excel für iPad, Excel für iPhone, Excel für Android-Tablets, Excel für Android-Smartphones, [Weniger](#)

Mithilfe der **FILTER**-Funktion können Sie einen Bereich von Daten anhand der von Ihnen definierten Kriterien filtern.

Abb. 3.2: Funktion FILTER() auf Microsoft Infoportal

⁵ <https://support.microsoft.com/de-de/office/excel-funktionen-nach-kategorie-5f91f4e9-7b42-46d2-9bd1-63f26a86c0eb>

Anm.: zum aktuellen Zeitpunkt (Anfang 2023) soll ich bei meiner Uni immer noch die Seminare mit Schwerpunkt **Office 2016** durchführen. Attraktive Übungen und Umsetzungen mit der Funktion **FILTER()** fallen da natürlich heraus.

Hinweis: Die Hilfe zu Funktionen kann man schnell über den Funktionsassistenten (fx-Symbol in Bearbeitungszeile) erreichen. Hierüber ist der Aufruf von integrierter oder Online-Hilfe möglich.

3.1 Standardfunktionen

Die hier folgenden Funktionen finden sich auch bei den am *häufigsten* genutzten **Excel Funktionen** in den Übersichtsportalen bei Microsoft.

Im folgenden ein Beispiel mit diversen Standardfunktionen bei einer kleinen Auswertung von Umsätzen.

	A	B	C	D	E	F	G	H	I
1	Umsätze	Artikel 1	Artikel 2	Artikel 3	Artikel 4	Gesamt:			
2	Januar	75,00 Liter	15,00 Liter	10,00 Liter	12,00 Liter	112,00 Liter			
3	Februar	55,00 Liter	33,00 Liter	55,00 Liter	30,00 Liter	173,00 Liter			
4	März	25,00 Liter	25,00 Liter	87,00 Liter	25,00 Liter	162,00 Liter			
5	Gesamt:	155,00 Liter	73,00 Liter	152,00 Liter	67,00 Liter	447,00 Liter			
6									
7	Proz. Umsatz:	34,676%	16,331%	34,004%	14,989%				
8									
9	Min:	10,00 Liter		=MIN(B2:E4)					
10	Max:	87,00 Liter		=MAX(B2:E4)					
11	Durchschnitt:	37,25 Liter		=MITTELWERT(B2:E4)					
12	Anzahl:	12		=ANZAHL(B2:E4)					
13	Anzahl2:	12		=ANZAHL2(B2:E4)		11.06.2022	=HEUTE()		
14									
15									

Abb. 3.3: Diverse Standardfunktionen (Funktionen 101)

In Kurzseminaren lassen sich auf jeden Fall Übungen und Beispiele mit den folgenden Excel-Funktionen durchführen:

- **SUMME()** - [MS Office Support Link SUMME\(\)](#)⁶
- **ANZAHL()** - [MS Office Support Link Funktion ANZAHL\(\)](#)⁷
- **ANZAHL2()** - [MS Office Support Link Funktion ANZAHL2\(\)](#)⁸
- **MIN()** - [MS Office Support Link MIN\(\)](#)⁹
- **MAX()** - [MS Office Support Link MAX\(\)](#)¹⁰

⁶ <https://support.microsoft.com/de-de/office/summe-funktion-043e1c7d-7726-4e80-8f32-07b23e057f89>

⁷ <https://support.microsoft.com/de-de/office/anzahl-funktion-a59cd7fc-b623-4d93-87a4-d23bf411294c>

⁸ <https://support.microsoft.com/de-de/office/anzahl2-funktion-7dc98875-d5c1-46f1-9a82-53f3219e2509>

⁹ <https://support.microsoft.com/de-de/office/min-funktion-61635d12-920f-4ce2-a70f-96f202dcc152>

¹⁰ <https://support.microsoft.com/de-de/office/max-funktion-e0012414-9ac8-4b34-9a47-73e662c08098>

- MITTELWERT() - MS Office Support Link MITTELWERT()¹¹
- HEUTE() - MS Office Support Link HEUTE()¹²

Diese Liste hat natürlich keinen Anspruch auf Vollständigkeit und wird oft in den praktischen Seminarübungen durch weitere Funktionen ergänzt.

Zum Abschluss dieser ersten Übersicht hier noch ein Beispiel für eine *versteckte* Funktionen in Excel.

DATEDIF()

Die Funktion DATEDIF() berechnet die Anzahl der Tage, Monate oder Jahre zwischen zwei Datumsangaben.

Das ist eigentlich erst einmal eine sehr einfache Problemstellung, da Datumzellen (als spezielle Zahlenzellen) sehr einfach für Differenzen genutzt werden können. Allerdings handelt es sich hier immer um die Differenz in Tagen. Oft möchte man *kalendarische* Infos erhalten!

Diese Funktion befindet sich nicht in der Excel-Hilfe oder im Portfolio des Excel Funktionsassistenten - aber sie funktioniert und ist online dokumentiert: [MS Office Support Link DATEDIF\(\)](#)¹³

Funktionsnutzung: Berechnung von Datumzellendifferenzen in

- Jahren (Schalter „y“),
- Monaten („m“),
- Tagen („d“)
- oder Kombinationen

Tipp: Suchmaschine nutzen mit Begriffen wie „Excel Jahre Differenz“

Beispiel-Ergebnis-Link: <http://0711office.de/excel/funktionen/datedif.htm>

3.2 SUMME()

Hier eine kurze Darstellung der Funktion **SUMME()** als Beispiel für die Integration in Formeln.

Online Infos zu **SUMME()**: [MS Office Support Link SUMME\(\)](#)¹⁴

¹¹ <https://support.microsoft.com/de-de/office/mittelwert-funktion-047bac88-d466-426c-a32b-8f33eb960cf6>

¹² <https://support.microsoft.com/de-de/office/heute-funktion-5eb3078d-a82c-4736-8930-2f51a028fdd9>

¹³ <https://support.microsoft.com/de-de/office/datedif-25dba1a4-2812-480b-84dd-8b32a451b35c>

¹⁴ <https://support.microsoft.com/de-de/office/summe-funktion-043e1c7d-7726-4e80-8f32-07b23e057f89>

Index	Datum	Werte	Eingänge	Ausgänge	Bemerkungen
1	02.05.2022	Postwertzeichen		5,50 €	
2	03.05.2022	Pfand	15,00 €		
3	04.05.2022	Getränke	10,00 €	25,00 €	
4	04.05.2022	Postwertzeichen		25,00 €	ausgelegt - JoeB
5	06.05.2022	Pfand	5,00 €		

Abb. 3.4: Excel Funktion Summe

Man kann sehr gut den Einsatz der benannten Tabellenbereiche erkennen:

- Tabelle: KassenbuchMai
- Spalte: Ausgänge

Wir erhalten also *sprechende Formeln/Funktionen* durch die Kombination mit **Als Tabelle formatierter Bereich**.

3.3 WENN()

Bei der Funktion **WENN()** handelt es sich um eine Logische Funktion mit drei Parametern:

- Prüfung/Bedingung - ergibt entweder **WAHR** oder **FALSCH**
- Dann-Wert (falls Prüfung **WAHR**)
- Sonst-Wert (falls Prüfung **FALSCH**)

Bitte bei der Eingabe die Syntax und Hilfestellung beachten (Gelbe Quicktips an Funktion).

Microsoft Online Info zu Funktion **WENN()**: [MS Office Support Link WENN\(\)](#)¹⁵

Die **WENN()**-Funktion lässt sich auch verschachteln:

¹⁵ <https://support.microsoft.com/de-de/office/wenn-funktion-69aed7c9-4e8a-4755-a9bc-aa8bbff73be2>

Abb. 3.5: WENN-Funktionen verschachtelt

Anm.: Verschachtelung von zwei WENN-Funktionen für drei Verzweigungen und so weiter bis maximal 9 Verschachtelungen (mit maximal 10 möglichen Zuweisungen).

Die Nutzung von Zellen für Vergleichswerte in Prüfungen und für Berechnungen in den Dann/Sonst-Zweigen verlangen wieder nach den absoluten Zellbezügen!

Somit lassen sich die Funktionen dann auch wieder sauber kopieren (nach unten ziehen).

3.4 SVERWEIS()

Verschachtelte WENN-Funktionen können nur maximal 10 Werte verzweigen und das würde ja auch 9 Verschachtelungen verlangen, die man kaum noch durchschauen könnte.

Daher bedienen wir uns bei solch vielfältigen Nachschlageversuchen der Funktion **SVERWEIS** (oder auch **WVERWEIS**).

Microsoft Online Portal zu **SVERWEIS**: [MS Office Support Link SVERWEIS\(\)](#)¹⁶

Abb. 3.6: SVERWEIS-Funktion

Mit SVERWEIS lassen sich aus Bereichen (Matrix) also die entsprechenden Ergebniswerte aus den benachbarten (2. / 3. / ...) Spalten auswerten:

=SVERWEIS(B4; \$A\$2:\$E\$6 ; 2)

=SVERWEIS(B4 ; Provisionstabellenblatt!\$A\$2:\$E\$6 ; 2)

¹⁶ <https://support.microsoft.com/de-de/office/sverweis-funktion-0bbc8083-26fe-4963-8ab8-93a18ad188a1>

Oder mit (als Tabellen formatierten) Bereichen: (Technik: Als Tabelle formatieren...)

=SVERWEIS(B4 ; Provisionstabelle[#Alle] ; 2)

Beachten:

- erste Spalte aufwärts alphanumerisch sortieren
- Einstiegswert (kleinster möglicher Wert) sollte in der ersten Zeile berücksichtigt werden.

Verwandte Funktion: **WVERWEIS()** - dort das Ganze dann in Zeilen, statt in Spalten bei SVERWEIS().

3.5 TEILERGEBNIS()

An anderer Stelle hatten wir bei Filterungen von dynamischen Tabellenbereiche kein Glück bei der Aktualisierung von Summen.

Mit der mathematischen Funktion **TEILERGEBNIS** lassen sich dynamische Tabellenbereiche auswerten.

Microsoft Online Portal Link zur Funktion **TEILERGEBNIS()**: [MS Office Support Link TEILERGEBNIS\(\)](#)¹⁷

Index	Firmenname	Ort	Region	Bestellungen	Bestelldat
1	Akusa GmbH	Wiesbaden	Mitte	21	01.04.2020
2	Akusa GmbH	Mainz	Mitte	64	02.04.2020
9	Colorino GmbH	Mainz	Mitte	40	09.04.2020
10	Colorino GmbH	Mainz	Mitte	10	10.04.2020
11	Colorino GmbH	Mainz	Mitte	5	11.04.2020
19	Möbel-Wolther	Frankfurt	Mitte	8	19.04.2020
20	Möbel-Wolther	Frankfurt	Mitte	8	20.04.2020

Abb. 3.7: TEILERGEBNIS-Funktion

Die Berechnung folgt dynamisch und aktuell den Filterungen im Tabellenbereich.

=TEILERGEBNIS(9 ; BestellungenSep[Bestellungen])

Hier definiert man durch die **9** die Funktionalität „**Summe erstellen**“!

Alternative: die 1 würde den Mittelwert erstellen - alles Weitere über die Hilfen zur Funktion oder die gelben Quicktipps bei der Eingabe.

¹⁷ <https://support.microsoft.com/de-de/office/teilergebnis-funktion-7b027003-f060-4ade-9040-e478765b9939>

3.6 Datenbankfunktionen

Erstes Beispiel mit **DBSUMME()** behandelt Tabellenbereiche wie Datenbanken und erlaubt Abfragen mit Kriterien (engl.: Queries).

Microsoft Online Portal zur Funktion **DBSUMME()**: [MS Office Support Link DBSUMME\(\)](#)¹⁸



Index	Firmenname	Ort	Region	Bestellungen	Bestelldati
1	Akusia GmbH	Wiesbaden	Mitte	21	01.04.2020
2	Akusia GmbH	Mainz	Mitte	84	02.04.2020
3	Adelhaus Schätze	München	Süd	4	03.04.2020

Abb. 3.8: DBSUMME-Funktion

Übungen: (Kriterien bei den Suchkriterienzellen)

- **ODER** (Kriterien auf unterschiedlichen Zeilen)
- **UND** (Kriterien befinden sich auf derselben Zeile)

Hinweis: Hilfe-Kategorie Datenbank-Funktionen oder wieder einfach über Funktionsassistenten die Hilfe aufrufen.

Die **DBSUMME()** stellt eine beispielhafte Umsetzung der DB-Technik dar.

Weitere DB-Funktionen:

- **DBSUMME()**
- **DBMAX()**
- **DBMIN()**
- **DBMITTELWERT()**,
- ...

Die Funktionsweisen und Nutzungen der Datenbank-Funktionen sind immer gleich.

3.7 KGRÖSSTE (engl. LARGE)

Weil diese Funktion gerade in einem anderen Seminar zum Einsatz kam, will ich sie hier kurz vorstellen. Außerdem wurde die Funktion auf der Infosite mit englischem Namen dargestellt, sodass es auch von Interesse ist, die **Übersetzungen** (ins Deutsche) parat zu haben.

Funktionsweise von KGRÖSSTE: Ermittle (immer) die drei größten Werte einer Spalte!

¹⁸ <https://support.microsoft.com/de-de/office/dbsumme-funktion-53181285-0c4b-4f5a-aaa3-529a322be41b>

The screenshot shows an Excel spreadsheet with the following data:

Country	Actual	Budget	3 Largest Sales
USA	15.300	19.400	15300
Canada	11.500	11.100	
Germany	12.380	13.455	12380
China	11.154	11.400	
Brazil	9.800	9.600	
India	11.640	9.000	11640
Australa	9.600	9.800	

The presentation slide contains the following text:

1. Select cell D3 and add the title "3 Largest Sales".
2. Select cell D4 and enter the following formula:


```
=IF(B4>=LARGE($B$4:$B$10;3);B4;\"")
```

This formula answers two questions:

1. (using LARGE) - Is the value of B4 greater than or equal to the 3rd largest value in range B4:B10?
2. (using IF) - If so, display the value in cell B4; otherwise, display nothing (two double-quotes).

Fill the formula down from cell D4 through cell D10 and observe the results.

The formula editor window shows the German equivalent: `=WENN(B4>=KGRÖSSTE(B4:B10;3);B4;\"")`

Abb. 3.9: KGRÖSSTE() - die 3 größten Werte berechnen

Für die Übersetzung des Originals

`=IF(B4>=LARGE(B4:B10;3);B4;\"")`

kann man neben einer allgemeinen Internetrecherche auch zum **Excel-Translator** (Excel Translator Website¹⁹) greifen.

The screenshot shows the homepage of the Excel-Translator website. The main navigation menu includes:

- Übersetzer
- Funktionen
- Argumente
- Fehler
- Netzwerk
- Blog
- Alle
- Finanzmathematik
- Datum & Zeit
- Math. & trigonomet.
- Statistik
- Matrix
- Datenbank
- Text

There is also a search bar and a 'Werbung' (Advertising) section.

Abb. 3.10: Excel Translator Website

Und schon erhält man mit dem Übersetzer die *deutsche Formel*:

`=WENN(B4>=KGRÖSSTE(B4:B10;3);B4;\"")`

Und somit immer die aktuell drei größten Werte in einer Spalte.

Beim Nachschlagen der Support-Infos bei Microsoft zur Funktion **KGRÖSSTE()** wird die Verfügbarkeit seit **Version 2007** dokumentiert.

¹⁹ <https://de.excel-translator.de/>

3.8 SORTIEREN(), EINDEUTIG()

Leider sind nicht immer alle Funktionen, die man für Problemlösungen mit Excel recherchiert solche Klassiker. Insbesondere sehr attraktive *neue* Funktionen wie **SORTIEREN()** [engl. SORT()] oder **EINDEUTIG()** [engl.:UNIQUE()] findet man leider nur bei den aktuellen **Excel 2021 und 365!**

Die folgende Darstellung zeigt eine Kombination dieser neueren Funktionen in Kombination mit **Drop-Down-Listen** mittels **Datenüberprüfung**.

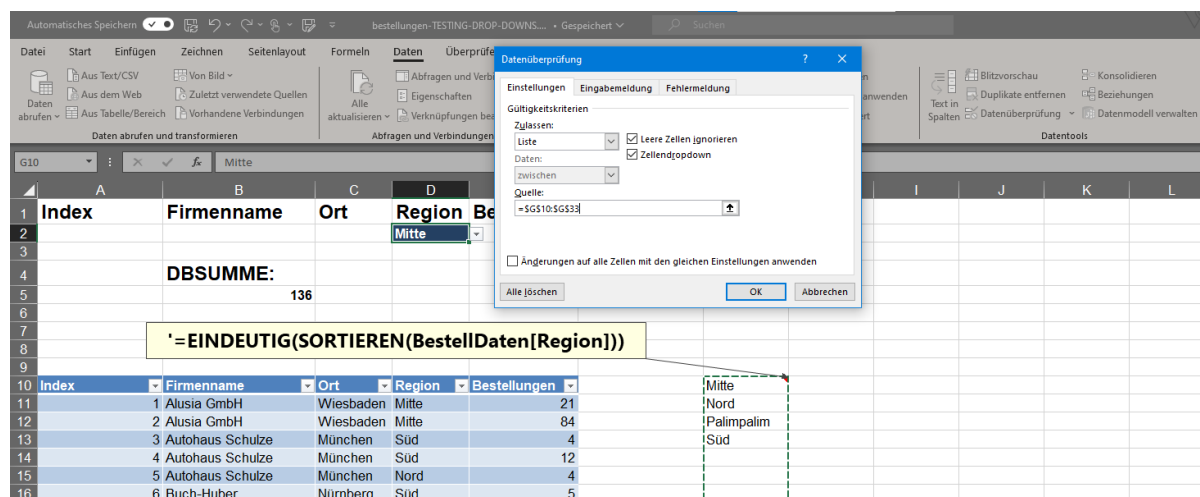


Abb. 3.11: Funktionen EINDEUTIG und SORTIEREN mit Datenüberprüfung für Liste

Hier eine Infoliste zu den beteiligten Techniken - die Funktionen stehen **erst mit Excel 2021 und 365** zur Verfügung!

- Funktion EINDEUTIG()

Link: <https://support.microsoft.com/de-de/office/eindeutig-funktion-c5ab87fd-30a3-4ce9-9d1a-40204fb85e1e>

- Funktion SORTIEREN()

Link: <https://support.microsoft.com/de-de/office/sortieren-funktion-22f63bd0-ccc8-492f-953d-c20e8e44b86c>

- Drop-Down-Listen mit Datenüberprüfung

Link: <https://support.microsoft.com/de-de/office/erstellen-von-dropdownlisten-7693307a-59ef-400a-b769-c5402>

Eine weitere spannende moderne Funktion ist **FILTER()**, die ich später noch kommentiere. Vorher wollen wir versuchen die Funktionen als *Nachbau* zu erhalten.

Fehlende Funktionen nachbauen

Eine seminarinterne Recherche hat folgende Quelle mit Nachbauten fehlender Funktionen in *Vorversionen Excel* ergeben Anm.: noch nicht von mir gecheckt/getestet - ich habe ja Zugriff auf 365 ;-)

Gefahr: Einsatz und Testing auf eigene Rechnung!

Nachbau mittels VBA-Code der Funktionen

- SORTIEREN
- SORTIERENNACH

- FILTER
- XVERWEIS
- XVERGLEICH
- EINDEUTIG
- SEQUENZ
- ZUFALLSMATRIX

Link: <https://hermann-baum.de/excel/hbSort/de/>

3.9 FILTER() - Excel 365

Die hier dargestellte Funktion FILTER() ist nicht nur technisch sehr interessant, sondern zeigt auch eine Abhängigkeit von der Excel-Version.

SPALTEN	Gibt die Anzahl der innerhalb eines Bezugs aufgeführten Spalten zurück
DROP-Funktion Office 365	Schließt eine angegebene Anzahl von Zeilen oder Spalten vom Anfang oder Ende eines Arrays aus.
EXPAND-Funktion Office 365	Erweitert oder polstert ein Array auf die angegebenen Zeilen- und Spaltenabmessungen.
FILTER-Funktion Office 365	Mithilfe der FILTER-Funktion können Sie einen Bereich von Daten anhand der von Ihnen definierten Kriterien filtern.
FORMELTEXT 2013	Gibt die Formel am angegebenen Bezug als Text zurück
PIVOTDATENZUORDNEN	Gibt die in einem PivotTable-Bericht gespeicherten Daten zurück
WVERWEIS	Sucht in der obersten Zeile einer Matrix und gibt den Wert der angegebenen Zelle zurück
HSTACK-Funktion Office 365	Fügt Arrays horizontal und nacheinander an, um ein größeres Array zurückzugeben.
HYPERLINK	Erstellt eine Verknüpfung oder einen Sprung, über die ein auf einem Netzwerkservers, in einem Intranet oder im Internet gespeichertes Dokument geöffnet wird
INDEX	Verwendet einen Index, um einen Wert aus einem Bezug oder einem Array auszuwählen

Abb. 3.12: Funktionsübersicht mit Funktion FILTER()

Link: <https://support.microsoft.com/de-de/office/filter-funktion-f4f7cb66-82eb-4767-8f7c-4877ad80c759>

Die Excel-Online-Support Darstellungen zeigen auch die Kompatibilität der verschiedenen Funktionen bezüglich der Excel-Versionen.

- **Office 365:** benötigt die Office 365 Suite von Microsoft mit Excel 365 und können häufig auch über die **WebApps** oder auch Smartphone Apps verfügbar sein!
- Ohne Zusatzinfo: klassische Excel-Techniken - ggf. zurück bis Excel-Version 95
- 2013: ab Excel 2013

Eine komplette Umsetzung mit Techniken aus dem Office 365 - also Excel 365 - folgt mit einem abgewandeltem Beispiel aus der Microsoft Support Seite zur Funktion FILTER().

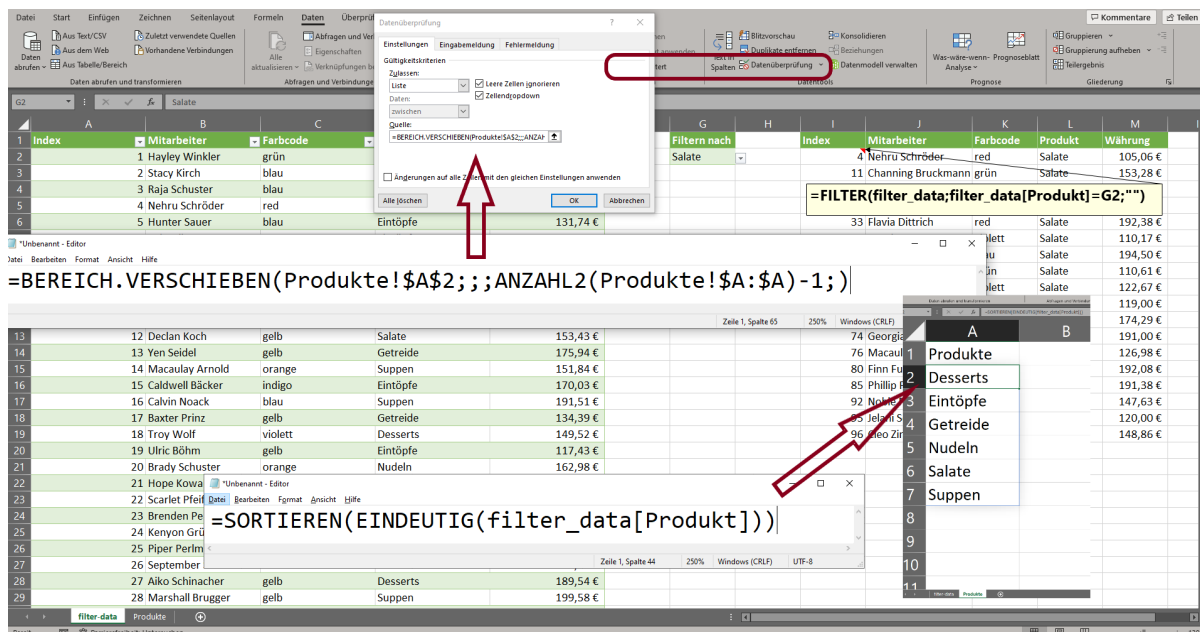


Abb. 3.13: Excel 365 Funktionen im Einsatz bei Filter-Technik

Im Gegensatz zu den klassischen Matrix-Funktionen arbeiten die moderneren Funktionsvarianten mit **Dynamischen Arrays** und sogenannten **Überläufen (engl. spills)**.

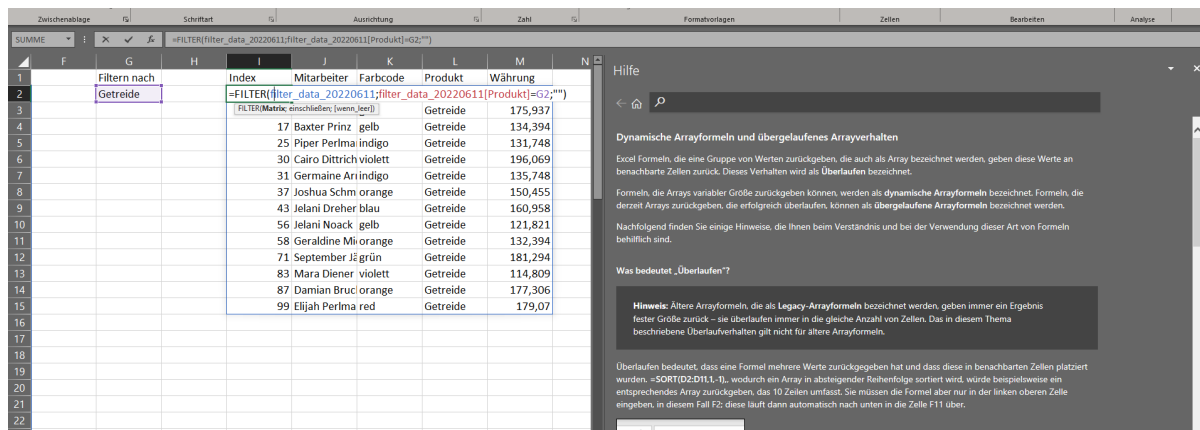


Abb. 3.14: Dynamische Arrayfunktion FILTER()

Insbesondere bei der Nutzung von *Als Tabelle formatierten Bereichen...* ergeben sich neue Herausforderungen, da es in diesen Bereichen zu **#Spill** Fehlern kommen kann (siehe [Microsoft Support](https://support.microsoft.com/de-de/office/-%C3%BCberlauf-fehler-%C3%BCberschreitet-den-rand-des-arbeitsblatts-b3406d00-be0b-4a68) zu **Überlauf/Spill**²⁰).

²⁰ <https://support.microsoft.com/de-de/office/-%C3%BCberlauf-fehler-%C3%BCberschreitet-den-rand-des-arbeitsblatts-b3406d00-be0b-4a68>

Pivot-Techniken (PivotTable oder auch PivotChart) werden klassisch auch als **Kreuztabellenabfragen** bezeichnet.

The screenshot shows an Excel spreadsheet with a PivotTable and the PivotTable Fields task pane. The PivotTable summarizes orders by region (Nord, Mitte, Süd) and company (Firmen). The task pane shows the data source and field assignments.

Summe von Bestellung	Regionen			
Firmen	Nord	Mitte	Süd	Gesamt
Ostermann & Co		10		
Möbel-Wolther			16	
Klemm-Schließanlagen		505		
Heinze & Söhne				27
Fliesen-Maier		9		
Colorino GmbH			55	
Buch-Huber				10
Brandes & Daughter		111		
Autohaus Schulze				20
Alusia GmbH			105	
Gesamtergebnis	635	176	105	57

Abb. 4.1: Pivot Tabelle (PivotTable)

Eine Aufbereitung und Analyse von Daten ist mit Hilfe der Pivot-Assistenten möglich.

Hinweis: Siehe auch Zusammenhang mit DB-Funktionen (DBSUMME, DBANZAHL) und auch hier wieder Einsatz von „Als Tabelle formatieren...“

Schauen wir uns die unterschiedlichen Schritte von der Erstellung bis zu weitergehenden Auswertungen mit Pivot-Technik an.

4.1 PivotTable erstellen

Eine Erstellung von Pivot-Technik kann man mittels **Menüband Einfügen** beginnen.

Am Einfachsten arbeitet man auf eigenem Tabellenblatt mit speziellen Menübändern für die Pivot-Konfiguration.

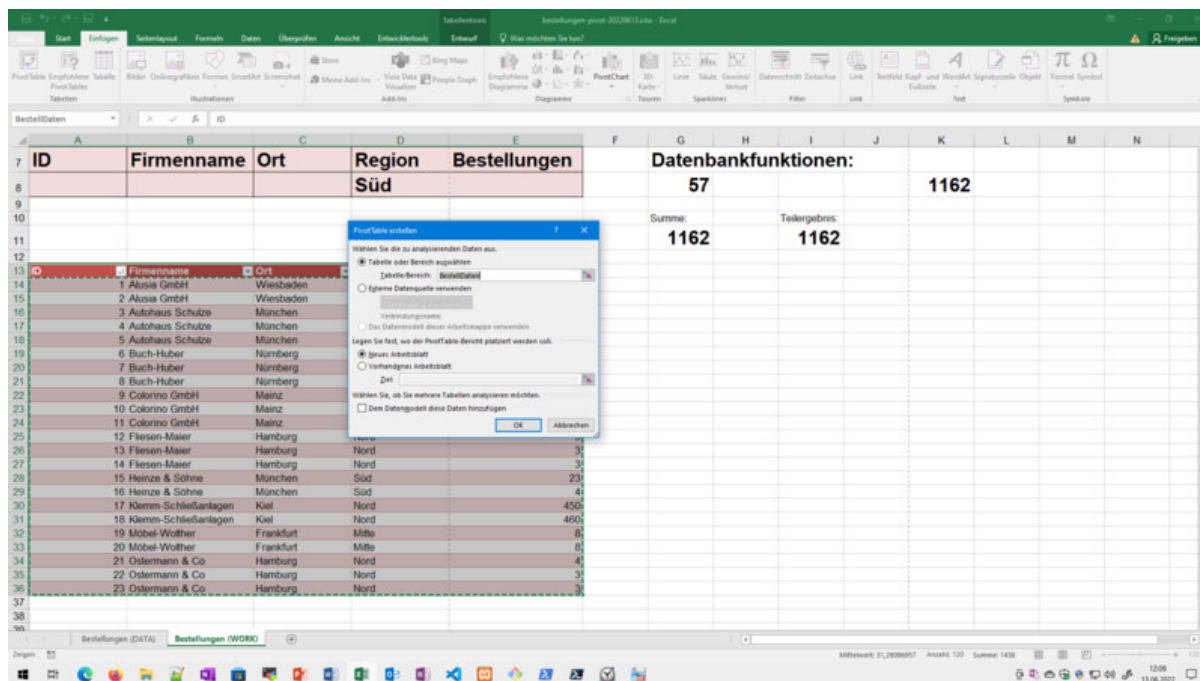


Abb. 4.2: Pivot erstellen

Eigenschaften von Feldern und Gesamt-PivotTable (z.B. automatisches Anpassen Spaltenbreiten deaktiviert)

Hinweis: Beim Ändern der Datenbereiche für die PivotTable muss man diese manuell aktualisieren

Eine Aktualisierung der Pivot-Daten erreicht man mittels Kontextmenü oder Schaltfläche **Aktualisieren** im Menüband Optionen der PivotTable-Tools.

Konfiguration / Layout aktualisieren

Über die Konfiguration des PivotTable Bereichs kann man einstellen, dass die **Spaltenbreiten** nicht immer wieder auf optimale Spaltenbreite *optimiert* wird.

So entsprechen die Spaltendarstellungen den eigenen Dimensionswünschen, auch wenn wir die PivotTable aktualisieren.

Es sind mehrere Auswertungen (Summe, Anzahl) in Bereich „Werte“ möglich.

4.2 PivotChart erstellen

Auch diese Erstellung von Pivot-Technik kann man mittels **Menüband Einfügen** beginnen.

Am Einfachsten arbeitet man auf wieder eigenem Tabellenblatt mit speziellen Menübändern für die Pivot-Konfiguration.

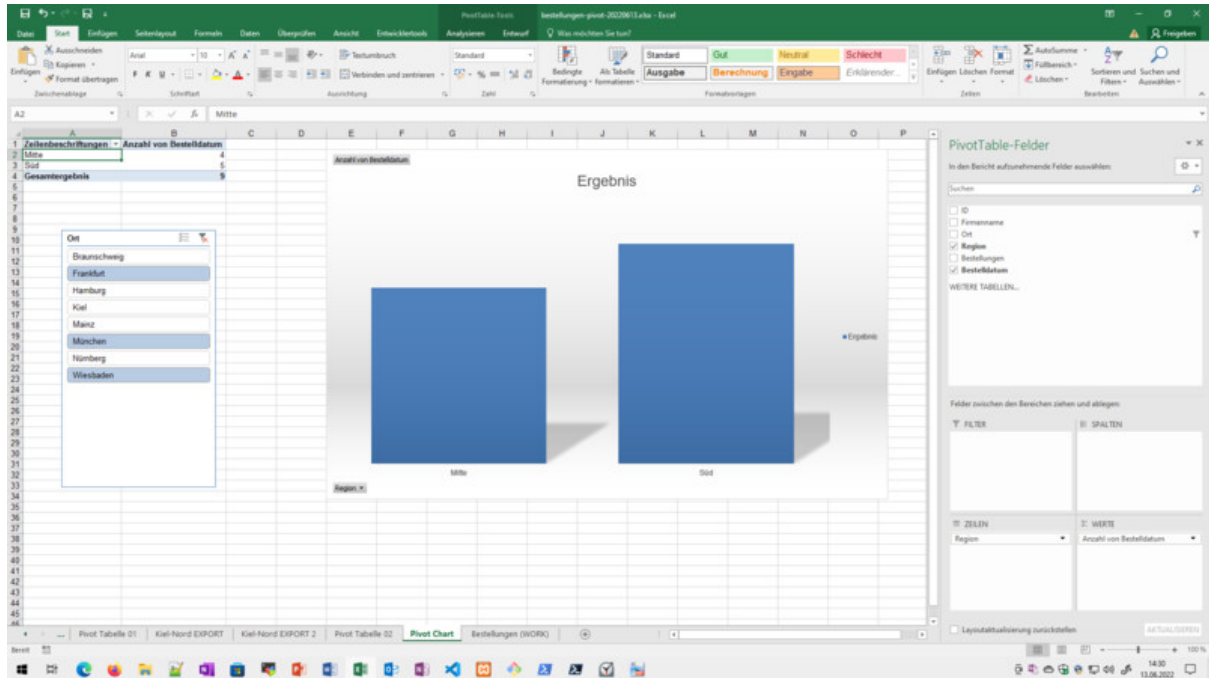


Abb. 4.3: Pivot Chart erstellen

Eigenschaften von Feldern und Gesamt-PivotTable (z.B. automatisches Anpassen Spaltenbreiten deaktiviert)

Hinweis: Beim Ändern der Datenbereiche für die PivotTable muss man diese manuell aktualisieren

Eine Aktualisierung der Pivot-Daten erreicht man mittels Kontextmenü oder Schaltfläche **Aktualisieren** im Menüband Optionen der PivotTable-Tools.

4.3 Pivot Gruppierungen

Gruppierte Bereiche (in Zeilen und/oder Spalten) möglich:

Summe von Bestellungen	Spaltenbeschriftungen	Nord	Süd	Gesamtergebnis
	Zeilenbeschriftungen			
	Braunschweig		15	
	Brandes & Co		15	
	Frankfurt	16		
	Möbel-Wolther	16		
	Hamburg		19	
	Fliesen-Maier		9	
	Ostermann & Co		10	
	Kiel		910	
	Klemm-Schließanlagen		910	
	Mainz	55		
	Colorino GmbH	55		
	München			47
	Autohaus Schulze			20
	Heinze & Söhne			27
	Nürnberg			10
	Buch-Huber			10
	Wiesbaden	105		
	Alusia GmbH	105		
	Gesamtergebnis	176	944	57

Abb. 4.4: Pivot Tabelle (gruppiert)

Hier wurde in erster Ebene nach **Ort** (Feldbezeichner) und in zweiter Ebene nach **Firmenname** gruppiert. Für die Aktivierung/Deaktivierung der Gruppenhierarchie- Symbole (+/-) kann man eine Schaltfläche im Menüband finden.

4.4 Datenschnitt / Zeitachse

Wenn man einen Pivot-Bericht erstellt hat, kann man mit den folgenden **Grafischen Filtermechanismen** weitere Analysen dynamisch vornehmen.

Orte	Region - Nord		Mitte		Süd		Gesamt	
	Anzahl	Summen	Anzahl	Summen	Anzahl	Summen	Anzahl	Summen
Braunschweig	1	15,0 Einh					1	15,0 Einh
Frankfurt			1	8,0 Einh			1	8,0 Einh
Hamburg	5	15,0 Einh					5	15,0 Einh
Mainz			2	15,0 Einh			2	15,0 Einh
München					3	20,0 Einh	3	20,0 Einh
Gesamtergebnis	6	30,0 Einh	3	23,0 Einh	3	20,0 Einh	12	73,0 Einh

Abb. 4.5: Datenschnitt und Zeitachsen bei PivotTable

Datenschnitt

Einfügen mittels Menüband PivotTable-Tools - Optionen - Gruppe Sortieren und Filtern. So lassen sich weitere attraktive Filterungen manuell durchführen.

Zeitachsen

Für Felder mit **Datum/Uhrzeit** lassen sich zeitliche Filterungen mittels der **Zeitachsen** durchführen.

4.5 Pivot Daten Export

Tip: mit Doppelklick auf Zusammenfassungen in dem Pivot-Bericht erhält man automatisch eine neues Tabellenblatt mit den Detailwerten der Pivot-Zusammenfassung!

ID	Firmenname	Ort	Region	Bestellungen	Bestelldatum
24	Brandes & Co	Braunschweig	Nord	15	03.01.2022
14	Fliesen-Maier	Hamburg	Nord	3	01.10.2021
13	Fliesen-Maier	Hamburg	Nord	3	30.03.2023
12	Fliesen-Maier	Hamburg	Nord	3	05.04.2021
18	Klemm-Schließel	Kiel	Nord	460	26.11.2024
17	Klemm-Schließel	Kiel	Nord	450	09.04.2025
23	Ostermann & Co	Hamburg	Nord	3	31.01.2022
22	Ostermann & Co	Hamburg	Nord	3	01.05.2023
21	Ostermann & Co	Hamburg	Nord	4	27.07.2020

Abb. 4.6: Pivot Datenexport per Doppelklick

PIVOTDATENZUORDNEN()

Die berechneten Werte im Pivot-Bericht lassen sich einfach per Klick in Formeln übernehmen und werden clever als Pivot-Zuweisungen und nicht als relative (oder gar absolute) Zellbezüge (z.B. H19) übernommen.

Beispiel:

=PIVOTDATENZUORDNEN("Gehalt";\$A\$3;"Abteilung";"IT")

4.6 Pivot Measure

Mit dieser Umsetzung kann man Berechnungen für neue Spalten basierend auf Pivot-Datenmodell-internen Spalten vornehmen.

The screenshot shows an Excel PivotTable with the following data:

Zeilenbeschriftungen	Summe von Bestellziele	Summe von Bestellungen	BestellungenDifferenz
Mitte	400	176	-224
Nord	350	944	594
Süd	500	57	-443
Gesamtergebnis	1250	1177	-73

The 'Messure' dialog box shows the following details:

- Tabellenname: BestellDaten
- Messurenname: BestellungenDifferenz
- Wertbeschreibung:
- Formel: $= [Summe von Bestellungen] - [Summe von Bestellziele]$
- Kategorie: Standard

Abb. 4.7: Pivot Measure (neue berechnete Spalte)

Bei dieser Umsetzung (Screenshot) kann man auch erkennen, dass wir hier zwei Tabellen für die Auswertung heranziehen (siehe Datenmodell).

4.7 Pivot - Beziehungen

Mit Pivot Technik kann man auch aus mehreren Tabellen Daten auswerten. Hierfür muss man die Tabellen in **Beziehungen** setzen.

Das heißt, dass man die Tabellen mit einer Verknüpfung zwischen Feldern der Tabellen ergänzt.

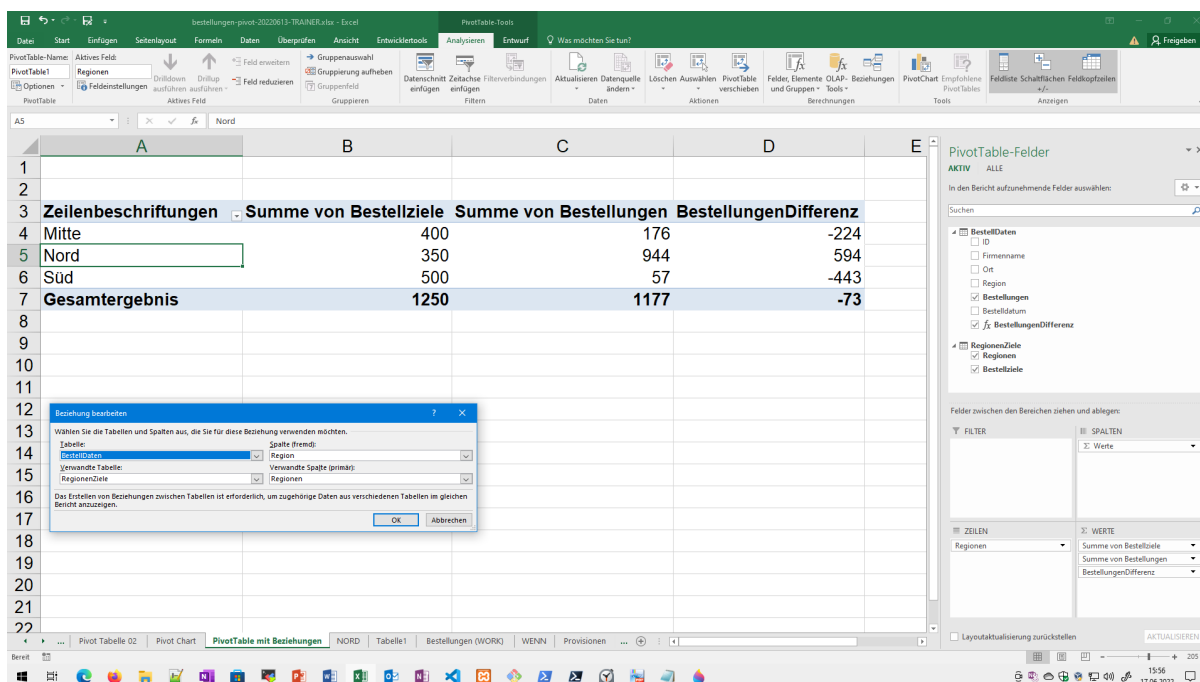


Abb. 4.8: Pivot Beziehungen

Hier wurden zwei Tabellen (BestellDaten, RegionenZiele) über die Felder **BestellDaten.RegionenZiele** und **Region.Regionen** miteinander verbunden.

4.8 Power Pivot

Für die professionelle Umsetzung von Datenmodellen und Pivot-Technik legt Microsoft (heute ein kostenloses) **COM-AddIn** bei.

Dieses lässt sich über die Excel-Optionen aktivieren und ist dann als eigenständiges Menüband in Excel verfügbar.

Die Power Pivot Datenmodellierung sieht dann so aus:

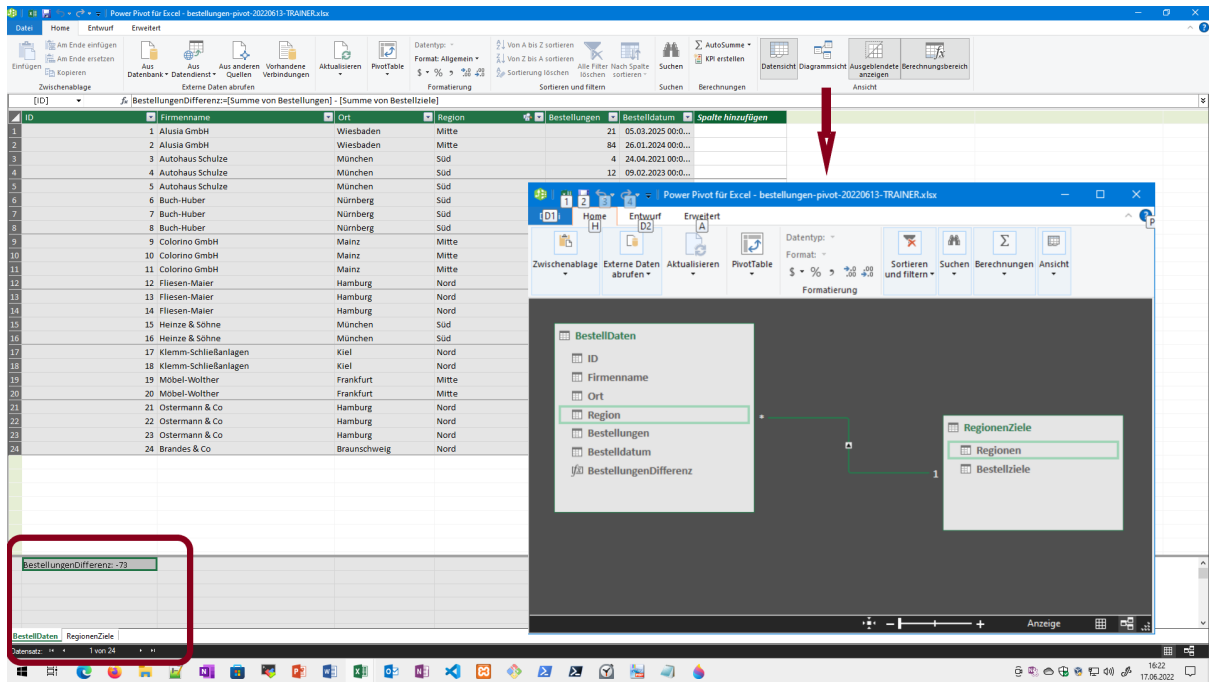


Abb. 4.9: Power Pivot (Daten- und Diagrammansicht)

Im Bereich unten links sieht man die Tabellen und Measures (Berechnungen) für die Daten.

Das Ganze stellt eine eigene Programmumgebung in Assistenz zu Excel dar und sprengt in den allermeisten Excel-Seminaren die zeitlichen Möglichkeiten.

Ein mehr als ordentliche Kurzeinführung liefert **Kevin Stratvert** auf seinem Youtube-Kanal.

[How to use Power Pivot in Excel | Full Tutorial](#)²¹

²¹ https://www.youtube.com/watch?v=rB_IiYbOo7w

#stratvert
How to use Power Pivot in Excel | Full Tutorial
549.512 Aufrufe · 05.03.2021

13.472 MAG ICH NICHT TEILEN THANKS CLIP SPEICHERN ...

Kevin Stratvert
1,41 Mio. Abonnenten

ABONNIEREN

In this step-by-step tutorial, learn how to use Power Pivot in Microsoft Excel. Power Pivot is a data modeling technology that lets you create data models, establish relationships, and create calculations. With Power Pivot you can work with large data sets, build extensive relationships, and create complex (or simple) calculations, all in a high-performance environment, and all within the familiar experience of Excel.

Additional resources:
- Sample files to follow along: <https://1drv.ms/u/s!AmxrofZZIz-whK1Sw...>
- In-depth Pivot Table tutorial: <https://youtu.be/m0wl61ahfLc>
- Learn the fundamentals of Excel in just 2 hours: <https://kevinstratvert.thinkific.com>

Abb. 4.10: Power Pivot (Kevin Stratvert Youtube)

Und er liefert auch noch gleich die Beispieldaten per OneDrive Download mit!

Diagramme 102

Wir können uns auch ein wenig tiefer mit Diagrammtechnik unter Excel beschäftigen.

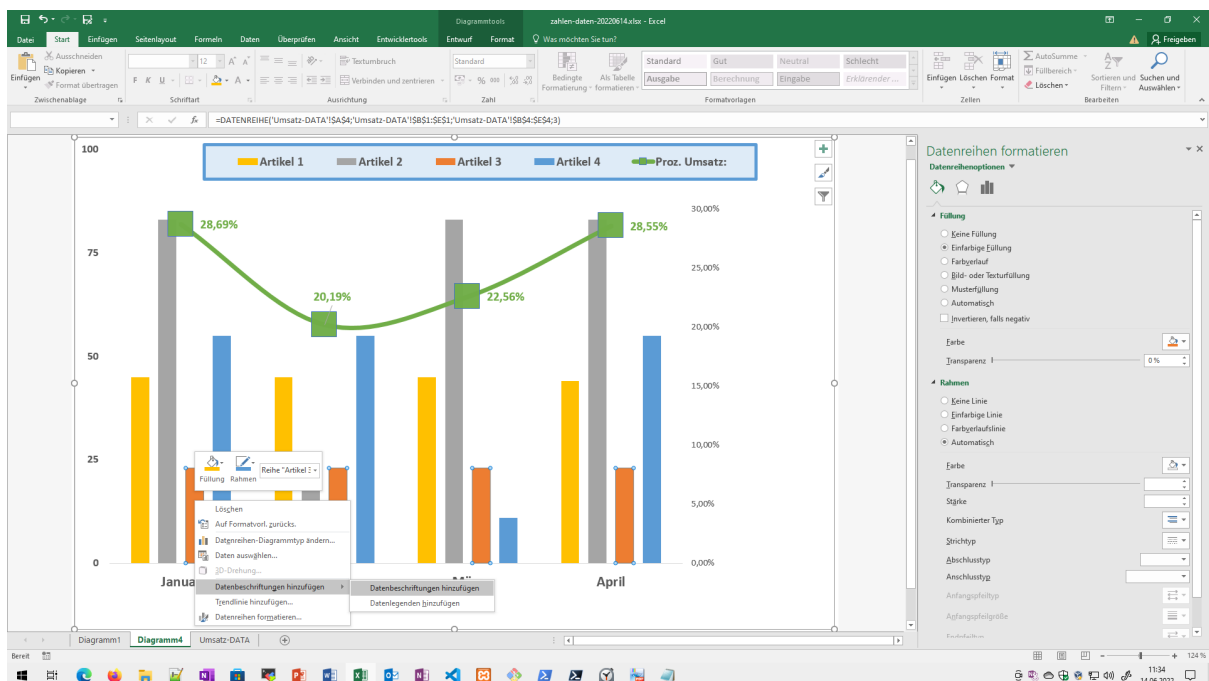


Abb. 5.1: Excel Diagramme (102)

5.1 Diagrammtypen

Hier soll es um die Diagrammtypen und ihre Umsetzung mit Excel gehen.

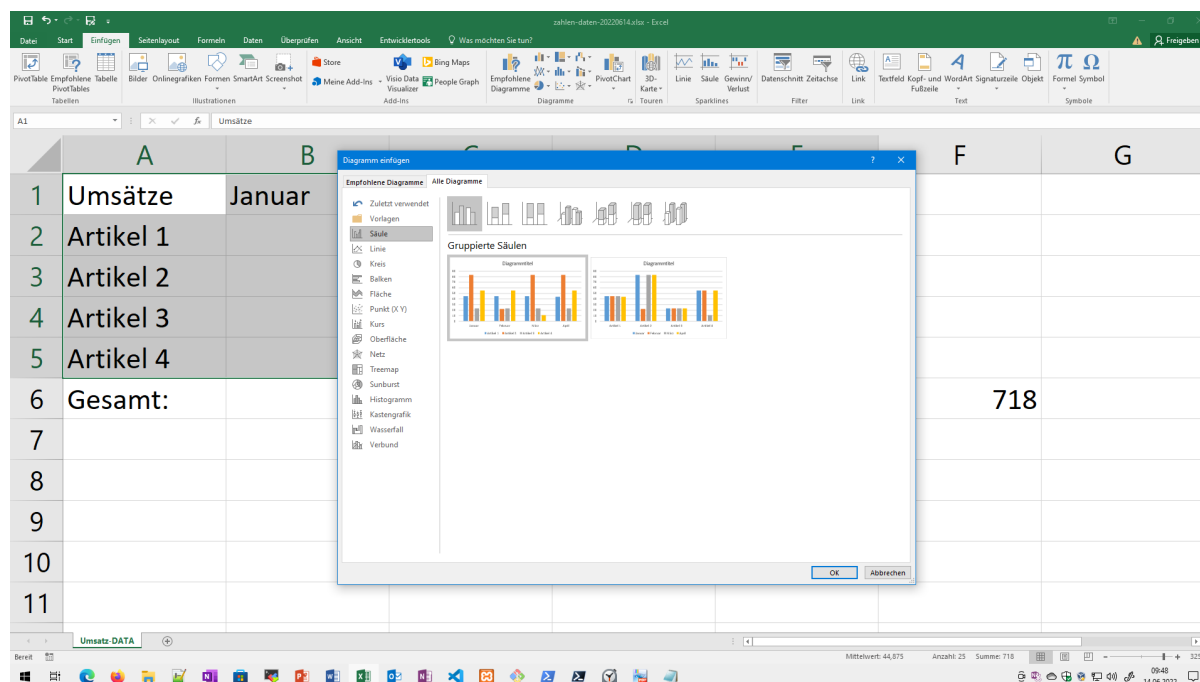


Abb. 5.2: Excel Diagrammtypen

Neben den Kategorien für Standarddiagramme sind zwei Kategorien von besonderem Interesse:

- **Vorlagen** - hier sind unsere Diagrammvorlagen auswählbar
- **Verbund** - mittels Verbunddiagrammen können Datenreihen spezifisch konfiguriert und auf die Sekundärachse zugewiesen werden

5.2 Verbunddiagramm

Mittels Verbunddiagramm kann man verschiedene Diagrammaspekte in einer Diagrammumsetzung beeinflussen.

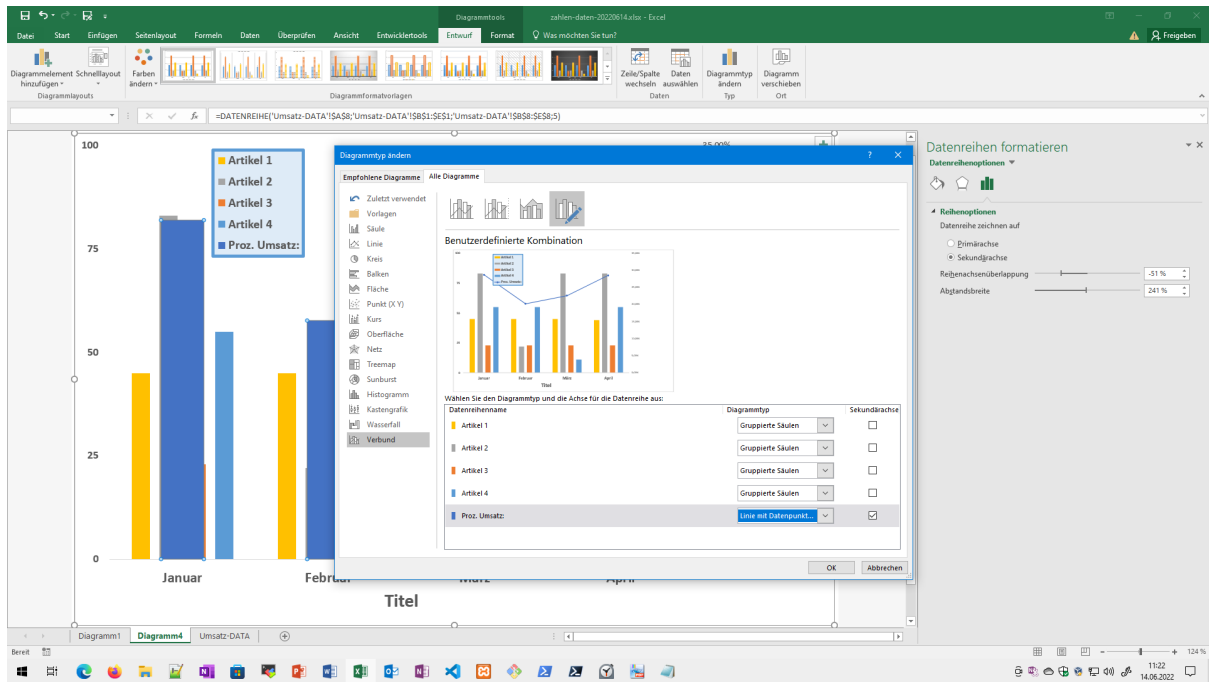


Abb. 5.3: Excel Verbunddiagramm (mit Sekundärachse)

Im Screenshot erkennt man den Verbund (der Datenreihen) und die individuelle Zuweisung von Diagrammtyp und Sekundärachse.

Auch im rechten Aufgabenbereich (hier: Datenreihen formatieren) kann die Einstellung zur Sekundärachse vorgenommen werden. Des Weiteren können Einstellungen zum Balkendesign (Breiten, Überlagerungen) eingestellt werden.

5.3 Diagrammvorlagen

Nachdem man die gewünschten Formatierungen an einem Beispieldiagramm vorgenommen hat, speichert man diese Technik einfach per Kontextmenü - Als Vorlage.

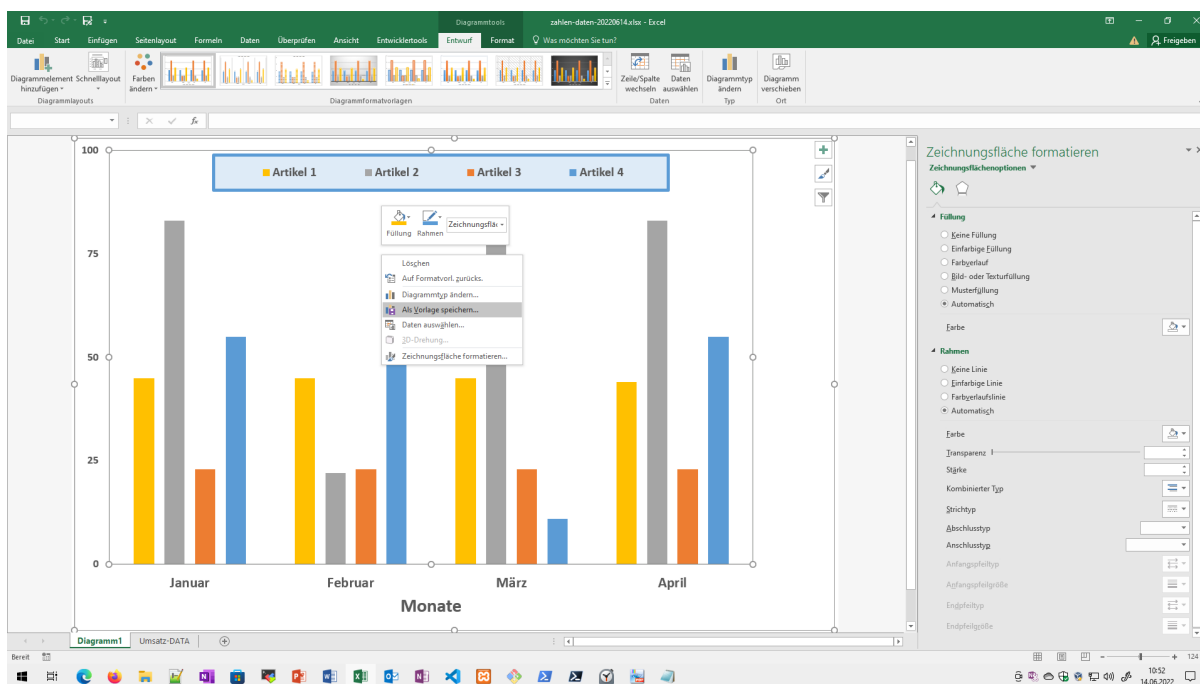


Abb. 5.4: Excel Diagrammvorlage speichern per Rechte Maus

In „älteren“ Excel-Versionen gab es außerdem noch eine Schaltfläche im Menüband.

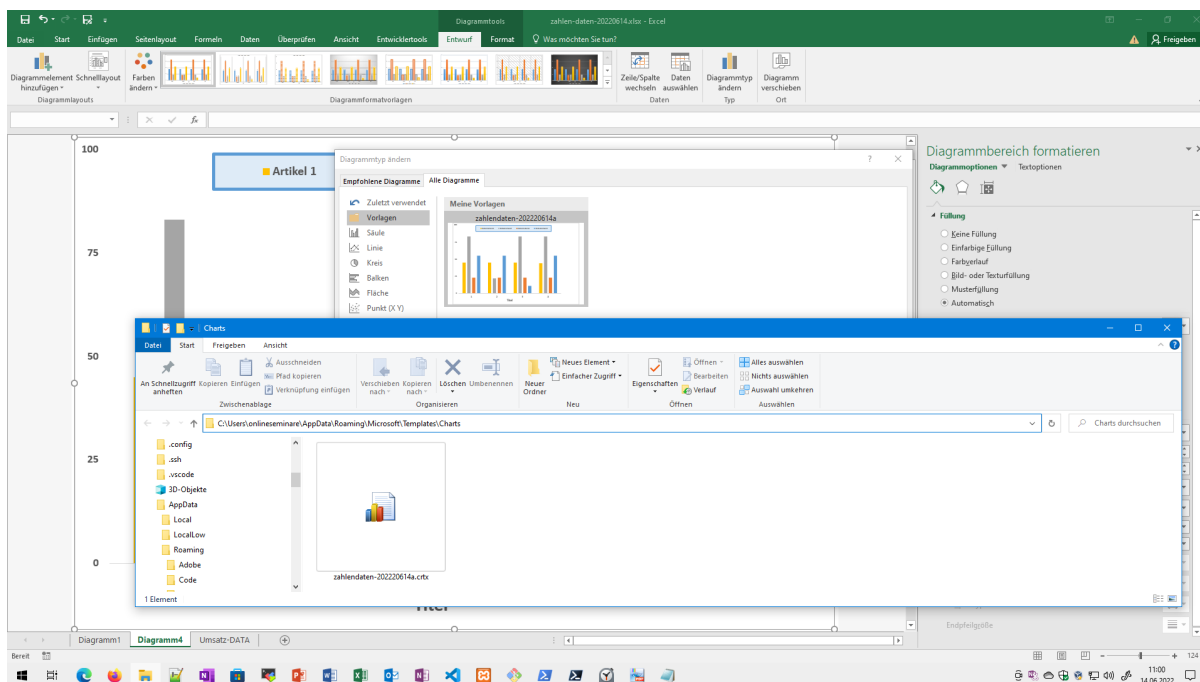


Abb. 5.5: Excel Diagrammvorlagen

Die Diagrammvorlagen liegen als *.crtx Dateien im AppData-Roaming-Profil des Windows Benutzers und sollten bei den Backupstrategien berücksichtigt werden!

5.4 Daten zu Diagramm hinzufügen

Das nachträgliche Hinzufügen von Datenreihen kann entweder per Copy & Paste durchgeführt werden oder über das Bearbeiten der Datenauswahl.

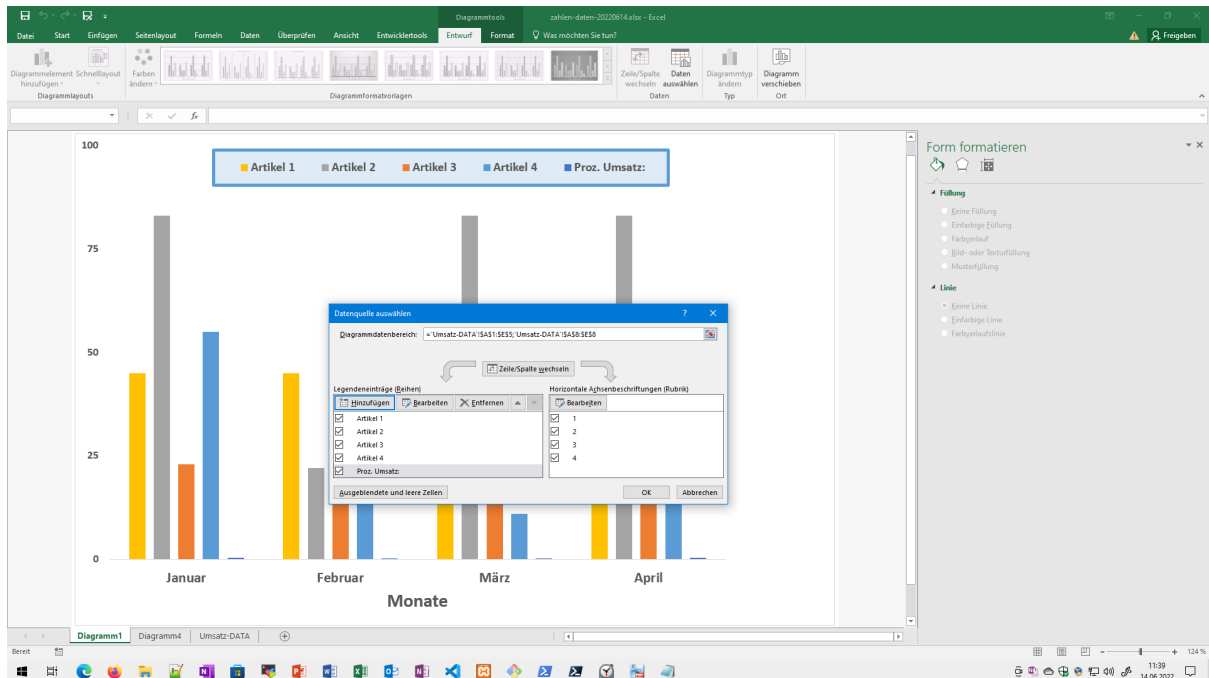


Abb. 5.6: Datenreihen hinzufügen

Über „Daten auswählen - Schaltfläche Hinzufügen“ kann man den Bezeichner und die Daten für die neue Datenreihe hinzufügen.

5.5 Datenvisualisierung

Für die Manipulierung und Darstellung von Excel-Diagramm kann man noch viele Beispiel anführen.

Oder man greift einfach zur Fachliteratur: *Datenvisualisierung mit Excel*

Das **Praxisbuch von Dirk Siegmann** zeigt auf knapp 400 Seiten viele Beispiele zur Datenvisualisierung.

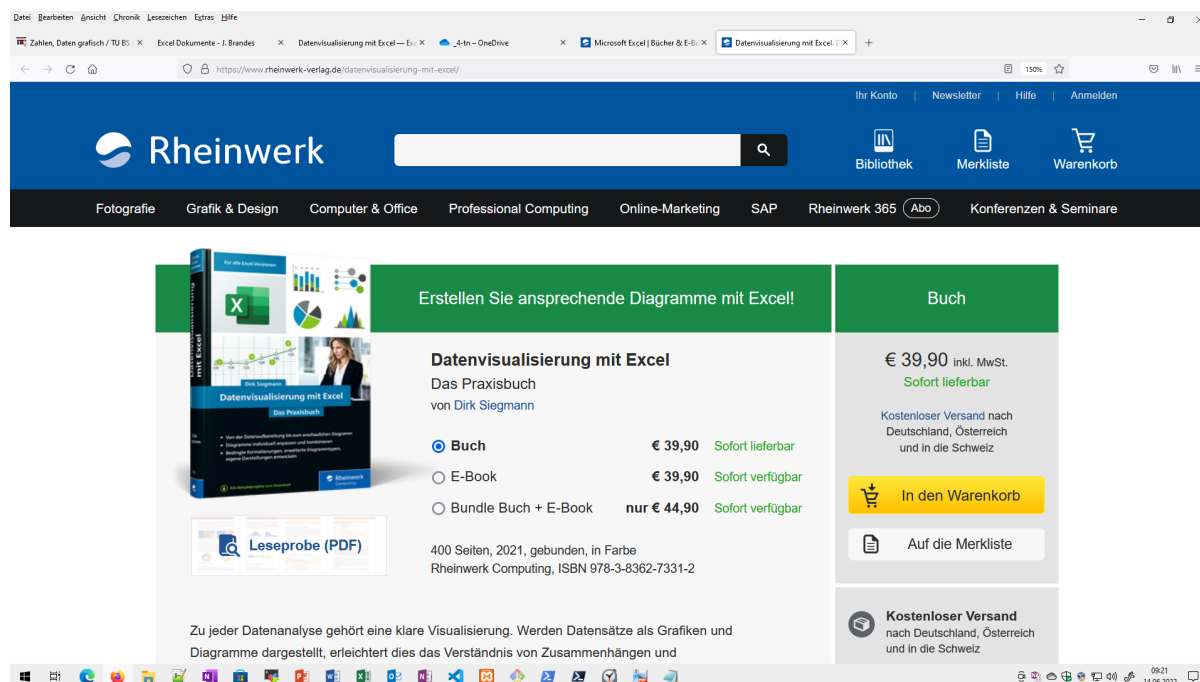


Abb. 5.7: Datenvisualisierung mit Excel (Dirk Siegmann)

Die Bücher des Rheinwerk-Verlag sind als Bundle (also Kombination mit PDF/E-Book/Mobi) DRM-frei erhältlich. Materialien zum Buch sind als Download verfügbar!

5.6 Animation mit PowerPoint

Für die Animation von Diagrammen lassen wir Excel und PowerPoint zusammenarbeiten. Man kann zwar auch mit Excel-Basistechniken (und oft in Kombination mit VBA) für Animationen sorgen, aber am Besten lässt man die Animationen für die Diagramme den Animations-Profi der Office-Suite übernehmen: PowerPoint.

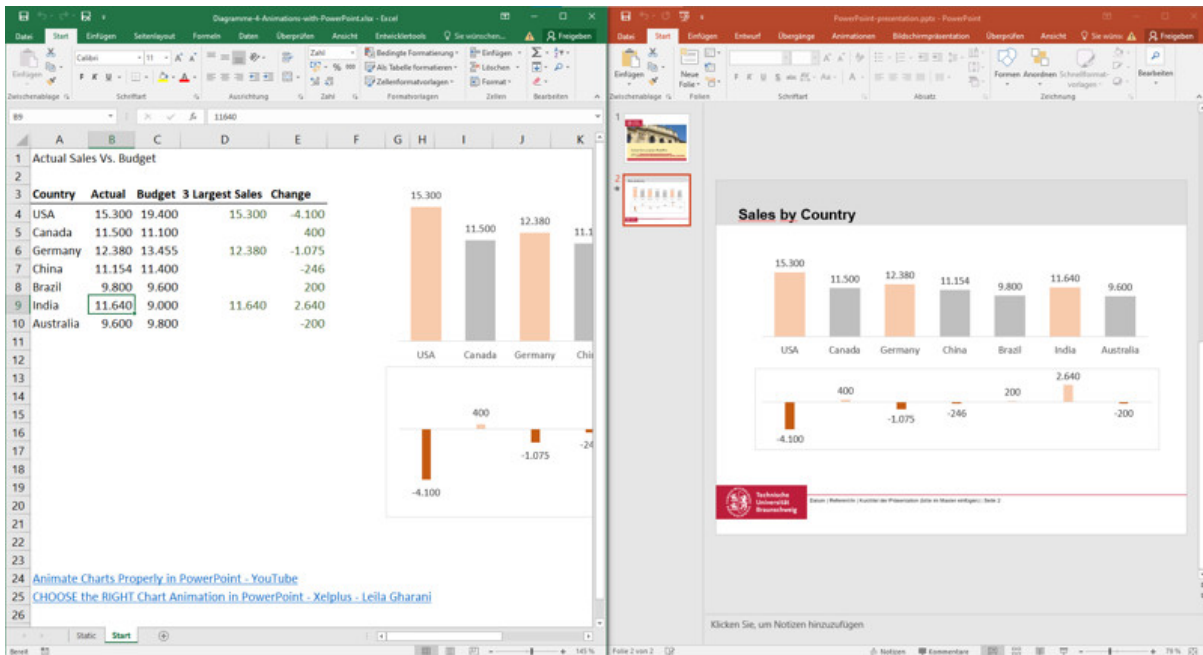


Abb. 5.8: Excel Diagramme mit PowerPoint Animieren

Das Ziel einer praktischen Übung im Seminar wird durch das „Header-Bild“ dieses Beitrags bereits deutlich.

Auf der linken Seite erkennt man vorbereitete Daten in Excel (Werte und zwei Diagramme) und auf der rechten Seite ist eine PowerPoint-Folie mit den verknüpften Diagrammdateien.

Für die Umsetzung ist die richtige Copy & Paste Technik wichtig.

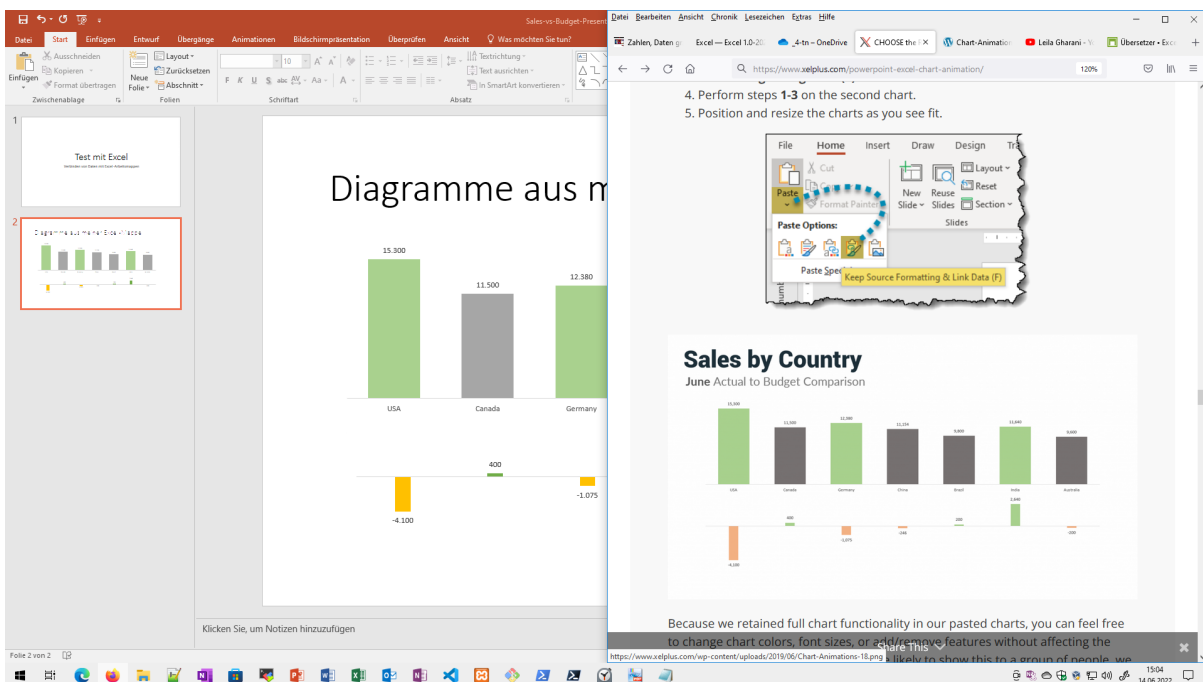


Abb. 5.9: Excel Diagramme mit PowerPoint verbinden

Durch diese Implementierung mit „Link Data (Datenverbindung)“ der Excel-Diagramme bekommt man

„PowerPoint-integrierte Diagrammtechnik“ mit den Daten aus der Excel-Mappe!

Das Übungsbeispiel resultiert aus einer Veröffentlichung eines sehr erfolgreichen Excel Portal (xelplus.com) und dem zugehörigen Youtube-Kanal (User: Leila Gharani).

Website zur beispielhaften Übung: <https://www.xelplus.com/powerpoint-excel-chart-animation/>

Arbeitsmappenvorlagen

Für Arbeitsmappenvorlagen hält Excel verschiedene Dateitypen parat:

- **.xlsm** (kann aktive Elemente / Makros enthalten)
- **.xltx** (klassische Vorlagen)

Die Arbeitsmappenvorlage fungieren wie in anderen Office Programmen: Sammeln von Inhalten, Funktionen oder auch **Makros** in einer Mappe und bereitstellen als Template für neue Arbeitsmappen.

Über neue Mappe - Meine Vorlagen lassen sich dann die Mappen inkl. der gewünschten Elemente nutzen.

Arbeitsmappenvorlagen werden mittels „Speichern unter...“ und Dateityp Excel-Vorlage ***.xltx** in den passenden Vorlagenordner des Office-Users speichern.

Vorlagen für Mappen mit Makros natürlich mit ***.xlsm** speichern!

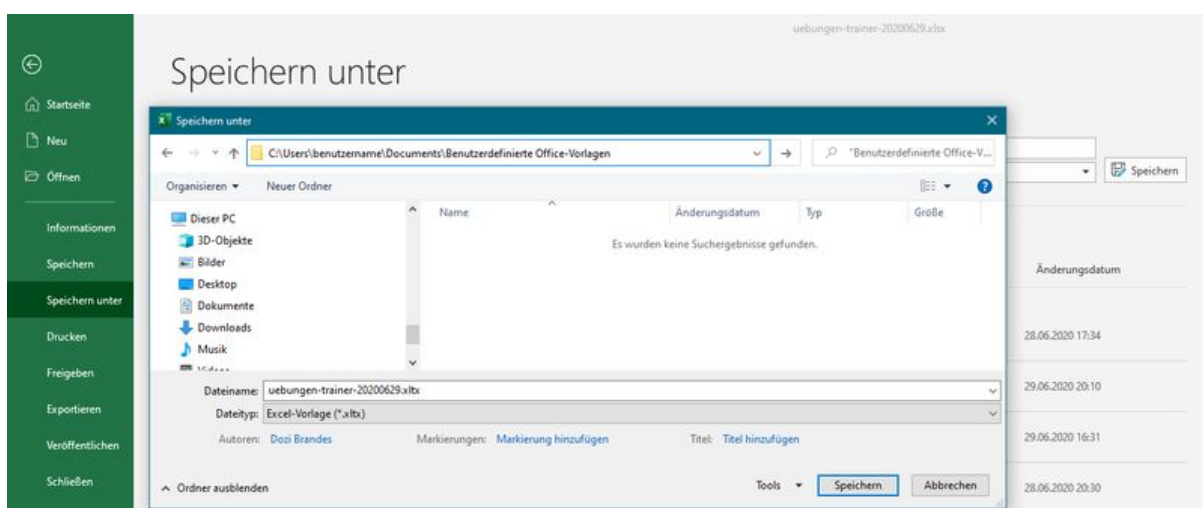


Abb. 6.1: Template speichern

Aktueller Windows Betriebssystem Standardordner: (mit Windows 10/11 und Office 365)

C:\Users\benutzername\Documents\Benutzerdefinierte Office-Vorlagen

Anm.: im Windows Explorer einfach deutschen Ordner „Dokumente“ aufsuchen...

Die Vorlagen können über das Menüband Datei - Neu - Rubrik: Persönlich genutzt werden.

Speicherort für Vorlagen in Vorgänger-Windows/Office-Kombinationen:

..\AppData\Roaming\Microsoft\Templates

Tipp: Mit Unterordnern im Templates-Ordner des Office lassen sich Registerkarten für Eigene/Meine Vorlagen erzeugen!

Was-wäre-wenn-Analysen

Auszug aus dem Hilfe Portal von Microsoft zum Thema:

Mithilfe von Tools für Was-wäre-wenn-Analysen in Excel können Sie unterschiedliche Wertegruppen in einer oder mehreren Formeln verwenden, um die verschiedenen Ergebnisse zu untersuchen.

Sie können z. B. eine What-If erstellen, um zwei Budgets zu erstellen, von der jedes ein bestimmtes Maß an Umsatz vorausnimmt. Sie können aber auch ein Ergebnis angeben, das eine Formel erzeugen soll, und dann ermitteln, welche Gruppen von Werten dieses Ergebnis erzeugen. Excel bietet verschiedene Tools, mit denen Sie den Typ von Analyse ausführen können, die Ihren Anforderungen entspricht.

Link Übersicht zu Analysen: [Support Microsoft Was-wäre-wenn-Analysen](#)²²

Excel verfügt über drei unterschiedliche Ansätze für diese Analysen

- Szenarien
- Datentabellen
- Zielwertsuche

Darüber hinaus hält Excel mit dem *sogenannten Solver* und diversen Add-Ins weitere Tools in diesem Technikbereich parat.

Konzentrieren wir uns wie gesagt auf die drei Grundansätze für die Zahlensimulationen mit Excel.

Wir starten mit den **Szenarien**.

²² <https://support.microsoft.com/de-de/office/einf%C3%BChrung-in-die-was-w%C3%A4re-wenn-analyse-22bffa5f-e891-4acc-bf7a-e4645c4>

7.1 Szenarien

Mit Hilfe des Szenario-Manager-Assistenten (siehe Menüband Daten) lassen sich verschiedenen Werte für Zellen festlegen, mit denen dann die Ergebnisse berechnet werden.

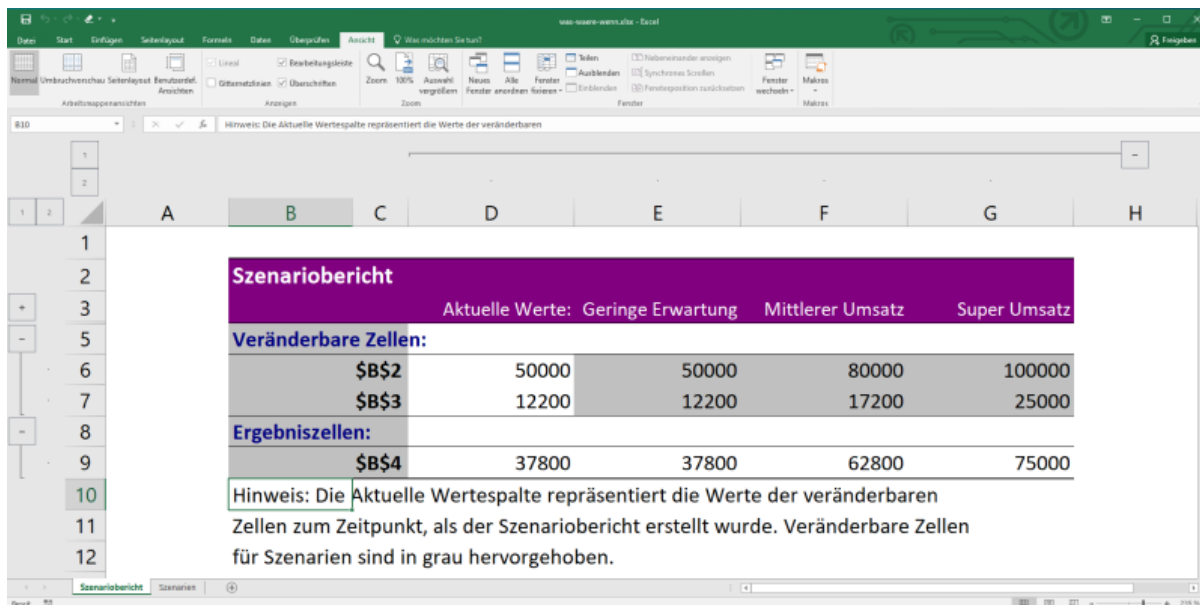


Abb. 7.1: Szenario

Link zu Anleitung/Beispiel: [Support Microsoft Szenarien²³](https://support.microsoft.com/de-de/office/wechseln-sie-zwischen-verschiedenen-wertgruppen-mithilfe-von-szenarien-2068afb1-ecdf-4956)

7.2 Datentabellen

Hier kann man einen Zellbereich mit unterschiedlichen Werten vorgeben und erhält entsprechende Antworten (Berechnungen) auf die eigenen Werte.

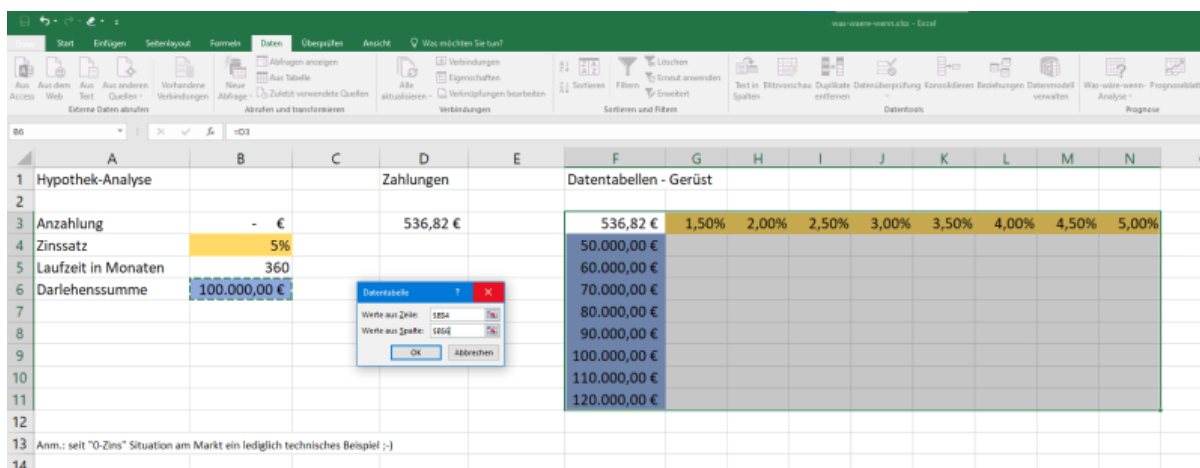


Abb. 7.2: Datentabelle

²³ <https://support.microsoft.com/de-de/office/wechseln-sie-zwischen-verschiedenen-wertgruppen-mithilfe-von-szenarien-2068afb1-ecdf-4956>

Link zu Anleitung/Beispiel: Support Microsoft Datentabelle²⁴

7.3 Zielwertsuche

Alles „rückwärts“: wir kennen das Ergebnis - bzw. geben dieses vor - und wollen den Wert / oder die nötigen entsprechenden Eingaben ermitteln.

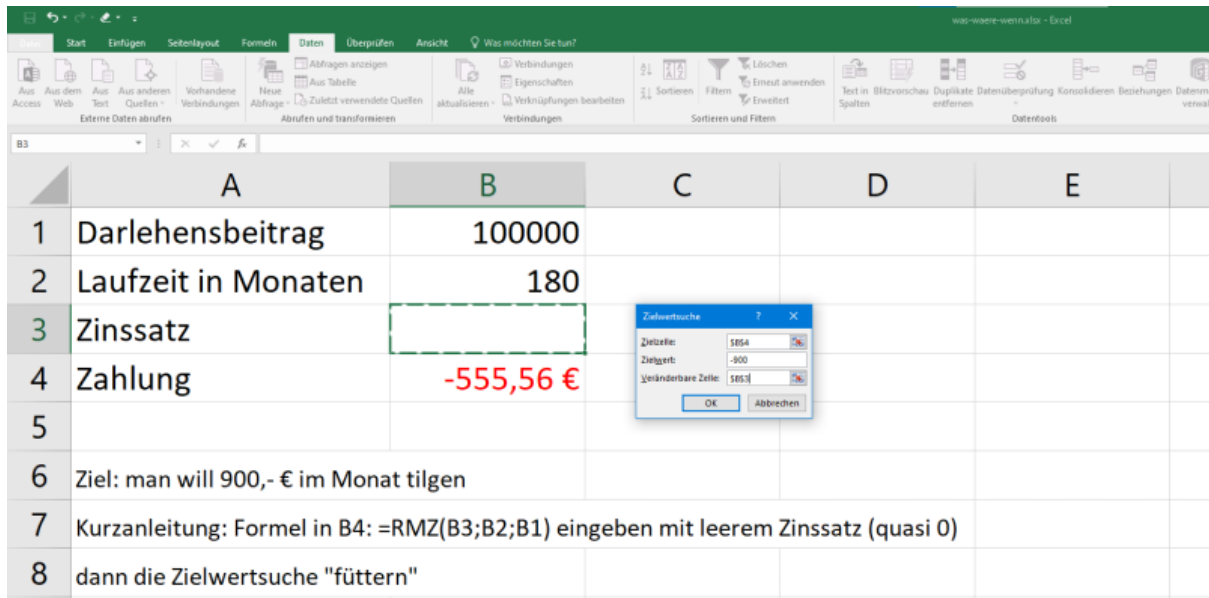


Abb. 7.3: Zielwertsuche

Link zu Anleitung/Beispiel: Support Microsoft Zielwertsuche²⁵

Spätestens für die Zielwertsuche gibt es auch freie Tools oder Erweiterungen für Excel: siehe Stichworte **Solver** oder **Solver-Add-Ins**.

²⁴ <https://support.microsoft.com/de-de/office/berechnen-von-mehreren-ergebnissen-mit-einer-datentabelle-e95e2487-6ca6-4413-ad12-77542a>

²⁵ <https://support.microsoft.com/de-de/office/verwenden-der-zielwertsuche-zum-ermitteln-des-gew%C3%BCnschten-ergebnisses-durch-anpassen-eines-eingabewerts-320cb99e-f4a4-417f-b1c3-4f369d6e66c7>

Tipp: Ich empfehle an dieser Stelle auch die Einblendung des **Menüband Entwicklertools** über die Optionen von Excel.

8.1 Vorlagen - Rekap

Allgemein und zur Wiederholung/Erinnerung: Vorlagen sammeln die Inhalte, Funktionen oder auch **Makros** in einer Mappe und stellen diese Techniken bereit.

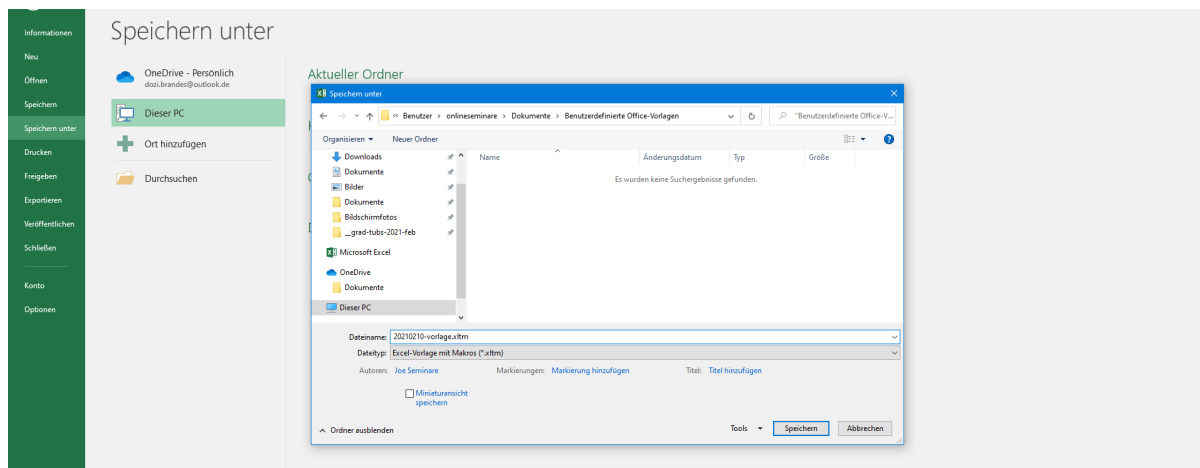


Abb. 8.2: Vorlage mit Datei xltn speichern

In Kürze:

- Über Menüband - Neue Mappe - Meine Vorlagen lassen sich dann die Mappen inkl. der **aktiven Elemente** nutzen.
- Arbeitsmappenvorlagen werden mittels „Speichern unter...“ und Dateityp in passenden Vorlagenordner des Office-Users speichern.
- Die Vorlagen für Mappen **mit Makros** müssen mit Dateityp ***.xltn** gespeichert werden!

Aktueller Vorlagen-Standardordner für Windows/Office Kombinationen:

C:\Users\benutzername\Documents\Benutzerdefinierte Office-Vorlagen

Anm.: im Windows Explorer natürlich den *deutschen* Ordner **Dokumente** öffnen.

Die Vorlagen können über Menüband Datei - Neu - Rubrik: Persönlich genutzt werden.

Speicherort für Vorlagen in Vorgänger-Windows/Office:

..\AppData\Roaming\Microsoft\Templates

Tipp: Unterordner im Templates-Ordner des Office lassen Registerkarten für Meine Vorlagen entstehen!

8.2 Vorlagen - mit Makros

Für die Umsetzung mit Makros müssen wir die Nutzung der richtigen Dokumenttypen beachten.

Tab. 8.1: Excel Dateiformate (inkl. Makrotypen)

Versionen	Dateiformate
Excel 97-2003	xls - das klassische Excel-Sheet (Arbeitsmappen) xlt - die passenden Vorlagen (engl.: Templates)
Excel 2007/2010/...	xlsx - XML basiertes Dokumentformat (ohne Makros) xltx - die passenden Vorlagen xlsm - XML basierte Mappen inkl. Makros xltm - die passenden Vorlagen inkl. Makros xlsb - spezielles Binärformat (siehe PERSONAL.xlsb)

Die *fett dargestellten* Formate **xlsm**, **xltm** und **xlsb** unterstützen aktive Inhalte (siehe Makros) und sollten besonders betrachtet und sicherheitstechnisch beobachtet und bewertet werden.

Die *Legacy* Formate xls und xlt sollten nicht mehr genutzt werden, da sie proprietäre und *nicht offene* Formate darstellen.

Die modernen Formate (mit x wie XML seit 2007) sind ein **offenes Dokumentformat** und stellen technisch **Zip-Archive** mit interner Dokumentstruktur dar.

8.3 PERSONAL.XLSB

Eine besondere Rolle bei der Nutzung von aktiven Techniken (Makros) mit Excel nimmt die **Persönliche Makroarbeitsmappe PERSONAL.XLSB** ein.

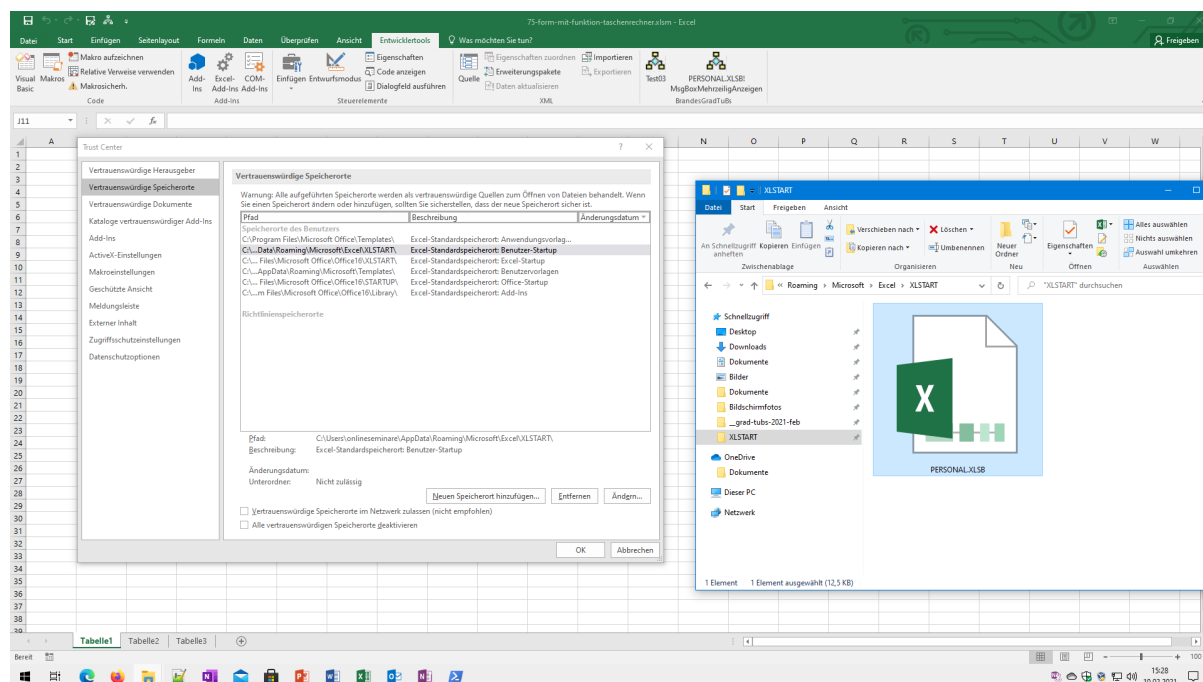


Abb. 8.3: Persönliche Makroarbeitsmappe (PERSONAL.XLSB)

Es handelt sich um eine *binäre* automatisch genutzte Arbeitsmappenvorlage inklusive Makros.

Excel Speicherort für diese Persönlichen Makro-Arbeitsmappen:

.. \AppData\Roaming\Microsoft\Excel\XLSTART\PERSONAL.XLSB

Wir bekommen sie also in unser Benutzerprofil kopiert und der .. \AppData\Roaming Pfad ermöglicht in Firmen (AD-Umgebungen) auch die zentrale Bereitstellung, sodass diese Mappe den Nutzern immer und überall zur Verfügung steht.

Die persönliche Makroarbeitsmappe steht nicht per Default zur Verfügung, sondern muss manuell erstellt werden.

Tip: Einfache Erstellung der PERSONAL.XLSB: Dummy-Makro erstellen und mit „Makro speichern in: Persönliche Makroarbeitsmappe“ sichern. Hierdurch wird die PERSONAL.XLSB erstellt.

Danach lässt sich diese Technikquelle über den VBA-Editor - Projektextplorer verwalten (Code hinzufügen/kopieren).

8.4 Makros - Sicherheit

Wir sprechen allgemein auch von sogenannten **Aktiven Elementen** integriert in Office-Dokumenten. Und das ist nicht unproblematisch und verlangt Überlegungen und Konfigurationen für Sicherheit. Nicht ohne Grund hatten wir jahrelang die Einführung von Dokumentformaten (MS Office) ohne diese Techniken angeregt und mit den Standard-Dokumenttypen **xlsx**, **docx** oder auch **pptx** erhalten.

Anlaufstelle für die Konfiguration: Excel-Optionen - Vertrauensstellungszentrum - Einstellungen für Vertrauensstellungszentrum

Alternativ sind die Konfigurationen über Schaltfläche „Makrosicherheit“ im Menüband Entwicklertools erreichbar.

Hier: Einstellungen für Makros (möglichst unverändert lassen)!

Vertrauenswürdige Speicherorte

Eine besondere Rolle spielen hier die **Vertrauenswürdige Speicherorte**, mit denen wir Dateien in bestimmten Ordnern automatisch als vertrauenswürdig kennzeichnen können.

Bei Laufwerkspfaden im Netzwerk (fachlich: UNC Pfade - Universal Naming Convention) muss extra ein Kontrollkästchen für diese Nutzung aktiviert werden!

Tip: Bei den „Vertrauenswürdigen Speicherorten“ findet man auch die Verzeichnisse für Benutzer und Templates (siehe `PERSONAL.xlsb` in Ordner `.\XLSTART`)!

Und wieder: Speicherorte auf Netzwerkressourcen müssen immer extra erlaubt werden.

8.5 Makros - Aufzeichnen

Ein sehr guter Einstieg in die Automatisierungen mittels VBA (Visual Basic for Applications) bietet sich mit der Aufzeichnung von Makros an.

Problemstellung für Makro

Einen Tabellenbereich markieren und in Zwischenablage kopieren. Auf anderem Tabellenblatt *transponiert* wieder einfügen.

Hierbei muss man nur die Aufzeichnungstechnik finden und startet dann einen **Makro Recorder**. Dieser zeichnet bis zum Stoppen der Aufzeichnung die relevanten Aktionen auf.

Übung: Aufzeichnungen von Makros

Je nach Herangehensweise ergeben sich verschiedene Codes. Das Schöne an diesen Makroaufzeichnungen ist ihr *Try and Error* Charakter, der viel Raum für Freiheit und Erkenntnisse beim Ausprobieren bereit hält.

Tip: Bei fehlerhaften Makros bitte die **Stopp-Taste** des VBA-Editor klicken.

In den Seminaren erhalten die Trainees die passenden Tricks & Kniffe, um die Problemchen hierbei zu meistern. Insbesondere der Umgang mit Makros, die fehlerhaft *unterbrechen* wird trainiert.

Schauen wir uns einen ersten möglichen Quellcode eines Makros an.

Quellcode 8.1: Erste Makroaufzeichnung

```

1 Sub Transponieren01()
2 '
3 ' Transponieren01 Makro
4 ' einfaches Transponieren - Markieren mit Maus!
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

5 '
6 '
7 '
8 Range("A1:G4").Select
9 Selection.Copy
10 Sheets("Tabelle2").Select
11 Range("A1").Select
12 Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
↳ _
13     False, Transpose:=True
14 Range("A1").Select
15 End Sub

```

Beim Aufrufen dieses Makros wurden nur die einfachsten Dinge für die Erstellung eines Makros beachtet, sodass das Makro in vielen Fällen seine Arbeit verweigert.

In der nächsten Variante wird eine verbesserte Markierung des zu transponierenden Bereichs vorgenommen. So kann man beliebige Bereiche (Dimensionen von Zellen) sauber markieren.

Quellcode 8.2: Makroaufzeichnung mit verbesserter Markierungstechnik

```

1 Sub Transponieren02()
2 '
3 ' Transponieren02 Makro
4 ' jetzt mit intelligenter Tastenmarkierung
5 '
6 '
7 '
8 Range("A1").Select
9 Range(Selection, Selection.End(xlToRight)).Select
10 Range(Selection, Selection.End(xlDown)).Select
11 Selection.Copy
12 Sheets("Tabelle2").Select
13 Range("A1").Select
14 Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
↳ _
15     False, Transpose:=True
16 Range("A1").Select
17 End Sub

```

Und zum *Schluss* kann man die Ausgangssituation zu Beginn des Makros noch verbessern.

Quellcode 8.3: Makroaufzeichnung mit verbesserter Tabellenblattwahl

```

1 Sub Transponieren_03()
2 '
3 ' Transponieren_03 Makro
4 ' mit neuer Tabelle für Transponierung
5 '

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

6 '
7 '
8 Sheets("Tabelle1").Select
9 Range("A1").Select
10 Range(Selection, Selection.End(xlToRight)).Select
11 Range(Selection, Selection.End(xlDown)).Select
12 Selection.Copy
13 Sheets.Add After:=Sheets(Sheets.Count)
14 Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
15     False, Transpose:=True
16 Range("A1").Select
17 Application.CutCopyMode = False
18 End Sub

```

Und hier wurde auch noch ein neues Tabellenblatt erzeugt, auf welchem dann der Tabellenbereich wieder eingefügt wurde.

8.6 Absolut vs Relativ

Gemeint ist hier die Art und weise, wie man bei der Aufzeichnung von Makros die *Positionierungen bzw. Bewegungen* im Zellbereich per VBA-Code interpretiert.

Hier mal zwei Codes, die einen Vergleich ermöglichen:

Quellcode 8.4: Beispiel Absolute Aufzeichnung

```

Sub AbsoluteAufzeichnung()
'
' AbsoluteAufzeichnung Makro
' In dieser Aufzeichnung werden nur absolute Zellbezüge verwendet.
'
' Tastenkombination: Strg+Umschalt+A
'
Range("A1:A4").Select
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .Color = 65535
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With
Range("A1").Select
ActiveCell.FormulaR1C1 = "Name:"
Range("A2").Select
ActiveCell.FormulaR1C1 = "Vorname:"
Range("A3").Select
ActiveCell.FormulaR1C1 = "Wohnort:"
Range("A4").Select

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
ActiveCell.FormulaR1C1 = "Telefon:"  
Range("A1:A4").Select  
Selection.Font.Bold = True  
End Sub
```

Bitte im Vergleich der folgenden Relativen Aufzeichnung genau auf die VBA-Befehle achten.

Quellcode 8.5: Beispiel Relative Aufzeichnung

```
Sub RelativeAufzeichnung()  
'  
' RelativeAufzeichnung Makro  
' In dieser Aufzeichnung werden nur relative Bezüge verwendet.  
'  
' Tastenkombination: Strg+Umschalt+B  
'  
ActiveCell.Range("A1:A4").Select  
With Selection.Interior  
    .Pattern = xlSolid  
    .PatternColorIndex = xlAutomatic  
    .Color = 65535  
    .TintAndShade = 0  
    .PatternTintAndShade = 0  
End With  
ActiveCell.Select  
ActiveCell.FormulaR1C1 = "Name:"  
ActiveCell.Offset(1, 0).Range("A1").Select  
ActiveCell.FormulaR1C1 = "Vorname:"  
ActiveCell.Offset(1, 0).Range("A1").Select  
ActiveCell.FormulaR1C1 = "Wohnort:"  
ActiveCell.Offset(1, 0).Range("A1").Select  
ActiveCell.FormulaR1C1 = "Telefon:"  
ActiveCell.Offset(-3, 0).Range("A1:A4").Select  
Selection.Font.Bold = True  
End Sub
```

Der Schalter für diese Option also bitte genaustens vor einer Makroaufzeichnung bedenken und für die eigenen Umsetzung hinterfragen und testen.

Visual Basic for Applications

Die Programmierung mit **VBA (Visual Basic for Applications)** ist selbst als Skriptsprache eine sehr mächtige und umfangreiche Technik, die sich mal nicht so eben in kurzer Seminarzeit komplett erlernen lässt.

Aber wie immer bei Skriptsprachen - im Gegensatz zu komplexen Programmier-Hochsprachen (wie C) - kann man es beim Skripting eben auch erheblich *lockerer* angehen lassen.

Vieles lässt sich per **Copy & Paste** nach vorheriger Recherche ganz gut erledigen und anpassen. Wir wollen uns hierfür das passende **VBA Know How** erarbeiten. Und los geht es...

9.1 VBA-Editor

Tastenkombination: Alt + F11

Übungen: Manuelles Einfügen von

- **Sub** Routinen (Makros) oder
- **Function** (neue eigene Funktionen; Beispiel: NettoMwst)

Wir verschieben das Programmieren so weit es geht nach *hinten*.

Hinweis: Falls noch keine Struktur **Module** im Projekt-Explorer vorhanden ist, dann bitte mittels Einfügen - Module diese notwendige Struktur erstellen.

Mit individuellen Modulen lassen sich die VBA-Codes besser und effizienter organisieren!

Ein paar Basics für den VBA-Editor:

- F8 - Manuelles (zeilenweise) Abarbeiten mit Funktionstaste F8
- F1 - Aufruf der Hilfen zu VBA Befehlen und Aufruf F1
- Manuelles (zeilenweise) Abarbeiten mit Strg + Leertaste

Debugging:

bei Fehlern lässt sich im Editor der fragliche Quellcode begutachten und korrigieren.

Für eine weitere Excel/VBA-Nutzung muss man den Fehler beheben oder die Ausführung des Makros **stoppen (Stoptaste im Editor)**.

Symbolleiste „Bearbeiten“ einfügen lassen im VBA-Editor:

- Einzüge
- Haltepunkte
- Sub/Makro (F5)
- Kommentierungen

Einstellungen:

- für Tab-Schrittweite (z.B. 2 statt 4 Leerzeichen)
- Schrift/Schriftgröße oder
- automatische „Option Explicit“ für neue Module

Codeeingaben gerne in Small (kleinen Zeichen) und ohne Leerzeichen:

```
worksheets("Tabelle1").range("a1").value=100  
' wird zu  
Worksheets("Tabelle1").Range("a1").Value = 100
```

Aber auch hier ein Hinweis: bitte den erstellten Code immer kontrollieren/checken!

9.2 Direktfenster

Tastenkombination: Strg + G

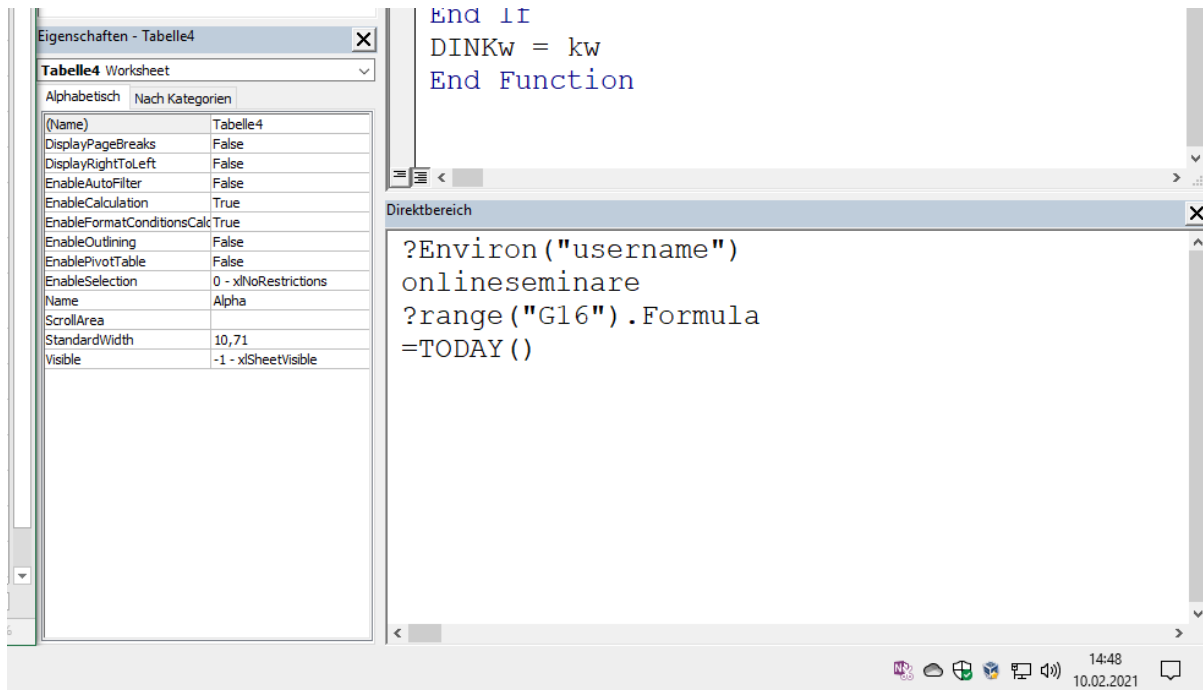


Abb. 9.1: Direktfenster (Strg + G)

Beispielhafte Direktaufrufe:

```
?Environ("username")
?Environ("computername")
```

Übung - Funktionsnamen im Englischen Original ermitteln:

Wie heißt die Funktion MITTELWERT() in Zelle A1?

```
?range("A1").Formula
=AVERAGE(D1:D8)
```

Tipp: Formel in Zelle eintragen:

Tabelle3.Range(„A1“).Formula = „=AVERAGE(D1:D8)“

Hinweis: FormulaLocal hat dann die länderspezifische Benennung!

9.2.1 Berechnungen

Grundeinstellungen über Optionen: automatisch / manuell

Anm.: SAP-Excel-Exports sind oft manuell eingestellt, sodass Excel diese Blätter nicht automatisch aktualisieren!

Aufrufe auch gerne auch im Direktfenster:

```
Application.Calculation=xlCalculationManual
Application.Calculation=xlCalculationAutomatic
```

9.2.2 Debug Ausgaben

z.B. zur Laufzeitmessung:

```
Debug.Print "Start: " & Now
....
Debug.Print "Ende: " & Now
```

oder einfach manuell einen Wert / Variable ausgeben!

9.3 Eigenschaften

Tastenkombination: F4

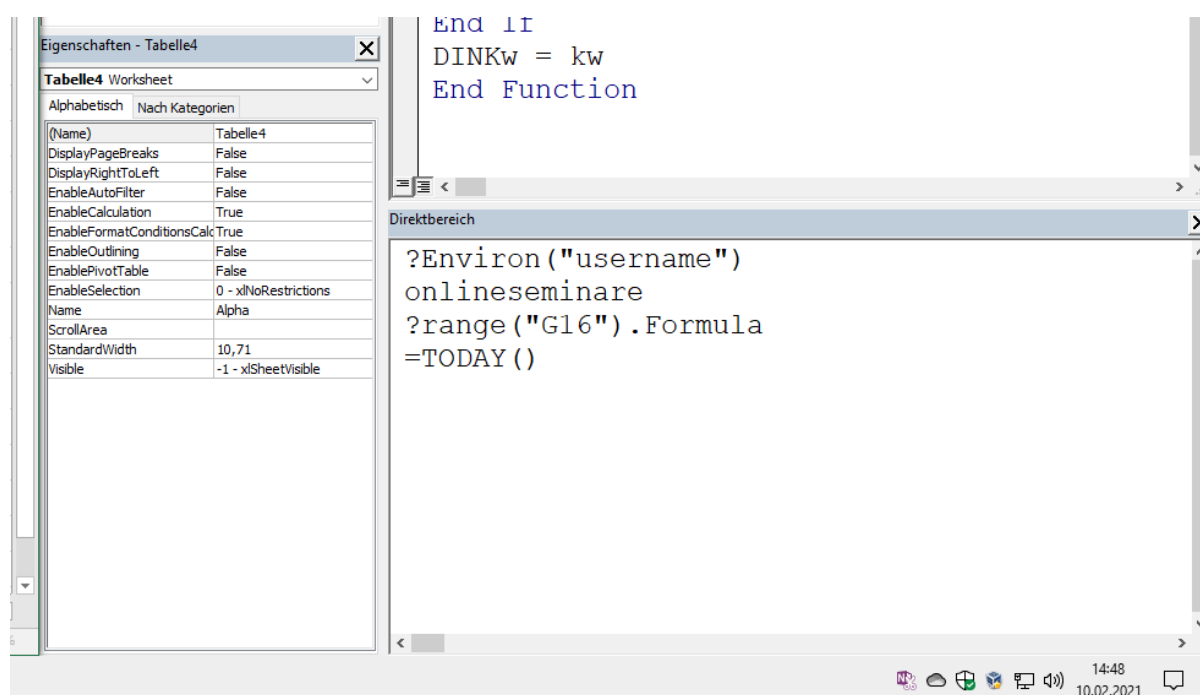


Abb. 9.2: Eigenschaften (F4)

Übung: Sichtbarkeiten von Tabellenblättern manipulieren

- **-1** / xlSheetVisible : Die Tabelle ist sichtbar.
- **0** / xlSheetHidden : Die Tabelle ist ausgeblendet.
- **2** / xlSheetVeryHidden : Die Tabelle ist sicher ausgeblendet, das heißt, der normale Anwender kann diese Tabelle nicht mehr einblenden.

Siehe auch: **ScrollArea**

9.4 Variablen

VBA Variablen werden mit dem Wort Dim deklariert.

Tipp: Bitte Sonderzeichen und natürlich *Leerzeichen* ;-) vermeiden.

lowerCamelCase Schreibweise empfohlen: `intBeispielVar`

Es wird die Nutzung eines Prefix empfohlen - hier: **int** für Integer-Variable.

Bitte die Variablen am Anfang der Sub-Routinen deklarieren (lokale Variablen).

Beispielhafte **Prefixes** für die Datentypen:

- bool Boolean
- byte Byte
- cur Currency
- dtm Date
- dec Decimal
- dbl Double
- int Integer
- lng Long
- obj Object
- sng Single
- str String
- var Variant
- rng Range
- wkb Workbook

Speziell: Typ Enum

Übersicht zu Variablentypen: (Beachten: VB weicht teils von VBA ab; z.B. decimal not supported, ...)

Beispielhafte Links:

- <https://docs.microsoft.com/en-us/office/vba/language/reference/user-interface-help/data-type-summary>
- <https://docs.microsoft.com/en-us/dotnet/visual-basic/language-reference/data-types/>
- <https://docs.microsoft.com/de-de/office/vba/language/reference/user-interface-help/dim-statement>
- <https://www.vba-tutorial.de/variablen/datentypen.htm>

Anm.: nur eine kleine Linkauswahl ohne Anspruch auf Vollständigkeit ;-)

Statische/Static Variablen bleiben nach Laufzeitende erhalten!

Quellcode 9.1: Static Definition für Variable

```
Sub VariablenInhaltBleibtBestehen()  
Static lngAufrufe As Long  
  
    lngAufrufe = lngAufrufe + 1  
    MsgBox "Makro wurde " & lngAufrufe & " Mal ausgeführt!"  
  
End Sub
```

Öffentliche/Public Variablen sind auch über Modulgrenzen hinweg nutzbar!

Quellcode 9.2: Sichtbarkeit über Modulgrenzen

```
' eigenes Modul 01  
Option Explicit  
  
Public dteStartzeit As Date  
  
Sub StartZeitSetzen()  
    dteStartzeit = Now  
End Sub  
  
' eigenes Modul 02  
Option Explicit  
  
Sub StartzeitAbfragen()  
    MsgBox dteStartzeit  
End Sub
```

Private/Private Variablen sind nur in ihrem Modul verfügbar!

```
Private strPasswort As String
```

Und wenn wir das Signalwort **Private** auf die Sub-Routinen anwenden erhalten wir die Möglichkeit den Code in Sichtbarkeit der **Makro Anzeige** zu beeinflussen.

Nur die letzte Routine ist in Excel Makros sichtbar!

Quellcode 9.3: Nutzung von Private für Sub Routinen

```
Option Explicit  
Private umfang As Single  
Private laenge As Single  
Private breite As Single  
Private Const faktor = 2  
  
Private Sub BerechneUmfang()  
    ' Umfang eines Rechtecks nach der Formel  $u = 2 \cdot (a+b)$  bestimmen  
    umfang = faktor * (laenge + breite)
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

End Sub

Private Sub EingabeDialog()
    ' Eingabewerte ermitteln
    laenge = InputBox("Bitte geben Sie die Länge des Rechtecks ein: ", "Eingabe
↳", 10)
    breite = InputBox("Bitte geben Sie die Breite des Rechtecks ein: ",
↳"Eingabe", 5)
End Sub

Sub BerechnungUmfang()
    ' startet den Eingabedialog und die Berechnung
    EingabeDialog
    BerechneUmfang
    MsgBox "Der Umfang des Rechtecks beträgt " & umfang & " Meter.",
↳vbInformation, "Ausgabe"
End Sub

```

9.5 Konstanten

VBA-Konstanten sind schneller als Variablen!

```

Const strArbeitsmappe as String = "Mappel.xls"
Const dteStartDatum As Date = #1/1/2015#
Const strFehlermeldung as String = "Fehler beim Druck!"
Const sngMWST as Single = 1.19

```

Auch hier wie bei Variablen bitte möglichst den Typ definieren.

9.6 Kontrollstrukturen

Und hier könnte man jetzt richtig für *Buchseiten* sorgen. Im Grunde sollte man für alle Grundkonstrukte einer Programmiersprache (hier: Skriptsprache) die üblichen Code-Gerüste parat haben.

Die Frage ist nur, ob man sich hier ausführliche Beispiele hinterlegt, oder einfach der Code Vervollständigung des Wahl-Editors folgt und ja doch oft die Codes fertig via *Copy & Paste* bereitstellt.

Wir schauen uns das mal *grob* an und schauen dann im VBA-Editor weiter.

9.6.1 Verzweigungen

Je nach Zeit werden wir in den Seminare die hier notwendigen Beispiele mehr oder weniger intensiv durchgehen.

- If ... End If
- If ... Then ... Else ... End If
- If(Bedingung, Then, Else)
- Select Case ... Case(s) ... End Select

Übungen: Beispiel-Codes für TN werden von mir bereitgestellt.

9.6.2 Schleifen

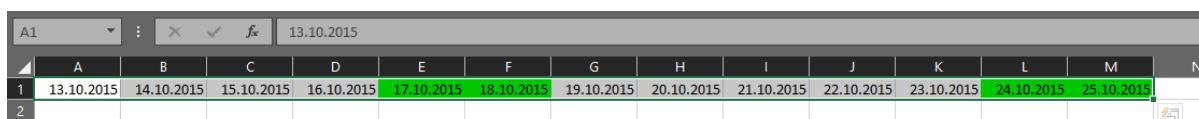
- For ... To ... Next
- For Each ... Next
- With ... End With
- Do Until ... Loop
- Do While ... Loop

Übungen: Beispiel-Codes für TN werden von mir bereitgestellt.

Aber ein wenig soll auch an dieser Stelle dargestellt werden.

9.6.3 Einfache Codes Kontrollstrukturen

Wochenenden kennzeichnen



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	13.10.2015	14.10.2015	15.10.2015	16.10.2015	17.10.2015	18.10.2015	19.10.2015	20.10.2015	21.10.2015	22.10.2015	23.10.2015	24.10.2015	25.10.2015	
2														

Abb. 9.3: Codebeispiel - Wochenenden kennzeichnen

Hier ein der Beispielcode (mit With ... End With):

Quellcode 9.4: Beispielcode With - Wochenenden kennzeichnen

```
Sub WochenendenKennzeichnen()  
Dim lngSpalte As Long  
Dim lngSpalteMax As Long  
  
With Tabelle4  
  
    .Rows(1).Interior.ColorIndex = xlColorIndexNone  
    lngSpalteMax = .Cells(1, .Columns.Count).End(xlToLeft).Column  
  
    For lngSpalte = 1 To lngSpalteMax
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

If IsDate(.Cells(1, lngSpalte).Value) Then

    If Weekday(.Cells(1, lngSpalte).Value, vbMonday) > 5 Then
        '.Cells(1, lngSpalte).Interior.ColorIndex = 4
        '.Cells(1, lngSpalte).Interior.Color = RGB(0, 0, 255)
        .Cells(1, lngSpalte).Interior.ThemeColor = xlThemeColorAccent1
    End If

End If

Next lngSpalte

End With

End Sub

```

Im Code kann man drei unterschiedliche Farbzweisungen erkennen. Hier ist die letzte Variante sicherlich die Professionellste, da wir die gewünschten Designfarben der Office-Umgebung nutzen (hier: Akzentfarbe 1).

Werte in Spalten suchen

Der Code soll einen Wert in einer Spalte suchen, finden und hervorheben.

	A	B	C
1	schlüssel		
2	4711		
3	4712		
4	4713		
5	4714		
6	4715		
7	4716		
8	4717		
9	4718		
10	4719		
11	4720		
12	4721		
13	4722		
14	4723		
15	4724		
16	4725		
17	4726		
18	4727		
19	4728		
20	4729		
21			
22			
23			
24			
25	4720		
26			
27			
28			
29			
30			
31			
32			

Abb. 9.4: Code Beispiel - Werte in Spalte suchen

Hier folgt der VBA-Code zur Umsetzung.

Quellcode 9.5: Wert in Spalte finden und hervorheben

```

' Variante: JoeB

Sub WerteInSpalteSuchen()
    Dim rngTreffer As Range
    Dim strNoTreffer As String
    Dim strSuchbegriff As String

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

Dim lngZeile As Long
Dim lngZeileMax As Long

Tabelle1.Range("A:A").Interior.ColorIndex = xlColorIndexNone
strSuchbegriff = InputBox("Suchbegriff eingeben!", "Direktsuche", 4720)

With Tabelle1
    lngZeileMax = .UsedRange.Rows.Count

    For lngZeile = 2 To lngZeileMax

        Set rngTreffer = Tabelle1.Range("A" & lngZeile).
↪Find(What:=strSuchbegriff, LookIn:=xlValues, LookAt:=xlWhole)

        If rngTreffer Is Nothing Then
            temp = Tabelle1.Range("A" & lngZeile).Address
            strNoTreffer = strNoTreffer & " " & temp
        Else
            rngTreffer.Interior.ColorIndex = 4
        End If

    Next lngZeile

End With

MsgBox "Wert in den Zellen " & strNoTreffer & " nicht gefunden!"

End Sub

```

9.7 Objektorientierung

Excel-VBA-Objekte folgen Grundmechanismen der Objektorientierten Basistechniken. Folglich haben VBA-Objekte

- Eigenschaften (Property - 80%) und
- Methoden (Methods - 20%)

Übung/Test mit Objekt **Application**:

```

1 Sub UnterschiedMethodeUndEigenschaft()
2 ' unterschiedlichen Zuweisungen mit = und := beachten
3
4 With tbl_Matrix.Range("A1:D10")
5 ' Eigenschaft
6 .Borders(xlInsideHorizontal).ColorIndex = 4
7 ' Methode
8 .BorderAround Weight:=xlThick, ColorIndex:=3

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
9      End With
10
11 End Sub
```

Bitte die unterschiedlichen Zuweisungen mit = und := beachten.

Bitte auch immer an **Code-Completion** mit Strg + Leertaste denken und die automatischen Auswahllistenfelder nutzen.

9.8 Workbook_Open()

Wir wollen Routinen direkt beim Laden einer Arbeitsmappe automatisch ausführen lassen.

Quellcode 9.6: Private Sub Workbook_Open()

```
Private Sub Workbook_Open()  
    'Dieses Ereignis wird beim Öffnen der Mappe ausgeführt  
    Tabelle1.Range("A1").Value = 100  
    'Die ScrollArea für Tabelle1 setzen  
    Tabelle1.ScrollArea = "A1:D10"  
End Sub
```

Wichtig: Dieser Code gehört jetzt mal nicht in die Module, sondern sozusagen wortwörtlich zu „Diese Arbeitsmappe“.

Eine solche Technik gibt es auch als Standardtechnik für Module einer Mappe.

Auto_Open() als Modul

Das folgende Macro (in einem Modul einer Arbeitsmappe) wird ebenfalls automatisch ausgeführt.

Quellcode 9.7: Auto_Open()

```
Sub Auto_Open()  
'  
' Auto_Open Makro  
'  
'  
ActiveCell.FormulaR1C1 = "Hallo"  
End Sub
```

In der Zelle des *letzten Speicherns* wird der Text **Hallo** eingefügt.

9.9 Tabellen ansprechen

... die Tabellennamen mit unterschiedlicher Ansprache ...

Die Tabellenblattnamen lassen sich in der Blattansicht und über das Eigenschaftenfenster ändern.

- **Codename:** Tabelle3
- **Normaler Name:** tbl_Daten (hier: Objekt Worksheets nutzen)

Empfehlung (bzw. Best Practise): **Codename**

Gründe:

- wegen besserer Code/Syntaxcompletion
- sicherer
- weniger Pflegeaufwand

Varianten zum Ansprechen eines Tabellenblatts:

Quellcode 9.8: Varianten für Tabellennamen

```
' Codename:  
Tabelle3.Range("A1").Value = 100  
' Vollqualifiziert mit Worksheets  
Worksheets("tbl_Daten").Range("A4").Value = 10  
' Vollqualifiziert mit Worksheets und Index  
Worksheets(3).Range("A2").Value = 10  
' teilweise qualifiziert - Sheets arbeitet auch mit Diagrammblättern  
Sheets("tbl_Daten").Range("A5").Value = 10  
' völlig ungenau/gefährlich  
Range("A6").Value = 10
```

Conclusio/Empfehlung: Tabellenblätter genau planen und strukturieren!

Zum Abschluss noch ein kleines Beispiel für das Handling und Erstellen von Tabellenblättern.

Quellcode 9.9: Tabellenblätter zählen und hinzufügen

```

Sub TabellenblaetterZaehlen()
    Dim Anzahl
    Anzahl = ActiveWorkbook.Worksheets.Count
    ActiveWorkbook.Worksheets.Add
    MsgBox "Die Anzahl der Tabellenblätter hat sich von " & Anzahl & " auf " _
        & ActiveWorkbook.Worksheets.Count & " erhöht.", , "1. Meldung"
    Anzahl = ActiveWorkbook.Worksheets.Count
    ActiveWorkbook.Worksheets.Add
    MsgBox "Die Anzahl der Tabellenblätter hat sich von " & Anzahl & " auf " _
        & ActiveWorkbook.Worksheets.Count & " erhöht.", , "2. Meldung"
End Sub

```

Am Besten würden die neuen Tabelleblätter auch noch gleich mit gewünschten Tabellenblattnamen ausgestattet und sollte an einer spezifischen Stelle in unserer Arbeitsmappe auftauchen.

Quellcode 9.10: Beispiele für neue Tabellenblätter einfügen

```

' Aktives Tabellenblatt benennen:
ActiveSheet.Name = "NewSheet"
' Name beim Erstellen definieren
Sheets.Add.Name = "NewSheet"
' Name für Tabellenblatt aus einer Zelle beziehen
Sheets.Add.Name = range("a3").value
' neues Tabellenblatt NACH Blatt Test mit Name TestNEU
Sheets.Add(After:=Sheets("Test")).Name = "TestNEU"
' oder auch VORHER
Sheets.Add(Before:=Sheets("Test")).Name = "TestNEU"
' am Ende neu einfügen
Sheets.Add After:=Sheets(Sheets.Count)
' am Anfang neu einfügen
Sheets.Add(Before:=Sheets(1)).Name = "FirstSheet"

```

Weiter Beispiel im Netz z.B. unter [Automateexcel - Add and name worksheets](https://www.automateexcel.com/vba/add-and-name-worksheets/)²⁶. Und das ist natürlich nur ein Beispiel von vielen - einfach mal mit unserem aufgebauten Wissen um die Techniken selber recherchieren.

Und wie immer: bitte nicht von den teils *altbackenen* Websites der VBA-Profis abschrecken lassen. In deren Community scheinen wohlgeformte oder gar moderne Webdesigns nicht zu interessieren ;-).

²⁶ <https://www.automateexcel.com/vba/add-and-name-worksheets/>

9.10 VBA Beispiele

VBA per Copy & Paste bereitstellen.

Eine Idee für unsere Einführung in die Automatisierungen mit Makros und VBA-Code ist die praktische Umsetzung „quasi fertiger“ VBA-Lösungen aus diversen Quellen:

- Fachbücher
siehe Herdt-Skript, VBA Buch Rheinwerk-Verlag (über 600 Beispiele), ...
- Online-Recherchen
neben der reinen Suche natürlich auch diverse Fachportale zu VBA

Hier mal ein paar Beispiele für diesen **Copy & Paste** Einsatz.

9.10.1 VBA Beispiel 01

Mehrzeiliger Code mit „_“ als Zeilenumbruch:

```
' Über mehrere Zeilen mit Leertaste _ (in MsgBox Konstante vbCrLf)
Sub MsgBoxMehrzeiligAnzeigen()
    MsgBox "Heute ist der " & Date & vbCrLf & _
        "Es ist jetzt genau: " & Time & vbCrLf & _
        "Sie sind der Anwender: " & Environ("username"), _
        vbOKOnly, "Meldung"
End Sub
```

Tipp: bei Code mit Vorgestellten Optionen bitte jeweils ein einzelnes Modul nutzen (siehe: Option Explicit)!

9.10.2 VBA Beispiel 02

Hier folgend ein Beispiel einer Internet-Recherche mit „excel VBA tabellenblätter sortieren“ ergibt aktuell mit Google gut 122.000 Suchergebnisse!

Darunter findet sich (z.B.) auch eines der erwähnten Portale rund um Excel:

<https://www.excel-inside.de/vba-loesungen/datei/951-tabellenbl%C3%A4tter-alphabetisch-sortieren>

Und schon hat man ein „Copy & Paste“ Beispiel für die Sortierung von Tabellenblättern:

```
1 Sub Sheets_alphabetisch_sortieren()
2   *** Dieses Makro sortiert die Blätter in alphabetischer Reihenfolge
3   *** Reihenfolge: Sonderzeichen, Ziffern, Buchstaben
4
5   *** Dimensionierung der Variablen
6   Dim intAnz As Integer
7   Dim a, b As Integer
8   Dim strSortKrit As String
9
10  *** Sortierreihenfolge festlegen
11  *** "<" = Aufsteigend von 0-9 und A-Z
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

12  '** ">" = Absteigend von Z-A und 9-0
13  strSortKrit = ">" 'Nach Bedarf anpassen!
14
15  '** Ermittlung Anzahl Tabellenblätter
16  intAnz = ActiveWorkbook.Worksheets.Count
17
18  '** Blätter nacheinander durchlaufen und sortieren
19  For a = 1 To intAnz
20      For b = a To intAnz
21          If strSortKrit = "<" Then
22              If Worksheets(b).Name < Worksheets(a).Name Then
23                  Worksheets(b).Move Before:=Worksheets(a)
24              End If
25          ElseIf strSortKrit = ">" Then
26              If Worksheets(b).Name > Worksheets(a).Name Then
27                  Worksheets(b).Move Before:=Worksheets(a)
28              End If
29          End If
30      Next b
31  Next a
32
33  End Sub

```

Den Code einfach in ein Modul Ihrer Wahl kopieren und austesten!

Und diese beispielhafte Suche kratzt nur an der Oberfläche der möglichen Umsetzungen - wie ich immer sage: „Das Rad wurde schon erfunden“. Was in diesem Zusammenhang heißen soll, dass es bereits Lösungen für Ihre Probleme gibt.

Es liegt an uns diese Beispiel sauber mit Excel-Technik umsetzen zu können.

9.11 UserForms

Nutzung von UserForms:

Hier ist es wichtig, dass die aktiven Elemente / VBA-Codes den Elementen des Formulars zugeordnet werden.

An den Code kommt man per Doppelklick auf User Form Elemente.

Das eigentliche Formular kann durch eine einfache Schaltfläche Arbeitsblatt per Klick aufgerufen!

Beispiele für Sub / Prozeduren:

- cmdMulti_Click() oder
- txtZahl1_Exit()

Es wird also immer der Name eines Form-Steuerelements und ein angehängtes Ereignis als Name benutzt!

Hier mal zwei Beispiele für UserForms.

9.11.1 Herdt-Taschenrechner

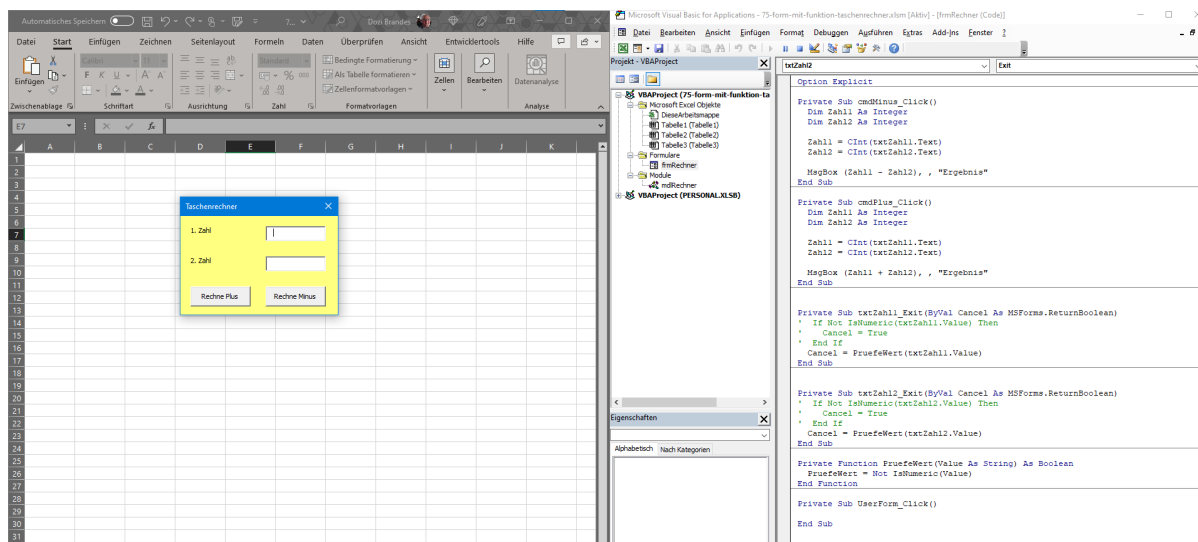


Abb. 9.5: User Form Beispiel Taschenrechner (Herdt Verlag)

Einbau einer neuen Schaltfläche (CommandButton cmdMultiplikation) und Code für eine Multiplikationsfunktion des Taschenrechners.

9.11.2 Eingabemasken mit Excel

Für weitere vertiefende Beschäftigungen mit dem Thema „UserForms“ aus einer beispielhaften Anleitung im Internet:

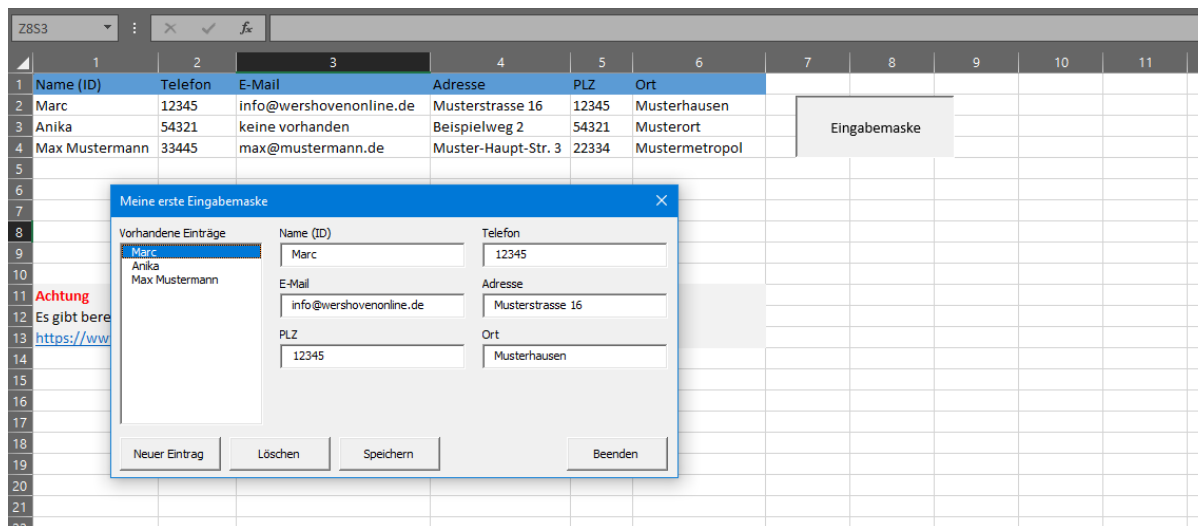


Abb. 9.6: User Form Beispiel Eingabemaske (online-vba.de)

VBA Tutorials - Eingabemasken mit Excel²⁷

Download der Trainingsdateien ist hier am Ende des Beitrags verfügbar.

²⁷ https://www.online-vba.de/vba_tutorialuserform.php

9.12 Eigene Funktion mit VBA

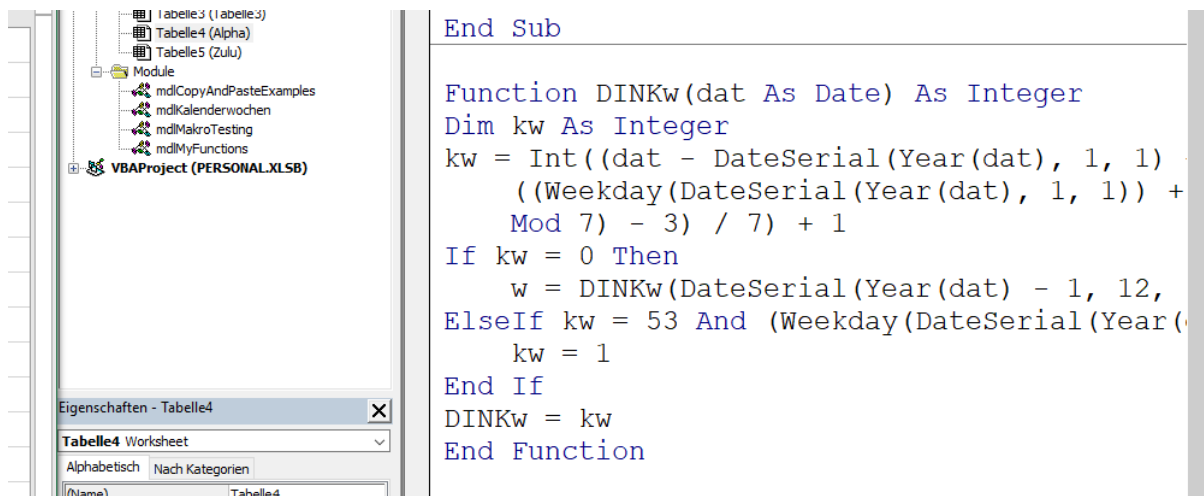


Abb. 9.7: Eigene Funktionen mit VBA

Mit ein wenig function VBA-Code lassen sich eigene Funktionen deklarieren, die man dann in Excel wie die eingebauten Funktionen nutzen kann.

```

Public Function NettoMwst(Betrag, Optional SteuerSatz As Single = 0.19)
'Quelle: Helmut Pirklbauer
Dim Netto As Double
Netto = Betrag / (1 + SteuerSatz)
NettoMwst = Excel.Application.Round(Netto, 4)
End Function

```

Hinweis: diese *functions* müssen sich bis auf Weiteres in den Modulen von Arbeitsmapen befinden (Anm.: also nicht in der personal.xlsb - der persönlichen Arbeitsmappe).

9.13 Textdateien einlesen

Übungen zu unterschiedlichen Varianten

```

....

Open ConstTxtDatei For Input As #1

Do Until EOF(1)
    Line Input #1, strText
    Debug.Print strText
Loop

Close #1

....

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
Open ConstTxtDatei For Input As #1
  VarDat = Input$(LOF(1), 1)
  Debug.Print VarDat
Close #1
```

Die Zweite Variante ist schneller, da man keine VBA-Schleife (Loop) nutzt.

KAPITEL 10

Was mit Medien

Für das SoSe 2022 hatte ich eine Anfrage für eine Teilveranstaltung im Rahmen der Serie **Was mit Medien**.

[Link zur Veranstaltungsreihe / Infos auf TU BS Website²⁸](#)

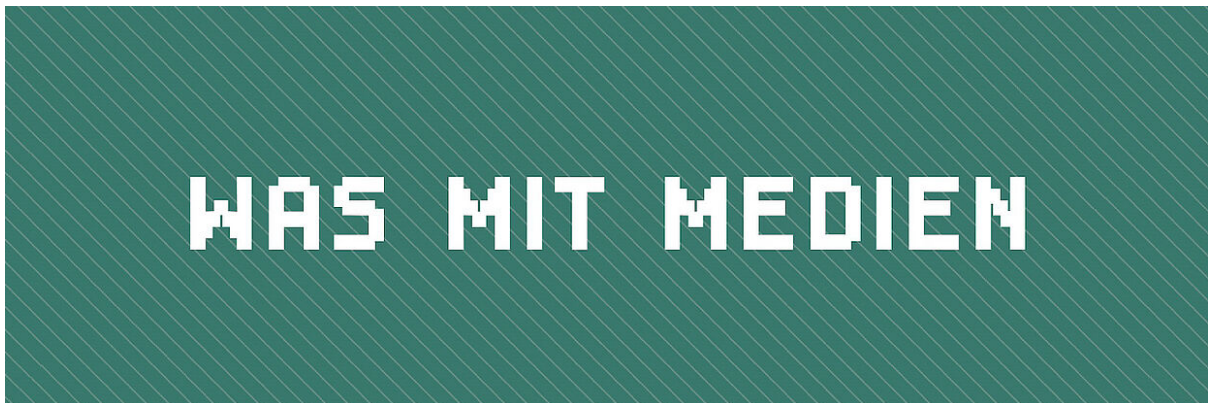


Abb. 10.1: Logo - Was mit Medien (siehe TU BS Website Link)

Für die nachhaltige Bereitstellung und die Ausarbeitung der Inhalte folgen weitere Abschnitte.

Die Seminarblöcke werden (meist) mit klassischen Präsentationen (pptx) und Seminarbeiträgen (per PDF) komplettiert.

²⁸ <https://www.tu-braunschweig.de/lehreundmedienbildung/angebote/angebote-fuer-studierende>

10.1 Seminar

Mein **90-minütiger Beitrag** zur Online-Reihe soll in einem Fachvortrag zur Verarbeitung von Daten mit Tabellenkalkulationstechnik bestehen.



Abb. 10.2: Was mit Medien - Onlinereihe

Der Vortrag schließt mit einem **Arbeitsauftrag** (Homework) für die Trainees ab, welcher vom **TU BS WMM Team** bewertet und gegebenenfalls von mir noch fachlich beleuchtet werden kann.

Hierbei sollte ich mich (möglichst) nach *Vorlage* des vorherigen Referenten orientieren.

Lernziele

Das führte zu den Schwerpunkten der **Datenverarbeitungsbasistechniken**:

- Import von Daten (Formate)
- Daten darstellen
- Daten ändern
- Export von Daten (Formate)

Arbeitsauftrag nach Seminar

Als Abschluss dieses Seminar-Blocks sollen die Trainees eine praktische Arbeit (**Arbeitsauftrag**) nach Vorgabe einer Problemstellung beim WMM-Team der TU Braunschweig einreichen.

Vermittlung Lehrgangsstoff

Die Kollegen nehmen für die Vermittlung in den Seminaren natürlich gerne PowerPoint und Co - und sehr wahrscheinlich kopiere ich mir auch noch *schnell so eine Foliensammlung zusammen*.

Allerdings bereite ich meine Unterlagen und Daten ja seit langer Zeit mit RST-Dokumenten auf. Also habe ich mich entschieden die Ausarbeitungen hier in gewohnter Weise bereitzustellen.

10.2 Tabellenformate

Damit sind hier alle möglichen Dateiformate für die Speicherungen und Verarbeitungen von *tabellarischen Informationen* gemeint.

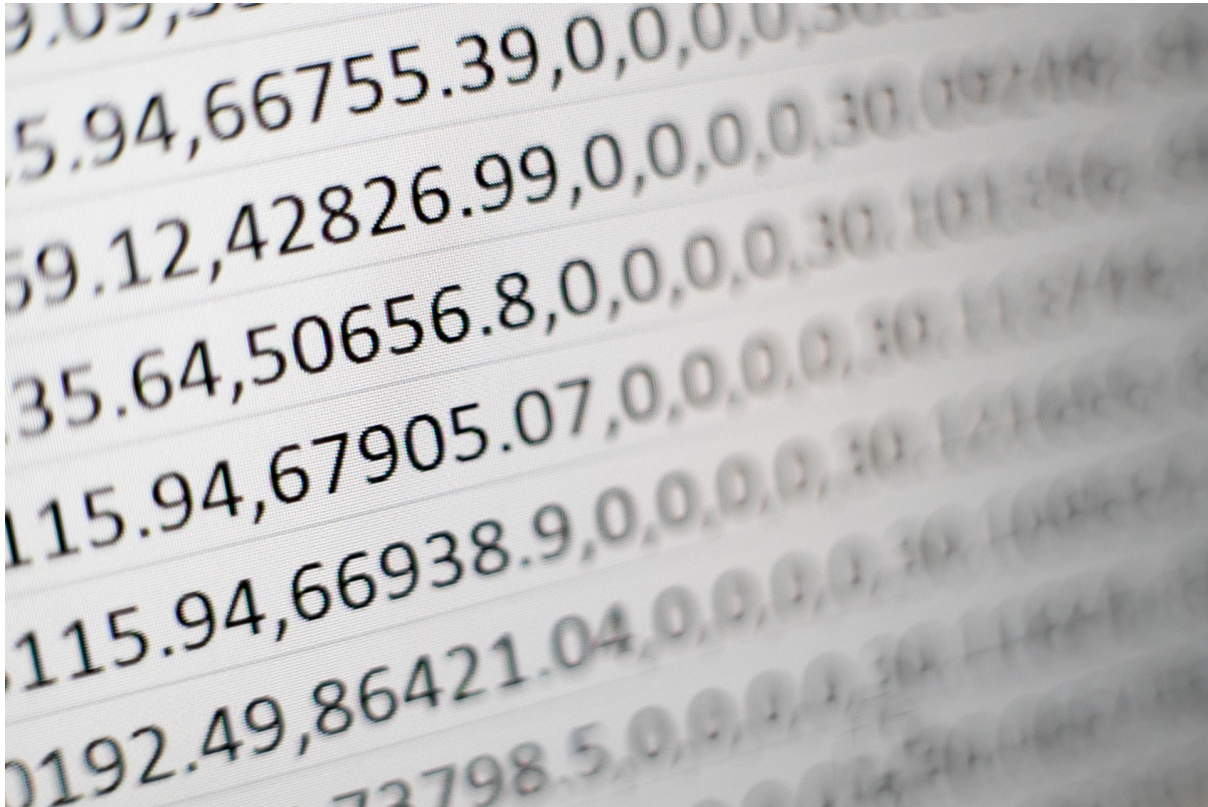


Abb. 10.3: Tabellarische Datenliste - Bild: mika-baumeister-Wpnoqo2plFA-unsplash.jpg

Hier stehen natürlich erst einmal die **IT-/Office-Industriestandards** der Excel Dateiformate im Mittelpunkt.

10.2.1 Excel-Dateiformate

Wir starten mit einer kurzen Erinnerung und Ausführung zu den Excel-Dokumentformate:

Tab. 10.1: Excel Dateiformate

Versionen	Dateiformate
Excel 97-2003	xls - das klassische Excel-Sheet (Arbeitsmappen) xlt - die passenden Vorlagen (engl.: Templates)
Excel 2007/2010/...	xlsx - XML basiertes Dokumentformat (ohne Makros) xltx - die passenden Vorlagen xlsm - XML basierte Mappen inkl. Makros xltm - die passenden Vorlagen inkl. Makros xlsb - spezielles Binärformat (siehe PERSONAL.xlsb)

Die *fett dargestellten* Formate **xlsm**, **xltm** und **xlsb** unterstützen aktive Inhalte (siehe **Makros** oder auch ActiveX) und sollten besonders betrachtet werden.

Die *Legacy* Formate xls und xlt sollten nicht mehr genutzt werden, da sie proprietäre und *nicht offene* Formate darstellen.

Die modernen Formate (mit x wie XML seit 2007) sind ein **offenes Dokumentformat** und stellen technisch **Zip-Archive** mit klarer Dokumentstruktur dar.

Tipp: Mit Packer-Tools wie **7-Zip** kann man eine **xlsx** einfach in einen Dateiordner entpacken. In den enthaltenen Dateien erkennt man dann die interne **XML** Technik - siehe das **x** am Ende von xls **x**.

Diese im Grunde *offen* Dokumentbeschreibung kann von anderen Programmen und Tools angewandt werden. Die Kompatibilität zu den Microsoft-Formate wird allerdings niemals gänzlich erreicht.

10.2.2 CSV

Der Klassiker in der IT für die Ablage von tabellarischen Informationen mit **Comma-separated Values**.

Das Format unterstützt nur einfache 2-dimensionale Datenstrukturen.

Außer einer Internet-Norm (RFC 4180) gibt es keine weiteren Festlegungen für Trennzeichen (Delimiter), Zeichenkodierungen oder die Behandlung von Spaltenüberschriften (Headings).

```
Stunde, Montag, Dienstag, Mittwoch, Donnerstag, Freitag
1, Mathematik, Deutsch, Englisch, Erdkunde, Politik
2, Sport, Deutsch, Englisch, Sport, Geschichte
3, Sport, "Religion (ev., kath.)", Kunst, , Kunst
```

Beispiel: [Wikipedia CSV](#)²⁹

Kommandozeilentools und Online-Ressourcen zu CSV:

²⁹ [https://de.wikipedia.org/wiki/CSV_\(Dateiformat\)](https://de.wikipedia.org/wiki/CSV_(Dateiformat))

- **xsv** - Kommandozeilentool für csv (<https://github.com/BurntSushi/xsv>), welches für diverse Betriebssysteme verfügbar ist
- **csvkit** - eine Sammlung von CSV-Tools: Github csvkit³⁰, Read the Docs csvkit³¹
- [Online CSV Editor and Viewer](https://www.convertcsv.com/csv-viewer-editor.htm)³²
- [Online CSV Tools](https://onlinecsvtools.com/)³³

Python Spezial

Auch bei der Python Skriptverarbeitung wird sehr häufig zu **CSV** als Datendatei gegriffen. Allerdings hat diese Community sich mit vielen Techniken beschäftigt und so auch für Datendateien *moderne* Wege gefunden.

Python CSV-Alternativen: Pickle, Parquet, Feather

Die hier genannten Dateikandidaten wurden im Youtube-Video

- [Medallion Data Science: The BEST way to store data in python & pandas.. spoiler: it's not CSV](https://www.youtube.com/watch?v=u4rsA5ZiTls)³⁴ genauer analysiert.

Hier mal ein Screenshot mit Dateigrößen:

```
[86]: !ls -lGFlash test.csv test.pickle test.parquet test.feather
215M -rw-rw-r-- 1 robmulla 215M Apr  2 14:41 test.csv
 49M -rw-rw-r-- 1 robmulla  49M Apr  2 14:45 test.feather
 33M -rw-rw-r-- 1 robmulla  33M Apr  2 14:45 test.parquet
 82M -rw-rw-r-- 1 robmulla  82M Apr  2 14:45 test.pickle
```

Abb. 10.4: Python Dataformats

Die Lesegeschwindigkeit in diesen Test war für das Einlesen von einer Test-CSV bei 39,3 s und verringerte sich bei den CSV-Alternativen in einen Bereich von 100 bis 500 ms (Millisekunden)!

Für Techniker, die mit großen (größten Datensätzen) zu tun haben, sollte sich also ein Blick auf die alternative Tabellen/Datenauswertung über Python lohnen.

10.2.3 JSON

Mit der **JavaScript Object Notation** stehen mehr Möglichkeiten zur Strukturierung zur Verfügung. Auch die technische Umsetzung von der Erstellung über die Weitergabe bis zur Verarbeitung ist sehr offen und vielseitig.

Hier ein JSON-Beispiel:

³⁰ <https://github.com/wireservice/csvkit>

³¹ <https://csvkit.readthedocs.io/en/latest/index.html>

³² <https://www.convertcsv.com/csv-viewer-editor.htm>

³³ <https://onlinecsvtools.com/>

³⁴ <https://www.youtube.com/watch?v=u4rsA5ZiTls>


```
{
  "Herausgeber": "Xema",
  "Nummer": "1234-5678-9012-3456",
  "Deckung": 2e+6,
  "Waehrung": "EURO",
  "Inhaber":
  {
    "Name": "Mustermann",
    "Vorname": "Max",
    "maennlich": true,
    "Hobbys": ["Reiten", "Golfen", "Lesen"],
    "Alter": 42,
    "Kinder": [],
    "Partner": null
  }
}
```

Beispiel: Wikipedia JSON³⁵

Auch für JSON gibt es gute Kommandozeilentools oder Online-Ressourcen:

- [jq - Command Line Tool jq](#)³⁶
- [jless - Command Line Tool jless](#)³⁷
- [20 Online Tools für JSON \(inkl. Erläuterungen\)](#)³⁸
- [Beispiel: JSON Editor Online](#)³⁹

Und natürlich kann man auch noch mit grafischen Oberflächen online experimentieren und visualisieren

- [Jsoncrack - Editor](#)⁴⁰

³⁵ https://de.wikipedia.org/wiki/JavaScript_Object_Notation

³⁶ <https://stedolan.github.io/jq/>

³⁷ <https://jless.io/>

³⁸ <https://geekflare.com/de/json-online-tools/>

³⁹ <https://jsoneditoronline.org/>

⁴⁰ <https://jsoncrack.com/editor>

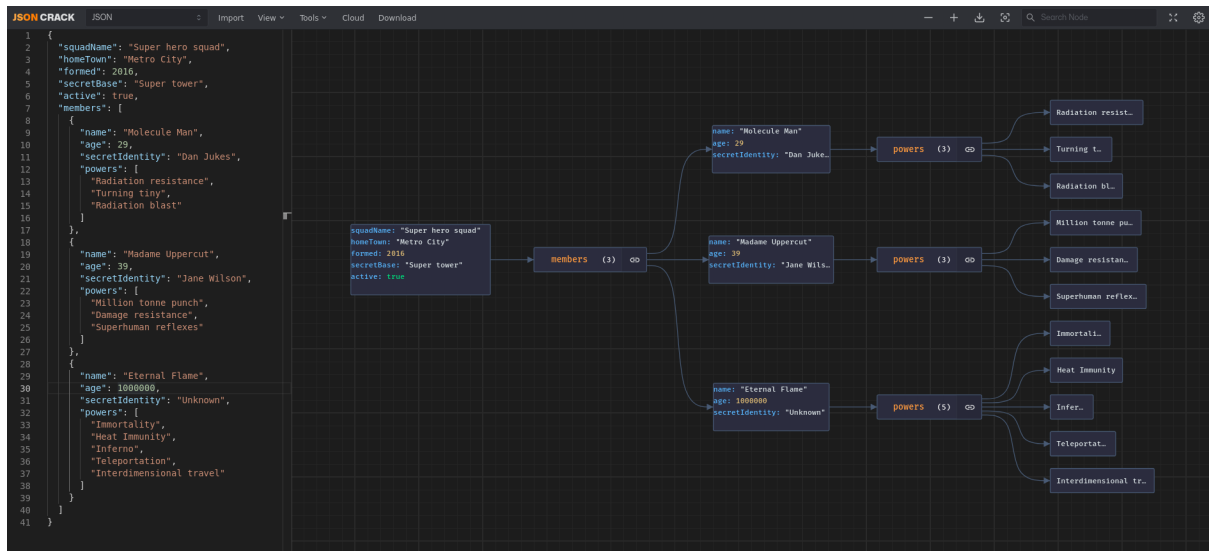


Abb. 10.5: Jsoncrack - Editor

Man kann den eigenen Json-Code einfach in das linke Fenster kopieren und sich dann die Technik und Hierarchie genauer anschauen.

Die Knotenpunkte kann man per Klick analysieren.

10.2.4 XML

Die **Extensible Markup Language** ist die umfassendste und mächtigste Auszeichnungssprache für strukturierte Datendateien.

Der Versuch einer Darstellung sprengt diesen Rahmen.

Beispiel einer XML-Datei:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<verzeichnis>
  <titel>Wikipedia Städteverzeichnis</titel>
  <eintrag>
    <stichwort>Genf</stichwort>
    <eintragstext>Genf ist der Sitz von ...</eintragstext>
  </eintrag>
  <eintrag>
    <stichwort>Köln</stichwort>
    <eintragstext>Köln ist eine Stadt, die ...</eintragstext>
  </eintrag>
</verzeichnis>

```

Beispiel: Wikipedia XML⁴¹

⁴¹ https://de.wikipedia.org/wiki/Extensible_Markup_Language

10.3 Import von Daten

Aktuelle Excel-Varianten haben neben der klassischen Herkunft der Tabellenkalkulation mittlerweile auch viele Datenbanktechniken implementiert.

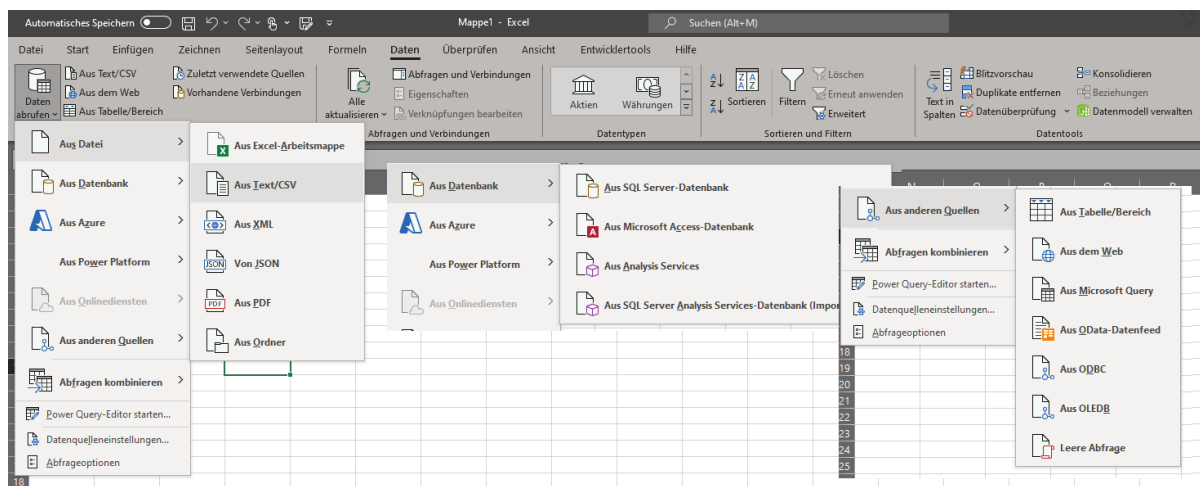


Abb. 10.6: Excel Import-Varianten

Oft sind auch beim Erstellen die jeweiligen Formate wählbar.

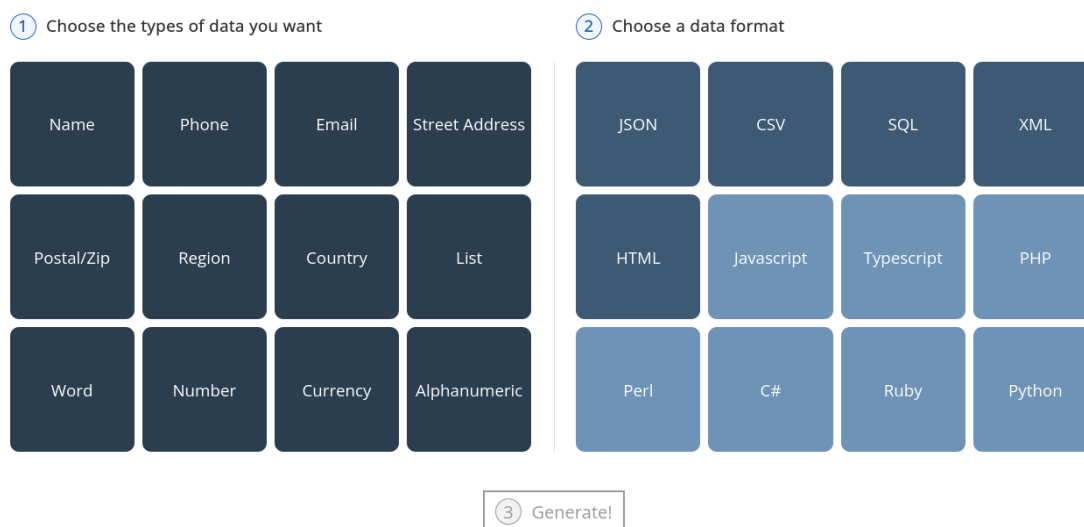


Abb. 10.7: Datensätze erstellen und Format festlegen

Bei Excel kann man über alle Versionen und Techniken den Import von textbasierten Dateien (xsv, json) grob in drei Stufen unterscheiden:

- Text direkt öffnen
Das führt meist zu einer ungewünschten Importierung ganzer Zeilen in Zellen.
- Texte über Datei einfügen
Hier wird man über einen einfachen Assistenten geleitet.

- Texte mittels Abfrage importieren

Wir erhalten Zugriff auf **Power Query** Techniken mit sehr weitreichenden technischen Möglichkeiten.

Meistgenutzt und historisch früh angesiedelt ist der Datenimport mittels

- **CSV-Format** - Comma-separated Values ([Wikipedia - CSV \(Dateiformat\)](#)⁴²).

Für verschachtelte hierarchisch geordnete Datenstrukturen gibt es Alternativen:

- **JSON** - [Wikipedia JSON](#)⁴³
- **XML** - [Wikipedia XML](#)⁴⁴

Wichtig: Beim Importieren der entsprechenden Datenspalten müssen sauber die entsprechenden **Datentypen** und auch die **Zeichenkodierungen** beachtet werden.

Entsprechende Umsetzungen - mit und ohne Probleme folgen in diesem Abschnitt.

10.3.1 Import-Quellen

Als Quellen für solchen Daten gibt es diverse Quellen im Netz, wo es offene (Public Domain) Statistikinfos oder andere Daten gibt.

Beispiele: (Anm.: Quellen übernommen aus Seminaraterial)

- [Govdata - Datenportal für Deutschland](#)⁴⁵
Sehr viele, aber recht ungeordnete Datensätze
- [Statista - das Statistikportal](#)⁴⁶
Account nötig - wenige kostenlose Infodaten

Und es gibt auch Portale auf denen man sich Testdatensätze bequem online zusammenstellen kann.

- [Generatedata.com](#)⁴⁷
- [Mackaroo - Fake Data](#)⁴⁸
- [Online Test Data Generator](#)⁴⁹

Nutzen eines Portals am Beispiel: [GenerateData.com](#)⁵⁰

⁴² [https://de.wikipedia.org/wiki/CSV_\(Dateiformat\)](https://de.wikipedia.org/wiki/CSV_(Dateiformat))

⁴³ https://de.wikipedia.org/wiki/JavaScript_Object_Notation

⁴⁴ https://de.wikipedia.org/wiki/Extensible_Markup_Language

⁴⁵ <https://www.govdata.de/>

⁴⁶ <https://de.statista.com/>

⁴⁷ <https://generatedata.com/generator>

⁴⁸ <https://www.mockaroo.com/>

⁴⁹ <https://www.onlinedatagenerator.com/>

⁵⁰ <https://generatedata.com/>

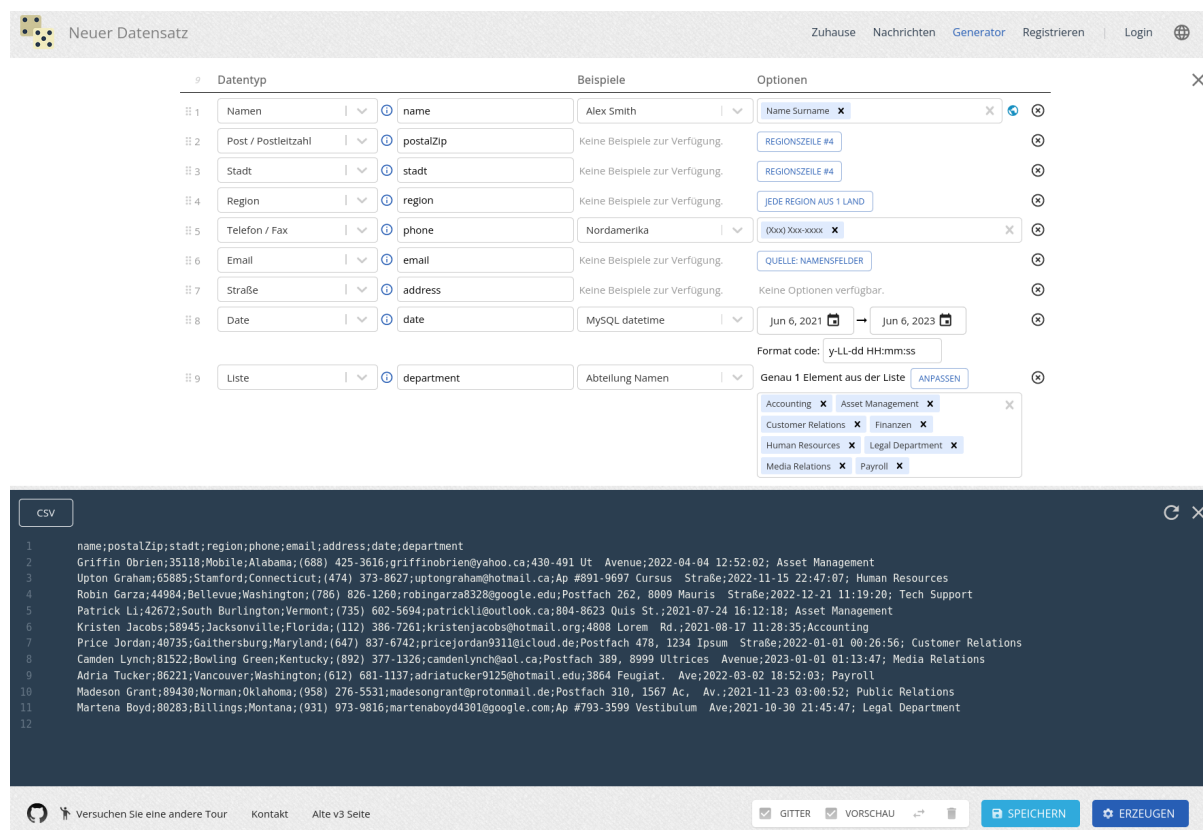


Abb. 10.8: generatedata.com - Website zum Erstellen von Datensätzen

Für die dargestellte (suboptimale) Datenerstellung ergibt sich für die ersten Datensatzzeilen die folgende Darstellung:

```

name;postalZip;stadt;region;phone;email;address;date;department
Xyla Austin;52939;Detroit;Michigan;(293) 443-0668;xylaaustin@aol.org;Postfach
↳950, 1049 Diam. St.;2022-01-25 02:49:58; Payroll
Vladimir Matthews;82586;Bloomington;Minnesota;(421) 713-1801;
↳vladimirmatthews7562@hotmail.org;207-8520 Molestie. St.;2023-01-04
↳06:56:26; Human Resources
Julie Gomez;71750;Virginia Beach;Virginia;(777) 572-2884;juliegomez@outlook.
↳org;763-5015 Sit Straße;2022-03-23 19:34:09; Asset Management
Shelly Murray;66516;Bloomington;Minnesota;(907) 143-7642;shellymurray@google.
↳org;Ap #906-2417 Donec Av.;2021-06-14 12:16:01; Human Resources
Idona Weber;45246;Bellevue;Nebraska;(798) 554-4446;idonaweber9729@aol.net;381-
↳3679 Maecenas St.;2022-02-25 23:43:00; Media Relations
James Guerrero;35836;Birmingham;Alabama;(278) 677-4598;
↳jamesguerrero1568@outlook.ca;912-3942 Arcu Straße;2022-04-22 15:57:06;
↳Accounting
Imogene Le;77421;Annapolis;Maryland;(193) 458-3454;imogenele@icloud.net;
↳Postfach 941, 5249 Metus. Straße;2022-03-17 02:00:22; Public Relations
...
    
```

Insgesamt wurden hier für weitere Darstellungen im Seminartraining 100 Datensätze erzeugt.

Für die erste Analyse ergeben sich folgende Fragen:

- Zeichencodierung: Was passiert mit „Straße“?
- Datum/Uhrzeit-Spalte: Wird in Tabellenkalkulation richtig formatiert?
- Letzte Spalte: Wie löst man Problem mit Leerzeichen for „department“?

Neben dem Online-Portal gibt es die Software zum Erstellen der Datensätze in den diversen Formaten auch als freie Software auf einem [Github Portal Generatedata](https://github.com/benkeen/generatedata)⁵¹

10.3.2 CSV-Import LibreOffice

Das Importieren von **CSV-Dateien** ist bei LibreOffice - wie auch seinem Vorgänger OpenOffice - denkbar einfach.

Beim Öffnen einer CSV-Datei startet ein Assistent und führt durch die möglichen Schritte bei allgemeinen Anpassungen bis hin zu speziellen Konfigurationen für die Datenspalten.

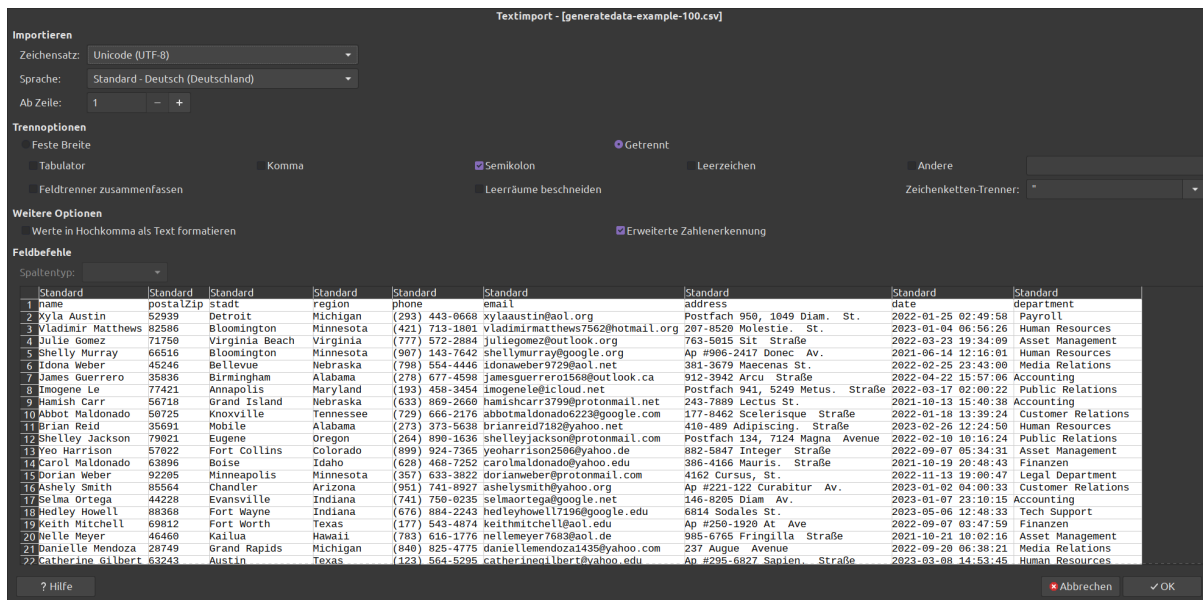


Abb. 10.9: LibreOffice Import Assistent

Von unseren Problemtopics ist auf jeden Fall noch die letzte Spalte gut erkennbar: Leerzeichen am Anfang der Zelle.

10.3.3 CSV-Import Excel

Auch Excel erlaubt das einfache Öffnen (*Importieren*) von CSV-Dateien (per Doppelklick). Allerdings wird bei diesem **Schnellimport** durch Meldungen deutlich gemacht, dass man für die weitere Verarbeitung auf ein *echtes* Excel-Format umstellen (umspeichern) möchte.

⁵¹ <https://github.com/benkeen/generatedata>

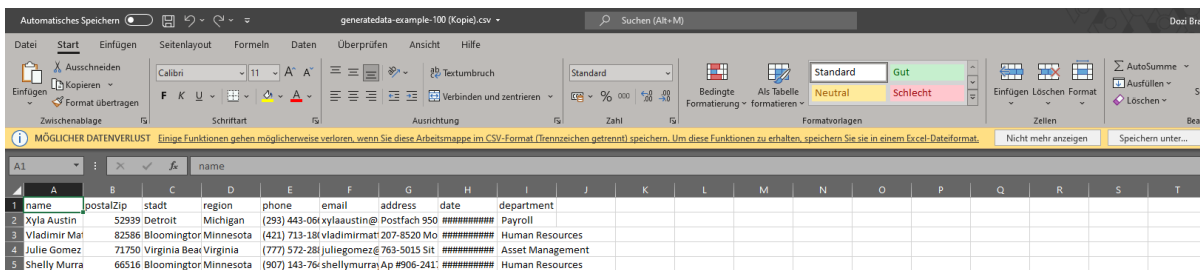


Abb. 10.10: Excel Meldung bei Einfach Import

Aber natürlich kann Excel professionell Daten importieren. Dafür allerdings bedient sich Excel der eigentlichen Profiteools aus der Datenbanktechnik, sodass man bei dieser Vorgehensweise dann eben auch bei diesen Profitechniken landet.

Der Excel Import beginnt wie schon einmal gezeigt über das Menüband **Daten - Daten abrufen und transformieren - Daten importieren**.

Siehe Bildmaterial *Import von Daten*.

Der Import stellt im ersten Schritt einen Datenspaltenanalyse bereit.

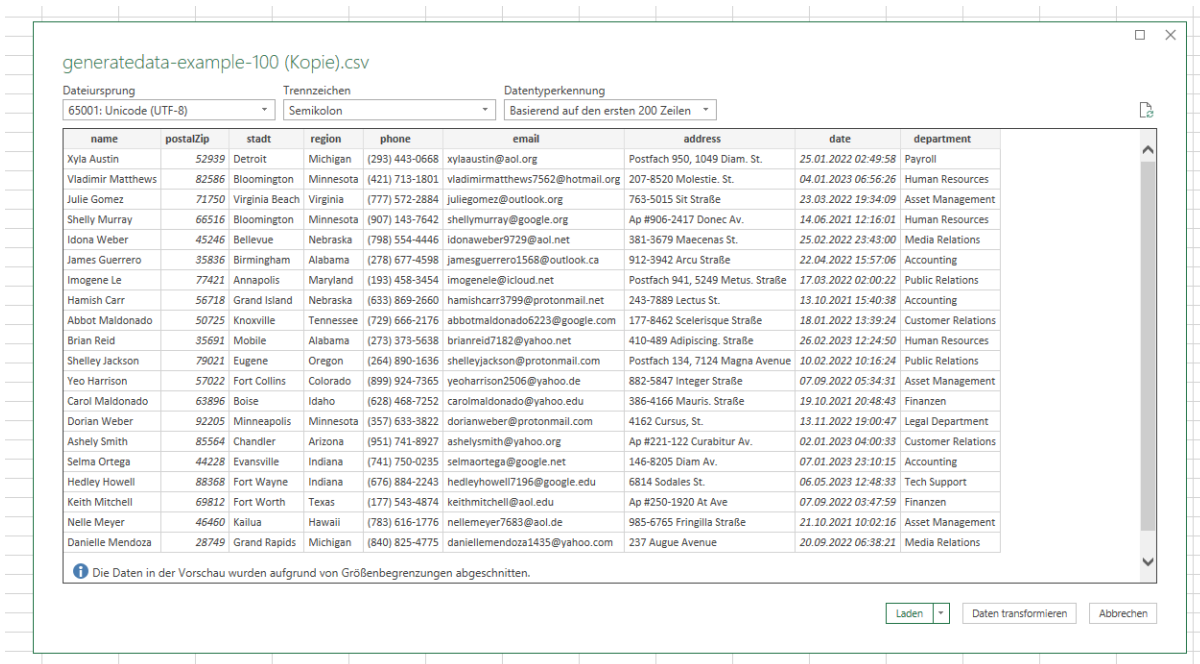


Abb. 10.11: Excel Import Analyse

Als schnelle Lösung kann man einfach auf die Schaltfläche **Laden** klicken und bekommt den Import sofort in Excel verfügbar.

Für das **Daten Transformieren** der Daten stellt Excel mit **Power Query** ein Profiteool aus den Anbindungen für Datenbanktechnik bereit.

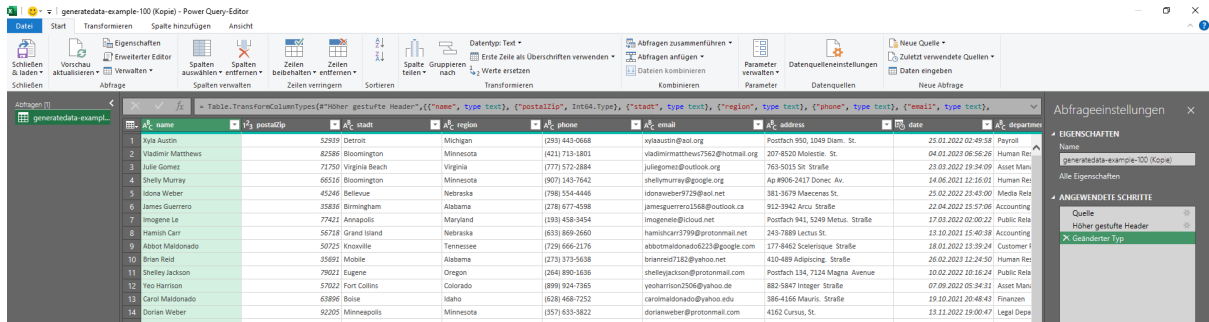


Abb. 10.12: Excel Power Query

Hier kann man jetzt die fraglichen Datentypen und Eigenschaften anpassen, bevor man die Datensätze dann letztlich lädt.

Die **Abfrage** mit der CSV-Datei wird in der Mappe hinterlegt. Es handelt sich hier aber nicht um eine Verknüpfung wie bei **Verbindungen** zu externen Tabellentechniken.

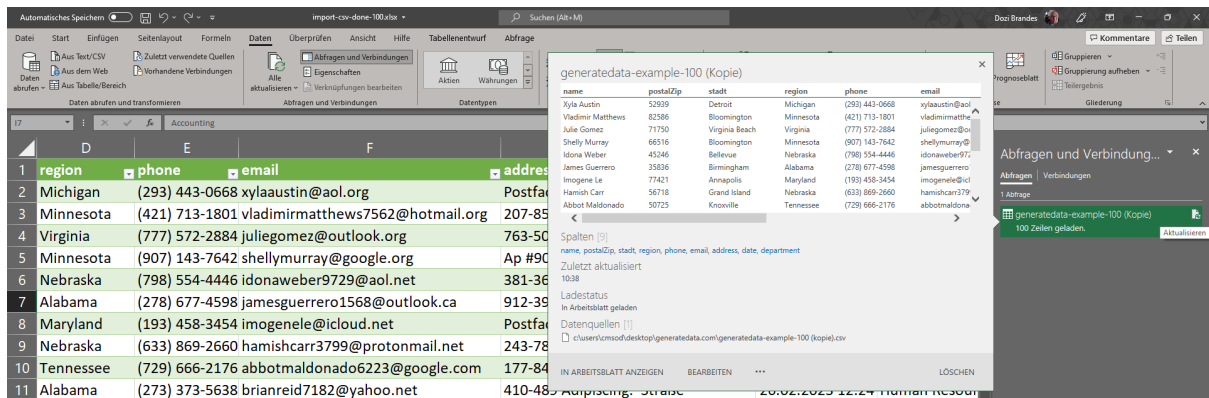


Abb. 10.13: Excel - Daten - Abfrage

Man könnte hier bei Änderungen von CSV-Dateien einfach auf **Aktualisieren** klicken, um die Datensätze immer wieder gleich zu importieren.

10.3.4 CSV-Import Excel WebApp

Diese Variante zeige ich eigentlich nur aus Abschreckungsgründen, das hier keinerlei sinnvolle Anpassungen vorgenommen werden können.

Selbst in meinem sehr einfachen Beispiel bekomme ich sofort Problem mit Zeichencodierungen.

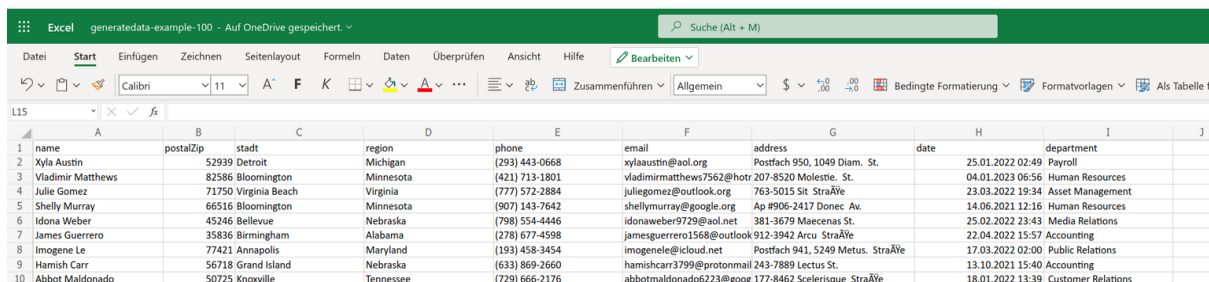


Abb. 10.14: Excel WebApp Import - Zeichencodierungsproblem

Bei den Import-Datensätzen befanden sich - trotz US-Schwerpunkt - auch Datensätze mit Text „Straße“, die sich nun falsch zeigen: Stra  e.

10.3.5 CLI Tools

F r die Analyse oder auch Anpassung von textbasierten Datenquellen in diversen Formaten nimmt man als Techniker auch immer gerne **CLI Tools** (Command Line Interface) - gem   dem Informationstechnik-Motto „Kiss - Keep it simple Stupid“.

```

> onlineseminare@vm-hpwin10sem ~ > Documents > tubs-wmm-2022-local > csv-json-xml
> cat .\csv-version.csv
name,company,country,postalZip
Colby Sanchez,Portitor Interdum Corp.,United States,94620
Linus Romero,Tortor Dictum Foundation,United States,68328
Damon Sanchez,Scelerisque Dui Suspendisse Incorporated,United States,27753
Dustin Hernandez,Orci LLP,United States,56728
Alexis West,A Malesuada Consulting,United States,64849
> onlineseminare@vm-hpwin10sem ~ > Documents > tubs-wmm-2022-local > csv-json-xml
> xsv count .\csv-version.csv
5
> onlineseminare@vm-hpwin10sem ~ > Documents > tubs-wmm-2022-local > csv-json-xml
> xsv table .\csv-version.csv | xsv table
name          company          country          postalZip
Colby Sanchez  Portitor Interdum Corp.  United States    94620
Linus Romero   Tortor Dictum Foundation  United States    68328
Damon Sanchez  Scelerisque Dui Suspendisse Incorporated  United States    27753
Dustin Hernandez Orci LLP          United States    56728
Alexis West    A Malesuada Consulting  United States    64849
> onlineseminare@vm-hpwin10sem ~ > Documents > tubs-wmm-2022-local > csv-json-xml
> xsv select name,postalZip .\csv-version.csv | xsv table
name          postalZip
Colby Sanchez  94620
Linus Romero   68328
Damon Sanchez  27753
Dustin Hernandez 56728
Alexis West    64849
> onlineseminare@vm-hpwin10sem ~ > Documents > tubs-wmm-2022-local > csv-json-xml
> xsv slice .\csv-version.csv -s 3 | xsv table
name          company          country          postalZip
Dustin Hernandez Orci LLP          United States    56728
Alexis West    A Malesuada Consulting  United States    64849
> onlineseminare@vm-hpwin10sem ~ > Documents > tubs-wmm-2022-local > csv-json-xml
> xsv index .\csv-version.csv
> onlineseminare@vm-hpwin10sem ~ > Documents > tubs-wmm-2022-local > csv-json-xml
> xsv stats .\csv-version.csv --everything | xsv table
field  type  sum  min  max  min_length  max_length  mean  stddev  median  mode  cardinality
name   Unicode  Alexis West  Linus Romero  11  16  62455.6  21502.734343334105  64849  N/A  5
company Unicode  A Malesuada Consulting  Tortor Dictum Foundation  8  40  5  5  62455.6  21502.734343334105  64849  N/A  5
country Unicode  United States  United States  13  13  5  5  62455.6  21502.734343334105  64849  N/A  5
postalZip Integer  312278  27753  94620  5  5  62455.6  21502.734343334105  64849  N/A  5
> onlineseminare@vm-hpwin10sem ~ > Documents > tubs-wmm-2022-local > csv-json-xml
> onlineseminare@vm-hpwin10sem ~ > Documents > tubs-wmm-2022-local > csv-json-xml
> cat .\json-simple-version.json | jq -wim64.exe '.[] | select(.name == "Ahmed Kelley")'
[
  {
    "name": "Ahmed Kelley",
    "company": "Sem Molestie Corp.",
    "country": "United States",
    "postalZip": "75532"
  },
  {
    "name": "Jasper Butler",
    "company": "Sociosqu Ad Ltd",
    "country": "United States",
    "postalZip": "72150"
  },
  {
    "name": "Lydia Moreno",
    "company": "Duis Volutpat PC",
    "country": "United States",
    "postalZip": "32650"
  },
  {
    "name": "Zenia George",
    "company": "Malesuada Malesuada Incorporated",
    "country": "United States",
    "postalZip": "11690"
  },
  {
    "name": "McKenzie Lee",
    "company": "Metus In Nec Limited",
    "country": "United States",
    "postalZip": "61805"
  }
]
> onlineseminare@vm-hpwin10sem ~ > Documents > tubs-wmm-2022-local > csv-json-xml
> cat .\json-simple-version.json | jq -wim64.exe '.[] | select(.name == "Ahmed Kelley")'
[
  {
    "name": "Ahmed Kelley",
    "company": "Sem Molestie Corp.",
    "country": "United States",
    "postalZip": "75532"
  }
]
> onlineseminare@vm-hpwin10sem ~ > Documents > tubs-wmm-2022-local > csv-json-xml
> cat .\json-simple-version.json | jq -wim64.exe '.[] | select(.name == "Ahmed Kelley")'
[
  {
    "name": "Ahmed Kelley",
    "company": "Sem Molestie Corp.",
    "country": "United States",
    "postalZip": "75532"
  }
]

```

Abb. 10.15: CLI Tools xsv (CSV) und jq (JSON)

Die Darstellung zeigt die Nutzung der beiden Tools in der PowerShell unter Windows.

- **xsv** - [xsv auf Github \(User: BurntSushi\)](https://github.com/BurntSushi/xsv)⁵²

Neben dem Online-Repo-Docs hier Hinweis auf Angabe eines **Delimiter** bei der Nutzung von xsv:

```
xsv headers datei.csv -d ',';
```

Standardm  ig erwartet das Tool (wortw rtlich) das Komma als Trenner.

- **jq** - [jq auf Github \(User: stedolan\)](https://github.com/stedolan/jq)⁵³

Die Nutzung von jq geschieht offensichtlich  ber Pipelining und spielt aktuell neben der reinen Datenbereitstellung auch bei Container/Docker/Podman Techniken oder in Form moderner Konfigurationsdateien (z.B. VS Code settings.json) eine Rolle.

Es soll an dieser Stelle nicht verschwiegen werden, dass professionelle CLI-Umgebungen am Besten in einer Linux/Unix- Betriebssystemumgebung durchgef hrt werden sollten (Anm.: Zeichencode, EOL, ...). Oder aber man ist sich dieser Unterscheide bei der Arbeit mit diesen CLI-Tools stets bewusst.

⁵² <https://github.com/BurntSushi/xsv>

⁵³ <https://github.com/stedolan/jq>

10.4 Grafische Darstellungen

Für die Darstellung, Analyse und Interpretation von Daten stellt das *moderne* Excel diverse Grafische Darstellungen - also Diagramme - bereit.

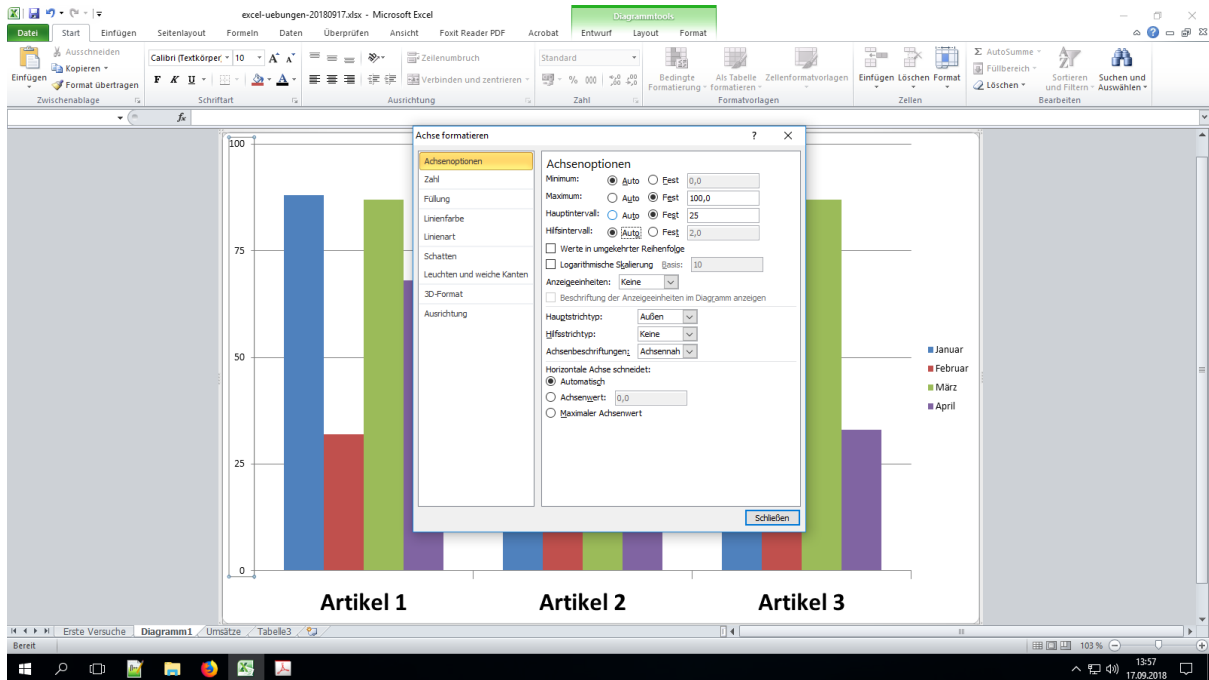


Abb. 10.16: Excel - Diagramme

Für das Erstellen eines einfachen Diagramms, um dann weitere individuelle Anpassungen vornehmen zu können, lassen sich Diagrammblätter per Tastendruck erstellen.

Tip: Nach dem sauberen Markieren der gewünschten Datenreihen drückt man die Funktionstaste F11 für ein Schnelldiagrammblatt.

Neben den klassischen Diagrammtypen und den speziellen Möglichkeiten wie z.B. den Pivot Charts (oder Power Pivot) sind in den aktuellen Excel-Versionen ab 2016 (bis heute 365) diverse interessante Diagrammtypen implementiert worden.

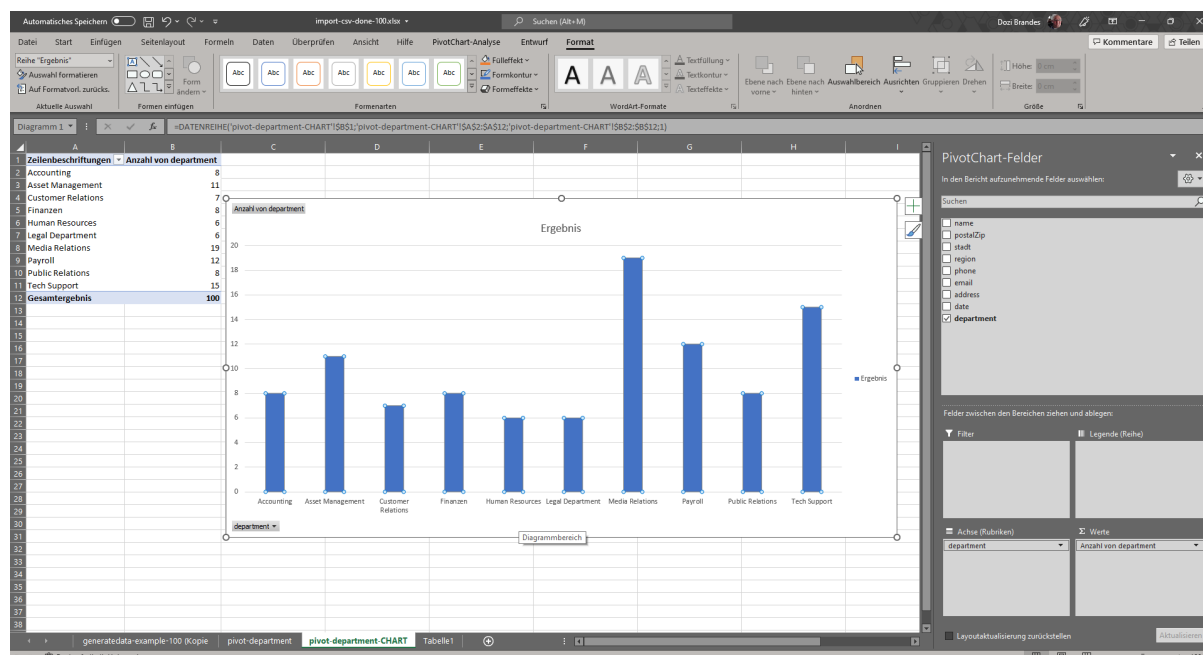


Abb. 10.17: Excel - Pivot Chart

Hier sind also tatsächlich Unterschiede bei Excel-Versionen zu finden! Es folgt eine kurze Übersicht ohne Anspruch auf Vollständigkeit.

- Überblick Diagramme⁵⁴
- Hierarchie-Diagramme⁵⁵
 - für Treemap und Sunburst (inkl. Beispieldaten)
 - Treemap⁵⁶
 - Sunburst⁵⁷
- Wasserfall⁵⁸ auch Brückendiagramme genannt
- Kastengrafik⁵⁹ engl.: Box and Whiskers
- Blasendiagramm⁶⁰ eigentlich schon seit Excel 2007 (!)
- 3D-Karte⁶¹ komplett neue 3D-/Visualisierungstechnik - sehr Performancehungrig!

Viel Spaß mit dieser Zusammenfassung und dem Ausprobieren der *neuen* Techniken.

⁵⁴ <https://www.microsoft.com/en-us/microsoft-365/blog/2015/07/02/introducing-new-and-modern-chart-types-now-available-in-office-2016->

⁵⁵ <https://www.microsoft.com/en-us/microsoft-365/blog/2015/08/11/breaking-down-hierarchical-data-with-treemap-and-sunburst-charts/>

⁵⁶ <https://support.office.com/de-de/article/Erstellen-eines-Diagramms-Treemap-in-Office-2016-dfe86d28-a610-4ef5-9b30-362d5c624b68?ui=de-DE&rs=de-DE&ad=DE>

⁵⁷ <https://support.office.com/de-de/article/Erstellen-eines-Sunburst-Diagramms-in-Office-2016-4a127977-62cd-4c11-b8c7-65b84a358e0c>

⁵⁸ <https://support.office.com/de-de/article/Erstellen-eines-Wasserfalldiagramms-in-Office-2016-8de1ece4-ff21-4d37-acd7-546f5527f185>

⁵⁹ <https://support.office.com/de-de/article/Erstellen-eines-Kastengrafikdiagramms-62f4219f-db4b-4754-aca8-4743f6190f0d>

⁶⁰ <https://support.office.com/de-de/article/Pr%C3%A4sentieren-von-Daten-in-einem-Blasendiagramm-424d7bda-93e8-4983-9b51-c766f3e33>

⁶¹ <https://support.office.com/de-de/article/Erste-Schritte-mit-3D-Karten-6b56a50d-3c3e-4a9e-a527-eea62a387030>

10.5 Daten verändern / auswerten

In diesem Abschnitt geht es häufig bei entsprechenden Seminaren oder Demos um die Nutzung von **Formeln und Funktionen**.

Funktionen 101 – eine erste Übersicht

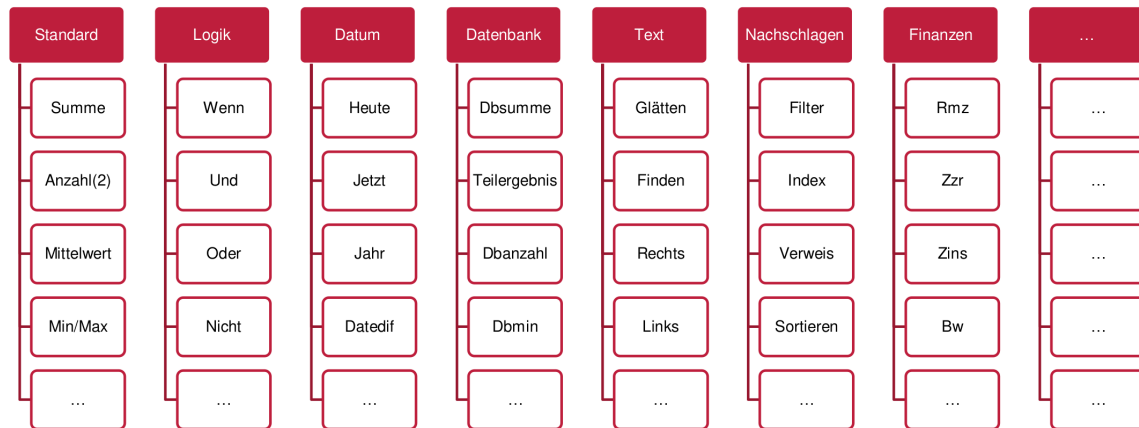


Abb. 10.18: Standardfunktionen Tabellenkalkulation

Und das Ganze natürlich ohne Anspruch auf Vollständigkeit.

Auch diese Darstellung ist etwas *Excel-lastig*, da die Namen der Funktionen dem *deutschsprachigem Excel* entnommen sind.

In anderssprachigen Ausgaben und anderen Tabellenkalkulationen können diese Bezeichner für die Funktionen abweichen!

Hier noch ein paar Infos zum Orientieren:

- Zellen können Texte, Zahlen und Formeln enthalten.
- In Formeln können wir Funktionen nutzen und kombinieren.
- Eingebaute Funktionen sind oft kategorisiert: Mathematisch, Finanzen, Logik, Datum/Uhrzeit, Statistik, Datenbank, Logik, Text (Strings), Nachschlagen, ...
- Funktionsnamen können je nach Tabellenkalkulation variieren:
ZÄHLENWENN() vs. COUNTIF()
- Mit Techniken wie VBA kann man sich eigene Funktionen erstellen.

Besondere Erwähnungen für Auswertungen:

- **Bedingte Formatierungen**

Tabellbereiche werden aufgrund von Eigenschaften auf gewünschte Art und Weise hervorgehoben.

- **Als Tabelle formatierte Bereiche...**

Mit dieser Technik erhalten Tabellenbereiche im Grunde schon Eigenschaften und Technikunterstützungen wie bei Datenbanktabellen.

Best Practise: Index-Spalte, Bereich sauber benennen

- **Pivot Tabellen bzw. Diagramme**

Mit diesen Techniken - ehemals auch als Kreuztabellenabfragen bekannt - kann man sehr effizient und schnell Daten auswerten.

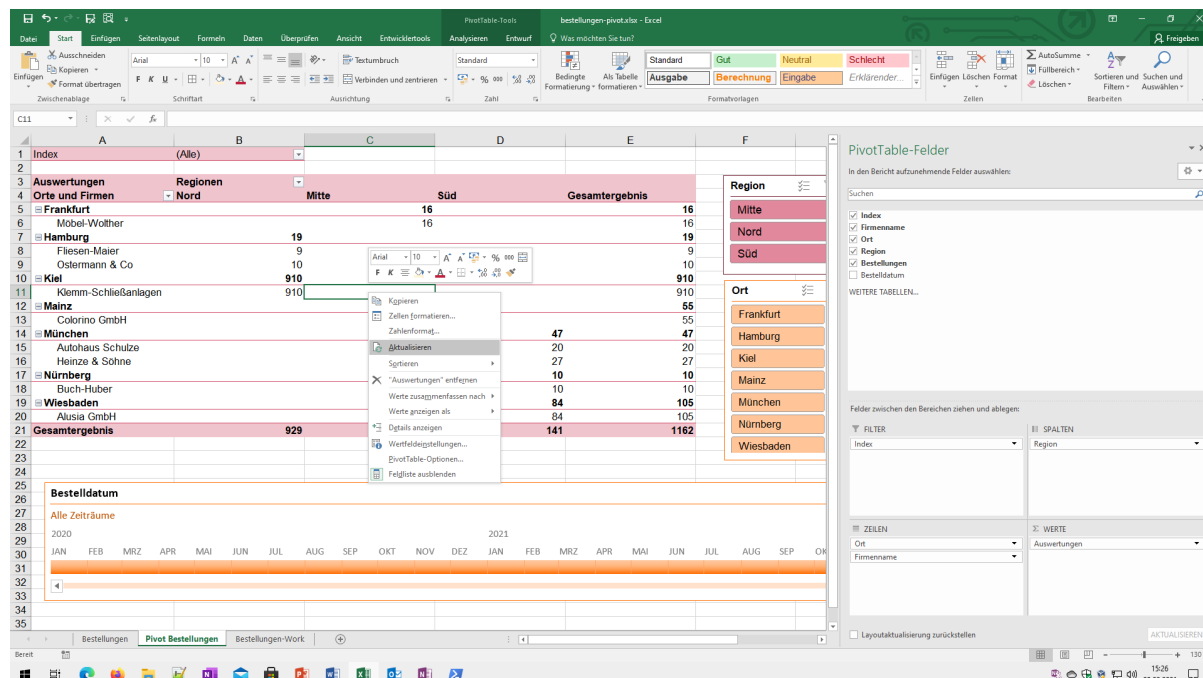


Abb. 10.19: Excel Pivot Table mit Datenschnitt und Zeitachse

Und spätestens mit Techniken wie **PowerPivot** ist man schnell im Bereich von BI (Business Intelligence) oder sogar Big Data.

10.6 Export von Daten

Für das **Exportieren** von Daten gelten alle gemachten Anmerkungen und technischen Beschreibungen aus dem Import-Kapitel.



Abb. 10.20: Daten exportieren - Bild: andy-li-CpsTAUPoScw-unsplash.jpg

In Programmen wie Excel finden sich wort-wörtlichen Aktionen über die Menübandtechniken.

Unter den diversen Techniklinks zu Datenformaten kann man auch Online-Ressourcen und Tools finden, mit denen man die gewünschten Formate erzeugen oder wandeln kann.

10.7 Arbeitsauftrag

Die Seminare in der Veranstaltungsreihe „Was mit Medien“ schließen für diese Vorlesung mit einer Aufgabe für die Teilnehmer (Studis) nach dem Seminar ab.



Technische
Universität
Braunschweig



Tabellenkalkulation – Was mit Medien 

19. Jul. 2023 08.00 – 09.30 Uhr | Joe Brandes | Homework

Abb. 10.21: WMM Homework Tabellenkalkulation

Diese Aufgabe muss gemäß Aufgabenstellung in der Vorlesung beim WMM-Team eingereicht werden.
Nach Ablauf der Eingabefristen werde ich meine/unsere Aufgabe aus diesem Semester hier als Beispiel veröffentlichen.

Zu den diversen Excel-Seminaren im Seminarportfolio ist jetzt auch ein Baustein der **Was mit Medien-Reihen** hinzugekommen: **Data Literacy**



Abb. 11.1: Header aus TU BS Website 2024

Der Schwerpunkt aus der folgenden Darstellung der Begrifflichkeit **Data Literacy** ist der **Workshop-Tag** für die **praktische Einführung in die Tabellenkalkulation mit Microsoft Excel**.

Zitat aus TU Braunschweig Website:

Data Literacy beschreibt die Kompetenz, Daten sammeln, verwalten, evaluieren und in kritischer Weise anwenden zu können.

In einer vernetzten und digitalen Welt, in Zeiten von Big Data, Industrie 4.0 und dem Internet of Things sind Daten eine unabdingbare Ressource, und der Umgang mit ihnen eine höchst relevante Kompetenz. In dieser Veranstaltung werden über ein Zusammenspiel aus eigenständigem Arbeiten im Onlinekurs und Blockveranstaltungen in Präsenz die Hintergründe eines datenbasierten Lebens, Arbeitens und Forschens behandelt und Grundlagen dazu vermittelt.

Die Studierenden erlernen, was Daten sind, wie sich die Datenkultur in unserer Gesellschaft gestaltet, wie sie Daten managen können und wie Daten und datenbasierte Erkenntnisse kritisch hinterfragt werden können.

In einem Praxisworkshop erlernen Sie, auf einfache Weise Daten zu analysieren und zu interpretieren und arbeiten nachfolgend an einem eigenen Datenprojekt.

Die Veranstaltung ist ein Angebot des Projekthaus und der Universitätsbibliothek der TU Braunschweig.

So weit die Infos aus unserer TU Braunschweig Website (Stand: 2024).

Aus meinen eigenen Recherchen folgen mal zwei Links mit Infos zu **Data Literacy** und netten Infographics:

- [Data literacy from theory to reality: How does it look?](#)⁶²
- [Data Literacy for the Data-phobic: 7 Things Beginners Need to Know](#)⁶³

Von meiner Seite viel Spaß und Erfolg bei unserem Workshop-Tag.

11.1 Praxistag

Im Rahmen der **DATA LITERACY** Seminare haben die Trainees/Studis einen Praxistag für die Einstimmung auf ein eigenes **Datenprojekt** als Arbeit im Rahmen der Ausbildung.

Als Trainer präsentiere ich eine praktische Einarbeitung in relevante Basistechniken von Excel in Abstimmung mit dem Erfahrungsprofil der TeilnehmerInnen.

Zum Abschluss präsentiere ich einen Datensatz und den Versuch der Datenanalyse mittels Excel. Hierbei gebe ich einfach mal eine möglich Pivot-Auswertung als Zielrichtung vor und versuche Probleme und einfache Lösungen darzustellen.

Für den Praxistag am 02.02.2024 habe ich hierfür den folgenden Online Datensatz über *Schulen in Deutschland* ausgewählt.

[Quelle/Link Schulen in Deutschland](#)⁶⁴

Es folgen ein paar Impressionen aus der Verarbeitungen der CSV:

⁶² https://www.researchgate.net/publication/335620777_Data_literacy_from_theory_to_reality_How_does_it_look

⁶³ <https://venngage.com/blog/data-literacy/>

⁶⁴ <https://gist.githubusercontent.com/k-nut/930136ffe8f36d535a451ea93c87e476/raw/424ed9ede13d7c75a136414cca10b38fe4790fa0/jedeschule-data-2023-01-07-reduced.csv>

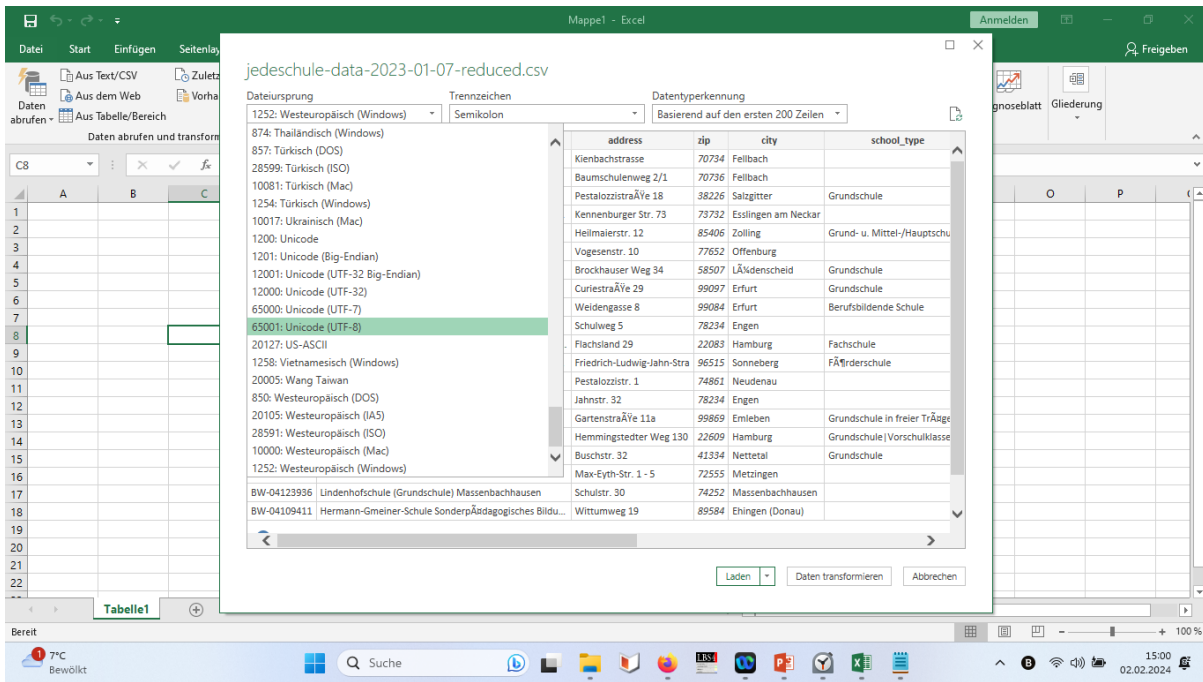


Abb. 11.2: Import Assistent Power Query Excel - hier: Zeichencode beachten

Falls man - z.B. per Pivot - nach Bundesländern auswerten wollte, dann muss man entsprechende IDs (z.B. TH-76015) noch sauber trennen.

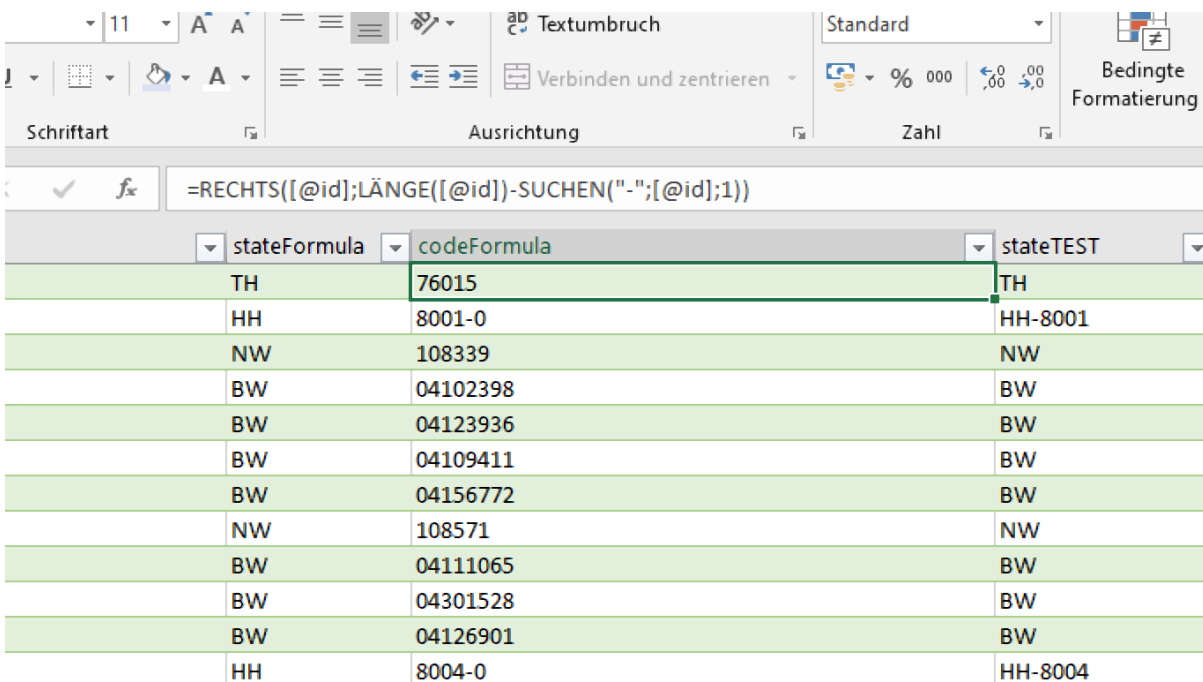


Abb. 11.3: Mit Formeln sauber die Anteile von IDs nach Wunsch trennen

Anm.: die Spalte **id** der Original-CSV hält allerdings noch weitere Stolpersteine bereit, da dort auch einfach freie Texte mit Bemerkungen mit bis zu über 90 Zeichen Länge eingetragen wurden!

Eine Persönliche Zusammenstellung von Links/Bookmarks zu diversen Excel-Themen.



Abb. 12.1: Bookmarks - Bild: markus-winkler-qg9JEHiPX98-unsplash.jpgg

... tbc ...

KAPITEL 13

Bücher

Bücher zum Thema **Tabellenkalkulation mit Excel** gibt es viele und ich will an dieser Stelle nur eine kleine eigene Übersicht anhängen.



Abb. 13.1: Books - Bild: matthew-feeney-Nwkh-n6l25w-unsplash.jpg

Hinweis: Texte und Anmerkungen zu den Büchern von Amazon bzw. den Verlagen.

13.1 Excel - Formeln und Funktionen

Mourad Louha

Dieses umfassende Handbuch hält auch in der aktuellen Neuauflage des Top sellers eine erstklassige Sammlung von über 600 Formeln, Funktionen und Tipps für engagierte Excel-User bereit und ist für alle Excel-Versionen von 2007 bis 2019 bestens geeignet

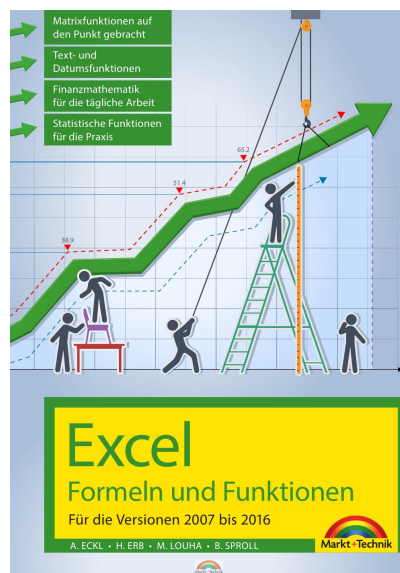


Abb. 13.2: Excel - Formeln und Funktionen

Aus dem Inhalt:

- Daten effektiv und komfortabel erfassen
- Formatierungen zielgerichtet einsetzen
- Arbeitsmappen und Tabellenblätter im Griff haben
- Formeln und Funktionen gekonnt anwenden
- Daten bearbeiten, auswerten und analysieren für die tägliche Praxis
- Professionelle Diagramme, Charts und Präsentationen
- Interne und externe Schnittstellen nutzen
- Nützliche Druck- und Mailfunktionen

Ob Sie komplexe Was-wäre-wenn-Analysen durchführen, mehrere Diagrammtypen kombinieren, Zeitfunktionen praxisbezogen einsetzen oder mit Steuerelementen und Formularen arbeiten: Die jeweilige Formel oder Funktion wird verständlich erklärt und anhand gut nachvollziehbarer Praxisbeispiele veranschaulicht. Zudem werden die neuen und geänderten Funktionen von Excel 2016 sowie 2019 detailliert vorgestellt und erleichtern Ihnen damit den Umstieg auf die aktuelle Version.

Zu den einzelnen Tipps und Beispielen werden Excel-Dateien zum Üben bereitgestellt.

Mourad Louha
Herausgeber: Markt + Technik Verlag (29. März 2019)
Sprache: Deutsch

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

Taschenbuch: 928 Seiten
 ISBN-10: 3959821662
 ISBN-13: 978-3959821667

Anm.: ich führe keine Aktualisierungen von Auflagen und Auflageinfos durch!

13.2 Excel Projektmanagement

Ignaz Schels

Datenanalyse und Reporting mit Power Query, Power Pivot und Power BI Desktop. Für Microsoft 365



Abb. 13.3: Excel Projektmanagement

Unternehmensdaten mit Excel-BI-Tools gekonnt analysieren und präsentieren!

Aus dem Inhalt:

- BI-Werkzeuge und Methoden (ETL, OLAP, ODBC u. a.)
- Externe Daten abrufen und analysieren mit Power Query
- Power Pivot im Datenmodell – Import, Verknüpfungen, Berechnungen
- Mit DAX-Funktionen kalkulieren und KPIs berechnen
- Excel-Daten mit Power BI visualisieren, Berichte und Dashboards in der Cloud veröffentlichen

Der aktuelle Zuwachs an Daten in allen Geschäftsprozessen stellt viele Anwender in der Praxis vor neue Herausforderungen.

Unternehmensdaten müssen aufbereitet werden verbunden mit intelligentem Reporting und Visualisierung. Analysen sollen zukünftig ständig, sofort und auf allen Medien verfügbar sein.

Die benutzerfreundlichen BI-Tools von Microsoft Excel sind hier die erste Wahl für Manager und Controller. Sie helfen Ihnen, Trends und Risiken zu erkennen, Zusammenhänge zu begreifen und die Grundlagen für Ihre Entscheidungen zu verbessern

Ignaz Schels
Herausgeber Carl Hanser Verlag GmbH & Co. KG; 1. Edition (11. Februar 2019)
Sprache: Deutsch
Gebundene Ausgabe 501 Seiten
ISBN-10: 3446457119
ISBN-13: 978-3446457119

Anm.: ich führe keine Aktualisierungen von Auflagen und Auflageinfos durch!

13.3 Business Intelligence mit Excel

Ignaz Schels

Datenanalyse und Reporting mit Power Query, Power Pivot und Power BI Desktop. Für Microsoft 365



Abb. 13.4: Business Intelligence mit Excel

Unternehmensdaten mit Excel-BI-Tools gekonnt analysieren und präsentieren!

Aus dem Inhalt:

- BI-Werkzeuge und Methoden (ETL, OLAP, ODBC u. a.)
- Externe Daten abrufen und analysieren mit Power Query
- Power Pivot im Datenmodell – Import, Verknüpfungen, Berechnungen
- Mit DAX-Funktionen kalkulieren und KPIs berechnen
- Excel-Daten mit Power BI visualisieren, Berichte und Dashboards in der Cloud veröffentlichen

Der aktuelle Zuwachs an Daten in allen Geschäftsprozessen stellt viele Anwender in der Praxis vor neue Herausforderungen.

Unternehmensdaten müssen aufbereitet werden verbunden mit intelligentem Reporting und Visualisierung. Analysen sollen zukünftig ständig, sofort und auf allen Medien verfügbar sein.

Die benutzerfreundlichen BI-Tools von Microsoft Excel sind hier die erste Wahl für Manager und Controller. Sie helfen Ihnen, Trends und Risiken zu erkennen, Zusammenhänge zu begreifen und die Grundlagen für Ihre Entscheidungen zu verbessern

Ignaz Schels
Herausgeber Carl Hanser Verlag GmbH & Co. KG; 1. Edition (11. Februar 2019)
Sprache: Deutsch
Gebundene Ausgabe 501 Seiten
ISBN-10: 3446457119
ISBN-13: 978-3446457119

Anm.: ich führe keine Aktualisierungen von Auflagen und Auflageinfos durch!

13.4 M - Daten abfragen und verarbeiten

Ignaz A. Schels

Copy & Paste war gestern! Wer heute Daten in ein Auswertungsmodell einliest, nutzt am besten eine automatisierte und standardisierte Lösung.



Abb. 13.5: M Daten abfragen und auswerten mit Excel und Power BI (1. Auflage)


Microsoft Excel und Microsoft Power BI Desktop enthalten hierfür den Power Query-Editor, mit dem Sie eine Verbindung zu fast jeder Datenquelle herstellen können.

- Automatisierter Datenimport
- Flexible und stabile Abfragen auch bei schwierigen Datenquellen
- Professionelle und effiziente Lösungen mit der Abfragesprache M
- Nützliche Beispiele Schritt für Schritt erklärt
- Ihr exklusiver Vorteil: E-Book inside beim Kauf des gedruckten Buches
- Neu in der 2. Auflage: KI-Funktionen und eigene Excel-Datentypen aus Power Query

In diesem Buch lernen Sie, dieses vielseitige »Programm im Programm« zu nutzen, um Ihren Datenimport zu automatisieren: Nicht nur das Einlesen, sondern auch die komplexe Datenaufbereitung wird einmalig als Abfrage gespeichert und kann anschließend immer wieder ausgeführt werden.

Grundlage ist die Technologie Power Query, die zuerst als Add-In für Excel 2010 entwickelt wurde und nun fester Bestandteil von Excel und Power BI ist.

Hier folgen die Daten für die 2. Auflage aus 2023:

Herausgeber: Carl Hanser Verlag GmbH & Co. KG; 2., überarbeitete Edition (14.  Juli 2023)

Sprache: Deutsch

Gebundene Ausgabe: 364 Seiten

ISBN-10: 3446477608

ISBN-13: 978-3446477605

Abmessungen: 18 x 2.7 x 24.7 cm

13.5 Excel programmieren

Kofler, Nebelo

Excel programmieren: Abläufe automatisieren, Apps und Anwendungen entwickeln mit Excel



Abb. 13.6: Excel Programmieren

Aus dem Inhalt:

- Der Klassiker zur Excel-Programmierung ; aktualisiert für alle Versionen von 2007 bis 2013
- So automatisieren Sie Abläufe und entwickeln Sie Makros und Anwendungen mit dem bordeigenen VBA
- Überwinden Sie mit Profi-Tools wie VSTO die Grenzen von VBA
- Setzen Sie Webtechniken wie XML, HTML und JavaScript für die Entwicklung von Office-Apps ein

- Mit vielen Programmbeispielen, Tipps und Lösungen für die Praxis

Die Möglichkeiten der Excel-Programmierung sind nahezu unbegrenzt. Die Excel-Versionen 2007 bis 2013 bringen aber nicht nur viele neue Objekte mit, sondern stellen den Entwickler auch vor Aufgaben, die sich nicht mehr oder nur noch teilweise mit VBA lösen lassen: die Anpassung von Menüband und Backstage-Ansicht, die Programmierung von individuellen Aufgabenbereichen, die Abfrage von Web Services, die Entwicklung von Office-Apps u.v.a.

Dahinter verbirgt sich ein radikal neues Erweiterungskonzept, das die klassischen Makros und Add-ins ablösen will. Mit diesem Buch finden Sie sich schnell in der komplexen Objektwelt von Excel zurecht und meistern auch fortgeschrittene Aufgaben mühelos. Sie lernen die erforderlichen Werkzeuge kennen und erhalten das notwendige Know-how für den erfolgreichen Einsatz. Das Spektrum reicht von einfachen Beispielen bis hin zu komplexen, menü- und dialoggesteuerten Programmen.

Dieses Buch hat sich aufgrund seiner praxisnahen Beispiele und der klaren Strukturierung zu einem Standardwerk für Excel-Programmierer entwickelt.

Kofler, Nebelo
 Herausgeber: Carl Hanser Verlag GmbH & Co. KG; 1. Edition (5. Dezember 2013)
 Sprache: Deutsch
 Gebundene Ausgabe: 857 Seiten
 ISBN-10: 3446438661
 ISBN-13: 978-3446438668

Anm.: ich führe keine Aktualisierungen von Auflagen und Auflageinfos durch!

13.6 VBA mit Excel

Bernd Held

VBA mit Excel: Das umfassende Handbuch für Einsteiger und fortgeschrittene Anwender



Abb. 13.7: VBA mit Excel

Aus dem Inhalt:

- Das Standardwerk für Einsteiger und Fortgeschrittene
- Konzepte und Techniken der VBA-Programmierung
- Programme für die tägliche Praxis

Ob Einsteiger oder fortgeschrittener Anwender: Mit diesem Buch werden Sie unter Anleitung des Erfolgsautors Bernd Held zum Profi in VBA mit Excel.

Von den Grundlagen und der Objektprogrammierung über Funktionen und Ereignisse bis hin zu UserForms oder Multifunktionsleisten: Alles drin und mit zahlreichen Beispielen illustriert. Best Practices helfen Ihnen, Ihren Code zu tunen und im Fehlerfall immer die passende Antwort parat zu haben. Das Buch ist damit Einführung, Arbeitsbuch und Nachschlagewerk in einem und unverzichtbar für jeden, der Excel-VBA lernen will.

Mit über 600 sofort einsetzbaren Makros zum Download. Geeignet für Excel 2010 bis Excel 2019 sowie Office 365.

Bernd Held
Herausgeber: Rheinwerk Computing; 4. Edition (28. Dezember 2019)
Sprache: Deutsch
Gebundene Ausgabe: 1016 Seiten
ISBN-10: 3836273985
ISBN-13: 978-3836273985

Anm.: ich führe keine Aktualisierungen von Auflagen und Auflageinfos durch!

13.7 VBA Referenz

Bernd Held

Excel-VBA-Referenz: Objekte, Eigenschaften, Methoden

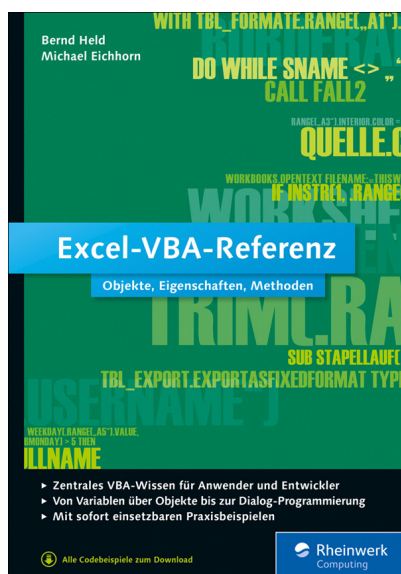


Abb. 13.8: VBA Referenz

Aus dem Inhalt:

- VBA-Wissen für Anwender und Entwickler auf einen Blick
- Von Variablen über Objekte bis zur Dialog-Programmierung
- Mit sofort einsetzbaren Praxisbeispielen und Beispielcode zum Download

Wenn Sie als Anwender oder Entwickler mit Excel-VBA arbeiten und schnellen Zugriff z. B. auf VBA-Objekte oder auch fertige VBA-Lösungen benötigen, dann macht diese Referenz auf Ihrem Schreibtisch eine perfekte Figur!

Neben einer prägnanten Einführung in die VBA-Programmierung und einer Übersicht über wichtige Excel-Funktionen enthält sie alle wichtigen VBA-Objekte mit zugehörigen Eigenschaften und Methoden. Alles illustriert mit häufig benötigten und getesteten, sofort einsetzbaren Beispielen.

Ein thematisch und ein zusätzlich alphabetisch sortiertes Inhaltsverzeichnis sowie der Index machen das Auffinden benötigter Inhalte für Sie zum Kinderspiel.

Bernd Held

Herausgeber: Rheinwerk Computing; 1. Edition (28. August 2015)

Sprache: Deutsch

Gebundene Ausgabe: 448 Seiten

ISBN-10: 3836238357

ISBN-13: 978-3836238359

Anm.: ich führe keine Aktualisierungen von Auflagen und Auflageinfos durch!

13.8 Datenvisualisierung mit Excel

Dirk Siegmann

Zu jeder Datenanalyse gehört eine klare Visualisierung. Werden Datensätze als Grafiken und Diagramme dargestellt, erleichtert dies das Verständnis von Zusammenhängen und Auffälligkeiten.



Abb. 13.9: Datenvisualisierung mit Excel - Das Praxisbuch

Dirk Siegmann zeigt Ihnen in seinem Praxisbuch, wie Sie Ihre Daten mit Microsoft Excel visuell richtig

aufbereiten. So lernen Sie, wie Sie Zahlen aus den unterschiedlichen Bereichen nur mit den Bordmitteln von Excel anschaulich visualisieren können.

- Anschauliche Diagramme und Grafiken mit Excel erstellen
- Von der Aufbereitung der Daten bis zum anschaulichen Diagramm
- Bedingte Formatierungen, erweiterte Diagrammtypen, eigene Darstellungen entwickeln

Dirk Siegmann

Herausgeber: Rheinwerk Computing; 2021

Sprache: Deutsch

400 Seiten, 2021, gebunden, in Farbe

ISBN-13: 978-3-8362-7334-3

E-Book-Formate: PDF, EPUB, MOBI/Kindle, Online

Anm.: ich führe keine Aktualisierungen von Auflagen und Auflageinfos durch!

Alle Bildmaterialien in dieser Ausarbeitung wurden nach bestem Wissen recherchiert und sind gemäß ihrer geforderten Bedingungen hinsichtlich des Autors kommentiert.

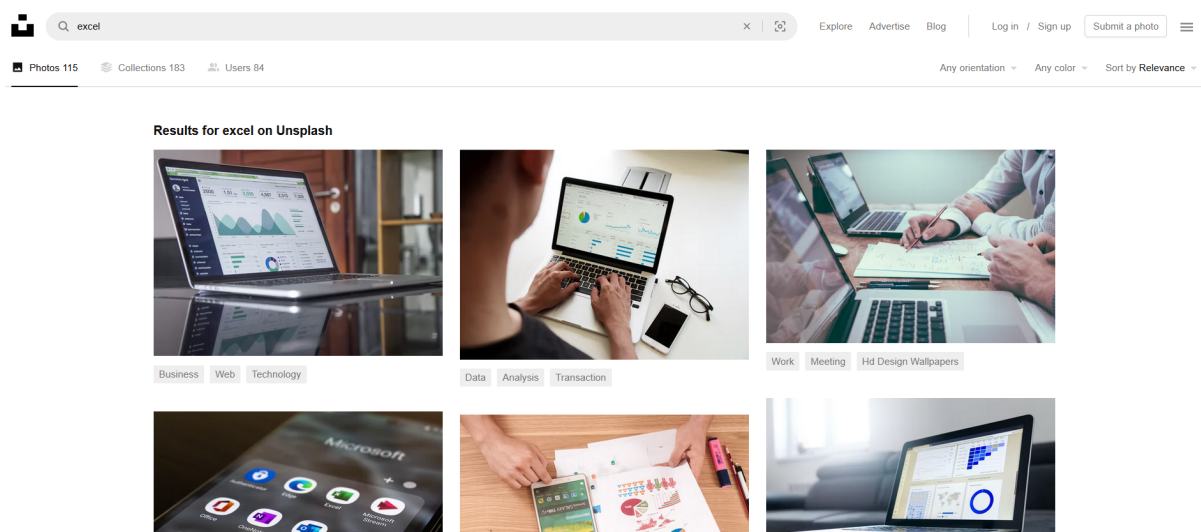


Abb. 14.1: Unsplash.com - Freie Bilder

Die entsprechenden Bildmaterialien entstammen folgenden Quellen:

- **Eigene Bildmaterialien**
oder Bildkompositionen
- **Wikipedia**
Link zum Wikipedia Bild (inkl. Autorennennung)

14.1 Unsplash

Aus dieser Quelle werden alle Bildmaterialien mit Bildname attributisiert.

Erklärung zum Bildcode:

Tab. 14.1: Bildcode Unsplash

Info	Inhalt
Bildname	magnet-me-beCkhUB5aIA-unsplash.jpg
Unsplash Autor	magnet-me
Bildcode	beCkhUB5aIA

Aus dem Unsplash-Bild-Code (hier: beCkhUB5aIA) lässt sich der Bildlink zum Unsplash-Portal zusammenbauen:

- [https://unsplash.com/photos/**Bildcode**](https://unsplash.com/photos/Bildcode)
- [https://unsplash.com/photos/**beCkhUB5aIA**](https://unsplash.com/photos/beCkhUB5aIA)

Abweichende Bild-Quellen wurden entsprechend kommentiert/dokumentiert.

Hier folgen abschließend noch die Meta-Infos zur Dokumentenversionspflege mit Hilfe von **reStructuredText**:

reStructuredText

reStructuredText (kurz *ReST*, *reST* oder *RST*) ist eine vereinfachte [Auszeichnungssprache](#) (Markup) mit dem Ziel, in der reinen Textform besonders lesbar zu sein. Weiterhin soll reStructuredText leicht in andere Formate umwandelbar sein.

Inhaltsverzeichnis [\[Verbergen\]](#)

- 1 [Hintergrund](#)
- 2 [Syntaxbeispiele](#)
- 3 [Weblinks](#)
- 4 [Einzelnachweise](#)

Hintergrund [\[Bearbeiten\]](#) [\[Quelltext bearbeiten\]](#)

Der reStructuredText-Parser ist eine Komponente der Docutils, einer in der [Programmiersprache Python](#) entwickelten Textverarbeitungsbibliothek. Version 0.6 unterstützt die Umwandlung in [ODT](#), [Manpage](#), [S5](#), [XHTML](#), [XML](#) sowie [LaTeX](#). Letzteres ermöglicht einen professionellen Satz mit Exportmöglichkeiten nach [PostScript](#) und [PDF](#). Die aktuelle Version von Docutils ist 0.18.1 (vom 23. Dezember 2021).^[1]

reStructuredText findet verbreitet Einsatz in der Formatierung der docstrings, der Python-typischen Kommentare zur Programmdokumentation. Das [Software-Dokumentationswerkzeug Sphinx](#) nutzt reStructuredText. Zudem bieten einige [Wiki-Systeme](#) neben ihrer eigenen [Wiki-Syntax](#) reStructuredText als Alternative an. [GitHub](#), [Bitbucket](#) und [GitLab](#) unterstützen reST.

reStructuredText	
<i>reStructuredText</i>	
Dateiendung:	.rst
MIME-Type:	text/x-rst
Entwickelt von:	David Goodger
Erstveröffentlichung:	2. April 2002
Art:	Vereinfachte Auszeichnungssprache
Website:	https://docutils.sourceforge.io/rst.html

Abb. 15.1: restructuredText

Kurzinfo auf Wikipedia:

reStructuredText (kurz ReST, reST oder RST) ist eine vereinfachte Auszeichnungssprache (Markup) mit dem Ziel, in der reinen Textform besonders lesbar zu sein.

[Wikipedia - restructuredText](#)⁶⁵

Die reST-Dokumenten lassen sich nahezu beliebig in gewünschte Formate wandeln:

- ODT - OASIS Open Document Format for Office Applications

⁶⁵ <https://de.wikipedia.org/wiki/ReStructuredText>

Kurzform: OpenDocument, ODF – engl. Offenes Dokumentformat für Büroanwendungen

- Textformate wie für LibreOffice, Microsoft Word
- HTML - diverse HTML-Varianten (Websitestyle)
- LaTeX
professioneller Satz mit Exportmöglichkeiten nach PostScript und PDF
- PDF (bzw. PostScript)
- Epub - Standard für eBooks

Desweiteren lassen sich die Dokumente

- professionell verlinken (taggen)
- indizieren und
- durchsuchen.

Versionierungsmerkmale für die reST-Dokumentstrukturen:

Release

1.1-2023

Version

1.1

Language

de

Description

Unterlagen zu **Excel** Seminaren Trainer Joe Brandes.
Erstellt mit **restructuredText** / **Sphinx** / **sphinx_rtd_theme** bzw.
sphinx_typo3_theme!

Keywords

Excel, Seminarunterlage, J. Brandes

Copyright

Joe Brandes

Author

Joe Brandes

License

GNU General Public License, either version 2 of the License or any later version.

Rendered

03.02.2024

Abbildungsverzeichnis

1	Excel Screenshot (Microsoft Website); Quelle: microsoft.com/excel	1
1.1	Roter Faden - Bild: amirali-mirhashemian-hTeYcjviZ-s-unsplash.jpg	3
2.1	Basics - Bild: lauren-richmond-490uCO8h7ZA-unsplash.jpg	5
2.2	Excel 365 (Jahrgang 2022)	6
2.3	Tabellen Dimensionen (16.384 Spalten x 1.048.576 Zeilen)	9
2.4	Text - Zahl - Formel	10
2.5	Zahlenformat Datum und Uhrzeit	11
2.6	Automatisches Ausfüllen mit Anfasser (kleines Fadenkreuz unten rechts)	12
2.7	Zellinhalte in ersten Übungen	13
2.8	Eingabetricks für Excel Zellen	13
2.9	Benutzerdefinierte Zahlenformate	14
2.10	Bedingte Formatierungen	15
2.11	Als Tabelle formatierter Bereich	16
2.12	Zellbezüge - relativ und absolut	17
2.13	Gemischte relative und absolute Bezüge	18
2.14	Fenstern Teilen	19
2.15	Fenster fixieren	20
2.16	Ausdruck optimieren	21
2.17	Kommentare und Notizen bei aktuellem Excel	22
2.18	Schnelldiagramm mittels F11	22
3.1	Excel Funktionen (Microsoft Website); Quelle: microsoft.com	26
3.2	Funktion FILTER() auf Microsoft Infoportal	26
3.3	Diverse Standardfunktionen (Funktionen 101)	27
3.4	Excel Funktion Summe	29
3.5	WENN-Funktionen verschachtelt	30
3.6	SVERWEIS-Funktion	30
3.7	TEILERGEBNIS-Funktion	31
3.8	DBSUMME-Funktion	32
3.9	KGRÖSSTE() - die 3 größten Werte berechnen	33
3.10	Excel Translator Website	33
3.11	Funktionen EINDEUTIG und SORTIEREN mit Datenüberprüfung für Liste	34
3.12	Funktionsübersicht mit Funktion FILTER()	35
3.13	Excel 365 Funktionen im Einsatz bei Filter-Technik	36

3.14	Dynamische Arrayfunktion FILTER()	36
4.1	Pivot Tabelle (PivotTable)	37
4.2	Pivot erstellen	38
4.3	Pivot Chart erstellen	39
4.4	Pivot Tabelle (gruppiert)	40
4.5	Datenschnitt und Zeitachsen bei PivotTable	41
4.6	Pivot Datenexport per Doppelklick	42
4.7	Pivot Measure (neue berechnete Spalte)	43
4.8	Pivot Beziehungen	44
4.9	Power Pivot (Daten- und Diagrammansicht)	45
4.10	Power Pivot (Kevin Stratvert Youtube)	46
5.1	Excel Diagramme (102)	47
5.2	Excel Diagrammtypen	48
5.3	Excel Verbunddiagramm (mit Sekundärachse)	49
5.4	Excel Diagrammvorlage speichern per Rechte Maus	50
5.5	Excel Diagrammvorlagen	50
5.6	Datenreihen hinzufügen	51
5.7	Datenvisualisierung mit Excel (Dirk Siegmann)	52
5.8	Excel Diagramme mit PowerPoint Animieren	53
5.9	Excel Diagramme mit PowerPoint verbinden	53
6.1	Template speichern	55
7.1	Szenario	58
7.2	Datentabelle	58
7.3	Zielwertsuche	59
8.1	Makros - aufgezeichneter Code im VBA Editor	61
8.2	Vorlage mit Datei xltm speichern	62
8.3	Persönliche Makroarbeitsmappe (PERSONAL.XLSB)	64
9.1	Direktfenster (Strg + G)	71
9.2	Eigenschaften (F4)	72
9.3	Codebeispiel - Wochenenden kennzeichnen	76
9.4	Code Beispiel - Werte in Spalte suchen	77
9.5	User Form Beispiel Taschenrechner (Herdt Verlag)	84
9.6	User Form Beispiel Eingabemaske (online-vba.de)	84
9.7	Eigene Funktionen mit VBA	85
10.1	Logo - Was mit Medien (siehe TU BS Website Link)	87
10.2	Was mit Medien - Onlinereihe	88
10.3	Tabellarische Datenliste - Bild: mika-baumeister-Wpnoqo2plFA-unsplash.jpg	89
10.4	Python Dataformats	91
10.5	Jsoncrack - Editor	93
10.6	Excel Import-Varianten	94
10.7	Datensätze erstellen und Format festlegen	94
10.8	generatedata.com - Website zum Erstellen von Datensätzen	96
10.9	LibreOffice Import Assistent	97
10.10	Excel Meldung bei Einfach Import	98
10.11	Excel Import Analyse	98
10.12	Excel Power Query	99

10.13	Excel - Daten - Abfrage	99
10.14	Excel WebApp Import - Zeichencodeproblem	99
10.15	CLI Tools xsv (CSV) und jq (JSON)	100
10.16	Excel - Diagramme	101
10.17	Excel - Pivot Chart	102
10.18	Standardfunktionen Tabellenkalkulation	103
10.19	Excel Pivot Table mit Datenschnitt und Zeitachse	104
10.20	Daten exportieren - Bild: andy-li-CpsTAUPoScw-unsplash.jpg	105
10.21	WMM Homework Tabellenkalkulation	106
11.1	Header aus TU BS Website 2024	107
11.2	Import Assistent Power Query Excel - hier: Zeichencode beachten	109
11.3	Mit Formeln sauber die Anteile von IDs nach Wunsch trennen	109
12.1	Bookmarks - Bild: markus-winkler-qg9JEHiPX98-unsplash.jpg	111
13.1	Books - Bild: matthew-feeney-Nwkh-n6l25w-unsplash.jpg	113
13.2	Excel - Formeln und Funktionen	114
13.3	Excel Projektmanagement	115
13.4	Business Intelligence mit Excel	116
13.5	M Daten abfragen und auswerten mit Excel und Power BI (1. Auflage)	117
13.6	Excel Programmieren	118
13.7	VBA mit Excel	119
13.8	VBA Referenz	120
13.9	Datenvisualisierung mit Excel - Das Praxisbuch	121
14.1	Unsplash.com - Freie Bilder	123
15.1	restructuredText	125

Tabellenverzeichnis

2.1	Excel Dateiformate (ohne. Makrotypen)	8
2.2	Excel Tabellenblatt Dimensionen	9
8.1	Excel Dateiformate (inkl. Makrotypen)	63
10.1	Excel Dateiformate	90
14.1	Bildcode Unsplash	124

Sonderzeichen

1904 Datumswerte, 11

A

Als Tabelle formatieren, 16

Ausfüllen, 11

B

benutzerdefiniert, 14

Benutzerdefinierte Liste, 12

C

CSV, 90

D

Data Literacy, 107

DATEDIF, 28

Datenimport, 94

Datenschnitt, 40

Datentabellen, 58

Datenüberprüfung, 34

Datum/Uhrzeit, 10

Diagramme, 22

Drop-Down-Listen, 34

E

EINDEUTIG(), 34

Eingabetricks, 13

Excel

 Dateiformate, 89

 Literatur, 113

Excel Translator, 33

F

Fenster fixieren, 19

Fenster teilen, 19

Filtern, 17

Format übertragen, 15

G

Große Tabellen, 18

I

Importformate, 95

Indexspalte, 16

K

Kommentar, 21

N

Notiz, 21

P

PERSONAL.XLSB, 63

Pivot Export, 41

PIVOTDATENZUORDNEN(), 42

R

restructuredText, 125

S

Sortieren, 17

SORTIEREN(), 34

Spalten, 8

Szenarien, 58

T

Tabellenblatt

 Dimension, 9

U

Unsplash.com, 123

W

Was mit Medien, 87

Z

Zahlenformat, 14

Zeilen, 8
Zeitachse, 41
Zellbezug
 absolut, 17
 relativ, 17
Zellinhalte, 10
Zielwertsuche, 59