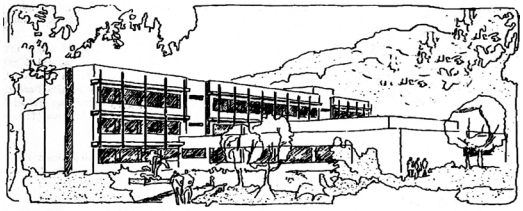
Hochschulrechenzentrum
Justus-Liebig-Universität Gießen



Excel für Microsoft 365

Funktion LAMBDA

HRZ
HOCHSCHULRECHENZENTRUM

JUSTUS-LIEBIG-
UNIVERSITÄT
GIESSEN

Inhaltsverzeichnis

Abbildungsverzeichnis

Einleitung

Excel für Microsoft 365 bietet bereits eine umfangreiche Sammlung an Funktionen, um komplexe Berechnungen durchführen zu können (siehe Skripte **Excel für Microsoft 365 – Funktionen (Übersicht)**, **Excel für Microsoft 365 – Funktionen (Beispiele)** und **Excel für Microsoft 365 – Funktionen (Matrix) und dynamische Arrays**). Trotzdem kommt es immer wieder vor, dass Sie Berechnungen durchführen müssen, bei denen Sie mehrere der in Excel vorgegebenen Funktionen ineinander verschachteln müssen. In solchen Fällen müssen Sie sich gut überlegen, welche Excel-Funktionen Sie überhaupt benötigen und wie Sie die Formel aufbauen müssen, damit das gewünschte Ergebnis bei der Berechnung zustande kommt. Dabei können sehr komplexe Formeln entstehen. Wenn Sie diese Berechnungen dann noch öfters in unterschiedlichen Arbeitsmappen benötigen, bedeutet das einen sehr hohen Aufwand, der nicht nur zeitintensiv ist, sondern auch fehleranfällig sein kann, wenn Sie sich bei der Formeleingabe vertippen. In solchen Fällen wäre es sinnvoll, wenn Sie eine neue benutzerdefinierte Funktion erstellen, die die gewünschte Berechnung durchführt und die Sie dann in jeder Arbeitsmappe einsetzen können. Zu diesem Zweck stellt Excel Ihnen die Programmiersprache **V**isual **B**asic for **A**pplications (kurz **VBA**) zur Verfügung (siehe Skript **Excel für Microsoft 365 – Funktionen (benutzerdefiniert)**). Allerdings hat die ganze Sache einen Haken: Sie müssen den Umgang mit einer Programmiersprache lernen, was in der Praxis aber sehr aufwendig ist. Zudem sind Programmiersprachen für viele Menschen zu komplex. Aber es gibt eine alternative Lösung: die Funktion **LAMBDA**. Mit Ihr können Sie benutzerdefinierte Funktionen erstellen, ohne dafür eine Programmiersprache lernen zu müssen. In diesem Skript bekommen Sie anhand von Beispielen gezeigt, wie Sie mit der Funktion **LAMBDA** umgehen. Bedenken Sie bitte, dass die Funktion **LAMBDA** nur in **Excel für Microsoft 365** zur Verfügung steht.

Aufbau der Funktion LAMBDA

Bevor Sie konkret an Beispielen den Einsatz der Funktion **LAMBDA** gezeigt bekommen, zunächst zur Syntax der Funktion:

=LAMBDA([Parameter1, Parameter2, …, Parameter253], Berechnung)

Dabei haben die Funktionsargumente folgende Bedeutung:

Funktionsargument	*Beschreibung*
Parameter1, **Parameter2**, …, **Parameter253**	Ein Wert, der an die Funktion übergeben werden soll. Das kann beispielsweise ein Zellbezug sein, aber auch eine Zeichenfolge oder eine Zahl. Sie können maximal 253 Parameter angeben. Das Funktionsargument ist optional.
Berechnung	Die Formel, die ausgeführt und als Ergebnis der Funktion zurückgegebene werden soll. Dieses Funktionsargument muss das letzte Argument der Funktion **LAMBDA** sein und ein Ergebnis zurückgeben. Außerdem ist dieses Funktionsargument erforderlich.

Um die Syntax besser verstehen zu können, hier ein kleines Einführungsbeispiel:

=LAMBDA(Zahl; Zahl+1)

Wenn Sie diese Formel nun direkt in eine Tabellenzelle eingeben und die Eingabe bestätigen, erhalten Sie kein Ergebnis, sondern den Fehlerwert **#KALK!**. Das liegt daran, dass Excel ja noch gar nicht weiß, was der Parameter **Zahl** bedeutet und welchen Wert er besitzt. Excel kann also gar kein richtiges Ergebnis liefern. Um diesen Fehlerwert zu vermeiden, wird an die Formel noch ein Wert, eingeschlossen in runden Klammern, hinter der Funktion angefügt. In diesem Beispiel ist es der Wert 4:

=LAMBDA(Zahl; Zahl+1)(4)

Nun liefert die Formel das Ergebnis **5** (4+1). Allerdings liefert dieses Beispiel immer das Ergebnis 5, da es sich bei dem zusätzlichen Wert um einen konstanten Wert handelt. Anstelle des konstanten Werts können Sie auch den Namen einer Tabellenzelle angeben. Dann nimmt Excel den Wert aus dieser Tabellenzelle für die Berechnung. Den Inhalt der Tabellenzelle können Sie dann anschließend jederzeit ändern und damit das Ergebnis der Formel. Wenn beispielsweise die Formel in der Tabellenzelle **A1** steht und der Wert für die Berechnung in der Tabellenzelle **B1**, dann geben Sie folgende Formel in die Tabellenzelle **A1** ein:

=LAMBDA(Zahl; Zahl+1)(B1)

Bei einem zweiten kleinen Beispiel sollen zwei Zahlen miteinander multipliziert werden. Die Formel wird beispielhaft in die Tabellenzelle **A2** eingetragen und für die Werte zur Berechnung der Multiplikation wird der Inhalt der Tabellenzellen **B2** und **C2** genommen. Die Formel sieht dann folgendermaßen aus:

=LAMBDA(x;y; x*y)(B2;C2)

An diesen beiden Beispielen können Sie hoffentlich das Grundprinzip erkennen, wie die Funktion **LAMBDA** funktioniert. Allerdings werden Sie sich jetzt vermutlich die Frage stellen, warum soll ich die Funktion **LAMBDA** verwenden, wenn ich doch einfach die Formel auch so in die Tabellenzelle eintragen kann? Bezogen auf das zweite Beispiel müssten Sie eigentlich in die Tabellenzelle **A2** nur die Formel **=B2*C2** eintragen und bestätigen. Das ist doch viel einfacher. Die Antwort auf diese Frage ergibt sich erst bei komplexen Berechnungen (siehe nachfolgendes Kapitel).

Anmerkung: Sie dürfen für die Funktionsargumente **Parameter1**, **Parameter2**, …, **Parameter253** nicht die Namen **Num1**, **Num2**, …, **Num253** verwenden, da es sich bei diesen Namen um Adressen von Tabellenzellen des Arbeitsblatts handelt. Wenn Sie diese Namen trotzdem als Parameter für die Funktion **LAMBDA** verwenden, erhalten Sie nach Bestätigung der Formeleingabe eine Fehlermeldung (siehe Abbildung 1), auch wenn diese nicht besonders aussagekräftig und eher verwirrend ist.

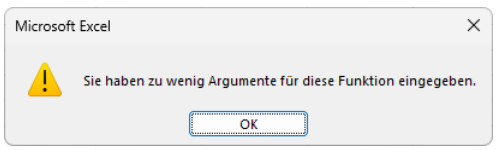


Abb. 1: *Fehlermeldung bei Verwendung der Namen **Num1**, **Num2**, usw.*

Komplexe Formeln mit der Funktion LAMBDA erstellen

Die Funktion **LAMBDA** kommt eigentlich erst bei komplexen Berechnungen zum Einsatz. Dazu wieder ein konkretes Beispiel. In einer Spalte (z.B. Spalte **A**) stehen die Vor- und Nachnamen von Personen in der Form: **Nachname, Vorname**. Nun sollen in einer anderen Spalte (z.B. Spalte **B**) diese Daten übernommen werden, aber in der Form: **Vorname Nachname**. Die eigentlichen Daten beginnen in der Tabellenzelle **A2**. Tragen Sie (z.B.) in die Tabellenzelle **B2** folgenden Formel ein[1]:

```
=TEIL(A2&" "&A2;FINDEN(", ";A2)+2;LÄNGE(A2)-1)
```

Die Formel funktioniert folgendermaßen (am Beispiel des Eintrags **Mustermann, Max**):

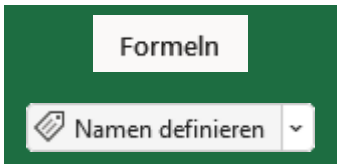
1. Zunächst wird der Inhalt der Tabellenzelle verknüpft mit einem Leerzeichen und erneut dem Inhalt der Tabellenzelle (**A2&" "&A2**). Ergebnis: **Mustermann, Max Mustermann, Max**

2. Nun wird die Position des ersten Buchstabens hinter dem ersten Komma und Leerzeichen ermittelt (**FINDEN(", ";A2)**). Ergebnis (in diesem Beispiel): **13**

3. Nun wird die Länge des Zelleintrags ermittelt (**LÄNGE(A2)-1**). Davon wird noch der Wert **1** abgezogen, da sonst am Ende beim Ergebnis hinter dem Nachnamen noch ein Komma zu sehen ist. Ergebnis (in diesem Beispiel): **14**

4. Nun kommt die Funktion **TEIL** zum Einsatz. Aus dem zusammengefügten Textteil (in diesem Beispiel: **Mustermann, Max Mustermann, Max**) werden ab der Position **13** (das ist der Großbuchstabe M vom Vornamen Max) insgesamt **14** Zeichen herausgenommen (in diesem Beispiel **Max Mustermann**).

Wenn Sie jetzt diese Formel öfters innerhalb der Arbeitsmappe in verschiedenen Arbeitsblättern einsetzen wollen, müssten Sie jedes Mal die Formel an die entsprechende Stelle kopieren und dann noch die Zellbezüge bearbeiten. Das ist auf Dauer zu aufwendig und auch fehleranfällig. Hier kommt nun die Funktion **LAMBDA** ins Spiel. Dazu wählen eine beliebige Tabellenzelle aus (mit Inhalt oder ohne spielt keine Rolle) und führen folgende Schritte aus:



1. Wählen Sie im Register **Formeln** in der Gruppe ***Definierte Namen*** das Symbol **Namen definieren**.

2. Im Dialogfeld **Neuer Name** tragen Sie in das Textfeld **Name** einen Namen für die neue Funktion ein (z.B. **NameTauschen**; Leerzeichen dürfen im Namen nicht vorkommen).

3. Löschen Sie den Inhalt des Textfelds **Bezieht sich auf** und tragen die nachfolgende Formel ein (siehe Abbildung 2, Seite 5):

   ```
   =LAMBDA(Name;TEIL(Name&" "&Name;FINDEN(", ";Name)+2;LÄNGE(Name)-1))
   ```

[1] Das Ganze geht eigentlich einfacher mit der Blitzvorschau (siehe Skript **Excel für Microsoft 365 – Blitzvorschau**), aber es soll ja nur ein Beispiel für den Einsatz der Funktion **LAMBDA** sein.

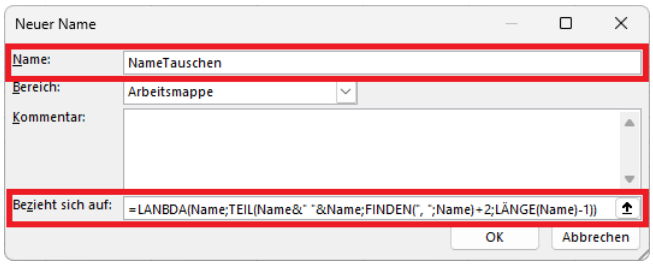


Abb. 2: *Dialogfeld* ***Neuer Name***

Anstelle von **Name** können Sie auch einen anderen Begriff für das erste Funktionsargument der Funktion **LAMBDA** verwenden. Sie müssen nur daran denken, dass dieser Begriff keine Leerzeichen enthalten darf. Groß-/Kleinschreibweise ist ohne Bedeutung.

4. **Optional**: Tragen Sie in das Textfeld **Kommentar** noch eine Beschreibung der Funktion ein.

5. Bestätigen Sie das Dialogfeld.

Wenn Sie jetzt die Funktion einsetzen wollen, wählen Sie die Tabellenzelle aus, wo das Ergebnis der Funktion eingetragen werden soll, und geben ein:

=NameTauschen(A2)

Wenn Sie nach der Eingabe der Formel in die Zellbearbeitung wechseln (z.B. mit der Funktionstaste F2), bekommen Sie im Infokästchen angezeigt **NameTauschen(Name)** (siehe Abbildung 3).

	A	B	C
1	Name, Vorname	Vorname Name	
2	Mustermann, Max	=NameTauschen(A2)	
3		NameTauschen(Name)	

Abb. 3: *Infokästchen mit Funktionsnamen und Funktionsargument*

Bei der Funktion **NameTauschen** gibt es nur ein Funktionsargument, nämlich **Name**. Viele der integrierten Excel-Funktionen besitzen aber mehr als nur ein Funktionsargument. Das funktioniert natürlich auch bei der Funktion **LAMBDA**. Als Beispiel wird der Tagesverdienst für die reine Arbeitszeit von Personen eines Unternehmens ermittelt. Dabei wird auch noch eine Mittagspause (Tabellenzelle **H1**) berücksichtigt und natürlich auch der Stundenlohn (Tabellenzelle **H2**). Die Ausgangstabelle sehen Sie in Abbildung 4.

	A	B	C	D	E	F	G	H
1	Person	Beginn	Ende	Tagesverdienst			Mittagspause:	00:30
2	1	08:03	16:56				Stundenlohn:	16,58 €
3	2	07:58	17:02					
4	3	07:53	16:47					
5	4	08:00	16:55					
6	5	08:05	17:13					

Abb. 4: *Ausgangstabelle für die Berechnung des Tagesverdiensts*

Wenn Sie jetzt ohne die Funktion **LAMBDA** arbeiten, müssten Sie folgende Formel in die Tabellenzelle **D2** eintragen:

```
=STUNDE(C2-B2-$H$1)*$H$2+MINUTE(C2-B2-$H$1)*$H$2/60
```

Nach Bestätigung der Eingabe können Sie die Tabellenzelle **D2** erneut auswählen und die Formel mit dem Hilfsmittel *Automatisches Ausfüllen* (siehe Skript **Excel für Microsoft 365 – Automatisches Ausfüllen**) in die Tabellenzellen **D3**, **D4**, **D5** und **D6** kopieren.

Auch bei diesem Beispiel lohnt sich der Einsatz der Funktion **LAMBDA**. Hier die einzelnen Schritte:

1. Wählen Sie eine beliebige Tabellenzelle.
2. Wählen Sie im Register **Formeln** in der Gruppe ***Definierte Namen*** das Symbol **Namen definieren**.

3. Im Dialogfeld **Neuer Name** tragen Sie in das Textfeld **Name** einen Namen für die neue Funktion ein (z.B. **Tagesverdienst**).
4. Löschen Sie den Inhalt des Textfelds **Bezieht sich auf** und tragen ein:

   ```
   =LAMBDA(Beginn;Ende;Mittagspause;Stundenlohn;STUNDE(Ende-Beginn-
   Mittagspause)*Stundenlohn+MINUTE(Ende-Beginn-Mittagspause)*Stundenlohn/60)
   ```

5. **Optional**: Tragen Sie in das Textfeld **Kommentar** noch eine Beschreibung der Funktion ein.
6. Bestätigen Sie die Eingabe.

Wählen Sie jetzt die Tabellenzelle **D2** und tragen folgende Formel ein und bestätigen anschließend die Eingabe:

```
=Tagesverdienst(B2;C2;$H$1;$H$2)
```

Anschließend wählen Sie erneut die Tabellenzelle **D2** und kopieren die Formeln in die Tabellenzellen **D3**, **D4**, **D5** und **D6** mit dem Hilfsmittel *Automatisches Ausfüllen*. Das Ergebnis sehen Sie in Abbildung 5.

	A	B	C	D	E	F	G	H	I
1	Person	Beginn	Ende	Tagesverdienst			Mittagspause:	00:30	
2	1	08:03	=Tagesverdienst(B2;C2;H1;H2)				Stundenlohn:	16,58 €	
3	2	07:58	Tagesverdienst(Beginn; Ende; Mittagspause; Stundenlohn)						
4	3	07:53	16:47	139,27 €					
5	4	08:00	16:55	139,55 €					
6	5	08:05	17:13	143,14 €					

Abb. 5: *Tabelle nach Berechnung des Tagesverdiensts*

Anmerkung: Die Namen *Beginn*, *Ende*, *Mittagspause* und *Stundenlohn* für die vier Parameter der Funktion **LAMBDA** sind wieder willkürlich gewählt und könnten auch völlig anders lauten. Beispielsweise könnten Sie auch einfach nur die Buchstaben a, b, c und d verwenden, aber das wäre zu abstrakt. Sie wissen später dann nicht, welcher Buchstabe was bedeutet. Sprechende Namen sind da deutlicher besser.

Optionale Funktionsargumente

Wie bereits erwähnt, besitzen viele der in Excel integrierten Funktionen mehr als ein Funktionsargument. Dabei kommt es häufig vor, dass einige Funktionsargumente angegeben werden müssen, während andere Funktionsargumente auch weggelassen werden können. Diese Funktionsargumente sind also optional. Auch solche Funktionsargumente können Sie bei der Funktion **LAMBDA** berücksichtigen. Dazu wieder ein Beispiel: Es soll für einen Nettobetrag eines Produkts der Steuerbetrag berechnet werden. Für die Berechnung des Ergebnisses wird also der Nettobetrag und der Steuersatz benötigt. Allerdings kann der Steuersatz auch weggelassen werden, wenn er **19%** beträgt. Als Beispiel steht der Nettobetrag in der Tabellenzelle **A1** und der Steuersatz in der Tabellenzelle **B1**. Nun soll in der Tabellenzelle **C1** der Steuerbetrag ermittelt werden. Für dieses Beispiel werden gleich die Schritte zur Erstellung der Funktion mit Hilfe der Funktion **LAMBDA** erklärt:

1. Wählen Sie eine beliebige Tabellenzelle.
2. Wählen Sie im Register **Formeln** in der Gruppe ***Definierte Namen*** das Symbol **Namen definieren**.

3. Im Dialogfeld **Neuer Name** tragen Sie in das Textfeld **Name** einen Namen für die neue Funktion ein (z.B. **Steuerbetrag**).
4. Löschen Sie den Inhalt des Textfelds **Bezieht sich auf** und tragen ein:

   =LAMBDA(Netto;[Steuersatz];WENN(WURDEAUSGELASSEN(Steuersatz);Netto+Netto*0,19;Netto+Netto*Steuersatz))

7. **Optional**: Tragen Sie in das Textfeld **Kommentar** noch eine Beschreibung der Funktion ein.
8. Bestätigen Sie die Eingabe.

Sie sehen, dass der Name **Steuersatz** bei den Parametern in eckige Klammer eingeschlossen ist. Damit weiß die Funktion **LAMBDA**, dass es sich um einen optionalen Parameter handelt. Mit Hilfe der Excel-Funktionen **WENN** und **WURDEAUSGELASSEN**[2] wird überprüft, ob für den Parameter **Steuersatz** ein Wert oder Zellbezug angegeben worden ist oder nicht. Ist das Ergebnis dieser Funktion **Wahr** (der Parameter wurde weggelassen), wird die Berechnung **Netto+Netto*0,19** ausgeführt, im anderen Fall die Berechnung **Netto+Netto*Steuersatz**. Nach Erstellung der benutzerdefinierten Funktion tragen Sie in die Tabellenzelle **C1** beispielsweise ein:

=Steuerbetrag(A1)

In diesem Fall nimmt die Funktion für den Steuersatz den Wert **19%** (also **0,19**). Tragen Sie stattdessen ein:

=Steuerbetrag(A1;B1)

wird für den Steuersatz der Inhalt der Tabellenzelle **B1** verwendet.

[2] Die Funktion **WURDEAUSGELASSEN** gibt es auch nur in der Version Excel für Microsoft 365.

Funktionen global bereitstellen

Bei der Benennung von Tabellenzellen (siehe Skript **Excel für Microsoft 365 – Tabellenzellen benennen**) gelten diese Namen nur innerhalb der Arbeitsmappe, wo Sie erstellt worden sind. Das gilt dann selbstverständlich auch für die benutzerdefinierten Funktionen, die mit der Funktion **LAMBDA** erstellt worden sind. Aber Sie wollen vielleicht die eine oder andere Funktion gerne auch in anderen Arbeitsmappen einsetzen.

Wenn Sie nur eine bestimmte benutzerdefinierte Funktion in eine andere Arbeitsmappe übertragen wollen, gehen Sie folgendermaßen vor:

1. Öffnen Sie die Arbeitsmappe (sofern sie nicht bereits geöffnet ist), die die benutzerdefinierte Funktion enthält, die Sie kopieren wollen.

2. Wählen Sie eine Tabellenzelle auf einem Arbeitsblatt aus, in der die zu kopierende Funktion eingesetzt worden ist.

3. Kopieren Sie den Inhalt der Tabellenzelle in die Zwischenablage (z.B. mit der Tastenkombination Strg C).

4. Öffnen Sie die Arbeitsmappe, wohin Sie die benutzerdefinierte Funktion kopieren wollen, sofern dies nicht bereits geschehen ist. Es kann sich natürlich auch um eine neue, leere Arbeitsmappe handeln.

5. Wählen Sie eine beliebige leere Tabellenzelle auf einem beliebigen Arbeitsblatt und fügen den Inhalt der Zwischenablage ein (z.B. mit der Tastenkombination Strg V). Lassen Sie sich nicht daran stören, dass evtl. ein Fehlerwert (vermutlich **#BEZUG!**) angezeigt wird.

6. **Optional**: Löschen Sie den Inhalt der ausgewählten Tabellenzelle.

7. Speichern Sie die Arbeitsmappe.

Damit existiert nun diese benutzerdefinierte Funktion in der Arbeitsmappe, auch wenn Sie in Schritt 6 den Inhalt der Tabellenzelle gelöscht haben sollten.

Wollen Sie mehrere benutzerdefinierte Funktionen aus der einen Arbeitsmappe in die andere Arbeitsmappe übernehmen, sollten Sie folgende Schritte abarbeiten:

1. Öffnen Sie die Arbeitsmappe (sofern sie nicht bereits geöffnet ist), die die benutzerdefinierte Funktion enthält, die Sie kopieren wollen.

2. Öffnen Sie die Arbeitsmappe, wohin Sie die benutzerdefinierte Funktion kopieren wollen, sofern dies nicht bereits geschehen ist. Es kann sich natürlich auch um eine neue, leere Arbeitsmappe handeln.

3. Wählen Sie die Arbeitsmappe aus, die die benutzerdefinierten Funktionen enthält.

4. Bewegen Sie das Maussymbol auf einen beliebigen Blattnamen und klicken die **rechte** Maustaste.

5. Wählen Sie im Kontextmenü den Befehl **Verschieben oder kopieren**.

6. Im Dialogfeld **Verschieben oder kopieren** wählen Sie in der Liste **Zur Mappe** die Arbeitsmappe aus, wohin die benutzerdefinierten Funktionen kopiert werden sollen (siehe Abbildung 6).

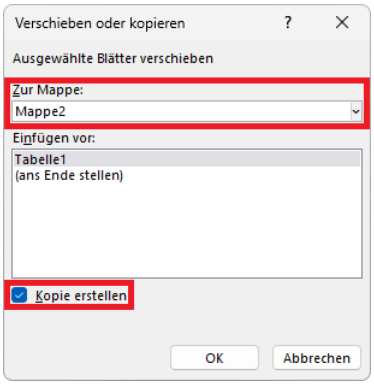


Abb. 6: *Dialogfeld* ***Verschieben oder kopieren***

7. **Optional**: Wählen Sie in der Liste **Einfügen vor** den Namen eines Arbeitsblatts, vor das das kopierte Arbeitsblatt eingefügt werden soll. Es spielt aber keine Rolle, wo innerhalb der Arbeitsmappe das Arbeitsblatt tatsächlich eingefügt wird.

8. Aktivieren Sie das Kontrollkästchen **Kopie erstellen**.

9. Bestätigen Sie das Dialogfeld.

Excel wechselt zur „neuen“ Arbeitsmappe und Sie können sofort die benutzerdefinierten Funktionen einsetzen. Sie können sogar das kopierte Arbeitsblatt löschen, ohne dass dadurch die benutzerdefinierten Funktionen verloren gehen.

Anmerkung: Auch wenn die zuvor genannten Schritte nicht besonders kompliziert sind, so müssen Sie sie dennoch jedes Mal aufs Neue abarbeiten, wenn Sie die benutzerdefinierten Funktionen in eine andere oder neue Arbeitsmappe übernehmen wollen. Eine andere Möglichkeit, die benutzerdefinierten Funktionen global zur Verfügung zu stellen, gibt es leider nicht. Das geht nur, wenn Sie die Funktionen in Form von Makros mit VBA erstellen. Aber wie bereits im Kapitel **Einleitung**, Seite 2, erwähnt, müssen Sie erst die Programmiersprache VBA lernen, bevor Sie damit benutzerdefinierte Funktionen erstellen können.

Unterstützerfunktionen

Im Zusammenhang mit der Funktion **LAMBDA** gibt es noch sogenannte Unterstützerfunktionen, die LAMBDA-Ausdrücke direkt verarbeiten können, d.h. bei einer der Funktionsargumente handelt es sich um die Funktion **LAMBDA**. In einigen Fällen handelt es sich bei den Ergebnissen um Matrizen, genauer gesagt um dynamische Arrays (siehe Skript **Excel für Microsoft 365 – Funktionen (Matrix) und dynamische Arrays**, Kapitel **Dynamische Arrays**, Seite 12). In den folgenden Unterkapitel sollen die einzelnen Unterstützerfunktionen kurz anhand von Beispielen vorgestellt werden.

Funktion MAP

Mit der Funktion **MAP** erstellen Sie eine Matrix, wobei die Ausgangswerte zur Ermittlung der Matrixwerte ebenfalls aus einer Matrix stammen. Die Ergebniswerte für die Zielmatrix werden mit Hilfe der Funktion **LAMBDA** ermittelt. Bei der Zielmatrix handelt es sich um ein dynamisches Array.

Die Syntax der Funktion **MAP** lautet:

=MAP(Matrix1;[Matrix2;…];LAMBDA-Funktion)

Dabei haben die Funktionsargumente folgende Bedeutung:

Funktionsargument	*Beschreibung*
Matrix1, **Matrix2**, …	Die Ausgangsmatrizen, deren Werte für die Berechnung der Funktion **LAMBDA** benötigt werden.
LAMBDA-Funktion	Die für die Berechnung der Ergebniswerte des dynamischen Arrays notwendige Formel. Die Funktion **LAMBDA** ist das letzte Argument der Funktion **MAP**.

Hier nun zwei Beispiele für den Einsatz der Funktion **MAP**.

Im ersten Beispiel enthält die Ausgangsmatrix dreißig Zahlen. In der Zielmatrix sollen die Quadratzahlen der Ausgangszahlen berechnet werden, allerdings nur für die Werte, die größer oder gleich 3 sind. Bei Zahlen, die kleiner als 3 sind, werden die Zahlen einfach nur übernommen. Die Ausgangswerte befinden sich im Zellbereich **A1:J3**. Für die Ermittlung der Ergebnisse wird die Tabellenzelle **A5** markiert und folgende Formel eingegeben:

=MAP(A1:J3;LAMBDA(x;WENN(x>=3;x*x;x)))

Die Eingabe der Formel bestätigen Sie mit der Taste  . Der Zellbereich **A5:J7** wird dank des *dynamischen Arrays* automatisch ausgefüllt (Sie müssen also nicht vorher den kompletten Zellbereich **A5:J7** markieren). Das Ergebnis sehen Sie in Abbildung 7 (der gelbe Zellbereich ist die Ausgangsmatrix, der hellgrüne Zellbereich ist die Zielmatrix).

	A	B	C	D	E	F	G	H	I	J
1	4	2	9	10	1	1	8	10	9	7
2	8	6	8	8	9	8	5	9	7	9
3	3	6	7	1	7	8	9	2	9	9
4										
5	16	2	81	100	1	1	64	100	81	49
6	64	36	64	64	81	64	25	81	49	81
7	9	36	49	1	49	64	81	2	81	81
8										
9	Formel in A5: =MAP(A1:J3;LAMBDA(x;WENN(x>=3;x*x;x)))									

Abb. 7: *Beispiel 1 für die Funktion **MAP**, Quadratzahlen berechnen*

Im zweiten Beispiel wird zunächst eine Liste mit zwei Spalten erstellt, in denen Wahrheitswerte (**WAHR** bzw. **FALSCH**) eingetragen sind. Die Wahrheitswerte stehen im Zellbereich **A2:B8** (in den Tabellenzellen **A1** und **B1** stehen die Überschriften der Spalten). Im Zellbereich **D2:D8** sollen nun

ebenfalls Wahrheitswerte stehen, die mit den Funktionen **MAP** und **LAMBDA** ermittelt werden. Das Ergebnis ist **WAHR**, wenn in der Spalte **A** und in der Spalte **B** jeweils der Wert **WAHR** ist. In allen anderen Fällen ist das Ergebnis **FALSCH**. Wählen Sie die Tabellenzelle **D2** und geben folgende Formel ein:

```
=MAP(A2:A8;B2:B8;LAMBDA(a;b;UND(a;b)))
```

Die Eingabe der Formel bestätigen Sie wieder mit der Taste  . Die restlichen Tabellenzellen **D3** bis **D8** werden wieder automatisch mit der Formel ausgefüllt. Abbildung 8 zeigt das Ergebnis.

	A	B	C	D	E	F	G	H	I	J	K
1	Wert 1	Wert 2		Wert 1 UND Wert 2							
2	FALSCH	FALSCH		FALSCH		Formel in D2: =MAP(A2:A8;B2:B8;LAMBDA(a;b;UND(a;b)))					
3	WAHR	FALSCH		FALSCH							
4	WAHR	WAHR		WAHR							
5	WAHR	FALSCH		FALSCH							
6	FALSCH	WAHR		FALSCH							
7	FALSCH	FALSCH		FALSCH							
8	WAHR	WAHR		WAHR							

Abb. 8: *Beispiel 2 für die Funktion **MAP**, Wahrheitswerte ermitteln*

Anmerkung: Anstelle der Verwendung der Funktion **MAP** können Sie das Ergebnis auch sehr einfach mit der Formel **=UND(A1;B1)** berechnen. Sie müssen dann anschließend die Formel nur noch mit dem Verfahren *Automatisches Ausfüllen* (siehe Skript **Excel für Microsoft 365 – Automatisches Ausfüllen**) bis zur Tabellenzelle **D8** kopieren.

Funktion REDUCE

Mit der Funktion **REDUCE** wird ein kumulierter Wert berechnet, wobei für jeden Wert einer Ausgangsmatrix eine Berechnung mit Hilfe der **LAMBDA**-Funktion durchgeführt wird. Dann werden diese Werte fortlaufend aufsummiert (kumuliert). Am Ende wird mit Hilfe der Funktion **LAMBDA** das Endergebnis berechnet.

Die Syntax der Funktion **REDUCE** lautet:

```
=REDUCE([Startwert];Matrix;LAMBDA(Akkumulator;Wert))
```

Dabei haben die Funktionsargumente folgende Bedeutung:

Funktionsargument	***Beschreibung***
Startwert	Legt den Startwert für den Akkumulator fest. Dieses Funktionsargument ist optional.
Matrix	Ein Zellbereich, dessen Inhalt für die Berechnung verwendet wird.
LAMBDA	Eine Funktion, mit deren Hilfe die Berechnung durchgeführt wird. Die Funktion erlaubt nur die beiden folgenden Funktionsargumente:
Akkumulator	Der Wert, der aufsummiert und als Endergebnis zurückgegeben wird.
Wert	Die Berechnung, die auf jedes Element der Matrix angewendet wird.

Mit den Beschreibungen der einzelnen Funktionsargumente werden Sie vermutlich nicht sehr viel anfangen können. Das wird hoffentlich deutlicher anhand von zwei Beispielen.

Im ersten Beispiel ist ein Zellbereich mit den Zahlen **1** bis **10** im Zellbereich **A1:J1** vorgegeben. Nun soll In der Tabellenzelle **A3** für jeden Wert des Zellbereichs **A1:J1** jeweils das Quadrat berechnet und anschließend alle Werte addiert werden (also **1**+**4**+**9**+**16**+…+**81**+**100**).

Die Formel in der Tabellenzelle **A3** lautet:

```
=REDUCE(0;A1:J1;LAMBDA(a;b;a+b^2))
```

Das Funktionsargument **a** bei der Funktion **LAMBDA** entspricht dem Startwert **0** der Funktion **REDUCE** und bleibt für die gesamte Berechnung konstant. Dem Funktionsargument **b** bei der Funktion **LAMBDA** werden nacheinander die Werte im Zellbereich **A1:J1** zugewiesen. Die Formel für die Berechnung lautet dann **a+b^2** (entspricht $a+b^2$). Da **a** ja immer den Wert **0** besitzt, werden also einfach nur Ergebnisse für **b^2** berechnet und anschließend fortlaufend aufaddiert. Abbildung 9 zeigt die Tabelle.

	A	B	C	D	E	F	G	H	I	J
1	1	2	3	4	5	6	7	8	9	10
2										
3	385		Formel in A3: =REDUCE(0;A1:J1;LAMBDA(a;b;a+b^2))							

Abb. 9: *Beispiel 1 für die Funktion **REDUCE**, Summe der Quadrate ermitteln*

Anmerkung: Anstelle der Verwendung der Funktion **REDUCE** können Sie das Ergebnis auch mit der Formel **=SUMME(A1:J1^2)** berechnen.

Im zweiten Beispiel sind im Zellbereich **A1:J1** verschiedene ganze Zahlen eingetragen. In der Tabellenzelle **A3** soll angegeben werden, bei wie vielen der 10 ganzen Zahlen es sich um gerade Zahlen handelt. Die Formel lautet:

```
=REDUCE(0;A1:J1;LAMBDA(a;b;WENN(ISTGERADE(b);a+1;a)))
```

Auch bei diesem Beispiel wird der Startwert (hier: **0**) dem Argument **a** der Funktion **LAMBDA** zugewiesen. Das Argument **b** der Funktion **LAMBDA** nimmt jeden einzelnen Wert aus dem Zellbereich **A1:J1** an. Die Funktion **ISTGERADE** liefert als Ergebnis **WAHR**, wenn es sich bei der zu überprüfenden Zahl um eine gerade Zahl handelt; im anderen Fall ist das Ergebnis **FALSCH**. Handelt es sich beim Argument **b** um eine gerade Zahl, wird die Berechnung **a+1** ermittelt (also das Ergebnis wird immer um den Wert **1** erhöht). Liefert die Überprüfung das Ergebnis **FALSCH** wird nur der Wert des Arguments **a** als Ergebnis zurückgeliefert (sollte es im Zellbereich **A1:J1** keine einzige gerade Zahl geben, wäre das Ergebnis **0** (entspricht dem Startwert)). Abbildung 10 zeigt die Tabelle.

	A	B	C	D	E	F	G	H	I	J
1	10	77	64	83	97	14	81	12	23	37
2										
3	4		Formel in A3: =REDUCE(0;A1:J1;LAMBDA(a;b;WENN(ISTGERADE(b);a+1;a)))							

Abb. 10: *Beispiel 2 für die Funktion **REDUCE**, Anzahl gerader Zahlen ermitteln*

Anmerkung: Anstelle der Verwendung der Funktion **REDUCE** können Sie das Ergebnis auch mit der Formel **=SUMMENPRODUKT((A1:J1=GERADE(A1:J1))*1))** berechnen.

Funktion SCAN

Die Funktion **SCAN** ist vom Prinzip her identisch mit der Funktion **REDUCE** (siehe Kapitel **Funktion REDUCE**, Seite 11). Der Unterschied besteht lediglich darin, dass nicht nur ein Ergebnis am Ende herauskommt, sondern es werden auch alle Zwischenschritte berechnet und angezeigt. Das Ergebnis bei der Funktion **SCAN** ist also auch wieder eine Matrix. Die Syntax der Funktion **SCAN** lautet:

=REDUCE([Startwert];Matrix;LAMBDA(Akkumulator;Wert))

Die Funktionsargumente sind mit denen bei der Funktion **REDUCE** identisch (siehe Seite 11). Auch hier sollen zwei Beispiele die Ausführung der Funktion **SCAN** deutlicher machen.

Im ersten Beispiel werden die einzelnen Werte einer Matrix miteinander multipliziert, wobei immer nur der eine Wert mit dem nächsten Wert multipliziert wird. Die Werte befinden sich wieder im Zellbereich **A1:J1**. Im Zellbereich **A3:J3** sollen nun die Zwischenergebnisse ermittelt und angezeigt werden. Wählen Sie die Tabellenzelle **A3** und geben folgende Formel ein:

=SCAN(1;A1:J1;LAMBDA(a;b;a*b))

Nach Bestätigung der Formeleingabe wird der Zellbereich **B3:J3** automatisch mit den Zwischenergebnissen ausgefüllt (siehe Skript **Excel für Microsoft 365 – Funktionen (Matrix) und dynamische Arrays**, Kapitel **Dynamische Arrays**, Seite 12). Zunächst wird dem Argument **a** der Funktion **LAMBDA** wieder der Startwert (hier: **1**) zugewiesen. Das Argument **b** der Funktion **LAMBDA** bekommt nacheinander die Werte des Zellbereichs **A1:J1** zugewiesen. Dann wird **a*b** berechnet und sämtliche Zwischenergebnisse in den Zellbereich **A3:J3** eingetragen. Für den ersten Zwischenwert ergibt sich **a=1**; **b=1** ⇒ **a*b=1*1=1**. Für den zweiten Zwischenwert wird dem Argument **a** das Ergebnis des ersten Zwischenwerts zugewiesen, also **1** und dem Argument **b** der zweite Wert des Zellbereichs **A1:J1**. Daraus ergibt sich **a=1**; **b=2** ⇒ **a*b=1*2=2**. Dieses Ergebnis wird nun wieder dem Argument **a** zugewiesen und das Argument **b** bekommt den dritten Wert aus **A1:J1**, also den Wert **3**. Damit ergibt sich für den dritten Zwischenwert **a=2**; **b=3** ⇒ **a*b=2*3=6**. Und so setzt sich die Reihe fort. Das Ergebnis sehen Sie in der Abbildung 11.

	A	B	C	D	E	F	G	H	I	J
1	1	2	3	4	5	6	7	8	9	10
2										
3	1	2	6	24	120	720	5040	40320	362880	3628800
4										
5	Formel in A3: =SCAN(1;A1:J1;LAMBDA(a;b;a*b))									

Abb. 11: *Beispiel 1 für die Funktion **SCAN**, aktuellen Wert mit nächstem Wert multiplizieren*

Anmerkung: Anstelle der Verwendung der Funktion **SCAN** können Sie das Ergebnis auch mit der Formel **=A3*B1** berechnen. Die Formel geben Sie in die Tabellenzelle **B3** ein. In die Tabellenzelle **A3** geben Sie die Formel **=A1** ein. Nach der Eingabe der beiden Formeln, markieren Sie die Tabellenzelle **B3** und kopieren die Formel mit Hilfe des Verfahrens *Automatisches Ausfüllen* (siehe Skript **Excel für Microsoft 365 – Automatisches Ausfüllen**) in die Tabellenzelle **C3:J3**.

Im zweiten Beispiel werden Textteile miteinander verkettet. Vom Prinzip her funktioniert dieses Beispiel wie das erste Beispiel. Im Zellbereich **A1:A6** stehen die Wörter, die nacheinander verkettet werden sollen. In die Tabellenzelle **B1** wird nun folgende Formel eingegeben:

=SCAN("";A1:A6;LAMBDA(a;b;a&b&" "))

Nach Bestätigung der Formeleingabe wird die Formel automatisch in den Zellbereich **B2:B6** übertragen. Bei diesem Beispiel ist der Startwert die leere Zeichenfolge (durch Angabe von zwei doppelten Anführungszeichen direkt hintereinander). Das Argument **a** der Funktion **LAMBDA** hat also zu Beginn praktisch keinen Inhalt. Dem Argument **b** der Funktion **LAMBDA** wird nach und nach der Inhalt der Tabellenzellen **A1:A6** zugewiesen. Dann wird der Inhalt des Arguments **a** mit dem Inhalt des Arguments **b** verkettet und zusätzlich noch ein Leerzeichen angefügt (damit bei den anderen Zwischenergebnissen und dem Endergebnis zwischen den Wörtern auch Leerzeichen zu sehen sind). In Excel werden Textteile mit dem Zeichen **&** verkettet. Beim ersten Zwischenwert wird also die leere Zeichenfolge mit dem Wort **Das** und einem Leerzeichen (symbolisiert durch das Zeichen _) verkettet, also **a&b&"_"** ⇒ **""&"Das"&"_"** ⇒ **Das_**. Für das zweite Zwischenergebnis wird das erste Zwischenergebnis dem Argument **a** zugewiesen und das Argument **b** bekommt den Inhalt der Tabellenzelle **A2** zugewiesen. Zusammen ergibt das **a&b&"_"** ⇒ **"Das_"&"ist"&"_"** ⇒ **Das_ist_**. Jetzt wird das zweite Zwischenergebnis dem Argument **a** zugewiesen und das Argument **b** bekommt den Inhalt der Tabellenzelle **A3**. Das Ergebnis ist dann **Das_ist_ein_**. So setzt sich das Ganze bis zur Tabellenzelle **B6** fort. Das Ganze können Sie sich nochmal in Abbildung 12 anschauen.

	A	B	C	D	E	F	G	H	I
1	Das	Das		Formel in B1: =SCAN("";A1:A6;LAMBDA(a;b;a&b&" "))					
2	ist	Das ist							
3	ein	Das ist ein							
4	Beispiel	Das ist ein Beispiel							
5	für	Das ist ein Beispiel für							
6	Excel	Das ist ein Beispiel für Excel							

Abb. 12: *Beispiel 2 für die Funktion **SCAN**, Textteile verketten*

Anmerkung: Anstelle der Verwendung der Funktion **SCAN** können Sie das Ergebnis auch mit der Formel **=B1&A2&" "** berechnen. Die Formel geben Sie in die Tabellenzelle **B2** ein. In die Tabellenzelle **B1** geben Sie die Formel **=A1&" "** ein. Nach der Eingabe der beiden Formeln, markieren Sie die Tabellenzelle **B2** und kopieren die Formel mit Hilfe des Verfahrens *Automatisches Ausfüllen* (siehe Skript **Excel für Microsoft 365 – Automatisches Ausfüllen**) in die Tabellenzelle **B3:B6**.

Funktion MATRIXERSTELLEN

Mit der Funktion **MATRIXERSTELLEN** wird eine Matrix mit einer bestimmten Anzahl von Spalten und Zeilen erstellt. Die einzelnen Werte der Matrix werden mit der Funktion **LAMBDA** berechnet und in die entsprechenden Tabellenzellen eingetragen.

Die Syntax der Funktion **MATRIXERSTELLEN** lautet:

=MATRIXERSTELLEN(Zeilen;Spalten;LAMBDA(Zeile;Spalte;Berechnung))

Dabei haben die Funktionsargumente folgende Bedeutung:

Funktionsargument	***Beschreibung***
Zeilen	Die Anzahl der Zeilen in der Matrix. Der Wert muss größer als 0 sein.

Funktionsargument	*Beschreibung*
Spalten	Die Anzahl der Spalten der Matrix. Der Wert muss größer als 0 sein.
LAMBDA	Enthält den Zeilenindex, den Spaltenindex und die eigentliche Berechnung für die einzelnen Tabellenzellen des zugehörigen Zellbereichs.
Zeile	Der Zeilenindex der Matrix.
Spalte	Der Spaltenindex der Matrix.
Berechnung	Die Berechnung, deren Ergebnis jeweils in die entsprechenden Tabellenzelle des Zellbereichs eingetragen wird.

Auch hier soll wieder ein Beispiel die Funktionsweise der Funktion **MATRIXERSTELLEN** besser verdeutlichen. Für die Formeleingabe wird die Tabellenzelle **A1** ausgewählt und folgende Formel eingegeben:

```
=MATRIXERSTELLEN(3;4;LAMBDA(a;b;a*b))
```

Nach Bestätigung der Formeleingabe, werden die Ergebnisse automatisch von Excel in den Zellbereich **A1:D3** eingetragen (siehe Skript **Excel für Microsoft 365 – Funktionen (Matrix) und dynamische Arrays**, Kapitel **Dynamische Arrays**, Seite 12). In die Tabellenzellen **A1:D3** wird jeweils das Produkt aus dem Spaltenindex und dem Zeilenindex eingetragen. Beispielsweise steht in der Tabellenzelle **D2** der Wert **8**, da es sich um die 4. Spalte und die 2. Zeile der Matrix handelt. Der Zeilenindex hat den Wert **2** und der Spaltenindex den Wert **4**, also **2*4=8**. Abbildung 13 zeigt das Ergebnis der Berechnung.

	A	B	C	D	E	F
1	1	2	3	4		
2	2	4	6	8		
3	3	6	9	12		
4						
5	Formel in A1: =MATRIXERSTELLEN(3;4;LAMBDA(a;b;a*b))					

Abb. 13: *Beispiel für die Funktion **MATRIXERSTELLEN***

Anmerkung: Anstelle der Verwendung der Funktion **MATRIXERSTELLEN** können Sie auch folgende Lösung verwenden: markieren Sie den Zellbereich **A1:D3**, geben die Formel **=ZEILE(A1:D3)*SPALTE(A1:D3)** ein und bestätigen die Eingabe mit der Tastenkombination 

(siehe Skript **Excel-Microsoft 365 – Funktionen (Matrix) und dynamische Arrays**).

Im englischen hat die Funktion **MATRIXERSTELLEN** den Namen **MAKEARRAY**. Dieser Hinweis deshalb, weil in der Hilfe zur Funktion **MATRIXERSTELLEN** nur der englische Name verwendet wird.

Funktion NACHSPALTE

Bei der Funktion **NACHSPALTE** wird mit Hilfe der Funktion **LAMBDA** eine Berechnung für die Werte, die sich in jeder Spalte einer Tabelle oder Liste befinden, durchgeführt und die Ergebnisse in einem Zellbereich dargestellt.

Die Syntax der Funktion **NACHSPALTE** lautet:

=NACHSPALTE(Matrix;LAMBDA(Spalte;Berechnung))

Dabei haben die Funktionsargumente folgende Bedeutung:

Funktionsargument	*Beschreibung*
Matrix	Ein Zellbereich, der mindestens eine Spalte besitzt.
LAMBDA	Entnimmt aus jeder Spalte einen bestimmten Wert, der für die Berechnung verwendet wird.
Spalte	Name der Spalte aus dem angegebenen Zellbereich.
Berechnung	Ermittelt den Wert für die neue Matrix aus den einzelnen Werten der jeweiligen Spalte.

Auch hier wird die Ganze erst mit einem Beispiel deutlicher. Im Zellbereich **A1:F5** stehen verschiedene ganzzahlige Werte. In den Zellbereich **A7:F7** soll nun aus jeder Spalte der kleinste und größte Wert ermittelt und miteinander multipliziert werden. Dazu wählen Sie die Tabellenzelle **A7** aus und geben folgende Formel ein:

=NACHSPALTE(A1:F5;LAMBDA(a;MIN(a)*MAX(a)))

Nach Bestätigung der Eingabe werden die Tabellenzellen **B7:F7** automatisch von Excel mit den Ergebniswerten ausgefüllt (siehe Skript **Excel für Microsoft 365 – Funktionen (Matrix) und dynamische Arrays**, Kapitel **Dynamische Arrays**, Seite 12). In diesem Beispiel wird aus jeder Spalte der kleinste und der größte Wert ermittelt und miteinander multipliziert und das jeweilige Ergebnis in die Tabellenzellen **A7:F7** eingetragen. Abbildung 14 zeigt das Ergebnis.

	A	B	C	D	E	F	G
1	8	4	4	7	5	3	
2	9	3	9	8	3	4	
3	4	1	2	4	7	5	
4	3	7	6	2	4	7	
5	5	9	3	3	8	6	
6							
7	27	9	18	16	24	21	
8							
9	Formel in A7: =NACHSPALTE(A1:F5;LAMBDA(a;MIN(a)*MAX(a)))						

Abb. 14: *Beispiel für die Funktion **NACHSPALTE**, kleinster und größter Wert multiplizieren*

Anmerkung: Anstelle der Verwendung der Funktion **NACHSPALTE** können Sie auch folgende Lösung verwenden: wählen Sie zunächst die Tabellenzelle **A7** und geben dann die Formel **=MIN(A1:A5)*MAX(A1:A5)** ein und bestätigen die Eingabe mit der Taste ↵. Anschließend kopieren Sie die Formel in die Tabellenzellen **B7:F7** (siehe Skript **Excel-Microsoft 365 – Automatisches Ausfüllen**).

Im englischen hat die Funktion **NACHSPALTE** den Namen **BYCOL**. Dieser Hinweis deshalb, weil in der Hilfe zur Funktion **NACHSPALTE** nur der englische Name verwendet wird.

Funktion NACHZEILE

Die Funktion **NACHZEILE** ist prinzipiell mit der Funktion **NACHSPALTE** (siehe Kapitel **Funktion NACHSPALTE**, Seite 15) identisch. Diesmal wird aus jeder Zeile des Zellbereichs ein Wert für die Berechnung entnommen. Die Ergebniswerte werden dann in einen Zellbereich eingetragen.

Die Syntax der Funktion **NACHZEILE** lautet:

=NACHZEILE(Matrix;LAMBDA(Zeile;Berechnung))

Dabei haben die Funktionsargumente folgende Bedeutung:

Funktionsargument	*Beschreibung*
Matrix	Ein Zellbereich, der mindestens eine Spalte besitzt.
LAMBDA	Entnimmt aus jeder Zeile einen bestimmten Wert, der für die Berechnung verwendet wird.
Zeile	Name der Zeile aus dem angegebenen Zellbereich.
Berechnung	Ermittelt den Wert für die neue Matrix aus den einzelnen Werten der jeweiligen Zeile.

Auch hier wieder ein Beispiel, um den Einsatz der Funktion **NACHZEILE** zu zeigen. Im Zellbereich **A1:F5** stehen wieder verschiedene ganzzahlige Werte. In den Zellbereich **H1:H5** soll nun aus jeder Zeile der Mittelwert der jeweils sechs Zeilenwerte ermittelt werden. Dazu wählen Sie die Tabellenzelle **H1** aus und geben folgende Formel ein:

=NACHZEILE(A1:F5;LAMBDA(a;MITTELWERT(a)))

Nach Bestätigung der Formeleingabe kopiert Excel automatisch die restlichen Ergebnisse in die Tabellenzellen **H2:H5**. Das Ergebnis sehen Sie in Abbildung 15.

	A	B	C	D	E	F	G	H
1	482	787	323	929	154	552		537,83
2	661	663	399	865	192	344		520,67
3	326	864	794	386	777	814		660,17
4	833	639	271	930	111	902		614,33
5	592	107	988	354	640	124		467,50
6								
7	Formel in H1: =NACHZEILE(A1:F5;LAMBDA(a;MITTELWERT(a)))							

Abb. 15: *Beispiel für Funktion **NACHZEILE**, Mittelwert pro Zeile berechnen*

Anmerkung: Anstelle der Verwendung der Funktion **NACHZEILE** können Sie auch folgende Lösung verwenden: wählen Sie zunächst die Tabellenzelle **H1** und geben anschließend die Formel **=MITTELWERT(A1:F1)** ein und bestätigen die Eingabe mit der Taste ↵. Anschließend kopieren Sie die Formel in die Tabellenzellen **H2:H5** (siehe Skript **Excel-Microsoft 365 – Automatisches Ausfüllen**).

Im englischen hat die Funktion **NACHZEILE** den Namen **BYROW**. Dieser Hinweis deshalb, weil in der Hilfe zur Funktion **NACHZEILE** nur der englische Name verwendet wird.

Funktion WURDEAUSGELASSEN

Die Funktion **WURDEAUSGELASSEN** nimmt eine Sonderstellung unter den Unterstützerfunktionen ein. Mit ihr können Sie bei der Funktion **LAMBDA** überprüfen, ob ein Argument für die eigentliche Berechnung fehlt.

Die Syntax der Funktion **WURDEAUSGELASSEN** lautet:

=WURDEAUSGELASSEN(Argument)

Dabei haben die Funktionsargumente folgende Bedeutung:

Funktionsargument	*Beschreibung*
Argument	Das Argument für die Funktion LAMBDA, das überprüft wird, ob es vorhanden ist oder nicht.

Wenn Sie folgende Formel beispielsweise in die Tabellenzelle **A1** eingeben, erhalten Sie als Ergebnis **Das zweite Argument fehlt**:

=LAMBDA(x;y;WENN(WURDEAUSGELASSEN(y);"Das zweite Argument fehlt";x+y))(1;)

Die Funktion **WURDEAUSGELASSEN** können Sie beispielsweise sinnvoll einsetzen, wenn Sie in der Funktion **LAMBDA** optionale Funktionsargumente verwenden (siehe Kapitel **Optionale Funktionsargumente**, Seite 7).

Anmerkung: Im englischen hat die Funktion **WURDEAUSGELASSEN** den Namen **ISOMITTED**. Dieser Hinweis deshalb, weil in der Hilfe zur Funktion **WURDEAUSGELASSEN** nur der englische Name verwendet wird.

Die Funktion **WURDEAUSGELASSEN** gibt es nur in der Version Excel für Microsoft 365.

Zusatzinformationen

Wenn Sie sich die Unterstützerfunktionen genauer angeschaut haben, insbesondere die Anmerkungen zu den meisten Unterstützerfunktionen, dann werden Sie sich vielleicht schon die Frage gestellt haben, warum soll ich so komplizierte Formeln verwenden, wenn es einfachere Formel zum Lösen des Problems gibt? Beispielsweise ist die Formel **=UND(A2;B2)** wesentlich verständlicher und kürzer als die Formel **=MAP(A2:A8;B2:B8;LAMBDA(a;b;UND(a;b)))**. Der Vorteil der Unterstützerfunktionen wird erst bei komplexen Problemstellungen deutlich, wenn Sie das Problem mit den „normalen“ Excel-Funktionen nicht mehr lösen können. Ebenfalls von Vorteil sind die dynamischen Arrays. Wenn es sich bei dem Ergebnis einer Unterstützerfunktion um eine Matrix handelt, müssen Sie keinen Zellbereich markieren, dann die Formel eingeben und mit der Tastenkombination Strg ⇧ ↵ bestätigen. Es reicht aus, eine Tabellenzelle auszuwählen, die Formel einzugeben und mit der Taste ↵ zu bestätigen. Das ist insbesondere dann von Vorteil, wenn Sie gar nicht im Vorfeld wissen,

wie umfangreich die Zielmatrix sein wird, die Sie benötigen, um alle Ergebniswerte darzustellen (das gilt in erster Linie für den Aufbau der Zielmatrix – Anzahl der Spalten und Zeilen). Alle anderen benötigten Tabellenzellen werden von Excel automatisch ausgefüllt. Insbesondere bei sehr großen Tabellen, wo die Ergebnismatrix einen sehr großen Zellbereich umfasst, ist dieser Weg deutlich von Vorteil.

Ein weiterer Vorteil wird erst dann sichtbar, wenn Sie den Zellbereich mit den Ausgangsdaten in eine Tabelle umwandeln (siehe Skript **Excel für Microsoft 365 – Tabellen**). Wird die Tabelle nachträglich durch weitere Werte erweitert, passt sich auch der Zellbereich mit der Zielmatrix automatisch an. Das passiert nicht, wenn Sie nur die „normalen" Excel-Funktionen einsetzen. Auch hierzu ein Beispiel. Nehmen wir das zweite Beispiel für die Funktion **MAP** (siehe Abbildung 8, Seite 11). Die Liste im Zellbereich **A1:B8** wird in eine Tabelle umgewandelt (beliebige Tabellenzelle im Zellbereich **A1:B8** auswählen, dann im Register **Einfügen** in der Gruppe ***Tabellen*** das Symbol **Tabelle** anklicken und das anschließende Dialogfeld bestätigen). Im Zellbereich **D2:D8** wird die Funktion **MAP** eingesetzt (genaue Formel siehe Seite 11). Im Zellbereich **F2:F8** wird dasselbe Ergebnis mit der Funktion **UND** berechnet (Formel: **=UND(A2;B2)**, **=UND(A3;B3)**, **=UND(A4;B4)**, usw.; siehe Abbildung 16).

	A	B	C	D	E	F
1	Wert 1	Wert 2		Wert 1 UND Wert 2		Wert 1 UND Wert 2
2	FALSCH	FALSCH		FALSCH		FALSCH
3	WAHR	FALSCH		FALSCH		FALSCH
4	WAHR	WAHR		WAHR		WAHR
5	WAHR	FALSCH		FALSCH		FALSCH
6	FALSCH	WAHR		FALSCH		FALSCH
7	FALSCH	FALSCH		FALSCH		FALSCH
8	WAHR	WAHR		WAHR		WAHR

Abb. 16: *Beispiel für die Funktion **MAP** mit einer Tabelle, Ausgangssituation*

Jetzt wählen Sie die Tabellenzelle **A9** aus und geben einen beliebigen Wahrheitswert ein (es spielt wirklich keine Rolle, welchen Wahrheitswert Sie eingeben). Das Gleiche machen Sie anschließend mit der Tabellenzelle **B9**. Sobald Sie den ersten Wahrheitswert in Tabellenzelle **A9** eingegebenen haben, wird die Tabelle automatisch erweitert und Sie sehen auch bereits in der Tabellenzelle **D9** ein Ergebnis. Die Tabellenzelle **F9** bleibt dagegen leer. Hier wurde die Formel nicht aus der vorherigen Tabellenzelle **F8** übernommen (siehe Abbildung 17). Natürlich können Sie mit dem Verfahren *Automatisches Ausfüllen* (siehe Skript **Excel für Microsoft 365 – Automatisches Ausfüllen**) die Formel aus der Tabellenzelle **F8** in die Tabellenzelle **F9** kopieren. Aber das ist zusätzlicher Aufwand.

	A	B	C	D	E	F
1	Wert 1	Wert 2		Wert 1 UND Wert 2		Wert 1 UND Wert 2
2	FALSCH	FALSCH		FALSCH		FALSCH
3	WAHR	FALSCH		FALSCH		FALSCH
4	WAHR	WAHR		WAHR		WAHR
5	WAHR	FALSCH		FALSCH		FALSCH
6	FALSCH	WAHR		FALSCH		FALSCH
7	FALSCH	FALSCH		FALSCH		FALSCH
8	WAHR	WAHR		WAHR		WAHR
9	WAHR	FALSCH		FALSCH		

Abb. 17: *Beispiel für die Funktion **MAP** mit einer Tabelle, neue zusätzliche Zeile*

In den ganzen gezeigten Beispielen für die Unterstützerfunktionen wurden bei der jeweiligen Funktion **LAMBDA** für die Parameter immer die Buchstaben **a** bzw. **b** verwendet. Das ist in diesem Fall reiner Zufall. Sie können den Parametern beliebige andere Namen geben. Nehmen wir als Beispiel die Formel für die Funktion **NACHZEILE** (siehe Seite 17). Anstelle des Buchstabens **a** könnten Sie beispielsweise das Wort **Zeile** verwenden. Dadurch wirkt die gesamte Formel nicht ganz so abstrakt. Die Formel hat dann folgendes neues Aussehen:

=NACHZEILE(A1:F5;LAMBDA(Zeile;MITTELWERT(Zeile)))