

---

# Tabellenkalkulation: Logikfunktionen

---

LES Pforzheim

Januar 2019

## Inhaltsverzeichnis

<b>1 Logikfunktionen: Einstieg</b>	<b>1</b>
1.1 Tests . . . . .	1
1.2 wenn() . . . . .	3
1.3 Formeln mit wenn() formulieren . . . . .	3
1.4 wenn() innerhalb komplexer Formeln . . . . .	4
<b>2 Anwendungen</b>	<b>4</b>
2.1 min() und max() . . . . .	4
2.2 Mehrere Tests kombinieren . . . . .	5
2.3 Eingaben prüfen und Fehler abfangen . . . . .	6
2.4 Zahlenkolonnen selektiv auswerten . . . . .	6
<b>3 Werte in einer Liste nachschlagen: SVerweis()</b>	<b>7</b>
<b>4 Optimierungen</b>	<b>9</b>

## 1 Logikfunktionen: Einstieg

Mit Logikfunktionen kann sich ein Arbeitsblatt flexibel an Bedingungen anpassen, mit Entscheidungen ist weit mehr möglich als mit linearen Berechnungen. Die meisten Logikfunktionen ermitteln *WAHR* oder *FALSCH* als Ergebnis ihrer Berechnungen. Falls dieses Ergebnis nicht direkt nützlich ist, kann man es mit `wenn()` weiter verarbeiten.

### 1.1 Tests

Nicht alle mathematischen Operatoren lassen sich einfach eintippen, deswegen gibt es in der Tabellenkalkulation (und in Programmiersprachen) Ersatzdarstellungen.

<i>Operator</i>	<i>Eintippen</i>	<i>Bedeutung</i>
>	>	größer als
≥	>=	mindestens so groß wie
<	<	kleiner als
≤	<=	höchstens so groß wie
=	=	gleich
≠	<>	verschieden

Bei Textaufgaben sollten Sie sorgfältig darauf achten, welcher Vergleichsoperator gemeint ist. Wenn Sie versehentlich keinen Vergleichsoperator angeben, wird implizit mit null verglichen. Vermeiden Sie das.

Falls Sie keinen passenden Operator haben, können Sie zum Negieren von Vergleichen `nicht()` innerhalb des Test-Teils verwenden oder einfach den *Dann*- und *Sonst*-Teil vertauschen. Als Ergänzung zum Notenbeispiel: Wenn die Note nicht besser als 4,0 ist, gibt es einen Tadel.

```

anderer Operator  =wenn(B1 >= 4,0; "Tadel"; "")
nicht()           =wenn(nicht(B1 < 4,0); "Tadel"; "")
Teile vertauscht  =wenn(B1 < 4,0; ""; "Tadel")

```

**Textvergleich** Operatoren wie »größer« beziehen sich auf die alphabetische Reihenfolge. Excel unterscheidet nicht zwischen Groß- und Kleinschreibung, für LibreOffice kommen Großbuchstaben vor Kleinbuchstaben. Denken Sie daran, dass Texte innerhalb von Formeln immer in Anführungszeichen müssen.

**Auf Fehler prüfen** Es gibt eine Reihe von Funktionen, mit denen Fehler oder bestimmte Zellzustände im Arbeitsblatt abgefragt werden können. Typische Beispiele:

`IstFehler()` : prüft, ob in einer Zelle ein Fehler vorliegt. Beispiel: Der folgende Befehl liefert *WAHR*, wenn die Formel in **B5** falsch ist.  
`IstFehler(B5)`

`IstLeer()` : prüft, ob eine Zelle leer ist.

`IstGerade()` : prüft, ob eine Zahl gerade ist

`IstFormel()` : prüft, ob eine Zelle eine Formel enthält

`IstNV()` : prüft, ob ein Wert in einer Liste für `SVerweis()` vorkommt

**Tests verwenden** Sie können Tests direkt in eine Zelle eintippen; denken Sie dabei an das = am Anfang. `=3<7` sollte *WAHR* liefern.

**Mehrere Tests verknüpfen** Es gibt zwei Funktionen, mit denen Sie Tests verknüpfen können. Sie können aber auch Konstruktionen aus mehreren `wenn()`-Befehlen machen.

`Oder()` : liefert *WAHR*, wenn mindestens einer der aufgeführten Tests zutrifft. Beispiel: Prüfen, ob mit dem Text in B5 eine Frau gemeint ist.  
`=Oder(B5="F"; B5="Frau"; B5="W"; B5="Fr1.")`

`Und()` : liefert *WAHR*, wenn alle aufgeführten Tests zutreffen. Beispiel: Prüfen, ob in B5 eine gerade Zahl zwischen 20 und 40 liegt.  
`=Und(IstGerade(B5); B5 >= 20; B5 <= 40)`

## 1.2 wenn()

Der Befehl `wenn()` wählt je nach Ausgang eines Tests eine von zwei Berechnungen aus. Damit kann man das *WAHR* oder *FALSCH* in Formeln, Texte oder anderes umsetzen. `wenn()` benötigt daher drei Argumente, die (wie immer) durch Strichpunkte getrennt werden. Das erste Argument ist der *Test*, das zweite der *Dann*-Teil, das dritte der *Sonst*-Teil.

`=wenn(Test; Dann; Sonst)`

Wenn der *Dann*- oder *Sonst*-Teil leer bleiben soll, tippen Sie dort zwei Anführungszeichen ein, um die Ausgabe von *WAHR* oder *FALSCH* zu unterdrücken. Beispiel: Bei einer Note (in Zelle B1) besser als 2,0 loben.

`=wenn(B1 < 2,0; "Lob"; "")`

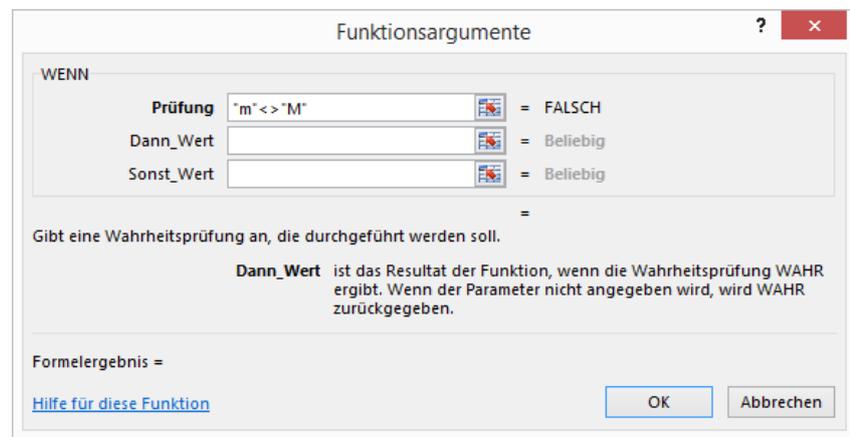


Abbildung 1: Das *FALSCH* kommt daher, dass der Test gleich ausgewertet wird. Es ist kein Fehler in Ihrer Formel.

Vor Vergleichen sollten Sie Ihre Daten vielleicht korrigieren. Bei Rundungsdifferenzen kann es sinnvoll sein, gerundete Zahlen zu vergleichen. Bei Textvergleichen möchten Sie den Text vielleicht vorher aufbereiten, indem Sie ihn in Großbuchstaben umwandeln, kürzen oder sonstwie behandeln.

## 1.3 Formeln mit `wenn()` formulieren

Falls Sie sich mit dieser Funktion schwertun, gehen Sie so vor:

1. Formulieren Sie die Aufgabenstellung in einen Satz um, der mit »Wenn« beginnt.
2. Verfeinern Sie den Satz, indem Sie den richtigen Vergleichsoperator verwenden.
3. Überlegen Sie, welche Zellen oder Werte in Ihrer Formel vorkommen.
4. Setzen Sie die Formel in der Tabellenkalkulation um.

Gleich ein Beispiel: *Ab 100 Euro gibt es 5 % Rabatt.*

1. *Wenn* der Rechnungsbetrag ab 100 Euro ist, gibt es 5 % Rabatt, sonst gibt es keinen Rabatt.

2. *ab* bedeutet  $\geq$ , also:  
Wenn der Rechnungsbetrag  $\geq 100$  ist, gibt es 5%, sonst 0%.
3. Angenommen, der Rechnungsbetrag steht in B2 und alle anderen Zahlen tippen Sie ein. Dann fängt Ihr Satz so an: *Wenn B2 größer oder gleich 100 ist, ...*
4. Und Ihre Formel sieht ungefähr so aus:  
`=wenn(B2 >= 100; 5%, 0%)`

Falls Sie den Rabatt gleich berechnen müssen, laufen die Überlegungen ein wenig anders:

1. Wenn der Rechnungsbetrag ab 100 Euro ist, gibt es 5% Rabatt, sonst gibt es keinen Rabatt.
2. *ab* bedeutet  $\geq$ , also:  
Wenn der Rechnungsbetrag  $\geq 100$  ist, liegt der Rabatt bei 5% vom Rechnungsbetrag, sonst bei 0 Euro.
3. Angenommen, der Rechnungsbetrag steht in B2 und alle anderen Zahlen tippen Sie ein. Ihr Satz lautet jetzt:  
*Wenn B2  $\geq 100$  ist, ist der Rabatt B2 \* 5%, sonst 0.*
4. Ihre Formel sieht dann ungefähr so aus:  
`=wenn(B2 >= 100; B2 * 5%, 0)`

#### 1.4 wenn() innerhalb komplexer Formeln

wenn() muss nicht alleine in einer Zelle stehen, es kann auch innerhalb einer Berechnung verwendet werden. Im folgenden Beispiel sind die Formeln in B3 und B4 gleichwertig. Sie dürfen sich für die entscheiden, mit der Sie besser zurecht kommen. Wenn beide Berechnungen bis auf einen Wert identisch sind, ist die »ausgeklammerte« Form übersichtlicher.

	A	B	C	D	E	F
1	Nettopreis	20,00		Steuersatz 1	19%	
2	Steuersatz (1 oder 2)	1		Steuersatz 2	7%	
3	Mehrwertsteuer	<code>=wenn(B2 = 1; B1 * E1; B1 * E2)</code>				
4	Mehrwertsteuer	<code>=B1 * wenn(B2 = 1; E1; E2)</code>				

## 2 Anwendungen

### 2.1 min() und max()

Diese beiden Befehle sind keine Logikfunktionen, Sie können sie aber immer dann einsetzen, wenn die kleinste oder größte von mehreren Zahlen benötigt wird. min() liefert den kleinsten Wert aus einer Liste, max() den größten. Einfaches Beispiel:

`=Min(4; 2; 12; 7; 23; 11)`

`=Max(4; 2; 12; 7; 23; 11)`

Die erste Formel liefert 2, die zweite 23.

Zwei Beispiele: Im Handyvertrag sind 3 GB Daten enthalten, jedes weitere GB kostet 20 €. Die verbrauchten GB stehen in B1.

Methode	Formel
wenn()	<code>=Grundgebühr+ wenn(B1 &lt; 3; 0; 20 * (B1 - 3))</code>
max()	<code>=Grundgebühr+ max(0; 20 * B1)</code>

Ein Prepaid-Vertrag fürs Handy hat eine Kostendeckelung. Pro Gesprächsminute werden 10 Cent bezahlt, höchstens jedoch 30 Euro im Monat. Die Minuten stehen in B1.

Method	Formel
wenn()	= wenn(B1 * 0,10 < 30; B1 * 0,10; 30)
max()	= min(B1 * 0,10; 30)

## 2.2 Mehrere Tests kombinieren

Es gibt viele Arten, mehrere Tests zu kombinieren, davon hängt die Umsetzung ganz maßgeblich ab.

**Voneinander unabhängige Tests** Wenn Sie einen Eisbecher kaufen, können Sie gegen Aufpreis Soße (1,00 €), Schokostreusel (0,50 €) oder Sahne (1,20 €) in beliebiger Kombination haben. Die Testergebnisse werden einfach addiert.

=Eispreis+ wenn(will Soße;1,00;0) + wenn(will Streusel;0,50;0)  
+ wenn(will Sahne;1,20;0)

**Alle Tests müssen zutreffen** Sie bekommen ein Lob, wenn Sie einen Notenschnitt besser zwei und ein tadelloses Verhalten haben. Der Folgetest wird im *Dann*-Teil des ersten Tests vorgenommen. Alternative: Verwenden von **und()** im Test des **wenn()** erspart das mehrfache Nennen des *Sonst*-Teils. Spätestens bei vier Bedingungen sind Sie mit **und()** deutlich besser darn: Weniger Klammern, Ihre Absicht ist klarer, einfachere Erweiterbarkeit.

=wenn(und(Note < 2;  
Verhalten ok); "Lob"; "")

=wenn(Note < 2;wenn(Verhalten ok; "Lob"; ""); "")

**Mindestens ein Test muss zutreffen** Sie bekommen ermäßigten Eintritt ins Theater, wenn Sie zur Schule gehen, Rente beziehen oder behindert sind. Der Folgetest wird im *Sonst*-Teil des ersten Tests vorgenommen. Alternative: Verwenden von **oder()** im Test des **wenn()** erspart das mehrfache Nennen des *Dann*-Teils. Auch hier gilt: Je komplizierter es wird, desto weiter kommen Sie mit **oder()**.

=wenn(oder(Schule; Rente;  
Behinderung);  
Rabatt; kein Rabatt)

=wenn(Schule; Rabatt;wenn(Rente; Rabatt;  
wenn(Behinderung; Rabatt; kein Rabatt)))

**Genau ein Test muss zutreffen** In älteren Excel-Versionen lässt sich das nicht vernünftig formulieren. Inzwischen gibt es die Funktion **xxoder()**, die im Unterschied zu **oder()** nur dann *WAHR* liefert, wenn genau eine Bedingung zutrifft. Ein Beispiel wäre: Wenn Sie in einem Prüfungsfach eine Fünf haben, können Sie eine Zusatzprüfung ablegen. Bei mehreren Fünfen sind Sie durchgefallen.

**Ein einzelner Wert wird mit einer Liste verglichen** Dies ist ein Sonderfall des »Mindestens ein Test...«. Wenn die Liste etwas länger ist, verwenden Sie **sverweis()** statt **wenn()**.

## 2.3 Eingaben prüfen und Fehler abfangen

Mit Befehlen wie `istLeer()` oder `istZahl()` können Sie Benutzereingaben prüfen. Mit `wenn()` machen Sie daraus Anweisungen für den Benutzer.

```
=wenn(istLeer(B1); "In B1 fehlt Ihre Eingabe."; "")
```

```
=wenn(istZahl(B1); ""; "In B1 gehört eine Zahl.")
```

`wenn()` eignet sich auch zum Abfangen von Fehlern in Arbeitsblättern. Typisches Beispiel: In B1 steht eine Funktion, die bei manchen Eingaben kein gültiges Ergebnis liefert. Das können Sie mit Funktionen wie `istFehler()` erkennen und mit `wenn()` in entsprechende Handlungen umsetzen.

```
=wenn(istFehler(B1); "Bitte Eingabe korrigieren"; "")
```

Am einfachsten ist es, wenn Sie Ihre Benutzer direkt beim Eintippen nur gültige Werte auswählen lassen. Hierfür brauchen Sie keine Formeln, Sie machen Vorgaben mit *Datentools/Gültigkeit*. Wenn Sie mit diesem Menüpunkt auch noch Hilfetexte ergänzen, bekommen Sie glückliche Benutzer.

Da man `=wenn(istFehler(...))` öfters braucht, gibt es die Komfortfunktion `WennFehler()`. Beispiel: In B2 und B3 stehen zwei Streckenlängen, der Tangens soll berechnet werden und aus dem Tangens dann der Winkel. Für den Tangens werden die beiden dividiert. Falls die zweite Länge gleich 0 ist, ist der Winkel 0 Grad. Die Division ergibt jedoch einen Fehler. In B6 steht dann entweder der in B5 berechnete Wert oder eben 0.

	A	B	C
1	<i>Bezeichnung</i>		<i>Wert</i>
2	Länge X		4,00
3	Länge Y		2,00
4	Quotient		=B2/B3
5	Winkel		=ATan(B4)*360/(2*PI())
6	Abgefangen		=WennFehler(B5;0)

## 2.4 Zahlenkolonnen selektiv auswerten

Mit `SummeWenn()` und `ZählenWenn()` können Sie gezielt Werte aus einer Liste heraus addieren oder zählen. Beispiel: Auf einer Rechnung stehen Produkte zu verschiedenen Steuersätzen. Sie möchten nur Anzahl und Summe von Produkten mit Steuersatz 1.

	A	B	C	D	E
1	<i>Bezeichnung</i>	<i>Nettopreis</i>	<i>Steuersatz</i>		
2	Brot	2,00	2		
3	Sprudel	1,00	1		
4	Käse	2,50	2		
5	Batterie	1,20	1		
6	<b>Auswertung</b>	für Steuersatz		1	
7	Summe	=SummeWenn(C2:C5; D6; B2:B5)			
8	Anzahl	=ZählenWenn(C2:C5; D6)			

Die Formeln in Zeile 7 und 8 kann man 1 : 1 in die Umgangssprache übertragen: Wenn im Bereich von C2 bis C5 das gleiche steht wie in Zelle D6 (in diesem Fall die 1), dann addiere die Zahl aus dem Bereich

B2 bis B5. Wenn im Bereich von C2 bis C5 das gleiche steht wie in Zelle D6 (in diesem Fall die 1), dann zähle die jeweilige Zeile mit.

Wenn Sie Vergleichsoperatoren für diese beiden Funktionen benötigen, wird es etwas komplizierter, Sie müssen den Vergleich dann in Anführungszeichen setzen. Beispiel: Artikel ab 2 Euro zählen:

`=ZählenWenn(C2:C5; ">=2")`

### 3 Werte in einer Liste nachschlagen: **SVerweis()**

Sie haben eine Tabelle von Werten. In der ersten Spalte steht der Wert, nach dem die Tabellenzeile ausgewählt wird, in weiteren Spalten stehen die Werte, die Sie in Abhängigkeit von der ersten Spalte in Berechnungen verwenden möchten. **SVerweis()** kann für exakte Übereinstimmung oder für Bereiche eingestellt werden. Beispiele:

- In der ersten Spalte steht eine Bestellnummer, in den Folgespalten Artikelname und so weiter. Hier ist exakte Übereinstimmung wichtig. Letztes Argument für **SVerweis()** ist **0**.
- In der ersten Spalte steht eine Körpergröße, in der Folgespalte die passende Konfektionsgröße für Kleidung. Hier brauchen Sie Bereichsauswahl. Letztes Argument für **SVerweis()** ist **1**, zudem müssen die Werte der ersten Spalte aufsteigend sortiert sein.

`=SVerweis(Wert; Tabelle; Ergebnisspalte; 0 oder 1)`

Geben Sie der Tabelle, in der nachgeschlagen wird, am besten einen symbolischen Namen (Adresskästchen links oben), falls Sie das nicht machen möchten, sind absolute Adressen (Dollarzeichen) empfehlenswert. Nehmen Sie die Überschriften *nicht* mit hinein. Die Ergebnisspalte wird nicht mit den üblichen Spaltenkennungen (A, B, C ...) angegeben, sondern mit einer Zahl; die erste Spalte Ihrer Nachschlagetabelle bekommt die 1.

**SVerweis()** ist mit Sicherheit besser als ein dreifach oder noch öfter verschachteltes **wenn()**. Neben der Übersichtlichkeit ist es auch sehr einfach, die Tabelle bei Bedarf zu erweitern. Falls die Tabelle mehr als zwei Spalten hat, ist **SVerweis()** auch schon bei zwei Fällen die bessere Wahl als **wenn()**.

**Kleidergrößen** Kinderkleidung geht nach der Körpergröße. Von 81 bis 86 cm wird Größe 86 verwendet, dann bis 92 cm Größe 92, weiter geht es mit 98, 104 und so fort.

Sie erstellen ein Arbeitsblatt wie folgt:

	A	B	C	D	E	F
1	Körpergröße (cm)			<i>Größe (cm) ab</i>	<i>Kleidergröße</i>	
2	Kleidergröße			0	zu klein	
3				81	86	
4				87	92	
5				93	98	
6				99	104	
7				105	zu groß	

Die Zahl, die wir nachschlagen wollen, steht in B1. Die Tabelle, in der nachgeschlagen wird, steht in D2 bis E7 (ohne Überschriften). Wir

sind an der zweiten Spalte unserer Tabelle interessiert. Die Werte der ersten Spalte sind sortiert, wir möchten bereichsweise nachschlagen, also wird das letzte Argument für `SVerweis()` eine 1. Damit ergibt sich in B2 folgende Formel:

```
=SVerweis(B1; $D$2:$E$7; 2; 1)
```

**Artikeltabelle** Firma Cherry stellt MX-Schalter für hochwertige Tastaturen her. Die Schalter haben unterschiedliche Eigenschaften und unterschiedliche Farbcodes.

	A	B	C	D	E
1	<i>Farbe</i>	<i>Klick</i>	<i>Kraft</i>	<i>taktil</i>	
2	rot	nein	45	nein	
3	schwarz	nein	60	nein	
4	braun	nein	55	ja	
5	blau	ja	60	ja	
6					
7	Farbe:				
8	Klick:		taktil:		

In B6 wird eine Farbe eingetippt. Die Nachschlagetabelle steht in A1 bis D4. In B8 soll stehen, ob die Taste klickt (zweite Spalte der Nachschlagetabelle) und in D8, ob sie taktiles Feedback gibt (vierte Spalte). Die erste Spalte der Nachschlagetabelle ist nicht sortiert, nicht aufgeführte Werte (z.B. grün) sind ungültig. In D8 kommt folgende Formel:

```
=SVerweis(B7; $A$2:$D$5; 4; 0)
```

Die Formel in B8 sieht genauso aus, mit einem Unterschied: statt der 4 steht als drittes Argument eine 2.

**Abfangen ungültiger Eingaben** Mit der Funktion `istNV()` (*nicht vorhanden*) können Sie prüfen, ob der Benutzer in B6 ein gültiges Wort eingegeben hat.

```
=istNV(D8)
```

Geprüft wird dabei nicht die Eingabe selbst, sondern man schaut nach, ob `SVerweis()` etwas mit der Eingabe anfangen kann.

`istNV()` für sich erzeugt nur *WAHR* oder *FALSCH*, für einen Benutzerhinweis müssen Sie es in `wenn()` einpacken.

```
=wenn(istNV(D8); "ungültige Farbe"; "")
```

Wenn das Ergebnis der Gültigkeitsprüfung in B8 gezeigt werden soll, wird es etwas komplizierter, da `SVerweis()` dann zwei Mal – innerhalb `istNV()` (zur Prüfung auf Gültigkeit) und innerhalb des *sonst*-Teils von `wenn()` (zur Anzeige des Wertes) – angeführt werden muss:

```
=wenn(istNV(SVerweis(B7; $A$2:$D$5; 2; 0));  
"ungültige Farbe"; SVerweis(B7; $A$2:$D$5; 2; 0))
```

Widerstrebt es Ihnen, die eigentliche Berechnung im *sonst*-Teil auszuführen, können Sie das Ergebnis von `istNV()` mit `nicht()` invertieren:

```
=wenn(nicht(istNV(SVerweis(B7; $A$2:$D$5; 2; 0)));  
SVerweis(B7; $A$2:$D$5; 2; 0); "ungültige Farbe")
```

Falls Sie solche Ausdrücke entziffern müssen, gehen Sie sie am besten von innen nach außen vor. Der Funktionsassistent von LibreOffice kommt – im Unterschied zu Excel – auch mit verschachtelten Funktionen klar.

Den zweimaligen Aufruf von `SVerweis()` können Sie vermeiden, indem Sie das Ergebnis von `SVerweis()` verstecken und an der Stelle, wo Sie es benötigen, `IstFehler()` verwenden. Im Beispiel lassen wir den `SVerweis()` in D8 und setzen die Prüfung anderswo hin.  
`=wennFehler(D8;"ungültige Farbe")`

## 4 Optimierungen

Mathematiksoftware löst Gleichungssysteme durch symbolische Umformungen – Tabellenkalkulationen können das nicht. Tabellenkalkulationen können immerhin systematisch Werte in ein Arbeitsblatt einsetzen, bis eine akzeptable Lösung gefunden wird. Die *Zielwertsuche* eignet sich, wenn eine einzige Zahl im Arbeitsblatt verändert werden soll und das Ergebnis ebenfalls in einer Zahl in einer einzelnen Zelle steht. Die Zielwertsuche kann schnell uns spontan eingesetzt werden. Mit dem *Solver* können Sie komplexere Aufgaben lösen, indem Sie Nebenbedingungen vorgeben, die möglichen Werte auf positive ganze Zahlen beschränken und so weiter. Der Solver muss in Excel gesondert installiert werden.

**Eigenfertigung oder Fremdbezug** Bei Eigenfertigung entstehen Fixkosten von 10 000 Euro für eine Maschine und variable Kosten von 10 Euro pro gefertigtem Teil. Bei Fremdbezug entstehen keine fixen Kosten, die variablen Kosten liegen pro Teil bei 20 Euro. Offensichtlich ist der Fremdbezug bei kleinen Stückzahlen günstiger, die Eigenfertigung bei großen. Wo liegt die Menge, bei der beides den selben Gesamtpreis ergibt?

	A	B	C	D
1		<i>Eigenfertigung</i>	<i>Fremdbezug</i>	
2	<i>Fixkosten</i>	10.000,00	0,00	
3	<i>Stückkosten</i>	10,00	20,00	
4	<i>Stückzahl</i>	1.000		
5	<i>Gesamtkosten</i>	= B2 + B3 * \$B4	= C2 + C3 * \$B4	
6	<i>Differenz</i>	= B5 - C5		

Entscheidend für die Zielwertsuche ist der Setup: Das Ergebnis muss sich in *einer* einzigen Zelle abbilden – hier wurde die Differenz der Kosten für beide Arten gewählt; ebenso darf es nur *eine* Zelle geben, deren Wert durchprobiert wird. Hier ist es die Stückzahl. Zielzelle für die Zielwertsuche ist **B6**, der gesuchte Wert ist 0 und die veränderbare Zelle ist **B4**. Im Unterschied zu Excel-Funktionen läuft die Zielwertsuche nicht ständig mit. Wenn Sie Fixkosten oder Stückkosten ändern, muss sie erneut aufgerufen werden. Wenn die Zielwertsuche kein exakt passendes Ergebnis findet, bietet sie einen Näherungswert an. Das ist oft akzeptabel.