

# Was ist ein Minimalbeispiel?

Christian Faulhammer

Version: 0.6

2009-08-18

## Inhaltsverzeichnis

<b>1</b>	<b>Was ist ein Minimalbeispiel überhaupt?</b>	<b>2</b>
<b>2</b>	<b>Die Werkzeuge</b>	<b>3</b>
2.1	Verschieben des Endes . . . . .	3
2.1.1	Eine Datei . . . . .	3
2.1.2	Zusammengesetzte Dateien . . . . .	4
2.2	Vereinfachungen . . . . .	4
2.2.1	Unnötige Pakete entfernen . . . . .	4
2.2.2	Eigene Befehle und Umgebungen . . . . .	4
2.2.3	Schrumpfen . . . . .	5
2.2.4	Grafiken entfernen . . . . .	5
2.2.5	Benötigte zusätzliche Dateien . . . . .	5
<b>3</b>	<b>Beispiele</b>	<b>6</b>
3.1	Syntaxfehler . . . . .	6
3.2	Unerwünschtes Verhalten . . . . .	7
3.3	Unerklärliches Verhalten :) . . . . .	9
3.4	Stolpersteine außerhalb des Quelltexts . . . . .	10
3.4.1	Ärger mit Editoren . . . . .	10
3.4.2	Probleme, die einfach nicht verschwinden . . . . .	11

Dieses Dokument entstand aus dem Bedürfnis heraus, für Neuankömmlinge in der Newsgroup `de.comp.text.tex` eine Beispielsammlung zu schaffen, die das Prinzip des Minimalbeispiels erläutert. Der Text ist Public Domain, die Version für die Newsgroup wird momentan von Christian Faulhammer gepflegt und in der regelmässig veröffentlichten Einführung erwähnt. Korrekturen, Änderungswünsche und Kommentare bitte an `christian(at)faulhammer.org` senden.

Wer ausführliche Informationen zur Interpretation von Fehlermeldungen haben möchte, sollte einen Blick in den „`LATEX` Companion“ werfen, der aber für Anfänger nicht direkt geeignet ist.

## Danksagungen

Danken möchte ich allen Teilnehmern der Newsgroup, die immer wieder darauf hinweisen, wie wichtig ein Minimalbeispiel ist und wertvolle Hinweise zum Erstellen desselben liefern. Speziell möchte ich Ulrich Schwarz für die Übersetzung ins Englische, Hilmar Preuße für die erste `LATEX`-Version und Kurt Lidwin für das Überarbeiten derselbigen erwähnen.

## 1 Was ist ein Minimalbeispiel überhaupt?

Das Erstellen eines Minimalbeispiels ist eine Methode zur Fehlererkennung und -behebung, sowie zum Bestimmen der Ursache eines bestimmten Verhaltens. Minimalbeispiele müssen möglichst klein und lauffähig sein, d. h. sie besitzen den Grundaufbau eines `LATEX`-Dokuments und sind ohne Ergänzungen ausführbar.

Das Minimalbeispiel muss vollständig und lauffähig sein, damit

- der Fragende nicht aus Unkenntnis, Informationen, die für die Diagnose wichtig sind, weglassen kann,
- der Antwortende das Beispiel einfach per Kopieren und Einfügen übertragen und ausprobieren kann.

Das Minimalbeispiel muss klein sein,

- damit der Antwortende sich nicht durch lange, unwichtige Codezeilen quälen muss,
- weil dadurch die möglichen Fehlerquellen eingegrenzt werden,

- weil man kurze Beispiele gut per E-Mail verschicken oder in Newsgroups veröffentlichen kann,
- weil sich viele Probleme, wie fehlende Klammern, falsche Syntax, vergessene Schriftumschaltbefehle usw., während der Minimierung von selbst erledigen,
- weil man bei kleinen Dokumenten nicht so schnell den Überblick verliert.

Selbst Anfänger sollten mit dem Erstellen keine Probleme haben, da hierfür nur ein wenig Geduld und gesunder Menschenverstand nötig sind. Meistens findet man das Problem selbst und spart sich die Frage in `de.comp.text.tex` schlichtweg. Nach dieser kurzen Erläuterung kommt dann eine Auflistung der Werkzeuge, gefolgt von einigen Beispielen. Dabei werden verschiedene Problemquellen (fehlerhafte Paketoptionen, Paketkombinationen, Syntaxfehler etc.) separat behandelt. Bei komplexeren Fällen sind natürlich alle Schritte gleichzeitig auf das Dokument anzuwenden.

## 2 Die Werkzeuge

Um ein Minimalbeispiel zu erstellen, gilt vor allem ein Prinzip: Teilen und (besser be-)herrschen. Teilen kann man auf verschiedene Arten, auch abhängig von der Struktur des Dokuments. Zuerst sollte auch der Befehl `\listfiles` in das Dokument aufgenommen werden, damit alle verwendeten Dateien (Pakete, Klassen etc.) inklusive Versionsnummer in der `Log`-Datei auftauchen. Diese Informationen können für die Antwortenden wichtig sein und sollten nicht fehlen.

Für das „richtige Leben“ gilt immer: Backup anfertigen und dann an einer Kopie arbeiten, um nicht schon vorhandenes Material zu vernichten!

### 2.1 Verschieben des Endes

Da  $\text{\LaTeX}$  grundsätzlich beim ersten vorkommenden Befehl `\end{document}` aufhört, die eventuell danach folgenden Zeilen zu interpretieren, eignet sich das Vorziehen dieses Befehls sehr gut zur Fehlersuche.

#### 2.1.1 Eine Datei

Ist der gesamte Quelltext auf eine Datei konzentriert, ist folgendes sinnvoll. Durch geschicktes Verschieben des Befehls `\end{document}` in Richtung *Beginn des Textes* hilft, das Problem grob einzukreisen. Dabei bewegen wir uns

blockweise nach oben. Damit sind logische Blöcke gemeint, also Absätze, Umgebungen (gekennzeichnet durch `\begin` und `\end`) usw. Durch das Verschieben bleibt der Originaltext erhalten, wir müssen nur immer eine einzige Zeile einfügen (und jedesmal übersetzen lassen). So bekommt man einen Anhaltspunkt, in welchem Block der Fehler liegt. Nun noch alles von `\begin{document}` bis zum Beginn des gefundenen Blocks löschen; tritt der Fehler dann immer noch auf, liegt das Problem genau dort.

### 2.1.2 Zusammengesetzte Dateien

Im Grunde ist das Vorgehen analog, nur dass man in einer Quelldatei, die mit Hilfe von `\input` eingebunden wird, das Ende durch `\endinput` gekennzeichnet wird. Zuerst sollte man aber die Teile identifizieren, die verantwortlich sind, durch das schrittweise Auskommentieren der `\input`- und/oder `\include`-Befehle oder durch den Einsatz von `\includeonly`.

## 2.2 Vereinfachungen

Wenn man den zuständigen Block gefunden hat, muss man es soweit vereinfachen, dass man sein Problem auf eine Zeile festnageln kann (darum drehen sich auch die Beispiele).

### 2.2.1 Unnötige Pakete entfernen

Es gibt Pakete, die vertragen sich nicht miteinander. Jetzt werden die Pakete wieder einzeln nach und nach herausgenommen. Damit dieser Test schnell durchgeführt werden kann, empfiehlt es sich, `\usepackage` so zu verwenden:

```
1 \usepackage{%  
2   amsmath  
3   ,listings % Gute Wiedergabe von Quelltexten  
4   ,ngerman  
5 }
```

Einzelne Pakete werden durch ein einziges `%` (in der ersten Spalte) auskommentiert und deaktiviert. Sobald es keinen Fehler mehr gibt, nimmt man das zuletzt ausgeklammerte Paket wieder rein und fährt mit dem Nächsten fort.

### 2.2.2 Eigene Befehle und Umgebungen

Selbstdefinierte Befehle und Umgebungen sollten

- einfach gelöscht werden, wenn sie im Restdokument nicht mehr verwendet werden
- andernfalls „geleert“ werden, also einfach nur ein #1 (usw.) als zweites Argument haben oder nichts tun.

Auch das sollte schrittweise erfolgen, falls der Fehler durch einen dieser Befehle verursacht wird.

### 2.2.3 Schrumpfen

Jetzt wenden wir im Grunde wieder „Verschieben des Endes“ an, nur nicht blockweise, sondern zeilenweise. Der Rest muss wohl nicht näher erläutert werden. Im Idealfall bleibt nur eine Zeile des Dokuments selber übrig, die man dann genauer untersuchen kann oder, wenn alle Stricke reißen, in `de.comp.text.tex` veröffentlichen kann (oder jeder anderen Hilfequelle). Das Lesen der Einführung ist dabei sowohl für Anfänger als auch Neulinge empfehlenswert.

Es gibt noch viele weitere Möglichkeiten, Bereiche auszuklammern, sei es mit dem `comment`-Paket, bedingten Anweisungen (`if`) u.ä.

### 2.2.4 Grafiken entfernen

Eingefügte Bilder sind fast immer problematisch. Eigentlich müsste man sie weitergeben; meist sind sie aber sehr groß und enthalten womöglich vertrauliche Informationen. Ein Ablegen auf dem eigenen Webspaces ist damit auch nicht möglich. Was also tun? Ganz einfach, wir ersetzen das Bild schlichtweg durch ein Rechteck, das mit `\rule` erzeugt wird.

### 2.2.5 Benötigte zusätzliche Dateien

Manchmal ist es unumgänglich, dass zusätzliche Dateien angefügt werden müssen, z. B. eine kleine Literaturdatenbank.  $\LaTeX$  bietet eine einfache Möglichkeit, alles in eine Datei zu stecken und bei Bedarf die zusätzlichen Dateien mit einem bestimmten Inhalt zu erzeugen: Die `filecontents`-Umgebung.

```

1 \begin{filecontents}{dateiname.bib}
2   Irgendein Quatsch
3 \end{filecontents}
4 \documentclass[a4paper]{article}
5 ...

```

## 3 Beispiele

Achtung! Die Zeilennummern sind nur zur Verdeutlichung da! So sind die Beispiele nicht übersetzbar und lauffähig, deswegen bitte bei eigenen Versuchen weglassen.

### 3.1 Syntaxfehler

Folgendes Beispiel ist eine einfache ausgerichtete Formel, die man auch mit Hilfe der `eqnarray`-Umgebung erstellen kann, aber das `amsmath`-Paket ist wesentlich leistungsfähiger.

```
1 \documentclass[a4paper]{article}
2 \usepackage{amsmath}
3 \begin{document}
4 \begin{align*}
5   S_1 > S > S_2 \\
6   \Rightarrow S_1 &= \mathrm{e}^{\{x\}} \\
7   \Rightarrow S_1 &= \mathrm{e}^{\{y\}} \\
8   \Rightarrow S_1 &= \mathrm{e}^{\{x\}} \\
9   &= \boxed{\mathrm{e}^{\{z\}}}
10 \end{align*}
11 \end{document}
```

Die Fehlermeldung lautet:

```
1 Runaway argument?
2 S_1 > S > S_2 \\ \Rightarrow S_1 &= \mathrm{e}^{\{x\}} \\
3 \Rightarrow S_1 &= \mathrm{e}^{\{y\}} \\
4 \\
5 ! File ended while scanning use of \align*.
6 <inserted text>
7 \par
```

Obiges lässt vermuten, dass der Fehler in den Zeilen 6 bis 9 zu suchen ist. Also kommentiert man diese alle in der `align`-Umgebung aus.

```
1 \documentclass[a4paper]{article}
2 \usepackage{amsmath}
3 \begin{document}
4 \begin{align*}
5   S_1 > S > S_2 \\
6   % \Rightarrow S_1 &= \mathrm{e}^{\{x\}} \\
7   % \Rightarrow S_1 &= \mathrm{e}^{\{y\}} \\
8   % \Rightarrow S_1 &= \mathrm{e}^{\{x\}} \\
9   % &= \boxed{\mathrm{e}^{\{z\}}}
10 \end{align*}
```

```
11 \end{document}
```

Jetzt tritt der Fehler nicht mehr auf, man hat also zuviel entfernt, ein einzelnes Einfügen der Zeilen mit dauernden Testläufen zeigt, dass ab der Zeile 8 wieder eine fehlerrelevante Stelle dabei ist. Daraufhin nimmt man alles darüber bis `\begin{align*}` heraus (als Kommentar) und der Fehler tritt immer noch auf. Also löscht man jetzt die Zeilen und erhält ein minimales Beispiel, da das einzige Paket, das man lädt, überhaupt erst die Umgebung `align` zur Verfügung stellt.

```
1 \documentclass[a4paper]{article}
2 \usepackage{amsmath}
3 \begin{document}
4 \begin{align*}
5   \rightarrow S_1 &= \mathrm{e}^{\{x\}}
6 \end{align*}
7 \end{document}
```

Jetzt sucht man ein bisschen und sieht, dass man nach `\mathrm{e}^{\{}` die schließende geschweifte Klammer vergessen hat (gute Editoren helfen dabei hervorragend). Es muss richtig heißen: `\mathrm{e}^{\}`.

### 3.2 Unerwünschtes Verhalten

Nehmen wir mal an, der folgende Quelltext läge vor:

```
1 \documentclass[a4paper]{article}
2 \usepackage{amsmath}
3 \newcommand{\im}{\mathrm{i}}
4 \newcommand{\e}{\mathrm{e}}
5 \begin{document}
6 \begin{align}
7   f_N(t) = & \frac{A_0}{2} + \sum_{k=1}^{\infty} \left( \frac{
8     1}{2} \left( A_k - \im B_k \right) \e^{\im \alpha t} + \frac{
9     1}{2} \left( A_k + \im B_k \right) \e^{\im \alpha t} \right) \backslash
10   & \text{mit } B_0=0 \text{ und } \alpha = \omega t
\end{align}
\end{document}
```

Nur stört der subjektiv zu kleine Abstand zwischen dem Bruch und dem Gleichheitszeichen, erzeugt durch Zeile 7. Dieses Problem möchte man den Leuten in `de.comp.text.tex` schildern, ist sich aber bewusst, dass das Beispiel alles andere als übersichtlich ist. Also beschränkt man das Beispiel auf das Wesentliche, dazu gehören in diesem Fall wohl die selbstdefinierten Befehle aus den

Zeilen 3 und 4, das finden wir mit einem einfachen Testlauf heraus. Weg damit!

```
1 \documentclass[a4paper]{article}
2 \usepackage{amsmath}
3 \newcommand{\im}{}
4 \newcommand{\e}{}
5 \begin{document}
6 \begin{align}
7 f_N(t) = & \frac{A_0}{2} + \sum_{k=1}^{\infty} \left(
8 \frac{1}{2} \left( A_k - \im B_k \right) e^{\im} \{\im
9 \alpha t\} + \frac{1}{2} \left( A_k + \im B_k \right)
10 e^{\im} \{- \im \alpha t\} \right) \\\
11 & \text{\textit{mit } } B_0=0 \text{\textit{ und } } \alpha
12 & = \omega t
13 \end{align}
14 \end{document}
```

Auf die Richtigkeit der Mathematik kommt es ja nicht an. Trotz dieser Entschlackung ist das Dokument immer noch recht komplex (zumindest behaupten wir das mal, sonst hätte dieser Text keinen Sinn). Nötig ist wirklich nur Zeile 7 und von dieser auch nur der Anfang, also entfernen wir den Rest und vereinfachen die Schreibweise noch weiter.

```
1 \documentclass[a4paper]{article}
2 \usepackage{amsmath}
3 \begin{document}
4 \begin{align}
5 f = & \frac{a}{b}
6 \end{align}
7 \end{document}
```

Und wir haben ein minimales Beispiel, das genau das Problem zeigt ohne Ballast mit sich herum zu schleppen. Als Antwort könnte man dann einen Vorschlag wie den folgenden bekommen.

```
1 \documentclass[a4paper]{article}
2 \usepackage{amsmath}
3 \begin{document}
4 \begin{align}
5 f & = \frac{a}{b}
6 \end{align}
7 \end{document}
```

### 3.3 Unerklärliches Verhalten :)

Dieses Beispiel dient vor allem dazu, wie man Dinge ersetzt, die man nicht auf einfache Weise in die Newsgroup als Text einbringen kann. Was das bedeuten soll? Ganz einfach: Wenn man Bilder einfügt, sind diese meist recht groß und können schlecht als Text in eine News gesetzt werden, da einige Server Binärnachrichten in Textgruppen (wie `de.comp.text.tex` eine ist) verbieten und filtern.

Man hat sich ein schönes kleines Dokument gebastelt, das ein Bild und ein wenig Text enthält, wir wollen noch ein bisschen Mathematik schreiben und haben deswegen ein paar mehr Pakete eingebunden.

```
1 \documentclass[a4paper,12pt]{scrreprt}
2 \usepackage{graphicx}
3 \usepackage{ngerman,amsmath,exscale}
4 \usepackage[latin1]{inputenc}
5 \begin{document}
6 Dies ist ein Testtext, der nur ein bisschen helfen soll,
   Neuankömmlingen in de.comp.text.tex das Prinzip des
   Minimalbeispiels zu erklären. Und wir fügen jetzt einfach mal
   ein Bild ein.
7 \includegraphics{bild} Und wir schreiben neben dem Bild
   einfach weiter. Leider ist die Oberkante nicht bündig mit
   dem folgenden Text, sowas dummes aber auch. Da sollte man
   doch glatt ein Minimalbeispiel
8   basteln und in de.comp.text.tex nachfragen, warum das so
   ist. Aber erst nach einer gründlichen
9   Eigenrecherche.
10 \end{document}
```

Zuerst einmal testen wir, ob es was mit den anderen Paketen zu tun haben könnte, die eingebunden werden. Ein Auskommentieren von Zeile 3 ändert aber nichts am Aussehen des Textes. Da keine Standardklasse zum Einsatz kommt, sondern ein Ersatz aus dem KOMA-Script-Paket, machen wir diese sehr empfehlenswerte Klasse verantwortlich. Ein Ersetzen von `scrreprt` durch `report` bringt aber keinerlei Verbesserung, wir lassen es aber in diesem Zustand, weil wir die Klasse „report“ auf jedem System erwarten können. Nun reduzieren wir noch das restliche Drumherum und kommen zu einem deutlich minimalerem Stand.

```
1 \documentclass{report}
2 \usepackage{graphicx}
3 \begin{document}
4 \includegraphics{bild} Testtext
5 \end{document}
```

Aber im Grunde ist es nicht wirklich minimal. Flugs noch die Grafik entfernt und fertig ist unser endgültiges Minimalbeispiel.

```
1 \documentclass{report}
2 %\usepackage{graphicx}
3 \begin{document}
4 \rule{3cm}{4cm} Testtext
5 \end{document}
```

### 3.4 Stolpersteine außerhalb des Quelltexts

Der Grund für einen Fehler kann auch außerhalb des  $\text{\LaTeX}$ -Quelltextes liegen. Sehen wir uns das ein wenig näher an.

#### 3.4.1 Ärger mit Editoren

Viele Leute werden eine der gebräuchlichen IDEs oder Editoren mit  $\text{\TeX}$ -Unterstützung (Emacs, Kile,  $\text{\TeX}$ nicCenter, WinEdt, um nur einige zu nennen) einsetzen. Damit können zum einen die Dateien bearbeitet werden, zum anderen hat man einfachen Zugriff auf  $\text{\LaTeX}$  und Konsorten (Tastenkürzel, Symbolleisten etc.). Aber manchmal verfahren sie sich in ihrem Versuch, dem Benutzer zu helfen: Entweder liegt Fehlkonfiguration vor, sie finden die Programme nicht, ihnen entgehen Fehlermeldungen oder sie sind einfach zu eingeschränkt in ihren Fähigkeiten. Es kann vorkommen, dass sie glauben, das Dokument liege im DIN A5-Papierformat vor, nur weil man  $\text{\input{a5}}$  für das Einfügen des fünften Anhangs benutzt. Wenn irgendwas schief geht und der  $\text{\TeX}$ -Code ist mit Sicherheit korrekt, sollte man folgendes bedenken:

- Rotationen, Grafiken oder Farben sind häufig ein Problem für Programme, die DVI-Dateien darstellen. Oft gibt es dabei Darstellungsprobleme. Wichtig ist die korrekte Ausgabe im PostScript- oder PDF-Format.
- `latex`, `pdflatex`, `bibtex` und `makeindex` usw. sollten einmal von Hand auf der Kommandozeile aufgerufen werden. Manchmal versteckt der Editor einfach die Fehlermeldungen vor dem Benutzer. `makeindex` zeichnet sich durch das Erzeugen von schwer zu erkennenden Fehlern aus.

```
1 This is makeindex, version 2.14 [02-Oct-2002]
2 (kpathsea + Thai support).
3 Scanning style file ./svind.ist...done
4 (4 attributes redefined, 0 ignored).
5 Scanning input file Approximative.idx...done
6 (152 entries accepted, 2 rejected).
```

```

7 | Sorting entries....done (1274 comparisons).
8 | Generating output file Approximative.ind....done
9 |   (210 lines written, 0 warnings).
10 | Output written in Approximative.ind.
11 | Transcript written in Approximative.ilg.

```

Und erkannt? Zwischen zwei  $\text{\LaTeX}$ -Läufen, die jeweils 120 Zeilen Ausgabe erzeugen wäre es sicherlich untergegangen. Jeden Schritt einzeln vorzunehmen, hilft die Ausgabe zu überprüfen, ohne dass man die Log-Dateien durchsehen muss, die meist sehr unübersichtlich sind.

### 3.4.2 Probleme, die einfach nicht verschwinden

Manchmal wird  $\text{\LaTeX}$  oder ein Paket durch den Inhalt der Hilfsdateien verwirrt. Als Beispiel das folgende:

```

1 | \documentclass{article}
2 | \begin{document}
3 | \makeatletter
4 | \immediate\write\@auxout{foo}
5 | \end{document}

```

Ohne tiefer einsteigen zu müssen, reicht es zu wissen, dass das Wort „foo“; in die `aux`-Datei geschrieben wird. Der erste Lauf wird klappen, der zweit nicht, da es nicht erlaubt ist, in die `aux`-Datei zu schreiben, wenn sie gerade gelesen wird.. Das Abbrechen von `latex` (mit `x`) bewirkt kein Neuerstellen der `aux`-Datei und jeder weitere Lauf wird ebenso missglücken. Das ist komplett unabhängig wie die `tex`-Datei abgeändert wird!

Um nachzuvollziehen was passiert, kann man Klammern zählen: Wenn  $\text{\TeX}$  eine Datei öffnet, wird (`dateiname` auf dem Bildschirm ausgegeben, und `)` wenn diese Datei wieder geschlossen wird. In unserem Fall erhalten wir sowas wie

```

1 | : ~ [0 514]; latex test
2 | This is pdf $\text{\TeX}$ k, Version 3.141592-1.20a-2.2 (Web2C 7.5.3)
3 |   %&-line parsing enabled.
4 | output format initialized to DVI
5 | entering extended mode
6 | (./test.tex
7 | LaTeX2e &lt;2003/12/01&gt;
8 | Babel &lt;v3.8d&gt; and hyphenation patterns for
9 | english, dumylang, nohyphenation, german, ngerman,
10 | ukenglish, loaded.
11 | (/opt/TL2004/texmf-dist/tex/latex/base/article.cls
12 | Document Class: article 2004/02/16 v1.4f Standard
13 | LaTeX document class

```

```

14 (/opt/TL2004/texmf-dist/tex/latex/base/size10.clo)
15 (./test.aux
16
17 ! LaTeX Error: Missing \begin{document}.
18
19 See the LaTeX manual or LaTeX Companion for explanation.
20 Type H <return> for immediate help.
21 ...
22
23 1.2 f
24     oo
25 ? x
26 No pages of output.
27 Transcript written on test.log.
28 : ~ [1 515];

```

Man sieht, dass `test.aux` die letzte Datei ist, die geöffnet wird, aber nicht mehr geschlossen wird, also wird sich der Fehler dort finden (Ok, das wurde vorher schon verraten).

Natürlich kann man das auch einfacher lösen: Löschen der entsprechenden Datei. Endungen der häufigsten Hilfs-Dateien mit ihrem Zweck und Inhalt folgen (so dass man weiß, wie man sie nach dem Löschen neu erstellen kann).

- `aux`: Label, Verweise, Literaturstellen, babel-Einstellungen, diverse andere Dinge. Herkunft: `tex`- oder `ltx`-Datei
- `toc`, `lof`, `lot`: Inhalts-, Abbildungs- und Tabellenverzeichnis. Quelle: `tex`- or `ltx`-Datei
- `bb1`: Literaturverzeichnis. Quelle: `bib`- und `aux`-Dateien mittels `bibtex`
- `ind`, `idx`: Index. Quelle: `ind` aus `idx` mittels `makeindex` oder `xindy`, `idx` aus `tex`- oder `ltx`-Datei
- `gls`, `glo`: Glossar. Analog zu `ind` und `idx`.