

```
%META:TOPICINFO{author="HerbertVoss" date="1381659743" format="1.1" reprev="1.5"
version="1.5"}% %META:TOPICPARENT{name="WebHome"}%
```

\verb-Anweisung, verbatim- und Kommentar-Umgebungen

 Wieso kann ich "%%\verb%" nicht im Argument einer anderen Anweisung wie "%%\footnote%", "%%\parbox%" u.ä. verwenden? Wieso geht dies mit LaTeX2e nicht mehr, obwohl es mit LaTeX 2.09 ging? Wieso bekomme ich die Fehlermeldung "%%! LaTeX Error: \verb ended by end of line.%", wenn ich im Argument von "%%\verb%" in der Eingabe ein Zeilenende habe? Wie kann ich eine eigene Umgebung "%%verbatim%" definieren? Wie kann ich Programm-Code in mein LaTeX-Dokument einbinden? Wie kann ich den Inhalt einer Datei innerhalb einer Umgebung "%%verbatim%" darstellen? Wie erreiche ich, dass innerhalb einer Umgebung "%%verbatim%" Umlaute und andere Sonderzeichen zu sehen sind? Wie kann ich Zeilennummern zu meinem Text hinzufügen? Gibt es eine Kommentarumgebung, mit der ich größere Textteile auskommentieren kann? Wie kann ich mehrere Versionen eines Textes erstellen oder bedingte Textteile verwenden? Ich will "%%\verb%" z.B. für Dateinamen, E-Mail-Adressen und URL verwenden. Wie schaffe ich es, dass innerhalb von "%%\verb%" an geeigneten Stellen getrennt wird?

 Wieso kann ich =\verb nicht im Argument einer anderen Anweisung wie \footnote, \parbox u.ä. verwenden? Wieso geht dies mit LaTeX2e nicht mehr, obwohl es mit LaTeX 2.09 ging?

Dies ging in LaTeX 2.09 auch nur, solange innerhalb des Arguments von \verb keine Zeichen mit besonderer Bedeutung verwendet wurden. Für diese Fälle hätte ein einfaches {\tt ...} auch ausgereicht. Die Eingabe \footnote{\verb+\cite+} liefert auch mit LaTeX 2.09 Fehler mit einer sehr kryptischen Fehlermeldung, deshalb wird diese falsche Benutzung in LaTeX2e mit der aussagekräftigeren Fehlermeldung ! LaTeX Error: \verb illegal in command argument. verhindert.

Benutzt man das Paket =fancyvrb= und verwendet man in der Präambel des Dokuments die von diesem Paket definierte Anweisung =\VerbatimFootnotes, so können, anders als in Standard-LaTeX, innerhalb von Fußnoten \verb-Anweisungen verwendet werden.

fancyvrb [macros/latex/contrib/fancyvrb](https://www.ctan.org/ctan/packages/macros/latex/contrib/fancyvrb)

 Wieso bekomme ich die Fehlermeldung ! LaTeX Error: \verb ended by end of line., wenn ich im Argument von \verb in der Eingabe ein Zeilenende habe?

LaTeX 2.09 war bei \verb sehr viel toleranter und hat den Zeilenwechsel innerhalb des Arguments von \verb erlaubt, obwohl dies keinen Sinn macht, da \verb nur für kurze Textstücke gedacht ist und im formatierten Ergebnis kein Zeilenumbruch sichtbar wurde.

Aus diesen Gründen wird nun in LaTeX2e ein eventuell vorhandenes Zeilenende im Argument von `\verb` als Fehler gemeldet. Dies hat den Vorteil, dass ein versehentlich vergessenes Argumentenzeichen sehr früh erkannt und mit einer sinnvoller Fehlermeldung angezeigt werden kann. Als Seiteneffekt kann hierdurch auch die Verwendung von `\verb` im Argument einer anderen Anweisung erkannt und angezeigt werden.

Eine Abhilfe ist sehr einfach möglich, indem man in der Eingabe einen Zeilenwechsel im Argument von `\verb` vermeidet.

Wie kann ich eine eigene Umgebung =verbatim definieren?

Die intuitive Definition

```
\newenvironment{myverbatim}{%
  \begin{verbatim}%
}{%
  \end{verbatim}%
}
```

einer solchen Umgebung ist nicht möglich, da `=verbatim ` ‚festverdrahtet‘ nach der Zeichenkette `\end{verbatim}` sucht, sodass die neue Umgebung das `\end{myverbatim}` nicht als Ende erkennen kann.

Um dennoch `=verbatim=`-ähnliche Umgebungen definieren zu können, gibt es mehrere Pakete: * `alltt` belässt die drei Zeichen `\`, `{` und `}` in ihrer besonderen Bedeutung, ist daher keine vollständige `verbatim`-Umgebung. * `verbatim` erlaubt die Definition eigener `=verbatim=`-Umgebungen, wie z.B. `<verbatim>\newenvironment{myverbatim}{% \verbatim % \leftarrow nicht \begin{verbatim} ! }{% \endverbatim % \leftarrow nicht \end{verbatim} ! }</verbatim>`

- `moreverb` und `fancyvrb` enthalten weitere Umgebungen, die unterschiedliche `=verbatim=`-ähnliche Umgebungen vordefiniert enthalten oder mit denen man solche definieren kann.

<code>alltt</code>	macros/latex/base/
	macros/latex/unpacked/
	macros/latex209/contrib/misc/alltt.sty
<code>verbatim</code>	macros/latex/required/tools/
<code>moreverb</code>	macros/latex/contrib/moreverb
<code>fancyvrb</code>	macros/latex/contrib/fancyvrb

Wie kann ich Programm-Code in mein LaTeX-Dokument einbinden?

Neben der Möglichkeit, kurze Abschnitte ‚`verbatim`‘ einzubinden, gibt es verschiedene Tools, die automatisch LaTeX-Code mit den entsprechenden Syntaxhervorhebungen erzeugen: * ` =listings` Reine LaTeX-Lösung; unterstützt viele Programmiersprachen; in neueren Versionen kann `listings` zusätzlich einige der Möglichkeiten des Pakets `fancyvrb` nutzen. Möchte man allerdings Umlaute im Programm-Listing darstellen, so mussfolgendes definiert werden: `<verbatim>\lstset{% literate={ä}a1 {ö}o1 {ü}u1 {Ä}a1 {Ö}o1 {Ü}u1 {\ß}{\ss}1}</verbatim>`

- `lgrind` C-Programm und LaTeX-Style, für LaTeX2e; unterstützt sehr viele Programmiersprachen.
- `tiny_C2l` ex-Programm und LaTeX-Style; unterstützt neben C auch C++ und Java; kann überlange Zeilen umbrechen.

<code>listings</code>	macros/latex/contrib/listings/
<code>lgrind</code>	nonfree/support/lgrind/
<code>tiny_C2l</code>	support/tiny_c2l/

Wie kann ich den Inhalt einer Datei innerhalb einer =verbatim=-Umgebung darstellen?

`verbatim` [](#) stellt für diese Zwecke die Anweisung `=\verbatiminput{ _Dateiname_ }` zur Verfügung. Die Pakete `moreverb` und `fancyvrb` stellen weitere Anweisungen und Umgebungen zur Verfügung.

<code>verbatim</code>	macros/latex/required/tools/
<code>moreverb</code>	macros/latex/contrib/moreverb
<code>fancyvrb</code>	macros/latex/contrib/fancyvrb

Wie erreiche ich, dass innerhalb einer ,verbatim`-Umgebung Umlaute und andere Sonderzeichen zu sehen sind?

Zu diesem Zweck ist es nötig, die Art der Kodierung dieser Zeichen anzugeben und für das Paket `listings` `<verbatim>\lstset{% literate={ä}a1 {ö}o1 {ü}u1 {Ä}a1 {Ö}o1 {Ü}u1 {ß}{\ss}1}</verbatim>`

Wie kann ich Zeilennummern zu meinem Text hinzufügen?

Zu diesem Zweck gibt es die beiden LaTeX-Pakete `,lineno`` [](#) und `,numline``. Handelt es sich um Programmcode, der numeriert werden soll, sind jedoch die in [Frage 7.3.4](#) genannten Lösungen zu bevorzugen.

<code>lineno</code>	macros/latex/contrib/lineno
<code>numline</code>	macros/latex/contrib/numline

Gibt es eine Kommentarumgebung, mit der ich größere Textteile auskommentieren kann?

Um Teile eines Textes auszukommentieren, kann man entweder zwei sehr einfache TeX-Konstrukte verwenden, oder man bedient sich spezieller Pakete, die unterschiedlich mächtige Kommentar-Umgebungen anbieten.

Zu den einfachen Konstrukten, die TeX selbst anbietet, zählt zum einen die Einklammerung des zu ignorierenden Textes mit den TeX-Primitiven `\iffalse ... \fi`:

```
<verbatim>\iffalse
  Kommentartext, wird sehr schnell ueberlesen
\fi</verbatim>
```

Hier sollten im Kommentartext keine `\if...-`, `\else-` und `\fi`-Tokens (bzw. diese müssen korrekt geschachtelt sein), keine `,outer``-Makros oder unerlaubte Zeichen auftauchen, da diese zu einem Fehler oder vielleicht zu doch nicht auskommentierten Textteilen führen können.

Ein anderes, einfaches Verfahren ist die Verwendung eines Makros, wobei der auszukommentierende Text in dessen Argument geschrieben wird:

```
<verbatim>\newcommand{\kommentar}[1]{%
% ...
\kommentar{%
  Kommentartext, wird als Argument gelesen und ignoriert
}%</verbatim>
```

Auch hier sollten die oben genannten Tokens nicht im Kommentartext auftauchen, außerdem darf der Kommentartext nicht zu lang sein, da sonst TeXs `,main memory`` überlaufen könnte.

Spezielle Kommentariumgebungen beseitigen diese Einschränkungen - und haben meist selbst eigene Einschränkungen. `,verbatim`` [](#) aus dem `,tools``-Bündel definiert eine sehr einfache Umgebung namens `,comment``, die man jedoch nicht schachteln kann. Weiterhin gibt es noch `=xcomment`, eine speziell an die Erfordernisse des Paketes `seminar` angepasste und abgespeckte Version des `verbatim`. Mit `comment` kann man neue Kommentariumgebungen definieren, die mit `\includecomment` und `\excludecomment` Deklarationen aus- und eingeblendet werden können.

<code>verbatim</code>	macros/latex/required/tools/
<code>xcomment</code>	macros/latex/contrib/seminar/inputs/xcomment.sty
<code>comment</code>	macros/latex/contrib/comment/

[](#)Wie kann ich mehrere Versionen eines Textes erstellen oder bedingte Textteile verwenden?

Auf der einen Seite kann man die in [Frage 7.3.8](#) beschriebenen Kommentariumgebungen entsprechend verwenden. Daneben gibt es mit den Paketen `,version`` [](#), `,optional`` und dem bereits erwähnten `,comment`` Erweiterungen, mit denen man über Flags im Dokumentkopf Textteile bedingt ein- oder ausblenden kann.

<code>version</code>	macros/latex/contrib/misc/version.sty
<code>optional</code>	macros/latex/contrib/misc/optional.sty
<code>comment</code>	macros/latex/contrib/comment/

[](#)Ich will `=\verb` z.B. für Dateinamen, E-Mail-Adressen und URL verwenden. Wie schaffe ich es, dass innerhalb von `\verb` an geeigneten Stellen getrennt wird?

Für einige Anwendungen, wo man evtl. `\verb` benutzen könnte, gibt es auch spezielle Lösungen, die

für den Anwendungszweck besser geeignet sind. So gibt es beispielsweise die Pakete `url=` und `=path` für Dateinamen, E-Mail- und WWW-Adressen, bei denen nur an bestimmten Stellen umbrochen werden darf, dies aber bereits automatisch geschieht.

<code>url</code>	macros/latex/contrib/misc/url.sty
<code>path</code>	macros/latex/contrib/misc/path.sty

Verwendet man das Paket `hyperref`, so wird automatisch das Paket `url` geladen. Für den Fall., dass man zum Übersetzen des Dokuments die Sequenz `latex → dvips → ps2pdf` benutzt, ist statt `url` das Paket `breakurl` zu laden und zwar *nach* `hyperref`!

- Main.HerbertVoss - 17 Mar 2009

From:
<https://wiki.dante.de/> - **DanteWiki**

Permanent link:
<https://wiki.dante.de/doku.php?id=dantefaq:verbatimmode>

Last update: **2023/02/28 10:39**

