

# Blindtext.sty: Creating text for testing/ Texterzeugung zum testen

Knut Lickert  
<http://tex.lickert.net/packages/blindtext/>

January 2, 2007

## Abstract

This packet create text with the commands `\blindtext` and `\Blindtext`. With this created text you can get an impression of the look of new classes and packages.

Mit diesem Paket kann man lange Texte erzeugen, ohne lange Texte einzugeben. Sinn ist die Erzeugung langer Beispieltexthe, ohne den Quelltext lang zu machen.

## 1 Overview/Übersicht

### 1.1 The commands/Die Kommandos

`\blinddocument` `\blinddocument` create a document with sections, subsections... and lists (`itemize`, `enumerate` and `description`). `\Blinddocument` create a document with bigger sections, subsections... and longer lists.

The smallest document to check the look of a class would be:

```
\documentclass{<class>}
\usepackage{blindtext}
\begin{document}
  \blinddocument
\end{document}
```

`\blindtext` `\blindtext` [*x*] create a text. The optional parameter defines a repetition. Default for the repetition is one and can be modified with `\setcounter{blindtext}{1}`.

`\Blindtext` `\Blindtext` [*x*] [*y*] create *x* paragraphs with the text of `\blindtext`. The first optional parameter defines a repetition of the paragraphs. Default for the repetition is five and can be modified with `\setcounter{Blindtext}{5}`. The second optional parameter is given to `\blindtext`.

It is possible to add a start and end sequence for each paragraph.

Command/Makro	englisch	german
<code>\blinddocument</code>	creates a document	erzeugt ein Dokument
<code>\Blinddocument</code>	creates a big document	erzeugt ein großes Dokument
<code>\blindtext</code>	creates text	erzeugt Text
<code>\Blindtext</code>	creates more text	erzeugt mehr Text
<code>\blindlist{env}</code>	creates a list	erzeugt eine Liste
<code>\blindlistoptional{env}</code>	creates a list with <code>\item[]</code>	erzeugt eine Liste mit <code>\item[]</code>
<code>\blindlistlist[level]{env}</code>	creates cascaded lists	erzeugt geschachtelte Listen
<code>\Blindlist{env}</code>	creates a big list	erzeugt eine große Liste
<code>\Blindlistoptional{env}</code>	creates a big list with <code>\item[]</code>	erzeugt eine große Liste mit <code>\item[]</code>
<code>\blinditemize</code>	creates an itemize list	erzeugt eine itemize-Liste
<code>\blindenumerate</code>	creates an enumerate list	erzeugt eine enumerate-Liste
<code>\blinddescription</code>	creates a description list	erzeugt eine description-Liste
<code>~[x]</code>	repetition	Wiederholungsfaktor

Figure 1: Command Overview

<code>\parstart</code>	The two macros <code>\parstart</code> and <code>\parend</code> define a starting and ending sequence
<code>\parend</code>	for each paragraph in <code>\Blindtext</code> .
<code>\blindlist</code>	<code>\blindlist{env}[x]</code> create a list, the type is defined by the obligatory parameter. The optional parameter defines a repetition. Default for the repetition is five and can be modified with <code>\setcounter{blindlist}{x}</code> .
<code>\blindlistlist</code>	<code>\blindlistlist[level]{env}[x]</code> creates cascaded lists up to level <code>level</code> , the type is defined by the second obligatory parameter. The optional parameter defines a repetition. Default is the same like in <code>\blindlist</code> .
<code>\Blindlist</code>	<code>\Blindlist{env}[x]</code> create a list with long texts. The optional parameter defines a repetition. Default for the repetition is five and can be modified with <code>\setcounter{blindlist}{x}</code> .
<code>\blindlistoptional</code>	<code>\blindlistoptional{env}[x]</code> create a list with <code>\item[]</code> . The list is defined by the obligatory parameter. The optional parameter defines a repetition. Default for the repetition is five and can be modified with <code>\setcounter{blindlist}{x}</code> .
<code>\Blindlistoptional</code>	<code>\Blindlistoptional</code> create a list like <code>\blindlistoptional</code> with long texts.
<code>\blinditemize</code>	This commands are a abbreviation for <code>\blindlist{itemize}</code> , <code>\blindlist{enumerate}</code>
<code>\Blinditemize</code>	and <code>\blindlist{description}</code> .
<code>\blindenumerate</code>	
<code>\Blindenumerate</code>	
<code>\blinddescription</code>	
<code>\Blinddescription</code>	

## 2 Implementation/Implementierung

```

1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{blindtext}[\filedate\space\fileversion\space%
3     blindtext-Package]
4 \RequirePackage{xspace}

```

Just in case the french quote are not defined

```

5 \providecommand{\frqq}{-}
6 \providecommand{\flqq}{-}

```

`\blind@checklanguage` It is possible, that you use `blindtext.sty` with undefined languages. This macro checks, if `blindtext` support the language. If it is not available, an error is reported and English is used.

```

7 \newcommand{\blind@checklanguage}{%
8   \@ifundefined{blindtext@\languagename}{%
9     \PackageWarning{blindtext}{\languagename\space not
10      defined, using english instead.\MessageBreak
11    }%
12    \selectlanguage{english}%
13  }{}%
14 }

```

First we need some counters for the number of repetitions of the text and the paragraphs.

```

15 \newcounter{blindtext}\setcounter{blindtext}{1}
16 \newcounter{Blindtext}\setcounter{Blindtext}{5}

```

`\blindtext` `\blindtext[x]` write  $x \times$  a sentence. The default is stored in the counter `blindtext`. This value can be changed with `\setcounter{blindtext}{1}`.

```

17 \newcount\blind@countxx
18 \newcommand{\blindtext}[1][\value{blindtext}]{%
19   \blind@checklanguage
20   \blind@countxx=1 %
21   \loop
22     \blindtext@text
23   \ifnum\blind@countxx<#1\advance\blind@countxx by 1 %
24   \repeat
25 }

\Blindtext \Blindtext[x][y] execute  $x \times$  \blindtext[y]. Each \blindtext[y] built a
paragraph. The default is stored in the counter Blindtext. This value can be
changed with \setcounter{Blindtext}{5}. The counter blindtext@numBlindtext
stores the first optional parameter to be used in \blindtext@Blindtext.
26 \newcount\blindtext@numBlindtext%
27 \newcommand{\Blindtext}[1][\value{Blindtext}]{%
28   \blind@checklanguage%
29   \blindtext@numBlindtext=#1%
30   \blind@Blindtext%
31 }

\blind@Blindtext This macro continue \Blindtext with a second optional parameter.
Each paragraph starts and end with a text, which can be defined with
\starttext or \endtext.
32 \newcount\blind@countyy%
33 \newcommand{\blind@Blindtext}[1][\value{blindtext}]{%
34   \blind@countyy=1 %
35   \loop%
36   {\blindtext@parstart\blindtext[#1]\blindtext@parent\par}%
37   \ifnum\blind@countyy<\blindtext@numBlindtext\advance\blind@countyy by 1 %
38   \repeat%
39 }

\parstart
\parent 40 \newcommand{\blindtext@parstart}{ } %Text at start of paragraph
41 \newcommand{\blindtext@parent}{ } %Text at end of paragraph
42 \newcommand{\parstart}[1]{\renewcommand{\blindtext@parstart}{#1}}
43 \newcommand{\parent}[1]{\renewcommand{\blindtext@parent}{#1}}

```

### 3 Lists

Here I define some flags to decide in the lists, if we have to create big items and if we need the optional parameter at \item. This flags are reset in \blind@list.

The counter blindlist define the default number of items in a list. The counter blindlistitem is needed by \blindtext@count to write a text like 'first', 'second'...

```
44 \newif\ifblind@long\blind@longfalse
```

```

45 \newif\ifblind@optional\blind@optionalfalse
46 \newcounter{blindlist}
47 \newcounter{blindlistlevel}%Up to X level
48 \newcounter{blindlist@level}%internal counter
49 \newcount\blind@listitem

\blindlist \blindlist{env}[x] write a env-list with x items. The default is stored in the
\blind@listtype counter blindlist. This value can be changed with \setcounter{blindlist}{1}.
The type of list is stored in \blind@listtype.
50 %% ^^A-----
51 \newcommand{\blindlist}[1]{
52 \blind@checklanguage
53 \def\blind@listtype{#1}
54 \setcounter{blind@levelcount}{1}
55 \blind@list
56 }

\blind@list This macro continue the macro \blindlist and with the optional parameter.
After the printout, we reset the flags and the counter for the list text.
57 \newcommand{\blind@list}[1][5]{%
58 \setcounter{blindlist}{#1}%
59 \addtocounter{blindlist@level}{1}%depth of list
60 \blind@listitem=1 %
61 \begin{\blind@listtype}%
62 \blind@items%
63 \end{\blind@listtype}%
64 \blind@longfalse%
65 \blind@optionalfalse%
66 }

\blindlistlist \blindlistlist[num]{env}[x] writes cascaded lists up to level num.
67 \newcommand{\blindlistlist}[2][4]{%
68 \setcounter{blindlistlevel}{#1}%
69 \setcounter{blindlist@level}{0}%deepest reached level
70 \setcounter{blind@levelcount}{1}%actual level for printout
71 \def\blind@listtype{#2}%
72 \blind@list%
73 }

\blindlistlistoptional Like \blindlistlist[num]{env}[x] but for environment with optional parame-
ters.
74 \newcommand{\blindlistlistoptional}{%
75 \blind@optionaltrue%
76 \blindlistlist%
77 }

\blinditemize Some default macros for the standard environments.
\blindenumerate 78 \newcommand{\blinditemize}{\blindlist{itemize}}
79 \newcommand{\blindenumerate}{\blindlist{enumerate}}

```

`\Blindlist[x]` write a list with  $x$  items. The default is stored in the counter `blindlist`. This value can be changed with `\setcounter{blindlist}{1}`.

`\Blindlist`

```
80 \newcommand{\Blindlist}[1]{%
81 \blind@longtrue%
82 \blindlist{#1}%
83 }
```

`\Blinditemize` Some default macros for the standard environments.  
`\Blindenumerate` 84 \newcommand{\Blinditemize}{\Blindlist{itemize}}  
85 \newcommand{\Blindenumerate}{\Blindlist{enumerate}}

`\blindlistoptional` Here we start a list with `\item[]`. So we set the flag for optional parameter and start the normal list.  
86 \newcommand{\blindlistoptional}[1]{%
87 \blind@optionaltrue%
88 \blindlist{#1}%
89 }

`\Blindlistoptional` Now a big list with optional parameter at `\item`.  
90 \newcommand{\Blindlistoptional}[1]{%
91 \blind@optionaltrue%
92 \blind@longtrue%
93 \blindlist{#1}%
94 }

`\blinddescription` Some default macros for the standard environment description.  
`\Blinddescription` 95 \newcommand{\blinddescription}{\blindlistoptional{description}}  
96 \newcommand{\Blinddescription}{\Blindlistoptional{description}}  
97 \newcounter{blind@listcount}  
98 \newcounter{blind@levelcount}

`\blind@items` Here the `\item` are written. A local counter is defined and the `\item` is written. The type of `\item` is influenced by the flags.  
99 \newcommand{\blind@items}{  
100 \setcounter{blind@listcount}{1}  
101 \loop  
102 \ifblind@optional  
103 \ifblind@long  
104 \item[\blindtext@countitem] \blindtext@text  
105 \else  
106 \item[\blindtext@count] \blindtext@item  
107 \fi  
108 \else % \blind@optionalfalse  
109 \ifblind@long  
110 \item \blindtext@text  
111 \else  
112 \item \blindtext@countitem

```

113 \fi
114 \fi % \ifblind@optional
    Loop for cascaded lists.
115 {%
116 \loop
117 \ifnum\value{blindlistlevel}>\value{blindlist@level}
118 \addtocounter{blind@levelcount}{1}
119 \blind@list[\value{blindlist}]\relax
120 \addtocounter{blind@levelcount}{-1}
121 \setcounter{blind@listcount}{1}
122 \repeat
123 }%
124 \ifnum\value{blind@listcount}<\value{blindlist}
125 \addtocounter{blind@listcount}{1}
    Correction for nested values.

    Close the loop
126 \repeat
127 }

```

## 4 A sample document

`\blinddocument` `\blinddocument` adds a document with `\chapter`, `\section...`, texts and lists (itemize, enumerate, description).

```

128 \newcommand{\blinddocument}{
129 \ifundefined{chapter}{}%
130 \chapter{\blindtext@heading 0 (chapter)}
131 \blindtext%
132 }
133 \section{\blindtext@heading 1 (section)}
134 \blindtext
135 \subsection{\blindtext@heading 2 (subsection)}
136 \blindtext
137 \subsubsection{\blindtext@heading 3 (subsubsection)}%
138 \blindtext
139 \paragraph{\blindtext@heading 4 (paragraph)}
140 \blindtext
141 \section{\blindtext@list}
142 \subsection{\blindtext@listEx (itemize)}
143 \blinditemize
144 \subsubsection{\blindtext@listEx (4*itemize)}
145 \blindlistlist{itemize}[2]
146 \subsection{\blindtext@listEx (enumerate)}
147 \blindenumerate
148 \subsubsection{\blindtext@listEx (4*enumerate)}
149 \blindlistlist{enumerate}[2]
150 \subsection{\blindtext@listEx (description)}
151 \blinddescription

```

```

152     \subsubsection{\blindtext@listEx (4*description)}
153     \blindlistlistoptional{description}[2]
154 }

\Blinddocument \blinddocument adds a document with \chapter, \section..., texts and lists
(itemize, enumerate, description).
155 \newcommand{\Blinddocument}{
156 \@ifundefined{chapter}{}{%
157     \chapter{\blindtext@heading 0 (chapter)}
158     \blindtext%
159 }%
160 \section{\blindtext@heading 1 (section)}
161 \Blindtext
162 \subsection{\blindtext@heading 2 (subsection)}
163 \Blindtext
164 \subsubsection{\blindtext@heading 3 (subsection)}
165 \Blindtext
166 \paragraph{\blindtext@heading 4 (paragraph)}
167 \Blindtext
168 \section{\blindtext@list}
169 \subsection{\blindtext@listEx (itemize)}
170     \Blinditemize
171     \subsubsection{\blindtext@listEx (4*itemize)}
172     \blind@longtrue
173     \blindlistlist{itemize}[2]
174 \subsection{\blindtext@listEx (enumerate)}
175     \Blindenumerate
176     \subsubsection{\blindtext@listEx (4*enumerate)}
177     \blind@longtrue
178     \blindlistlist{enumerate}[2]
179 \subsection{\blindtext@listEx (description)}
180     \Blinddescription
181     \subsubsection{\blindtext@listEx (4*description)}
182     \blind@longtrue
183     \blindlistlistoptional{description}[2]
184 }

```

## 5 The texts

```

\addto If babel is not loaded, we need the command \addto.
185 %\providecommand{\addto}[2]{#2}
186 \@ifundefined{addto}{
187 \@ifpackageloaded{babel}{}{
188 \PackageWarning{blindtext}{%
189 You are not using babel.sty -> Language by random\MessageBreak
190 Kein babel.sty verwendet -> Zufallssprache}}
191 \newcommand{\addto}[2]{#2}
192 }{}

```

## 5.1 Default without language

First all texts without any language. If you make a new translation, please define all this macros.

```
193 \def\blindtext@text{Text}
194 \def\blindtext@count{%
195   \ifcase\blind@listitem\or
196     First\or Second%...
197   \else
198     Last%
199   \blind@listitem=0 %
200   \fi
201   \global\advance\blind@listitem by 1 %
202 }%\blindtext@count}
203 \def\blindtext@item{itemtext}
204 \def\blindtext@countitem{\blindtext@count\ \blindtext@item}
205 \def\blindtext@heading{Heading on level\xspace}
206 \def\blindtext@list{Lists}
207 \def\blindtext@listEx{Example for list\xspace}
```

## 5.2 English Texts

`\blindtext@english` Define flag, so we can check, if language is defined.

```
208 \def\blindtext@english{}

209 \addto\extrasenglish{%
210 \def\blindtext@text{
211 Hello, here is some text without a meaning. This text should show,
212 how a printed text will look like at this place. If you read this
213 text, you will get no information. Really? Is there no
214 information? Is there a difference between this text and some
215 nonsense like \frqq Huardest gefburn\flqq? Kjift -- Never mind! A
216 blind text like this gives you information about the selected
217 font, how the letters are written and the impression of the look.
218 This text should contain all letters of the alphabet and it should
219 be written in of the original language. There is no need for a
220 special contents, but the length of words should match to the
221 language.}%\blindtext@text
222 }

223 \addto\extrasenglish{%
224   \def\blindtext@count{%
225     \ifcase\value{blind@listcount}\or
226       First\or Second \or Third \or Fourth\or Fifth\or
227       Sixth\or Seventh\or Eighth \or Ninth\or Tenth\or
228       Eleventh\or Twelfth%
229     \else Another%
230     \fi
231   }%\blindtext@count
232   \def\blindtext@item{item in a list}%, level~\arabic{blindlist@level}}
```

```

233 }% \addto\extrasenglish
234 \addto\extrasenglish{%
235     \def\blindtext@heading{Heading on level\xspace}%
236     \def\blindtext@list{Lists}%
237     \def\blindtext@listEx{Example for list\xspace}%
238 }%\addto\extrasenglish
239 %

```

### 5.3 German Texts

\blindtext@german Define flag, so we can check, if language is defined.

```

240 \def\blindtext@german{}

241 \addto\extrasgerman{%
242 \def\blindtext@text{
243 Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer
244 diesen Text liest, ist selbst schuld. Der Text gibt lediglich den
245 Grauwert der Schrift an. Ist das wirklich so? Ist es
246 gleich-g"ul-tig ob ich schreibe: \frqq Dies ist ein
247 Blindtext\flqq\ oder \frqq Huardest gefburn\flqq? Kjift --
248 mitnichten! Ein Blindtext bietet mir wichtige Informationen. An
249 ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie
250 harmonisch die Figuren zueinander stehen und pr"u-fe, wie breit
251 oder schmal sie l"auft. Ein Blindtext sollte m"og\lichst viele
252 verschiedene Buchstaben enthalten und in der Originalsprache
253 gesetzt sein. Er mu\ss\ keinen Sinn ergeben, sollte aber lesbar
254 sein. Fremdsprachige Texte wie \frqq Lorem ipsum\flqq\ dienen
255 nicht dem eigentlichen Zweck, da sie eine
256 falsche Anmutung vermitteln.}%\blindtext@text
257 }

258 \addto\extrasgerman{%
259 \def\blindtext@count{
260     \ifcase\value{blind@listcount}\or
261     Erster\or Zweiter \or Dritter \or Vierter \or F{"u}nfter \or
262     Sechster \or Siebter\or Achter \or Neunter \or Zehnter\or
263     Elfter \or Zw{"o}lfter \or Dreizehnter \or Vierzehnter
264     \else
265     Noch ein
266     \fi
267 }%\blindtext@count
268 \def\blindtext@item{Listenpunkt, Stufe~\arabic{blind@levelcount}}%
269 }%\addto\extrasgerman

270 \addto\extrasgerman{%
271     \def\blindtext@heading{"U}berschrift auf Ebene\xspace}%
272     \def\blindtext@list{Listen}%
273     \def\blindtext@listEx{Beispiel einer Liste\xspace}%
274 }%\addto\extrasgerman
275 %

```

```

276 % If the packet \Lpack{german} is loaded, select the language.
277 %   \begin{macrocode}
278 \@ifpackageloaded{german}{\selectlanguage{german}}{}

```

## 5.4 German - New typesetting

```

\blindtext@ngerman Define flag, so we can check, if language is defined.
279 \def\blindtext@ngerman{}

280 \addto\extrasngerman{%
281 \def\blindtext@text{
282 Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer
283 diesen Text liest, ist selbst schuld. Der Text gibt lediglich den
284 Grauwert der Schrift an. Ist das wirklich so? Ist es
285 gleich-g"ul"-tig ob ich schreibe: \frqq Dies ist ein
286 Blindtext\flqq\ oder \frqq Huardest gefburn\flqq? Kjift --
287 mitnichten! Ein Blindtext bietet mir wichtige Informationen. An
288 ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie
289 harmonisch die Figuren zueinander stehen und pr"u"-fe, wie breit
290 oder schmal sie l"auft. Ein Blindtext sollte m"og"-lichst viele
291 verschiedene Buchstaben enthalten und in der Originalsprache
292 gesetzt sein. Er mu"ss\ keinen Sinn ergeben, sollte aber lesbar
293 sein. Fremdsprachige Texte wie \frqq Lorem ipsum\flqq\ dienen
294 nicht dem eigentlichen Zweck, da sie eine
295 falsche Anmutung vermitteln.}%\blindtext@text
296 }

297 \addto\extrasngerman{%
298 \def\blindtext@count{
299   \ifcase\value{blind@listcount}\or
300   Erster\or Zweiter \or Dritter \or Vierter \or F{"u}nfter \or
301   Sechster \or Siebter\or Achter \or Neunter \or Zehnter\or
302   Elfter \or Zw{"o}lfter \or Dreizehnter \or Vierzehnter\else
303   Noch ein\fi
304 }%\blindtext@count
305 \def\blindtext@item{Listenpunkt, Stufe~\arabic{blind@levelcount}}%
306 }%\addto\extrasngerman

307 \addto\extrasngerman{%
308   \def\blindtext@heading{{\U}berschrift auf Ebene\xspace}%
309   \def\blindtext@list{Listen}%
310   \def\blindtext@listEx{Beispiel einer Liste\xspace}%
311 }%\addto\extrasngerman
312 %
313 % If the packet \Lpack{ngerman} is loaded, select the language.
314 %   \begin{macrocode}
315 \@ifpackageloaded{ngerman}{\selectlanguage{ngerman}}{}

```

## 5.5 Latin Texts

```

\blindtext@latin Define flag, so we can check, if language is defined.

```

```
316 \def\blindtext@latin{}
```

I don't speak latin, but I think the classic "Lorem ipsum" should be available.

```
317 \addto\extraslatin{%
318 \def\blindtext@text{
319 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam
320 lobortis facilisis sem. Nullam nec mi et neque pharetra
321 sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper,
322 felis non sodales commodo, lectus velit ultrices augue, a
323 dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie
324 ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in
325 sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
326 Duis fringilla tristique neque. Sed interdum libero ut metus.
327 Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit
328 amet ante lobortis sollicitudin. Praesent blandit blandit mauris.
329 Praesent lectus tellus, aliquet aliquam, luctus a, egestas a,
330 turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum
331 turpis accumsan semper.}%\blindtext@text
332 }
```

The following "Latin" texts are not really correct latin. (I never learned Latin)  
If you want correct texts, please provide them to me.

```
333 \addto\extraslatin{%
334 \def\blindtext@count{
335   \ifcase\value{blind@listcount}\or
336     Primus\or Duo\or Tres\or Quattuor\or Quinque\or
337     Sex\or Septem\or Octo\or Novem\or Decem\or
338     Undecim\or Duodecim
339   \else
340     Nova
341   \fi
342 }%\blindtext@count
343 \def\blindtext@item{, altum~\arabic{blind@levelcount}}}%
344 }%\addto\extrasgerman

345 \addto\extraslatin{%
346   \def\blindtext@heading{Sectio\xspace}%
347   \def\blindtext@list{Caudex}%
348   \def\blindtext@listEx{Exemplum caudex\xspace}%
349 }%\addto\extraslatin
```

## 6 Some other macros

\alotoftext

```
350 \let\alotoftext=\blindtext
```

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

<b>A</b>	<code>\Blinddocument</code> . . . . .	<u>155</u>	<code>\Blindlistoptional</code> .	<u>90</u>
<code>\addto</code> . . . . .	<code>\blinddocument</code> . . . . .	<u>128</u>	<code>\blindlistoptional</code> .	<u>86</u>
<code>\alotof</code> . . . . .	<code>\Blindenumerate</code> . . . . .	<u>84</u>	<code>\Blindtext</code> . . . . .	<u>26</u>
	<code>\blindenumerate</code> . . . . .	<u>78</u>	<code>\blindtext</code> . . . . .	<u>17</u>
<b>B</b>	<code>\Blinditemize</code> . . . . .	<u>84</u>	<code>\blindtext@english</code> .	<u>208</u>
<code>\blind@Blindtext</code> . . . . .	<code>\blinditemize</code> . . . . .	<u>78</u>	<code>\blindtext@german</code> .	<u>240</u>
<code>\blind@checklanguage</code> . . . . .	<code>\Blindlist</code> . . . . .	<u>80</u>	<code>\blindtext@latin</code> . .	<u>316</u>
<code>\blind@items</code> . . . . .	<code>\blindlist</code> . . . . .	<u>50</u>	<code>\blindtext@ngerman</code> .	<u>279</u>
<code>\blind@list</code> . . . . .	<code>\blindlistlist</code> . . . . .	<u>67</u>	<b>P</b>	
<code>\blind@listtype</code> . . . . .	<code>\blindlistlistoptional</code>		<code>\parend</code> . . . . .	<u>40</u>
<code>\Blinddescription</code> . . . . .	. . . . .	<u>74</u>	<code>\parstart</code> . . . . .	<u>40</u>
<code>\blinddescription</code> . . . . .				