

# The mleft mright package

Heiko Oberdiek  
<heiko.oberdiek at gmail.com>

2010/09/25 v1.0

## Abstract

TeX sets subformulas by `\left` and `\right` as inner formulas with additional surrounding spaces in some situations. This package provides `\mleft` and `\mright` that call `\left` and `\right`, but the delimiters will act as normal `\mathopen` and `\mathclose` delimiters without the additional space of an inner formula.

## Contents

<b>1 Documentation</b>	<b>1</b>
1.1 Use . . . . .	2
<b>2 Implementation</b>	<b>2</b>
<b>3 Test</b>	<b>7</b>
3.1 Catcode checks for loading . . . . .	7
<b>4 Installation</b>	<b>8</b>
4.1 Download . . . . .	8
4.2 Bundle installation . . . . .	9
4.3 Package installation . . . . .	9
4.4 Refresh file name databases . . . . .	9
4.5 Some details for the interested . . . . .	9
<b>5 Acknowledgement</b>	<b>10</b>
<b>6 References</b>	<b>10</b>
<b>7 History</b>	<b>10</b>
[2010/09/25 v1.0] . . . . .	10
<b>8 Index</b>	<b>10</b>

## 1 Documentation

The package is a result of a thread in the newsgroup `comp.text.tex` with the subject *spacing after \right) and before \left)* [1]. The problem: `\left` and `\right` adjust the size of the delimiters automatically. However, TeX treats the whole expression as inner formula. In some circumstances TeX adds extra space before or after an inner formula. Example:

```
$\sin(x^2), x$            $\Rightarrow \sin(x^2), x$ 
$\sin\left(x^2\right), x$  $\Rightarrow \sin\left(x^2\right), x$ 
$\sin\mleft(x^2\mright), x$  $\Rightarrow \sin(x^2), x$ 
(\mleft and \mright are provided by this package.)
```

In the newsgroup Donald Arseneau answered with clever macros [2]:

```
\newcommand\lft{\mathopen{}\left}
\newcommand\rgt{\aftergroup\mathclose\aftergroup{\aftergroup}\right}
```

However one problem remains, a following subscript or superscript is not applied to the right delimiter but the empty `\mathclose`. Thus Philipp Stephani provided an improvement [3]:

```
\mathopen{} \mathclose{\left\| A^2 \right\|}_2
```

Heiko Oberdiek converted this into macro form [4]:

```
\newcommand\lft{\mathopen{}\mathclose\bgroup\left}
\newcommand\rgt{\aftergroup\egroup\right}
```

The package uses longer macro names `\mleft` and `\mright` to avoid name clashes. Also it adds some checks for error conditions.

## 1.1 Use

`\mleft<delimL> ... \mright<delimR>`

Macros `\mleft` and `\mright` are used in the same way as `\left` and `\right`. Also `\middle` can be used inbetween if  $\varepsilon$ -TeX is present.

`\mleftright`

Macro `\mleftright` redefines `\left` as `\mleft` and `\right` as `\mright`. The redefinition is local to the group.

`\mleftrighrestore`

Macro `\mleftright` restores `\left` and `\right` with the original meaning if they were previously redefined by `\mleftright` (also locally).

## 2 Implementation

```
1 (*package)
```

Reload check, especially if the package is not used with L<sup>A</sup>T<sub>E</sub>X.

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
3 \catcode13=5 % ^^M
4 \endlinechar=13 %
5 \catcode35=6 % #
6 \catcode39=12 % '
7 \catcode44=12 % ,
8 \catcode45=12 % -
9 \catcode46=12 % .
10 \catcode58=12 % :
11 \catcode64=11 % @
12 \catcode123=1 % {
13 \catcode125=2 % }
14 \expandafter\let\expandafter\x\csname ver@mleftright.sty\endcsname
15 \ifx\x\relax % plain-TeX, first loading
16 \else
17 \def\empty{}%
18 \ifx\x\empty % LaTeX, first loading,
19 % variable is initialized, but \ProvidesPackage not yet seen
20 \else
```

```

21 \expandafter\ifx\csname PackageInfo\endcsname\relax
22 \def\x#1#2{%
23 \immediate\write-1{Package #1 Info: #2.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{mletright}{The package is already loaded}%
29 \aftergroup\endinput
30 \fi
31 \fi
32 \endgroup%

```

Package identification:

```

33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34 \catcode13=5 % ^~M
35 \endlinechar=13 %
36 \catcode35=6 % #
37 \catcode39=12 % '
38 \catcode40=12 % (
39 \catcode41=12 % )
40 \catcode44=12 % ,
41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51 \def\x#1#2#3[#4]{\endgroup
52 \immediate\write-1{Package: #3 #4}%
53 \xdef#1{#4}%
54 }%
55 \else
56 \def\x#1#2[#3]{\endgroup
57 #2[#3]}%
58 \ifx#1\@undefined
59 \xdef#1{#3}%
60 \fi
61 \ifx#1\relax
62 \xdef#1{#3}%
63 \fi
64 }%
65 \fi
66 \expandafter\x\csname ver@mletright.sty\endcsname
67 \ProvidesPackage{mletright}%
68 [2010/09/25 v1.0 Usual delimiter spacing with left and right (HO)]%
69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^~M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76 \expandafter\edef\csname mletright@AtEnd\endcsname{%
77 \endlinechar=\the\endlinechar\relax
78 \catcode13=\the\catcode13\relax
79 \catcode32=\the\catcode32\relax
80 \catcode35=\the\catcode35\relax
81 \catcode61=\the\catcode61\relax

```

```

82      \catcode64=\the\catcode64\relax
83      \catcode123=\the\catcode123\relax
84      \catcode125=\the\catcode125\relax
85      }%
86      }%
87      \x\catcode61\catcode48\catcode32=10\relax%
88      \catcode13=5 % ^~M
89      \endlinechar=13 %
90      \catcode35=6 % #
91      \catcode64=11 % @
92      \catcode123=1 % {
93      \catcode125=2 % }
94      \def\TMP@EnsureCode#1#2{%
95        \edef\mleftright@AtEnd{%
96          \mleftright@AtEnd
97          \catcode#1=\the\catcode#1\relax
98        }%
99        \catcode#1=#2\relax
100     }
101     \TMP@EnsureCode{38}{4}% &
102     \TMP@EnsureCode{39}{12}% '
103     \TMP@EnsureCode{40}{12}% (
104     \TMP@EnsureCode{41}{12}% )
105     \TMP@EnsureCode{42}{12}% *
106     \TMP@EnsureCode{43}{12}% +
107     \TMP@EnsureCode{44}{12}% ,
108     \TMP@EnsureCode{45}{12}% -
109     \TMP@EnsureCode{46}{12}% .
110     \TMP@EnsureCode{47}{12}% /
111     \TMP@EnsureCode{60}{12}% <
112     \TMP@EnsureCode{91}{12}% [
113     \TMP@EnsureCode{93}{12}% ]
114     \edef\mleftright@AtEnd{%
115       \mleftright@AtEnd
116       \escapechar\the\escapechar\relax
117       \noexpand\endinput
118     }
119     \escapechar=92 %
120     \begingroup\expandafter\expandafter\expandafter\endgroup
121     \expandafter\ifx\csname RequirePackage\endcsname\relax
122       \input infwarerr.sty\relax
123       \input ltxcmds.sty\relax
124     \else
125       \RequirePackage{infwarerr}[2010/04/08]%
126       \RequirePackage{ltxcmds}[2010/04/26]%
127     \fi

```

The original commands `\left` and `\right` are saved and later used in `\mleft` and `\mright` in order to deal with:

```

\let\left\mleft
\let\right\mright

```

`\mleftright@OrgLeft`

```

128 \let\mleftright@OrgLeft\left

```

`\mleftright@OrgRight`

```

129 \let\mleftright@OrgRight\right

```

`\mleftright@Def` Macro `\mleftright@Def` defines a macro as robust macro if  $\varepsilon$ -TeX or L<sup>A</sup>T<sub>E</sub>X is available.

```

130 \ltx@ifundefined{protected}{%
131   \ltx@ifundefined{DeclareRobustCommand}{%

```

```

132 \def\mletright@Def{\def}%
133 }{%
134 \def\mletright@Def{\DeclareRobustCommand*}%
135 }%
136 }{%
137 \def\mletright@Def{\protected\def}%
138 }
139 \edef\mletright@Def#1{%
140 \noexpand\ltx@ifundefined{%
141 \noexpand\expandafter\noexpand\ltx@gobble\noexpand\string#1%
142 }{%
143 \expandafter\noexpand\mletright@Def#1%
144 }{%
145 \noexpand\@PackageError{mletright}{%
146 Command \noexpand\string#1 already defined%
147 }\noexpand\@ehd
148 \noexpand\ltx@gobble
149 }%
150 }

```

In case of  $\varepsilon$ -TEX the group status after the left symbol is saved and later checked at the beginning of `\mright`.

```

151 \ltx@ifundefined{currentgrouplevel}{%
152 \catcode38=14 % & = comment
153 }{%
154 \catcode38=9 % & = ignore_
155 }

```

`\mletright@GroupLevel`

```

156 & \def\mletright@GroupLevel{-1}%

```

`\mletright@WrongGroup`

```

157 & \def\mletright@WrongGroup#1(#2){%
158 & \ifnum\mletright@GroupLevel<\ltx@zero
159 & \@PackageError{mletright}{%
160 & Missing previous \string\mleft
161 & }\@ehc
162 & \else
163 & \@PackageError{mletright}{%
164 & Unexpected group status for \string\mright%
165 & \ifnum\mletright@GroupLevel=#1 %
166 & \else
167 & .\MessageBreak
168 & Group level is #1, %
169 & expected is \mletright@GroupLevel
170 & \fi
171 & \ifnum16=#2 %
172 & \else
173 & .\MessageBreak
174 & Group type is #2 (%)
175 & \ifcase#2 %
176 & bottom level%
177 & \expandafter\expandafter\expandafter\ltx@gobblefour
178 & \expandafter\ltx@gobbletwo
179 & \or simple%
180 & \or hbox%
181 & \or adjusted hbox%
182 & \or vbox%
183 & \or vtop%
184 & \or align%
185 & \or no align%
186 & \or output%

```

```

187 & \or math%
188 & \or disc%
189 & \or insert%
190 & \or vcenter%
191 & \or math choice%
192 & \or semi simple%
193 & \or math shift%
194 & \or math left%
195 & \else
196 & unknown%
197 & \fi
198 & \space group),\MessageBreak
199 & expected is 16 (math left group)%
200 & \fi
201 & }\@ehd
202 & \fi
203 & }%

```

\mleft

```

204 \mleftright@Def\mleft{%
205 \mathopen{ }\mathclose\bgroup
206 & \edef\mleftright@GroupLevel{\the\numexpr\the\currentgrouplevel+1}%
207 \mleftright@OrgLeft
208 }

```

\mright

```

209 \mleftright@Def\mright{%
210 & \ifnum\mleftright@GroupLevel=\currentgrouplevel
211 & \ifnum16=\currentgrouptype
212 & \aftergroup\egroup
213 & \else
214 & \expandafter\mleftright@WrongGroup
215 & \the\expandafter\currentgrouplevel
216 & \expandafter(\the\currentgrouptype)%
217 & \fi
218 & \else
219 & \expandafter\mleftright@WrongGroup
220 & \the\expandafter\currentgrouplevel
221 & \expandafter(\the\currentgrouptype)%
222 & \fi
223 \mleftright@OrgRight
224 }

```

\mleftright

```

225 \mleftright@Def\mleftright{%
226 \let\left\mleft
227 \let\right\mright
228 }

```

\mleftrightrestore

```

229 \mleftright@Def\mleftrightrestore{%
230 \ifx\left\mleft
231 \let\left\mleftright@OrgLeft
232 \fi
233 \ifx\right\mright
234 \let\right\mleftright@OrgRight
235 \fi
236 }

237 \mleftright@AtEnd%
238 \end{package}

```

## 3 Test

### 3.1 Catcode checks for loading

```
239 <*test1>
240 \catcode'\{=1 %
241 \catcode'\}=2 %
242 \catcode'\#=6 %
243 \catcode'\@=11 %
244 \expandafter\ifx\csname count@\endcsname\relax
245   \countdef\count@=255 %
246 \fi
247 \expandafter\ifx\csname @gobble\endcsname\relax
248   \long\def\@gobble#1{%
249 \fi
250 \expandafter\ifx\csname @firstofone\endcsname\relax
251   \long\def\@firstofone#1{#1}%
252 \fi
253 \expandafter\ifx\csname loop\endcsname\relax
254   \expandafter\@firstofone
255 \else
256   \expandafter\@gobble
257 \fi
258 {%
259   \def\loop#1\repeat{%
260     \def\body{#1}%
261     \iterate
262   }%
263   \def\iterate{%
264     \body
265     \let\next\iterate
266   \else
267     \let\next\relax
268   \fi
269   \next
270 }%
271 \let\repeat=\fi
272 }%
273 \def\RestoreCatcodes{}
274 \count@=0 %
275 \loop
276   \edef\RestoreCatcodes{%
277     \RestoreCatcodes
278     \catcode\the\count@=\the\catcode\count@\relax
279   }%
280 \ifnum\count@<255 %
281   \advance\count@ 1 %
282 \repeat
283
284 \def\RangeCatcodeInvalid#1#2{%
285   \count@=#1\relax
286   \loop
287     \catcode\count@=15 %
288     \ifnum\count@<#2\relax
289       \advance\count@ 1 %
290     \repeat
291 }
292 \def\RangeCatcodeCheck#1#2#3{%
293   \count@=#1\relax
294   \loop
295     \ifnum#3=\catcode\count@
296     \else
```

```

297     \errmessage{%
298         Character \the\count@\space
299         with wrong catcode \the\catcode\count@\space
300         instead of \number#3%
301     }%
302     \fi
303     \ifnum\count@<#2\relax
304         \advance\count@ 1 %
305     \repeat
306 }
307 \def\space{ }
308 \expandafter\ifx\csname LoadCommand\endcsname\relax
309     \def\LoadCommand{\input mleftright.sty\relax}%
310 \fi
311 \def\Test{%
312     \RangeCatcodeInvalid{0}{47}%
313     \RangeCatcodeInvalid{58}{64}%
314     \RangeCatcodeInvalid{91}{96}%
315     \RangeCatcodeInvalid{123}{255}%
316     \catcode'\@=12 %
317     \catcode'\=0 %
318     \catcode'\%=14 %
319     \LoadCommand
320     \RangeCatcodeCheck{0}{36}{15}%
321     \RangeCatcodeCheck{37}{37}{14}%
322     \RangeCatcodeCheck{38}{47}{15}%
323     \RangeCatcodeCheck{48}{57}{12}%
324     \RangeCatcodeCheck{58}{63}{15}%
325     \RangeCatcodeCheck{64}{64}{12}%
326     \RangeCatcodeCheck{65}{90}{11}%
327     \RangeCatcodeCheck{91}{91}{15}%
328     \RangeCatcodeCheck{92}{92}{0}%
329     \RangeCatcodeCheck{93}{96}{15}%
330     \RangeCatcodeCheck{97}{122}{11}%
331     \RangeCatcodeCheck{123}{255}{15}%
332     \RestoreCatcodes
333 }
334 \Test
335 \csname @@end\endcsname
336 \end
337 </test1>

```

## 4 Installation

### 4.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[CTAN:macros/latex/contrib/oberdiek/mleftright.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/mleftright.pdf](#) Documentation.

**Bundle.** All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#)

TDS refers to the standard “A Directory Structure for T<sub>E</sub>X Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

---

<sup>1</sup>[ftp://ftp.ctan.org/tex-archive/](http://ftp.ctan.org/tex-archive/)



## 4.2 Bundle installation

**Unpacking.** Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

**Script installation.** Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

## 4.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain  $\mathrm{\TeX}$ :

```
tex mleftright.dtx
```

**TDS.** Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

<code>mleftright.sty</code>	$\rightarrow$ <code>tex/generic/oberdiek/mleftright.sty</code>
<code>mleftright.pdf</code>	$\rightarrow$ <code>doc/latex/oberdiek/mleftright.pdf</code>
<code>test/mleftright-test1.tex</code>	$\rightarrow$ <code>doc/latex/oberdiek/test/mleftright-test1.tex</code>
<code>mleftright.dtx</code>	$\rightarrow$ <code>source/latex/oberdiek/mleftright.dtx</code>

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

## 4.4 Refresh file name databases

If your  $\mathrm{\TeX}$  distribution (te $\mathrm{\TeX}$ , mi $\mathrm{\TeX}$ , ...) relies on file name databases, you must refresh these. For example, te $\mathrm{\TeX}$  users run `texhash` or `mktextlsr`.

## 4.5 Some details for the interested

**Attached source.** The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk mleftright.pdf unpack_files output .
```

**Unpacking with  $\mathrm{\LaTeX}$ .** The `.dtx` chooses its action depending on the format:

**plain  $\mathrm{\TeX}$ :** Run `docstrip` and extract the files.

**$\mathrm{\LaTeX}$ :** Generate the documentation.

If you insist on using  $\mathrm{\LaTeX}$  for `docstrip` (really, `docstrip` does not need  $\mathrm{\LaTeX}$ ), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{mleftright.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

**Generating the documentation.** You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfL<sup>A</sup>T<sub>E</sub>X:

```
pdflatex mlefttright.dtx
makeindex -s gind.ist mlefttright.idx
pdflatex mlefttright.dtx
makeindex -s gind.ist mlefttright.idx
pdflatex mlefttright.dtx
```

## 5 Acknowledgement

**Donald Arsenau:** He provided the main trick and the first macros.

**Philipp Stephani:** He solved the subscript problem.

## 6 References

- [1] Dave94705, *spacing after \right) and before \left)*, newsgroup comp.text.tex, Message-ID: 5d264909-7c3d-4c9d-9b22-434178b2bf90@g21g2000prn.googlegroups.com, 2010-08-12.  
<http://groups.google.com/group/comp.text.tex/msg/e5b6833da7dc29bf>
- [2] Donald Arseneau, *Re: spacing after \right) and before \left)*, newsgroup comp.text.tex, Message-ID: yfivd6svl8y.fsf@mutant.triumf.ca, 2010-08-30.  
<http://groups.google.com/group/comp.text.tex/msg/e0b2e4386e5d04e4>
- [3] Philipp Stephani, *Re: spacing after \right) and before \left)*, newsgroup comp.text.tex, Message-ID: 4c8c8c1e\$0\$6981\$9b4e6d93@newsspool4.arcor-online.net, 2010-09-12.  
<http://groups.google.com/group/comp.text.tex/msg/87ac1f61321de3ef>
- [4] Heiko Oberdiek, *Re: spacing after \right) and before \left)*, newsgroup comp.text.tex, Message-ID: i6jcc2\$8of\$1@news.eternal-september.org, 2010-09-12.  
<http://groups.google.com/group/comp.text.tex/msg/257aa6119bef878b>

## 7 History

[2010/09/25 v1.0]

- The first version.

## 8 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; plain numbers refer to the code lines where the entry is used.

<b>Symbols</b>	<code>\@</code> . . . . .	243, 316
<code>\#</code> . . . . .	<code>\@PackageError</code> . . . . .	242 145, 159, 163
<code>\%</code> . . . . .	<code>\@ehc</code> . . . . .	318 161

<code>\@ehd</code> .....	147, 201	<code>\ltx@gobblefour</code> .....	177
<code>\@firstofone</code> .....	251, 254	<code>\ltx@gobbletwo</code> .....	178
<code>\@gobble</code> .....	248, 256	<code>\ltx@ifundefined</code> ..	130, 131, 140, 151
<code>\@undefined</code> .....	58	<code>\ltx@zero</code> .....	158
<code>\@</code> .....	317		
<code>\{</code> .....	240		
<code>\}</code> .....	241		
	<b>A</b>		<b>M</b>
<code>\advance</code> .....	281, 289, 304	<code>\mathclose</code> .....	205
<code>\aftergroup</code> .....	29, 212	<code>\mathopen</code> .....	205
		<code>\MessageBreak</code> .....	167, 173, 198
	<b>B</b>	<code>\mleft</code> .....	2, 160, 204, 226, 230
<code>\body</code> .....	260, 264	<code>\mleftright</code> .....	2, 225
		<code>\mleftright@AtEnd</code>	95, 96, 114, 115, 237
	<b>C</b>	<code>\mleftright@Def</code>	130, 204, 209, 225, 229
<code>\catcode</code> 2, 3, 5, 6, 7, 8, 9, 10, 11, 12,		<code>\mleftright@GroupLevel</code> .....	
13, 33, 34, 36, 37, 38, 39, 40, 41,		.....	156, 158, 165, 169, 206, 210
42, 43, 44, 45, 46, 47, 48, 49, 69,		<code>\mleftright@OrgLeft</code> ...	128, 207, 231
70, 72, 73, 74, 78, 79, 80, 81, 82,		<code>\mleftright@OrgRight</code> ..	129, 223, 234
83, 84, 87, 88, 90, 91, 92, 93, 97,		<code>\mleftright@WrongGroup</code>	157, 214, 219
99, 152, 154, 240, 241, 242, 243,		<code>\mleftrightrestore</code> .....	2, 229
278, 287, 295, 299, 316, 317, 318		<code>\mright</code> .....	164, 209, 227, 233
<code>\count@</code> .....	245, 274,		
278, 280, 281, 285, 287, 288,			
289, 293, 295, 298, 299, 303, 304			
<code>\countdef</code> .....	245		
<code>\csname</code> .....	14, 21, 50, 66,		
76, 121, 244, 247, 250, 253, 308, 335			
<code>\currentgrouplevel</code> .	206, 210, 215, 220		
<code>\currentgrouptype</code> ....	211, 216, 221		
	<b>D</b>		<b>N</b>
<code>\DeclareRobustCommand</code> .....	134	<code>\next</code> .....	265, 267, 269
		<code>\number</code> .....	300
	<b>E</b>	<code>\numexpr</code> .....	206
<code>\empty</code> .....	17, 18		
<code>\end</code> .....	336		
<code>\endcsname</code> .....	14, 21, 50, 66,		
76, 121, 244, 247, 250, 253, 308, 335			
<code>\endinput</code> .....	29, 117		
<code>\endlinechar</code> .....	4, 35, 71, 77, 89		
<code>\errmessage</code> .....	297		
<code>\escapechar</code> .....	116, 119		
	<b>I</b>		<b>P</b>
<code>\ifcase</code> .....	175	<code>\PackageInfo</code> .....	26
<code>\ifnum</code> .....	158, 165,	<code>\protected</code> .....	137
171, 210, 211, 280, 288, 295, 303		<code>\ProvidesPackage</code> .....	19, 67
<code>\ifx</code> ....	15, 18, 21, 50, 58, 61, 121,		
230, 233, 244, 247, 250, 253, 308			
<code>\immediate</code> .....	23, 52		
<code>\input</code> .....	122, 123, 309		
<code>\iterate</code> .....	261, 263, 265		
	<b>L</b>		<b>R</b>
<code>\left</code> .....	128, 226, 230, 231	<code>\RangeCatcodeCheck</code> .....	
<code>\LoadCommand</code> .....	309, 319	.....	292, 320, 321, 322, 323, 324,
<code>\loop</code> .....	259, 275, 286, 294		325, 326, 327, 328, 329, 330, 331
<code>\ltx@gobble</code> .....	141, 148	<code>\RangeCatcodeInvalid</code> .....	
		.....	284, 312, 313, 314, 315
		<code>\repeat</code> .....	259, 271, 282, 290, 305
		<code>\RequirePackage</code> .....	125, 126
		<code>\RestoreCatcodes</code> ..	273, 276, 277, 332
		<code>\right</code> .....	129, 227, 233, 234
			<b>S</b>
		<code>\space</code> .....	198, 298, 299, 307
			<b>T</b>
		<code>\Test</code> .....	311, 334
		<code>\the</code> .....	77, 78, 79,
		80, 81, 82, 83, 84, 97, 116, 206,	
		215, 216, 220, 221, 278, 298, 299	
		<code>\TMP@EnsureCode</code> .....	94,
		101, 102, 103, 104, 105, 106,	
		107, 108, 109, 110, 111, 112, 113	
			<b>W</b>
		<code>\write</code> .....	23, 52
			<b>X</b>
		<code>\x</code> 14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87	