

# substr.sty | Substring Functions with L<sup>A</sup>T<sub>E</sub>X

Harald Harders<sup>✓</sup>

March 30, 2010

## Abstract

This documentation of substr.sty has been typeset by automatical ASCII/UTF-8 / L<sup>A</sup>T<sub>E</sub>X enhancement provided by makedoc.sty and niceverb.sty's \auto mode."<sup>1</sup> Some extended formatting | kind of sectioning, switches to \tt | have been achieved by string replacements speci c to the comment text of the package le (see source le substr.tex).

substr.sty especially demonstrates the rather rare `%% ' style of package comments that is somewhat favoured by makedoc dcommentssoe easily343(sodisng | kuishab)-342(from343(some)-

`\SubStringsToCounter{counter}{substring}{string}` Counts the number of appearances of *hsubstringi*

tests if #1 in #2. If yes execute #3, else #4

```

15 \newcommand*\IfSubStringInString[2]{%
16   \su@ExpandTwoArgs{#1}{#2}\su@IfSubStringInString
17 }
18 \newcommand*\su@IfSubStringInString[2]{%
19   \def\su@compare##1##2\nil{%
20     \def\su@param{##2}%
21     \if\su@param\empty
22       \expandafter\@secondoftwo
23     \else
24       \expandafter\@firstoftwo
25     \fi
26   }%
27   \su@compare#2\nil#1\nil
28 }

```

tests if #1 in #2. If yes execute #3, else #4

```

29 \newcommand*\IfCharInString{}
30 \let\IfCharInString\IfSubStringInString

```

returns the part of the string behind the found substring

```

31 \newcommand*\BehindSubString[2]{%
32   \su@ExpandTwoArgs{#1}{#2}\su@BehindSubString
33 }
34 \newcommand*\su@BehindSubString[2]{%
35   \def\su@rest##1##2\nil{##2}%
36   \IfSubStringInString{#1}{#2}{\su@rest#2\nil}{\command*\IfSubStringInString[2]{%

```

