substr.sty*—Substring Functions with LaTeX

Harald Harders†

March 18, 2012

Abstract

This documentation of substr.sty has been typeset by automatical ASCII/UTF-8 → LaTeX enhancement provided by makedoc.sty and niceverb.sty's "auto mode."[1] Some extended formatting—kind of sectioning, switches to `\tt`—have been achieved by string replacements specific to the comment text of the package file (see source file substr.tex).

substr.sty especially demonstrates the rather rare '`%%␣`' style of package comments that is somewhat favoured by makedoc because this way "true comments" are easily distinguishable from mere "commenting out."

```
%%
%% substr.sty
%%
```

This package provides commands to deal with substrings in strings: Determine if a string contains a substring, count appearances of a substring in a string.

Commands:

`\IfSubStringInString{substring}{string}{true part}{false part}` This command searches ⟨*substring*⟩ in ⟨*string*⟩ and executes the ⟨*true part*⟩ if it is and else the ⟨*else part*⟩

`\IfCharInString{char}{string}{true part}{false part}` Actualy the same as `\IfSubStringInString`.

`\BehindSubString{substring}{string}` Returns the part of ⟨*string*⟩ that is on the behind ⟨*substring*⟩. Always the first appearance of ⟨*substring*⟩ is taken.

`\BeforeSubString{substring}{string}` Returns the part of ⟨*string*⟩ that is on the before ⟨*substring*⟩. Always the first appearance of ⟨*substring*⟩ is taken.

`\CountSubStrings{substring}{string}` Counts the number of appearances of ⟨*substring*⟩ in ⟨*string*⟩ and returns it as text.

*See `http://ctan.org/pkg/substr` for more about substr.sty.

†Joint work with Heiko Oberdiek, one line + documentation tool by Uwe Lück.

[1] See `http://ctan.org/pkg/nicetext` for more on these packages.

`\SubStringsToCounter{counter}{substring}{string}` Counts the number of appearances of ⟨*substring*⟩ in ⟨*string*⟩ and sets the counter ⟨*counter*⟩ to that value.

`\IfBeforeSubStringEmpty{substring}{string}{true part}{false part}` Calls ⟨*true part*⟩ if ⟨*substring*⟩ is equal to the beginning of ⟨*string*⟩. Else call ⟨*false part*⟩.

`\IfBehindSubStringEmpty{substring}{string}{true part}{false part}` Calls ⟨*true part*⟩ if ⟨*substring*⟩ is equal to the end of ⟨*string*⟩. Else call ⟨*false part*⟩.

History of this package:

The package arises from a posting of me in the newsgroup `de.comp.text.tex` in which I asked how to find out if a substring is included in a string. Heiko Oberdiek ⟨*oberdiek@ruf.uni-freiburg.de*⟩ posted the commands `\IfSubStringInString` and `\IfCharInString` and suggested to write a command which counts the appearances in a string. So, I wrote the commands `\CountSubStrings` and `\SubStringsToCounter`. After I wrote this package I sent it to Heiko Oberdiek who improved and rewrote many parts of it.

ChangeLog

2009/10/20 v1.2 Uwe Lück - Fix `\IfSubStringInString` which did not work if the string was contained in the substring

2005/11/29 v1.1 Harald Harders - Add `\IfBeforeSubStringEmpty` and `\IfBehindSubStringEmpty`.

2009-10-20 Harald Harders `h.harders@tu-bs.de`

```
\ProvidesPackage{substr}[2009/10/20  v1.2  Handle substrings]
% expands the first and second argument with
% \protected@edef and calls #3 with them:
\newcommand\su@ExpandTwoArgs[3]{%
  \protected@edef\su@SubString{#1}%
  \protected@edef\su@String{#2}%
  \expandafter\expandafter\expandafter#3%
  \expandafter\expandafter\expandafter{%
    \expandafter\su@SubString\expandafter
  }\expandafter{\su@String}%
}
```

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

```
\newcommand*\IfSubStringInString[2]{%
  \su@ExpandTwoArgs{#1}{#2}\su@IfSubStringInString
}
\newcommand*\su@IfSubStringInString[2]{%
  \def\su@compare##1#1##2\@nil{%
    \def\su@param{##2}%
    \ifx\su@param\@empty
      \expandafter\@secondoftwo
    \else
      \expandafter\@firstoftwo
    \fi
  }%
  \su@compare#2\@nnil#1\@nil
}
```

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

```
\newcommand\IfCharInString{}
\let\IfCharInString\IfSubStringInString
```

returns the part of the string behind the found substring

```
\newcommand*\BehindSubString[2]{%
  \su@ExpandTwoArgs{#1}{#2}\su@BehindSubString
}
\newcommand*\su@BehindSubString[2]{%
  \def\su@rest##1#1##2\@nil{##2}%
  \IfSubStringInString{#1}{#2}{\su@rest#2\@nil}{}%
}
```

returns the part of the string before the found substring

```
\newcommand*\BeforeSubString[2]{%
  \su@ExpandTwoArgs{#1}{#2}\su@BeforeSubString
}
\newcommand*\su@BeforeSubString[2]{%
  \def\su@rest##1#1##2\@nil{##1}%
  \IfSubStringInString{#1}{#2}{\su@rest#2\@nil}{#2}%
}
```

calls `#3` if part of string before substring is empty, otherwise calls `#4`.

```
\newcommand*\IfBeforeSubStringEmpty[2]{%
  \su@ExpandTwoArgs{#1}{#2}\su@IfBeforeSubStringEmpty
}
\newcommand*\su@IfBeforeSubStringEmpty[4]{%
  \def\su@rest##1#1##2\@nil{##1}%
  \IfSubStringInString{#1}{#2}{%
    \edef\su@resta{\su@rest#2\@nil}%
    \ifx\@empty\su@resta #3\else #4\fi
  }{#4}%
```

```
}
```

calls `#3` if part of string after substring is empty, otherwise calls `#4`.

```
\newcommand*\IfBehindSubStringEmpty[2]{%
  \su@ExpandTwoArgs{#1}{#2}\su@IfBehindSubStringEmpty
}
\newcommand*\su@IfBehindSubStringEmpty[4]{%
  \def\su@rest##1#1##2\@nil{##2}%
  \IfSubStringInString{#1}{#2}{%
    \edef\su@resta{\su@rest#2\@nil}%
    \ifx\@empty\su@resta #3\else #4\fi
  }{#4}%
}
```

counter for counting appearances

```
\newcounter{su@anzahl}
% #1: String
% #2: Substring
% #3: Counter
\newcommand*\su@StringSubstringCounter[3]{%
  \su@IfSubStringInString{#2}{#1}{%
    \stepcounter{#3}%
    \def\su@rest##1#2##2\@nil{##2}%
    \expandafter\su@StringSubstringCounter\expandafter
    {\su@rest#1\@nil}{#2}{#3}%
  }{}%
}
\newcommand*\CountSubStrings[2]{%
  \su@ExpandTwoArgs{#1}{#2}\su@CountSubStrings
}
\newcommand*\su@CountSubStrings[2]{%
  \setcounter{su@anzahl}{0}%
  \su@StringSubstringCounter{#2}{#1}{su@anzahl}%
  \thesu@anzahl
}
% #1: counter
% #2: substring
% #3: string
\newcommand*\SubStringsToCounter[3]{%
  \su@ExpandTwoArgs{#2}{#3}\su@SubStringsToCounter{#1}%
}
% #1: substring
% #2: string
% #3: counter
\newcommand*\su@SubStringsToCounter[3]{%
  \setcounter{#3}{0}%
  \su@StringSubstringCounter{#2}{#1}{#3}%
}
\endinput
```

EOF