The `alltt` environment*

Johannes Braams

2024/07/07

This file is maintained by the LaTeX Project team.
Bug reports can be opened (category `latex`) at
https://latex-project.org/bugs.html.

Abstract

This package defines the alltt environment, which is like the verbatim environment except that \, {, and } have their usual meanings.

Thus, other commands and environments can appear within an alltt environment.

1 Introduction

`alltt` (*env.*) Here are some things you may want to do in an alltt environment:

- Change fonts–e.g., by typing `\emph{emphasized text}`
- Insert text from a file `foo.tex` by typing `\input{foo}`. Beware that each `<return>` starts a new line, so if `foo.tex` ends with a `<return>` you can wind up with an extra blank line if you're not careful.
- Insert a math formula. Note that `$` just produces a dollar sign, so you'll have to type `\(...\)` or `\[...\]`. Also, `^` and `_` just produce their characters; use `\sp` or `\sb` for super- and subscripts, as in `\(x\sp{2}\)`.

NB When you are using OT1 encoded fonts you might be surprised when you switch to italics, because those fonts have a different set of glyphs:

```
%  The glyph at the position of the $ in a slanted font: $.
%  The glyph at the position of the $ in an italic font: £.
%
```

2 The Implementation

1 ⟨*package⟩

`alltt` (*env.*) The alltt environment is similar to the verbatim environment, except that \, { and } have their usual behaviour.

*This file has version number v2.0g, last revised 2024/07/07.

```
\begingroup
\lccode`\~=`\'
\lowercase{\endgroup
\newenvironment{alltt}{%
  \trivlist
  \item\relax
    \if@minipage
    \else
      \vskip\parskip
    \fi
    \leftskip\@totalleftmargin
    \rightskip\z@skip
    \parindent\z@
    \parfillskip\@flushglue
    \parskip\z@skip
    \@@par
    \@tempswafalse
    \def\par{%
      \if@tempswa
        \leavevmode\null\@@par\penalty\interlinepenalty
    \else
      \@tempswatrue
      \ifhmode\@@par\penalty\interlinepenalty\fi
    \fi}
    \obeylines
    \verbatim@font
    \let\org@prime~%
    \@noligs
    \everymath\expandafter{\the\everymath
      \catcode`\'=12 \let~\org@prime}
    \everydisplay\expandafter{\the\everydisplay
      \catcode`\'=12 \let~\org@prime}
    \let\org@dospecials\dospecials
    \g@remfrom@specials{\\}
    \g@remfrom@specials{\{}
    \g@remfrom@specials{\}}
    \let\do\@makeother
    \dospecials
    \let\dospecials\org@dospecials
    \frenchspacing\@vobeyspaces
    \everypar \expandafter{\the\everypar \unpenalty}}
{\endtrivlist}}
```

`\g@remfrom@specials` In the old implementation of the alltt package a fixed `\dospecials` was used. However nowadays the `\dospecials` command might contain more special characters at run-time then as was defined in the format. Therefore we remove the necessary special character from `\dospecials` at the start of the alltt environment. For this we need a macro. Remember that the list of special characters in `\dospecials` contains the control sequence `\do` between the characters. We use that to check whether a character has to be removed.

The macro `\g@remfrom@specials` takes one argument, the character to be removed from the list.

```
\def\g@remfrom@specials#1{%
```

We build up a new list in `\@new@speicals`.

```
\def\@new@specials{}
```

The command `\@remove` compares its argument with the argument of `\g@remfrom@specials`.

```
\def\@remove##1{%
  \ifx##1#1\else
```

When they are not the same the argument of `\@remove` is added (together with `\do`) to the new list.

```
    \g@addto@macro\@new@specials{\do ##1}\fi}
```

Now we `\let \do` be equal to `\@remove` and execute `\dospecials`.

```
\let\do\@remove\dospecials
```

All that's left is to make `\dospecials` point to the new list.

```
\let\dospecials\@new@specials
}
```

⟨/package⟩