The xnewcommand package

Josselin Noirel
http://www.jnoirel.fr/

21st April 2007 (v1.2)

Contents

Abstract

For most purposes, the features of `\newcommand` suffice. Nonetheless, `\newcommand` lacks some interesting features like the ability to make definition global or to use the `\protected` prefix supplied by ε-TEX. This—surprisingly small—package makes possible to pass an optional argument `\newcommand` so that is becomes possible to use the `\global` and `\protected` prefixes.

Usage

Load the package with

```
\usepackage{xnewcommand}
```

Then `\newcommand` can take an additional optional argument:

`\newcommand[`⟨*prefix*⟩`]{`⟨*command*⟩`}[`⟨*number*⟩`][`⟨*default*⟩`]{`⟨*definition*⟩`}`

where ⟨*prefix*⟩ can be any valid prefix or combination of them:

`\long` The command will accept 'long arguments', in other words, its arguments may contain ends of paragraph (explicit `\par` or empty lines);

`\global` The command is defined globally, in other words, the definition won't be confined to the current group;

`\protected` The command will be robust in moving arguments and won't undergo systematic expansion at the beginning of tabular cells;

`\outer` Makes the macro `\outer` (not very useful, but included for the sake of completeness).

The same syntax applies to the commands `\renewcommand`, `\DeclareRobustCommand`, `\newenvironment` and `\renewenvironment`.

Remarks

It should not break anything. The normal syntax is preserved:

`\newcommand{`⟨*command*⟩`}{`⟨*definition*⟩`}`

is equivalent to

`\newcommand[\long]{`⟨*command*⟩`}{`⟨*definition*⟩`}`

and

`\newcommand*{`⟨*command*⟩`}{`⟨*definition*⟩`}`

is equivalent to

`\newcommand[]{`⟨*command*⟩`}{`⟨*definition*⟩`}`

Any macro that uses `\star@or@long` will inheritate the features described above.

Examples

Global definition:

`\newcommand[\global]{`⟨*command*⟩`}{`⟨*definition*⟩`}`

Protected definition:

`\newcommand[\protected]{`⟨*command*⟩`}{`⟨*definition*⟩`}`

Combination of both:

`\newcommand[\global \protected]{`⟨*command*⟩`}{`⟨*definition*⟩`}`