

easyformat

the manual for the 2017/04/09 version (v1.1.0)

Evert Provoost

Abstract

`easyformat` is a package that allows the use of `_` to begin/end *italic* or **bold** formatting as an alternative to the standard $\text{\LaTeX 2}\epsilon$ `\textit{italic}` and/or `\textbf{bold}`.

1 Introduction

Usually when you want to make something *italic* or **bold** in L^AT_EX you insert `\textit{italic}` or `\textbf{bold}` respectively.

However when writing a book or another text where this formatting is common, this quickly becomes annoying to type.

Quick disclaimer:

I'm not saying that the L^AT_EX way is bad. I just want to explain that it's not ideal under all circumstances.

Simple markuplanguages (eg. Markdown) have a more elegant solution. If you, for example, wanted to write:

Stop!, you *have* to wait!

In Markdown, you'd type:

****Stop!****, you **have** to wait!

As you can see it's a matter of *wanting* italic or bold formatting, adding *'s where needed and done; no (long) macros to type.

With the help of `easyformat` we can do something similar, here we get the same result with:

Stop!, you have to wait!

(Why underscores? See: 4.1)

2 Usage, syntax and examples

Like usual you import the package with: `\usepackage{easyformat}`

`easyformat` makes `_` an active character, however it was already 'special' before so you still have to type `_` if you want to insert an underscore.

As already said, `easyformat` uses a Markdown-like syntax, so:

`_italic, _ _bold, _ _ and _ _bolditalic! _ _`

Gives:

italic, **bold**, and ***bolditalic!***

However we can also do more complex stuff, eg.:

`_Lorem _ _ipsum_ totalem _ _ da _ _givea _zin_ doram _ _zet_tim, liefkan.`

Becomes:

Lorem ipsum **totalem** da **givea zin** **doram zettim**, liefkan.

Those with a good eye might have noticed that `easyformat` adds italic-correction when needed. Just like `\textit{zet}tim` would have.

3 Utilities

3.1 `\nrshape`

Reverts the fontshape to the default.

3.2 `\nrseries`

Reverts the fontseries to the default.

3.3 `\setundact`

Sets the catcode of `_` to 13 (active), effectively *(re-)enabling* the `easyformat` syntax.

3.4 `\setundsub`

Sets the catcode of `_` to 8 (subscript), effectively *disabling* the `easyformat` syntax.

4 Technical details

4.1 Why `_` and not `*`?

To make `*` work we would have to make it an active character, however this would break things like `\section*{Art}`. The `_` is already protected by `TEX` because of its meaning in mathmode. However, it has little to no use in textmode. This is why we can use `_` for this purpose and not `*`.

4.2 Does this break mathmode?

No it doesn't. It keeps behaving like before in mathmode, its meaning only changes in textmode. (We first check whether we are in mathmode or not and behave accordingly.)

5 Contact

If you want to make suggestions or have any questions whose answer could be included in a future version of this document, you can email to this address: <mailto:evert.provoost@gmail.com>

6 Changelog

We only include important changes from v1.0.0 onwards, since earlier versions barely worked.

2017/04/09 v1.1.0

Made usage of syntax in `\section{}`, and similar macros, possible. Changed the behaviour so it works more like `\emph{}` (this also simplifies the code).

2017/04/07 v1.0.0

First stable version.

Made with ♡ in Berlaar, Belgium.