

easyformat

the manual for the 2017/06/03 version (v1.4.0)

Evert Provoost

Abstract

easyformat is a package that allows the use of `_` to begin/end *italics* or **boldface** and `^` to begin/end SMALLCAPS. As an alternative to the standard L^AT_EX `\textit{italic}`, `\textbf{bold}` and `\textsc{smallcaps}`.

Contents

1	Introduction	2
2	Usage, syntax and examples	2
3	Troubleshooting	3
4	Macros	3
4.1	Special characters	3
4.2	Fonts	4
4.3	easyformat-syntax	4
5	Technical details	5
5.1	Why <code>_</code> and not <code>*</code> ?	5
5.2	Does this break mathmode?	5
6	Contact	5
7	Changelog	6

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: 5.1)

And since it is good practice to add abbreviations using SMALLCAPS, `easyformat` gives you `^smallcaps^`.

So:

`^nasa^` and `^esa^` are probably the best known space agencies.

Results in:

NASA and ESA are probably the best known space agencies.

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.

The same can be said of `^`, however since this one is difficult to add in ordinary L^AT_EX, the `easyformat` package gives you `\cir` to get `^`.

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

```
_italic,_ __bold,__ ___bolditalic___ and ^smallcaps!^
```

Gives:

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

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 italics correction when needed. Just like `\textit{zet}tim` would have.

3 Troubleshooting

`easyformat` tends to conflict with the loading of other packages, therefore you should probably load `easyformat` last.

“For some or other reason I get some weird formatting...”

It could be a bug—in which case I’d love to hear from you (see 6)—however, you probably forgot to close your formatting somewhere and because of how `easyformat` works you can get weird results.

4 Macros

4.1 Special characters

`\cir` `\cir`

Insert a circumflex (^) character.

4.2 Fonts

<hr/> <hr/> <code>\nrfamily</code>	<code>\nrfamily</code> Reverts the font family to the default.
<hr/> <hr/> <code>\nrshape</code>	<code>\nrshape</code> Reverts the font shape to the default.
<hr/> <hr/> <code>\nrseries</code>	<code>\nrseries</code> Reverts the font series to the default.
<hr/> <hr/> <code>\setffamily</code>	<code>\setffamily {⟨font family⟩}</code> Quickly change the font family.
<hr/> <hr/> <code>\setfshape</code>	<code>\setfshape {⟨font shape⟩}</code> Quickly change the font shape.
<hr/> <hr/> <code>\setfseries</code>	<code>\setfseries {⟨font series⟩}</code> Quickly change the font series.

4.3 `easyformat-syntax`

<hr/> <hr/> <code>\enableeasyformat</code>	<code>\enableeasyformat</code> Enables the <code>easyformat-syntax</code> .
<hr/> <hr/> <code>\disableeasyformat</code>	<code>\disableeasyformat</code> Disables the <code>easyformat-syntax</code> .
<hr/> <hr/> <code>\setciract</code>	<code>\setciract</code> Sets the catcode of <code>^</code> to 13 (active).
<hr/> <hr/> <code>\setcirsup</code>	<code>\setcirsup</code> Sets the catcode of <code>^</code> to 7 (superscript).
<hr/> <hr/> <code>\setundact</code>	<code>\setundact</code> Sets the catcode of <code>_</code> to 13 (active).
<hr/> <hr/> <code>\setundsub</code>	<code>\setundsub</code> Sets the catcode of <code>_</code> to 8 (subscript).

5 Technical details

5.1 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 `*`.

5.2 Does this break mathmode?

No it doesn't. `_` and `^` keep behaving like before in mathmode, their meaning only changes in textmode. (We first check whether we are in mathmode or not and then behave accordingly.)

6 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>

7 Changelog

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

2017/06/03 v1.4.0

Improved code readability. We now fully use `expl3`, which eliminates most of the possible future issues with `LATEX3`. Fixed a kerning issue with `\cir` in mathmode.

2017/05/28 v1.3.0

Removed forced re-enabling of the `easyformat`-syntax at the start of the document. Rewritten to use `expl3`, this makes the code shorter and a future move to `LATEX3` easier. Fixed an error that could occur with `\cir`. `easyformat` now also behaves correctly when a fontpackage is loaded.

2017/04/17 v1.2.0

Improved handling of mixed styles. Added `^smallcaps` for `SMALLCAPS`.

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.