Introduction to LaFiC

Sebastian Meisel

December 17, 2019

LaFiC means *layout and format in comments*, as all layout and format information is put into comment lines. So layout and content are *fully* separated. For details see Writing text in LaFiC.

1 Why LaFiC

I've been working with LaTeX / XeLaTeX for many years now. Mostly I'm writing prose (no math at all). I often found it disturbing, that I'm forced to create a preamble instead of just start writing.

I started using markdown / multimarkdown. Being quite inflexible it didn't convince me either. Also I didn't like the cryptic syntax so much. The LaTeX output was quite cryptic as well.

The I remembered my father saying, he'd like to be able to just start writing as with his old typewriter only with a better formating to end with.

Last but not least I was thinking a lot about how the layout and formation of a text could be cleaner separated from the content.

With LaFiC I can start writing without a thought about the Layout. Still I get a well structured HTML or (Xe)LaTeX[1] document, that I can further render to PDF.

When I'm ready with writing, I can start format by adding human readable comments, beeing my own lector.

[1] The standard templates for LaFiC are all based on XeLaTeX to support UTF-8. Still using LaTeX should be possible.

Part I

Installation and Usage

2 Prerequisites

LaFiC requires Perl > 5.10.1 (tested with Perl 5.26.1).

The standard templates require a resent XƎLaTeX installation with a least graphicx, hyperref, microtype and xspace.

The Gnu Emacs lisp files where tested with Gnu Emacs 25.2.2.

`LaFiC2pdf` also requires latexmk (tested with version 4.41).

3 Installation

Get source from github using:

```
git clone https://github.com/SebastianMeisel/LaFiC.git
```

Add LaFiC directory to $PATH, e.g.:

```
export PATH=${PATH}:~/LaFiC
```

See `LaFiC-mode.el` for installation instructions, if you want to use in in Gnu Emacs[2].

4 Usage

For now the LaFiC distribution consists of three scripts that you call with the name of the LaFiC file.

```
# LaFiC2html Datei.LaFiC
# LaFiC2tex Datei.LaFiC
# LaFiC2pdf Datei.LaFiC
```

[2]GNU Emacs is available as free Software under a GNU General Public License for most modern operating systems (Unix, GNU/Linux, macOS und Windows).

The last of these is a bash script, first calling `LaFiC2tex` and then `latexmk`.

Calling these three script would result in the following files:

```
Datei.html
Datei.tex
Datei.pdf
```

5 LaFiC major mode in GNU Emacs

After installing and activation `LaFiC-mode.el` (see Installation), the LaFiC major mode is activated on opening any file with a `*.LaFiC` extension.

This gives you basic syntax highlighting and some keyboard shortcuts with a `C-c` prefix. The shortcuts are similar to those used in AUCTeX.

Part II
Writing text in LaFiC

6 Lines and paragraphs

The content is presented in two forms, which also include the most basic layout: There are *lines* and *paragraphs*.

The difference is not so much the length, but lines include none of the punctation marks *(., ?, !, :)*. If no further layout information is provided, these are interpreted as headings.

The *first* line is interpreted as the title and presented as <h1>, when converted to HTML, and \title, when converted to LaTeX .

Further *lines* will be converted to <h3> (HTML) or \section (LaTeX), if no otherwise specified.

This way simple documents may be structured with no explicit layout information at all.

7 Comments

You can add comments to your text, by starting a paragraph with two % chars with **no leading spaces**:

```
%% This is a comment.

%% This is a longer comment, that spreads over several
%% lines. It is important that it is not connected to a line
%% of the general content.
```

8 Formated paragraphs

Paragraphs can be formated by adding a line before the paragraph, that starts with a % char, followed by a single word. There are some predefined keywords, like quote or quotation for – well a quotation. If the keyword is unknown, it will be converted to an environment name in LaTeX or the name of a <div> in Html.

```
% quote
This is a quotation.
```

> This is a quotation.

Two paragraph starting with the same keyword will be concatenated to one block / environment.

```
% center
This paragraph is centered

% center
This one, too.
```

Becomes:

This paragraph is centered
This one, too.

The following keywords are available at the moment:

- quote for quote environment / <blockquote>
- quotation for quotation environment / <blockquote>
- center for center environment / <div class="center">, with text-align=center

9 Formated lines

Line are formated in the same way, only they are converted to macros (LaTeX) oder <span> names (HTML). Know keywords are:

- `"title"`, “h1” or `"heading1"` for \title / <h1>
- `"part"`, “h2”, `"heading"` or `"chapter"`[3] for \part, \chapter / <h2>
- `"section"`, “h3” or `"heading3"` for \section / <h3>
- `"subsection"`, “h4” or `"heading4"` for \subsection / <h4>
- `"subsubsection"`, “h5” or `"heading5"` for \subsubsection / <h5>
- `"paragraph"`, “h6” or `"heading6"` for \paragraph / <h6>
- `"h"` or `"heading"` for \addsec
- `"marginpar"` or `"annote"` for \marginpar / <span class=`"annote"`>

```
% heading4
This is a subsection
```

9.1 This is a subsection

10 Inline formation

If you want to format words or sequences in a paragraph (or line if needed), you add format lines with a leading % after a paragraph. It has two parts:

1. the word or the sequence to be formated in the form start…end.
2. a keyword.

The both are separated by a colon.

```
Hallo dear old world!
% Hallo: bold
% ol…ld: emphasize
```

Becomes: **Hallo** dear *old world*!

Known format keywords are:

[3]Chapter is not available in standard template as it is not available in the document class used.

- `"bold"` for \textbf / <b>
- `"emphasize"` for \emph / <em>
- `"italic"` for \textit / <i>
- `"mono"` or `"typewriter"` for \texttt / <span class=`"tt"`>
- "smallcaps" for \textsc / <span class=`"sc"`>
- "superscript" for \textsuperscript / <sup>
- `"subscript"` for \textsubscript / <sub>

If the keyword is unknown, it is converted to a macro (LaTeX) oder <span> (HTML) name.

Some keywords need a second argument which is added after a second colon:

```
This is a green world!
% green: color: red
```

becomes: This is a green world!

Know keywords of that kind are:

- "url" or "link" for \href / <a href='[url]'>
- "see" for \nameref / <a href='#[label]'>
- "footnote"[4] for \footnote / <a class='fn' href='xfn[x]'>
- "color" for \textcolor / <span style='color: [color]'>

11 Parameters

It is also possible to add some additional parameters to the whole paragraph or line. This is done quite similar to the inline formats, but with a equal sign separating the keyword from the value:

```
This text is white on blue and aligned to the right.
% background = blue
% color = white
% align = right
```

[4]In HTML documents footnote are presented in an <ol> list that is placed in a <div id="footnotes"> container at the end of the document. Each footnote is placed in a <li id="fnx"> element.

becomes:

This paragraph has a red on blue text and is aligned to the right.

Known parameters are:

- “name” or “label” for \label / <?? id=“[id]”> that is referred to by the “see” keyword.
- “background” for \colorbox / <div style=‘ background: [color]’>.
- “color” for \textcolor / <div style=‘ color: [color]’>.
- “align” for \raggedleft, \centering or \raggedright / <div style=‘ text-align: [align]’>.

12 Lists

Lists are the only things, that need some kind of markup: You have to start each topic of the list with one of the following chars: –, *, +, -. It doesn’t matter, which one you choose. You may indent the lines, but that has no influence on the layout.

```
* Top 1.
- Top 2.
```

- Top 1.
- Top 2.

For multilevel lists you have to choises to raise or decrease the level: The clean LaFiC style would be, to start a new paragraph and add the keyword »`% level+`« or »`% level-`« at the end.

```
 * Top 1.
 * Top 2.


 * Top 2a.
 * Top 2b.
 % level+
```

- Top 1.
- Top 2.
  - Top 2a.
  - Top 2b.

Or you can write the list in one paragraph, marking the raise or decrease of the level with a `>` or `<` at the beginning of a single line.

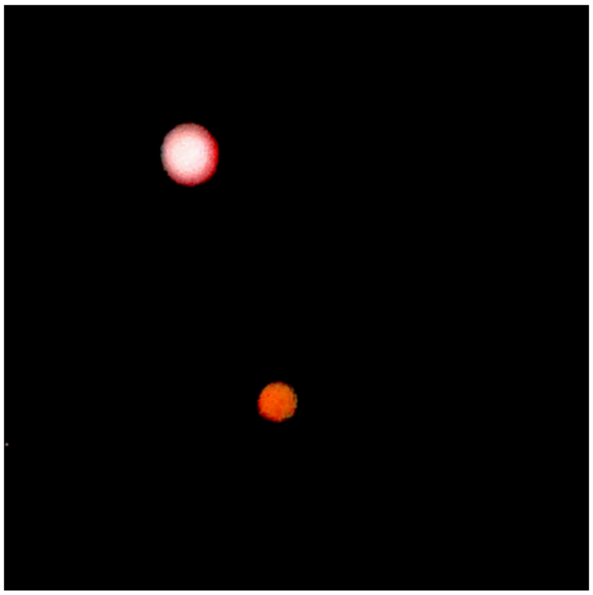
```
* Top 1.
* Top 2.
>
  * Top 2a.
  * Top 2b.
<
* Top 3
```

- Top 1.
- Top 2.
  - Top 2a.
  - Top 2b.
- Top 3

13 Images

The simplest way to put an image into a LaFiC file is a line with the image name, with a know extention: png, jpg, jpeg, gif.

```
Image.png
% height = 40%
% align = center
```



The following parameters are available for formating:

- height
- width or length

Note that this will not put an figure environment in LaTeX files, so the image won't float this way. For this to achieve to have to put % image, %img or %figure before the line.

```
%image
Image.png
% width = 40%
% caption = "Moon and Mars"
```

This also gives you two more parameters to use:

- name or label
- caption

14 Colors

The following colors are accepted as argument to parameters for color or background:

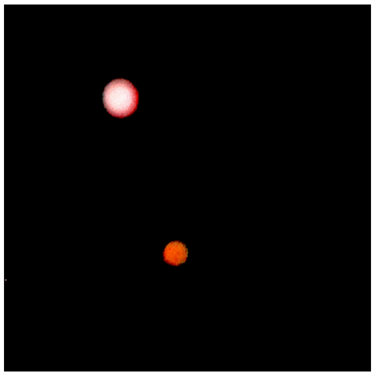


Figure 1: ”Moon and Mars”

- gray
- green
- yellow
- blue
- red
- white

Part III

Customization

It is possible to customize LaFiC in various ways.

15 Templates

LaFiC uses templates for LaTeX and HTML output. These templates are locates in the `/templates` subdirectory of the `/LaFiC` directory. The suffix of these files is .tmp.tex for LaTeX and .tmp.html for HTML output.

You can create your own templates an place them there. LaFiC requires for LaTeX templates at least the `graphicx` package for images, `hyperref` for links and `xcolor` with `x11names, dvipsnames`* and `svgnames` option for color support:

```
\usepackage{graphicx}
\usepackage[x11names, dvipsnames*, svgnames]{xcolor}
\usepackage[hyperindex=false, pdfpagelabels,
                    pageanchor, hyperfootnotes=false, bookmarksopen,
         pdfpagemode=UseOutlines]{hyperref}
```

The templates are called by putting it's name without suffix in the very first line of a document with a leading % char:

```
       % template
```

It is also possible to create a template for a specific document. Such templates must be placed in the same directory as the document and have the same basename:

```
$ ls -1
   .
   ..
   Beispiel.LaFiC
   Beispiel.tmp.tex
   Beispiel.tmp.html
```

The templates must contain placeholders for metadata like title and author and for the content of the document.

For LaTeX that would be %TITLE% and %TEXT%:

```
\documentclass{article}
 …
%TITLE%
\begin{document}
%TEXT%
\end{document}
```

For Html it's <!– TITLE –> and <!– TEXT –>:

In Html-Vorlagen heißen die Platzhalter <!– TITLE –> und <!– TEXT –>:

```
<html>
        <head>
<!-- TITLE -->
        </head>
        <body>
<!-- TEXT -->
        </body>
</html>
```

16 Advanced: Custom keywords

By creating custom LaTeX environment and macros as well as the corresponding css classes, you can use your own keywords.

17 Stylesheets

In the `/styles` subdirectory there is a `standard.css` file, which contains the CSS stylesheet used by the standard HTML templates for LaFiC. You can place your own stylesheets here to refer to them from custom templates. For now you have to manually copy or link the `/styles` directory to your working directory.

18 `LaFiC.config.pl`

In this file in the LaFiC directory you set some preferences. At the moment this are just two:

- `lang` where you have the option to set another `lang`uage then English as your preferred `lang`uage. At the moment the only option here is `de_DE` for German, as this is my first `lang`uage.
- `author` is for now the only way to specify the `author`'s name for your LaFiC documents. I hope to provide an in-document option soon.

The content of this document is actually a perl construct an must be written in the form:

```
key => "value",
```

Please observe the comma at the end of each line.