

# Introduction to LaFiC

Sebastian Meisel

December 11, 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  $\text{\LaTeX}$  /  $\text{\XeLaTeX}$  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  $\text{\LaTeX}$  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<sup>1</sup> 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.

---

<sup>1</sup>The standard templates for LaFiC are all based on  $\text{\XeLaTeX}$  to support UTF-8. Still using  $\text{\LaTeX}$  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 recent  $\text{\LaTeX}$  installation with at least graphicx, hyperref, microtype and xspace.

The Gnu Emacs lisp files were 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 it in Gnu Emacs<sup>2</sup>.

### 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
```

---

<sup>2</sup>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 punctuation 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 L<sup>A</sup>T<sub>E</sub>X .

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 formatted 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 L<sup>A</sup>T<sub>E</sub>X 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 formatted 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"<sup>3</sup> 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 formatted 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:

---

<sup>3</sup>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"<sup>4</sup> 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
```

---

<sup>4</sup>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 choose, 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.

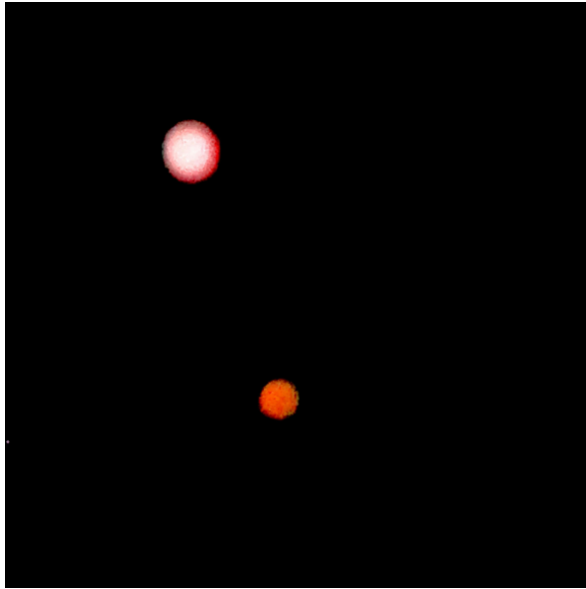
- ```
* Top 1.
* Top 2.
>
  * Top 2a.
  * Top 2b.
<
* Top 3
```
- Top 1.
  - Top 2.
    - Top 2a.
    - Top 2b.
  - Top 3

## 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%
```





Note that this will not put an figure environment in L<sup>A</sup>T<sub>E</sub>X files, so the image won't float this way. For this to achieve to have to put % image, %img or %figure before the line. You don't need the extension then.

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

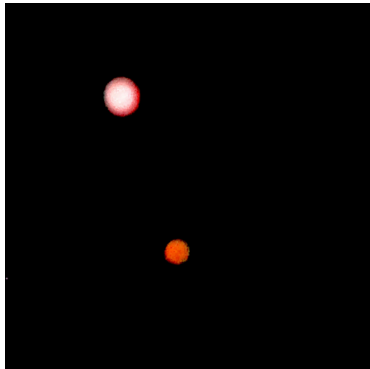


Figure 1: "Moon and Mars"