

Introduction to LaFiC

Sebastian Meisel

June 10, 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.

Part I

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 Emacs¹.

Part II

Writing text in LaFiC

1 Lines and paragraphs

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

¹GNU Emacs ist als freie Software unter der GNU General Public License erhältlich und läuft auf den meisten heute üblichen Betriebssystemen (Unix, GNU/Linux, macOS und Windows).

The difference is not 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^AT_EX .

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

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

2 Formation lines

You can add layout *informations and formations in lines before and after* the paragraph. These *format lines* start with a %-char (with no leading spaces) and must not be separated from the paragraph by blank lines.

2.1 Paragraph and line formats

You can specify a format for a paragraph or line by a leading format line with a single *keyword*. The *keyword* will be translated to a block element name (Html) or a macro / environment (LaTeX). The use of *keywords* is unique to l^aT_EX, as you can define multiple *keywords* for on block element, macro or environment. So *h4*, *heading4* and *subsection* will all be translated to <h4> (Html) and \subsection (LaTeX).

You can force lines (with no ., ?, !, :) to be interpreted as paragraphs with a leading paragraph format (like quotation) or a leading empty format line – just the %-char.

```
% quotation
This is a quotation
```

```
%
This is a paragraph
```

This is a quotation

This is a paragraph

On the other hand you can...

3 Comments

You can add comments to your text, by starting a single line or each *line* of a paragraph with a % char with no leading spaces. These lines or paragraphs must, however, be separated by empty line from the content.

```
% 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.
```

“It is recommended, however, to start comments with two % chars. Else there may occur ‘problems’, when there” is a »:« somewhere in the comment. You also can start a longer comment this way and don’t need to repeat it ‘in every’ line.

```
%% This is a comment! No mistake!  
Even when you go on with no leading % it's still a comment.
```

4 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.

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

```
    This is a quotation.
```

If the keyword is unknown, it will be converted to an environment name in L^AT_EX or the name of a <div> in Html.

5 Formated lines

Line are formatted in the same way, only they are converted to macros (LaTeX) oder `` names (HTML). Known keywords are H1 ... H6 for headings.

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

5.1 This is a subsection

6 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 `»:«`.

Known environment keywords are e.g. quote or quotation.

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

7 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

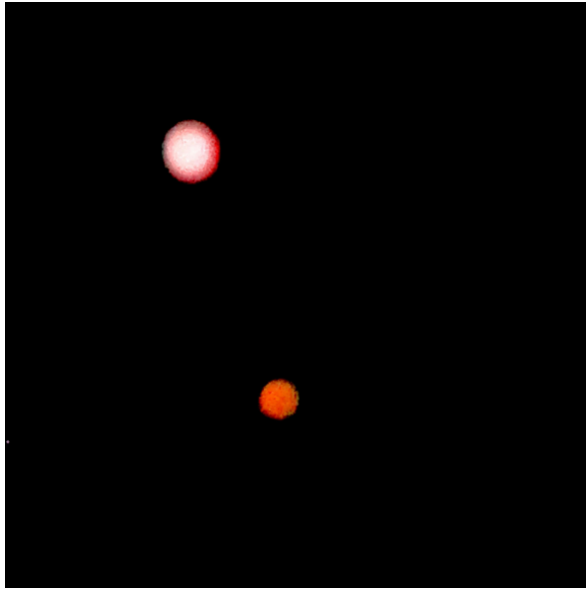
• Top 1.
• Top 2.
  – Top 2a.
  – Top 2b.

• Top 3
```

8 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 \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. You don't need the extension then.

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

