

# Einführung in LaFiC

Sebastian Meisel

December 17, 2019

LaFiC bedeutet *layout and format in comments*, also „Layout und Format in Kommentaren“, denn sämtliche Formatierungen werden in LaFiC in Kommentarzeilen ausgeführt. Das ermöglicht eine *vollständige* Trennung von Inhalt und Gestalt.

## 1. Warum LaFiC

Ich arbeite nun schon viele Jahre mit  $\text{\LaTeX}$  /  $\text{\XeTeX}$ . Ich schreibe vor allem Prosa (ganz frei von mathematischen Formeln). Dabei fand ich es zum Teil störend, dass ich immer erst mit der Preamble beginnen muss und nicht einfach losschreiben kann.

Daraufhin probierte ich markdown / multimarkdown aus. Dabei störte mich aber, dass es sehr unflexibel ist. Außerdem ist die Syntax an manchen Stellen etwas kryptisch, was mich störte. Außerdem funktionieren die erstellten  $\text{\LaTeX}$ -Dateien zwar irgendwie, sind aber auch sehr kryptisch aufgebaut.

Außerdem hatte ich immer die Worte meines Vaters im Ohr, der sagte, er wolle am liebsten, wie einst mit der Schreibmaschine losschreiben können und dennoch ein ordentliches Ergebnis bekommen.

Als letzte Motivation kam hinzu, dass ich immer wieder darüber nachdachte, wie sich Inhalts- und Gestaltungselemente des Textes noch sauberer trennen lassen, als dies mit bisherigen Ansätzen möglich ist.

Mit LaFiC kann ich direkt losschreiben und bekomme von anfang an ein ordentlich strukturiertes Ergebnis als Html- und (Xe)LaTeX-<sup>1</sup> Datei, wodurch auch eine Ausgabe als Pdf möglich ist.

---

<sup>1</sup>Die Standard-Vorlagen, die LaFiC nutzt basieren auf  $\text{\XeTeX}$ , da ich immer in Utf-8 schreibe. Die Nutzung von  $\text{\LaTeX}$  sollte aber auch möglich sein.

Die eigentliche Formatierung geschieht erst im Anschluss. Als mein eigener Lektor gehe ich durch den Text und formatiere ihn mit für Menschen lesbaren und verständlichen Kommentaren.

## Part I.

# Installation und Benutzung

### 2. Voraussetzungen

LaFiC setzt Perl > 5.10.1 voraus (getestet unter Perl 5.26.1).

Das Standard  $\text{\LaTeX}$  -Template setzt eine aktuelle  $\text{\TeX}$  -Installation sowie graphicx, hyperref, microtype und xspace voraus.

Die Gnu-Emacs-Einbindung wurde unter Gnu Emacs 25.2.2 getestet.

lafic2pdf setzt zudem latexmk (getestet mit Version 4.41) voraus.

### 3. Installation

Holen Sie den Quelltext von github:

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

Fügen Sie den Pfad zu \$PATH Umgebungsvariable hinzu:

```
export PATH=${PATH}:/lafic
```

Zur Nutzung des Major Modes in Emacs, kopieren (oder verlinken) die Dateien lafic-mode.el und lafic-german-keywords.el in ein Verzeichnis in Ihrem load-path. Fügen Sie folgende Zeilen in Ihre Init-Datei (~/.emacs) ein:

```
(setq lafic-use-german t)
(require 'lafic-mode)
```

Wenn Sie keine Unterstützung von deutschen Schlüsselwörtern wünschen oder benötigen, lassen Sie die erste Zeile weg.

## 4. Nutzung

Momentan besteht die LaFiC-Distribution aus drei Skripten. Der Aufruf erfolgt jeweils analog:

```
# lafic2html Datei.lafic  
# lafic2tex Datei.lafic  
# lafic2pdf Datei.lafic
```

Letzteres ruft zuerst `lafic2tex` und `latexmk` auf.

Durch den Aufruf dieser drei Skripte würden folgende Dateien erstellt:

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

## 5. Nutzung des LaFiC-Major-Modes in GNU Emacs

Nachdem Sie `lafic-mode.el` wie oben (siehe Installation) beschrieben in Emacs eingebunden haben, wird der LaFiC-Major-Mode beim Öffnen einer Datei mit der Endung `*.lafic` aktiviert.

Dadurch stehen eine einfache Syntaxhervorhebung und einige Tastenkürzel zur Verfügung. Als Prefix wird dabei jeweils `C-c` verwendet. Die Tastenkürzel orientieren sich an AUCTeX, sodass man gut zurecht kommen sollte, wenn man damit schon gearbeitet hat.

# Part II. Schreiben mit LaFiC

## 6. Zeilen und Absätze

Der Inhalt wird durch die Unterscheidung von *Zeilen* und *Absätzen* gegliedert.

Dabei besteht der Unterschied nicht so sehr in der Länge. Vielmehr unterscheiden sich Zeilen von Absätzen dadurch, dass sie kein Satzschlusszeichen (`.`, `?`, `!`, `:`). Wenn nicht anders festgelegt, werden sie als Überschriften interpretiert.

Die erste *Zeile* wird als Titel interpretiert und zu `\title` umgewandelt, wenn die Datei in  $\text{\LaTeX}$  umgewandelt wird, bei HTML-Ausgabe wird es in `<h1>` umgewandelt.

Weitere *Zeilen* werden in `<h3>` (HTML) oder `\section` (LaTeX) umgewandelt, wenn es nicht anders angegeben wird.

Auf diese Weise können einfache Texte ganz ohne Formatierung gegliedert werden.

## 7. Kommentare

Man kann im Text jeder Zeit Kommentare einführen in dem man einen Absatz einfügt, der mit zwei %-Zeichen beginnt:

```
%% Dies ist ein Kommentar.
```

```
%% Dies ist ein längerer Kommentar. Es ist wichtig, dass  
%% Kommentare immer durch eine Leerzeile vom eigentlichen  
%% Inhalt getrennt sind.
```

## 8. Formatierte Absätze

Absätze können formatiert werden, in dem eine Zeile vorangestellt wird, die mit einem %-Zeichen beginnt dem ein Schlüsselwort folgt. Ist das Schlüsselwort bekannt, wird die entsprechende Umgebung (LaTeX), bzw. der entsprechende Block (Html) ausgegeben. Ansonsten dient das Schlüsselwort (umgewandelt in Kleinschreibung) als Name der Umgebung (LaTeX), bzw. eines `<div>`-Blocks (Html).

```
% Zentriert  
Dieser Absatz ist zentriert.
```

Dieser Absatz ist zentriert.

Folgen zwei Absätze mit dem selben Schlüsselwort hintereinander, werden sie in einer Umgebung / einem Block zusammengefasst.

```
% Zitat  
Dies ist ein Zitat.
```

```
% Zitat  
Hier geht das Zitat weiter.
```

Wird zu:

Dies ist ein Zitat.  
Hier geht das Zitat weiter.

Zur Zeit werden folgende Schlüsselworte unterstützt:

- „Zitat“ für quote-Umgebung / `<blockquote>`-Block.
- „Langzitat“ / LZitat für quotation-Umgebung / `<blockquote>`-Block.
- „Zentriert“ / Z für center-Umgebung / `<div class=„center“>`-Block mit `text-align=„center“`
- „Verbatim“ für verbatim-Umgebung / `<pre>`-Block

## 9. Formatierte Zeilen

Zeilen werden auf die gleiche Weise formatiert, nur werden sie in Makros (LaTeX) oder `<span>`-Ids (Html) umgewandelt. Zeilen sind in der Regel Überschriften.

```
% Unterabschnitt  
Das ist ein Unterabschnitt
```

### 9.1. Dies ist ein Unterabschnitt

Folgende Schlüsselwörter werden zur Zeit unterstützt:

- „Überschrift“ für `\addsec` / `<h3>`
- „Titel“, „Überschrift1“ für `\title` / `<h1>`
- „Teil“ / „Überschrift2“ für `\part` / `<h2>`
- „Kapitel“ für `\chapter` / `<h2>`<sup>2</sup>
- „Abschnitt“, „Überschrift3“ für `\section` / `<h3>`

---

<sup>2</sup>Wird nicht von der Standard-Vorlage unterstützt. Es muss eine `\LaTeX`-Klasse verwendet werden, die `\chapter` unterstützt.

- „Unterabschnitt" / „Überschrift4" für \subsection / <h4>
- „Unterunterabschnitt" / „Überschrift5" für \subsubsection / <h5>
- „Absatz" / „Überschrift6" für \paragraph / <h6>

## 10. Textformatierung im Absatz

Wörter und Textabschnitte in einem Absatz werden in einem Block nach dem Absatz formatiert. Jede Zeile beginnt mit einem »%« gefolgt von:

1. dem Wort
2. dem ersten und letzten Wort den Abschnittes, getrennt durch 3 Punkte, bzw. durch »...«.

Sie sind durch einen Doppelpunkt getrennt.

```
Hallo gute alte Welt!
% Hallo: fett
% al...lt: hervorheben
```

wird zu: **Hallo** gute *alte Welt*!

Folgende Schlüsselworte werden zur Zeit unterstützt:

- „fett" für \textbf / <b>
- „hervorheben" für \emph / <em>
- „kursiv" für \textit / <i>
- „Schreibmaschine" oder „nicht proportional" für \texttt / <span class=,tt'>
- „Kapitälchen" für \textsc / <span class=,sc'>
- „hochgestellt" für \textsuperscript / <sup>
- „tiefgestellt" für \textsubscript / <sub>

Unbekannte Schlüsselworte werden in Makros (LaTeX) oder <span>-Elemente (Html) umgewandelt.

Einige Schlüsselworte benötigen ein zweites Argument, dass nach einem zweiten Doppelpunkt angefügt wird:

```
Die Welt ist grün!  
% grün: Farbe: rot
```

Wird zu: Die Welt ist **grün!**

Folgende solcher Schlüsselworte werden zu Zeit unterstützt:

- „url“ oder „link“ für \href / <a href=,[url]>
- „siehe“ für \nameref / <a href='#[label]'>
- „Fußnote“ für \footnote / <a class=,fn href=,xfn[x]<sup>3</sup> >
- „Farbe“ für \textcolor / <span style=,color: [color]>

## 11. Parameter

Es ist weiterhin möglich einer Zeile oder einem Absatz zusätzliche Parameter hinzuzufügen. Die geschieht äquivalent zu Formatierung im Absatz, nur dass ein Ist-gleich-Zeichen statt des Doppelpunktes gesetzt wird:

```
Dieser Text ist weiß auf blauem Untergrund und  
rechtsbündig.  
% Hintergrundfarbe = blau  
% Farbe = weiß  
% Ausrichtung = right
```

wird zu:

**Dieser Text ist rot auf blauem Untergrund und rechtsbündig.**

Folgende Parameter werden derzeit unterstützt:

- „Name“, „Bezeichnung“ oder „Label“ für \label / <?? id=,[id]> that is referred to by the „see“ keyword.
- „Hintergrund“ für \colorbox / <div style=,background: [color]>.
- „Farbe“ für \textcolor / <div style=,color: [color]>.

---

<sup>3</sup>In Html-Dokumenten werden Fußnoten in einer <ol>-Liste in einem <div id=„footnote“>-Container am Ende des Dokumentes dargestellt. Die einzelnen Fußnoten stehen in einem <li id=„fn[x]“>-Element.

- „Ausrichtung“<sup>4</sup> für \raggedleft, \centering or \raggedright / <div style=,text-align:[align]>.

## 12. Listen

List sind das Einzige, das eine Art Auszeichnung benötigt: Sie müssen jeden Punkt in der List mit einem der folgenden Zeichen beginnen: -, \*, +, -. Dabei spielt es keine Rolle, welches Sie wählen. Sie können Zeilen einrücken, aber das beeinflusst das Layout nicht.

- \* Top 1.
- Top 2.
- Top 1.
  - Top 2.

Für Listen mit mehreren Ebenen haben Sie zwei Möglichkeiten, die Ebene zu wechseln: Die reine LaFiC-Stil wäre, einen neuen Abschnitt einzufügen und dahinter eines der Schlüsselworte »% Level+« oder »% Level-« anzuhängen.

- \* Top 1.
- \* Top 2.
- \* Top 2a.
  - \* Top 2b.
  - % level+
  - Top 1.
  - Top 2.
  - Top 2a.
    - Top 2b.

Die andere Möglichkeit ist die Liste in *einem* Abschnitt zu setzen und die Ebenenwechsel mit einem Größer- bzw. Kleiner-als-Zeichen in einer ansonsten leeren Zeile anzuzeigen:

---

<sup>4</sup>Die eigentlich Ausrichtung muss derzeit noch auf Englisch angegeben werden.



```

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

• Top 1.

• Top 2.
  – Top 2a.
  – Top 2b.

• Top 3

```

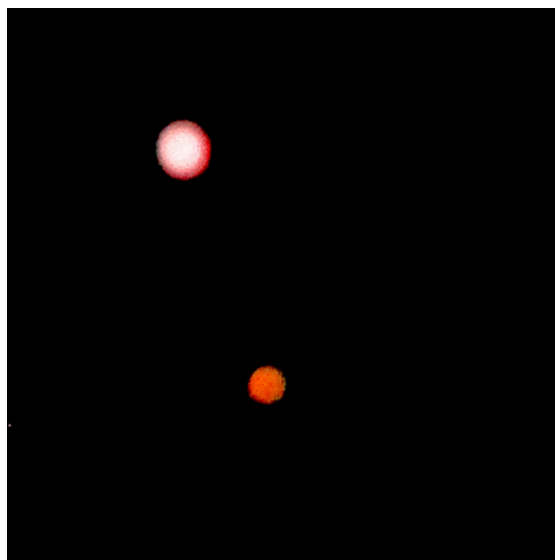
### 13. Bilder

Die einfachste Möglichkeit ein Bild in eine LaFiC-Datei einzubinden ist es, eine Zeile mit dem Namen des Bildes mit einer der Dateiendungen png, jpg, jpeg oder gif einzufügen:

```

Image.png
% Breite = 50%
% Ausrichtung = center

```



Zur Formatierung stehen folgende Parameter zur Verfügung:

- Höhe
- Breite
- Ausrichtung

Auf diese Art wird das Bild aber nicht als Flusselement eingefügt. Dafür muss das Bild mit %Bild formatiert werden.

```
% Bild
Image.png
% Höhe = 40%
% Beschriftung = Mars and Moon
% Name = bild1
% Ausrichtung = right
```

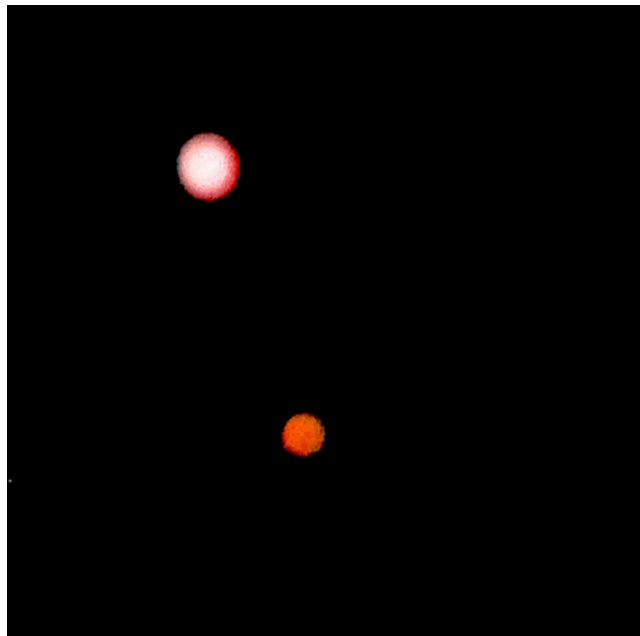


Figure 1: Mars und Mond

Damit stehen dann zwei weitere Parameter zur Verfügung:

- Name
- Beschriftung

## 14. Farben

Folgende Farben werden zur Zeit als Argument für die Parameter Farbe und Hintergrund unterstützt:

- grau
- grün
- gelb
- blau
- rot
- weiß

## Part III. Anpassung

LaFiC lässt sich auf verschiedene Weise anpassen.

## 15. Vorlagen

LaFiC nutzt Vorlagen. Diese liegen im Unterverzeichnis `/templates` des LaFiC-Verzeichnisses und tragen die Dateiendung `.tmp.tex` für  $\text{\LaTeX}$  -, bzw. `.tmp.html` für Html-Dokumenten. Für deutschsprachige Projekte sollten die Dateinamen auf `_de_DE` enden.

Sie können eigene Vorlagen erstellen und in diesem Verzeichnis ablegen. LaFiC erwartet in  $\text{\LaTeX}$ -Vorlagen dabei zumindest `graphicx` für das Einbinden von Bildern, `hyperref` für die Nutzung von Links und `xcolor` mit den Optionen `x11names`, `dvipsnames*` und `svgnames`:

```
\usepackage{graphicx}
\usepackage[x11names, dvipsnames*, svgnames]{xcolor}
\usepackage[hyperindex=false, pdfpagelabels,
```

```

        pageanchor, hyperfootnotes=false, bookmarksopen,
pdfpagemode=UseOutlines]{hyperref}

```

Die Vorlage wird aufgerufen, indem der Dateiname ohne Dateiendung nach einem Prozentzeichen in die erste Zeile eines Dokumentes eingefügt wird:

```
% Vorlage_de_DE
```

Außerdem ist es möglich Vorlagen für eine konkretes Dokument zu erstellen, oder anzupassen. Solche Vorlagen müssen im selben Verzeichnis wie das Dokument liegen und denselben Dateinamen tragen:

```

$ ls -1
.
..
Beispiel.lafic
Beispiel.tmp.tex
Beispiel.tmp.html

```

In den Vorlagen müssen Platzhalter für die Metadaten und den eigentlichen Dokumenttext enthalten sein.

In  $\text{\LaTeX}$ -Vorlagen sind das %TITLE% und %TEXT%:

```

\documentclass{article}
...
%TITLE%
\begin{document}
%TEXT%
\end{document}

```

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

```

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

```

## 16. Für Fortgeschrittene: Eigene Schlüsselwörter

Wenn sie eigene  $\text{\LaTeX}$  -Umgebungen und Makros, sowie die dazu passenden CSS-Klassen definieren, können Sie eigene Schlüsselwörter benutzen.

## 17. Stylesheets

Im Unterverzeichnis `/styles` befindet sich die Datei `standard.css`. Sie enthält das Stylesheet für die Standard-HTML-Vorlagen von LaFiC. Sie können hier eigene Stylesheets ablegen und Ihren Vorlagen verlinken. Zur Zeit muss das Verzeichnis von Hand in das Arbeitsverzeichnis kopiert oder dort verlinkt werden.

## 18. `lafic.config.pl`

In dieser Datei im LaFiC-Verzeichnis können Voreinstellungen vorgenommen werden. Momentan gibt es nur zwei (mögliche) Einträge:

- `lang` für die Standardsprache. Statt dem Englisch wird hier zur Zeit nur `de_DE` als Eintrag für die Deutsche Sprache unterstützt.
- `author` ist Momentan die einzige Möglichkeit den Autor für ihre Dokumente zu setzen.

Die Einträge in dieser Datei sind Perlkonstrukte und müssen in der Form:

```
Schlüssel => "Wert",
```

geschrieben werden. Bitte beachten sie das Komma am Ende der Zeile.