

Viele Ziele Veröffentlichung

Axel Kielhorn

2013-02-03

Anmerkung

Dieser Artikel wurde ursprünglich für die dante Vereinszeitschrift „Die TeXnische Komödie“ 3/2011 in LaTeX geschrieben. Eine englische Übersetzung erschien im „TUGboat“ Volume 32 (2011), No. 3.

Mit dem Erscheinen von pandoc 1.9 wurde er nach `markdown` konvertiert. Durch eine angepasste `template` Datei wird das Aussehen des Originalartikels nachgebildet.

Ein Weg führt zu einem Ziel

Bisher war das Ziel meiner Veröffentlichungen immer Papier: DIN A4, DIN A5 oder auch mal 9 cm × 12 cm. Auf dem Weg dahin entstand immer eine PDF-Datei, daher lag es nahe, diese am Bildschirm zu lesen, was zumindest bei DIN A5 bequem möglich ist.

Doch dann kamen die Mobilgeräte. Einige davon können auch PDF anzeigen und mit etwas Aufwand kann man den Text so formatieren, das er auf *einem* Mobilgerät mit wenig scrollen lesbar ist.

Günstiger wäre natürlich ein Format, bei dem der Leser die Textgröße bestimmen kann und das Gerät den Text passend zur Anzeigengröße umbricht. Ein solches Format ist z. B. ePUB. Im Prinzip ist das nichts anderes als ein ZIP-Archiv mit einer definierten Struktur und ein paar XML-Dateien, die den eigentlichen Text enthalten. Das Aussehen kann durch eine css-Datei gesteuert werden.

Eine Umleitung

Zum Glück gibt es ein Programm, das LaTeX lesen und ePUB schreiben kann: pandoc^{[5]¹}. Wenn die LaTeX-Datei nicht zu kompliziert ist, kann pandoc sie verstehen und konvertieren. Aber was ist zu kompliziert? Am einfachsten konvertiert man ein Dokument von LaTeX nach LaTeX und sieht was übrig bleibt.

```
pandoc -f latex -t latex --template=./default-de.latex
-o quelle-pd.tex quelle.tex
```

Dieser Befehl benutzt eine an die deutsche Sprache angepasste Version der Standardvorlage `default.latex` um eine neue LaTeX-Datei zu erstellen.²

¹pandoc ist unter GPL lizenziert.

²Das Begleitmaterial zu diesem Artikel befindet sich auf <https://github.com/AKielhorn/Markdown-Intro/blob/master/Ziele.zip>

Die Steine auf dem Weg

Pandoc arbeitet mit UTF-8 Dateien. Latin-1 Texte lassen sich leicht konvertieren, Umlaute gemäß `german.sty` gehen jedoch verloren. Auch TeX-Akzente `\^o` oder Einbuchstabenbefehle `\o` führen manchmal zu Problemen. Das lässt sich aber mit ein paar Zeilen `sed` beheben (siehe Abschnitt Reisevorbereitung).

Einfache Auszeichnungsbefehle für **fett** und *kursiv* werden unterstützt, doch schon beim geschachteltem `\emph` ist Schluss.

Ein neuer Anfang

Steht die LaTeX-Datei wirklich am Anfang? Oder sollten wir nicht lieber LaTeX als *ein* Backend verstehen und die LaTeX-Datei somit nur als ein Zwischenprodukt?

Eine ungewöhnliche Richtung: Markdown statt markup

Markdown ist eine von John Gruber^[3] entwickelte Auszeichnungssprache, die so aussieht, als ob sie keine Auszeichnungen enthält:

A Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions.

Der Anfang dieses Artikels sieht in Markdown so aus:

```
# Ein Weg führt zu einem Ziel
```

```
Bisher war das Ziel meiner Veröffentlichungen immer Papier: DIN A4, DIN A5 oder auch mal 9 cm  $\times$  12 cm. Auf dem Weg dahin entstand immer eine PDF-Datei, daher lag es nahe, diese am Bildschirm zu lesen, was zumindest bei DIN A5 bequem möglich ist.
```

```
Doch dann kamen die Mobilgeräte. Einige davon können auch PDF anzeigen und mit etwas Aufwand kann man den Text so formatieren, das er auf einem Mobilgerät mit wenig scrollen lesbar ist.
```

Dieser Text wurde mit dem Befehl

```
pandoc -f latex -t markdown -o Ziele.md Ziele.tex
```

direkt aus der LaTeX-Datei erzeugt.

Markdown ist eine sehr eingeschränkte Sprache, die Manpage, die den Sprachumfang beschreibt ist nur 16 Seiten lang. Im Vergleich dazu ist `l2kurz` dreimal so lang und kratzt gerade mal an der Oberfläche von LaTeX.

Bei der Konvertierung von LaTeX zu Markdown sollte man beachten, dass pandoc kein TeX versteht. Es nutzt Reguläre Ausdrücke um LaTeX zu interpretieren. Das hat zur Folge, dass bei einigen Befehlen zusätzliche Leerzeilen erforderlich sind, damit pandoc die Befehle von normalem Text unterscheiden kann.

Auf neuem Weg zum alten Ziel: PDF aus Markdown über LaTeX

```
pandoc -f markdown -t latex --template=./default-de.latex
-o quelle.tex quelle.md
```

Die default-de.latex Dateien

Die mitgelieferte `default.latex` Datei ist eine Minimalversion. Für deutsche Texte empfiehlt sich die erweiterte `default-de.latex` Datei, die im Begleitmaterial [4] zu finden ist. Die Änderungen zur Originalversion sind durch ein `-ak-` gekennzeichnet.

Seit Version 1.9 liefert Pandoc eine universelle Template-Datei mit. Diese kann durch zuweisen einer Variablen auf die gewünschte Sprache eingestellt werden. Das Argument ist hier eine von `babel` unterstützte Sprache.

```
pandoc -f markdown -t latex -V lang=ngerman -s
-o quelle.tex quelle.md
```

In diesem Fall muss `pandoc` mit der Option `-s` (standalone) mitgeteilt werden, das ein vollständiges Dokument erstellt werden soll.

Leider fehlt in der offiziellen `default.latex` Datei die Zeile

```
\usepackage[T1]{fontenc}
```

daher ist `default-de.latex` immer noch erforderlich.

Die T1 Zeichensatzkodierung ist notwendig, da OT1 die Anführungszeichen unten nicht enthält. Französische Anführungszeichen, «Das sind diese» funktionieren zwar ohne `fontenc`, das Ergebnis sieht aber furchtbar aus.

Mit Version 1.9.4 wurde die template Datei erweitert und enthält nun die Zeilen

```
\usepackage[T1]{fontenc}
\usepackage{lmodern}
```

Damit ist die Datei `default-de.latex` nicht mehr erforderlich, alle Einstellungen können über Variablen vorgenommen werden.

Anführungszeichen führen noch zu einem weiteren Problem. Als TeX noch auf eine 7-Bit Eingabekodierung beschränkt war, wurden einige Zeichen über Ligaturen eingegeben. Dazu zählten auch die englischen Anführungszeichen “ und ”. Aus Kompatibilitätsgründen gibt Pandoc die Anführungszeichen immer noch so aus. In deutschen Texten werden die “ Zeichen als *schließende* Anführungszeichen verwendet. Endet nun ein Zitat mit einem ! oder einem ?, so greift ein anderer Ligaturmechanismus, der die Kombination ! ‘ in ein ¡ und ? ‘ in ein ¿ umwandelt. Dies lässt sich umgehen, indem man bei der Eingabe " benutzt. Wenn im LaTeX template das Paket `csquotes` benutzt wird, werden die geraden Anführungsstriche in `\enquote` Befehle übersetzt. In Verbindung mit der Option `german` werden daraus dann die richtigen Anführungsstriche. Bei anderen Formaten muss man damit leben, das hier englische Anführungsstriche verwendet werden.

Alternativ kann man natürlich die LaTeX Datei mit `sed` bearbeiten und die Zeichen nach Unicode konvertieren.

Bei der Verwendung von französischen Anführungszeichen gibt es keine Probleme.

Seit pandoc 1.9.3 gibt es einen neuen Schalter `--no-tex-ligatures` mit dem die Übersetzung von Unicode-Zeichen in TeX-Ligaturen abgeschaltet werden kann. „, und “ werden dann unverändert in den TeX Text übernommen. Gleichzeitig wird die Übersetzung von `--` in `-` und `---` in `--` abgeschaltet. Diese Zeichen müssen dann durch TeX konvertiert werden. Bei `pdflatex` ist das standardmäßig der Fall, bei `XeTeX` und `LuaTeX` muss die `fontspec` Option `Ligatures=TeX` gesetzt werden.

Dieser Schalter schaltet auch die Option `--smart` ab. Schreibmaschinenanführungszeichen " werden dann nicht mehr in typographische Anführungszeichen konvertiert. Möchte man diese Funktion erhalten, so ist die Option `--smart` (Kurzform `-S`) zusätzlich anzugeben.

Eine einfache Möglichkeit, die komfortable Eingabe von " zu nutzen und gleichzeitig richtige Anführungszeichen im Zieldokument zu erhalten ist:

```
pandoc --smart file.md -t markdown
| sed -f smart2de
| pandoc -f markdown -o file.epub
```

Die Datei `smart2de` enthält die folgenden Befehle:

```
s/"/,/g
s/'/'/'g
```

Natürlich kann man alternativ auch » und « als Ersetzungstext benutzen

Abkürzung

Der schnellste Weg aus einer Markdown-Datei ein PDF zu erstellen ist:

```
pandoc --template=./default-de.latex -o quelle.pdf quelle.md
```

Am Zielformat `pdf` erkennt pandoc, dass es die PDF-Datei direkt erzeugen soll. Im Hintergrund wird natürlich LaTeX aufgerufen, es bleiben jedoch keine temporären Dateien zurück. Wird eine Inhaltsverzeichnis gewünscht, ist ein zweiter LaTeX-Lauf erforderlich. Pandoc erkennt das am Befehl `\tableofcontents`.

Mit den Schaltern `--xetex` bzw. `--luatex` kann man von pdfTeX auf XeLaTeX oder LuaLaTeX umschalten. Im `default.latex` wird mit Hilfe von `ifxetex` und `ifluatex` ermittelt, welches TeX verwendet wird.

Auf diesem Weg kann man natürlich kein BibTeX/Biber verwenden. Zum Erstellen des Literaturverzeichnisses muss man auf die pandoc internen Funktionen zurückgreifen.

Bei älteren pandoc Versionen übernimmt das mitgelieferte Programm `markdown2pdf` die Konvertierung nach PDF.

```
markdown2pdf --template=./default-de.latex quelle.md
```

Nacharbeit

Die automatisch erstellte LaTeX-Datei ist fast so gut wie eine von einem Anfänger manuell erstellte. Natürlich wird man auch hier mit Trennhilfen und ähnlichem die overfull und underfull hboxes beseitigen müssen.

Der Weg zu einem neuen Ziel: ePUB

Der ursprüngliche Wunsch war es, eine ePUB Datei zu erstellen. Dies geschieht mit dem Befehl:

```
pandoc -f markdown -t epub --epub-cover-image=cover-image.gif -s
-o Ziele.epub Ziele.md
```

Der Text wird anhand der Gliederungsbefehle in verschiedene Dateien aufgeteilt, das erleichtert die Nachbehandlung, z. B. mit Sigil [7].

Seit Version 1.8.1.2 kann ein Titelbild mit der Option `--epub-cover-image` eingebunden werden.

Damit ein eBook-Reader den Text richtig umbrechen und bei Bedarf trennen kann, ist es erforderlich ihm die Dokumentsprache mitzuteilen. Pandoc wertet die Umgebungsvariable `$LANG` aus und setzt die Sprache entsprechend. Auf einem deutschen Rechner hat diese Variable normalerweise den Wert `de_DE`, in Österreich den Wert `de_AT`.

Möchte man einen Text in einer anderen Sprache schreiben, so kann man diesen Wert durch zuweisen einer Variablen ändern.

```
pandoc -f markdown -t epub -s -V lang=en_US
-o Ziele.epub Ziele.md
```

führt zu einem Text in amerikanischem English.

Auf einigen ebook Readern kann es zu Darstellungsproblemen kommen, da die installierten Zeichensätze nur einen kleinen Teil des Unicode Zeichenvorrats enthalten. Feste Abstände, wie sie z. B. im Mathematiksatz verwendet werden, erscheine dann als Fragezeichen. Oft fehlt auch der Pfeil mit Haken, der als Zurück-Symbol in Fußnoten verwendet wird.

Bei den meisten Geräten lassen sich eigene Zeichensätze installieren. Gute Erfahrungen habe ich mit GNU Freefont (<http://savannah.gnu.org/projects/freefont/>) und DejaVu (<http://dejavu-fonts.org/wiki/index.php>) gemacht.

Ein Lagerfeuer

Es gibt zwei Wege, um aus ePUB Dateien Dokumente für den Kindle³ zu erstellen:

1. Der offizielle Amazon Weg führt über das Programm `kindlegen`, das von der Amazon Webseite heruntergeladen werden kann. Ab pandoc Version 1.9 kann `kindlegen` die von pandoc erzeugten Dateien direkt in das `mobi`-Format übersetzen.

`Kindlegen` mag es nicht, wenn als Sprache `de_DE` angegeben ist. Soll ein ePUB später mit `kindlegen` weiterverarbeitet werden, so ist als Sprache `de` für deutsch, ohne die Landesangabe DE oder AT zu wählen. Schweizer Deutsch `de_CH` ist jedoch in Ordnung, es wird wahrscheinlich wegen des fehlenden ß anders behandelt. Beim Aufruf von pandoc ist die Option

```
-V lang=de
```

anzugeben.

Ältere pandoc Versionen erzeugen ein ePUB, bei dem `kindlegen` Probleme hat, wenn im Titel oder den Kapitelüberschriften Umlaute vorkommen. Man kann das Problem umgehen,

³Kindle ist wahrscheinlich ein eingetragenes Warenzeichen von Amazon, auch wenn ich auf den ersten Blick keinen entsprechenden Hinweis auf der Webseite gefunden habe.

indem man das Dokument in **sigil** öffnet und gleich wieder speichert. Alternativ kann man es auch mit den Kommandozeilenwerkzeugen aus **calibre** [2] von ePUB nach ePUB konvertieren. Danach verarbeitet **kindlegen** die Datei ohne Probleme.

```
ebook-convert quelle.epub quelle_fixed.epub
-no-default-epub-cover
```

Mit dem Erscheinen des Kindle fire hat Amazon ein neues Dateiformat KF8 eingeführt. Zur Konvertierung in dieses Format ist **kindlegen** Version 2.0 erforderlich.

2. Eine Open Source Alternative ist das Programm **calibre** [2]. Im Menüpunkt **Einstellungen** kann man unter **Erweitert-> Verschiedenes-> Kommandozeilen-Tools** installieren Kommandozeilenwerkzeuge installieren. Diese stehen dann z. B. für ein Makefile zur Verfügung. Beim Ausgabeformat **mobi** für den Kindle lautet die Kommandozeile:

```
ebook-convert quelle.epub quelle.mobi --output-profile=kindle
```

Die erstellte Datei ist größer als bei **kindlegen**, da der Text weniger gut komprimiert wird. Mehr Informationen zu dem Befehl erhält man mit:

```
ebook-convert quelle.epub quelle.mobi -h
```

Sehen wo ich gehe: WYSIWYG

„Kann ich das als Word-Datei haben?“ Wer kennt diese Frage nicht? Mit **pandoc** 1.9 ist es möglich eine **docx** Datei zu erstellen:

```
pandoc -f markdown -t docx --reference-docx=./reference-de.docx -s
-o quelle.odt quelle.md
```

Die Sprache der Datei ist englisch und muss nach jeder Konvertierung in Formatvorlage **Standard** auf deutsch zurückgesetzt werden.

Wer lieber mit StarOffice / OpenOffice / LibreOffice arbeitet, kann mit

```
pandoc -f markdown -t odt --reference-odt=./reference-de.odt -s
-o quelle.odt quelle.md
```

eine passende Datei erstellen.

Die Dateien **reference-de.docx** bzw. **reference-de.odt** dienen hier als Basis für die Absatzvorlagen und das Dokumentlayout. Dies können eine beliebige Dateien sein, am einfachsten ist es jedoch eine von **pandoc** erstellte Datei an die eigenen Layoutwünsche anzupassen. So ist sichergestellt, dass die internen Bezeichnungen der Absatzformate mit den von **pandoc** vergebenen übereinstimmen.

Die Datei **reference-de.odt** basiert auf einer von **pandoc** erstellten Datei, bei der die Sprache auf deutsch umgestellt wurde. Bis **pandoc** 1.8.1.2 gab es ein Problem beim Einbinden von Bildern. OpenOffice meldet eine korrupte Datei und bietet an, diese zu reparieren. Dieser Fehler wurde in Version 1.8.1.3 behoben.

Eingebundene Bilder werden auf Einheitsgröße zusammengestaucht und müssen jeweils einzeln auf Originalgröße expandiert werden. Dieser Fehler wurde mit **pandoc** 1.9 behoben.

Reisevorbereitung

Mit einem kleinen `sed`-Skript lässt sich diese `dtk`-basierte LaTeX-Quelle in eine neutrale LaTeX-Datei umwandeln, die von `pandoc` weitgehend verlustlos gelesen werden kann. Lediglich die `bibliography`-Umgebung geht verloren.

```
s/\\LaTeX/LaTeX/g
s/\\TeX/TeX/g
s/\\ConTeXt/ConTeXt/g
s/\\Program{\\([a-zA-Z]*\\)}/\\1/g
s/\\Acronym{\\([a-zA-Z]*\\)}/\\1/g
s/\\Package/\\texttt/g
s/[style=DTKlstNoNumber]//
s/\\File/\\texttt/g
s/\\Macro{/\\texttt{\\textbackslash /g
s/\\begingroup//
s/\\endgroup//
```

Mit dem Aufruf

```
sed -f dtk2mdtex.sed Ziele.tex >Ziele-clean.tex
```

werden die LaTeX-Befehle, die `pandoc` nicht kennt, aus der Quelldatei entfernt. Das Ergebnis kann dann mit

```
pandoc -f latex -t markdown -s -o Ziele-clean.md Ziele-clean.tex
```

nach Markdown konvertiert werden.

Wegbeschreibung

Gliederungsbefehle

Markdown unterstützt bis zu 6 Gliederungsebenen. Die Gliederungsstufe wird durch die Anzahl der `#` Zeichen vorgegeben. Vor jeder Überschrift ist eine Leerzeile erforderlich.

```
# Oberste Ebene

## Zweite Ebene

### Dritte Ebene

#### *Wichtige Information* in der vierten Ebene versteckt
```

Eine alternative Form der Gliederungsbefehle unterstützt nur zwei Ebenen:

```
Erste Ebene
=====

Zweite und letzte Ebene
-----
```

Weiter Ebenen müssen dann durch `#` Zeichen angegeben werden. Standardmäßig erzeugt `pandoc` beim konvertieren nach Markdown dieses Format. Mit der Option `--atx-headers` werden Überschriften im ersten Format erstellt.

Zitatumgebung

Zitiert wird wie in (alten) E-Mail Programmen mit einem >

```
> Hier zitiere ich eine alte E-Mail
>
> > Und hier wird in der alten E-Mail eine noch ältere zitiert.
>
> Normalerweise erhält jede Zeile ein >-Zeichen. Einige Editoren
> fügen dann in der Folgezeile das > gleich mit ein.
>
> Sollte ein Editor nicht über diese Funktion verfügen,
reicht es, nur die erste Zeile eines Absatzes zu markieren.
```

Der Text wird dabei normal umgebrochen. Möchte man den Zeilenumbruch der Quelldatei im fertigen Dokument erhalten, so gibt es dafür den Zeilenblock. Er wird durch | eingeleitet. Somit lassen sich Gedichte formatieren. Leerzeichen am Zeilenanfang bleiben erhalten. Anders als Programmlisings, die in Schreibmaschinenschrift gesetzt werden, wird hier die normale Textschrift benutzt.

```
| Vom Eise befreit sind Straßen und Wege,
| Durch des Salzes ätzende Kraft.
|     Und so weiter.
```

Eine besondere Form von Zitaten sind Zitate aus Programmen. Diese werden standardmäßig in einer nichtproportionalen Schrift gesetzt. Programmzitate werden durch 4 Leerzeichen oder einen Tabulatorschritt eingegeben:

```
    \documentclass[11,ngerman]{dtk}
    \usepackage[utf8]{inputenc}
```

Auch vor Zitaten ist eine Leerzeile erforderlich.

Will man bei längeren Listings nicht jede Zeile einrücken, so gibt es in pandoc eine alternative Form, diese ist jedoch nicht Standard-Markdown. Durch eine Zeile mit mindestens 3 ~-Zeichen wird das Programmlisting eingeleitet und durch eine Zeile mit mindestens genauso vielen ~-Zeichen wieder beendet. Eine Zeile mit weniger ~-Zeichen ist Bestandteil des Listings.

```
~~~~~
Dies ist ein Programm Listing
~~~~
Header
~~~~
Body
~~~~~
```

Pandoc 1.9 wird standardmäßig mit Unterstützung für Syntaxhervorhebung übersetzt. Bei älteren Version musste man das explizit aktivieren. Eine Liste der unterstützten Sprachen erhält man mit:

```
pandoc -v
```

Die Sprache teilt man dem Listing mit, außerdem ist es möglich die Zeilen zu nummerieren:

```
~~~~~{.Latex .numberLines startFrom="10"}
\documentclass[11,ngerman]{dtk}
\usepackage[utf8]{inputenc}
~~~~~
```


Mit dem Schalter `--no-highlight` kann man die Hervorhebung abschalten.

Mit dem Schalter `--highlight-style` lassen sich verschiedene Stile auswählen. Zur Auswahl stehen: *pygments* (the default), *kate*, *monochrome*, *espresso*, *zenburn*, *haddock*, und *tango*.

Die Hervorhebung funktioniert nur bei den Ausgabeformaten HTML und LaTeX.

Listen

Aus LaTeX kennen wir verschiedene Listenformen:

Die itemize Liste

Eine Liste wird durch ein Aufzählungszeichen (*, + oder -) eingeleitet.

```
* eins
* zwei
* drei
  - drei a
  - drei b
* vier
  Wie immer verstecken wir die wichtigen Informationen ganz unten,
  in der Hoffnung, das niemand sie liest.
```

Besteht ein Listeneintrag aus mehreren Absätzen, so sind die Folgeabsätze mit 4 Leerzeichen oder einem Tabulatorschritt einzurücken.

```
* eins
* zwei
* drei
  - drei a
  - drei b
* vier

  Wie immer verstecken wir die wichtigen Informationen ganz
  unten, in der Hoffnung, das niemand sie liest.

  Um ganz sicher zu gehen, erwähnen wir erst im letzten Absatz
  die vier Leerzeichen Regel.
```

Die enumerate Liste

Das erste Aufzählungszeichen gibt hier das Format der Aufzählungszeichen vor (1., (1) oder i.). Es ist nicht notwendig, dass die weiteren Aufzählungszeichen in der richtigen Reihenfolge erscheinen (auch wenn das seltsam aussieht).

Es wird automatisch das `enumerate`-Paket geladen. Möchte man das verhindern, so kann man stattdessen die Aufzählungspunkte mit #. markieren, dann wird die Standardeinstellung der verwendeten Dokumentklasse benutzt.

Ab Version 1.10 wird das `enumerate`-Paket nicht mehr benötigt, die Anpassungen erfolgen direkt im TeX-Code.

```
1. ein
2. zwei
```

```

4. drei
    a) drei a
    b) drei b
5. vier
    Wie immer verstecken wir die wichtigen Informationen ganz unten,
    in der Hoffnung, das niemand sie liest.

```

Pandoc kann auch Listen mit römischen Zahlen nummerieren, dabei gibt es jedoch einen Stolperstein: Damit nicht abgekürzte Vornamen die zufällig am Anfang einer Zeile stehen als Aufzählzeichen für eine Liste erkannt werden, ist bei den Zahlen I, V, X, C, D und M ein doppeltes Leerzeichen erforderlich. Bei Kleinbuchstaben ist das nicht notwendig.

```

1. ein
2. zwei
4. drei
    I.  drei a (mit doppeltem Leerzeichen)
    II. drei b
5. vier
    i. vier a (ohne doppeltes Leerzeichen)
    ii. vier b

```

Die description Liste

Endlich kommen wir zu den beliebten `l2kurz`-Tieren. Das beschriebene Objekt steht allein in einer Zeile, die Beschreibung wird durch einen Doppelpunkt oder eine Tilde eingeleitet.

Wenn *hinter* der Beschreibung eine Leerzeile steht, wird sie als Absatz formatiert, also gegebenenfalls mit Einzug und größerem Zeilenabstand. Um eine kompakte Liste zu erreichen, muss man auf Absätze verzichten, bzw. diese müssen als mehrere Beschreibungen definieren werden. Wichtig ist hierbei auch der Abstand hinter der letzten Beschreibung, sonst wird u.U. der letzte Punkt der Liste anders formatiert als die übrigen Punkte.

Beim OpenDocument- oder LaTeX-Export werden beide Fälle gleich behandelt. Ab Version 1.10 wird auch beim LaTeX-Export zwischen kompakten Listen und Listen mit mehr Freiraum unterschieden.

```

Gelse
: ein kleines Tier, das östlich des Semmering Touristen verjagt.
Gemse
: ein großes Tier, das westlich des Semmering von Touristen
  verjagt wird.
: Langer Absatz über die Frage ob es Gemen oder Gämsen heißt.
Gürteltier
~ ein mittelgroßes Tier, das hier nur wegen der Länge seines
  Namens vorkommt.
~ Gürteltiere sind in Österreich außerhalb von Zoologischen
  Gärten selten anzutreffen.
Nächster Absatz

```

Die `l2kurz` Tiere, hier mit etwas mehr Platz:

```

Gelse
: ein kleines Tier, das östlich des Semmering Touristen verjagt.

Gemse
: ein großes Tier, das westlich des Semmering von Touristen

```

```

    verjagt wird.
    : Langer Absatz über die Frage ob es Gensen oder Gämsen heißt.

Gürteltier
~ ein mittelgroßes Tier, das hier nur wegen der Länge seines
  Namens vorkommt.
~ Gürteltiere sind in Österreich außerhalb von Zoologischen
  Gärten selten anzutreffen.

Nächster Absatz

```

Die fortlaufende Liste

Normalerweise fängt jede neue Liste wieder bei 1 an. Es gibt jedoch eine besondere Form, die fortlaufend über das gesamte Dokument läuft, vergleichbar mit den caption-Zählern für **figure** oder **table** Umgebungen.

Auf diese Zähler kann später Bezug genommen werden.

```

(@Behauptung) Hier behaupte ich etwas.

Die Behauptung (@Behauptung) wird in (@Beweis) bewiesen.

(@Beweis) Und hier der versprochene Beweis.

```

Diese Liste benutzt nicht den LaTeX `\label/\ref`-Mechanismus und erfordert daher keinen zweiten LaTeX-Lauf.

Tabellen

Seit Version 1.8.1.2 werden Tabellen mit dem `ctable`-Paket gesetzt. Dadurch hat sich die Qualität der Ausgabe erheblich verbessert.

Mit der Version 1.10 wurde auf `longtable` umgestellt. Dadurch sind Tabellen keine Gleitobjekte mehr sondern erscheinen dort wo sie definiert wurden. Lange Tabellen werden am Seitenende umgebrochen. Leider landen dadurch auch manchmal Teile von kurzen Tabellen auf der nächsten Seite.

Die `longtable` Umgebung ist nicht mit `beamer` kompatibel, nach einer Lösung wird noch gesucht.

Es gibt drei Arten von Tabellen. Bei der Eingabe von Tabellen sollte man auf Tabulatoren verzichten und die Spalten mit Leerzeichen ausrichten.

Rechts	Links	Mitte	Standard
-----	-----	-----	-----
12	12	12	12
123	123	123	123
ab	ab	ab	ab

Table: Eine einfache Tabelle

Der Tabellenkopf und die einzelnen Tabellenzeilen müssen in eine Zeile passen. Die Ausrichtung der Spalten hängt von der Unterstreichung der Kopfzeile ab.

- Ist die Unterstreichung rechtsbündig und steht auf der linken Seite hervor, ist die Ausrichtung rechtsbündig.

- Ist die Unterstreichung linksbündig und steht auf der rechten Seite hervor, ist die Ausrichtung linksbündig.
- Steht die Unterstreichung auf beiden Seite hervor, ist die Ausrichtung zentriert.
- Ist die Unterstreichung genauso lang wie der Text, wird die Standardeinstellung (linksbündig) benutzt.

Eine Tabelle muss mit einer Leerzeile abgeschlossen werden. Optional darf ein Tabellentitel vergeben werden, diese wird durch das Schlüsselwort **Table:** (oder nur **:**) eingeleitet. Wird ein Titel verwendet, so wird die Tabelle als Gleitobjekt in einer **table**-Umgebung formatiert, ansonsten als Tabelle im Fließtext.

Mehrzeilige Tabellen müssen durch jeweils eine Reihe **--**-Zeichen eingeschlossen werden. Die Tabellenzeilen werden durch Leerzeilen getrennt.

Zentrierter Kopf	Standard Kopf	Rechtsbündig	Linksbündig
Erste	Zeile	12.0	Beispiel einer mehrzeiligen Zeile
Zweite	Zeile	5.0	Noch ein mehrzeilige Zeile, durch eine Leerzeile abgetrennt

Beim dritten Typ handelt es sich um sogenannte Rastertabellen. Der äußere Rahmen wird natürlich bei LaTeX-Ausgabe nicht gesetzt. In den Tabellenzellen können z. B. enthalten sein.

Frucht	Preis	Vorteile
Banane	\$1.34	- eingebaute Verpackung - leuchtende Farben
Orange	\$2.10	- heilt Scorbut - ist lecker

Mit Version 1.10 wurde ein vierter Tabellentyp eingeführt, die Pipe-Tabelle. (Pipe ist ein englischer Name für das **|** Zeichen.) Die Ausrichtung der Zellen wird durch den **:** bestimmt. Die Kopfzeile darf leer sein, die Zeile mit den **-** Zeichen jedoch nicht, da sie die Ausrichtung definiert. Die **|** Zeichen in der ersten und letzten Spalte sind optional.

Rechts	Links	Standard	Zentriert
12	12	12	12
123	123	123	123
1	1	1	1

Titelei

Angaben zu Titel des Dokuments, den Autoren und dem Datum der Erstellung werden am Anfang der Datei angegeben:

```
% Viele Ziele Veröffentlichung
% Axel Kielhorn
% DTK 2011-3
```

Lange Titel und mehrere Autoren bricht man auf mehrere Zeilen um:

```
% Viele Ziele Veröffentlichung\
  (Multi Target Publishing)
% Axel Kielhorn
  Babel Fisch (Übers.)
% DTK 2011-3
```

Der \ am Ende der ersten Zeile erzeugt ein \\ in der LaTeX-Ausgabe. Leider führt das bei einigen eBook Lesegeräten zu Problemen, hier sollte man auf das \ verzichten.

Das ePUB Format fordert ein echtes Datum (2012-02-12) als Argument.

```
% Viele Ziele Veröffentlichung
% Axel Kielhorn
% 2012-02-12
```

Fußnoten

Fußnoten bestehen aus zwei Teilen, der Fußnotenmarkierung und dem Fußnotentext.

```
Dies ist eine Fußnotenmarkierung[^1] und dies eine
weitere[^fussnote]

[^1]: Hier ist der Fußnotentext

[^fussnote]: Diese Fußnote ist etwas länger.

    Sie enthält einen zweiten Absatz.
```

Hervorhebungen

Kursiver Text wird durch *ein* * oder _ eingeschlossen.

Fetter Text wird durch *zwei* * oder _ eingeschlossen.

Sollen nur Wortteile hervorgehoben werden, ist zwingend der * zu nutzen, da in Variablenamen häufig _ als Bestandteil des Namens vorkommen.

```
Dieser Text wurde _mit dem Unterstrich hervorgehoben_
und dieser *mit dem Sternchen*.

Für __fetten Text__ benutzt man **zwei** Zeichen.

Variablen_namen_ können Unterstriche enthalten, daher benutzt man
zum hervor*heben* das Sternchen.
```

Zum Hochstellen benutzt man das bekannte ^, zum Tiefstellen muss auf die ~ ausgewichen werden, da der Unterstrich ja schon zum auszeichnen verwendet wird. Die Befehle schließen das Argument ein.

```
H2O ist Wasser, 210 ist 1024.  
222 ist 222.
```

Die letzte Zeile mag zuerst verwundern, allerdings handelt es sich hier nicht um Mathematiksatz sondern um Textexponenten und -indices.

Mathematik

Inline-Mathematik wird wie aus LaTeX bekannt in $\$$ eingeschlossen. Der Inhalt wird direkt an LaTeX weitergeben, daher ist alles erlaubt, was in LaTeX möglich ist.

```

$$2^{2^2} \neq 2^{22}$$

```

Bei anderen Ausgabeformaten hängt es von jeweiligen Format und den Optionen beim Aufruf von pandoc ab, was als Ausgabe erzeugt wird.

Abgesetzte Formeln können als rohes LaTeX eingegeben werden.

Rohes LaTeX

Alles was zwischen einem `\begin/\end` Paar steht wird direkt an LaTeX (oder ConTeXt) weitergegeben und in allen anderen Formaten ignoriert.

Rohes HTML

Da Markdown ursprünglich zum Erstellen von HTML-Seiten entwickelt wurde, bietet es die Möglichkeit HTML direkt einzugeben. Bei nicht HTML basierten Ausgabeformaten wird es ignoriert.

Links

Bei einem Format, das ursprünglich zum Erstellen von Webseiten gedacht war, sollte man annehmen, das es mit Hyperlinks umgehen kann. Und so ist es natürlich auch. Alles, was in einer Zeile zwischen spitzen Klammern steht, ist ein Link.

```
<http://johnmacfarlane.net/pandoc/>
```

Ein Link kann natürlich auch im Text auftauchen:

```
Die Dokumentation zu pandoc befindet sich auf der  
[pandoc Webseite](http://johnmacfarlane.net/pandoc/).
```

Bilder

Bilder sind eine besondere Form des Links. Wenn vor dem Link ein `!` steht, wird dieser als Link auf ein Bild interpretiert.

```
![Ein Blaues Bild](blau.jpg "Blaues Bild")
```

Das Bild wird in einer **figure**-Umgebung gesetzt, der Text in den eckigen Klammern wird als Bildunterschrift verwendet.

Möchte man das Bild im Fließtext einbinden, darf der Link nicht allein in einer Zeile stehen.

```
Das ![Rotes Quadrat](rot.png "rote Quadrat") im Fließtext.
```

Es gibt keine Möglichkeit, die Bilder zu skalieren, sie müssen in der richtigen Größe und Auflösung vorliegen.

Am Wegesrand

Pandoc kann auch ConTeXt-Dateien erstellen. Damit könnte man vorhandene LaTeX-Dateien konvertieren, um den Einstieg in ConTeXt zu finden.

ConTeXt kann mithilfe des Filter Moduls sogar direkt Markdown bearbeiten indem es pandoc als Filter aufruft. Näheres hierzu gibt es im Pandoc Extra Wiki [\[6\]](#).

Reiseliteratur: Große Dokumente ...

Standardmäßig erstellt pandoc Dokumente ohne Abschnittsnummerierung und ohne Inhaltsverzeichnis. Bei kurzen Texten ist das sicher kein Problem, bei längere Texten muss man nicht darauf verzichten. Mit:

```
pandoc -f markdown -t latex --number-sections
-o quelle.tex quelle.md
```

erhält man die Abschnittsnummern und mit:

```
pandoc -f markdown -t latex --toc -o quelle.tex quelle.md
```

das Inhaltsverzeichnis. Beides kann man natürlich auch kombinieren.

Die Tiefe des Inhaltsverzeichnisses kann man ab Version 1.10 beeinflussen:

```
pandoc -f markdown -t latex --toc-depth=3 -o quelle.tex quelle.md
```

Die Option `--toc` wird dabei implizit gesetzt, muss also nicht angegeben werden. Das funktioniert auch bei HTML und epub.

... oder ganze Bücher

Wenn keine Dokumentklasse angegeben wird, benutzt pandoc standardmäßig die Klasse **article**, bzw. wenn die Option `--chapters` gewählt wurde, die Klasse **book**.

Durch das setzen einer Variablen, kann man die gewünschte Dokumentklasse auswählen, dabei werden **report**, **book**, **memoir**, **scrreprt** und **scrbook** automatisch erkannt und die Option `--chapters` gesetzt.

Bei älteren pandoc Versionen (<1.9) war dies nicht möglich. Pandoc hat hier die Dokumentklasse aus der Template Datei gelesen und bei **report**, **book** und **memoir** auf Kapitel als oberste Gliederungsebene umgeschaltet.

Bei **scrreprt** und **scrbook** musste man Kapitel durch eine Option aktivieren:

```
# pandoc >=1.9
pandoc -f markdown -t latex -V documentclass=srcreprt -s
-o quelle.tex quelle.md

# pandoc <1.9
pandoc -f markdown -t latex --template=./komascript.latex
--chapters -o quelle.tex quelle.md
```

Die Datei `monster.latex` ist ein Versuch möglichst viele konfigurierbare Optionen in eine Vorlage zu bringen. Durch die Vielzahl an Optionen ist diese Datei nur noch über Skripte sinnvoll nutzbar, da kaum jemand alle Optionen jedes mal von Hand eintippen möchte.

Der Quelltext kann in mehrere Dateien aufgeteilt werden, allerdings bietet pandoc keinen mit `\include/\includeonly` vergleichbaren Mechanismus. Stattdessen hängt man die Dateinamen einfach an den Befehlsaufruf an:

```
pandoc -f markdown -t latex --number-sections --toc
--template=./report.latex -o md-test.tex
md-test-intro.md
md-test-ch1.md
md-test-ch2.md
md-test-ch3.md
```

Kein Ziel ohne Quellen

Verweise ins Literaturverzeichnis werden in eckige Klammern gesetzt und erhalten das `@`-Zeichen sowie den Zitierschlüssel. Es können mehrere Quelle in einem Befehl zitiert werden. Kann ein Verweis nicht aufgelöst werden, so bleibt der Zitierschlüssel im Text, es gibt keine Fehlermeldung!

```
<http://johnmacfarlane.net/pandoc/>[@pandoc]
Dieser Artikel erschien in [@Ziele-dtk; @Ziele-TUGboat; @Ziele-ct]
```

Pandoc kann `bibTeX`-Dateien lesen und mit Hilfe von CSL (Citation Style Language) formatieren. Stile für verschiedene Zeitungen und Organisationen stehen unter [1] zur Verfügung. Ohne explizite Angabe wird ein Chicago Autor–Datum Stil verwendet.

Das Literaturverzeichnis wird an das Dokument angehängt, dabei wird die letzte Überschrift als Titel des Verzeichnisses benutzt. Sollte es nach dieser Überschrift noch weiteren Text geben, so wird er vor dem eigentlichen Verzeichnis ausgegeben, z. B. für eine Einleitung zum Literaturverzeichnis.

```
pandoc -f markdown -t latex --bibliography quelle.bib --cs1 apa.cs1
-o quelle.tex quelle.md
```

Diese Methode hat den Vorteil, dass sie mit allen Ausgabeformaten funktioniert und keinen weiteren TeX-Lauf benötigt. Sie hat jedoch den Nachteil, dass im erzeugten PDF keine Links auf das Literaturverzeichnis verweisen. Außerdem gibt es Probleme mit LaTeX-Befehlen in der `bib`-Datei, das TeX in TeXnische Komödie wird falsch wiedergegeben.

Bei der Ausgabe einer TeX-Datei kann man alternativ auch `biblatex` verwenden. In Verbindung mit `hyperref` entstehen so Links in das Literaturverzeichnis. Die Sortierung erfolgt dann mit `bibtex` oder `biber`, die Formatierung über entsprechende `biblatex` Stile. Es sind zusätzliche LaTeX Läufe erforderlich.

```
pandoc -f markdown -t latex --bibliography quelle.bib --biblatex
-o quelle.tex quelle.md
```


Fazit

Anders als bei LaTeX ist die Einarbeitung in Markdown sehr einfach. Solange die Standardmöglichkeiten ausreichen bietet Markdown als Eingabeformat nur Vorteile: weniger Tipparbeit, übersichtlicherer Text. Trotzdem befindet sich im Hintergrund ein komplettes LaTeX System, auf das jederzeit zurückgreifen kann. Zum Anpassen der Template Datei sind LaTeX Kenntnisse erforderlich und man sollte auch im Betrieb einen LaTeX Experten zur Hand haben, falls es mal hakt. Aber es hakt deutlich weniger, jedenfalls solange man auf rohes LaTeX verzichtet.

Reichen irgendwann die Fähigkeiten von Pandoc nicht mehr aus, hat man ja immer noch eine LaTeX Datei als Zwischenprodukt. Diese kann dann z. B. mit `\index` Befehlen erweitert werden.

Die Möglichkeit neben PDF auch andere Formate zu erstellen ist ein deutlicher Gewinn. Nun kann aus einer Quelle ein pBuch, ein eBuch und eine Webseite erstellt werden.

Was fehlt ist ein Editor der Pandoc per Knopfdruck aufruft und somit den Ausflug in die Kommandozeile erspart. Natürlich ist eine Integration in Vim oder Emacs kein Problem, aber das sind nicht die Editoren, die ein Einsteiger nutzen wird. Für TeXshop gibt es bereits engines für die wichtigsten Aufgaben (Zielformate PDF und ePUB), bei anderen Editoren sieht es noch schlecht aus.

Lizenz

Dieser Artikel und das Begleitmaterial unterliegen eine Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen Lizenz der Version 3.0. Weitere Informationen unter <http://creativecommons.org/licenses/by-sa/3.0/de/>

Anhang

Im Begleitmaterial befinden sich folgende Dateien:

md-test.md Die Beispiele aus diesem Artikel

md-test.bib Eine bibtex Datei für das Literaturverzeichnis.

md-test.tex Konvertiert nach LaTeX.

md-test.pdf Mit pdfLaTeX gesetzt.

md-test.epub Konvertiert nach ePUB

default-de.latex, report-de.latex, komascript.latex, monster.latex Ein paar LaTeX-Vorlagen zum Erstellen deutscher Texte.

reference-de.odt Eine OpenOffice-Vorlage zum Erstellen deutscher Texte.

dtk2mdtex.sed Eine sed Datei zum Entfernen der dtk-spezifischen Auszeichnung.

pandoc.pdf Die pandoc-Manpage.

pandoc-markdown.pdf Die Manpage, die den Markdown Syntax von pandoc beschreibt.

Kommandozeilen.txt Eine Sammlung von Kommandozeilen, um Tipparbeit zu sparen.

Literatur

- [1] Bruce D’Arcus, Frank G. Bennett Jr. und Rintze M. Zelle. *Citation style language: Styles*. <https://github.com/citation-style-language/styles>. 2011.
- [2] Kovid Goyal. *Calibre – E-book management*. <http://calibre-ebook.com/>. 2011.
- [3] John Gruber. *Daring Fireball: Markdown*. <http://daringfireball.net/projects/markdown/>. 2004.
- [4] Axel Kielhorn. *Ziele.zip*. <https://github.com/AKielhorn/Markdown-Intro/blob/master/Ziele.zip>. 2011.
- [5] John MacFarlane. *Pandoc – About pandoc*. <http://johnmacfarlane.net/pandoc/>. 2011.
- [6] John MacFarlane. *Pandoc Extras*. <http://github.com/jgm/pandoc/wiki/Pandoc-Extras>. 2011.
- [7] Strahinja Marković. *Sigil – A WYSIWYG ebook editor*. <http://code.google.com/p/sigil/>. 2011.