

Artikel und Texte mit Markdown schreiben

Justus Holzberger

7. November 2018, CC-By 4.0, v. 1.4

Inhaltsverzeichnis

1	Einleitung	1
1.1	Markdown? Hä?	1
1.1.1	Markdown-Basics	2
1.1.2	Bilder und Tabellen	3
1.1.3	Querverweise	4
2	Workflow mit Markdown	4
2.1	Ein Texteditor	5
2.2	Umwandeln mit pandoc	5
3	Ansprüche an wissenschaftliche Texte	5
3.1	Titelseite(n)	6
3.1.1	Ganz simpel - stumpf ist Trumpf!	6
3.1.2	Simpel aber mit ohne Seitennummern	6
3.1.3	Fancy Ferzeichnisse	7
3.2	Schriftart und Seitenformatierung	8
3.3	Quellenangaben	9
3.3.1	Zitierstile, Literaturverzeichnis	10
4	Anleitung für Eilige	11
5	Fazit und noch mehr Möglichkeiten	12
6	Links auf einen Blick	12

1 Einleitung

Vor einiger Zeit bin ich auf einige [Video-Tutorials von Nicholas Cifuentes-Goodbody](#) gestoßen. Er erklärt darin, wie er wissenschaftliche Artikel in Markdown schreibt und sie mit pandoc in formatierte RTF-Dateien (Rich Text Format) umwandelt. Ich fand das recht elegant, zumal ich schon seit längerer Zeit aus Gründen immer mal wieder mit Markdown liebäugle. Außerdem habe ich schon einige Erfahrungen mit [pandoc](#) sammeln können und bin *sehr* angetan von den vielen Möglichkeiten.

Und so begab es sich, dass ich mich fünf Stunden am Stück mit Markdown-Formatierungen wissenschaftlicher Texte auseinandergesetzt habe... Damit die Mühe nicht umsonst war und alle davon profitieren können, möchte ich das in diesem Text festhalten.

Ich werde kurz Erklären was Markdown ist, einige nützliche Befehle auflisten, (m)einen Workflow beschreiben und zeigen, wie ich meinen Text mit pandoc in eine PDF umwandle.

Als Kurzversion zum gucken empfiehlt sich das oben genannte [Video!](#)

Alle Schritte in sehr komprimierter Form gibt es unter [Anleitung für Eilige](#).

1.1 Markdown? Hä?

[Markdown](#) ist eine Auszeichnungssprache für Texte und folgt damit einem sogenannten [WYSIWYM](#)-Ansatz (What You See Is What You Mean).

Schreibt man einen Text in Markdown, markiert man mit Sonderzeichen und Einrückungen im Text, wie diese so *ausgezeichneten* Stellen später einmal aussehen sollen. Ein Text wie

```
# eine Überschrift
```

wird zu einer Überschrift 1, mit zwei ## zu einer Überschrift 2 und so weiter. Möchte man ein Wort (oder mehr) kursiv hervorheben, schreibt man einfach Sternchen (*) davor und dahinter.

So ein Vorgehen hat einige Vorteile:

- einen einfacher Texteditor genügt zum Schreiben - den findet man überall
- Schreibprozess und Layouting sind voneinander getrennt - das hilft dabei, konzentriert zu schreiben
- die Befehle bzw. Syntax ist einfach zu lernen
- Markdown ist auch als Quelltext gut lesbar
- es lässt sich in (fast) alle anderen Dokumentenformate konvertieren
- die Dateien verbrauchen wenig Speicherplatz und sind robust, da sie ja nur Text enthalten und kein umfangreiches Dateiformat

Mehr zu der Philosophie und den Befehlen gibt es auf der Webseite von [Markdown](#), wo alle Formatierungen einzeln aufgeführt sind.

Es gibt unterschiedliche Varianten von Markdown, die sich meist in Anwendungsgebiet und Syntax geringfügig unterscheiden. Neben dem hier vorgestellten Pandoc-Markdown sind weitere verbreitete Varianten: *GitHub-flavored Markdown (GFM)* und *R-Markdown*.

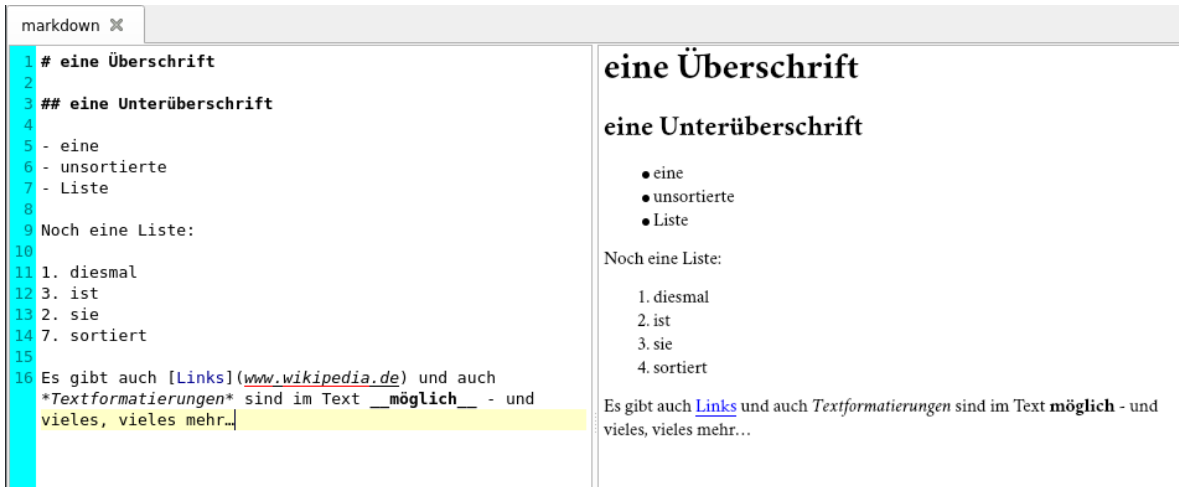


Abbildung 1: Beispiel der Markdown-Syntax: Links Markdown, rechts der formatierte Text (HTML).

1.1.1 Markdown-Basics

Ein paar dieser Markdown-Befehle für alltägliche Textarbeit sind:

- **unnummerierte Listen** werden wahlweise mit * oder - und einem Leerzeichen dahinter erstellt.
- **nummerierte Listen** werden mit 1., 2. etc. und einem Leerzeichen dahinter erstellt. Die Zahlen müssen dabei nicht in der richtigen Reihenfolge stehen, das passiert automatisch.

- **Links** sehen so aus:

[Linkname](https://www.beispielseite.de)

oder einfach:

<https://www.beispielseite.de>

- **Bilder** so:

![Textbeschreibung](beispiel.png)

oder so:

![Textbeschreibung](URL)

- **Tabellen** macht man so:

```
| **Name** | **Wert** |
| :---: | :---: |
| Gold | 100 |
| Blei | 10 |
```

- **Fußnoten** einfach im Text mit Eckigen Klammern und vorangestelltem Circumflex:

^[Text der Fußnote]

- **Zitatblöcke** werden mit > vor dem jeweiligen Absatz eingefügt

- Drei Arten von **Zitaten** im Text (mittels BibTeX-Datei, s. **Quellenangaben**):

- [@quellen_key] wird zu: (Autor_in Jahr)
- [-@quellen_key, S. 30] wird zu: (Jahr, S. 30)
- @quellen_key [S. 30] wird zu: Autor_in (Jahr, S. 30)

- **Formeln** sind dank LaTeX kein Problem:

$$D = -2 \sum \lim_{i \rightarrow 1} \ln(n_i/n) * n_i$$

wird zu:

$$D = -2 \sum_{i=1}^I \ln(n_i/n) * n_i$$

- **Code** kann mittels “backticks”, also ` ` eingefügt werden oder durch Einrücken mit einem Tabstopp.

1.1.2 Bilder und Tabellen

Bilder und Tabellen werden immer den Seitenrändern entsprechend mittig im Text platziert. Sie sind nur bei Angabe einer Beschreibung Beschriftet.

Tabelle 1: Eine Beispieldabelle

Name	Wert
Gold	100
Blei	10

Tipp: Einige Markdown-Editoren können Tabellen automatisch anlegen oder entsprechend einrücken, was eine enorme Erleichterung ist. Für alle, die dazu eine grafische Oberfläche brauchen, gibt es z.B. auch Online-Dienste.

1.1.3 Querverweise

Querverweise lassen sich `-pandoc-crossref` gut bewerkstelligen. Sie funktionieren dann ähnlich die die Referenzen in LaTeX mit Markern wie `{#fig:label}`. Eine Anleitung dazu gibt es auf der [Projektseite von pandoc-crossref](#).

Label müssen immer eine eindeutige Bezeichnung haben!

Ein Bild sieht dann im Text folgendermaßen aus:

```
![Beschriftung](DATEI.xxx){#fig:Label}
```

Soll irgendwo auf das Bild verwiesen werden, genügt etwas wie:

```
Verweis auf eine Abbildung (s. [ @fig:Label ])
```

Dieser Teil wird später ersetzt durch das entsprechende Kürzel des Objekts (`fig:` für eine Abbildung, `tbl:` für eine Tabelle, `eq:` für mathematische Gleichungen, `sec:` für Überschriften etc.).

Die entsprechenden [Kürzel sind in der Dokumentation beschrieben](#). Sie können auch angepasst werden, um etwa aus "Figure" ein "Foto" o.ä. zu machen. Das geschieht über die bereits genannten [YAML-Header](#).

2 Workflow mit Markdown

Anmerkung: ich benutze [Linux](#) und kann die vorgestellte Software daher über mein Paketmanagement installieren. Unter Windows werden diese Programme anders installiert und benutzt werden. Einige sind eventuell auch gar nicht unter Windows oder MacOS verfügbar.

Voraussetzung zum Schreiben und Umwandeln von Markdown sind grundsätzlich:

- ein Texteditor
- [pandoc](#) zum Umwandeln
- eine LaTeX-Umgebung für den eigentlichen Satz. (Hier: [TeX Live](#), [pandoc](#) empfiehlt [MiKTeX](#))

Der Ablauf ist ebenfalls denkbar einfach:

1. Text schreiben und dabei die Markdown-Befehle nutzen
 - dabei können auch LaTeX-Befehle benutzt werden
2. `pandoc` mit den gewünschten Optionen aufrufen und den geschriebenen Text in ein (PDF-)Dokument übersetzen
3. sich über die wunderschöne PDF freuen

2.1 Ein Texteditor

Ich verwende gerade den Editor [Gedit](#) vom [GNOME-Projekt](#) [Vim](#) zum Schreiben. Markdown kann hier als Hervorhebungsmodus eingestellt werden, so dass die verschiedenen Text-Teile verschiedenfarbig hervorgehoben werden.

Außerdem gibt es für Gedit Vim Plugins für “Schnipsel” ([UltiSnips](#)). Mit Schnipseln lassen sich bestimmte Text-Bausteine einrichten, die man mit einem kurzen Schlagwort und der Autovervollständigung (Taste: *TAB*) einfügen kann. Das erleichtert die Arbeit enorm, da die Befehle mit mehreren Klammern etc. nicht mehr ausgeschrieben werden müssen.

Anmerkung: Es gibt unzählige Editoren, die unterschiedlichste Funktionen bieten. Aus meiner Sicht sind dabei folgende Möglichkeiten sehr sinnvoll:

- *Syntax-Highlighting*
- *Wort- bzw. Code-Vervollständigung und (Snippets)*
- *Rechtschreibprüfung*
- *“Einklappen” von Codeblöcken bzw. Kapiteln / Überschriften (gerade bei langen Dokumenten)*
- *Lesezeichen bzw. Übersichten über Kapitel / Überschriften*

2.2 Umwandeln mit pandoc

Ist der Text getippt, wandelt ihn mir [pandoc](#) in eine PDF um. Es greift dabei auf das Satzsetzprogramm [LaTeX](#) zurück, das sehr guten Satzsetz ermöglicht. So entsteht eine wundervolle PDF, die einem gedruckten Artikel gleichkommt.

Um aus einer Markdown-Datei eine hübsche PDF zu zaubern, braucht pandoc eigentlich nicht viele Einstellungen:

```
pandoc INPUT.md -o OUTPUT.pdf
```

Damit wird pandoc aufgerufen und nimmt meine INPUT-Datei (hier `.md`, kann aber auch einfach `.txt`), um sie auszugeben als Output (`-o`) - und zwar in die Datei `OUTPUT.pdf`. Das Dateiformat wird dabei automatisch durch die Dateiendung erkannt - es könnte also auch `OUTPUT.html` oder `OUTPUT.ODT` heißen.

3 Ansprüche an wissenschaftliche Texte

Wissenschaftliche Texte brauchen aber meist weitere Funktionen und Text-Teile, die sie von anderen Texten unterscheiden:

- Titelblatt
- Abstract

- Quellenangaben im Text
- Fußnoten
- Zitat-Blöcke
- (beschriftete) Bilder
- (beschriftete) Tabellen
- Literaturverzeichnis
- Abbildungsverzeichnis
- Seitennummerierungen

Je nach eigenem Geschmack, Format oder Verlag wird auch eine andere Schriftart vorausgesetzt. Meine persönliche Favoritin ist dabei die *Linux Libertine*, die etwas “runder” als die *Times New Roman* ist.

Im Folgenden möchte ich zeigen, welche Gestaltungsmöglichkeiten pandoc dabei bieten kann.

3.1 Titelseite(n)

Anmerkung: nicht alles hier lässt sich einfach mit purem Markdown umsetzen. Einige der Befehle sind möglicherweise spezifisch für die Markdown-Interpretation von pandoc und / oder LaTeX.

Die weiteren Punkte beziehen sich auf das extrem empfehlenswerte(!) [pandoc Manual](#).

3.1.1 Ganz simpel - stumpf ist Trumpf!

Sehr einfach lässt sich eine Titelseite generieren, indem die ersten drei Zeilen des Dokuments mit einem % begonnen werden (s. Pandocs [Metadata Blocks](#)).

- Alles hinter dem *ersten* % wird zum **Titel**
- Alles hinter dem *zweiten* % wird zum **Autor_innennamen**
- Alles hinter dem *dritten* % wird zur **Datumsangabe**

3.1.2 Simpel aber mit ohne Seitennummern

Erweitert man diese drei Zeilen mit ein wenig LaTeX-Code, erhält man eine nicht nummerierte Titelseite:

```
% Titel
% Autor_in
% Datum
\pagenumbering{gobble}
\pagebreak
\pagenumbering{arabic}
```

Der LaTeX-Befehl `\pagenumbering{gobble}` “verschluckt” dabei die Seitenzahlen so lange, bis sie mit `\pagenumbering{arabic}` aktiviert wird - hier nach einem Seitenumbruch (`\pagebreak`).

Anmerkung: Mit diesem Block am Anfang kann man auch pandoc direkt den Parameter `--toc` mitgeben. Damit wird automatisch ein Inhaltsverzeichnis auf der ersten Seite erstellt.

3.1.3 Fancy Ferzeichnisse

Da manchmal auch ein Abstract des Artikels gut wäre, nun eine umfangreichere Lösung. Sie enthält die Angaben zu *Titel*, *Autor_in*, *Datum*, ein *Abstract*, sowie ein *Inhaltsverzeichnis* und ein *Abbildungsverzeichnis*. Diese ersten Seiten sind jeweils nicht nummeriert. Die Nummerierung startet erst ab der dritten Seite.

Achtung! Die Verzeichnisse werden bei dieser Lösung *mit LaTeX-Befehlen* (`\tableofcontents` und `\listoffigures`) *im Dokument* erstellt, die von pandoc beim Umwandeln interpretiert werden.

Es gibt drei Varianten:

1. Die Verzeichnisse mit LaTeX-Befehlen in die Datei schreiben (wie gerade beschrieben)
2. pandoc beim Umwandeln die Option `--toc` mitgeben, um ein Inhaltsverzeichnis zu erstellen
3. pandoc beim Umwandeln (nur zu einer PDF!) die Optionen (LaTeX-Variablen) `-v toc` (Inhaltsverz.), `-v lof` (Abbildungsverz.) und `-v lot` (Tabellenverz.) mitgeben ¹

Meta-Angaben des Dokuments lassen sich bei pandoc als **YAML-Header** (s. unten, alles zwischen `---` und `...`) angeben. Damit sind Schlagwörter und ein Abstract am Anfang des Dokuments möglich. Außerdem werden durch die Option `link-citations: true` im Dokument die Zitate später in anklickbare Links umgewandelt.

Die “Rahmenbedingungen” so einer Datei können dann dann so aussehen:

```
---
title: 'Titel der Arbeit'
author:
- Autor_in
date: \today
tags: [schreiben, markdown, text]
abstract: |
```

¹`--toc` und `-v toc` sind das Selbe

Text des Abstracts hier.

```
link-citations: true
```

```
...
```

```
\pagenumbering{gobble}
```

```
\pagebreak
```

```
\tableofcontents
```

```
\pagebreak
```

```
\listoffigures
```

```
\pagebreak
```

```
\pagenumbering{arabic}
```

... Hier steht das der ganze Textkram ...

```
# Literaturverzeichnis
```

```
\footnotesize
```

Alles zwischen --- und ... ist dabei der besagte YAML-Header. Es folgen einige Zeilen LaTeX-Befehle:

- `\pagenumbering{gobble}` “verschluckt” die Seitenzahlen, bis sie mit `{arabic}` weiter unten aktiviert werden
- die ersten Seiten mit Inhaltsverzeichnis (`\tableofcontents`) und Abbildungsverzeichnis (`\listoffigures`) sind daher nicht nummeriert
- zwischen den Verzeichnissen befinden sich jeweils feste Seitenumbrüche (`\pagebreak`)

Das Literaturverzeichnis am Ende wird von pandoc erstellt (dazu gleich mehr) und automatisch am Ende des Dokuments eingefügt. Der LaTeX-Befehl `\footnotesize` stellt dabei für allen Text nach dem Befehl die Schriftgröße auf die Größe der Fußnoten ein.

3.2 Schriftart und Seitenformatierung

Um Überschriften zu nummerieren, eine andere Schriftart zu wählen und die Seiten des Textes zu formatieren (Rand, Schriftgröße etc.), müssen pandoc einige zusätzliche Optionen mitgegeben werden:

- `--number-sections` : aktiviert die Nummerierung vor den jeweiligen Überschriften (1., 1.1, 1.2 usw.)
- `-V papersize=a4paper` : Seitengröße ist A4-Papier
- `-V geometry:margin=3cm` : 3cm Seitenrand überall
- `-V lang=de-DE` : Sprache ist Deutsch - damit das automatische “Inhaltsverzeichnis” nicht “Content” heißt...
 - `--pdf-engine=lualatex` : Um die folgenden Schriften einzustellen ist eine andere PDF-Engine nötig als die voreingestellte `pdflatex`
 - * `-V fontfamily=libertine` : Schriftart ist meine geliebte *Linux Libertine*

* -V monofont=inconsolata : Schriftart für Code bzw. Schreibmaschinen-Schrift ist die *Inconsolata*

- -V fontsize=12pt : Schriftgröße ist 12pt
- -V breakurl : zu lange URLs im Text werden umgebrochen
- -V hyphens=URL : URLs werden korrekt getrennt
- -V colorlinks : Links werden im Text farbig hervorgehoben (farbige Schrift)

Ein Aufruf von pandoc wird also für diese Einstellungen und obige “Rahmenbedingungen” etwas länger:

```
pandoc INPUT.md --number-sections --pdf-engine=lualatex -V papersize=a4paper -V geometry:margin=3cm -V lang=de-DE -V fontfamily=libertine -V monofont=inconsolata -V fontsize=12pt -V breakurl -V hyphens=URL -V colorlinks -o OUTPUT.pdf
```

3.3 Quellenangaben

Für wissenschaftliche Literatur sind Quellenangaben im Text und ein Literaturverzeichnis ein unbedingtes Muss!

Ich arbeite mit der Literaturverwaltung [Zotero](#). Die darin erfassten Quellen exportiere ich als BibTeX-Datei (dazu bitte in der [Zotero Hilfe](#) nachsehen). Es gibt ein wundervolles Plug-In für Zotero, das die BibTeX-Datei synchron hält und stetig aktualisiert - es heißt [Better Bib\(La\)TeX](#).

BibTeX ist das Format, in dem LaTeX Quellenangaben verwaltet. Dazu stehen alle Quellen mit ihren Daten in einer .bib-Datei. Die Einträge in dieser Datei lassen sich dann in einer Markdown-Datei zitieren. Jede Quelle hat darin einen *Key* (meist als: Name_Datum), der sie eindeutig identifiziert.

Es wird in diesem Beispiel angenommen, dass die BibTeX-Datei (quellen.bib) im gleichen Verzeichnis wie die Markdown-Datei liegt.

Auf folgende Arten lassen sich dann [Quellen aufrufen](#):

- [@quellen_key] wird zu: (Autor_in Jahr)
- [-@quellen_key, S. 30] wird zu: (Jahr, S. 30)
- @quellen_key [S. 30] wird zu: Autor_in (Jahr, S. 30)

Um das Literaturverzeichnis am Ende zu erstellen, braucht pandoc noch die Informationen, wie:

- die quellen.bib mit den Angaben heißt und wo sie liegt (hier: “quellen.bib” im gleichen Verzeichnis)

Für die Verarbeitung von Literaturangaben braucht Pandoc ein zusätzliches Modul: [pandoc-citeproc](#).

Dann bekommt pandoc als Optionen mit:

- `--filter pandoc-citeproc` : ([Citeproc](#) ist Voraussetzung für das Arbeiten mit BibTeX-Zitationen)
- `--bibliography quellen.bib` : Bibliographische Informationen liegen in der Datei `quellen.bib`

Achtung: werden sowohl `pandoc-citeproc` als auch `pandoc-crossref` benutzt, muss `pandoc-citeproc` vor `pandoc-crossref` aufgerufen werden: `pandoc -F pandoc-crossref -F pandoc-citeproc file.md -o file.html`

3.3.1 Zitierstile, Literaturverzeichnis

Je nach Uni und Fachbereich sind unterschiedliche Stile für Zitate und Literaturverzeichnisse üblich. Literaturverwaltungen wie Zotero bieten tausende Stile über CSL-Dateien (Citation Style Language). Diese können vom [Zotero Style repository](#) heruntergeladen werden. Diese legt man dann ins Verzeichnis mit der Markdown-Datei und gibt folgende Optionen für pandoc mit an (hier die *Kölner Zeitschrift für Soziologie und Sozialpsychologie*):

- `--csl kolner-zeitschrift-fur-soziologie-und-sozialpsychologie.csl`

4 Anleitung für Eilige

Der Workflow von [Markdown](#) und [pandoc](#) hier noch einmal “für Eilige” skizziert:

1. Schnapp dir den Texteditor deiner Wahl

2. Schreibe in Markdown mit einigen LaTeX-Befehlen. Hier ein Beispielgerüst:

```
---
title: 'Titel der Arbeit'
author:
- Autor_in 1
- Autor_in 2
date: \today
tags: [schreiben, markdown, text]
abstract: |
  Text des Abstracts hier.
link-citations: true
...
\pagenumbering{gobble}
\pagebreak
\tableofcontents
\pagebreak
\listoffigures
\pagebreak
\pagenumbering{arabic}
... Hier steht das der ganze Textkram ...
# Literaturverzeichnis
\footnotesize
```

3. Wandle die Datei mit pandoc um:

```
pandoc INPUT.md --number-sections --pdf-engine=lualatex -V papersize=a4paper -
V geometry:margin=3cm -V lang=de-DE -V fontsize=12pt -V breakurl -V hyphens=URL
-V colorlinks -o OUTPUT.pdf
```

1. für eine andere Schriftart (z.B.) zusätzlich:

- -V fontfamily=libertine
- -V monofont=inconsolata

2. für Quellenangaben und Literaturverzeichnis im Stil der *Kölner Zeitschrift für Soziologie und Sozialpsychologie* zusätzlich:

- --filter pandoc-citeproc
- --bibliography quellen.bib

- --csl kolner-zeitschrift-fur-soziologie-und-sozialpsychologie.csl

5 Fazit und noch mehr Möglichkeiten

Wer mit *wenig Aufwand* einfach *schöne Dokumente* produzieren möchte, sollte sich Markdown und pandoc einmal genauer ansehen! *Unterschiedliche Ausgabeformate* sind damit kein Problem - genau so wenig, wie die anspruchsvollen Anforderungen *wissenschaftlicher Texte*.

Hausarbeiten, Paper, Mitschriften, Notizen oder Exzerpte sind mit Markdown schnell geschrieben und *an kein Dateiformat oder Schreibprogramm gebunden*. Mit pandoc können schnell PDF-Dokumente erstellt werden, genau wie Dateien gängiger Office-Programme, Webseiten, E-Books oder sogar interaktive Präsentationsfolien.

Mit dem gezeigten Workflow sind *professionell aussehende Ergebnisse* möglich, ohne sich zu sehr um Formatierungen des eigenen Textes kümmern zu müssen.

Wer sich eine solche Vorlage für Texte anlegt, kann einfach mit einem Texteditor der Wahl in die Tasten hauen. Dabei lohnt es sich, den Befehl zum Umwandeln mit pandoc als einen *benutzerdefinierten Befehl* anzulegen: so kann *jeder beliebige Markdown-Text mit einem Klick in ein fertiges Dokument umgewandelt* werden.

6 Links auf einen Blick

Academic Writing in Markdown (en) <https://www.youtube.com/watch?v=hpAJMSS8pvs>

pandoc (en) <http://pandoc.org/>

pandoc Dokumentation (en) <http://pandoc.org/MANUAL>

Markdown (en) <https://daringfireball.net/projects/markdown/>

Wikipedia zu LaTeX (de) <https://de.wikipedia.org/wiki/LaTeX>

LaTeX Webseite (en) <https://www.latex-project.org/>

Zotero (en/de) <https://www.zotero.org/>