Literate Programming

Dokumente erstellen und konvertieren mit pandoc

Meik Teßmer*

2014

Bereich *Computergestützte Methoden* (CoMet), Fakultät für Wirtschaftswissenschaften, Universität Bielefeld

Version: 2

Inhaltsverzeichnis

*mtessmer@wiwi.uni-bielefeld.de

1 Was ist pandoc?

pandoc bezeichnet sich selbst als „swiss army-knife“ für die Konvertierung von Dokumenten, die mit Hilfe sog. Auszeichnungssprachen erstellt wurden. Das Werkzeug unterstützt neben den gängigen Markup-Sprachen wie HTML und LaTeX auch Markdown, reStructuredText, MediaWiki uvm. Dieser Beitrag gibt einen kurzen Überblick über die Verwendung und das von pandoc unterstützte Markup.

2 Installation

Die meisten GNU/Linux-Distributionen bieten relativ alte Versionen von pandoc an. Glücklicherweise kann eine lokale Installation vergleichsweise einfach vorgenommen werden. Auf Debian-basierten Systemen ist dafür zunächst die Installation einiger Paketen notwendig:

```
aptitude install libz-dev ghc alex happy cabal-install
```

Anschließend kann mit Hilfe von cabal die Installation beginnen:

```
cabal update
cabal install pandoc pandoc-citeproc
```

Die Installation ist aufgrund von Paketabhängigkeiten umfangreich und kann etwas dauern. Die neue pandoc-Version liegt nun im Verzeichnis `~/.cabal/bin`, eine entsprechende Anpassung der `PATH`-Variable ist demnach sinnvoll.

3 Dokumente konvertieren

pandoc liest entweder von der Standard-Eingabe oder verbindet alle aufgeführten Dateien mit einer Leerzeile, bevor es konvertiert (ausgenommen sind Binärformate wie `EPUB` und `docx`). An Stelle einer Eingabedatei kann auch eine URL genannt werden. Ohne Angabe einer Ausgabedatei schreibt pandoc auf die Standardausgabe.

pandoc versucht, anhand der Dateiendung auf die jeweilige Auszeichnungssprache zu schließen. Die Dateiformate für Ein- und Ausgabe lassen sich aber auch explizit mit `-f|--from` und `-t|--to` spezifizieren:

```
pandoc -f html -t markdown hallo_welt.html > hallo_welt.md
```

Die Option `-o` ermöglicht die Angabe einer Ausgabedatei:

```
pandoc -f html -t markdown -o hallo_welt.md hallo_welt.html
```

4 Dokumente erstellen

pandocs Flexibilität erlaubt es, Dokumente in allen unterstützten Auszeichnungssprachen zu erstellen. Beliebt ist Markdown, da es nur wenig Aufwand erfordert und auch ohne Konvertierung gut zu lesen ist.

4.1 Markdown-Syntax

4.1.1 Absätze und Überschriften

Absätze werden mit einer oder mehreren Leerzeilen eingeleitet. Überschriften unterstützt pandoc mit zwei Varianten:

- Beim *Setext*-Stil werden Überschriften mit = (1. Ebene) und - (2. Ebene). unterstrichten. Der Setext-Stil unterstützt nur zwei Ebenen.
- Der *Atx*-Stil fasst Überschriften mit Doppelkreuz ein: `# Überschrift#`, `## Überschrift ##` usw.

Vor den Überschriften muss eine Leerzeile stehen, es denn, es dort befindet sich ebenfalls eine Überschrift.

Für jede Überschrift erzeugt pandoc automatisch einen *Identifier*. Leer- und Sonderzeichen werden dabei durch Bindestriche ersetzt[1]. Diese Identifier lassen sich über *interne Links* referenzieren:

```
Näheres findet sich in der [Einleitung](#einleitung).
```

Block-Zitate werden in Markdown wie Zitate in E-Mails mit einer schließenden spitzen Klammer gefolgt von einem Leerzeichen ausgezeichnet:

```
> Dies ist ein
> Blockzitat.
```

4.1.2 Verweise

Werden E-Mail-Adressen oder URLs in spitze Klammern eingefasst, erzeugt pandoc automatisch Verweise: `<http://www.wiwi.uni-bielefeld.de>`. Sollen an die Stelle der URLs eigens definierte Bezeichner treten, kann dies in Form von *Inline Links* geschehen:

```
Bitte folgen Sie [diesem Link](http://www.wiwi.uni-bielefeld.de/).
```

Neben Inline Links unterstützt pandoc auch sog. *Reference Links*. Sie bestehen aus zwei Teilen: Der erste Teil besteht aus dem Bezeichner und einem Identifier. Der zweite Teil definiert über den Identifier das Ziel:

```
[Mein Link][foo] leitet Sie weiter.
...

[foo]: http://www.wiwi.uni-bielefeld.de/ "Optionaler Titel"
```

Der Titel kann auch eingerückt in der nächsten Zeile aufgeführt werden.

[1] Für Details zu den Ersetzungsregeln siehe Header identifiers in HTML, LaTeX, and ConTeXt

4.1.3 Bilder und Fußnoten

Bilder verarbeitet pandoc auf ähnliche Weise wie Links. Ein Ausrufezeichen direkt vor einem Link wird als Referenz auf eine Bilddatei interpretiert: `![identifier](mt.png "Alternativer Text")`. Eine Einbindung via Reference Link ist ebenfalls möglich:

```
Ein Bild ![identifier] im Text.
...

[identifier]: mt.png
```

Üblicherweise werden Bilder mit einer Überschrift oder Legende versehen, auch *caption* genannt. Steht ein interner Link auf ein Bild allein in einem Absatz, erzeugt pandoc daraus ein Bild mit dem Identifier als caption-Inhalt.

Fußnoten können direkt im Text (inline) oder über einen Reference Link definiert werden:

```
Als Inline-Formatierung^[Dies ist die Fußnote.]...

Mittels Reference Link[^1] kann ...

[^1]: siehe dazu ...
```

An Stelle der Ziffer kann wieder ein Identifier gewählt werden. Soll eine lange Fußnote Absätze oder ähnliche Blockelemente enthalten, müssen diese vier Leerzeichen eingerückt werden:

```
Verweis auf eine [^lange_fußnote].

[^lange_fußnote]: Eine Fußnote mit Absätzen.

    Folgeabsätze müssen eingerückt sein, damit ihre
    Zugehörigkeit zu Fußnote deutlich wird.
```

4.1.4 Listen

Einfache Listen werden wie Absätze durch eine Leerzeile abgetrennt. Die Markierung der Listenelemente ist flexibel: Es können `*`, `+` oder `-` in beliebiger Mischung auftreten. Eingeschachtelte Listen erfordern ein Einrücken mit vier Leerzeichen:

```
Eine Liste:

* Element 1
    - eingerückte Liste
    - weiteres Element
- Element 2
+ Element 3
```

Listenelemente können wie Fußnoten Absätze und ähnliche Blockelemente enthalten; Voraussetzung ist wieder ein Einrücken um vier Leerzeichen.

Nummerierte Listen müssen durch ein Doppelkreuz gefolgt von einem Punkt samt Leerzeichen markiert werden:

```
#. blau
#. rot
```

Für Definitionslisten kann die Syntax von PHP Markdown Extra verwendet werden, jedoch bietet pandoc dazu eine kompaktere Syntax an:

```
Term 1
  ~ Definition

Term 2
  ~ Definition 1
  ~ Definition 2
```

Die Definitionen werden mit zwei Leerzeichen, einer Tilde und einem weiteren Leerzeichen eingeleitet. Die Begriff-Definitionspaare müssen mit einer Leerzeile voneinander getrennt werden. Je Begriff sind zudem mehrere Definitionen möglich.

4.1.5 Tabellen und horizontale Linien

Die Spaltenüberschriften bestimmen, wie der jeweilige Spalteninhalt ausgerichtet werden soll. Eine Tabelle muss mit einer Leerzeile enden oder mit einer gestrichelten Linie, wiederum gefolgt von einer Leerzeile. Beginnt die darauffolgende Zeile mit `Table:`, so wird diese automatisch zur Tabellenunterschrift gemacht.

```
 rechts     links   zentriert    Standard
-------     ------ -----------   --------
     ab     ab        ab           ab
    abd     abd       abd          abd
      x     x          x             x

Table: Eine einfache Tabelle.
```

Sind Spaltentitel nicht nötig, können sie weggelassen werden. Dann beenden gestrichelte Linien die Tabelle. Die Ausrichtung der Spalten wird der ersten Zeile entnommen.

```
-------     ------ -----------   --------
     ab     ab        ab           ab
    abd     abd       abd          abd
      x     x          x             x
-------     ------ -----------   --------
```

Spaltenüberschriften und Zellen können auch mehrere Zeilen beinhalten. Dazu muss die erste und letzte Zeile durchgehend gestrichelt und die Zeilen durch Leerzeilen getrennt sein.

```
-------------------------------------------------------------
 zentrierte standard-              rechts links
  Spalte    ausgerichtet    ausgerichtet ausgerichtet
----------- ------------ -------------- --------------------
   erste    Zeile                     0 mehrzeilige
                                        Zelle

  Second    row                       1 weitere mehrzeilige
                                        Zelle
-------------------------------------------------------------

Table: Auch die Tabellenüberschrift kann
mehrzeilig sein.
```

pandoc unterstützt noch weitere Formate; Details dazu sind im Abschnit Tables der Dokumentation zu finden.

Horizontale Linien können mit drei oder mehr Bindestrichen gesetzt werden: `---`.

4.1.6 Inline-Formatierung

Für die Formatierung von Wörtern lehnt sich pandoc an andere einfache Auszeichnungssprachen an:

- Betonung: `*emphasized*` → *emphasized*, aber auch `emph*asized*` → emph*asized*
- Starke Betonung: `**strong**` oder `__strong__` → **strong**
- Durchstreichung: `~~strikeout~~` → ~~strikeout~~
- hoch setzen: `a^2^+b^2^=c^2^` → $a^2+b^2=c^2$
- tief setzen: `N~0~` → N_0
- gesperrt gesetzt: `'verbatim'` → 'verbatim'

Ein Backtick muss von zwei Backtick-Paaren eingefasst werden:

```
Dies ist ein Backtick ``` ` ```.
```

4.1.7 Code-Blöcke

Gesperrt gesetzte Absätze, wie sie oft für die Darstellung von Quellcode zum Einsatz kommen, werden mit einem Tabulator oder vier Leerzeichen ausgezeichnet:

```
Jetzt folgt Code:

    def my_function():
      pass
```

In sog. *Fenced* Code-Blöcke ist das Einrücken nicht nötig, außerdem sind zusätzliche Angaben zur Formatierung möglich:

```
~~~ {#code .python .numberLines startFrom="100"}

def my_function():
  pass
~~~
```

Der Code-Abschnitt erhält hier einen Bezeichner (`#code`) und soll das Syntax Highlighting der Sprache Python nutzen. Die Zeilen werden außerdem ab 100 nummeriert.

pandoc unterstützt auch eine Kurzform:

```
‘‘‘python
def my_function():
  pass
‘‘‘
```

4.1.8 Mathematische Formeln

Alles, was zwischen zwei Dollar-Zeichen gesetzt wird, interpretiert pandoc als TeX-Formelausdruck. Wie die Formel im Ausgabeformat umgesetzt wird, hängt vom Format ab. Wenn möglich, wird sie direkt mit den Mitteln des Zielformats beschrieben, ansonsten wird der Formeltext selbst dargestellt.

4.1.9 Direkte Formatierungen

pandoc erlaubt die direkte Verwendung von HTML- oder DocBook-Tags überall im Text. Das gilt auch für TeX-Befehle.

4.2 Metadaten

Metadaten wie Titel, Autor etc. stehen in einem YAML[2]-formatierten Metadatenblock:

```
% Titel
% Autor
% Jahr
---
subtitle: Untertitel
email: mtessmer@wiwi.uni-bielefeld.de
publishers: Bereich *Computergestützte Methoden* (CoMet), Fakultät für
  Wirtschaftswissenschaften, Universität Bielefeld
lang: german
```

[2] *Yet Another Markup Language*; s. http://yaml.org/

```
documentclass: scrartcl
bibliography: literatur.bib
csl: din-1505-2.csl
abstract: |
  Dieser Beitrag ...
...
```

pandoc interpretiet die Einträge abhängig vom Ausgabeformat bzw. dem dazu vorliegenden Template. Die im Template definierten Variablen können entweder wie im Beispiel als Metadaten oder beim Aufruf von pandoc als Option übergeben werden: `-V lang=german`.

4.3 Literaturangaben

Mit der Erweiterung `pandoc-citeproc` ist pandoc in der Lage, Bibliografie-Dateien unterschiedlicher Formate zu verarbeiten. Dazu gehören BibTeX, BibLaTeX, EndNote usw. Ein Literaturverweis wird in eckige Klammern eingefasst und dem Schlüssel ein `@`-Zeichen vorangestellt: `[siehe @Doe2001, S. 22]`, `[@Doe2001, pp. 1-23]`, `[@Schulz2000; @Doe2001]`. Ein Minus-Zeichen vor dem `@` unterdrückt die Nennung des Autors: `Wie Schulz in [-@Schulz2000] schreibt ...` Im YAML-Metadatenblock definiert der Schlüssel `bibliography:` die zu verwendende Bibliografie-Datei.

Der Zitierstil lässt sich im Metadaten-Block mit dem Schlüssel `csl:` oder auf der Kommandozeile mit der Option `--csl din.csl` bestimmen. Welche Metadaten wie ausgewertet und dargestellt werden, hängt von der verwendeten CSL-Datei ab. Auf der Seite The Citation Style Language finden sich viele dieser CSL-Dateien.

Die Liste der Literaturhinweise werden von pandoc nach dem letzten Absatz eingefügt, d.h. als letzter Eintrag sollte im Dokument eine Überschrift wie `Literatur` o.Ä. stehen.

5 Arbeiten mit mehreren Dateien

Werden mehrere Dateien zugleich an pandoc übergeben, so muss der Metadaten-Block entweder in der ersten Datei de Metadaten stehen in einer separaten Metadaten-Datei: `pandoch kapitel1.md kapitel2.md metadaten.yaml`. Damit Überschriften etc. korrekt gesetzt werden, sollte am Ende jeder Datei eine Leerzeile sein.

Bei umfangreichen Texten wird neben der Einteilung in Kapitel manchmal eine weitere Gliederungsebene hinzugefügt, die die Kapitel zu Textteilen zuordnet. In LaTeX wird dies mit dem Gliederungsbefehl `\parts{...}` umgesetzt. Da pandoc vor der Verarbeitung alle eingehenden Dateien aneinander hängt, können die Befehle für die Einteilung einfach in eigenen Dateien stehen, bspw. `part_konzepte.md`:

```
\part{Konzepte}
```

Diese Dateien werden einfach an die passende Stelle zwischen die Eingabedateien gestellt:

```
pandoc ... -o buch.pdf vorwort.md part_konzepte.md monitoring.md ... \
    part_umsetzung.md ...
```

Diese zusätzliche Gliederung wirkt sich jedoch nur auf den Export via LaTeX/PDF aus; andere Exportformate wie bspw. HTML ignorieren die `\part`-Anweisungen.

6 Templates

Standardmäßig nutzt pandoc für die Konvertierung die im Verzeichnis `~/.pandoc/templates` hinterlegten Vorlagen. Um eine eigene LaTeX-Vorlage zu nutzen, kann die zugehörige Standardvorlage aus dem Verzeichnis `~/.cabal/share/pandoc-<version>/data/templates` dorthin kopiert und angepasst werden.

Soll pandoc ein anderes Template-Verzeichnis verwenden, muss dies mit der Option `--data-dir=` angegeben werden. In diesem Verzeichnis muss sich das Unterverzeichnis `templates` mit der zu verwendenden Template-Datei befinden. Ein Beispiel: Mit `pandoc --data-dir=. --template=default.latex ...` wird auf ein Template-Verzeichnis zugegriffen, das im selben Ordner wie das Dokument `article.md` liegt:

```
.
├── article.md
├── din-1505-2.csl
├── Makefile
└── templates
    ├── default.epub3
    ├── default.html
    ├── default.html5
    ├── default.html_fragment
    ├── default.latex
    ├── default.markdown
    └── default.plain
```

7 Fazit

pandoc besticht durch seine Fähigkeit, eine Vielfalt an Auszeichnungssprachen zu übersetzen. Die Erstellung eigener Dokumente wird durch viele kleine Erweiterungen der klassischen Markdown-Syntax erleichtert. Hier wurde nur ein kleiner Teil des Funktionsumfangs betrachtet und auf das Schreiben von Filtern oder Erweiterungen nicht eingegangen. Der interessierte Leser findet auf der Website des Projekts eine umfangreiche Dokumentation mit vielen Beispielen.

8 Revisionen

1 - 2014-11-10: erste Veröffentlichung
2 - 2014-11-21: Verarbeitung mehrerer Dateien um Hinweis auf `\part` erweitert