

HTML und PDF Dateien mit pandoc und gvim erstellen

Stefan Blechschmidt

Oktober 2013

Inhaltsverzeichnis

Vorwort	1
Ausgabeformate erstellen	2
Die Gvim Kurztaste	2
Formatierung erweitern	3
Rumex?	4

Vorwort

Rumex verwendet pandocs markdown weil man damit sehr einfach und schnell Text erstellen und verschiedenen Formate wandeln kann. Diese Funktion kann man aber auch außerhalb vom Rumex, also für die alltäglichen Beschreibungen und Notizen die so anfallen, verwenden.

Abgesehen aber ich es bei dieser Beschreibung auf die beiden Formate PDF und HTML in der Grundform wie sie pandoc Standardmäßig liefert. Also ohne Anpassung des Layouts weder bei HTML noch bei PDF.

Für PDF muss jedoch LaTeX auf dem Rechner installiert sein. Außerdem wurde für die Darstellung, siehe weiter unten, der PDF Dateien das Programm zathura gewählt.

Ausgabeformate erstellen

Um die unterschiedlichen Ausgabeformate mittels pandoc zu erstellen braucht man einige Optionen die auf der [Homepage von pandoc](#) sehr gut beschrieben sind.

Um den aktuellen Text nach HTML zu wandeln kann dieser Befehl verwendet werden.

```
pandoc -f markdown -t html5 --toc -s -o test.html test.md
```

Für PDF¹ funktioniert dieser Befehl

```
pandoc -f markdown -t latex --toc -V lang=ngermanb -o test.pdf test.md
```

Dabei ist der Parameter `--toc` für die Anzeige des Inhaltsverzeichnis zuständig.

Da diese beiden Formate innerhalb des Editors gvim zur Verfügung stehen sollten brauchte es noch die Definition der gvim Kurztasten.

Die Gvim Kurztaste

Die Unterstützung soll nur für gvim aber auch ausserhalb Rumex funktionieren darum werden die benötigten Befehle in die Datei `~/.gvimrc` geschrieben.

Verwendet werden, in diesem Beispiel, die Tasten F8 und F9.

F8 für HTML und F9 für das PDF Format.

Ausserdem werden noch die zwei Tasten ALT-F8 und ALT-F9 für die Anzeige konfiguriert wobei zathura² für die PDF Anzeige verwendet wird.

Nachfolgende Zeilen in die `~/.gvimrc` Datei kopieren und fertig ist diese Anpassung.

```
" HTML Datei erstellen
map <F8> :w<cr>:!pandoc -f markdown -t html5 --toc -s -o <C-R>=expand("%:r")<CR>.html %<CR>
" HTML Datei anzeigen
map <A-F8> :!x-www-browser <C-R>=expand("%:r")<CR>.html&<CR><CR>

" PDF Datei erstellen
map <F9> :w<cr>:!pandoc -f markdown -t latex --toc -V lang=ngermanb -o <C-R>=expand("%:r")<CR>.pdf %<CR>
" PDF Datei anzeigen
map <A-F9> :!zathura <C-R>=expand("%:r")<CR>.pdf&<CR><CR>
```

¹Für das Erstellen des PDF Formats muss jedoch LaTeX auf dem System installiert werden.

²Zathura wurde deshalb gewählt weil dieses Programm ähnlich wie vi über die Tastatur bedient werden kann.

Formatierung erweitern

Die erzeugte HTML Datei besitzt keine Formatierung bzw. verwendet die Standard Darstellung des Browsers.

Kopf- und Fusszeile werden dadurch nicht, vom restlichen Text, unterschieden. Auch das Inhaltsverzeichnis ist im ersten Moment als solches nicht zu erkennen.

Es muss also eine kleine Format Änderung eingebaut werden.

```
/* gvim_f8.css */

header {
    text-align: center;
    border-bottom: 1px solid silver;
}

nav#TOC {
    border-bottom: 1px solid silver;
    font-size: 0.8em;
}

section.footnotes {
    color: gray;
    font-size: 0.8em;
    margin-top: 4em;
}

section.footnotes hr {
    border: none;
    border-top: 1px solid silver;
    margin-left: 0;
    width: 40%;
}
```

Die Zeile in der Datei `~/.gvimrc` ändert sich dadurch ein wenig. Sie wird um die beiden Optionen `--self-contained` und `--css ~/.pandoc/gvim_f8.css` erweitert.

" HTML Datei erstellen

```
map <F8> :w<cr>:!pandoc -f markdown -t html5 --toc --self-contained --css ~/.pandoc/gvim_f8.css
```

Parameter Erweiterung

--self-contained Durch diesen Parameter wird die CSS Datei in den HTML Quellcode eingebunden. Funktioniert übrigens auch mit Bildern. Es muss also nur die HTML Datei hoch geladen werden.

–css Die CSS Formatierungsdatei. Da die Datei mittels `--self-contained` in die HTML Datei eingebunden wird muss diese nicht auf dem Server mit hoch geladen werden.

Rumex?

In Rumex kann man diese Funktion natürlich auch verwenden. Am besten erstellt man sich dazu ein eigenes Unterverzeichnis und dort die Datei `index.md` mit den Texten.

Die PDF Datei dieser Beschreibung kann man sich [hier](#) ansehen. Die Markdown Quelldatei kann man sich [hier](#) holen.
