HTML und PDF Dateine mit pandoc und gvim erstellen

Stefan Blechschmidt

Oktober 2013

Inhaltsverzeichnis

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 habe ich es bei dieser Beschreibung auf die beiden Formate `PDF`[1] und `HTML`.

[1] Für die Umwandlung nach PDF muss jedoch LaTeX mit den entsprechenden Paketen auf dem Rechner installiert sein. Außerdem wird für die Darstellung der PDF Dateien das Programm zathura verwendet, was natürlich auch installiert sein sollte.

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 PDF 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 zathura[2] 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>.htm %<CR><C
" HTML Datei anzeigen
map <A-F8> :!x-www-browser <C-R>=expand("%:r")<CR>.htm&<CR><CR>


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

[2] Zathura 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 gleich zu erkennen. Dieses kann mit ein wenig CSS geändert 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%;
}


/* ---------------------------------
    Umformatieren der Überschriften
    --------------------------------- */

/* Ab der zweiten h1 Überschrift bekommt diese
    einen größeren Abstand. */

h1:nth-of-type(n+2) {
    margin-top: 4em;
}
```

```
/* Der Link der Überschriften sollte
    nicht unterstrichen
    und in schwarz dargestellt werden. */
h1 a,
h2 a,
h3 a,
h4 a,
h5 a,
h6 a {
    text-decoration: none;
    color: #000;
}
```

Gespeichert wird diese Datei in einem separaten Unterverzeichnis. Ich verwende dazu `~/.pandoc`.

Die Zeilen für die HTML Erstellung, in der Datei `~/.gvimrc`, ändert sich dadurch ein wenig. Es sind die Optionen `--self-contained` und `--css ~/.pandoc/gvim_f8.css` hinzu gekommen.

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

Anmerkung zu den neuen Parametern

`--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.

Warum `.htm` verwendet wird

Um nicht mit der git Verwaltung, innerhalb Rumex / Unterverzeichnis `.rx`, in Konflikt zu kommen wurde für die HTML Dateien, die per F8 erstellt werden, die Endung `.htm` verwendet. Durch einen Eintrag in der Datei `.gitignore` können diese so von der git Verwaltung ausgenommen werden.

```
.rx/*.htm
.rx/*.pdf
```

Zwar werden diese Dateien im lokalen Arbeitsverzeichnis gespeichert aber nicht über den Befehl `make online` hoch geladen.

`.htm` Dateien im `.rx` Verzeichnis aufräumen

Um die evtl. angefallenen `.htm` Dateien, die sich im `.rx` Verzeichnis befinden, löschen zu können gibt es auch einen entsprechenden make Befehl unter Rumex. Mit

```
make f8clean
```

werden alle `.htm` Dateien im `.rx` Verzeichnis gelöscht. Dieser Befehl wurde auch in

```
make clean
```

aufgenommen.

Die PDF Datei dieser Beschreibung kann man sich hier ansehen. Die Markdown Quelldatei kann man sich hier holen.
