

DARIAH Working Paper Workflow

Redaktionsumgebung – Work in Progress

Thorsten Vitt

Mirjam Blümm



Thorsten Vitt, Mirjam Blümm: „DARIAH Working Paper Workflow“. [DARIAH-DE Working Papers](#) Nr. 0.
Göttingen: DARIAH-DE, 2016. URN: **TODO: urn**.

Dieser Beitrag erscheint unter der
Lizenz [Creative-Commons Attribution 4.0](#) (CC-BY).

Die *DARIAH-DE Working Papers* werden von Mirjam Blümm,
Thomas Kollatz, Stefan Schmunk und Christof Schöch
herausgegeben.



Zusammenfassung

Für die Publikation der DARIAH-Working-Papers gibt es einen Workflow auf der Basis von Markdown, das mit Pandoc und LuaLatex formatiert wird.

Dieser Artikel beschreibt die Installation der Working-Paper-Vorlage und einige weitere Themen für Redakteure. Autoren finden Hinweise zur Formatierung des Texts in den [Autorenhinweisen](https://github.com/DARIAH-DE/dwp-template/blob/master/DWP-Autorenhinweise.pdf)¹.

¹<https://github.com/DARIAH-DE/dwp-template/blob/master/DWP-Autorenhinweise.pdf>

Inhaltsverzeichnis

1	Voraussetzungen	4
2	Installation	4
3	Benutzung	5
4	Troubleshooting	5
4.1	irgendwas mit UTF-8	5
4.2	PDF-Datei kann nicht erzeugt werden, keine ordentliche Fehlermeldung	5
4.3	Fehlermeldung wegen <code>\bibname</code>	5
4.4	<code>.bib</code> vorhanden und kein Literaturverzeichnis	6
4.5	Schwieriges \LaTeX	6
5	Umgang mit Dateien in Office-Formaten	7

1 Voraussetzungen

- Pandoc
- LaTeX-Installation mit LuaLaTeX, z.B. aktuelles TeX-Live
- pandoc-citeproc (für das Literaturverzeichnis-Processing)
- Python 3 (für das Convenience-Skript)

2 Installation

(Es gibt ein einfaches, experimentelles Installationsscript `install.sh`, das auf Linux und MacOS X funktionieren sollte.)

Die Schriften [WeblySleek UI](#)² und [losevka in der Default-Variante \(Download 01-*\)](#)³ entsprechend dem Betriebssystem installieren. Für übliche Linux-Distributionen ist es ausreichend, die TTF-Dateien in den Ordner `~/.fonts` zu legen.

Einige Dateien aus diesem Verzeichnis müssen über das Dateisystem verteilt werden. Ich empfehle, die entsprechenden Dateien per symbolischem Link zu verlinken statt sie zu kopieren, um für Aktualisierungen gerüstet zu sein:

- **DWP.latex** ist das Pandoc-Template, es muss in das Verzeichnis `~/.pandoc/templates`.
- Die ***.cs1-Dateien** sind die Styles für die Literaturverwaltung, sie stammen aus dem [Zotero Style Repository](#)⁴. `dwp.py` sucht diese Styles ebenfalls im Verzeichnis `~/.pandoc/templates`
- Die Bilder aus dem `img`-Ordner werden für die Titelseite benötigt. Sie werden in einem LaTeX-Baum gesucht.
- `dwp.py` ist ein Script zum einfachen Aufrufen von Pandoc mit entsprechenden Parametern. Es sollte irgendwo in den `$PATH`.

Unter der Annahme, dass dieses Verzeichnis unter `~/projects/dwp-template` liegt:

```
mkdir -p ~/.pandoc/templates
cd ~/.pandoc/templates
ln -s ~/projects/dwp-template .
mkdir -p `kpsexpand '$TEXMFHOME/tex/latex'`
cd `kpsexpand '$TEXMFHOME/tex/latex'`
ln -s ~/projects/dwp-template .
cd /usr/local/bin
sudo ln -s ~/projects/dwp-template .
```

²<http://www.dafont.com/weblysleek-ui.font>

³<https://be5invis.github.io/losevka/>

⁴<https://www.zotero.org/styles?q=chicago&format=author-date>

Nach erfolgreicher Installation sollte es in jedem Verzeichnis möglich sein, mit `dwp datei.md` die entsprechende Datei in PDF zu übersetzen.

3 Benutzung

Das beigefügte Skript `dwp.py` kann einfach mit `dwp artikeldatei.md` aufgerufen werden. Es ruft Pandoc mit den richtigen Parametern auf, um `artikeldatei.pdf` zu erzeugen. Das kann dann z. B. so aussehen:

```
pandoc -o article.pdf --latex-engine=lualatex --template=DWP \
--filter=pandoc-citeproc --bibliography=article.bib \
--csl=$HOME/.pandoc/templates/chicago-author-date.csl \
--metadata=link-citations:true \
article.md
```

`dwp` kann auch mit weiteren Pandoc-Parametern aufgerufen werden, es reicht alle Parameter an Pandoc weiter und lässt dafür ggf. die selbst generierten Versionen weg.

4 Troubleshooting

4.1 irgendwas mit UTF-8

Eingabedatei ist nicht UTF-8-codiert. Im Texteditor öffnen und als UTF-8 speichern.

4.2 PDF-Datei kann nicht erzeugt werden, keine ordentliche Fehlermeldung

1. TeX-Datei erzeugen lassen – das geht entweder auf die entsprechende Rückfrage oder mit `dwp -o article.tex article.md`
2. TeX-Datei manuell mit LuaLaTeX übersetzen lassen: `lualatex article.tex`, Fehlermeldungen prüfen

4.3 Fehlermeldung wegen `\bibname`

Metadatenfeld `lang: de` oder `lang: en` angeben!

4.4 .bib vorhanden und kein Literaturverzeichnis

Heißt die .bib so wie die .md?

Wird auch wirklich zitiert, dh mit @mueller2012 (entsprechend den Keys im .bib)? Siehe DWP-Autorenhinweise.md und DWP-Autorenhinweise.bib: Im .md muss @Eijkhout1991 oder [@Eijkhout1991] stehen, um den u.a. Eintrag zu zitieren:

```
@BOOK{Eijkhout1991,
  title = {\TeX\ by Topic. A \TeX nician's Reference},
  publisher = {Addison-Wesley},
  year = {1991},
  author = {Victor Eijkhout},
  address = {London},
  keywords = {general},
}
```

Literatur, die nicht in Pandoc-Syntax zitiert wird, landet auch nicht im Literaturverzeichnis.

Wenn Autoren .bib anliefern, aber *nicht* korrekt zitieren, kann man sich behelfen, indem man exakt folgenden Metadateneintrag zum YAML-Header hinzufügt:

```
nocite: '@*'
```

In diesem Fall werden alle Einträge der .bib-Datei im Literaturverzeichnis gesetzt, auch solche, die nicht zitiert werden. Convenience-Features wie einheitliche Zitationskeys und Links von der Jahreszahl ins Literaturverzeichnis gehen damit natürlich nicht.

4.5 Schwieriges \LaTeX

Im Prinzip kann man einfache \LaTeX -Kommandos und -Umgebungen direkt ins Markdown schreiben. Wenn das nicht klappt, muss man Pandoc sagen, dass ein entsprechender Abschnitt rohes TeX ist. Dazu verwendet man die [Raw-Attribute](https://pandoc.org/MANUAL.html#extension-raw_attribute)⁵-Funktion: Man schreibt einen Codeblock und unmittelbar dahinter {=latex}, also für größere Blöcke

```
```${=latex}
\begin{tabular}{ll}
 \begin{tabular}{l} foo \end{tabular} & x \end{tabular}
\end{tabular}
```
```

⁵https://pandoc.org/MANUAL.html#extension-raw_attribute

5 Umgang mit Dateien in Office-Formaten

Wenn Dateien in Office-Formaten angeliefert werden empfiehlt sich folgender Workflow:

1. OpenOffice-Dateien nach DOCX konvertieren, die Pandoc-Unterstützung von DOCX ist besser
2. `pandoc -o article.md --extract-media=article_assets article.docx`. Erzeugt aus der Word-datei eine Markdown-Datei namens `article.md`, extrahiert die Bilder und speichert sie im Verzeichnis `article_assets` (wird ggf. angelegt)
3. Markdown-Datei nachbearbeiten

Vorteil: Bilder werden gleich ausgepackt, Grundformatierungen bleiben erhalten