



Pandoc-Vorlage für Arbeiten an der DHBW-Mosbach

Unterstützt Projektarbeiten, Studienarbeiten und Bachelorarbeiten

Bachelorarbeit T3_3300

des Studiengangs Angewandte Informatik

an der Dualen Hochschule Baden-Württemberg Mosbach

von

Maxi Muster

Max Muster

10. August 2022

Bearbeitungszeitraum
Matrikelnummer, Kurs
Dualer Partner
Betreuer*in des Dualen Partners
Gutachter*in der Dualen Hochschule

12 Wochen
1234567 & 1234568, MOS-TINF19X
My Cool Company, Berlin
Herr Müller
Frau Prof. Dr. Meier

Erklärung

Wir versichern hiermit, dass wir unsere Bachelorarbeit T3_3300 mit dem Thema: *Pandoc-Vorlage für Arbeiten an der DHBW-Mosbach* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Mosbach, 10. August 2022

Maxi Muster

Max Muster

Zusammenfassung

Dies ist ein zu kurzes Abstract. Das Abstract für tatsächliche Projektarbeiten sollte 200 bis 250 Wörter lang sein.

Abstract

This is a similarly too short abstract, but in English. The abstract for actual papers should be between 200 and 250 words in length.

Inhaltsverzeichnis

Abbildungsverzeichnis	V
Tabellenverzeichnis	VI
1 Einleitung	1
2 Verwendung	2
2.1 Drone	2
2.2 Docker	2
2.3 Manuell	2
2.3.1 Windows	3
2.3.2 Linux	4
3 Demo	6
3.1 Überschrift-Ebene 2	7
3.1.1 Überschrift-Ebene 3	7
3.2 Extensions	7
Literaturverzeichnis	9

Abbildungsverzeichnis

3.1 Firmen-Logo	7
---------------------------	---

Tabellenverzeichnis

3.1 Tabellen-Beschriftung	7
-------------------------------------	---

Abkürzungen

DHBW Duale Hochschule Baden-Württemberg

1 Einleitung

Eine Vorlage, um wissenschaftliche Arbeiten für die Duale Hochschule Baden-Württemberg (DHBW) in Pandoc verfassen zu können.

Dabei werden die in [1] beschriebenen Richtlinien nach bestem Gewissen umgesetzt.

2 Verwendung

2.1 Drone

Am einfachsten ist die Verwendung mit [Drone](#). Wenn Drone und Docker installiert sind, reicht folgender Befehl:

```
drone exec
```

Basierend auf Drone lässt sich auch alternativ zu GitHub-Actions eine Build-Automatisierung nutzen.

2.2 Docker

Um diese Vorlage zu verwenden kann das Docker-Image [siphakor/extended-pandoc](#) verwendet werden.

Unter Linux kann folgender Befehl zum Kompilieren der PDF mit Docker verwendet werden:

```
docker run --rm --volume $(pwd):/data --entrypoint make siphakor/extended-pandoc
```

2.3 Manuell

Alternativ können die nötigen Extensions selbst installiert werden. Die Liste der Extensions findet sich in [der Readme des Docker-Images](#).

Anschließend kann mit `make index.pdf` die PDF-Datei kompiliert werden. Unter Windows kann sich der entsprechende Befehl aus dem `Makefile` entnommen werden.

2.3.1 Windows

Im Folgenden wird die Installation unter Windows beschrieben. Die folgenden Befehle sollten in PowerShell ausgeführt werden (alternativ [pwsh](#)).

Als Package Manager wird Chocolatey genutzt, da diese einfache Installationen und Updates ermöglichen. Eine Shell mit Administrativen rechten wird benötigt! Diese kann mit [Win]+[X], [A] geöffnet werden.

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
irm https://get.scoop.sh | iex
irm https://community.chocolatey.org/install.ps1 | iex
choco feature enable -n=allowGlobalConfirmation
```

allowGlobalConfirmation schaltet Rückfragen bei der installation ab.

Zunächst wird die VCS git installiert:

```
choco install git
```

Weiterhin wird pip benötigt um die Python-basierten Extensions zu installieren:

```
choco install python.install
python -m ensurepip
```

Nun benötigen wir noch den `make` command aus den GNU coreutils. Hierzu werden alle GNU coreutils installiert:

```
choco install gnuwin32-coreutils.install
```

Abschließend werden jetzt die Pandoc- und Latex-Umgebung installiert. In den folgenden Befehlen wird der LATEX compiler [MikTeX](#) installiert. Dies bietet unter anderem eine grafische Oberfläche und eine einsteigerfreundlichere Paket-Verwaltung. Das Argument `\ThisUser` installiert MikTeX nur für den aktuellen Benutzer, das ist zu installieren von Updates und Abhängigkeiten empfohlen.

```
choco install miktex.install \ThisUser
```

Nach der installation müssen in MikTeX 1. Updates gesucht und 2. Dependencies installiert werden. Es wird empfohlen automatisch Dependencies nachzuladen, mit der „update dependencies on-the-fly“ option.

Nun werden der [Pandoc](#) converter und ein paar Erweiterungen installiert.

```
choco install pandoc pandoc-crossref
pip install pandoc-include --user
```

Der Python installer für Pandoc Acronyms fügt keinen PATH Eintrag hinzu, deshalb verwenden wir den offiziellen installer

```
mkdir -p ~/apps
pushd ~/apps || exit
git clone https://github.com/kprussing/pandoc-acro.git
pushd pandoc-acro || exit
python setup.py install
popd
popd
```

[SVG](#) können mit GNOME librsvg eingebettet werden. Windows muss diese Library nachziehen.

```
choco install rsvg-convert
```

2.3.1.1 Automatischer Setup

Ein [automatischer Setup](#) welcher die obg. Schritte durchführt kann wie folgt durchgeführt werden:

```
./setup.ps1
```

2.3.2 Linux

Der Einfachheit halber am besten zunächst den [Homebrew](#) Package Manager installieren:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Python ist in den allermeisten Linux-Distributionen vorinstalliert, eventuell ist es als `python` statt `python3` verfügbar:

```
python3 -m ensurepip
```

Schlussendlich können Pandoc, die nötigen Extensions und die [TeX Live](#) LaTeX-Umgebung installiert werden.

```
brew install pandoc pandoc-crossref texlive  
pip install pandoc-acro pandoc-include --user
```

[SVG](#) können mit GNOME `librsvg` Eingebettet werden. Die Library ist auf einigen Distros bereits installiert, ansonsten muss die Library bezogen werden.

```
brew install librsvg
```

3 Demo

Unterstützt werden alle typischen Markdown-Features, sowie die nativen Erweiterungen von Pandoc:

- *kursiv*, **fett**, ***beides***, ~~durchgestrichen~~, unterstrichen, KAPITÄLCHEN
- Definitionen und Listen:

Wort: Dies ist eine Beschreibung über mehrere Zeilen.

Unsortierte Listen:

- Eins
- Zwei
- Drei

Unsortierte Listen:

1. Eins
2. Zwei
3. Drei

- Fußnoten¹²
- Zitation/Quellenangaben [1, Abschn. 9.1], [1, S. 23]
- Latex-Ausdrücke, zum Beispiel für mathematische Ausdrücke $\sum_{i=1}^n \frac{1}{a_b} \cdot i$
- Bilder
- Tabellen:

¹Dies ist eine Fußnote.

²Dies ist eine zweite Fußnote.



Abbildung 3.1: Firmen-Logo

Tabelle 3.1: Tabellen-Beschriftung

Spalte 1	Spalte 2	Spalte 3
A	B	C

- Code und Code-Blöcke:

```
print("Hello World");
```

Listing 3.1 Ein Code-Block der nicht im Fließtext ist!

```
print("Floating Code Block!")
```

Hier geht der Stichpunkt weiter.

Für weitere native Pandoc-Features, siehe [die offizielle Dokumentation](#).

3.1 Überschrift-Ebene 2

3.1.1 Überschrift-Ebene 3

3.1.1.1 Überschrift-Ebene 4

3.1.1.1.1 Überschrift-Ebene 5

3.2 Extensions

Durch Extensions wird zusätzliche Funktionalität zur Verfügung gestellt:

pandoc-acro — **Abkürzungen:** Abkürzungen können in der `acronyms.yaml` definiert werden:

```
acronyms:
  options:
    list/heading: chapter*
    make-links: true
```

Und anschließend verwendet werden: **DHBW**; Duale Hochschule Baden-Württemberg

pandoc-crossref — **Referenzen:** Referenzen für Abschnitte (Abschnitt 3.2), Bilder (Abbildung 3.1), Code-Blöcke (Listing 3.1) und Tabellen (Tabelle 3.1).

Literaturverzeichnis

- [1] DHBW Fachkommission Technik, „Leitlinien für die Bearbeitung und Dokumentation von schriftlichen Arbeiten - Fachkommission Technik“. Okt-2021 [Online]. Verfügbar unter: https://www.mosbach.dhbw.de/fileadmin/user_upload/dhbw/studiengaenge/ai/191212_Leitlinien_Praxismodule_Studien_Bachelorarbeiten.pdf. [Zugegriffen: 10-Juni-2022]