

README

Jan Unger

10. August 2020

Inhaltsverzeichnis

1	Software	3
2	Erste Schritte	4
3	Github-Repository erstellen – klonen	5
4	Script Beschreibung	6
5	Schreiben in Markdown	7
6	Markdown – Latex – PDF erstellen	8
7	Quellen	8
8	Listen	8
9	Anführungszeichen	9
10	Grafik – Abbildung	9
11	Tabelle	10
12	Mathe	10
13	Texthervorhebung	11
14	Code	11
15	Links	11
16	Absätze	12

Abbildungsverzeichnis

1	Grafiken-Bsp	9
2	Abbildung-Bsp	10

Tabellenverzeichnis

1	Tabelle-Bsp	10
---	-----------------------	----

Listings

1	HalloWelt	11
---	---------------------	----

Erstellt Websites & Latex-Files mit Markdown und Pandoc. Projekt wurde getestet unter »Ubuntu 18.04.3 LTS« und »Win10« (erfordert **Git Bash**)

1 Software

- Git Bash¹
- Github-Repository klonen²
- Texlive³
- Pandoc⁴
- Imagemagick⁵
- Editor Visual Studio Code⁶
- Editor Atom⁷
- Editor Notepad++⁸
- TeXstudio⁹
- Tablesgenerator¹⁰
- hpi-dokumentvorlagen-latex¹¹
- Zotero¹²
- Wordpress¹³
- XAMPP Apache + MariaDB + PHP¹⁴
- Filezilla¹⁵

¹<https://git-scm.com/downloads>

²<https://github.com/jul-eu/Vektorgrafiken-SVG-EPS.git>

³<https://www.tug.org/texlive/>

⁴<https://pandoc.org/installing.html>

⁵<https://imagemagick.org/script/download.php>

⁶<https://code.visualstudio.com/>

⁷<https://atom.io/>

⁸<https://notepad-plus-plus.org/downloads/>

⁹<https://www.texstudio.org/>

¹⁰https://www.tablesgenerator.com/latex_tables

¹¹<https://osm.hpi.de/theses/tipps#dokumentvorlagen-latex>

¹²<http://www.zotero.org/styles/ieee>

¹³<https://de.wordpress.org/download/>

¹⁴<https://www.apachefriends.org/de/index.html>

¹⁵<https://filezilla-project.org/>

- VM VirtualBox¹⁶
- Ubuntu¹⁷

2 Erste Schritte

Files anpassen:

1. `scripteBash/sed.sh`
 - `codelanguage`: HTML5, Python, Bash, C, C++, TeX
 - CMS Server Pfad: `https://bw-ju.de/#`
 - Bildformat: svg, png, jpg, webp
2. `scripteBash/gitversionieren.sh`
 - `»/media/jan/usb/repos/notizenUbuntu«`
 - `»/media/jan/virtuell/repos/notizenUbuntu«`
3. `projekt.sh`
 - `THEMA=»Vektorgrafiken-SVG-EPS«`
 - `»/media/jan/usb/backup/notizenUbuntu«`
 - `»/media/jan/virtuell/backup/notizenUbuntu«`
 - `»/media/jan/usb/archiv/notizenUbuntu«`
 - `»/media/jan/virtuell/archiv/notizenUbuntu«`
4. `content/metadata.tex`
 - Datum, Titel, Autor
5. `content/titelpage.tex`
 - `»Grafiken/logo.pdf«`

Markdown-Files erstellen

1. Erstelle eine Datei `»neu.md«` im Ordner `»md/«`
 - Bilder nach `images/` kopieren
 - Vektorgrafiken nach `Grafiken/` kopieren
2. Script ausführen: `projekt.sh`

¹⁶<https://www.virtualbox.org/>

¹⁷<https://ubuntu.com/download>

Linux-Terminal öffnen oder unter Win10 Git Bash-Terminal öffnen

```
$ ./projekt.sh
```

- 0) Projekt aufräumen
- 1) Projekt erstellen
- 2) Markdown in (tex, html5) + sed (Suchen/Ersetzen)
- 3) Kapitel erstellen + Scripte ausführen
- 4) Fotos optimieren (Web, Latex)
- 5) www + index.html
- 6) git init
- 7) git status + git log
- 8) Git-Version erstellen
- 9) Backup + Archiv erstellen
- 10) Beenden?

Eingabe Zahl >_

3. Latex-PDFs erstellen: make

```
$ make
$ make clean
$ make distclean
```

4. Repository auf Github erstellen

3 Github-Repository erstellen – klonen

GitHub's maximum file size of **50 MB**

Repository auf Github erstellen

HTTPS oder SSH

HTTPS: <https://github.com/jul-eu/Vektorgrafiken-SVG-EPS.git>

SSH: <git@github.com:jul-eu/Vektorgrafiken-SVG-EPS.git>

create a new repository

```
echo "# README" >> README.md
```

```
git init
```

```
git add .
```

```
git commit -m "git init"
```

or push an existing repository

```
git remote add origin https://github.com/jul-eu/Vektorgrafiken-SVG-  
↪ EPS.git
```

```
git push -u origin master
```

Github-Repository klonen

```
git clone https://github.com/jul-eu/Vektorgrafiken-SVG-EPS.git
```

4 Script Beschreibung

```
$ ./projekt.sh
```

1. Projekt erstellen

- Verz. erstellen, wenn nicht vorhanden

2. Markdown in *.tex und *.html

- Markdown in Latex + HTML5 + Wordpress
- sed > Wordpress
- sed > Latex

3. Kapitel erstellen + Scripte ausführen

- Alle Abbildungen »images/« in Markdown speichern.
 - »archiv/input-img.txt«
- Latex Kapitel erstellen.
 - Kopiere »tex-pandoc/.tex« nach »tex/«
 - »tex/« **Handarbeit...** für opt. Ergebnisse!
 - Kopiere »archiv/inhalt.tex« nach »content/«
 - make – Latex-PDF erstellen
- Tabellen als PDFs in Latex einfügen. »Tabellen/ ?«
- Inhalt vom Projektverzeichnis.
 - »archiv/Projekt-Inhalt.txt«
- Quellcode »code/« in Latex speichern.
 - »archiv/Quellcode-files.tex« HTML, Python, Bash, C, C++, TeX
- Artikel aus den Ordnern erstellen
 - »tex/«
 - »archiv/«
 - »Tabellen/«
 - »content/beispiele/tex/«
 - wird gespeichert in »Artikel/«

- Alle Abbildungen »images/« in Latex speichern
 - »archiv/Pics-files.tex«
 - Bildgröße: `width=.80\\textwidth`
- 4. Fotos optimieren (Web, Latex)
- 5. www + index.html
 - »html/alle-pics.html« erstellen
 - »index.html« erstellen
- 6. `git init`
- 7. `git status + git log`
- 8. Git-Version erstellen
 - **Pfade** anpassen in `gitversionieren.sh`
 - lokales Repository: master
 - Github Repository: origin/master
 - Backup Repository: backupUSB/master
 - »/media/jan/usb/repos/notizenUbuntu«
 - Backup Repository: backupHD/master
 - »/media/jan/virtuell/repos/notizenUbuntu«
- 9. Backup + Archiv erstellen
 - **Pfade** anpassen in `projekt.sh`
 - THEMA=»Vektorgrafiken-SVG-EPS«
 - »/media/jan/usb/backup/notizenUbuntu«
 - »/media/jan/virtuell/backup/notizenUbuntu«
 - »/media/jan/usb/archiv/notizenUbuntu«
 - »/media/jan/virtuell/archiv/notizenUbuntu«

5 Schreiben in Markdown

1. Markdown
2. Textauszeichnung – Was ist wichtig?, Tabellen, Bilder, Quellcode, Literatur, Links
3. Rechtschreibprüfung ¹⁸
4. Literatur ¹⁹

¹⁸<https://languagetoolplus.com/?pk-campaign=addon2-popup-logo>

¹⁹<https://www.zotero.org/user/login>

6 Markdown – Latex – PDF erstellen

1. Markdown > Latex: `$ projekt.sh` Script (pandoc)
2. Hand-Kopie: `tex_pandoc/` `tex/`
3. Referenzen: Links prüfen
 - Bild (`\autoref\{fig:bild\}`).
 - Tabelle (`\autoref\{tab:tabellen\}`).
 - Kapitel (`\autoref\{sec:zusammenfassung\}`).
 - Code (`\autoref\{code:hallowelt\}`).
4. Latex > PDF: `$ make Makefile` (latexmk)

7 Quellen

Quelle: Monk [2]

Quelle: Norbert [3]

Quelle: Kofler, Zingsheim, Gebeshuber, Widl, Aigner, Hackner, Kania, Kloep und Neubauer [1]

Quelle: `[@monk:2016:action]`

Quelle: `[@homofaciens:2018:projekt]`

Quelle: `[@kofler:2018:hacking]`

8 Listen

ungeordnete Liste

- a
 - b
 - bb
 - c
-
- a
 - b
 - bb
 - c

Sortierte Liste

1. eins

2. zwei

3. drei

1. eins

2. zwei

3. drei

Sortierte Liste

a) a

b) b

c) c

a) a

b) b

c) c

9 Anführungszeichen

»Anführungszeichen«

"Anführungszeichen"

10 Grafik – Abbildung

Grafiken-Bsp vgl. (Abbildung 1).



Abbildung 1: Grafiken-Bsp

![Grafiken-Bsp] (Grafiken/logo.eps) {width=30%}

Abbildung-Bsp vgl. (Abbildung 2).

![Abbildung-Bsp] (images/Chili-1.pdf) {width=60%}

11 Tabelle

Tabelle-Bsp vgl. (Tabelle 1).

Tabelle 1: Tabelle-Bsp		
Nr.	Begriffe	Erklärung
1	a1	a2
2	b1	b2
3	c1	c2
4	a1	a2

```
|**Nr.**|**Begriffe**|**Erklärung**|
|---:|:-----|:----|
| 1      | a1          | a2
| 2      | b1          | b2
| 3      | c1          | c2
| 4      | a1          | a2
```

12 Mathe

$[V] = [\Omega] \cdot [A]$ o. $U = R \cdot I$ o. $R = \frac{U}{I}$
 $[V] = [\Omega] \cdot [A]$ o. $U = R \cdot I$ o. $R = \frac{U}{I}$
 $\rightarrow \{I\}$



Abbildung 2: Abbildung-Bsp

5 cm, $a \cdot b$, \dots , Ω
100°C
80 %

```
$5~cm$, $a \cdot b$, $\cdots$, $\Omega$  
$100^\circ\text{C}$  
// ACHTUNG: Prozentzeichen macht Probleme in HTML und Latex  
// Z.B. 80 %  
$80~\%$ // in Latex  
$80~\%$ // in HTML
```

Matheumgebung:

$$\sum_{i=1}^5 a_i = a_1 + a_2 + a_3 + a_4 + a_5$$

```
\begin{align*}  
    \sum_{i=1}^5 a_i = a_1 + a_2 + a_3 + a_4 + a_5  
\end{align*}
```

13 Text hervorhebung

Fett oder *Kursiv*

****Fett**** oder **Kursiv**

14 Code

HalloWelt vgl. (Listing 1).

Listing 1: HalloWelt

```
// hallowelt.c  
#include <stdio.h>  
int main(void) {  
    printf("Hallo Welt!\n");  
    return 0;  
}
```

15 Links

<https://google.de> oder Google

<<https://google.de>> oder [Google] (<https://google.de>)

Fussnote²⁰

Fussnote[^1]

[^1]: <<https://bw-ju.de/>>

16 Absätze

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: »Dies ist ein Blindtext« oder »Huardest gefburn«? Kjift - mitnich-ten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein.

Fremdsprachige Texte wie »Lorem ipsum« dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Literaturverzeichnis

- [1] M. Kofler, A. Zingsheim, K. Gebeshuber, M. Widl, R. Aigner, T. Hackner, S. Kania, P. Kloep und F. Neugebauer. *Hacking & Security – Das umfassende Handbuch*. Rheinwerk Verlag GmbH, 2018. ISBN: 9783836245487.
- [2] S. Monk. *Das Action-Buch für Maker – Bewegung, Licht und Sound mit Arduino und Raspberry Pi – Experimente und Projekte*. 1. Auflage. Heidelberg: dpunkt, 2016. ISBN: 978-3-86490-385-4.
- [3] H. Norbert. *HomoFaciens*. URL: <http://www.homofaciens.de/project:ge.htm> (besucht am 20. Mai 2018).

²⁰<https://bw-ju.de/>