



Projekt

## Notizen-T<sub>E</sub>X-Web

**L<sup>A</sup>T<sub>E</sub>X**

von

Jan Unger

4. November 2020



# Inhaltsverzeichnis

<b>1</b>	<b>README</b>	<b>1</b>
1.1	README . . . . .	1
1.2	Kurzbefehle . . . . .	1
1.3	Software . . . . .	2
1.4	Erste Schritte . . . . .	3
1.5	Github-Repository erstellen – klonen . . . . .	4
1.6	Script Beschreibung . . . . .	4
<b>2</b>	<b>Spickzettel-Markdown</b>	<b>7</b>
2.1	Schreiben in Markdown . . . . .	7
2.2	Markdown – Latex – PDF erstellen . . . . .	7
2.3	Quellen . . . . .	7
2.4	Listen . . . . .	8
2.5	Anführungszeichen . . . . .	8
2.6	Grafik – Abbildung . . . . .	8
2.7	Tabelle . . . . .	9
2.8	Mathe . . . . .	10
2.9	Texthervorhebung . . . . .	10
2.10	Code . . . . .	10
2.11	Links . . . . .	11
2.12	Absätze . . . . .	11
	<b>Literatur</b>	<b>11</b>

# 1 README

## 1.1 README

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

## 1.2 Kurzbefehle

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

# Schreiben in Markdown o. Illustrator o. Excel

```
./projekt.sh # Schritt 2, 3, 5
#####
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
#####

# PDF erstellen
make distclean
make
make clean

# Git Version
git add .
git commit -a
git push

# Backup
./projekt.sh # Schritt 9
```

## 1.3 Software

- Git Bash<sup>1</sup>
- Github-Repository klonen<sup>2</sup>
- Texlive (Latex)<sup>3</sup>
- Pandoc (Dokumentenconverter)<sup>4</sup>
- Imagemagick (Bildbearbeitung)<sup>5</sup>
- Editor Visual Studio Code<sup>6</sup>
- Editor Atom<sup>7</sup>
- Editor Notepad++<sup>8</sup>
- TeXstudio (Latexeditor)<sup>9</sup>
- Tablesgenerator (Latex / Markdown)<sup>10</sup>
- hpi-dokumentvorlagen-latex (Hasso-Plattner-Institut (HPI) Potsdam)<sup>11</sup>
- Zotero (Literaturverwaltung)<sup>12</sup>
- Wordpress<sup>13</sup>
- XAMPP Apache + MariaDB + PHP<sup>14</sup>
- Filezilla<sup>15</sup>
- VM VirtualBox<sup>16</sup>
- Ubuntu (Desktop / Server)<sup>17</sup>
- Wordpress-themes<sup>18</sup>
- themecheck (Wordpress-themes)<sup>19</sup>
- ghostscript Z.B eps in pdf<sup>20</sup>

---

<sup>1</sup><https://git-scm.com/downloads>

<sup>2</sup><https://github.com/jul-eu/Notizen-TeX-Web.git>

<sup>3</sup><https://www.tug.org/texlive/>

<sup>4</sup><https://pandoc.org/installing.html>

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

<sup>6</sup><https://code.visualstudio.com/>

<sup>7</sup><https://atom.io/>

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

<sup>9</sup><https://www.texstudio.org/>

<sup>10</sup>[https://www.tablesgenerator.com/latex\\_tables](https://www.tablesgenerator.com/latex_tables)

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

<sup>12</sup><http://www.zotero.org/>

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

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

<sup>15</sup><https://filezilla-project.org/>

<sup>16</sup><https://www.virtualbox.org/>

<sup>17</sup><https://ubuntu.com/download>

<sup>18</sup><https://de.wordpress.org/themes/>

<sup>19</sup><https://themecheck.info/>

<sup>20</sup><https://www.ghostscript.com/>

## 1.4 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=»Notizen-TeX-Web«
  - »/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.eps«

### 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

### Linux-Terminal 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

## 1.5 Github-Repository erstellen – klonen

GitHub's maximum file size of 50 MB

### Repository auf Github erstellen

```
# HTTPS oder SSH
HTTPS: https://github.com/ju1-eu/Notizen-TeX-Web.git
SSH: git@github.com:ju1-eu/Notizen-TeX-Web.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/ju1-eu/Notizen-TeX-Web.git
git push -u origin master
```

### Github-Repository klonen

```
git clone https://github.com/ju1-eu/Notizen-TeX-Web.git
```

## 1.6 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=»Notizen-TeX-Web«
- »/media/jan/usb/backup/notizenUbuntu«
- »/media/jan/virtuell/backup/notizenUbuntu«
- »/media/jan/usb/archiv/notizenUbuntu«
- »/media/jan/virtuell/archiv/notizenUbuntu«



## 2 Spickzettel-Markdown

### 2.1 Schreiben in Markdown

1. Markdown
2. Textauszeichnung – Was ist wichtig?, Tabellen, Bilder, Quellcode, Literatur, Links
3. Rechtschreibprüfung <sup>21</sup>
4. Literatur <sup>22</sup>

### 2.2 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)`

### 2.3 Quellen

Quelle: Monk [2]

Quelle: Norbert [3]

Quelle: Kofler, Zingsheim, Gebeshuber, Widl, Aigner, Hackner, Kania, Kloop und Neugebauer [1]

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

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

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

---

<sup>21</sup><https://languagetoolplus.com/?pk-campaign=addon2-popup-logo>

<sup>22</sup><https://www.zotero.org/user/login>

## 2.4 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

## 2.5 Anführungszeichen

»Anführungszeichen«

"Anführungszeichen"

## 2.6 Grafik – Abbildung

Grafiken-Bsp

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

Abbildung-Bsp

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

2.7 Tabelle

Tabelle-Bsp

Tabelle 2.1

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
```



Abbildung 2.1: Grafiken-Bsp



Abbildung 2.2: Abbildung-Bsp

| 4 | a1 | a2

## 2.8 Mathe

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

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

$$5 \text{ cm}, a \cdot b, \dots, \Omega$$

$$100^\circ\text{C}$$

$$80 \%$$

$$5 \text{ cm}, a \cdot b, \dots, \Omega$$

$$100^\circ\text{C}$$

$$\text{// ACHTUNG: Prozentzeichen macht Probleme in HTML und Latex}$$

$$\text{// Z.B. } 80 \%$$

$$80\% \text{ // in Latex}$$

$$80\% \text{ // in HTML}$$

### Matheumgebung:

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

$\begin{aligned} \end{aligned}$

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

## 2.9 Text hervorhebung

**Fett** oder *Kursiv*

**Fett** oder *Kursiv*

## 2.10 Code

HalloWelt

*hallowelt.c*

`#include <stdio.h>`

`int main(void) {`

`printf("Hallo Welt!\n");`

`return 0;`

`}`

## 2.11 Links

<https://google.de> oder Google

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

Fussnote<sup>23</sup>

Fussnote<sup>[1]</sup>

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

## 2.12 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 - mitnichten! 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.

---

<sup>23</sup><https://bw-ju.de/>

# 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).