

## Selbstreferentieller Markdowntext

**Autor:** Karsten Reincke

**Zweck:** Eine selbstreferentielle Anleitung zum Schreiben von Markdown-Texten im GitHub-Flavor

**Motto:** **Spicken ist in**, Guck Dir einfach ab, was Du brauchst

**Lizenz:** [CC0-Lizenz](#)

Die Idee ist, diese Datei `markdown-selftutorial.md` (`README.md`) in einen Editor (z.B. [VSCODE](#)) zu laden, im Code nachzusehen, wie man etwas *markdownisch* schreibt und im View-Modus (bzw. mittels eines PDF-Konverters) zu betrachten, wie es interpretiert aussehen soll.

### 0.) Inhalt

- 1.) Überschriften
- 2.) Absätze
- 3.) Textstile
- 4.) Listen
- 5.) Zitate
- 6.) Coding
- 7.) Linien
- 8.) Links
- 9.) Tabellen
- A.) Konverter
- B.) Grenzen

### 1.) Überschriften

#### Überschrift 1

#### Überschrift 2

#### Überschrift 3

### 2.) Absätze

Dies ist ein *deutscherister* Absatz, mit *hinreichend Länge* für einen guten Zeilenumbruch - und getestetem zweisprachigem Spellchecking. (Der Check funktioniert, wenn hier *deutscherister* und mit *hinreichend Länge* ‘angemeckert’ werden.)

*This* is an *English written* paragraph, with sufficient length and a fitting line feed - and verified bilingual spell checker. (That checker works, id *This* and *English written* are criticized.)

Und hier ein selbstgesetzter Zeilenumbruch.

Wird durch zwei Blanks am Ende ausgelöst.

Klappt.

### 3.) Textstile

normal text

**bold text**

*italic text*

***bold and italic Text***

Alternative Notation in unsortierter Liste: \* **bold Text** \* *italic Text* \* ***bold and italic text***

### 4.) Listen

Verschachtelte Liste

- Level 1
  - Level 1.A
    - \* Level 1.A.a
  - Level 1.B
- Level 2

Sortierte Liste

1. Level a
  1. Level a.x
  2. Level a.  
Note: 1.1 does not work
2. Level b

### 5.) Zitate

Dies wird als Zitat wiedergegeben. > *Everything* is going according to **plan**.

**Darin sind Formatierungen möglich. Und der Umbruch ist wieder automatisch weich.**

## 6.) Coding

Codeblocks werden mit 4 Blanks ausgezeichnet:

```
1  if (Zeile 1)
2  then
3    while(true) do echo $value; done
4  else
5  fi
```

Inline codes sind über Backticks möglich **AND**. Das Rendern hängt von der Maschine ab.

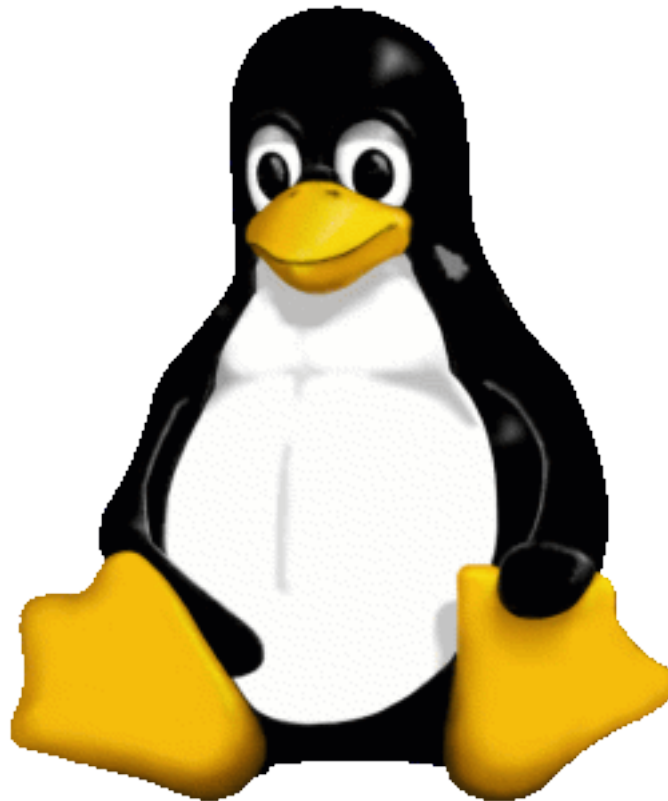
## 7.) Linien

---

Drei \*\*\* oder --- \*\*\*

## 8.) Links

- nach draußen: `fodina.de` = [`fodina.de`] (`http://fodina.de` "Eine Fundgrube")



- Lokale Bilder:
- Anchor Links:
  - Anchor: `<a id="anchor-name"/>`
  - Anchor-Link: `[Link-Text] (#anchor-name)`  
Note: anchor links only work in texts, interpreted by GitHub markdown viewer, not in normal viewers
- Fußnote<sup>1</sup>

Und hier in den Text per html-Tag 'img' eingebundene Bilder<sup>2</sup>: Hallo. Ich bin ein kleiner Blindtext. Und zwar schon so lange ich denken kann. Es war nicht leicht zu verstehen, was es bedeutet, ein blinder Text zu sein: Man ergibt keinen Sinn. Wirklich keinen Sinn. Man wird zusammenhangslos eingeschoben und rumgedreht – und oftmals gar nicht erst gelesen. Aber bin ich allein deshalb ein schlechterer Text als andere? Na gut, ich werde nie in den Bestsellerlisten stehen. Aber andere Texte schaffen das auch nicht. Und darum stört es mich nicht besonders blind zu sein. Und sollten Sie diese Zeilen noch immer lesen, so habe ich als kleiner Blindtext etwas geschafft, wovon all die richtigen und wichtigen Texte meist nur träumen.

Fußnoten:

---

<sup>1</sup>gehen nur im GitHub-Flavor

<sup>2</sup>wird von Pandoc nicht immer korrekt ins PDF übersetzt

## 9.) Tabellen

1	2	3	4	5
a	ab	abc	abcd	abcde
xyz	xy	x	xy	xyz
a	b	c	d	e

## A.) Konverter

Es gibt mehrere Converter, mit denen diese Datei in eine PDF-Datei umgewandelt werden kann. Getestet haben wir die Umwandlung mit

- Visual Studio Code in den folgenden Varianten:
  - VSCODE für [Windows](https://code.visualstudio.com/docs/setup/windows) (<https://code.visualstudio.com/docs/setup/windows>) oder [Linux](https://code.visualstudio.com/docs/?dv=linux64_deb) ([https://code.visualstudio.com/docs/?dv=linux64\\_deb](https://code.visualstudio.com/docs/?dv=linux64_deb))
  - VSCODIUM für [Windows](#) oder [Linux](#)
  - + jeweils den Plugins/Extensions für
    - \* VSCODE erweitert aus [Visual Studio Marketplace](#):
      - [markdown-all-in-one v. yzhang](#)
      - [markdown converter v. manuth](#)
    - \* VSCODIUM :
      - [markdown-all-in-one v. yzhang](#)
      - [markdown converter v. manuth](#)
- PANDOC in den folgenden Varianten:
  - [Windows](#)
  - [Linux](#)

Für die Konvertierung unter Linux mit [PANDOC](#) enthält unser GitHub Repository [proMdTutorials](#) das Script [convert-with-pandoc.sh](#), die Datei [pandoc-header.md](#), in und mit der das Seitenlayout konfiguriert wird, und das 3rd-Party-Template [EISVOGEL v. Pascal Wagler](#). Die Konvertierung nutzt das Kommando:

```
1 pandoc pandoc-header.md markdown-selftutorial.md \  
2 -o markdown-selftutorial.pdf --from markdown \  
3 --template ./3ps/eisvogel/eisvogel.latex \  
4 --listings -V lang=de-DE
```

Das Script `convert-with-pandoc.sh` kann auch für die Konvertierung eigener Markdowndateien verwendet werden. Dazu übergibt man nur den Pfad auf die entsprechende Markdowndatei als Parameter:

```
1 convert-with-pandoc.sh ../dies/ist/meine/Markdowndatei.md
```

## B.) Grenzen

Es gibt einen Github spezifischen "Flavor" von Markdown (GFM bzw. <https://docs.github.com/de/enterprise-cloud@latest/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax>). So verfeinerte Markdown-Dateien werden auf/von GitHub korrekt interpretiert, aber nicht unbedingt von den PDF-Konvertern:

- **Fußnoten:**

- Der GitHub-Interpreter interpretiert die Fußnoten im GitHub-Flavor korrekt.
- Der Preview-Mode von der **VSCODE**-Extension `markdown-all-in-one` interpretiert die erste Fußnote nicht, die nachfolgenden dafür unerwartet.
- Die **VSCODE**-Extension `markdown converter v. manuth` interpretiert die Fußnotenmarkanten nicht.
- Der Konverter `pandoc` mit `Eisvogel`-Template interpretiert die Fußnotenmarkanten nicht.

- **Bilder:**

- *Pandoc* mit *Eisvogel*-Template verschiebt mit `style="float:right"` gesetzte Bilder manchmal unerwartet,

- **Inhaltsverzeichnis:**

- *Markdown All In One* für **VSCODE** bietet die Option, das Inhaltsverzeichnis automatisch zu generieren. Allerdings kommt der Algorithmus gelegentlich durcheinander. Deshalb empfiehlt es sich, ein `<!-- no token -->` vor die Inhaltsverzeichnisliste zu setzen.
- Der Konverter `pandoc` setzt die Einträge eines Inhaltsverzeichnis nur richtig, wenn der Name des Kapitels mit der Kapitelüberschrift - unbesehen der Groß- bzw. Kleinschreibung - übereinstimmt:
  - \* Überschrift: `## B.) <a id="grenzen"></a>Grenzen`
  - \* Inhaltsverzeichnis: `– [B.)Grenzen] (#grenzen)`