
title: Markdown

author: Dr. Daniel Appel

author: Dr. Daniel Appel

Einführung

Was ist Markdown eigentlich?

Auszeichnungssprachen

Vorteile von Auszeichnungssprachen

Spezielle Vorteile von Markdown

Grenzen von Markdown

Beispiel zum Kennenlernen

Wie können wir Markdown einsetzen?

Syntax von Markdown

Hervorhebungen

Überschriften

Listen

Zitate

Code

Code-Blöcke

Inline-Code

Links

Externe Links

Interne Links

E-Mail-Adressen und URLs

Bilder

Trennlinie

Fußnoten

Tabellen

Escape-Sequenz

Einführung

Die folgende Kurzanleitung für Markdown entstand zur Vorbereitung der Projekttag an unserer Schule. Hiermit haben einige Schülerinnen und Schüler in gerade einmal einer Doppelstunde den Umgang mit Markdown erlernt. Zuer Erstellung habe ich selbst Markdown verwendet.

Die Anleitung musste an einigen Stellen angepasst werden, um sie hier online präsentieren zu können. Zusätzlich ist sie hier als PDF erhältlich.

Was ist Markdown eigentlich?

Auszeichnungssprachen

Markdown ist eine sogenannte *Auszeichnungssprache* (englisch: Markup Language). Das berühmteste Beispiel für eine solche Sprache ist HTML, die Sprache mit der Webseiten beschrieben werden.

Ein Text, der mit einer Auszeichnungssprache verfasst wird, besteht letztlich aus normalem Text, in dem einige zusätzliche Angaben gemacht werden. Zum Beispiel werden Überschriften oder besonders wichtige Textpassagen markiert.

Wer mit HTML vertraut ist weiß, dass man dort eine Überschrift z.B. so markieren kann:

```
<h1>
  Das ist eine Überschrift in HTML
</h1>
```

In Markdown werden wir dies so machen:

```
#Das ist eine Überschrift in Markdown
```

Vorteile von Auszeichnungssprachen

Ein Vorteil der Verwendung einer Auszeichnungssprache besteht darin, dass der **Inhalt und die Struktur eines Dokuments vom Design getrennt** werden.

Nun mag man sich fragen, warum dies ein Vorteil ist. Aber vielleicht kennen viele von uns folgende Situation. Wir möchten ein Dokument wie zum Beispiel einen Praktikumsbericht erstellen. Bevor wir aber endlich mit dem Schreiben beginnen, verlieren wir viel Zeit dafür, zu überlegen, welche Schriftart wir wählen, wie groß die Überschriften sein sollen und denken über viele Details nach, die uns vom wesentlichen, nämlich dem Inhalt, fernhalten.

Verwenden wir eine Auszeichnungssprache, kümmern wir uns einzig auf den Inhalt. Das Design kommt später oder kann auch aus einer Vorlage übernommen werden.

Ein weiterer Vorteil besteht in der **Nachhaltigkeit der geleisteten Arbeit**. Unser Dokument ist, wie oben erwähnt, einfach eine Textdatei. Solche Dateien werden wir immer, auch in ferner Zukunft, noch öffnen können. Unsere alten Word 6.0-Dateien eventuell nicht mehr ganz so einfach.

Spezielle Vorteile von Markdown

Markdown ist **sehr einfach zu erlernen**. Das ist für sich natürlich schon ein Vorteil, bringt aber den weiteren Vorteil mit sich, dass Texte, die in Markdown verfasst sind auch sehr **leicht lesbar** sind. Sogar jemand, der Markdown nicht kennt, wird ein Markdown-Dokument schnell verstehen.

Markdown lässt sich mit entsprechenden Werkzeugen in zahlreiche andere Formate exportieren. Umgekehrt, kann man auch viele Dokumente in Markdown importieren. Das kann oftmals die **gemeinsame Arbeit an einem Text erleichtern**. Wenn z.B. ein Windows-LibreOffice-Benutzer mit einem MacOS-Pages-Benutzer zusammenarbeitet, kann es schon mal vorkommen, dass die Formatierungen nicht ganz so aussehen, wie sie sollen. Kommt dann noch ein Verfechter von LaTeX hinzu, muss dieser wahrscheinlich sowieso wieder in die Röhre gucken.

Markdown lässt sich unter Anderem nach HTML exportieren. Das ist prima, wenn man einen Text online veröffentlichen will. Es lässt sich auch ins EPUB-Format exportieren, so dass man damit ein eigenes EBook erstellen kann!

Grenzen von Markdown

Aufgrund der Einfachheit von Markdown, muss man natürlich auch Einschränkungen in Kauf nehmen. Mit HTML & CSS beispielsweise könnten wir eine Website viel mehr nach unseren Wünschen gestalten als mit Markdown. Ein anderes Beispiel ist das Verschmelzen von Zellen in Tabellen. Möchte man so etwas erreichen, muss man ggf. in der exportierten HTML-Datei manuell nachbessern.

Sehr schön zusammengefasst wird dies auf markdown.de wie folgt:

Die Idee hinter Markdown ist es, Text so einfach wie möglich zu lesen, zu schreiben und zu bearbeiten. HTML ist ein *Publikations-Format*; Markdown ist ein *Schreib-Format*. Daher berücksichtigt seine Syntax nur Inhalte, die mit purem Text zu vermitteln sind.

Beispiel zum Kennenlernen

Betrachten wir mit ein paar einfachen Textformatierungen:

Eingabe	Ergebnis
<code>*kursiv*</code>	<i>kursiv</i>
<code>**fett**</code>	fett
<code>***kursiv und fett***</code>	<i>kursiv und fett</i>

Damit können wir zum Beispiel einen Text so formatieren:

Eingabe	Ergebnis
<code>Das hier ist mein *erster* Text, den ich ***ganz alleine*** mit *Markdown* erstellt habe.</code>	Das hier ist mein <i>erster</i> Text, den ich ganz alleine mit Markdown erstellt habe.

Hier sollten wir festhalten, dass die Beschreibungen (fett, kursiv, fett und kursiv) vielleicht nicht ganz korrekt sind. Besser wäre eine solche Beschreibung: `*betont*`, `**wichtig**`, `***sehr wichtig***`. Warum aber ist diese Beschreibung korrekter?

Wer mit HTML und CSS vertraut ist, ahnt vielleicht schon, wie die Antwort lautet: Das Aussehen unseres Texts wird von CSS bestimmt! Standardmäßig führt ein Sternchen tatsächlich zu einem Kursivdruck, aber mit CSS können wir dies auch verändern und stattdessen z.B. betonte Textpassagen oder Wörter farblich hervorheben. Allerdings sollte uns dies beim Schreiben unseres Texts zunächst nicht weiter kümmern, denn wie schon erwähnt, konzentrieren wir uns bei der Verwendung einer Auszeichnungssprache auf den Inhalt.

Wie können wir Markdown einsetzen?

Im Grunde könnten wir sofort mit einem beliebigen Texteditor starten, ein Dokument in Markdown zu erstellen. Allerdings bietet dieser höchstwahrscheinlich nicht die Möglichkeit, den erstellten Text zu exportieren.

Ein sehr professionelles Werkzeug zur Umwandlung des erstellten Texts in viele Formate wie PDF, HTML, DOC, EPUB etc. ist *Pandoc*. Dieses Tool dient einzig und allein der Umwandlung von einem Format in ein anderes. Auch die Umwandlung von anderen Formaten in Markdown ist möglich. Allerdings ist dies ein Kommandozeilenprogramm, was vielen vielleicht (leider) nicht ganz so sympathisch ist.

Es gibt auch die Möglichkeit, einen Editor zu verwenden, der Im- und Exportfunktionen schon mitliefert. Hier ist *Typora* ein schönes Beispiel, unter anderem, da dieser Editor für Windows, Linux und MacOS verfügbar ist. Außerdem liefert er fünf Themes mit. Dies sind vorgefertigte Designvorlagen, mit denen man schöne Ergebnisse erzielen kann.

Syntax von Markdown

Hervorhebungen

Wie weiter oben schon erwähnt, stehen drei Möglichkeiten zur Hervorhebung bereit:

Eingabe	Ergebnis
<code>*kursiv*</code>	<i>kursiv</i>
<code>**fett**</code>	fett
<code>***kursiv und fett***</code>	<i>kursiv und fett</i>

Überschriften

Überschriften markiert man mit #. Es gibt Überschriften 1. bis 6. Grades. Die Anzahl der verwendeten # bestimmt den Grad. Zum Beispiel:

```
# Kapitel 1
## Erste Schritte
Hier machen wir die ersten Schritte.

## Weitere Schritte
Es geht weiter.
```

Die Überschriften dienen auch automatisch zur Gliederung des Dokuments. Dies ist z.B. beim Erstellen eines Inhaltsverzeichnisses sehr komfortabel.

Listen

Wir können Listen **ohne Nummerierung** so erstellen:

- Wir starten mit einem -.
- Ein Aufzählungspunkt erhält ebenfalls ein -.
 - Mit einem Tab am Anfang der Zeile starten wir einen Unterpunkt.
 - Dieser ist eingerückt.
- Das ist also recht simpel.

Das Ergebnis:

- Wir starten mit einem -.
 - Ein Aufzählungspunkt erhält ebenfalls ein -.
 - Mit einem Tab am Anfang der Zeile starten wir einen Unterpunkt.
 - Dieser ist eingerückt.
 - Das ist also recht simpel.
-

Eine Aufzählung **mit Nummerierung** erstellen wir so::

1. Wir starten mit einer Zahl.
1. Ein Aufzählungspunkt erhält ebenfalls eine Zahl.
 3. Die richtige Reihenfolge spielt keine Rolle.
 6. Wir müssen also nicht mitzählen.
4. Das ist also auch recht simpel.

Das Ergebnis:

1. Wir starten mit einer Zahl.
 2. Ein Aufzählungspunkt erhält ebenfalls eine Zahl.
 3. Die richtige Reihenfolge spielt keine Rolle.
 4. Wir müssen also nicht mitzählen.
 3. Das ist also auch recht simpel.
-

Zitate

Zitate markiert man mit > wie man es aus E-Mails kennt.

```
Gleich kommt ein Zitat!  
> Hier steht ein Zitat.  
> Es ist gut als solches erkennbar
```

Hier ist das Ergebnis:

Gleich kommt ein Zitat!

Hier steht ein Zitat. Es ist gut als solches erkennbar

Code

Code-Blöcke

Hier fügen wir als Beispiel einen HTML-Code ein:

```
```HTML
<h1>
 Das ist eine Überschrift in HTML
</h1>
```
```

Das Ergebnis:

```
<h1>
  Das ist eine Überschrift in <strong>HTML</strong>
</h1>
```

Inline-Code

Im Fließtext kann man Code mit einfachen `` markieren:

```
In Java kann man mit `System.out.print(3+2);` ausrechnen, wie viel 3 plus 2 ist.
```

Ergebnis:

In Java kann man mit `System.out.print(3+2);` ausrechnen, wie viel 3 plus 2 ist.

Links

Externe Links

Kommen wir zunächst zur einfachsten Variante:

```
Klicke \[hier\](http://drdanielappel.de), um auf meine Seite zu kommen!
```

Ergebnis:

Klicke [hier](#), um auf meine Seite zu kommen!

Zusätzlich kann man einen **Titel** einfügen. Dieser wird angezeigt, wenn man mit dem Mauszeiger über den Link fährt:

```
Klicke \[hier\](http://drdanielappel.de "Mathe- und Infoseite"), um auf meine Seite zu kommen!
```

Ergebnis:

Klicke [hier](#), um auf meine Seite zu kommen!

Möchte man Links mehrmals verwenden, kann man eine **Referenz** erstellen:

```
Klicke \[hier\][ref1], um auf meine Seite zu kommen!
```

```
\[ref1\]: http://drdanielappel.de "Mathe- und Infoseite"
```

Ergebnis:

Klicke [hier](#), um auf meine Seite zu kommen!

Interne Links

Man kann auch innerhalb eines Dokuments auf eine Überschrift verlinken.

```
Zum Beispiel auf \[die vom Abschnitt zu Codes\](#code).
```

Ergebnis:

Zum Beispiel auf [die vom Abschnitt zu Codes](#).

E-Mail-Adressen und URLs

Eine E-Mail-Adresse oder eine URL können so angegeben werden, dass sie am Rechner auch direkt angeklickt werden können.

```
Schreiben Sie mir unter <test@test.de>!  
Oder besuchen Sie <meineSeite.de/kontake>!
```

Ergebnis:

Schreiben Sie mir unter test@test.de! Oder besuchen Sie <http://meineSeite.de/kontake>!

Bilder

Um Bilder einzufügen, müssen wir den Dateinamen inklusive einem möglichen Pfad angeben. Im Beispiel befindet sich die Bilddatei von der Markdown-Datei gesehen im Unterordner `Grafiken`. Wie in HTML sollte man auch immer einen Alternativtext angeben. Dieser steht in den eckigen Klammern.

```
Hier kommt ein Bild:  
![Logo Analysis](Grafiken/Bild.png)
```

Ergebnis:

Hier kommt ein Bild:
Logo Analysis

Auch Bilder kann man **referenzieren**:

```
Hier kommt ein Bild:  
![Logo Analysis][bild1]  
  
[bild1]: Grafiken/Bild.png
```

Trennlinie

Eine Trennlinie erhalten wir mit drei Unterstrichen `___`. So sieht sie aus:

Fußnoten

```
Hier[^note1] beziehen wir uns auf eine Fußnote.  
  
[^note1]: Das ist sie!
```

Ergebnis:

Hier ¹ beziehen wir uns auf eine Fußnote.

Tabellen

Um die Ausrichtung in den verschiedenen Spalten zu bestimmen, muss man die Doppelpunkte entsprechend setzen:


```
| Spalte 1 | Spalte 2 | Spalte 3 |
| :-----: | :----- | -----: |
| zentriert| links    | rechts   |
```

Ergebnis:

Spalte 1	Spalte 2	Spalte 3
zentriert	links	rechts

Escape-Sequenz

Möchte man ein Zeichen verwenden, dass in Markdown eine syntaktische Bedeutung hat, kann man diese Bedeutung mit einen Backslash \ deaktivieren.

```
*Kursiv* oder nicht \kursiv\*?
```

Ergebnis:

Kursiv oder nicht **kursiv**?

1. Das ist sie![e](#)