Markdown & Pandoc

Michael Kofler

```
Syntax • Optionen, Templates
HTML = LaTeX = EPUB = Mobi
   PDF Folien (Beamer)
    Tools und Techniken
```

Impressum

Markdown & Pandoc, 2. Auflage

Markdown-Syntax. Werkzeuge. LaTeX und HTML. E-Books erstellen. Präsentationen.

© Michael Kofler und ebooks.kofler 2018

Autoren	Michael Kofler (mit einem Beitrag von Axel Dürkop und Tina Ladwig)
Korrektorat	Markus Hinterreither
ISBN PDF	978-3-902643-33-9
ISBN EPUB/Mobi	978-3-902643-34-6
Verlag	ebooks.kofler, Schönbrunngasse 54c, 8010 Graz, Österreich

Die PDF- und EPUB-Ausgabe dieses Buchs können Sie hier kaufen:

https://kofler.info/ebooks/markdown_pandoc

Inhaltsverzeichnis

lm	pres	2	
Vo	Vorwort		12
1	Einf	führung	15
	1.1	Hello World!	15
	1.2	Hintergrund, Geschichte	19
		Das originale Markdown von John Gruber	19
		Markdown-Varianten und -Implementierungen	19
		Markdown-Alternativen	21
2	Wer	rkzeuge	22
	2.1	Pandoc installieren	22
		Pandoc unter Windows installieren	22
		Pandoc unter Linux installieren	23
		Pandoc unter macOS installieren	24
		Pandoc verwenden	25
		Original-Markdown installieren	25
	2.2	Markdown-Editoren	26
		Markdown im Webbrowser	26
		Editoren für Einsteiger	27
		und für Profis	29

3	Die l	Markdown-Syntax	3′
	3.1	Absätze	32
		Textausrichtung ändern	33
		Zeilenumbruch	33
		Linienblöcke (Pandoc)	34
	3.2	Zeichensätze und Sonderzeichen	34
		Sonderzeichen	3
		Programmcode	3'
		Feste Leerzeichen (Pandoc)	3
		Gedankenstriche (Pandoc)	38
		Weiche Trennzeichen (Pandoc)	38
		Anführungszeichen	38
	3.3	Kommentare	39
	3.4	Dokumentstruktur und Überschriften	4(
		Zusatzinformationen in Überschriften (Pandoc)	4
	3.5	Textformatierung (fett, kursiv)	42
		Fetter und kursiver Text	42
		Pandoc-Eigenheiten	43
		Hochstellen, Tiefstellen, Durchstreichen und Kapitälchen (Pandoc)	43
	3.6	Aufzählungen und Listen	44
		Verschachtelte Listen	40
		Nummerierte Listen	4
		Beispiellisten (Pandoc)	48
		Definitionslisten (Pandoc)	49
		Todo-Listen (GFM)	50

3.7	Einrückungen bzw. Zitate	50
3.8	Listings und Programmcode	51
	Kurze Passagen	51
	Mehrzeilige Listings	52
	Listings ohne Einrückungen (Pandoc, GFM)	53
	Syntax-Hervorhebung (Pandoc, GFM)	54
3.9	Links und Querverweise	55
	Einfache Links	55
	Inline-Links	56
	Referenzen	56
	Querverweise innerhalb von Markdown-Dokumenten (Pandoc)	58
3.10	Bilder	59
	Beschriftung von Bildern (Pandoc)	59
	Bildgröße und andere Parameter (Pandoc)	60
3.11	Tabellen	61
	Linientabellen (PHP Markdown Extra, Pandoc, GFM)	62
	Einfache Tabellen (Pandoc)	63
	Tabellen mit mehrzeiligen Einträgen (Pandoc)	64
	Gittertabellen (Pandoc)	65
	Horizontale Linien	66
3.12	Fußnoten	67
	Eingebettete Fußnoten (Pandoc)	67
	Referenzierte Fußnoten (Pandoc)	67
3.13	HTML- und LaTeX-Code	68
	HTML-Code	68

		Markdown-Formatierung innerhalb des HTML-Codes (Pandoc)	70
		LaTeX-Formeln (Pandoc)	70
		Formeln LaTeX-Makros (Pandoc)	72
		LaTeX-Code (Pandoc)	72
		HTML- und LaTeX-Code kombinieren (Pandoc)	73
	3.14	Weitere Pandoc-spezifische Funktionen	74
		Literaturangaben und Literaturverzeichnisse	74
		Dokument-Metadaten	75
		Dokument-Metadaten im YAML-Format	77
		Import anderer Dateien	78
4	Das	Pandoc-Kommando	79
	4.1	Input- und Output-Formate	80
		Einschränkungen	83
		Markdown-Kompatibilität	84
	4.2	Pandoc-Erweiterungen	85
	4.3	Pandoc-Optionen	86
		Allgemeine Optionen	87
		Allgemeine Reader- und Writer-Optionen	88
		Writer-Optionen für HTML	90
		Writer-Optionen für LaTeX und PDF	91
		Writer-Optionen für Folien	91
		Writer-Optionen für E-Books im EPUB-Format	92
	4.4	Templates	92
		Template-Syntax	93
		Template-Beispiel	95

		Eigene Templates	96
	4.5	Filter	97
		AST-Aufbau	98
		Traditionelle JSON-Filter	99
		Lua-Filter (ab Pandoc 2)	101
		Voraussetzungen und Einschränkungen	101
		Fertige Filter	102
		Beispiel 1: Code-Dateien einfügen	102
		Beispiel 2: Abkürzungen typografisch korrekt darstellen	103
5	нтм	IL-Dokumente	105
	5.1	Der Aufbau von HTML-Dokumenten	105
		Eigene CSS- und JavaScript-Dateien	107
		Sonderzeichen und äöüß werden falsch dargestellt	107
	5.2	Die Formatierung von HTML-Dokumenten	108
		Überschriften und Inhaltsverzeichnis	110
		Nummerierte Überschriften	111
		Tabellen	113
		Abbildungen	114
6	E-B	ooks im EPUB- und Mobi-Format	116
	6.1	EPUB	117
		EPUB-Reader und -Viewer	117
		EPUB-Dateien mit Pandoc erzeugen	118
		Metadaten und Cover	118
		Blocksatz und Formatierung	119

		Bilder und Listings	120
		Inhaltsverzeichnis	121
		Mathematische Formeln und HTML-Code	122
		Eigene Schriften	122
		EPUB-Dateien validieren	125
		EPUB-Interna	126
		EPUB-Dateien aus HTML-Dokumenten erzeugen	128
	6.2	Mobi	128
		KindleGen	129
		EPUB zu Mobi konvertieren	130
		HTML zu Mobi konvertieren	130
		Vorschau	134
		Mobi-Interna ansehen	135
		Fiodi interna andenen	133
7	LaTe	eX und PDF	137
7	LaTe 7.1		
7		eX und PDF	137
7		eX und PDF LaTeX- und PDF-Export	137 138
7		eX und PDF LaTeX- und PDF-Export PDF-Engines	137 138 139
7		eX und PDF LaTeX- und PDF-Export PDF-Engines LaTeX-spezifische Pandoc-Optionen	137 138 139 140
7		eX und PDF LaTeX- und PDF-Export PDF-Engines LaTeX-spezifische Pandoc-Optionen Defaultformatierung	137 138 139 140 141
7	7.1	eX und PDF LaTeX- und PDF-Export PDF-Engines LaTeX-spezifische Pandoc-Optionen Defaultformatierung LaTeX-Template und Template-Variablen	137 138 139 140 141 142
7	7.1	eX und PDF LaTeX- und PDF-Export PDF-Engines LaTeX-spezifische Pandoc-Optionen Defaultformatierung LaTeX-Template und Template-Variablen Individuelle Formatierung	137 138 139 140 141 142 146
7	7.1	LaTeX- und PDF-Export PDF-Engines LaTeX-spezifische Pandoc-Optionen Defaultformatierung LaTeX-Template und Template-Variablen Individuelle Formatierung Seitenformat und Seitenlayout	137 138 139 140 141 142 146 147
7	7.1	LaTeX- und PDF-Export PDF-Engines LaTeX-spezifische Pandoc-Optionen Defaultformatierung LaTeX-Template und Template-Variablen Individuelle Formatierung Seitenformat und Seitenlayout Kopf- und Fußzeilen	137 138 139 140 141 142 146 147

		Silbentrennung	150
		Fonts und Schriftgröße	151
		Fließtext	152
		Listen und Aufzählungen	153
		Tabellen	154
		Abbildungen	159
		Einrückungen bzw. Zitate	160
		Listings mit dem fancyvrb-Paket	162
		Listings mit dem listings-Paket	164
		Beispieldateien	167
8	Folie	en	168
	8.1	Grundlagen	168
		PDF oder HTML?	168
		Strukturierung von Dokumenten	169
		Titelseite	171
	8.2	HTML-Folien	171
		Standalone-Folien	173
	8.3	PDF-Folien (Beamer)	173
		Strukturierung	175
		LaTeX-Code	177
		Zweispaltige Folien	177
		Abbildungen	178
		Handouts mit pgfpages	178
		Handouts mit pdfpages	179

9	Fort	geschrittene Arbeitstechniken	180
	9.1	Atom und Pandoc kombinieren	181
		Absatz neu umbrechen	181
		Atom-Paketverwaltung	182
		Markdown-Unterstützung ohne Konfiguration	183
		Markdown-Preview-Enhanced und Pandoc	184
	9.2	Der Satz dieses E-Books	185
		PDF-Version	185
		Pandoc- und LaTeX-Aufruf trennen	185
		Preprocessing	187
		Postprocessing	188
		LaTeX-Dateien	190
		LaTeX-Code in den Markdown-Dateien	190
		Automatischer Aufruf der Scripts	191
		Der Satz von »echten« Büchern	191
	9.3	Docker-Container für Pandoc	192
		Dockerfile	193
		Image erzeugen	194
		Container erzeugen und starten	195
	9.4	Kollaborativ schreiben mit GitLab	196
		Das große Ganze	197
		Bereitstellen des Docker-Images auf Docker Hub	198
		Bei gitlab.com anmelden	199
		GitLab-Repository einrichten	200
		Konfiguration des GitLab-Runners	200

ebooks.kofler	
Jobs und Pipelines	201
Artefakte finden und herunterladen	203
Kollaborativ schreiben	203
Abschließende Betrachtung	205

INHALTSVERZEICHNIS

11

Markdown & Pandoc (2. Aufl.)

Vorwort

Als ich vor mehr als fünf Jahren die erste Auflage dieses E-Books verfasst habe, war die Markdown-Syntax für viele Anwender noch ein wenig exotisch. Das hat sich mittlerweile grundlegend geändert: Blog-Autoren und Software-Entwickler verfassen täglich Webseiten, Bug-Reports und Dokumentationstexte in der Markdown-Syntax. Selbst »gewöhnliche« Anwender stoßen im Internet immer häufiger auf Formulare, die die Markdown-Syntax (zumindest teilweise) erlauben.

Markdown ...

Die Grundidee von Markdown besteht darin, Texte ausschließlich durch Einrückungen bzw. durch »gewöhnliche« Zeichen so zu formatieren. Damit bleibt der Text für das menschliche Auge gut lesbar, gleichzeitig ist die Syntax für den Computer (also für Markdown-Programme wie Pandoc) eindeutig. Ein Beispiel macht das sofort klar:

```
Das ist ein Markdown-Text mit *kursiv* und **fett
hervorgehobenen** Wörtern. Hier beginnt eine Aufzählung:
* Der erste Punkt der Aufzählung.
* Und der zweite Punkt.
```

Im Vergleich zu herkömmlichen Textverarbeitungsprogrammen wie Word hat das Arbeiten mit Markdown-Texten den Vorteil, dass jeder beliebige Texteditor geeignet ist. Da es sich bei Markdown-Dateien um reine Textdateien handelt, können diese unter Linux oder macOS mit Kommandos wie grep, sed, wc verarbeitet und mit Versionsverwaltungssystemen wie SVN oder Git verwaltet werden. Im Klartext bedeutet das: schnelles, effizientes, plattformunabhängiges Arbeiten auch im Team, und das ganz ohne Abstürze.

Markdown bietet sich aber auch für all jene an, die im naturwissenschaftlichen Umfeld Texte verfassen müssen, vom kurzen Paper bis zur Dissertation. In diesem Segment galt (und gilt zum Teil noch immer) LaTeX als das Maß aller Dinge. Allerdings ist die Syntax von LaTeX *viel* komplizierter und unübersichtlicher als die Markdown-Syntax. Außerdem brilliert LaTeX zwar bei der Erzeugung formvollendeter PDF-Dokumente, scheitert aber bei der Umwandlung der Texte in die Formate HTML, EPUB oder Mobi. LaTeX ist also zum zeitgemäßes Publizieren nur mit großen Einschränkungen geeignet.

... und Pandoc

An dieser Stelle kommt *Pandoc* ins Spiel: Die Hauptfunktion dieses Programms besteht darin, Markdown-Dokumente in das HTML-Format umzuwandeln. Pandoc sieht sich aber weniger als Markdown-Programm denn vielmehr als universellen Konverter zwischen allen erdenklichen Textformaten (und zwar in beide Richtungen!). Zu den wichtigen von Pandoc unterstützten Formaten zählen neben Markdown und HTML auch LaTeX, EPUB und Word. Die Dokumentation zählt fast 50 verschiedene Formate auf.

Sie können Pandoc also beispielsweise dazu verwenden, um einen mit Word verfassten Text in das Markdown-Format umzuwandeln und daraus dann eine PDF-Datei und ein E-Book im EPUB-Format zu erstellen. In der Praxis sind derartige Umwandlungen zwischen verschiedenen Formaten allerdings mit vielen Einschränkungen verbunden. Dieses E-Book konzentriert sich deswegen auf die Markdown-Funktionen von Pandoc.

Pandoc unterstützt eine erweiterte Markdown-Syntax, die auch das Hoch- und Tiefstellen von Text, die Gestaltung von Tabellen und Fußnoten sowie das Bilden von Querverweisen erlaubt. Pandoc macht Markdown zu einem universellen Ausgangsformat für nahezu jede Art von Dokument, von Blog-Beiträgen bis zur Diplomarbeit, von Vortragsfolien bis zum kompletten Buch.

Dieses E-Book

Dieses E-Book gibt eine Einführung in die großartige Markdown-Welt und ihre vielfältigen Anwendungsmöglichkeiten. Nach einem kurzen Einführungskapitel werden die folgenden Themen im Detail behandelt:

- Markdown-Tools (Installation, Markdown-Editoren)
- Markdown-Syntax inklusive der Pandoc-spezifischen Erweiterungen
- Pandoc-Optionen und -Anwendungskonzepte

- HTMI -Dokumente erstellen
- E-Books (EPUB und Mobi)
- LaTeX-Export und PDF-Erzeugung
- Vortragsfolien (Slides)
- Docker-Setup und andere fortgeschrittene Arbeitstechniken

Natürlich freue ich mich über jeden Leser, jede Leserin; wenn es Ihnen aber nur darum geht, für Ihr Content Management System (CMS) hin und wieder einen Blog-Beitrag in Markdown-Syntax zu verfassen, dann reicht es vermutlich aus, wenn Sie sich schnell die Markdown-Syntaxzusammenfassung in der Wikipedia durchlesen.

Die eigentliche Zielgruppe dieses E-Books sind fortgeschrittene Benutzer, die längere Texte (Diplomarbeiten, Bücher etc.) schreiben und publizieren möchten, die also z. B. PDFs und EPUBs erzeugen wollen. Bei aller Begeisterung für Markdown und Pandoc muss ich zugeben, dass die Realisierung eigener Layout-Wünsche jedoch mit viel Mühe und Handarbeit verbunden ist. Oftmals sind eigene Header-Dateien (LaTeX) oder gar Scripts erforderlich, die den von Pandoc generierten Code nachbearbeiten. Benutzer mit Programmier- und LaTeX-Erfahrung sind hier klar im Vorteil. Insofern eignet sich die Kombination aus Markdown und Pandoc gut für *technical writing*, wo dieser Hintergrund zumeist gegeben ist.

Für die hier vorliegende zweite Auflage habe ich das E-Book umfassend überarbeitet und im Hinblick auf die Pandoc-Version 2.3 aktualisiert. Naturgemäß ist dieser Text – ebenso wie die meisten meiner im Eigenverlag bzw. im Rheinwerk Verlag erschienen E-Books und Bücher – selbst als Markdown-Dokument entstanden. Die Textdateien wurden mit Pandoc (für die Mobi- und EPUB-Ausgabe) und außerdem mit LaTeX (für die PDF-Ausgabe) verarbeitet.

Wenn Sie selbst HTML-Seiten, Artikel, Folien, Dokumentationen, Bücher oder E-Books effizienter als bisher verfassen möchten – lesen Sie weiter! Lernen Sie die wunderbare Welt von Markdown und Pandoc kennen! Viel Erfolg beim Publizieren mit Markdown und Pandoc wünscht Ihnen

Michael Kofler im Oktober 2018 https://kofler.info

1 Einführung

1.1 Hello World!

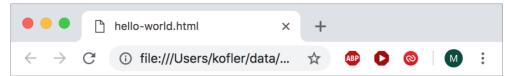
Die folgenden Zeilen zeigen einen einfachen Markdown-Text. Die Formatierung des Texts ist auch im Textmodus offensichtlich. Zur Texteingabe ist jeder beliebige Texteditor geeignet. Besonders komfortabel ist das Verfassen von Markdown-Dokumenten, wenn der Editor die Markdown-Syntax kennt und Textelemente farblich hervorhebt bzw. eventuell gleich eine Vorschau des formatierten Texts ermöglicht.

ebooks.kofler 1.1 Hello World!

Üblicherweise werden Markdown-Dokumente in Dateien mit der Kennung *.md gespeichert. Um den Text nun in das HTML-Format umzuwandeln, benötigen Sie einen Markdown-Konverter, z. B. das Perl-Script markdown oder das Kommando pandoc. Die Installation und Bedienung dieser Werkzeuge ist Thema des nächsten Kapitels. Die resultierende HTML-Datei können Sie nun mit einem beliebigen Webbrowser ansehen (siehe die folgende Abbildung)

Eine individuelle und ansprechendere Darstellung des Texts erreichen Sie, indem Sie in die HTML-Datei eine CSS-Datei (Cascading Stylesheets) einbinden (siehe das Kapitel HTML-Dokumente).

ebooks.kofler 1.1 Hello World!



Die Markdown-Syntax

Markdown-Dokumente sind simple Textdateien. Die Textformatierung erfolgt durch Auszeichnungselemente, die den Textfluss kaum stören. Ein paar Beispiele: Kursiver Text, fetter Text, Text mit Sonderzeichen *in* Listing-Schrift.

Einfache Links werden zwischen < und > gestellt: https://de.wikipedia.org/wiki/Markdown.

Alternativ kann auch zwischen dem dargestellten Text und dem Link differenziert werden: Wikipedia.

Eingerückter Text wird wie in E-Mails mit dem Zeichen > eingeleitet. Die Einrückung kann natürlich über mehrere Zeilen reichen.

Aufzählungen

Nicht nummerierte Aufzählungspunkte werden durch Sterne markiert:

- Der erste Aufzählungspunkt
- Der zweite Aufzählungspunkt
- Der dritte Aufzählungspunkt

Listings/Programmcode

Programmlistings müssen im Markdown-Quelltext um vier Zeichen eingerückt werden. Hier können alle Sonderzeichen verwendet werden. <>(){}*&"'

Abbildung 1.1: HTML-Darstellung des Hello-World-Textes

ebooks.kofler 1.1 Hello World!



Abbildung 1.2: HTML-Darstellung des Hello-World-Textes mit CSS-Formatierung

1.2 Hintergrund, Geschichte

Das originale Markdown von John Gruber

Markdown wurde 2004 von John Gruber mit Unterstützung von Aaron Swartz entwickelt. John Gruber ist in Apple-Kreisen als Betreiber der beliebten Website Daring Fireball bekannt. Auf dieser Website finden Sie auch die Referenz zur originalen Markdown-Syntax sowie den Download-Link eines relativ kleinen Perl-Scripts, das Markdown-Texte in das HTMI -Format übersetzt.

John Gruber sieht Markdown ausschließlich als Werkzeug, um HTML-Seiten bequemer zu verfassen. Aus diesem Grund ist seine Markdown-Syntax auf das absolute Minimum beschränkt; sein Markdown-Konverter sieht einzig eine Übersetzung von Markdown nach HTML vor.

Markdown-Varianten und -Implementierungen

Die Grundidee von Markdown hat viele Web-Autoren und -Entwickler auf Anhieb begeistert. Websites wie Stack Overflow oder GitHub akzeptieren die Markdown-Syntax bei der Eingabe von Texten. Darüber hinaus existieren Markdown-Parser in allen erdenklichen Programmiersprachen und machen es so einfach, Markdown in eigenen Programmen bzw. auf eigenen Websites zu integrieren.

Mit der zunehmenden Verbreitung von Markdown wuchsen freilich auch die Wünsche der Anwender: Das Erzeugen simpler HTML-Seiten war nur der Anfang. Viele Autoren wollen mit Markdown auch Folien für Vorträge gestalten, Handbücher, Hilfetexte, Artikel, wissenschaftliche Arbeiten, Bücher bzw. E-Books verfassen, E-Mails schreiben und ihre Markdown-Texte schließlich in alle anderen erdenklichen Formate umwandeln (z. B. PDF, LaTeX, LibreOffice, Microsoft Word).

Der originalen Markdown-Syntax fehlen dazu aber viele Features: Die Gestaltung von Tabellen, die spezifische Hervorhebung verschiedener Sprachelemente in Listings (Syntax-Highlighting), Querverweise im Text, Fußnoten, mathematische Formeln,

Literaturangaben und -verzeichnisse etc. sind mit dem originalen Markdown nicht bzw. nur mit großen Einschränkungen möglich.

Insofern verwundert es nicht, dass im Verlauf der Zeit diverse Markdown-Dialekte entstanden sind, die zwar auf Grubers Markdown basieren, darüber hinaus aber alle möglichen Zusatzfunktionen realisieren. Die Markdown-Projektseite auf GitHub listet über 50 Markdown-Implementationen auf!

Am weitesten geht in dieser Hinsicht das Programm *Pandoc*, das im Mittelpunkt dieses Buchs steht: Eine stark erweiterte Markdown-Syntax dient hier als Fundament, um Dokumente zwischen allen erdenklichen Formaten zu konvertieren. Zur Zeit gibt es kein anderes Markdown-Programm, das derart viele und weitreichende Anwendungsmöglichkeiten bietet.

https://pandoc.org

Vielleicht haben Sie es schon befürchtet: Auch wenn die originale Markdown-Syntax der unumstrittene gemeinsame Nenner aller Markdown-Varianten ist, sind diverse Erweiterungen unterschiedlich implementiert und führen zu Syntaxinkompatibilitäten. Wenn Sie Markdown-Erweiterungen nutzen möchten, müssen Sie sich daher auf ein bestimmtes Werkzeug festlegen.

Bevor Sie jetzt verzweifelt zu lesen aufhören: So schlimm, wie es hier möglicherweise klingt, ist es nicht. Die Basis ist bei allen Markdown-Dialekten dieselbe. Wenn Sie darüber hinaus Markdown-Erweiterungen nutzen möchten, müssen Sie sich für einen Markdown-Dialekt entscheiden – und dabei hängt es ganz davon ab, was Sie eigentlich mit Markdown machen möchten. Wenn Sie primär kurze Texte schreiben (Blogs, E-Mails), brauchen Sie die meisten Markdown-Erweiterungen gar nicht.

Erst wenn Sie daran denken, umfassende Dokumente mit Markdown zu gestalten (also Bücher, E-Books, technische Dokumentationen), müssen Sie sich für eine Markdown-Variante entscheiden. Für solche Fälle empfehle ich Ihnen Pandoc. Dieses Programm bietet momentan bei weitem die meisten Möglichkeitkeiten; zudem wird Pandoc aktiv weiter entwickelt und gewinnt ständig neue Benutzer.

Markdown-Alternativen

Markdown ist natürlich nicht die einzige textbasierte Syntax, um Texte zu verfassen und dann in andere Formate umzuwandeln. Zu den wichtigsten Alternativen zählen die folgende Formate:

- Wikitext (Link) ist die Syntax der Wikipedia; sie kommt in verschiedenen Varianten in vielen anderen Wiki-Systemen zum Einsatz.
- **AsciiDoc** (Link) ist ein etabliertes Werkzeug zum Verfassen technischer Dokumente, die dann in zahlreiche Formate konvertiert werden können (HTML, PDF, ePub etc.).
- reStructuredText (ReST) (Link) ist Bestandteil der Dokumentationswerkzeuge der Programmiersprache Python. ReST kann aber auch losgelöst von Python verwendet werden, um technische Dokumente zu verfassen und diese dann in diverse Formate zu übersetzen (HTML, LaTeX, XML etc.)

Werkzeuge

Dieses Kapitel beschreibt die Installation von Pandoc bzw. des originalen Markdown-Konverters. Außerdem stelle ich Ihnen einige Editoren mit Markdown-Unterstützung vor.

2.1 Pandoc installieren

Damit Sie Ihre Markdown-Texte in das HTML-Format oder in ein anderes Format umwandeln können, benötigen Sie einen Markdown-Konverter. Ich empfehle Ihnen die Installation von Pandoc, dem zur Zeit vielseitigsten Markdown-Konverter. Alternativ können Sie aber auch das Perl-Script Markdown.p1 von John Gruber oder einen anderen Markdown-Konverter installieren.

Hinweis

Es gibt einige Editoren, die einen Markdown-Konverter gleich mitbringen. Für erste Experimente ist das durchaus bequem. Allerdings legen Sie sich damit auf die vom jeweiligen Editor unterstützte Markdown-Variante fest, die weniger Funktionen bietet als Pandoc.

Pandoc unter Windows installieren

Pandoc steht auf der folgenden GitHub-Seite zum kostenlosen Download als MSI-Datei bereit:

https://github.com/jgm/pandoc/releases

Ein Doppelklick auf die MSI-Datei und eine Bestätigung der Open-Source-Lizenz GPL – mehr ist nicht zu tun, um Pandoc zu installieren. Wenn Sie sich vergewissern möchten,

ebooks.kofler 2.1 Pandoc installieren

dass alles funktioniert hat, öffnen Sie ein Eingabeaufforderungsfenster und führen dort pandoc -v aus.

```
Microsoft Windows [Version 10.0.17134.285]
(c) 2018 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\ms>pandoc -v
pandoc 2.3
Compiled with pandoc-types 1.17.5.1, texmath 0.11.1, skylighting 0.7.2
Default user data directory: C:\Users\ms\AppData\Roaming\pandoc
Copyright (C) 2006-2018 John MacFarlane
Web: http://pandoc.org
This is free software; see the source for copying conditions.
There is no warranty, not even for merchantability or fitness
for a particular purpose.

C:\Users\ms>
```

Abbildung 2.1: Pandoc-Versionsinformationen unter Windows lesen

Pandoc unter Linux installieren

Unter Linux ist die Installation besonders einfach: Die meisten Distributionen liefern standardmäßig entsprechende Pakete mit, die Sie unkompliziert mit dem Paketverwaltungswerkzeugen installieren können. Unter Ubuntu führen Sie beispielsweise einfach apt install pandoc aus.

Wenn Ihre Distribution keine Pandoc-Pakete anbietet oder wenn diese veraltet sind (leider keine Seltenheit), müssen Sie zuerst die exotische Programmiersprache Haskell installieren (unter Ubuntu mit apt install haskell-platform). Haskell verfügt mit dem Kommando cabal über ein eigenes Paketverwaltungssystem. Damit installieren Sie nun die gerade aktuelle Pandoc-Version aus einer Haskell-Paketquelle:

```
cabal update
cabal install pandoc
```

Pandoc wird nun heruntergeladen und kompiliert. Dieser Prozess nimmt einige Minuten in Anspruch. Sollten dabei Fehler auftreten, stellen Sie sicher, dass alle Pakete

ebooks.kofler 2.1 Pandoc installieren

Ihrer Linux-Distribution aktuell sind. Unter Ubuntu führen Sie dazu apt update sowie apt full-upgrade aus. Anschließend versuchen Sie es nochmals. Das eigentliche Pandoc-Kommando wird in das Verzeichnis /home/<username>/.cabal/bin installiert. Damit Sie Pandoc ohne Angabe des Pfads direkt aufrufen können, sollten Sie dieses Verzeichnis Ihrer PATH-Umgebungsvariable hinzufügen. Dazu laden Sie die Datei .bashrc in einen Editor und fügen am Ende der Datei die folgende Zeile hinzu:

```
# am Ende von .bashrc
...
export PATH=$PATH:~/.cabal/bin
```

Nachdem Sie das Terminal geschlossen und neuerlich geöffnet haben, vergewissen Sie sich, dass alles funktioniert hat:

```
pandoc -v
pandoc 2.3
Compiled with pandoc-types 1.17.5.1, ...
Default user data directory: /home/kofler/.pandoc
Copyright (C) 2006-2018 John MacFarlane
```

Tipp

Wenn Sie zu einem späteren Zeitpunkt Pandoc auf eine neue Version aktualisieren möchten, wiederholen Sie einfach die beiden oben angegebenen cabal-Kommandos.

Pandoc unter macOS installieren

Ein fertiges Installationspaket für macOS finden Sie auf der Pandoc-Download-Seite. Die Installation verläuft unkompliziert und schnell. Nach der Installation öffnen Sie ein Terminalfenster und führen dort wie unter Linux das Kommando pandoc -v aus. Pandoc zeigt dann die Versionsnummer an.

Pandoc verwenden

Pandoc enthält keine grafische Benutzeroberfläche und ist somit nach der Installation unsichtbar; es gibt kein neues Icon oder Programm im Startmenü. Pandoc wird ausschließlich als Kommando in einem Eingabeaufforderungsfenster (Windows) bzw. in einem Terminal (Linux, macOS) ausgeführt.

Eine kurze Bedienungsanleitung zu Pandoc liefert man pandoc (nur unter Linux und macOS). Eine lange Liste aller Optionen erhalten Sie mit pandoc --help. Im einfachsten Fall, d. h., wenn Sie eine Markdown-Datei in das HTML-Format konvertieren möchten, rufen Sie pandoc so auf:

```
pandoc -o output.html input.md
```

Wenn Sie weder die Option -o noch eine Markdown-Datei angeben, liest pandoc die zu verarbeitenden Daten aus der Standardeingabe und schreibt das resultierende HTML-Dokument in die Standardausgabe.

Zum Erzeugen von HTML-Dokumenten ist es meist zweckmäßig, auch die Optionen -s sowie -c name.css zu verwenden. Damit erreichen Sie, dass das HTML-Dokument mit den vom HTML-Standard vorgeschriebenen Headern ausgestattet und die angegebene CSS-Datei eingebunden wird. Einen Überblick über weitere Optionen und andere Anwendungsformen des pandoc-Kommandos gibt das Kapitel Das Pandoc-Kommando.

Original-Markdown installieren

Anstelle von Pandoc können Sie auch das Perl-Script Markdown.p1 von John Gruber oder einen anderen Markdown-Konverter installieren. Markdown.p1 ist in einem kleinen ZIP-Archiv enthalten, das Sie von der Website Daring Fireball herunterladen.

Hinweis

Zur Ausführung von Markdown.pl benötigen Sie die Programmiersprache Perl. Der Perl-Interpreter steht unter Linux und macOS standardmäßig zur Verfügung. Sollten Sie unter Windows arbeiten, müssen Sie Perl selbst installieren. Download-Links finden Sie hier.

ebooks.kofler 2.2 Markdown-Editoren

Nach dem Download von Markdown können Sie mit perldoc Markdown.pl die Bedienungsanleitung lesen. Um die Markdown-Datei input.text zu verarbeiten und daraus output.html zu erzeugen, führen Sie Markdown so aus:

Markdown.pl input.text > output.html

2.2 Markdown-Editoren

Grundsätzlich können Sie Markdown-Dokumente mit jedem beliebigen Editor verfassen. Spezielle Markdown-Editoren bzw. -Erweiterungen machen diesen Prozess aber komfortabler, sei es durch Syntax-Highlighting, durch eine Live-Vorschau des Dokuments oder durch integrierte Export-Kommandos.

Diesen Vorteil erkaufen Sie aber in vielen Fällen mit einem wesentlichen Nachteil: Die meisten Markdown-Editoren unterstützen nur einen bestimmten Markdown-Dialekt, dessen Syntax weit weniger reichhaltig ist als bei Pandoc.

Markdown im Webbrowser

Ganz ohne Installation können Sie Markdown in einem Web-basierten Editor ausprobieren und erlernen. Dabei stehen gleich mehrere Angebote zur Auswahl. Empfehlenswert sind die beiden folgenden Seiten:

- https://stackedit.io: StackEdit sieht mit seiner Symbolleiste fast wie ein »echtes« Programm aus (siehe die folgende Abbildung). Es unterstützt einige Markdown-Erweiterungen, z. B. Tabellen, mathematische Formeln und einfache Diagramme.
- https://dillinger.io: Dillinger bietet zwar weniger Funktionen als StackEdit, enthält dafür aber eine praktische eine PDF-Export-Funktion.

2.2 Markdown-Editoren

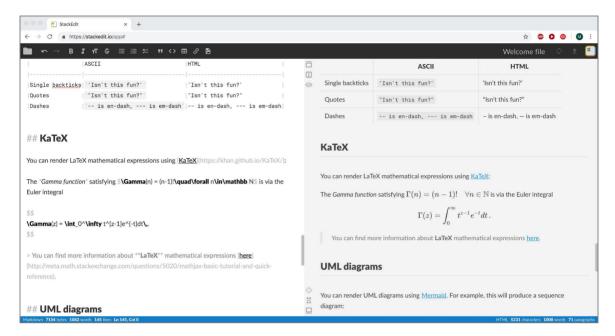


Abbildung 2.2: SlackEdit - eln Markdown-Editor im Webbrowser

Editoren für Einsteiger ...

In den vergangenen Jahren wurden unzählige Programme vorgestellt, die das Verfassen von Markdown-Texten leichter machen sollten. Langfristig durchsetzen konnte sich bisher jedoch keines dieser Programme. Vielfach ist die Entwicklung nach einem vielversprechenden Start schnell wieder eingeschlafen. Aus Platzgründen nenne ich an dieser Stelle nur einige populäre Vertreter.

- Als bekanntester Markdown-Editor unter Windows gilt vermutlich noch immer MarkdownPad. Seine Bekanntheit verdankt dieses Programm seiner Vorreiterrolle.
 Ende 2014 wurde die Entwicklung eingestellt.
- Vielversprechend sieht der plattformunabhängige Editor Haroopad aus (siehe die folgende Abbildung). Allerdings stammt der letzte GitHub-Commit auch in diesem Fall aus dem Jahr 2014.

 Unter dem Namen Typora steht aktuell ein kommerzieller Editor für die Plattformen Windows, macos und Linux in Entwicklung. Als ich dieses E-Book verfasste, befand

sich das Programm noch im Beta-Test und war kostenlos verfügbar.

- Nur unter macOS ist das kommerzielle Programm Marked2 verfügbar. Anders als der Name vermuten lässt, handelt es sich dabei nicht um einen Editor. Marked überwacht lediglich Markdown-Dateien und erzeugt nach jedem Speichervorgang eine neue Vorschau des aktuellen Dokuments. Dieser minimalistische Ansatz erweitert jeden Texteditor um eine Markdown-Vorschau
- Unter Linux k\u00f6nnen die Standardeditoren Gedit und KWrite zumindest rudiment\u00e4r Markdown-Syntaxelemente hervorheben.

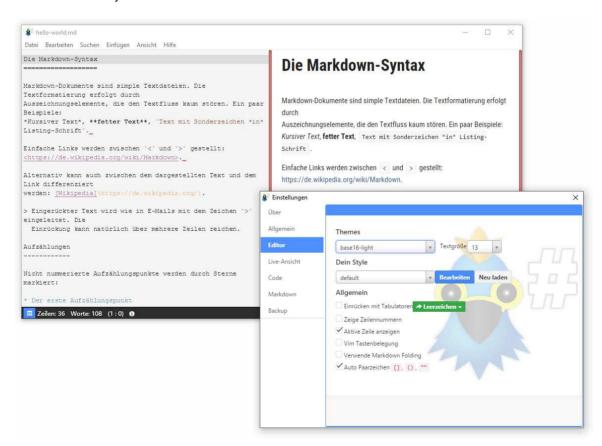


Abbildung 2.3: Haroopad bietet viele Einstellmöglichkeiten und steht für Windows, Linux und macOS zur Verfügung.

... und für Profis

ebooks.kofler

Entwicklern und fortgeschrittenen Anwendern empfehle ich, ganz einfach bei ihren Lieblings-Editoren zu bleiben. Für nahezu alle gängigen Editoren, z. B. für *Atom* und *VS-Code*, aber auch für »Urgesteine« wie *Emacs* oder *Vi* (vim) gibt es Markdown-Erweitungen. Deren Installation ist in der Regel nicht schwierig.

Wenn Sie mit keinem dieser Editoren vertraut sind, sollten Sie sich die Mühe machen, einen derartigen Editor zu erlernen. Das kostet zugegebermaßen anfangs Zeit, lohnt sich aber längerfristig. Heute würde meine Wahl wohl auf *Atom* fallen (https://atom.io). Dieses für Windows, macOS und Linux kostenlos verfügbare Programm ist nahezu uneingeschränkt konfigurier- und erweiterbar und erfüllt nahezu jeden Wunsch. Tipps zur Integration von Atom mit Pandoc finden Sie im Abschnitt Atom und Pandoc kombinieren.

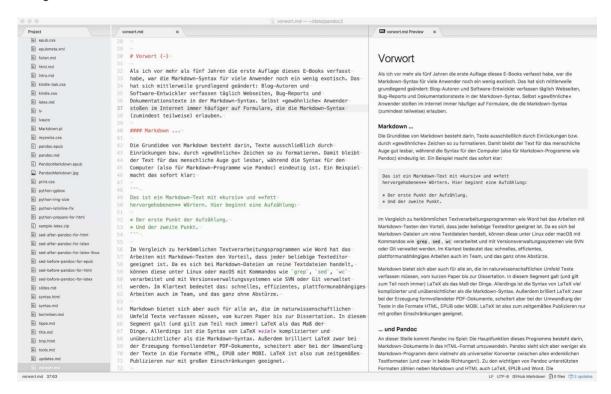


Abbildung 2.4: Mit der Installation weniger Zusatz-Packages wird Atom zum perfekten Markdown-Fditor.

ebooks.kofler 2.2 Markdown-Editoren

Emacs über alles ...

Wenn man sich einmal an einen Editor gewöhnt hat, wechselt man nur ungern sein wichtigstes Schreibwerkzeug. Das erklärt, warum ich dieses E-Book – wie schon zuvor fast 100 andere Bücher zuvor – mit meinem Lieblingseditor Emacs verfasst habe. Für das Syntaxhighlighting habe ich die folgende Markdown-Erweiterung installiert:

https://jblevins.org/projects/markdown-mode

Zur Vorschau des fertigen Markdown-Dokuments verwende ich üblicherweise einen Webbrowser oder PDF-Viewer. Vorher muss ein Markdown-Konverter, in meinem Fall Pandoc, natürlich manuell ausgeführt werden. In meinem Setup kümmert sich darum ein Script, das im Hintergrund läuft und nach jedem Speichervorgang automatisch startet.

3 Die Markdown-Syntax

Dieses Kapitel fasst die Markdown-Syntax zusammen. Syntaxelemente ohne nähere Erläuterungen gelten für alle Markdown-Dialekte, beziehen sich also auf die originale Markdown-Syntax von John Gruber. Daneben gehe ich aber auch auf die Erweiterungen der wichtigsten Markdown-Dialekte ein, wobei der Schwerpunkt bei *Pandoc* und *GitHub Flavored Markdown* (kurz GFM) liegt.

Ich habe mich bemüht, die Markdown-Syntax mit all ihren Erweiterungen in diesem Kapitel möglichst vollständig wiederzugeben. Auf die Beschreibung exotischer Sonderfälle habe ich allerdings verzichtet, damit die Darstellung nicht allzu unübersichtlich wird. In der folgenden Tabelle finden Sie Querverweise auf die jeweilige offizielle Syntaxbeschreibung:

Kürzel	Syntaxbeschreibung
Original	https://daringfireball.net/projects/markdown/syntax
Pandoc	https://pandoc.org
GitHub Flavored Markdown (GFM)	https://github.github.com/gfm
MultiMarkdown	https://github.com/fletcher/MultiMarkdown
Markdown Extra	https://michelf.ca/projects/php-markdown/extra
kramdown	https://kramdown.gettalong.org/

Tabelle 3.1: Die Syntaxbeschreibungen der wichtigsten Markdown-Dialekte

ebooks.kofler 3.1 Absätze

3.1 Absätze

In Markdown werden Absätze und andere Textelemente durch leere Zeilen voneinander getrennt. Zeilenumbrüche spielen hingegen keine Rolle für die Formatierung eines Absatzes. Der Absatz wird automatisch im Webbrowser umbrochen, abhängig davon, wie breit das Browserfenster bzw. die jeweilige Textspalte gerade ist.

Whitespace (also Leer- und Tabulatorzeichen) zwischen Wörtern wird so behandelt, als würde es sich um ein einziges Leerzeichen handeln. Kritischer sind Leerzeichen am Beginn der Zeile, durch die der Text eingerückt wird. Ab vier Zeichen Einrückung interpretiert Markdown den Text als Code-Listing und formatiert ihn anders. Kleinere Einrückungen werden zwar toleriert, sollten aber dennoch vermieden werden.

Das ist ein Absatz. Lange Abstände im Quelltext werden ignoriert. Der Absatz endet mit einer leeren Zeile.

Hier beginnt der nächste Absatz.

Das ist ein Absatz. Lange Abstände im Quelltext werden ignoriert. Der Absatz endet mit einer leeren Zeile.

Hier beginnt der nächste Absatz.

Hinweis

Viele Beispiele in diesem Kapitel folgen dem obigen Muster: Zuerst ist der Markdown-Quellcode abgedruckt, dann folgt eine Box, die zeigt, wie der Text in einem resultierenden HTML- oder PDF-Dokument aussieht

Absatzende je nach Markdown-Dialekt!

Beachten Sie, dass manche, gerade in Blog- oder Content-Management-Systemen weit verbreitete Markdown-Dialekte jeden Zeilenumbruch als Absatzende betrachten und daher mit jeder neuen Zeile einen neuen Absatz beginnen. In diesem Fall müssen Sie Absätze in extrem langen Zeilen formulieren.

ebooks.kofler 3.1 Absätze

Es ist schwer zu begreifen, warum sich die Markdown-Entwickler in einem so elementaren Punkt uneinig sind und vom Standardverhalten abweichen. Bei manchen Markdown-Editoren können Sie mit einer Option zwischen den beiden Absatztrennungsverfahren umschalten. Wenn Sie möchten, dass sich auch Pandoc so verhält, rufen Sie das Kommando mit der Option – hard_line_breaks auf. Weitere Details finden Sie im Abschnitt Pandoc-Kompatibilität.

Textausrichtung ändern

Markdown sieht keine Möglichkeit vor, die Ausrichtung von Text zu beeinflussen (z.B. um einen Text zu zentrieren) oder um Blocksatz ein- und auszuschalten. Natürlich können Sie durch eine CSS-Datei festlegen, dass im resultierenden HTML-Dokument *alle* Absätze im Blocksatz gesetzt werden sollen, aber Sie können im originalen Markdown nicht gezielt einige Absätze so und einige weitere anders formatieren.

Tipp

Wenn Sie als einziges Ausgabeformat HTML wünschen, können Sie den betreffenden Text in <div class="myclass"> verpacken und in der CSS-Datei für myclass jede beliebige Formatierung vornehmen. Diese Vorgehensweise funktioniert aber nicht, wenn Sie Ihr Markdown-Dokument mit Pandoc in andere Formate als HTML umwandeln.

Zeilenumbruch

Wenn Sie innerhalb eines Absatzes einen Zeilenumbruch erzwingen möchten, geben Sie am Ende der betreffenden Zeile zwei oder mehr Leerzeichen an. Beachten Sie aber, dass viele Editoren davon ausgehen, diese Leerzeichen wären überflüssig, und sie aus Ihrem Text wieder eliminieren. Dann scheitert die Kennzeichnung des Zeilenumbruchs. Suchen Sie nach der entsprechenden Option Ihres Editors und deaktivieren Sie diese!

Viele Markdown-Implementierungen sehen zusätzliche Möglichkeiten zur Kennzeichnung fester Zeilenumbrüche vor. In Pandoc gilt: Wenn eine Zeile mit dem Zeichen \ endet, erfolgt an dieser Stelle ein fester Zeilenumbruch.

Linienblöcke (Pandoc)

Pandoc kennt sogenannte Linienblöcke. Dabei wird der Markdown-Quelltext durch eine vertikale Linie am rechten Rand gekennzeichnet. Bei derartig formatiertem Text bleiben Zeilenumbrüche und Einrückungen erhalten. Ansonsten gelten aber die üblichen Markdown-Regeln, etwa für Hervorhebungen. Linienblöcke sind insbesondere für Gedichte oder Verse geeignet.

```
| Ach, was muss man oft von bösen
| Kindern hören oder lesen!
| Wie zum Beispiel hier von diesen,
| welche **Max und Moritz** hießen.
```

Ach, was muss man oft von bösen Kindern hören oder lesen! Wie zum Beispiel hier von diesen, welche **Max und Moritz** hießen.

3.2 Zeichensätze und Sonderzeichen

Markdown schreibt keinen bestimmten Zeichensatz vor. Alle Markdown-Formatierungszeichen wie * oder # sind ASCII-Zeichen und werden daher in den meisten gängigen Zeichensätzen identisch dargestellt. Zeichen mit Codes größer als 127 belässt Markdown, wie sie sind.

Mit anderen Worten: Wenn Sie Ihr Markdown-Dokument im Zeichensatz Unicode (UTF-8) verfassen, verwendet auch das resultierende HTML-Ergebnis diesen Zeichensatz. Sollten Zeichen wie ä, ö, ü oder ß im Webbrowser falsch dargestellt werden, dann fehlt dem HTML-Dokument lediglich ein Header, der den Zeichensatz festschreibt. Details dazu können Sie im Kapitel HTML-Dokumente nachlesen.

Verwenden Sie UTF-8!

Auch wenn die Markdown-Syntax Ihnen in diesem Punkt fast jede Freiheit gibt, empfehle ich Ihnen dringend, den Zeichensatz UTF-8 einzusetzen. Unter Linux und macOS stellt sich diese Frage gar nicht – dort kommt UTF-8 in allen Editoren standardmäßig zum Einsatz.

Anders sieht es unter Windows aus: Manche Windows-Editoren verwenden noch immer den Windows-typischen Zeichensatz CP1252, der weitgehend dem Zeichensatz Latin-1 entspricht. Falls Sie einen Markdown-Text im CP1252-Zeichensatz später in ein anderes Format als HTML umwandeln möchten, werden höchstwahrscheinlich Probleme auftreten. Diese lassen sich durch diverse Zusatzoptionen, Preprocessing etc. in den Griff bekommen – aber wozu die Mühe?

Verwenden Sie UTF-8 und ersparen Sie sich den Ärger! Das gilt insbesondere für Pandoc, das im Gegensatz zur originalen Markdown-Syntax explizit den UTF-8-Zeichensatz vorschreibt!

Sonderzeichen

Sie müssen sich keine Gedanken um die korrekte Behandlung der Zeichen <, > und & machen. Markdown kümmert sich selbst darum und ersetzt diese Zeichen bei der HTML-Konvertierung durch die Codes &1t;, > und &.

Wenn Sie Markdown-Formatierungszeichen wie * direkt im Code verwenden möchten, müssen Sie unter Umständen das Zeichen \ voranstellen – also z. B. *, wenn Sie das Zeichen * am Beginn einer Zeile wünschen, ohne damit einen Aufzählungspunkt einzuleiten. Die folgende Tabelle gibt an, welche Zeichen im originalen Markdown durch \-Codes ausgedrückt (»quotiert«) werden können. Einige Markdown-Dialekte erweitern diese Liste um weitere Zeichen. Noch einfacher ist die Regel bei Pandoc: *Alle* Symbolzeichen dürfen quotiert werden.

Markdown-Code	Ergebnis	Markdown-Code	Ergebnis
\\	\	\(\)	()
/,	•	\[\]	[]
_	_	\{ \}	{}
١.		\!	!
*	*	\+	+
\-	-	\#	#

Tabelle 3.2: Sonderzeichen

```
HTML-Sonderzeichen: a < b, a > b, a & b

Markdown-Sonderzeichen: * _ # +

Quotierte Markdown-Sonderzeichen: \* \_ \# \+

Kursiv: *a+b*. Auch kursiv: *a\*b*. So geht's nicht: *a*b*.
```

HTML-Sonderzeichen: a < b, a > b, a & b

Markdown-Sonderzeichen: * _ # +

Quotierte Markdown-Sonderzeichen: * _ # +

Kursiv: a+b. Auch kursiv: a*b. So geht's nicht: ab*.

Hinweis

Wie die obigen Beispiele zeigen, erkennt Markdown in vielen Fällen aus dem Kontext, dass alleinstehende Zeichen wie * oder # nicht zur Formatierung gedacht sind, und interpretiert Ihre Eingabe auch ohne vorangestelltes \-Zeichen korrekt. Das macht Ihren Markdown-Quelltext besser lesbar, was ja der Intention von Markdown entspricht. Das Voranstellen von \ ist also immer zulässig, aber nur selten notwendig.

ebooks.kofler 3.2 Zeichensätze und Sonderzeichen

Programmcode

Mit `leiten Sie in den Fließtext eingebetteten Programmcode ein. Ein weiteres derartiges Zeichen beendet die Codepassage. Der gesamte darin eingeschlossene Text wird unverändert in Listing-Schrift (z. B. Courier) dargestellt.

```
Kennzeichnung von Programmcode: `if(a*b>c) { run(); }`
```

```
Kennzeichnung von Programmcode: if(a*b>c) { run(); }
```

Damit auch das Zeichen `selbst dargestellt werden kann, muss der entsprechende Code doppelt eingegrenzt werden:

```
blabla ~~
```

```
bla`bla
```

Feste Leerzeichen (Pandoc)

In Pandoc gilt ein quotiertes Leerzeichen (also »\ «) als festes Leerzeichen. Aus $3\ km$ wird also $3\ km$, und Sie können sich sicher sein, dass zwischen $3\ und\ km$ kein Zeilenumbruch erfolgt. Sofern Ihr Editor es unterstützt, können Sie anstelle von »\ « auch das Unicode-Zeichen $160\ (U+00A0)$ eingeben.

Bei der Konvertierung des Markdown-Texts wird aus festen Leerzeichen das Unicode-Zeichen 160 (HTML) bzw. eine Tilde ~ (LaTeX). Beachten Sie, dass viele Programme das Unicode-Zeichen 160 wie ein gewöhnliches Leerzeichen darstellen.

Vorsicht

Die Syntax »\ « ist an sich komfortabel, führt aber nach meinen Erfahrungen oft zu Problemen – und zwar dann, wenn der Editor den Text eines Absatzes neu umbricht. Dann kann es passieren, dass der Backslash am Ende einer Zeile landet und dort als fester Zeilenumbruch interpretiert wird.

Wenn Pandoc mit der Erweiterung smart ausgeführt wird (also mit der Option -- from markdown+smart), fügt es nach manchen Abkürzungen (z. B. Mr.) ein festes Leerzeichen ein. Zufriedenstellend funktioniert dieser Automatismus leider nur für englischsprachige Texte.

Gedankenstriche (Pandoc)

Wenn Pandoc mit der gerade erwähnten Erweiterung smart ausgeführt wird, ersetzt es -- durch einen kurzen und --- durch einen langen Gedankenstrich, also durch - bzw. —. Außerdem wird ... durch das Ellipsis-Zeichen ... ersetzt.

Weiche Trennzeichen (Pandoc)

Für manche Ausgabeformate können Sie Vorschläge für die korrekte Silbentrennung machen. Sofern Ihr Editor dazu in der Lage ist, fügen Sie in das Wort ein weiches Trennzeichen ein ($soft\ hyphen$, Unicode U+00AD). Unter Windows gelingt das zumeist mit der Tastenkombination AltGr + -.

Pandoc belässt das Zeichen unverändert, die meisten HTML-Browser und manche E-Book-Viewer können auch korrekt damit umgehen. Auch aktuelle LaTeX-Versionen interpretieren das Zeichen richtig. (Die Fehlermeldung *unicode char u8 not set up for use with LaTeX* deutet auf eine zu alte LaTeX-Version hin.)

Anführungszeichen

Markdown und Pandoc belassen Anführungszeichen grundsätzlich unverändert. Wenn Sie Pandoc mit der Option -- from markdown+smart ausführen (diese Option kommt auch bei diesem E-Book zum Einsatz), werden zusammengehörende einfache bzw. doppelte Anführungszeichen jeweils nach oben bzw. nach unten gerichtet, wie dies im englischen Sprachraum üblich ist. Zu sehen ist dies im folgenden Beispiel bei den Wörtern amet und sadipscing.

ebooks.kofler 3.3 Kommentare

Wenn Sie wie im folgenden Beispiel bei den Wörtern diam und eirmod die im deutschen Sprachraum gebräuchlichen Anführungszeichen wünschen, müssen Sie die entsprechenden Sonderzeichen im Editor direkt eingeben (siehe auch hier in der Wikipedia).

Zeichen	Unicode	Bedeutung
,	U+201A	einfaches deutsches Apostroph unten
•	U+2018	einfaches deutsches Apostroph oben
,,	U+201E	doppelte deutsche Apostrophe unten
"	U+201C	doppelte deutsche Apostrophe oben

Tabelle 3.3: Deutsche Apostrophe

```
Lorem ipsum »dolor« sit "amet",
consetetur 'sadipscing' elitr,
sed "diam" nonumy 'eirmod' tempor . . .
```

3.3 Kommentare

Markdown sieht keine direkte Möglichkeit vor, Kommentare in den Text zu integrieren – also Text, der im endgültigen Dokument nicht sichtbar ist. Da Markdown aber das Einbetten von HTML-Code in den Text erlaubt (siehe auch den Abschnitt HTML- und LaTeX-Code), können Sie auf HTML-Kommentare zurückgreifen.

```
<!-- Kommentar in einem
Markdown-Dokument -->
```

Vorsicht

Beachten Sie, dass diese Kommentare im resultierenden HTML-Dokument zwar unsichtbar, aber sehr wohl enthalten sind und z.B. im HTML-Quelltext gelesen werden können! Wenn Sie sich Passwörter oder andere vertrauliche Informationen merken müssen, sind HTML-Kommentare also nicht der ideale Ort! Verwenden Sie gegebenenfalls beim Aufruf von Pandoc die Option --strip-comments, um alle Kommentare zu eliminieren.

Falls Sie vorhaben, Ihr Dokument später in ein E-Book im EPUB-Format umzuwandeln, vermeiden Sie mehrere aufeinanderfolgende Minuszeichen innerhalb des Kommentars! EPUB basiert auf XHTML, und die XHTML-Syntax verbietet mehrere aneinandergereihte Minuszeichen in Kommentaren; der Kommentar<!-- Option --abc --> ist somit ein Syntaxfehler und führt dazu, dass die resultierende EPUB-Datei nicht dargestellt werden kann!

3.4 Dokumentstruktur und Überschriften

Es gibt zwei Möglichkeiten, Überschriften zu kennzeichnen.

- #-Syntax: Hier stellen Sie dem Überschriftentext ein bis sechs #-Zeichen voran und kennzeichnen auf diese Weise Überschriften erster bis sechster Ordnung. Optional dürfen nach dem Überschriftentext weitere #-Zeichen folgen; diese haben aber nur kosmetische Funktion und werden von Markdown ignoriert. Deswegen spielt ihre Anzahl auch keine Rolle (d. h., eine Überschrift dritter Ordnung wird mit drei #-Zeichen eingeleitet, kann aber am Ende ohne weiteres sechs #-Zeichen aufweisen).
- Die andere Variante sieht ein Unterstreichen der Überschrift durch die Zeichen = oder vor; sie funktioniert nur für Überschriften erster und zweiter Ordnung. Es ist gleichgültig, wie weit sie die Überschrift unterstreichen. Für die Markdown-Syntax reicht sogar ein einziges Zeichen aus (= oder -).

```
# Kapitelüberschrift (entspricht <h1> in HTML)

## Abschnittsüberschrift (<h2>)

### überschrift dritter Ordnung (<h3>)

#### überschrift vierter Ordnung (<h4>)

##### überschrift fünfter Ordnung (<h5>)

##### überschrift sechster Ordnung (<h6>)
```