

Markdown & Pandoc

Michael Kofler

Markdown-Syntax

HTML- und LaTeX-Export

eBooks erstellen (EPUB, MOBI, PDF)

Werkzeuge ■ Tipps & Tricks

ebooks.kofler

Markdown & Pandoc

Markdown-Syntax. Werkzeuge. LaTeX und HTML.

eBooks im EPUB-, MOBI- und PDF-Format.

Präsentationen (Folien).

© Michael Kofler und ebooks.kofler 2013

Autor	Michael Kofler
Korrektorat	Markus Hinterreither
ISBN	978-3-902643-11-7
Verlag	ebooks.kofler , Schönbrunnngasse 54c, 8010 Graz, Austria

Die PDF-Ausgabe dieses Buchs ist hier erhältlich:

<http://kofler.info/ebooks/markdown-pandoc/>

Viele in diesem eBook genannten Hard- und Software-Bezeichnungen sind geschützte Markennamen.

Dieses eBook wurde mit großer Sorgfalt verfasst. Dennoch sind Fehler nicht ganz auszuschließen. Für allfällige Fehler kann keine Verantwortung oder Haftung übernommen werden. Verbesserungsvorschläge oder Korrekturen sind selbstverständlich willkommen (ebooks@kofler.info). Vielen Dank dafür!

Dieses eBook ist durch das österreichische Urheberrecht geschützt. Sie dürfen das eBook für den persönlichen Gebrauch kopieren und ausdrucken, aber nicht an andere Personen weitergeben, weder in elektronischer noch in anderer Form.

Inhaltsverzeichnis

Vorwort	5
1 Einführung	8
1.1 Hello World!	8
1.2 Hintergrund, Geschichte	12
2 Werkzeuge	15
2.1 Pandoc installieren	15
2.2 Markdown-Editoren	19
3 Die Markdown-Syntax	25
3.1 Absätze	26
3.2 Zeichensätze und Sonderzeichen	28
3.3 Kommentare	31
3.4 Dokumentstruktur und Überschriften	32
3.5 Textformatierung (fett, kursiv ...)	34
3.6 Aufzählungen und Listen	35
3.7 Einrückungen bzw. Zitate	42
3.8 Listings und Programmcode	43
3.9 Links und Querverweise	47
3.10 Bilder	50
3.11 Tabellen	52
3.12 Fußnoten	60
3.13 HTML- und LaTeX-Code	61
3.14 Weitere Funktionen	68

4	Pandoc	73
4.1	Formate	73
4.2	Pandoc-Optionen	78
4.3	Templates	85
5	HTML-Dokumente	90
5.1	Der Aufbau von HTML-Dokumenten	90
5.2	Die Formatierung von HTML-Dokumenten	94
6	EPUB- und MOBI-eBooks	104
6.1	EPUB	105
6.2	MOBI	117
7	LaTeX und PDF	128
7.1	LaTeX- und PDF-Export	129
7.2	Individuelle Formatierung	136
7.3	Die PDF-Version dieses eBooks	153
8	Folien	156
8.1	Grundlagen	156
8.2	HTML-Folien (Slidy, Slideous, DZSlides, S5)	159
8.3	PDF-Folien (Beamer)	164

Vorwort

Das Problem ...

Seit rund zwanzig Jahren verwende ich LaTeX zum Verfassen meiner Bücher. Ich habe LaTeX-Code für rund 20.000 Manuskriptseiten hinter mir! Seit beinahe ebenso langer Zeit verfasse ich Webseiten und Blog-Artikel direkt als HTML-Code, also ohne einen WYSIWYG-Editor. Ich beherrsche somit die wichtigsten Syntaxelemente von LaTeX und HTML im Schlaf.

LaTeX und HTML haben beide einen entscheidenden Vorteil im Vergleich zu Microsoft Word, OpenOffice/LibreOffice, Framemaker und Co.: Die zugrunde liegenden Dateien können mit jedem beliebigen Texteditor verfasst werden. Im Klartext: schnelles, effizientes, plattformunabhängiges Arbeiten ohne Abstürze, auch bei extrem langen Texten.

Diesem Vorteil steht aber ein entscheidender Nachteil gegenüber: LaTeX- und HTML-Quelltexte sind gleichermaßen hässlich, unübersichtlich und selbst mit Syntax-Highlighting schwer zu lesen. Und so habe ich LaTeX und HTML über 20 Jahre professionell, aber ohne echte Begeisterung genutzt. Bis ich vor einigen Jahren mit der Markdown-Syntax und dem darauf aufbauenden Programm Pandoc endlich eine brauchbare Alternative gefunden habe.

... und seine Lösung

Die Markdown-Syntax ermöglicht es, formatierte Texte besonders einfach mit einem Texteditor zu verfassen. Die Textstrukturierung und -formatierung erfolgt so, dass der Text für das menschliche Auge weiterhin gut lesbar bleibt. Markdown-Texte unterscheiden sich in dieser Hinsicht dramatisch von anderen Formaten (HTML, XML, LaTeX), bei denen der Inhalt zwischen unzähligen Formatierungscodes untergeht und der eigentliche Text erst mit einem Browser oder nach einer Umwandlung in ein anderes Format (oft PDF) vernünftig gelesen werden kann.

Ursprünglich war das Markdown-Konzept nur zum Verfassen von HTML-Texten gedacht. Die minimalistische Markdown-Syntax war eine Reaktion darauf, dass einerseits nur wenige Menschen Lust haben, Blogs und andere Artikel direkt in der HTML-Syntax einzugeben; dass aber andererseits die meisten HTML-Editoren nach dem WYSIWYG-Ansatz (*What you see is what you get*) in der Praxis jämmerlich versagen. Einen ganz ähnlichen Weg beschreitet übrigens die Wikipedia, auch wenn die Wiki-Syntax in vielen Details anders aussieht als die Markdown-Syntax.

Nachdem die Markdown-Idee einmal in der Welt war (übrigens schon 2004!), hat es nicht lange gedauert, bis das Konzept auf alle erdenklichen Weisen erweitert wurde. Am weitesten ging dabei John MacFarlane mit Pandoc: Dieses Programm nutzt Markdown als Basis, um Texte in diverse Formate zu konvertieren. Mit Pandoc können Sie nicht nur HTML-Seiten erstellen, sondern auch Folien für Vorträge, Handbücher, Romane, eBooks etc. Bei Bedarf können Sie aus Ihren Markdown-Quellen sogar Word- oder OpenOffice/LibreOffice-Dokumente machen.

Dieses eBook

Dieses eBook 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 aller relevanten Erweiterungen
- Pandoc-Optionen und -Anwendungskonzepte
- HTML-Dokumente erstellen
- EPUB- und MOBI-eBooks
- LaTeX-Export und PDF-Erzeugung
- Vortragsfolien (Slides)

Naturgemäß ist dieser Text – ebenso wie die meisten anderen unter ebooks.kofler erschienenen eBooks – als Markdown-Dokument entstanden. Die EPUB-Ausgabe wurde direkt mit Pandoc erstellt, die MOBI-Variante mit Pandoc und 'kindlegen', die PDF-Fassung mit Pandoc und LaTeX.

Wenn Sie selbst HTML-Seiten, Artikel, Folien, Dokumentationen, Bücher oder eBooks effizienter als bisher verfassen möchten – lesen Sie weiter! Lernen Sie die wunderbare Welt von Markdown und Pandoc kennen! Viel Spaß dabei wünscht Ihnen

Michael Kofler im Mai 2013

<http://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.

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:
`<http://de.wikipedia.org/wiki/Markdown>`.

Alternativ kann auch zwischen dem dargestellten Text und dem Link differenziert werden:
`[Wikipedia] (http://de.wikipedia.org/)`.

> 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. `<>(){}*&"`~'`

Um den Text 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 Abbildung 1.1).



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

Eine individuelle und ansprechendere Darstellung des Texts erreichen Sie, indem Sie in die HTML-Datei eine CSS-Datei (Cascading Stylesheets) einbinden (siehe Abbildung 1.2).

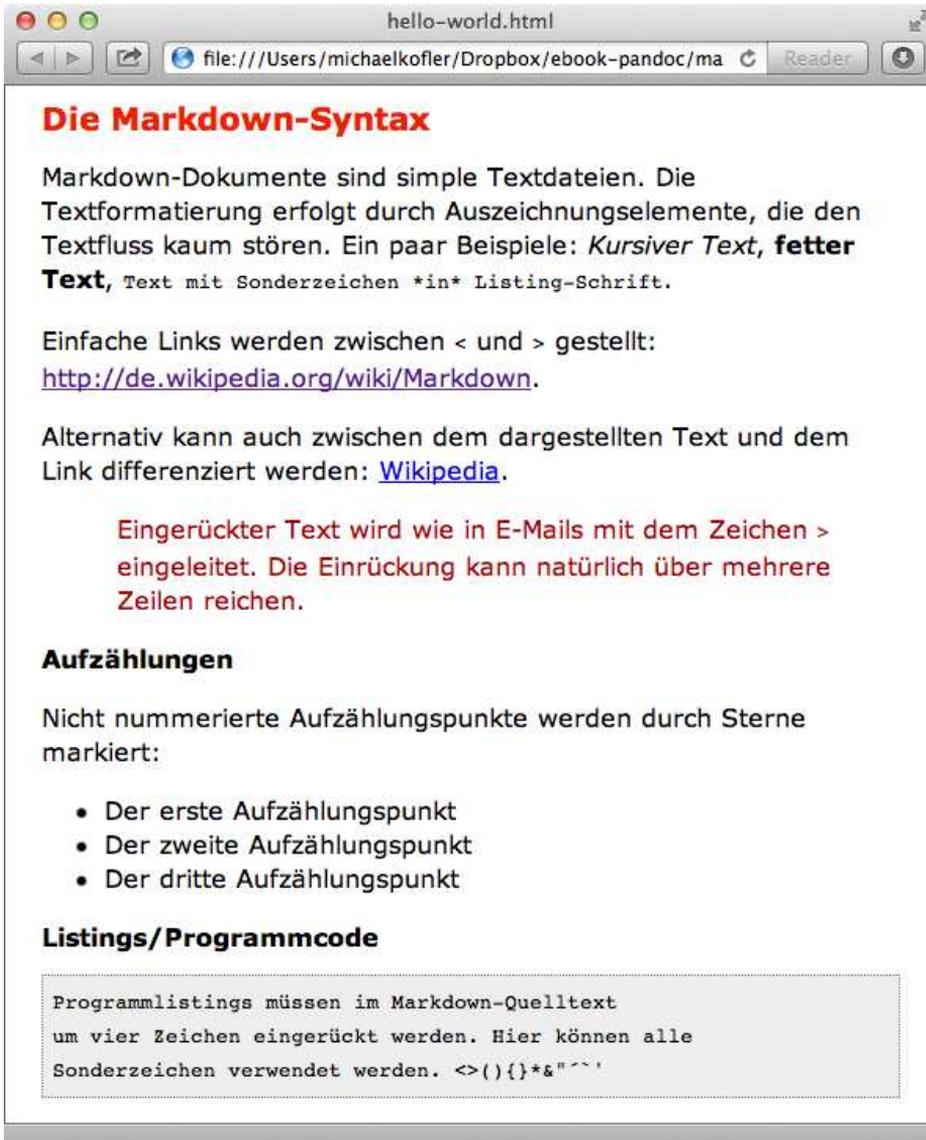


Abbildung 1.2: HTML-Darstellung des Hello-World-Textes mit CSS-Formatierung

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 HTML-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](#), [github](#) oder [Reddit](#) akzeptieren die Markdown-Syntax bei der Eingabe von Texten bzw. Blog-Beiträgen. 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. eBooks verfassen, E-Mails schreiben und ihre Markdown-Texte schließlich in alle anderen erdenklichen Formate umwandeln (z. B. PDF, LaTeX, OpenOffice/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 vergangenen Jahre immer mehr Markdown-Werkzeuge entstanden sind, die zwar auf Grubers Markdown basieren, darüber hinaus aber alle möglichen Zusatzfunktionen realisieren. Am weitesten geht in dieser Hinsicht das Programm Pandoc, das im Mittelpunkt dieses eBooks 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.

- **Pandoc** ([Link](#)) ist ein universelles Konvertierungswerkzeug zwischen Markdown-Texten und zahlreichen anderen Formaten. Pandoc unterstützt neben HTML unter anderem auch PDF, ePub, LaTeX, textile, ReST sowie diverse Office-Formate. Einige Formate können sogar gelesen werden, um beispielsweise ein LaTeX-Dokument in die Markdown-Syntax umzuwandeln.
- **MultiMarkdown** ([Link](#)) hat eine ähnliche Zielsetzung wie Pandoc, unterstützt aber weniger Formate. [Hier](#) ist ein Vergleich zwischen Pandoc und MultiMarkdown.
- **kramdown** ([Link](#)) ist eine Markdown-Implementierung in der Programmiersprache Ruby. Der kramdown-Konverter kann neben HTML- auch LaTeX-Dateien erzeugen.
- **PHP Markdown** bzw. dessen Variante **Markdown Extra** ([Link](#)) ist eine Markdown-Implementierung in der Programmiersprache PHP. PHP Markdown kann auch als Smarty-Plugin in WordPress und in anderen Blog-Systemen genutzt werden.
- **python-markdown2** ([Link](#)) ist eine Markdown-Implementierung in der Programmiersprache Python.
- **Erlmarkdown** ([Link](#)) ist eine Markdown-Implementierung in der Programmiersprache Erlang.
- **Markdent** ([Link](#)) ist eine Markdown-Implementierung in der Programmiersprache Perl.

Das waren die wichtigsten Markdown-Dialekte, aber bei weitem noch nicht alle. Die [Markdown-Projektseite](#) auf GitHub listet über 50 Markdown-Implementationen auf!

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.

Immerhin gibt es [Bemühungen](#), einen gemeinsamen Markdown-Standard zu finden. Wie weit diese fruchten, bleibt abzuwarten. Lesenswert ist in diesem Zusammenhang auch [eine Seite des Pandoc-Entwicklers John MacFarlane](#), auf der dieser die originale Markdown-Syntax kritisiert, weil sie diverse Sonderfälle nicht ausreichend genau festlegt.

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, eBooks, 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 Möglichkeiten; 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-Programmcode verwendet werden, um technische Dokumente zu verfassen und diese dann in diverse Formate zu übersetzen (HTML, LaTeX, XML etc.)

2 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.pl` von John Gruber oder einen anderen Markdown-Konverter installieren.

Hinweis

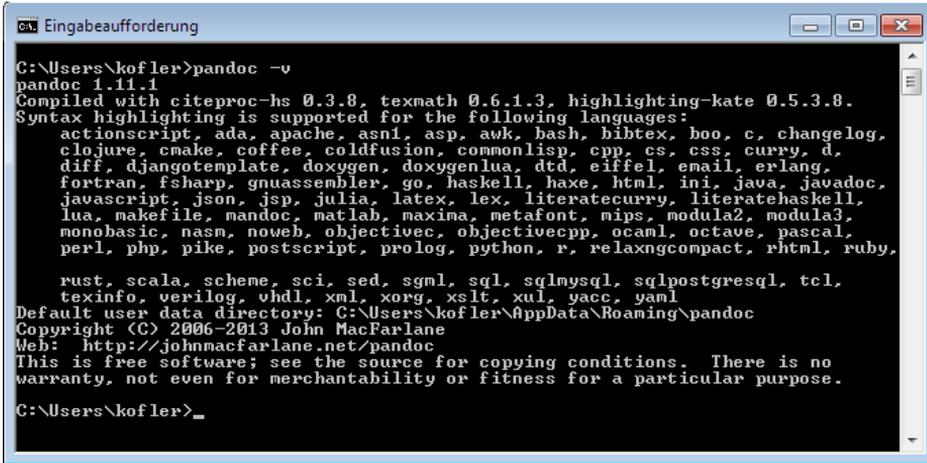
Es gibt einige Editoren, die einen Markdown-Konverter gleich mitbringen. Für erste Experimente ist das durchaus bequem. Allerdings legen Sie sich damit auf die vom jeweiligen Editor unterstützte Markdown-Variante fest, die weniger Funktionen bietet als Pandoc.

Pandoc unter Windows installieren

Pandoc steht auf der Website des Berkeley-Professors John MacFarlane zum kostenlosen Download als MSI-Datei bereit:

<http://johnmacfarlane.net/pandoc/installing.html>

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



```
C:\Users\kofler>pandoc -v
pandoc 1.11.1
Compiled with citeproc-hs 0.3.8, texmath 0.6.1.3, highlighting-kate 0.5.3.8.
Syntax highlighting is supported for the following languages:
actionscript, ada, apache, asnl, asp, awk, bash, bibtex, boo, c, changelog,
clojure, cmake, coffee, coldfusion, commonlisp, cpp, cs, css, curry, d,
diff, djangotemplate, doxygen, doxygenlua, dtd, eiffel, email, erlang,
fortran, fsharp, gnuassembler, go, haskell, haxe, html, ini, java, javadoc,
javascript, json, jsp, julia, latex, lex, literatecurry, literatehaskell,
lua, makefile, mandoc, matlab, maxima, metafont, mips, modula2, modula3,
monobasic, nasm, noweb, objectivec, objectivecpp, ocaml, octave, pascal,
perl, php, pike, postscript, prolog, python, r, relaxngcompact, rhtml, ruby,
rust, scala, scheme, sci, sed, sgml, sql, sqlalchemy, sqlpostgresql, tcl,
texinfo, verilog, vhdl, xml, xorg, xslt, xul, yacc, yaml
Default user data directory: C:\Users\kofler\AppData\Roaming\pandoc
Copyright (C) 2006-2013 John MacFarlane
Web: http://johnmacfarlane.net/pandoc
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\kofler>_
```

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-get 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-get 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 Ihrer Linux-Distribution aktuell sind. Unter Ubuntu führen Sie dazu `apt-get update && apt-get dist-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, vergewissern Sie sich, dass alles funktioniert hat:

```
pandoc -v
pandoc 1.11.1
Compiled with xxx ...
Copyright (c) 2006-2013 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 OS X installieren

Ein fertiges Installationspaket für OS X 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 also kein neues Icon oder Programm im Startmenü. Pandoc wird ausschließlich als Kommando in einem Eingabeaufforderungsfenster (Windows) bzw. in einem Terminal (Linux, OS X) ausgeführt.

Eine kurze Bedienungsanleitung zu Pandoc liefert man `pandoc` (nur unter Linux und OS X). Eine 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.text
```

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.

Häufig ist es zudem zweckmäßig, 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 [Pandoc](#).

Original-Markdown installieren

Anstelle von Pandoc können Sie auch das Perl-Script `Markdown.pl` von John Gruber oder einen anderen Markdown-Konverter installieren. `Markdown.pl` 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 OS X standardmäßig zur Verfügung. Sollten Sie unter Windows arbeiten, müssen Sie Perl selbst installieren. Download-Links finden Sie hier.

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 – und leider in keinem Fall die reichhaltige Pandoc-Syntax!

Hinweis

Dieses eBook habe ich in meinem Lieblingseditor Emacs verfasst. Zur Vorschau des Markdown-Dokuments verwende ich üblicherweise einen Webbrowser. Vorher muss ein Markdown-Konverter, in meinem Fall `pandoc`, natürlich manuell ausgeführt werden.

Windows-Editoren, MarkdownPad

`MarkdownPad` ist der zurzeit populärste Markdown-Editor unter Windows. Es gibt zwei Varianten, eine kostenlose Version und eine Pro-Version um US-\$ 15, die diverse Zusatzfunktionen bietet. Die freie Version kennt ausschließlich die minimale Markdown-Syntax laut John Gruber; die Pro-Version unterstützt immerhin auch die Markdown-Erweiterungen *Flavored Markdown* (GitHub) bzw. *Markdown Extra* (PHP Markdown). Beiden Versionen gemeinsam ist die Live-Vorschau des gerade verfassten Dokuments; sie hilft vor allem Markdown-Einsteigern.

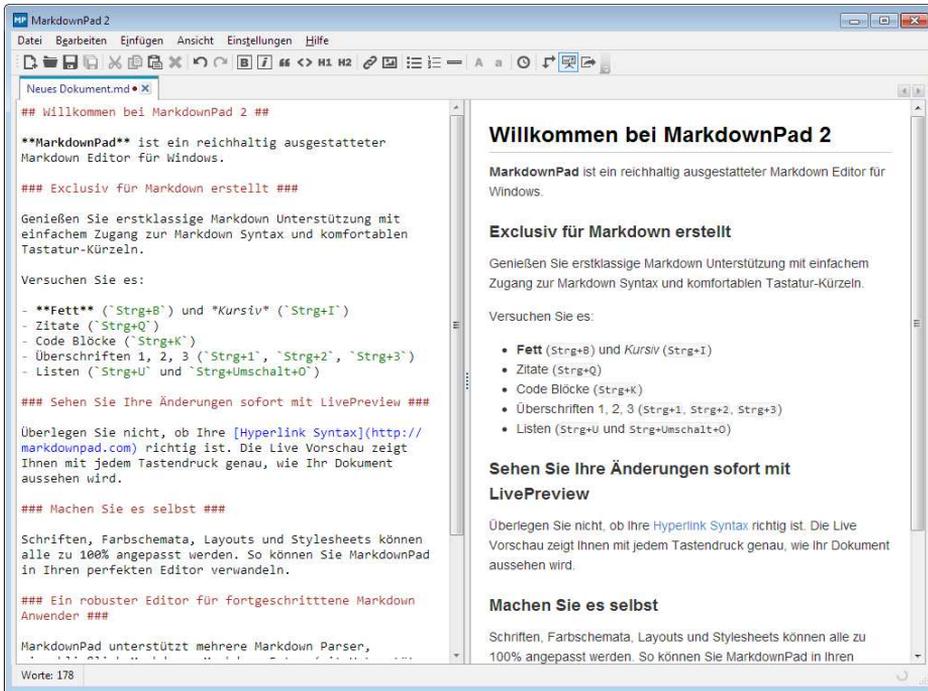


Abbildung 2.2: MarkdownPad

Linux-Editoren, Emacs und Vi

Die Gnome- bzw. KDE-Editoren Gedit und KWrite standardmäßig die Hervorhebung von Markdown-Syntaxelementen, auch wenn beim Syntax-Highlighting mitunter Fehler passieren. Es kann sein, dass die Editoren den Markdown-Modus nicht selbstständig erkennen; dann müssen Sie die Hervorhebung explizit mit *Ansicht/Hervorhebung* bzw. mit *Extras/Hervorhebung* aktivieren.

Eine andere Variante besteht darin, auf die Editor-Urgesteine Emacs oder Vi bzw. vim zurückzugreifen. Für beide Programme existieren Erweiterungen, die die Eingabe und Verarbeitung von Markdown-Text erleichtern und den Text durch Syntax-Highlighting übersichtlicher darstellen. Diese Erweiterungen müssen allerdings manuell installiert werden.

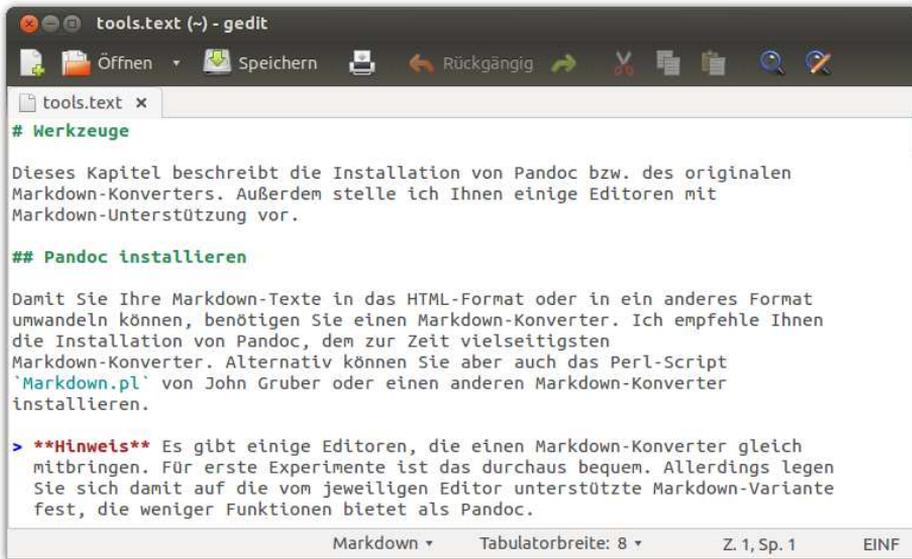


Abbildung 2.3: Gedit unter Ubuntu Linux

- **Emacs Markdown Mode:** Diese Emacs-Erweiterung definiert unzählige Tastenkürzel und führt ein effizientes Syntax-Highlighting durch. Die Erweiterung ist kompatibel zu den wichtigsten Markdown-Erweiterungen.
- **Emacs Pandoc Mode:** Diese Emacs-Erweiterung ist speziell für die erweiterte Markdown-Syntax des Pandoc-Projekts konzipiert, stellt aber weniger Tastenkürzel als der Markdown Mode zur Verfügung.
- **Vim-Markdown:** Diese Vi-Erweiterung kümmert sich nur um die Syntax-Hervorhebung. Allerdings scheint das Projekt verweist zu sein – das letzte Update ist mehr als zwei Jahre alt.

OS-X-Editoren, Marked, Texts und Scrivener

Marked ist ein preisgünstiges OS-X-Programm, das parallel zu einem vorhandenen Editor eingesetzt wird. (Marked stellt selbst keinen Editor zur Verfügung!) Es überwacht Ihre Markdown-Datei und erzeugt nach jedem Speichervorgang eine neue Vorschau des aktu-

ellen Dokuments. Dieser minimalistische Ansatz erweitert jeden Texteditor um eine Markdown-Vorschau. Genial!

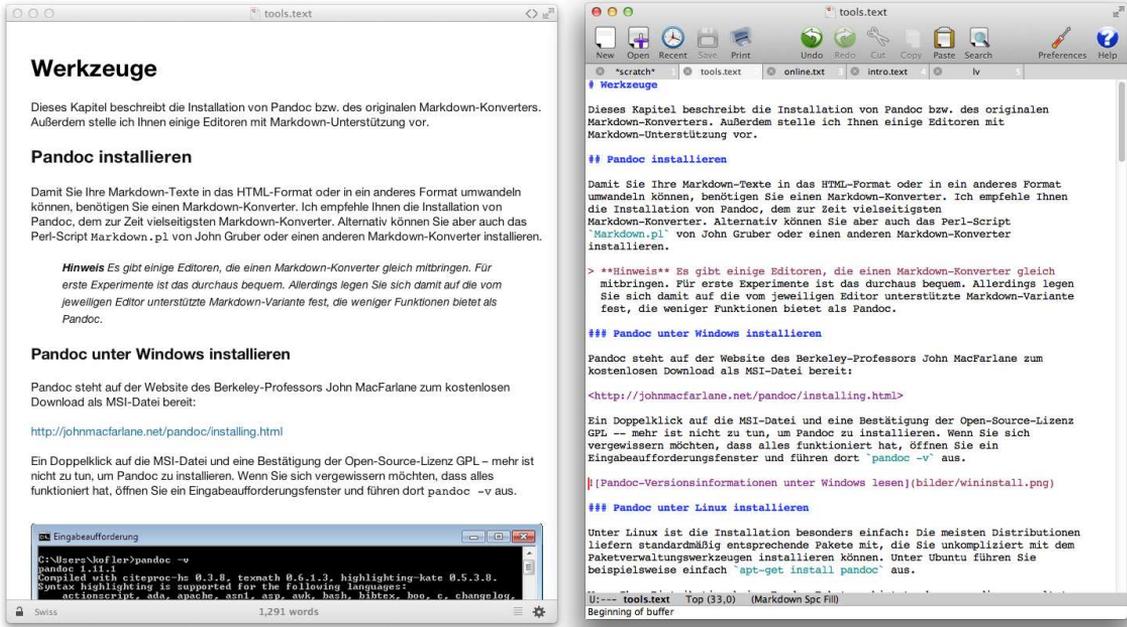


Abbildung 2.4: Links eine Vorschau durch Marked, rechts Aquamacs Emacs unter OS X mit aktivem Markdown-Modus

Das Layout der Vorschau kann durch verschiedene vordefinierte Layouts oder durch eine eigene CSS-Datei eingestellt werden. Standardmäßig verwendet Marked den Markdown-Dialekt MultiMarkdown. Um Marked in Kombination mit Pandoc zu nutzen, müssen Sie im Dialogblatt *Behavior* des Einstellungsdialogs die Option *Custom Markdown Processor* aktivieren und den genauen Ort des pandoc-Kommandos angeben. Grundsätzlich ist jedes Kommando geeignet, das Daten aus der Standardeingabe verarbeitet und das Ergebnis in die Standardausgabe schreibt. Sollte Ihr Markdown-Kommando dazu nicht in der Lage sein, müssen Sie den Aufruf gegebenenfalls in ein kleines Bash-Script verpacken. Nach der erfolgreichen Konfiguration schaltet cmd+alt+C dann zwischen MultiMarkdown und Ihrem eigenen Markdown-Konverter um.



Abbildung 2.5: Pandoc-Konfiguration für Marked

In der Praxis hat sich leider gezeigt, dass Marked bei längeren Dokumenten zunehmend langsamer wird. Beim Verfassen dieses Buchs habe ich letztlich einem gewöhnlichen Webbrowser den Vorzug gegeben, um die HTML-Fassung meines Texts anzusehen.

Einen Überblick über rund 30 weitere OS-X-Programme gibt die Website [appstorm](#) vor. Nicht in der Liste enthalten ist mein persönlicher Lieblingseditor, der auch unter OS X Emacs heißt (siehe die obige Abbildung). Dezierte OS-X-Editoren wie BBEdit oder TextWrangler unterstützen Markdown ganz selbstverständlich und zum Teil schon seit vielen Jahren. Abschließend noch einige Anmerkungen (oder sollte ich vielleicht sagen: Warnungen) zu zwei Programmen, die in jedem Suchergebnis nach *os x markdown editor* ganz oben auftauchen:

- **Scrivener** ist ein kommerzieller WYSIWYG-Texteditor für OS X und Windows mit optionalem MultiMarkdown-Export. Standardmäßig werden die Textdateien aber in einem eigenen XML-Format gespeichert. Die Besonderheit von Scrivener sind diverse Hilfsmittel für den kreativen Schreibprozess. Diese Werkzeuge machen Scrivener zu einem ganz besonderen Programm, aber wenn Sie primär einen Markdown-Editor suchen, ist Scrivener nicht die erste Wahl.

- Auch [Texts](#) ist ein kommerzieller WYSIWYG-Editor für Windows und OS X (weitere Versionen sind in Arbeit), der hinter den Kulissen Markdown und Pandoc verwendet. Ich persönlich habe mich mit dem Programm nicht anfreunden können. Besonders die fehlende Möglichkeit, den Markdown-Quelltext direkt zu bearbeiten, widerspricht der Markdown-Philosophie.

Markdown im Webbrowser und in Mail-Programmen

[Markdown Here](#) ist eine Erweiterung für Firefox, Chrome und Thunderbird. Sie ermöglicht es, innerhalb dieser Programme HTML-Seiten und E-Mails direkt in der Markdown-Syntax zu verfassen. Markdown Here verwendet eine Variante des GitHub-Flavored-Markdown-Dialekts. Das bedeutet insbesondere, dass abweichend zu den meisten anderen Markdown-Dialekten bei jedem Zeilenende innerhalb eines Absatzes ein fester Zeilenumbruch durchgeführt wird. Wenn Sie einen Absatz *ohne* festen Zeilenumbruch verfassen möchten, müssen Sie den gesamten Absatz in *eine* ganz lange Zeile schreiben!

Das kommerzielle OS-X-Programm [MailMate](#) ist ein professioneller Mail-Client für IMAP-Accounts, der beim Verfassen von HTML-E-Mails die Markdown-Syntax unterstützt.

iOS- und Android-Programme

Für iPhone, iPad und Co. gibt es diverse Markdown-Editoren, z. B. [Elements](#), [Writing Kit](#), [Byword](#) oder [iA Writer](#). Eine Websuche nach *ios markdown* liefert zahllose weitere Vertreter. Ganz ähnlich ist die Situation für Android: Es herrscht kein Mangel an Programmen, es ist aber nicht einfach, das optimale Programm für die eigenen Anforderungen zu finden. Ein gemeinsames Merkmal aller iOS- und Android-Apps besteht darin, dass diese zumeist nur grundlegende Markdown-Elemente kennen. Markdown-Erweiterungen werden, wenn überhaupt, nur vereinzelt unterstützt.

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 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	Standard Markdown
Pandoc	Pandoc
MultiMarkdown	MultiMarkdown (PDF-Handbuch)
Markdown Extra	PHP Markdown Extra
kramdown	kramdown
Flavored Markdown	GitHub Flavored Markdown

Tabelle 3.1: Die Syntaxbeschreibungen der wichtigsten Markdown-Dialekte

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 den resultierenden Text in einem HTML- oder PDF-Dokument zeigt.

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.

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:

- Pandoc: Wenn eine Zeile mit dem Zeichen `\` endet, erfolgt an dieser Stelle ein fester Zeilenumbruch.
- kramdoc: Wenn eine Zeile mit einem doppelten Backslash `\\` endet, erfolgt an dieser Stelle ein fester Zeilenumbruch.
- Flavored: Abweichend von fast allen anderen Markdown-Implementierungen gilt für das Github Flavored Markdown das Ende jeder Zeile als fester Zeilenumbruch! Wenn Sie einen Absatz *ohne* festen Zeilenumbruch verfassen möchten, müssen Sie also ganz lange Zeilen schreiben! (In Pandoc kann dieses Verhalten optional durch die Erweiterung `hard_line_breaks` aktiviert werden – siehe auch den Abschnitt [Pandoc-Kompatibilität](#).)

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](#) nachlesen.

Tipp

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

Anders sieht es unter Windows aus: Viele 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 `<`, `>` 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
<code>\\</code>	<code>\</code>	<code>\(\)</code>	<code>()</code>
<code>\'</code>	<code>'</code>	<code>\[\]</code>	<code>[]</code>
<code>_</code>	<code>_</code>	<code>\{ \}</code>	<code>{}</code>
<code>\.</code>	<code>.</code>	<code>\!</code>	<code>!</code>
<code>*</code>	<code>*</code>	<code>\+</code>	<code>+</code>
<code>\-</code>	<code>-</code>	<code>\#</code>	<code>#</code>

Alternativ können Sie Text auch in schräge Anführungszeichen (*Backticks*) stellen (``a*b*c`` liefert `a*b*c`). ``` ist insbesondere zur Kennzeichnung von Programmcode gedacht, weswegen in solchen Fällen eine Listingschrift verwendet wird (`<code>` in HTML).

```
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**.

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

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**.

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

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.

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. Bei der Konvertierung des Markdown-Texts in die Formate HTML, XML und LaTeX wird daraus `\ `, `\ ` bzw. `~`.

Achtung

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.