

# Markdown & Pandoc

Michael Kofler

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

2. Auflage

ebooks.kofler

# Impressum

**Markdown & Pandoc**, 2. Auflage

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

© Michael Kofler und ebooks.kofler 2018

<b>Autoren</b>	Michael Kofler (mit einem Beitrag von Axel Dürkop und Tina Ladwig)
<b>Korrektorat</b>	Markus Hinterreither
<b>ISBN PDF</b>	978-3-902643-33-9
<b>ISBN EPUB/Mobi</b>	978-3-902643-34-6
<b>Verlag</b>	<a href="https://ebooks.kofler.info">ebooks.kofler</a> , Schönbrunnngasse 54c, 8010 Graz, Österreich

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

[https://kofler.info/ebooks/markdown\\_pandoc](https://kofler.info/ebooks/markdown_pandoc)

# Inhaltsverzeichnis

<b>Impressum</b>	<b>2</b>
<b>Vorwort</b>	<b>4</b>
<b>1 Einführung</b>	<b>7</b>
1.1 Hello World!	7
1.2 Hintergrund, Geschichte	11
Das originale Markdown von John Gruber	11
Markdown-Varianten und -Implementierungen	11
Markdown-Alternativen	13
<b>2 Werkzeuge</b>	<b>14</b>
2.1 Pandoc installieren	14
Pandoc unter Windows installieren	14
Pandoc unter Linux installieren	15
Pandoc unter macOS installieren	16
Pandoc verwenden	17
Original-Markdown installieren	17
2.2 Markdown-Editoren	18
Markdown im Webbrowser	18
Editoren für Einsteiger ...	19
... und für Profis	21

<b>3 Die Markdown-Syntax</b>	<b>23</b>
3.1 Absätze	24
Textausrichtung ändern	25
Zeilenumbruch	25
Linienblöcke (Pandoc)	26
3.2 Zeichensätze und Sonderzeichen	26
Sonderzeichen	27
Programmcode	29
Feste Leerzeichen (Pandoc)	29
Gedankenstriche (Pandoc)	30
Weiche Trennzeichen (Pandoc)	30
Anführungszeichen	30
3.3 Kommentare	31
3.4 Dokumentstruktur und Überschriften	32
Zusatzinformationen in Überschriften (Pandoc)	33
3.5 Textformatierung (fett, kursiv ...)	34
Fetter und kursiver Text	34
Pandoc-Eigenheiten	35
Hochstellen, Tiefstellen, Durchstreichen und Kapitälchen (Pandoc)	35
3.6 Aufzählungen und Listen	36
Verschachtelte Listen	38
Nummerierte Listen	39
Beispiellisten (Pandoc)	40
Definitionslisten (Pandoc)	41
Todo-Listen (GFM)	42

3.7	Einrückungen bzw. Zitate	42
3.8	Listings und Programmcode	43
	Kurze Passagen	43
	Mehrzeilige Listings	44
	Listings ohne Einrückungen (Pandoc, GFM)	45
	Syntax-Hervorhebung (Pandoc, GFM)	46
3.9	Links und Querverweise	47
	Einfache Links	47
	Inline-Links	48
	Referenzen	48
	Querverweise innerhalb von Markdown-Dokumenten (Pandoc)	50
3.10	Bilder	51
	Beschriftung von Bildern (Pandoc)	51
	Bildgröße und andere Parameter (Pandoc)	52
3.11	Tabellen	53
	Linientabellen (PHP Markdown Extra, Pandoc, GFM)	54
	Einfache Tabellen (Pandoc)	55
	Tabellen mit mehrzeiligen Einträgen (Pandoc)	56
	Gittertabellen (Pandoc)	57
	Horizontale Linien	58
3.12	Fußnoten	59
	Eingebettete Fußnoten (Pandoc)	59
	Referenzierte Fußnoten (Pandoc)	59
3.13	HTML- und LaTeX-Code	60
	HTML-Code	60

Markdown-Formatierung innerhalb des HTML-Codes (Pandoc)	62
LaTeX-Formeln (Pandoc)	62
Formeln LaTeX-Makros (Pandoc)	64
LaTeX-Code (Pandoc)	64
HTML- und LaTeX-Code kombinieren (Pandoc)	65
3.14 Weitere Pandoc-spezifische Funktionen	66
Literaturangaben und Literaturverzeichnisse	66
Dokument-Metadaten	67
Dokument-Metadaten im YAML-Format	69
Import anderer Dateien	70
<b>4 Das Pandoc-Kommando</b>	<b>71</b>
4.1 Input- und Output-Formate	72
Einschränkungen	75
Markdown-Kompatibilität	76
4.2 Pandoc-Erweiterungen	77
4.3 Pandoc-Optionen	78
Allgemeine Optionen	79
Allgemeine Reader- und Writer-Optionen	80
Writer-Optionen für HTML	82
Writer-Optionen für LaTeX und PDF	83
Writer-Optionen für Folien	83
Writer-Optionen für E-Books im EPUB-Format	84
4.4 Templates	84
Template-Syntax	85
Template-Beispiel	87

Eigene Templates	88
4.5 Filter	89
AST-Aufbau	90
Traditionelle JSON-Filter	91
Lua-Filter (ab Pandoc 2)	93
Voraussetzungen und Einschränkungen	93
Fertige Filter	94
Beispiel 1: Code-Dateien einfügen	94
Beispiel 2: Abkürzungen typografisch korrekt darstellen	95
<b>5 HTML-Dokumente</b>	<b>97</b>
5.1 Der Aufbau von HTML-Dokumenten	97
Eigene CSS- und JavaScript-Dateien	99
Sonderzeichen und äöüß werden falsch dargestellt	99
5.2 Die Formatierung von HTML-Dokumenten	100
Überschriften und Inhaltsverzeichnis	102
Nummerierte Überschriften	103
Tabellen	105
Abbildungen	106
<b>6 E-Books im EPUB- und Mobi-Format</b>	<b>108</b>
6.1 EPUB	109
EPUB-Reader und -Viewer	109
EPUB-Dateien mit Pandoc erzeugen	110
Metadaten und Cover	110
Blocksatz und Formatierung	111

Bilder und Listings	112
Inhaltsverzeichnis	113
Mathematische Formeln und HTML-Code	114
Eigene Schriften	114
EPUB-Dateien validieren	117
EPUB-Interna	118
EPUB-Dateien aus HTML-Dokumenten erzeugen	120
6.2 Mobi	120
KindleGen	121
EPUB zu Mobi konvertieren	122
HTML zu Mobi konvertieren	122
Vorschau	126
Mobi-Interna ansehen	127
<b>7 LaTeX und PDF</b>	<b>129</b>
7.1 LaTeX- und PDF-Export	130
PDF-Engines	131
LaTeX-spezifische Pandoc-Optionen	132
Defaultformatierung	133
LaTeX-Template und Template-Variablen	134
7.2 Individuelle Formatierung	138
Seitenformat und Seitenlayout	139
Kopf- und Fußzeilen	140
Überschriften	140
Sprache	142



Silbentrennung	142
Fonts und Schriftgröße	143
Fließtext	144
Listen und Aufzählungen	145
Tabellen	146
Abbildungen	151
Einrückungen bzw. Zitate	152
Listings mit dem fancyvrb-Paket	154
Listings mit dem listings-Paket	156
Beispieldateien	159
<b>8 Folien</b>	<b>160</b>
8.1 Grundlagen	160
PDF oder HTML?	160
Strukturierung von Dokumenten	161
Titelseite	163
8.2 HTML-Folien	163
Standalone-Folien	165
8.3 PDF-Folien (Beamer)	165
Strukturierung	167
LaTeX-Code	169
Zweispaltige Folien	169
Abbildungen	170
Handouts mit pgfpages	170
Handouts mit pdfpages	171

<b>9 Fortgeschrittene Arbeitstechniken</b>	<b>172</b>
9.1 Atom und Pandoc kombinieren	173
Absatz neu umbrechen	173
Atom-Paketverwaltung	174
Markdown-Unterstützung ohne Konfiguration	175
Markdown-Preview-Enhanced und Pandoc	176
9.2 Der Satz dieses E-Books	177
PDF-Version	177
Pandoc- und LaTeX-Aufruf trennen	177
Preprocessing	179
Postprocessing	180
LaTeX-Dateien	182
LaTeX-Code in den Markdown-Dateien	182
Automatischer Aufruf der Scripts	183
Der Satz von »echten« Büchern	183
9.3 Docker-Container für Pandoc	184
Dockerfile	185
Image erzeugen	186
Container erzeugen und starten	187
9.4 Kollaborativ schreiben mit GitLab	188
Das große Ganze	189
Bereitstellen des Docker-Images auf Docker Hub	190
Bei gitlab.com anmelden	191
GitLab-Repository einrichten	192
Konfiguration des GitLab-Runners	192

Jobs und Pipelines	193
Artefakte finden und herunterladen	195
Kollaborativ schreiben	195
Abschließende Betrachtung	197

# 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 zeitgemäßen 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

- HTML-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 erschienenen 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.

```
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](https://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  
    fast alle Sonderzeichen verwendet werden. <>(){}*&"'
```

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](#)).





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

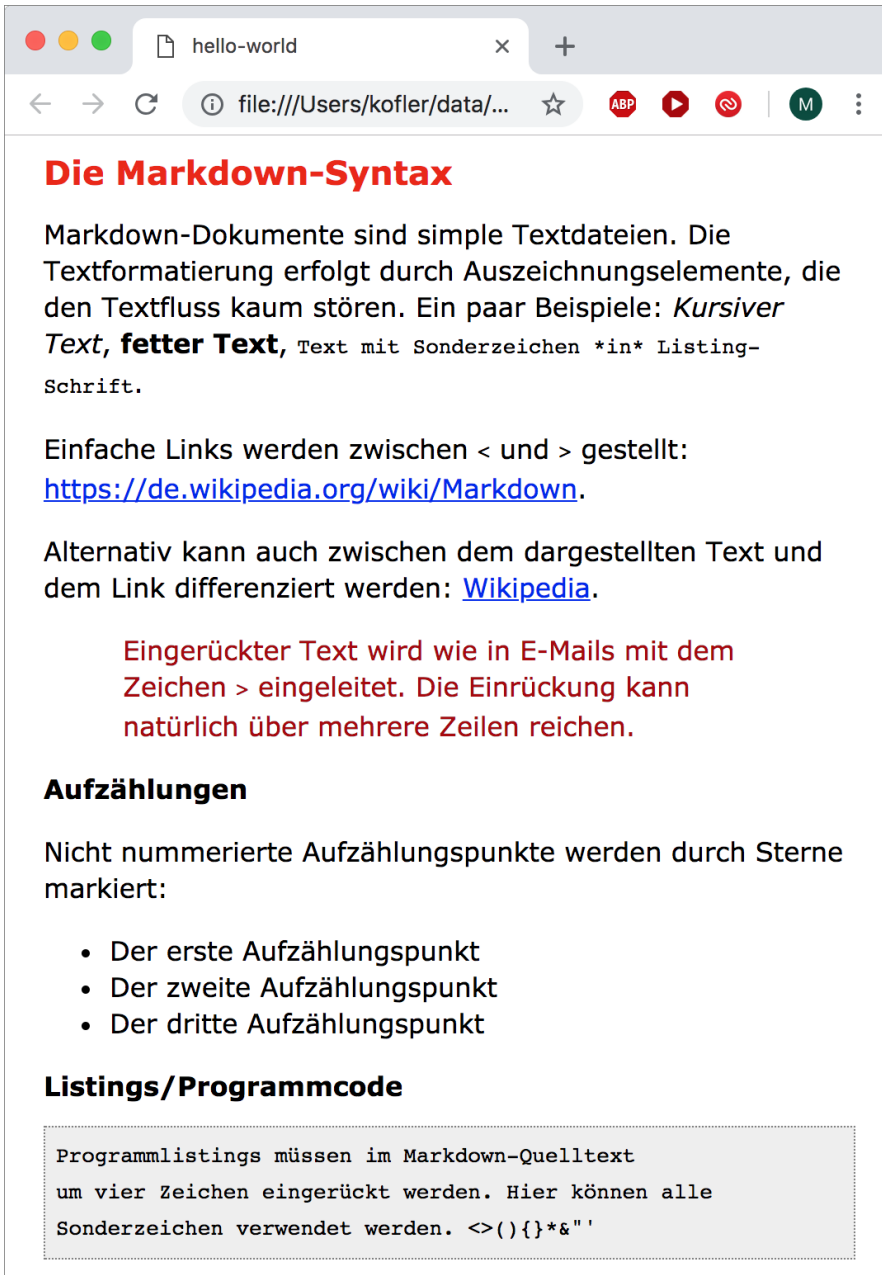


Abbildung 1.2: HTML-Darstellung des Hello-World-Textes mit CSS-Formatierung

## 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](#) 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 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 verwendet werden, um technische Dokumente zu verfassen und diese dann in diverse Formate zu übersetzen (HTML, LaTeX, XML etc.)