

DIE VOLLSTÄNDIGE MARKDOWN-REFERENZ

Markdown Cheat Sheet

Jede Syntax mit Beispielen • 2026

Jede Funktion von CommonMark und GitHub Flavored Markdown, mit einem Beispiel zum Kopieren, von Überschriften und Tabellen bis zu Code, Mathematik und Diagrammen.

[Markdific herunterladen](#)

Das vollständigste Markdown Cheat Sheet im Internet. Jede Funktion von CommonMark und GitHub Flavored Markdown, mit einem funktionierenden Beispiel für jede.

Setzen Sie ein Lesezeichen für diese Seite. Oder holen Sie sich die kostenlose Version als [Markdown Cheat Sheet PDF](#), um sie offline griffbereit zu haben.

Markdown–Syntax auf einen Blick

Element	Syntax
Überschrift	<code># H1</code> bis <code>##### H6</code>
Fett	<code>**fett**</code>
Kursiv	<code>*kursiv*</code>
Durchgestrichen	<code>~durchgestrichen~</code>
Link	<code>[Text](https://example.com)</code>
Bild	<code>![alt](image.jpg)</code>
Inline-Code	<code>`code`</code>
Codeblock	drei Backticks, Ihr Code, dann drei Backticks
Blockquote	<code>> zitierter Text</code>
Ungeordnete Liste	<code>- Element</code>
Nummerierte Liste	<code>1. Element</code>
Aufgabenliste	<code>- [] zu erledigen</code> und <code>- [x] erledigt</code>
Tabelle	<code> A B </code> mit <code> --- --- </code> darunter
Horizontale Linie	<code>---</code>

Was ist Markdown?

Markdown ist eine Formatierungssyntax im Klartext, die 2004 von John Gruber und Aaron Swartz entwickelt wurde. Sie verwendet einfache Symbole, um Text auszuzeichnen, sodass er in HTML und andere Formate umgewandelt werden kann. Das Ziel war Lesbarkeit. Eine Markdown-Datei sollte als Klartext sauber aussehen und nur beim Rendern hübscher werden.

Heute ist Markdown überall. GitHub-READMEs, Reddit-Beiträge, Discord-Nachrichten, Notion-Seiten, Obsidian-Notizen, Antworten von KI-Chatbots und die meisten Dokumentationsseiten verwenden es. Wenn Sie 2026 irgendetwas an einem Computer schreiben, produzieren Sie mit ziemlicher Sicherheit Markdown, ob Sie es merken oder nicht.

Mehr zum Format selbst erfahren Sie unter [was eine .md-Datei ist, wie man eine .md-Datei öffnet und liest](#) oder [Markdown vs HTML](#), um zu verstehen, wie sich die beiden Formate unterscheiden. Eine vollständige Referenz Element für Element finden Sie in der [Markdown-Dokumentation](#).

Überschriften

Verwenden Sie `#`-Symbole am Anfang einer Zeile, um Überschriften zu erstellen. Die Anzahl der Rauten legt die Überschriftsebene fest, von H1 (größte) bis H6 (kleinste).

```
# Überschrift 1
## Überschrift 2
### Überschrift 3
#### Überschrift 4
##### Überschrift 5
##### Überschrift 6
```

Alternative Überschriften-Syntax (Setext)

Nur für H1 und H2 können Sie Überschriften im Unterstreichungsstil verwenden:

```
Überschrift 1
=====

Überschrift 2
-----
```

Die meisten Schreibenden bleiben bei der `#`-Syntax, weil sie auf jeder Ebene funktioniert und leichter zu überblicken ist.

Tipps:

- Setzen Sie immer ein Leerzeichen zwischen `#` und dem Überschriftstext
- Verwenden Sie nur eine H1 pro Dokument (sie ist der Titel)
- Lassen Sie vor und nach Überschriften eine Leerzeile für das sicherste Rendern

Vollständige Anleitung: [Überschriften in Markdown](#) →

Absätze und Zeilenumbrüche

Absätze werden durch eine Leerzeile getrennt. Drücken Sie einfach zweimal die Eingabetaste.

```
Dies ist der erste Absatz.
```

```
Dies ist der zweite Absatz.
```

Zeilenumbrüche innerhalb eines Absatzes

Um eine Zeile umzubrechen, ohne einen neuen Absatz zu beginnen, beenden Sie die Zeile mit **zwei Leerzeichen** gefolgt von der Eingabetaste oder verwenden Sie einen Backslash `\`:

```
Erste Zeile.  
Zweite Zeile in einer neuen Zeile, gleicher Absatz.  
  
Erste Zeile.\nZweite Zeile mit Backslash.
```

Beide erzeugen ein `<br>`-Tag in HTML. Die Methode mit nachgestellten Leerzeichen ist traditioneller, aber im Editor unsichtbar, was Verwirrung stiftet. Die Backslash-Methode ist klarer und wird von GitHub Flavored Markdown unterstützt.

Vollständige Anleitung: [Absätze](#) → · [Zeilenumbrüche](#) →

Fett, Kursiv und andere Textformatierung

Dies sind die Formatierungsgrundlagen, die Sie ständig verwenden werden.

```
**fetter Text** oder __fetter Text__  
*kursiver Text* oder _kursiver Text_  
***fett und kursiv*** oder ___fett und kursiv___  
~~durchgestrichener Text~~
```

Fett und Kursiv gehören zum Kern von CommonMark und funktionieren überall. Durchgestrichen ist eine GFM-Erweiterung, die weiter unten ausführlicher behandelt wird.

Wann Sternchen und wann Unterstriche verwenden

Sowohl `**` als auch `__` erzeugen Fett. Sowohl `*` als auch `_` erzeugen Kursiv. Sie sind austauschbar, wählen Sie also eines und bleiben Sie zur Konsistenz dabei.

Der praktische Unterschied zeigt sich innerhalb eines Wortes. CommonMark und GFM ignorieren Unterstriche zwischen Buchstaben, sodass Unterstriche wörtlich bleiben, während Sternchen immer Hervorhebung auslösen:

```
Bleibt wörtlich:  this_word_has_underscores  
Wird kursiv:     this*word*has*asterisks
```

Verwenden Sie Unterstriche in Prosa, die Unterstriche enthält (wie `file_name`), um versehentliche Kursivstellung zu vermeiden, und verwenden Sie Sternchen, wenn Sie tatsächlich eine Hervorhebung innerhalb eines Wortes möchten:

```
un**glaub**lich
```

Vollständige Anleitung: [Hervorhebung: Fett und Kursiv](#) →

Listen

Listen gibt es in zwei Varianten: ungeordnet (Aufzählungspunkte) und geordnet (nummeriert).

Ungeordnete Listen

Verwenden Sie `-`, `*` oder `+` gefolgt von einem Leerzeichen. Alle drei funktionieren identisch. Die meisten Styleguides bevorzugen `-`.

- Erstes Element
- Zweites Element
- Drittes Element
 - Verschachteltes Element (2 oder 4 Leerzeichen einrücken)
 - Weiteres verschachteltes Element
 - Tief verschachteltes Element
- Viertes Element

Nummerierte Listen

Verwenden Sie Zahlen gefolgt von einem Punkt. Die tatsächlichen Zahlen spielen keine Rolle, Markdown nummeriert automatisch neu:

1. Erstes Element
2. Zweites Element
3. Drittes Element
 1. Verschachteltes Element
 2. Weiteres verschachteltes Element
4. Viertes Element

Sie können auch dies schreiben und dieselbe Ausgabe erhalten:

1. Erstes Element
1. Zweites Element
1. Drittes Element

Das ist ein Feature, kein Fehler. Es bedeutet, dass Sie Listenelemente umordnen können, ohne von Hand neu zu nummerieren.

Gemischte Listen

Sie können ungeordnete Listen in geordnete verschachteln und umgekehrt:

1. Schritt eins
2. Schritt zwei
 - Unterpunkt A
 - Unterpunkt B
3. Schritt drei

Vollständige Anleitung: [Listen in Markdown](#) →

Links

Links verwenden eckige Klammern für den sichtbaren Text und runde Klammern für die URL.

Einfache Inline-Links

```
[Markdific](https://markdific.com)
[Besuchen Sie unseren Blog](https://markdific.com/blog "Optionaler Hover-Titel")
```

Automatische Links

Setzen Sie eine URL in spitze Klammern, um sie automatisch zu verlinken:

```
<https://markdific.com>
<hello@example.com>
```

Referenzstil-Links

Nützlich, wenn derselbe Link mehrfach vorkommt oder wenn Sie die Prosa sauber halten möchten:

```
Lesen Sie unser [Cheat Sheet][1] oder den [Blog][2].

Später im Dokument oder am Ende:

[1]: https://markdific.com/resources/markdown-cheat-sheet
[2]: https://markdific.com/blog/
```

Sie können auch benannte Referenzen verwenden:

```
Schauen Sie sich [Markdific][md-home] an.

[md-home]: https://markdific.com
```

Auf Abschnitte verlinken (Anker-Links)

GitHub und die meisten Renderer erzeugen Anker-IDs automatisch aus dem Überschriftstext (kleingeschrieben, Leerzeichen durch Bindestriche ersetzt):

```
[Zu Tabellen springen](#tables)
[Zu GitHub-Alerts springen](#github-alerts)
```

Vollständige Anleitung: [Links](#) → · [Automatische Links](#) → · [Überschriften-IDs](#) →

Wikilinks und Embeds

Obsidian und einige andere Notiz-Tools ergänzen den Standard-Markdown um Wikilinks und Embeds mit doppelten Klammern:

<code>[[Notiztitel]]</code>	Link zu einer anderen Notiz
<code>[[Notiztitel Alias]]</code>	Link mit eigenem Text
<code>![[Notiztitel]]</code>	Inhalt einer anderen Notiz einbetten
<code>![[image.png]]</code>	ein Bild einbetten

Dies ist eine Obsidian-Erweiterung, kein Standard-Markdown, daher werden sie auf GitHub oder den meisten statischen Seiten nicht aufgelöst. Den vollständigen Satz finden Sie unter [Markdown in Obsidian](#).

Bilder

Die Bildsyntax ist fast identisch mit Links, mit einem Ausrufezeichen davor.

```
![[Alt-Text, der das Bild beschreibt]](path/to/image.jpg)
![[Alt-Text]](path/to/image.jpg "Optionaler Hover-Titel")
```

Bildgröße

Reines Markdown unterstützt keine Bildgrößen. Sie haben zwei Möglichkeiten:

Option 1: HTML direkt verwenden

```

```

Option 2: Plattformspezifische Erweiterungen verwenden

Obsidian zum Beispiel unterstützt:

```
![[image.jpg|500]]
```

Referenzstil-Bilder

```
![[Markdific-Logo]][logo]
```

```
[logo]: /images/logo.png "Markdific"
```

Ein Bild verlinken

Um ein Bild anklickbar zu machen, umschließen Sie die Bildsyntax mit einem Link:

```
[![[Alt-Text]](image.jpg)](https://destination-url.com)
```

Für Fälle, die reines Markdown nicht abdeckt, lokale Dateien, präzise Größen oder Base64-eingebettete Bilder, verwenden Sie das oben gezeigte rohe HTML-Tag `<img>`.

Vollständige Anleitung: [Bilder in Markdown](#) →

Code und Codeblöcke

Die Codeformatierung ist eine der nützlichsten Funktionen von Markdown.

Inline-Code

Umschließen Sie Code mit einzelnen Backticks:

Verwenden Sie die Funktion ``print()`` in Python.

Wenn Ihr Code einen Backtick enthält, umschließen Sie ihn mit doppelten Backticks:

Das Zeichen ``` ` ``` heißt Backtick.

Codeblöcke (Fenced)

Verwenden Sie drei Backticks für mehrzeilige Codeblöcke. Fügen Sie nach den öffnenden Backticks einen Sprachbezeichner hinzu, um Syntaxhervorhebung zu erhalten:

```
```python
def hello_world():
 print("Hello, world!")
```
```

```
```javascript
function helloWorld() {
 console.log("Hello, world!");
}
```
```

```
```bash
npm install markdown-it
```
```

Gängige Sprachbezeichner

- `python`, `py`
- `javascript`, `js`
- `typescript`, `ts`
- `bash`, `sh`, `shell`
- `html`, `xml`
- `css`, `scss`
- `json`, `yaml`
- `sql`, `graphql`
- `markdown`, `md`
- `diff` (für Git-Diffs)
- `plaintext`, `text` (keine Hervorhebung)

Eingerückte Codeblöcke (Legacy)

Rücken Sie eine beliebige Zeile um vier Leerzeichen oder einen Tabulator ein, um sie zu einem Codeblock zu machen. Das funktioniert, ist aber weniger lesbar als Fenced-Blöcke. Vermeiden Sie es.

Vollständige Anleitung: [Code](#) → · [Fenced-Codeblöcke](#) → · [Syntaxhervorhebung](#) →

Blockquotes

Verwenden Sie `>` am Anfang einer Zeile, um ein Blockquote zu erstellen.

```
> Dies ist ein Blockquote.  
> Es kann sich über mehrere Zeilen erstrecken.  
  
> Sie können auch ein einzeliliges Blockquote haben.
```

Verschachtelte Blockquotes

Stapeln Sie `>`-Zeichen:

```
> Äußeres Zitat.  
>  
> > Verschachteltes Zitat.  
> >  
> > > Tief verschachteltes Zitat.
```

Blockquotes mit anderem Markdown

Blockquotes können Überschriften, Listen, Code und andere Formatierungen enthalten:

```
> ### Eine Überschrift innerhalb eines Zitats  
>  
> - Ein Listenelement  
> - Ein weiteres Element  
>  
> Etwas `Inline-Code` und fetter Text.
```

Vollständige Anleitung: [Blockquotes](#) →

Horizontale Linien

Drei oder mehr Bindestriche, Sternchen oder Unterstriche in einer eigenen Zeile:

```
---  
***  
---
```

Alle drei werden als `<hr>` gerendert. Verwenden Sie sie sparsam. Die meisten modernen Dokumente nutzen stattdessen eine Überschriftenhierarchie.

Vollständige Anleitung: [Horizontale Linien](#) →

Tabellen

Tabellen sind eine Erweiterung von GitHub Flavored Markdown, kein Standard-CommonMark, werden aber inzwischen fast überall unterstützt.

Einfache Tabelle

```
Kopf 1	Kopf 2	Kopf 3
Zelle 1	Zelle 2	Zelle 3
Zelle 4	Zelle 5	Zelle 6
```

Spaltenausrichtung

Fügen Sie Doppelpunkte zur Trennzeile hinzu:

```
Linksbündig	Zentriert	Rechtsbündig
Text	Text	Text
Längerer Text	Längerer Text	Längerer Text
```

- `:---` linksbündig (Standard)
- `:---:` zentriert
- `---:` rechtsbündig

Tipps für sauberere Tabellen

- Äußere Pipes sind optional, verbessern aber die Lesbarkeit
- Spaltenbreiten in der Quelle beeinflussen die Ausgabe nicht (Renderer ignorieren zusätzliche Leerzeichen)
- Pipe-Zeichen innerhalb von Zellen müssen als `\|` escaped werden
- Standard-GFM-Tabellen können keine mehrzeiligen oder verbundenen Zellen haben; verwenden Sie dafür HTML

Eine vollständige Anleitung, einschließlich des Umgangs mit komplexen Tabellen, sowie unseren kostenlosen [Markdown-Tabellengenerator](#), finden Sie in unserem [vollständigen Leitfaden zu Markdown-Tabellen](#).

Vollständige Anleitung: [Tabellen in Markdown](#) →

Aufgabenlisten

Aufgabenlisten (auch To-do-Listen genannt) sind eine GFM-Erweiterung, die von GitHub, GitLab und den meisten modernen Editoren unterstützt wird.

- [x] Erledigte Aufgabe
- [] Offene Aufgabe
- [] Weitere offene Aufgabe
 - [x] Verschachtelte erledigte Unteraufgabe
 - [] Verschachtelte offene Unteraufgabe

Das `[x]` unterscheidet in manchen Renderern zwischen Groß- und Kleinschreibung. Verwenden Sie zur Sicherheit ein kleines `x`.

Vollständige Anleitung: [Aufgabenlisten](#) →

Fußnoten

Fußnoten sind eine GitHub-Erweiterung und werden auch von MultiMarkdown und Pandoc unterstützt. Sie eignen sich perfekt für Zitate und Randbemerkungen.

```
Dies ist ein Satz mit einer Fußnote.[^1]

Sie können mehrere Fußnoten haben.[^note]

[^1]: Dies ist der Inhalt der ersten Fußnote.
[^note]: Dies ist eine benannte Fußnote mit längerem Inhalt,
        der sich über mehrere Zeilen erstrecken kann, wenn eingerückt.
```

Fußnoten nummerieren sich automatisch und erscheinen am Ende des gerenderten Dokuments mit Rücklinks.

Vollständige Anleitung: [Fußnoten](#) →

Definitionslisten

Definitionslisten werden von Pandoc, MultiMarkdown und einigen anderen Varianten unterstützt, aber nicht vom Kern-GFM.

```
Begriff
: Definition des Begriffs.

Markdown
: Eine leichtgewichtige Auszeichnungssprache zum Erstellen von formatiertem Text.
: Erstellt von John Gruber im Jahr 2004.
```

Prüfen Sie Ihren Ziel-Renderer, bevor Sie sich darauf verlassen. Sie sind über Plattformen hinweg uneinheitlich.

Vollständige Anleitung: [Definitionslisten](#) →

Abkürzungen

MultiMarkdown und PHP Markdown Extra erlauben es, eine Abkürzung einmal zu definieren und sie überall dort zu erweitern, wo der Begriff erscheint:

Die HTML-Spezifikation wird vom W3C gepflegt.

*[HTML]: HyperText Markup Language

*[W3C]: World Wide Web Consortium

Dies ist weder Teil von CommonMark noch von GFM, daher zeigen die meisten Renderer (einschließlich GitHub) die Definitionszeilen als wörtlichen Text an. Verwenden Sie es nur dort, wo Sie wissen, dass das Ziel es unterstützt.

Durchgestrichen

Umschließen Sie Text mit doppelten Tilden:

~~Dieser Text ist durchgestrichen~~

Gerendert: Dieser Text ist durchgestrichen

GFM-Erweiterung. In modernen Renderern universell unterstützt.

Vollständige Anleitung: [Durchgestrichen](#) →

Tiefgestellt, Hochgestellt und Hervorhebung

Diese drei Inline-Markierungen sind erweiterte Funktionen. Keine davon ist Teil des Kern-CommonMark oder von GitHub Flavored Markdown, daher variiert die Unterstützung, aber jede hat einen zuverlässigen HTML-Fallback, der überall funktioniert.

Tiefgestellt und Hochgestellt

Pandoc, MultiMarkdown und eine Handvoll anderer Varianten unterstützen die Tilde- und Caret-Syntax:

Wasser ist H_2O .

Die Fläche beträgt 10 m^2 .

Wo diese Syntax nicht unterstützt wird (GitHub und die meisten CommonMark-Renderer), verwenden Sie stattdessen die HTML-Tags, sie werden überall gerendert:

Wasser ist H₂O.

Die Fläche beträgt 10 m².

Hervorhebung

Obsidian und einige andere Tools heben Text hervor, der in doppelte Gleichheitszeichen eingeschlossen ist:

Markdown hebt ==diesen Teil== hervor.

Für Renderer, die == nicht unterstützen, verwenden Sie das HTML-Tag `<mark>`:

`<mark>hervorgehoben</mark>`.

Vollständige Anleitung: [Tiefgestellt & Hochgestellt](#) → · [Hervorhebung](#) →

Emoji

Drei Möglichkeiten, Emoji in Markdown zu verwenden:

1. Direkt einfügen

Tippen oder fügen Sie einfach das eigentliche Emoji-Zeichen ein: 🎉 ✅ 🚀

Das funktioniert überall, weil Emoji einfach Unicode-Zeichen sind.

2. Shortcodes (GFM)

GitHub und GitLab unterstützen eine Shortcode-Syntax, die Textcodes in Emoji umwandelt:

```
:tada: :white_check_mark: :rocket:
```

Diese werden auf unterstützten Plattformen als 🎉 ✅ 🚀 gerendert. Discord und Slack verwenden eine ähnliche `:code:`-Syntax, aber mit ihren eigenen Emoji-Bibliotheken, daher unterscheiden sich die verfügbaren Codes. Siehe [Markdown in Discord](#) und [Markdown in Slack](#), um zu erfahren, wie jede Chat-App die Formatierung handhabt.

3. HTML-Entities

Für maximale Kompatibilität können Sie auch HTML-Entity-Codes verwenden, obwohl Shortcodes weitaus lesbarer sind.

Vollständige Anleitung: [Emojis in Markdown](#) →

HTML in Markdown

Sie können rohes HTML in Markdown einbetten, für Fälle, die die Syntax nicht abdeckt:

```
<details>
<summary>Zum Aufklappen klicken</summary>
```

Dieser Inhalt ist standardmäßig verborgen und wird angezeigt, wenn der Benutzer klickt.

```
</details>
```

Häufige Verwendungszwecke für rohes HTML:

- `<details>` und `<summary>` für aufklappbare Abschnitte
- `<sub>` und `<sup>` für Tiefgestelltes und Hochgestelltes
- `<kbd>` für Tastaturtasten (wird als Ctrl + C dargestellt)
- `<mark>` für Hervorhebungen
- `<img>` mit Breitenattributen für Bilder mit fester Größe
- Tabellen mit verbundenen Zellen, mehrzeiligen Zellen oder komplexen Layouts

Beispiel für Tastaturtasten

Drücken Sie `<kbd>Ctrl</kbd>` + `<kbd>C</kbd>` zum Kopieren.

Einschränkungen

Manche Plattformen entfernen oder bereinigen HTML aus Sicherheitsgründen (Reddit zum Beispiel). GitHub erlaubt das meiste HTML, blockiert aber `<script>` und einige andere Tags. Testen Sie immer auf Ihrer Zielplattform.

Vollständige Anleitung: [Inline-HTML in Markdown](#) →

Kommentare

Markdown hat keine offizielle Kommentarsyntax, aber es gibt zwei zuverlässige Wege, Notizen in einer Datei zu hinterlassen, die nie in der gerenderten Ausgabe erscheinen.

HTML-Kommentare

Jeder Renderer, der HTML erlaubt, berücksichtigt auch HTML-Kommentare:

```
<!-- Diese Notiz ist auf der gerenderten Seite unsichtbar. -->
```

Sichtbarer Inhalt geht hier weiter.

Der Kommentar wird übersprungen, wenn die Datei gerendert wird. Er existiert jedoch weiterhin im HTML-Quelltext, behandeln Sie ihn also als vor Lesern verborgen und nicht als wirklich privat.

Der Referenzlink-Trick

Für Renderer, die HTML entfernen (manche Chat-Apps und sandboxierte Viewer), verwenden Sie eine leere Referenzlink-Definition. Sie erzeugt keine Ausgabe, weil nie etwas darauf verlinkt:

```
[//]: # (Dies ist ein Kommentar, der nichts rendert.)
```

```
[comment]: # (Eine andere Art, dasselbe zu schreiben.)
```

Das funktioniert in fast jedem Parser, weil eine Linkdefinition, auf die nie verwiesen wird, einfach verworfen wird. Es ist der portabelste Weg, eine Notiz zu verbergen.

Escapezeichen

Um ein Markdown-Zeichen wörtlich anzuzeigen, stellen Sie ihm einen Backslash voran:

```
\* Dies ist nicht kursiv \  
\# Dies ist keine Überschrift  
\[Dies ist kein Link\](not-a-url)
```

Zeichen, die Sie escapen können:

`\ ` * _ { } [] () # + - . ! | ``

Vollständige Anleitung: [Zeichen escapen](#) →

Mathematische Gleichungen (LaTeX)

Mathematik wird von GitHub, GitLab, Obsidian, Notion und den meisten modernen Dokumentationstools als Renderer-Funktion unterstützt. Verwenden Sie die LaTeX-Syntax innerhalb von Dollarzeichen.

Inline-Mathematik

Einzelne Dollarzeichen für Inline-Mathematik:

Der Satz des Pythagoras lautet $a^2 + b^2 = c^2$.

GitHub unterstützt auch einen alternativen Inline-Begrenzer, der die Mathematik auf jeder Seite in ein Dollarzeichen und einen Backtick einschließt, für Fälle, in denen Ihre Mathematik Zeichen enthält, die mit Markdown in Konflikt geraten:

Hier verwendet `$`a + b = c`$` die alternativen Begrenzer.

Block-Mathematik

Doppelte Dollarzeichen für Gleichungen auf Blockebene:

```
$$  
\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}  
$$
```

Gängige LaTeX-Muster

Brüche:	$\frac{a}{b}$
Quadratwurzel:	$\sqrt{x}$
Potenzen:	x^2
Indizes:	x_i
Griechische Buchstaben:	$\alpha, \beta, \gamma, \pi, \theta$
Summe:	$\sum_{i=1}^n i$
Integral:	$\int_0^1 x \, dx$
Matrix:	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Renderer unterscheiden sich in der Syntax. GitHub verwendet MathJax. Manche Plattformen erfordern KaTeX, das leicht andere Konventionen hat.

Vollständige Anleitung: [LaTeX-Mathematik](#) →

Mermaid-Diagramme

Mermaid ist eine JavaScript-Bibliothek, die Text in Diagramme umwandelt. GitHub, GitLab, Notion, Obsidian und die meisten modernen Markdown-Renderer unterstützen es nativ. Es ist eine der nützlichsten Funktionen im modernen Markdown.

Flussdiagramm

```
```mermaid
graph TD
 A[Start] --> B{Funktioniert es?}
 B -->|Ja| C[Ausliefern]
 B -->|Nein| D[Debuggen]
 D --> B
```
```

Sequenzdiagramm

```
```mermaid
sequenceDiagram
 Benutzer->>Browser: .md-Datei öffnen
 Browser->>Markdific: Markdown rendern
 Markdific-->>Browser: HTML zurückgeben
 Browser-->>Benutzer: Formatierte Seite anzeigen
```
```

Weitere Mermaid-Diagrammtypen

- `flowchart` (Prozessabläufe)
- `sequenceDiagram` (Interaktionen im Zeitverlauf)

- `classDiagram` (UML-Klassen)
- `stateDiagram-v2` (Zustandsautomaten)
- `erDiagram` (Datenbankschemata)
- `gantt` (Projektzeitpläne)
- `pie` (Kreisdiagramme)
- `mindmap` (Mindmaps)
- `journey` (Nutzerreisen)
- `gitGraph` (Git-Branch-Visualisierungen)

Mermaid unterstützt weit mehr Diagrammtypen als hier gezeigt. Welche gerendert werden, hängt von der Mermaid-Version Ihres Tools ab, GitHub und GitLab halten ihre aktuell, daher funktionieren die meisten Diagrammtypen sofort.

Vollständige Anleitung: [Mermaid-Diagramme](#) →

GitHub–Alerts

GitHub-Alerts (auch "Callouts" oder "Admonitions" genannt) sind eine GitHub-Funktion, die Ende 2023 eingeführt wurde. Sie rendern spezielle Blockquotes mit Symbolen und Farben und werden inzwischen von vielen anderen Markdown-Renderern breit unterstützt.

```
> [!NOTE]
> Nützliche Informationen, die Benutzer kennen sollten.

> [!TIP]
> Hilfreiche Ratschläge, um Dinge besser zu machen.

> [!IMPORTANT]
> Wichtige Informationen, die Benutzer kennen müssen.

> [!WARNING]
> Dringende Informationen, die sofortige Aufmerksamkeit erfordern.

> [!CAUTION]
> Weist auf Risiken oder negative Folgen hin.
```

Die fünf unterstützten Typen sind `NOTE`, `TIP`, `IMPORTANT`, `WARNING` und `CAUTION`. Andere Markdown-Tools (Obsidian, MkDocs, Docusaurus) verwenden eine ähnliche, aber leicht abweichende Syntax für Callouts. Siehe [Markdown auf GitHub](#), um zu erfahren, wie Alerts und andere GitHub-Funktionen gerendert werden.

YAML–Frontmatter

Frontmatter ist ein Block aus YAML-Metadaten am Anfang einer Markdown-Datei, der von Generatoren für statische Seiten (Hugo, Jekyll, Astro, Next.js), CMS-Plattformen und Notiz-Apps verwendet wird.

```
---
title: "Mein Artikeltitel"
date: 2026-01-15
author: Jane Doe
tags:
  - markdown
  - tutorial
  - referenz
draft: false
description: Eine kurze Zusammenfassung für SEO.
---

# Der Artikelinhalt beginnt hier
```

Drei Bindestriche öffnen und schließen den Block. Verwenden Sie darin die Standard-YAML-Syntax. Gängige Felder:

- `title`
- `date`
- `author` oder `authors`
- `tags` oder `categories`
- `description` (für Meta-Tags)
- `draft` (boolean)
- `slug` oder `permalink`
- `image` (Pfad zum Beitragsbild)

Frontmatter ist in der gerenderten Ausgabe unsichtbar. Es sind reine Metadaten für das System, das die Datei verarbeitet.

Wenn Ihr Frontmatter in der Ausgabe als wörtlicher Text erscheint, unterstützt der Parser es wahrscheinlich nicht, prüfen Sie, ob die Datei in Zeile 1 mit `---` beginnt und keine Leerzeile darüber hat.

Vollständige Anleitung: [Front Matter](#) →

Markdown–Varianten (Flavors) im Vergleich

Markdown hat mehrere "Varianten" (Flavors) mit unterschiedlichen Funktionen. Zu wissen, welche Ihre Zielpattform verwendet, verhindert fehlerhaftes Rendern.

| Variante | Wo verwendet | Hauptfunktionen |
|--------------------------------|--|--|
| CommonMark | Reddit, Stack Overflow | Der standardisierte Kern. Vorhersehbar, aber begrenzt. |
| GitHub Flavored Markdown (GFM) | GitHub, GitLab, die meisten modernen Tools | CommonMark + Tabellen, Aufgabenlisten, Durchgestrichenes, Autolinks, Fußnoten, Alerts |
| MultiMarkdown | Akademisches und technisches Schreiben | Ergänzt Tabellen, Fußnoten, Mathematik, Zitate, Metadaten |
| Pandoc Markdown | Pandoc-Konverter | Am leistungsfähigsten. Ergänzt Definitionslisten, Mathematik, Tabellen, Zitate, rohes HTML |
| Obsidian Markdown | Obsidian-Notiz-App | GFM + Wikilinks <code>[[note]]</code> , Callouts, Embeds, Tags |
| R Markdown | R, RStudio, Quarto | Ergänzt ausführbare Codeblöcke für die Datenanalyse |
| Discord / Slack | Chat-Apps | Eigene Teilmengen mit plattformspezifischer Syntax (Spoiler, Erwähnungen, Emoji-Codes) |

Im Zweifel schreiben Sie GFM. Es kommt einem universellen Markdown-Standard 2026 am nächsten. Für plattformspezifisches Verhalten siehe [Markdown in Obsidian](#) und [Markdown in Notion](#).

Vollständige Anleitung: [Markdown-Varianten im Vergleich](#) →

Häufige Fehler

Diese bringen Schreibende ständig durcheinander. Vermeiden Sie sie.

1. Leerzeilen vergessen

Markdown benötigt oft eine Leerzeile zwischen Elementen:

```
Dies ist ein Absatz.
- Diese Liste wird möglicherweise nicht gerendert
- je nach Renderer.
```

Beheben Sie es mit einer Leerzeile:

```
Dies ist ein Absatz.

- Diese Liste wird korrekt gerendert.
- Jeder Renderer stimmt zu.
```

2. Einrückung mischen

Beim Verschachteln von Listen oder Hinzufügen von Absätzen zu Listenelementen verwenden Sie eine einheitliche Einrückung (2 oder 4 Leerzeichen). Das Mischen von Tabulatoren und Leerzeichen bricht das Rendern.

3. Nachgestellte Leerzeichen für Zeilenumbrüche

Der Trick mit "zwei Leerzeichen am Zeilenende" für weiche Zeilenumbrüche ist unsichtbar und bricht leicht. Verwenden Sie stattdessen einen Backslash oder akzeptieren Sie einfach, dass Absätze meist das

sind, was Sie wollen.

4. Sonderzeichen nicht escapen

`func_name_with_underscores` in Prosa zu schreiben ist in CommonMark und GFM sicher, die Unterstriche zwischen Buchstaben ignorieren, obwohl einige ältere Parser sie noch als Hervorhebung lesen. Sternchen sind hier riskanter, da `a*b*c` kursiv setzt. Um überall sicher zu sein, umschließen Sie Fachbegriffe mit Backticks für Inline-Code. Weitere häufige Kollisionen: `<` und `>` werden als HTML-Tags gelesen, und `|` innerhalb von Tabellenzellen bricht das Tabellenlayout.

5. Sich auf nicht standardisierte Funktionen verlassen

Wenn Sie Obsidian-Wikilinks `[[note]]` verwenden und dann auf einer statischen Seite veröffentlichen, die sie nicht unterstützt, brechen Ihre Links. Kennen Sie Ihren Ziel-Renderer.

6. Tabellen, die zu breit sind

GFM-Tabellen können nicht horizontal scrollen. Wenn eine Tabelle zu breit für die Seite ist, läuft sie über. Teilen Sie breite Tabellen in mehrere schmalere auf oder verwenden Sie HTML.

7. Leere Überschriften oder Überschriften nur mit Formatierung

```
## **Fette Überschrift**
```

Das funktioniert, aber manche Renderer entfernen die Formatierung aus Überschriften oder erzeugen seltsame Anker-IDs. Halten Sie Überschriften schlicht.

Das PDF-Cheat-Sheet herunterladen

Möchten Sie dieses Cheat Sheet offline haben? [Laden Sie die druckbare PDF-Version herunter](#), um sie auf Ihrem Schreibtisch zu haben oder mit Ihrem Team zu teilen.

Häufig gestellte Fragen

Was ist ein Markdown Cheat Sheet?

Ein Markdown Cheat Sheet ist eine Kurzreferenz, die jedes Markdown-Syntaxelement zusammen mit einem Beispiel auflistet, wie man es schreibt, Überschriften, fetten und kursiven Text, Listen, Links, Bilder, Codeblöcke, Tabellen und mehr. Diese Seite deckt ganz CommonMark plus die Erweiterungen von GitHub Flavored Markdown ab.

Wie mache ich Text in Markdown fett?

Umschließen Sie den Text mit zwei Sternchen: `**fetter Text**`. Zwei Unterstriche (`__fetter Text__`) funktionieren ebenfalls. Für Kursiv verwenden Sie ein einzelnes Sternchen oder einen Unterstrich; für fett und kursiv zusammen verwenden Sie drei: `***text***`.

Wie erstelle ich eine Tabelle in Markdown?

Trennen Sie Spalten mit Pipe-Zeichen und fügen Sie unter der Kopfzeile eine Reihe von Bindestrichen hinzu, um sie als Tabelle zu kennzeichnen:

```
Name	Rolle
Ada	Lead
```

Oder überspringen Sie die Syntax ganz und bauen Sie eine visuell mit unserem kostenlosen [Markdown-Tabellengenerator](#).

Was ist der Unterschied zwischen Markdown und GitHub Flavored Markdown?

CommonMark ist die standardisierte Kernsyntax. GitHub Flavored Markdown (GFM) erweitert sie um Tabellen, Aufgabenlisten, Durchgestrichenes, Autolinks, Fußnoten und Alerts. GFM ist die am breitesten unterstützte Variante und daher 2026 der sicherste Standard.

Ist Markdown dasselbe wie HTML?

Nein. Markdown ist eine leichtgewichtige Klartext-Syntax, die in HTML umgewandelt wird. Markdown dient zum Schreiben und Bearbeiten; HTML dient zur Anzeige. Unser Leitfaden zu [Markdown vs HTML](#) erklärt, wann man welches verwendet.

Wie füge ich einen Link in Markdown hinzu?

Setzen Sie den sichtbaren Text in eckige Klammern, gefolgt von der URL in runden Klammern:

`[Markdific](https://markdific.com)`. Um ein Bild in einen Link zu verwandeln, umschließen Sie die Bildsyntax mit denselben Klammern.

Wie füge ich einen Kommentar in Markdown hinzu?

Markdown hat keine offizielle Kommentarsyntax. Die portabelste Methode ist ein leerer Referenzlink, `[//]: # (Ihre Notiz)`, der nichts rendert. In jedem Renderer, der HTML erlaubt, funktioniert auch ein HTML-Kommentar `<!-- Ihre Notiz -->`.

Wie erstelle ich eine Checkliste in Markdown?

Verwenden Sie eine Aufgabenliste: Beginnen Sie jedes Element mit `- [ ]` für eine offene Aufgabe oder `- [x]` für eine erledigte. Aufgabenlisten sind eine Funktion von GitHub Flavored Markdown, die von GitHub, GitLab und den meisten modernen Editoren unterstützt wird.

Wie füge ich ein Bild in Markdown hinzu?

Verwenden Sie ein Ausrufezeichen, Alt-Text in eckigen Klammern und den Pfad in runden Klammern: `![Alt-Text](path/to/image.jpg)`. Um ein Bild in der Größe anzupassen, verwenden Sie stattdessen ein HTML-Tag `<img>` mit einem Attribut `width`.

Wie schreibe ich einen Codeblock in Markdown?

Umschließen Sie den Code mit drei Backticks in eigenen Zeilen. Fügen Sie nach den öffnenden Backticks einen Sprachnamen für die Syntaxhervorhebung hinzu. Für einen kurzen Ausschnitt innerhalb eines Satzes verwenden Sie einzelne Backticks um den `code`.

Wie erstelle ich ein Inhaltsverzeichnis in Markdown?

Markdown hat kein integriertes Inhaltsverzeichnis, aber Sie können eines aus Anker-Links wie `[Abschnittstitel](#section-title)` aufbauen, indem Sie die kleingeschriebene Überschrift mit durch Bindestriche ersetzten Leerzeichen verwenden. GitHub, GitLab und viele Editoren erzeugen ebenfalls automatisch eines.

Wie konvertiere ich Markdown in Word oder PDF?

Verwenden Sie einen Konverter. Markdific hat kostenlose Online-Tools, um Markdown in [Word](#), [PDF](#), [HTML](#) und mehr zu konvertieren, alle laufen in Ihrem Browser ohne Anmeldung.

Verwandte Anleitungen und Tools

Markdown nach Plattform. Sehen Sie genau, wie sich Markdown in den Tools verhält, die Sie täglich verwenden:

- [Markdown auf GitHub](#)
- [Markdown in Obsidian](#)
- [Markdown in Notion](#)
- [Markdown in VS Code](#)
- [Markdown in Discord](#)
- [Markdown in Slack](#)
- [Markdown in Jupyter](#)
- [Markdown in Confluence](#)
- [Wie man eine README schreibt](#)
- [Markdown-Dateien für KI-Agenten](#)

Konvertieren Sie Ihr Markdown. Kostenlose Online-Konverter, die in Ihrem Browser laufen, ohne Anmeldung:

- [Markdown zu Word](#)
 - [Markdown zu PDF](#)
 - [Markdown zu HTML](#)
 - [Markdown zu Text](#)
 - [Markdown zu Bild](#)
 - [Markdown zu Google Docs](#)
-

Fazit

Das ist jede Markdown-Syntax, der Sie 2026 realistisch begegnen werden. Die wichtigsten Erkenntnisse:

- **CommonMark + GFM** deckt 95 % der realen Markdown-Nutzung ab
- **Tabellen, Aufgabenlisten, Fußnoten, Mathematik, Mermaid und Alerts** sind die Erweiterungen, die es sich zu merken lohnt
- **HTML-Fallbacks** erledigen die Sonderfälle, die Markdown nicht abdeckt
- **Frontmatter** ist für jede statische Seite oder jedes Content-System unverzichtbar

- **Mermaid-Diagramme** sind eine der am wenigsten genutzten Funktionen in Markdown

Das Format wird sich weiterentwickeln, während KI-generierte Inhalte das Web mit .md-Dateien überfluten, aber die Kernsyntax ist seit 2004 stabil und wird sich nicht ändern. Lernen Sie sie einmal, und sie zahlt sich über Jahrzehnte aus.

Möchten Sie Ihr Markdown sauber gerendert sehen? [Markdific](#) ist ein schneller, dedizierter Markdown-Viewer und -Editor für Mac und Windows. Mehr dazu finden Sie unter [Markdown vs HTML](#), in unseren [Markdown-Ressourcen](#) oder in der vollständigen [Markdown-Dokumentation](#).

Markdific

Markdific verwandelt rohes Markdown auf jedem Gerät in saubere, lesbare Dokumente. Öffnen Sie eine beliebige `.md`-Datei und sehen Sie sofort korrekte Überschriften, Tabellen, Code, Mathematik und Bilder, bearbeiten Sie dann mit Autospeichern, wechseln Sie Themes und exportieren Sie nach PDF, Word oder HTML.



Markdific ist ein leichtgewichtiger Markdown-Viewer und -Editor, der jede .md-Datei öffnet und sofort rendert und rohes Markdown in saubere, formatierte Überschriften, Tabellen, Code, Mathematik und Bilder verwandelt. Bearbeiten Sie mit Autospeichern, wechseln Sie Themes und exportieren Sie nach PDF, Word oder HTML.

Markdific herunterladen