**Erstellung von Markdown-Dokumentation mit markdownlint**

[Wikipedia-Artikel zur Markdown](https://de.wikipedia.org/wiki/Markdown)

Markdown ist eine vereinfachte Auszeichnungssprache, welche das Ziel hat, das Ausgangsformat eines Textes leicht lesbar zu machen, ohne dass eine weitere Konvertierung nötig ist. Diese Sprache wird oft für die Erstellung von Readme-Dateien auf Entwicklungsplattformen benutzt oder als Sprache für Software-Dokumentation.

Im Gegensatz zu anderen Sprachen ist Markdown nicht vollständig standardisiert. Da die Standard-Implementation nicht immer ausreichend ist, hat man Variationen entwickelt (die sog. Dialekte), welche die Nutzung gebräuchlicher Elemente erlauben (wie z.B. Tabellen). Des Weiteren können auch HTML-Elemente benutzt, um die Syntax noch weiter zu erweitern. Da die Mehrheit der Dialekte für die Erfüllung spezifischer Bedürfnisse entwickelt worden sind,

Markdown-Dateien besitzen das Suffix .md. Für die Erstellung und Bearbeitung von Markdown-Dokumenten wird ein Text-Editor benötigt. Auch die gängigen IDE`s unterstützen die Erstellung von Markdown-Inhalten, mithilfe von Interpretationsplugins. Die Beispiele in dieser Anleitung wurden in Visual Studio Code erstellt, für das Syntax-Highlight wurde die Erweiterung markdownlint genutzt.

Markdownlint ist eine Extension für die Entwicklungsumgebung Visual Studio Code, welches die Erstellung von Markdown-Dokumentation unterstützt. Es funktioniert sowohl als Style Checker, als auch als Linter, welcher der Code gegenüber einen Regelsatz analysiert. Falls eine Regel verletzt wird, gibt der Linter eine Warnung zurück. Die Installation der Erweiterung erfolgt direkt in VSCode über dem Extensions-Menü.

Die Erweiterung markdownlint steht unter [diesem Link](https://marketplace.visualstudio.com/items?itemName=DavidAnson.vscode-markdownlint) zur Verfügung. Eine weitere hilfreiche Erweiterung, welche aber optional ist, findet man [hier](https://marketplace.visualstudio.com/items?itemName=hnw.vscode-auto-open-markdown-preview)

[Installation von Extensions in Visual Studio Code](https://code.visualstudio.com/docs/editor/extension-gallery)

Es wird der voreingestellte Regelsatz der Extension verwendet. Obwohl diese Regelliste nicht offiziell ist, ist die Benutzung ohne Änderungen empfehlenswert. Der Grund dafür ist, dass er sich an dem „puren“ markdown-Syntax ohne HTML orientiert, was ein hoher Grad an Kompatibilität bringt. Bei Bedarf kann dieser Regelsatz erweitert, geändert oder sogar überschrieben werden. Davon wird aber abgeraten, falls weitgehende Kompatibilität höher priorisiert wird.

Der komplette Regelsatz findet man unter [diesem Link](https://github.com/DavidAnson/markdownlint/blob/main/doc/Rules.md).

**Syntax –** Hier beschrieben, wie man Markdown-Elemente erstellt. In der beigefügten Datei beispiel.md stehen Beispiele für alle Elemente zur Verfügung.

* **Überschriften -** Eine Überschrift wird mit einer Raute (#) bezeichnet. Es sind insgesamt 6 Ebenen von Überschriften verfügbar. Ein markdown-Dokument beginnt immer mit einer Überschrift der Ebene 1.Für die ersten 2 Ebenen existiert auch eine alternative Schreibweise.
* **Fließtext –** Der Fließtext wird wie eingegeben dargestellt. Ein Absatz wird mit der Eingabe einer leeren Zeile erzeugt.
* **Escaping –** Liegen Symbole vor, welche die Formatierung bewirken, können diese mit einem Backslash (\) escaped werden. Mit einem doppelten Backslash kann man einen Backslash hinzufügen.
* **Schriftart –** Mit Sternen (\*) oder Underscores (\_) kann man den Text kursiv (\*), fett (\*\*) oder fett und kursiv (\*\*\*) setzen. Die Benutzung von Sternen wird empfohlen, um Verwirrungen zu vermeiden
* **Ungeordnete Listen –** gibt man ein Leerzeichen nach einem Stern, kann eine ungeordnete Liste erstellt werden. Durch Einrücken kann man bis auf 5 Unterpunkte erstellen.
* **Geordnete Listen –** Die Eingabe einer Ziffer mit Punkt, gefolgt von einem Leerzeichen (z.B. 1.) erzeugt eine geordnete Liste. Die Nummerierung erfolgt automatisch. Es können bis 3 weitere Unterpunkte erstellt werden.
* **Zitate –** Das Symbol > am Anfang der Zeile bezeichnet ein Zitat. Er wird automatisch in einem HTML-Blockquote-Element eingepackt
* **Hyperlinks –** Man benutzt die folgende Syntax, um Hyperlinks einzubetten: [Text](URL). Der Text in den eckigen Klammern wird dann zu einem Hyperlink, der zu der URL in den einfachen Klammern weiterleitet. Optional kann nach der URL einen Titel hinzugefügt werden, welche beim Überfahren mit dem Maus angezeigt wird. Der Titel wird direkt nach der URL in Anführungszeichen geschrieben, als Teiler dient ein Leerzeichen
* **Quellcode –** Wird ein Textstück mit dem Symbol ` eingerahmt, wird dieses Text als Inline Quellcode angezeigt.

**Weitere Elemente und Beispiele findet man in der Beispielsdatei.**

**Hilfreiche Links**

* [Ausführliche Liste mit Editoren und Einrichtungsmöglichkeiten für Markdown](https://softwarerecs.stackexchange.com/questions/5746/markdown-editor-for-windows-with-live-rendering-in-the-editing-pane-not-in-a-se)
* [Markdown Cheat Sheet](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)
* [Erstellung von Anchor Links mit einfachem Markdown](https://stackoverflow.com/questions/6695439/how-to-link-to-a-named-anchor-in-multimarkdown)
* [Erstellung von Inhaltsverzeichnissen](https://stackoverflow.com/questions/11948245/markdown-to-create-pages-and-table-of-contents?page=1&tab=votes#tab-top)