

Markdown mit Visual Studio Code und Pandoc

IceDynamix

2020-11-23

Inhaltsverzeichnis

1	Markdown	1
1.1	Warum Markdown?	1
1.2	Syntax	1
2	Visual Studio Code (VSCode)	2
2.1	Setup	2
2.2	Command Palette	2
2.3	Extensions	2
2.4	Einstellungen	3
2.5	Keymaps	3
3	Pandoc	3
3.1	Konvertieren	3
3.2	Pandoc Markdown	4
3.3	YAML Header	4

1 Markdown

1.1 Warum Markdown?

A Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – John Gruber

Markdown (.md) ist ein Format mit dem Dokumente aufgezeichnet werden können. Der große Vorteil daran ist, dass es relativ simpel ist und schnell zu schreiben ist.

Das Format ist außerdem dafür gemacht ohne speziellen Viewer gelesen werden zu können. Das bedeutet auch, dass man mit jedem gängigen Texteditor ohne Setup, sei es Microsoft Notepad, Notepad++ oder Vim, eine Markdown Datei lesen und schreiben kann. Somit ist man, wie zum Beispiel bei OneNote oder Word, nicht an ein proprietäres Programm gebunden und es ist auch definitiv schneller zu schreiben und leichter aufzusetzen als LaTeX (es gibt kein Setup).

Durch die grundlegende Architektur lässt es sich auch leicht in andere Formate konvertieren, was ich später im Kapitel Pandoc weiterführen werden. Falls das Dokument gerade als PDF oder als Webseite gelesen wird, besteht immer die Möglichkeit alles als Quellcode bei mir auf GitHub zu finden!

1.2 Syntax

Grundlegende Syntax Dokumentation ist hier einzusehen.

Wichtig ist dass ein Zeilenumbruch in Markdown nicht exakt einem Zeilenumbruch entspricht, sondern dass man zwei Zeilenumbrüche machen muss um einen neuen Paragraphen zu starten.

Im Kern ist Markdown nur stark vereinfachtes HTML, ein Subset, wenn man es so nennen möchte. Das bedeutet auch, dass man gängiges HTML auch pur in Markdown einbinden kann, falls gewollt. Scripts sind generell ausgeschlossen, inline Style aber erlaubt. Hier ein kleines Beispiel mit dem `details/summary` Tag, welches man anklicken und aufklappen kann (funktioniert nur auf Webseiten):

Klapp mich auf! Hallo Welt!

In manchen Markdown Variationen lässt sich LaTeX Mathematik benutzen! Dafür einfach zwischen zwei bzw. vier Dollar Zeichen packen.

$$\int_a^b f(x)dx = [F(x)]_a^b = |F(b) - F(a)|$$

2 Visual Studio Code (VSCode)

Das Schreiben von Markdown mag zwar in jedem Editor möglich sein, aber hier möchte ich speziell das Arbeiten mit Visual Studio Code in den Vordergrund stellen. Dieser ist ein sehr flexibler Editor, den man für alle vorstellbaren Programmiersprachen benutzen kann, da es für gefühlt alles eine Extension gibt. Darunter auch Markdown.

2.1 Setup

Von hier herunterladen und Installer ausführen. Eine detaillierte Anleitung ist hier zu finden.

Markdown funktioniert out-of-the-box, man muss theoretisch keine Extension installieren. Dafür einfach eine Markdown Datei erstellen und im Editor **Ctrl+K Ctrl+V** drücken (**Ctrl** kann man während beiden Eingaben gedrückt halten). Dies wird eine Preview des momentanen Markdown Dokuments auf der rechten Seite öffnen.

2.2 Command Palette

Wichtig im generellen Umgang mit VSCode ist, dass man *alles* unter der **Command Palette** (**Ctrl+Shift+P**) findet. Das ist mit Abstand der wichtigste Shortcut, da er Zugriff zu allen anderen Shortcuts geben kann. Zum Beispiel kann man auch von dort die Preview öffnen, einfach in der Command Palette nach “Markdown:” suchen. Bei jedem Command steht auch der Shortcut neben dran, falls man diesen vergessen haben sollte.

2.3 Extensions

Die Extensions sind wahrscheinlich das stärkste Tool von VSCode. Damit lässt sich Support für unzählige Sprachen einrichten, sei es Python, JavaScript, Java, C++, Assembly verschiedener Art oder gar AutoHotKey! Auch lässt sich damit das Verhalten einstellen, wie z.B. dass **Ctrl+Links** bzw. **Ctrl+Rechts** (was normalerweise ein Cursor-Schritt in Wortgruppen ist) auch auf Wortgruppen in CamelCase Format einrichten. Links gibt es ein Icon welches wie ein Puzzle aussieht, das ist das Extensions Menü. Oder einfach über die Command Palette aufrufen.

Die Extension die für Markdown am relevantesten ist ist “Markdown All in One”. Wer dazu noch Style-Checking in Form eines Linters haben möchte, kann sich auch “markdownlint” installieren.

Markdown All in One fügt folgende Features hinzu:

- LaTeX Mathematik
- Shortcuts für verschiedene Textformatierungen (z.B. **Ctrl+B** für Fett)
- Automatische Formatierung von Listen oder Tabellen
- Automatisches Generieren und Updaten eines Inhaltsverzeichnis (in Command Palette nach “Table of Contents” suchen)
- Automatisches Generieren von Kapitelnummern und Updaten
- Autocomplete (**Ctrl+Space**) für Links (Header Links, Link Definitionen, Pfade)
- und mehr...

2.4 Einstellungen

Die Einstellungen lassen sich auch von der Command Palette aufrufen, dabei gibt es zwei Wege diese einzusehen und zu bearbeiten.

- **UI Ansicht**, dort wo man sich durchklicken kann für alles was man braucht, also das was man von anderen Programmen auch gewohnt ist.
- **JSON Ansicht**, welche alle gesetzten Einstellungen im JSON Format auflistet und man diese auch direkt bearbeiten kann. Das macht es z.B. leicht Einstellungen einer anderen Person direkt zu übernehmen, da man einfach nur die JSON Einstellungen kopieren und einfügen muss.

Einstellungen die ich direkt empfehlen kann sind folgende:

```
{  
  // Formatierung eines Dokuments  
  "editor.formatOnSave": true,  
  "editor.formatOnType": true,  
  "editor.formatOnPaste": true,  
  
  // Optionen um den Editor generell *smooth* aussehen und fühlen zu lassen  
  "editor.cursorBlinking": "phase",  
  "editor.cursorSmoothCaretAnimation": true,  
  "editor.smoothScrolling": true,  
}
```

Falls man möchte, kann man sich die Einstellungen auf einer Cloud entweder über einen GitHub Account oder über einen Microsoft Account synchronisieren lassen. Dabei über dem Zahnrad unten links auf das Benutzer Icon klicken. Dies speichert auch die selbst-festgelegten Keyboard Shortcuts, Snippets und installierte Extensions.

2.5 Keymaps

Wer schon mit anderen Editoren/IDEs, wie zum Beispiel Eclipse, Atom, Sublime oder Vim gearbeitet hat, wird hier schnell fündig, denn es gibt Keymap Extensions welche die vertrauten Keybinds von dort in VSCode einbinden. Man muss sich also kaum umgewöhnen! Einfach in der Command Palette nach “Preferences: Keymap” suchen.

3 Pandoc

Pandoc ist das letzte Tool um die gesamte Combo wahrlich scheinen zu lassen. Es ist nämlich das schweizer Taschenmesser an Dokumentkonvertern. HTML zu Markdown? Markdown zu PDF? Microsoft Word zu LaTeX?? Alles möglich.

(PDF/LaTeX bedingt aber, dass eine LaTeX Engine installiert ist, wie zum Beispiel MikTeX oder TeXLive)

3.1 Konvertieren

Um den Bezug zu Markdown zu halten, hier ein paar Terminal Commands um Dokumente zu konvertieren. Vollständige Dokumentation für Pandoc ist hier zu finden.

```
# Dieses Dokument zu PDF (Latex) oder einer HTML Datei  
# -o steht für output  
pandoc ./de.md -o de.pdf  
pandoc ./de.md -o de.html  
  
# Für eine vollständige HTML mit Header und allem drum und dran
```

```
# -s steht für standalone
pandoc ./de.md -s -o de.html
```

Der vollständige Command mit dem ich die PDF generiert habe ist der obere von beiden.

3.2 Pandoc Markdown

Pandoc kann aber mehr als nur konvertieren. Es implementiert mehr Features zu Markdown als es der Standard hat und kann diese auch lesen und konvertieren. Hier ein paar Beispiele von der Pandoc Markdown Documentation:

Extension: line_blocks

A line block is a sequence of lines beginning with a vertical bar (|) followed by a space. The division into lines will be preserved in the output, as will any leading spaces; otherwise, the lines will be formatted as Markdown. This is useful for verse and addresses:

```
| The limerick packs laughs anatomical
| In space that is quite economical.
|   But the good ones I've seen
|   So seldom are clean
| And the clean ones so seldom are comical
```

Extension: strikethrough

To strikethrough a section of text with a horizontal line, begin and end it with `~~`. Thus, for example,

```
This ~~is deleted text.~~
```

Extension: footnotes

Pandoc's Markdown allows footnotes, using the following syntax:

Here is a footnote reference, `[^1]` and another. `[^longnote]`

`[^1]`: Here is the footnote.

`[^longnote]`: Here's one with multiple blocks.

Da diese Features nicht im Standard Markdown sind, weiß VSCode nicht was es mit diesen anfangen soll. Dafür gibt es die Pandoc Markdown Preview Extension, welche man anstelle der normalen Preview benutzen kann. Pandoc Markdown unterstützt natürlich auch LaTeX und die anderen Features von der Markdown All in One Extension sind auch noch da.

3.3 YAML Header

Wer in den Quellcode dieser Datei geschaut hat, hat vielleicht folgendes am Anfang bemerkt:

```
---
# YAML Header für Pandoc Metadaten, siehe letztes Kapitel
title: Markdown mit Visual Studio Code und Pandoc
author: IceDynamix
date: "2020-11-23"
documentclass: article
lang: de
geometry: margin=3cm # verändert die Randgröße
toc: true # erstellt ein Inhaltsverzeichnis
numbersections: true # generiert Kapitelnummern
papersize: a4
---
```

Dies ist ein YAML Header, welcher weitere Metadaten für die Konvertierung von Dokumenten durch Pandoc enthält. Zum Beispiel kann Pandoc in PDF LaTeX jetzt eine Titelseite generieren, da es jetzt den vollständigen Titel, Autor und Datum kennt. Auch kann man weitere Metadaten übergeben, wie zum Beispiel `geometry: margin=2cm`, welcher in PDF den Rand verändert, wie mit dem Geometry Package in LaTeX selbst.