
Ups and Downs of Markup

kratenko

2023-04-27

Inhaltsverzeichnis

Ups and Downs of Markup	3
Markup: Auszeichnungssprachen	3
HTML	3
Wikitext	4
Web 2.0	5
Markdown	5
Übersetzen	7
PDF	7
HTML	8
Weitere Formate	8

Ups and Downs of Markup

Wir wissen jetzt, was eine Textdatei ausmacht, und ahnen, wieso sie besonders und mächtig ist. Wir haben HTML und MediaWiki als Sprachen zum Beschreiben von Dokumenten kennengelernt. Es handelt sich dabei um Markup-Sprachen oder Auszeichnungssprachen. In dieser Lektion vertiefen wir, was das bedeutet, und wir lernen eine weitere Markup-Sprache kennen.

Markup: Auszeichnungssprachen

Diese Sprachen beschreiben die Struktur bzw. den Aufbau und die Gliederung eines Dokumentes. Eine Datei enthält Quellcode, bestehend aus dieser Gliederung und dem Inhalt als reinem Text. Ein Programm erzeugt aus dieser Quelldatei das eigentliche Dokument, das dann von einem Menschen betrachtet wird. Für HTML-Dateien ist dies ein Browser. Für MediaWiki-Wikitext ist dies die MediaWiki-Software und ein Browser (aka: Die Wikipedia).

HTML

HTML - *Hypertext Markup Language* ist, was das WWW ausmacht. Anweisungen werden in Tags geschrieben, die in spitzen Klammern geschrieben sind und einen englischen Namen haben, z.B.: `<body>`, `
`, `<title>`, ``.

Manche Tags haben Attribute: ``.

Tags wie für Bilder `` oder Zeilenumbruch `
` stehen alleine. Andere haben ein zugehöriges, schließendes Tag, mit dem gleichen Namen, aber mit einem Slash / vorgestellt. Sie beeinflussen alles, was zwischen öffnendem und schließendem Tag steht: `<h1>Überschrift</h1>`.

Ein simples HTML-Dokument, das ein Bild anzeigt (das Bild muss als separate Datei vorliegen):

```
<html lang="de">
  <head>
    <title>Eine Rakete</title>
  </head>
  <body>
    <h1>Eine krasse Rakete</h1>
    
    Gleich hebt sie ab, die <b>Rakete</b>!
  </body>
</html>
```

HTML gibt eine Gliederung eines Dokumentes an, mit Überschriften, absetzen, etc. Mit grundlegenden Englischkenntnissen und ein wenig Übung ist es möglich ein HTML-Dokument im Quellcode nachzuvollziehen. Es ist jedoch oft nicht einfach, einen Text im HTML-Quellcode zu lesen, da die Tags recht stark vom Inhalt ablenken. Die Syntax und übliche Formatierung/Einrückung legen den Schwerpunkt auf die Struktur und lassen den Text oft untergehen. Gerade komplexere Seiten (was heutzutage auf fast alle Seiten im WWW zutrifft), sind oft mit HTML überladen, sodass ein Lesen unzumutbar wird. Als Beispiel kann man sich mal den Wikipediaartikel zu HTML im Browser anschauen, und dann mal den Quellcode dazu lesen (dies erreicht man in einigen Browsern über Strg-U, bzw. Rechtsclick, "Seitenquelltext anzeigen"):

https://de.wikipedia.org/wiki/Hypertext_Markup_Language

Ein Beispiel für eine minimalistische HTML-Seite ist das Blog von Fefe. Dort kann den Quelltext noch besser lesen, angenehm ist es aber auch dort nicht:

<https://blog.fefe.de/>

HTML ist nicht nur nervig zu lesen, es ist auch nervig zu schreiben. Außerdem ist es fehleranfällig; wenn geöffnete Tags nicht in der richtigen Reihenfolge wieder geschlossen werden, ist das Dokument fehlerhaft. Dies liegt auch daran, dass HTML nicht mehr ganz neu ist; es ist von 1992. Damals sind nur wenige Leute in die Verlegenheit gekommen, Internetseiten zu gestalten, und diese Leute waren hauptsächlich an Universitäten unterwegs. Es hat viel ermöglicht, aber es hat sich als ungeeignet gezeigt, das Erzeugen von Webinhalten der breiten Masse zu ermöglichen.

Wikitext

Einen wichtigen Durchbruch in der Nutzung und Verbreitung des Netzes haben Wikis gebracht. Das bekannteste Wiki ist die Wikipedia. Sie war jedoch weder das erste Wiki, noch ist sie der Namensgeber.

Wiki bedeutet *schnell*. Der beschreibt die Grundidee, die hinter dem Konzept des Wikis steht, und die der Grund für deren Erfolg ist. Ein Wiki präsentiert ein Dokument in für den Menschen gut lesbarer Form. Dazu bietet es direkt eine einfache Funktion, mit der dieser Inhalt einfach und unproblematisch bearbeitet werden kann.

Wikis senken die Hürde, einen Text zu bearbeiten (also ihn zu erweitern oder kurz zu berichtigen) enorm. Zusätzlich werden dort Sprachen benutzt, die zugänglicher und besser lesbar sind, als HTML. Hierdurch wird es sehr viel mehr Menschen möglich, Inhalte im Web zu erzeugen.

Die Syntax für Wikitext ist nicht wirklich einheitlich über die verschiedenen Wikis, die es gibt. Als Beispiel sei hier der Wikitext der Wikipedia herangezogen. Hier ist ein Auszug aus dem Wikipediaartikel über Wikis:

=== Vorläufer ===

Einer der ersten Vorläufer des Wikis war das in der [[Carnegie-Mellon University]] 1972 entwickelte [[ZOG (Hypertext)|ZOG]]-Datenbanksystem, das für mehrere Nutzer ausgelegt war und die Daten in strukturierten Textrahmen darstellte; verbunden waren sie durch [[Hyperlink]]s.

Dieses System wurde 1981 von [[Donald McCracken]] und [[Robert Akscyn]] zum [[Knowledge Management System]] (KMS) erweitert, bei dem Änderungen an den Datenblättern im gesamten Netzwerk sofort sichtbar wurden. In diesem System waren bereits Grafiken und Bilder integrierbar, auch sie konnten mit Hyperlinks versehen werden.

Diese Markupsprache legt den Fokus auf einen lesbaren Text. Insbesondere sind Links dezenter umgesetzt, mit eckigen Klammern: [[Robert Akscyn]]. Dies stört den Textfluss deutlich weniger als die ungefähre HTML-Entsprechung `Robert Akscyn`. Es ist auch weniger anfällig für Fehler; bei HTML kann ein einziges falsches Zeichen dazu führen, dass ein komplettes Dokument unlesbar wird.

Web 2.0

Wikis und andere Weiterentwicklungen des WWW werden unter dem Begriff Web 2.0 zusammengefasst. Hierzu gehören neben Wikis auch Blogs und Social Media. Die entscheidende Entwicklung ist hier, dass die breite Masse der Internetnutzer vom reinen Konsumenten erhoben wurde zum Inhalte schaffenden. Dies wurde auch dadurch ermöglicht, dass simple Sprachen genutzt werden, um Inhalte zu erzeugen, die dann von Software in HTML umgesetzt werden. Es sind dabei viele oft ähnliche aber doch unterschiedliche Sprachen entstanden.

Markdown

Eine simple Auszeichnungssprache, die sich besonders in der Entwicklerszene durchgesetzt hat, ist Markdown. Sie zeichnet sich dadurch aus, dass sie auch im Quelltext gut zu lesen ist. Sie ist ursprünglich dazu gedacht, in HTML übersetzt zu werden.

Ein kleines Beispieldokument in Markdown:

Eine Überschrift erster Ordnung

Dieser Text zeigt ein kleines Markdowndokument.

Absätze

Ein zusammenhängender Text wird als ein Absatz verstanden. Einzelne Zeilenumbrüche spielen hierbei keine Rolle. Andere Whitespaces, wie z.B. Leerzeichen werden immer nur als ein trennendes Zeichen ausgegeben.

Nach einer Leerzeile beginnt ein neuer Absatz. Es kann auch mehr als eine Leerzeile eingefügt werden; trotzdem entsteht nur ein neuer Absatz.

Einfache Formatierung

Texte können einfach **kursiv**, ****fett**** oder auch *****fett** und *kursiv**** gestaltet werden. Hierdurch kann man Teile betonen oder hervorheben.

Listen

Listen werden in einer intuitiven Form erzeugt:

1. Der erste Punkt komm hier
2. Ein zweiter Punkt folgt
 1. Dieser Punkt ist ein Unterpunkt von 2.
3. Jetzt reicht es aber auch mal mit der Liste.

Es ist noch einiges mehr in Markdown möglich. Dieses gesamte Dokument wurde in Markdown erstellt.

Auch innerhalb von Markdown gibt es einen Wildwuchs an Erweiterungen und Standardisierungen. Die Grundlagen sind mehr oder weniger gleich. Manche Varianten unterstützen z.B. auch Tabellen.

Viele Blogs und viel Dokumentation ist in Markdown geschrieben - auch wenn man das häufig nicht sieht, da man nur den erzeugten HTML-Text kennt.

Markdown ermöglicht ein schnelles, strukturiertes Schreiben von Texten. Man kann darin z.B. auch Notizen zu einem Meeting führen.

Wenn ein Text einmal in einer strukturierten Form vorliegt, kann man ihn gut mit Computerprogrammen weiterverarbeiten.

Übersetzen

Dieses Markdowndokument beschreibt die Eigenschaften und Vorteile einer Markupsprache. Es sollte alleinstehend als reine Textdatei lesbar und verständlich sein. Der eigentliche Inhalt, der Text, nimmt die Hauptrolle ein. Für Menschen, die es nicht gewohnt sind, Quellcode zu lesen, ist dies sicher irritierend, aber das wichtigste, der Text, sollte auch ihnen zugänglich sein. Die Gliederung erschließt sich dem neugierigen Leser hoffentlich nach dem Lesen einiger Markdowndokumente von selbst zumindest im Groben.

Wer dieses Dokument in einem modernen Texteditor mit Markdownverständnis öffnet (oder in github betrachtet), dürfte sogar schon auf den ersten Blick verstehen, welche Vorzüge sich aus dem Definieren der Gliederung ergeben.

Um zu zeigen, was möglich ist, wenn der Computer die Gliederung des Textes versteht, sind der Datei noch einige Übersetzungen beigelegt. Diese wurden ohne weiteres eingreifen mit einem Computerprogramm aus diesem Markdowndokument als Quelltext erzeugt.

Das Programm, das für das Generieren genutzt wurde, heißt Pandoc. Es ist ein mächtiges Werkzeug, das strukturierte Dokumente über verschiedenste Formate hin und her konvertieren kann. Im Folgenden werden die Befehle dokumentiert, mit denen verschiedene Formate erzeugt wurden.

PDF

Das Format, das heute jeder öffnen kann. Wir können unser Markdowndokument ohne weitere Angaben einfach in pandoc reinwerfen. Das Ergebnis ist ein brauchbares PDF-Dokument: lektion.pdf

```
# Erzeugen eines einfachen PDFs:  
pandoc Lektion.md -o lektion.pdf
```

Da das Programm die Gliederung des Textes versteht, ist es möglich, diese zu Nutzen. Wir können sehr einfach ein Inhaltsverzeichnis (Table of Contents, TOC) einfügen: lektion-toc.pdf

```
# Erzeugen eines PDFs mit Inhaltsverzeichnis:  
pandoc Lektion.md -o lektion-toc.pdf --toc
```

Das Standarddesign für PDFs aus pandoc sieht etwas dröge aus; es ist das übliche Design von LaTeX, welches im Hintergrund benutzt wird. Das Design ist jedoch gar nicht Teil des Dokuments; wir folgen dem Prinzip der Trennung von Inhalt und Design. Im Markdowndokument ist der (strukturierte) Inhalt definiert worden. Die Darstellung wird an anderer Stelle definiert. Wir können mit ein paar einfachen ein Template für die Darstellung wählen (und an unsere Wünsche leicht anpassen). Das Ergebnis liegt in lektion-eisvogel.pdf.

```
# Erzeugen eines PDFs mit Inhaltsverzeichnis und Designvorlage:
pandoc Lektion.md -o lektion-eisvogel.pdf --toc --template eisvogel
↪ -Mttitlepage -Mtoc-own-page
```

HTML

Wir können das Dokument auch als HTML-Seite auf unserem Webserver ausliefern: lektion.html

```
# Erzeugen einer HTML-Seite:
pandoc Lektion.md -s -o lektion.html
```

Auch bei HTML können wir wieder Templates nutzen, um das Design zu beeinflussen: lektion-easy.html

```
# Installieren der genutzten Templates in ~/.pandoc/templates`
curl 'https://raw.githubusercontent.com/ryangroose/easy-pandoc-
↪ templates/master/copy_templates.sh' |
↪ bash
# Erzeugen einer HTML-Seite mit Designvorlage:
pandoc Lektion.md -s -o lektion-easy.html --template easy_template --toc
```

Weitere Formate

Pandoc unterstützt noch andere Formate. Hier noch ein paar weitere Beispiele. Dies soll andeuten, welche mächtigen Werkzeuge einem zur Verfügung stehen, wenn ein Dokument so geschrieben ist, dass der Computer die Gliederung versteht.

- Epub E-Reader Buch: lektion.epub
- Office open document text: lektion.odt
- MediaWiki Wikitext (Wikipedia): lektion.wiki

```
# Als epub E-Book rendern:
pandoc Lektion.md -o lektion.epub --toc
# Als Open Document Text (Office) rendern:
pandoc Lektion.md -o lektion.odt
# Als Mediawiki Wikitext (Wikipedia) rendern:
pandoc Lektion.md -o lektion.wiki -t mediawiki
```