

# Interpretation AsciiDoc

Nico Ulsamer, Christina Frei

Version 1.0

# Inhaltsverzeichnis

Vorwort .....	1
Codebeispiele .....	2
Vergleich zu Microsoft Office Word 2010 .....	5
Editoren .....	8
Konvertierung von .adoc in andere Formate .....	10
Anwendungsbeispiele im Betrieb .....	12
Bewertung von AsciiDoc .....	13
Vorteile .....	13
Nachteile .....	13
Fazit .....	15

# Vorwort

Im Zuge unseres Deutschunterrichts haben wir uns der Analyse der Markup Sprache *Asciidoc* gewidmet. Hierzu haben wir uns mit der Syntax, verschiedenen Asciidoc Editoren und der Konvertierung von adoc in andere Formate beschäftigt sowie einen direkten Vergleich zu dem gängigen Programm Microsoft Office Word 2010 gezogen.

Zu Beginn möchten wir an *Asciidoc* heranzuführen, indem wir einige Codebeispiele erläutern.

Zur Erarbeitung der Syntax von Asciidoc und zur Erstellung dieses Dokuments wurden die Editoren AsciidocFX und Atom, sowie GitHub verwendet.

# Codebeispiele

Mit den folgenden Syntaxkenntnissen kann man bereits die Basics in Sachen Formatierung umsetzen.

- Überschriften

Ein einfaches Gleichheitszeichen bewirkt, dass der darauffolgende Text als Hauptüberschrift angesehen und dementsprechend fett geschrieben wird sowie die Schriftgröße höher ausfällt. Je nachdem wie viele Gleichheitszeichen man benutzt, desto stärker wird dieser Effekt reduziert.

Vergleichbar ist diese Funktion mit den Überschriften in HTML, denn `<h1></h1>` (HTML) verhält sich wie ein `=` (AsciiDoc), sowie `<h2></h2>` zwei Gleichheitszeichen `==` entsprechen würde.

Eine weitere Möglichkeit, Überschriften zu setzen, ist das Setzen von Minuszeichen unter das Geschriebene, welches als Überschrift hervorgehoben werden soll. Die Anzahl der hier benötigten Minuszeichen ist abhängig von der Länge des als Überschrift zu markierenden Wortes. Um unter den verschiedenen Überschriften Unterüberschriften zu setzen, muss man für die erste Unterüberschrift mehrere Tilde setzen, für die zweite Unterüberschrift mehrere Zirkumflexe und für die dritte Unterüberschrift mehrere Pluszeichen. Dies ist auch die maximal mögliche Anzahl an Unterüberschriften, welche in der Standardkonfiguration von AsciiDoc verwendet werden kann.

Syntax:

```
Erste Überschrift
-----
Unterüberschrift 1
~~~~~
Unterüberschrift 2
^^^^^^^^^^^^^^^^
Unterüberschrift 3
+++++
```

- Aufzählungen

Indem man am Zeilenanfang einen Punkt setzt beginnt man eine Aufzählung. Diese Durchnummerierung wird solange durchgeführt, bis die nächste Zeile nicht mehr mit einem Punkt anfängt. So startet, wie in nachfolgendem Beispiel gut erkennbar, die erste Zeile mit Punkt am Anfang mit einer 1 und die Darauffolgende mit einer 2 und so weiter.

1. Erster Aufzählungspunkt
2. Zweiter Aufzählungspunkt

Text

1. Erster Aufzählungspunkt

Code hierfür:

- . Erster Aufzählungspunkt
- . Zweiter Aufzählungspunkt

Text

- . Erster Aufzählungspunkt

## • Doctype

In AsciiDoc besteht die Möglichkeit, durch das Bestimmen eines sogenannten "Doctypes", ein bereits voreingestelltes Format für das Dokument zu wählen. Hier besteht die Möglichkeit, das vordefinierte Format "book", "article", oder "manpage" zu verwenden. Im book-Format sind, wie bei einem Buch üblich, die Seitenzahlen jeweils auf der rechten, und daraufhin auf der linken Seite, es besteht zudem die Möglichkeit ein Vorwort, ein Nachwort, ein Glossar und Fußnoten hinzuzufügen. Bei dem Doctype article handelt es sich um ein Format, welches für kurze Artikel oder Dokumentationen verwendet werden kann. Der Doctype manpage wird verwendet um UNIX Handbücher zu schreiben.

Die Syntax, um den Doctype zu setzen ist folgende:

```
:doctype: book  
:doctype: article  
:doctype: manpage
```

Jeder der drei Doctypes hat seine Vor- und Nachteile, wir verwendeten vor allem aufgrund dem optischer Aspekte den Doctype book, da dieser von der Formatierung am ansprechendsten wirkte.

## • Table of Content

Eine weitere interessante und vorallem wichtige Funktion von AsciiDoc ist das Einbinden eines Table of Contents (zu Deutsch: Inhaltsverzeichnis). Dieses erkennt, sobald gesetzt, die einzelnen Überschriften und ermöglicht durch Verlinkungen in dem fertigen Dokument (PDF, Docbook) zu den entsprechenden Kapiteln zu springen. Außerdem bietet es dem Leser einen Überblick über die verschiedenen Kapitel. Unterüberschriften werden im Table of Content leicht eingerückt dargestellt. Der Table of Content kann entweder links oder rechts im Dokument platziert werden. Die Syntax, um das Inhaltsverzeichnis mit benutzerdefiniertem Namen und individueller Ausrichtung einzufügen, ist folgende:

```
:toc:  
:toc: right  
:toc-title: Name des Table of Content
```

## • Codesnippets mit Built-in Syntax Highlighting

In AsciiDoc hat man (was vor allem zur Dokumentation von Programmen von Nutzen ist) die Möglichkeit, Code mit voreingestelltem Syntax-Highlighting einzufügen. Hierzu muss der Autor lediglich die verwendete Programmiersprache deklarieren, um das zugehörige Syntax-Highlighting

zu erhalten.

So sieht folgender Code..

```
public class HelloWorld
{
    public static void main (String[] args)
    {
        System.out.println("Hello World!");
        int beispielwert = 3;
    }
}
```

... mit Syntax-Highlighting so aus:

```
public class HelloWorld
{
    public static void main (String[] args)
    {
        System.out.println("Hello World!");
        int beispielwert = 3;
    }
}
```

# Vergleich zu Microsoft Office Word 2010

Heutzutage werden Texte am Computer überwiegend mit Textverarbeitungsprogrammen wie Microsoft Office Word (im Folgenden der Einfachheit halber "Word") verfasst. Der Vorteil solcher Programme gegenüber Markup Sprachen wie AsciiDoc scheint auf den ersten Blick eindeutig, nämlich die einfache und sofortige Nutzung selbiger, ohne zuvor Kenntnisse über das Programm zu haben. Doch in Word muss der Benutzer alle Formatierung händisch selbst erledigen, während AsciiDoc dies schlicht durch voreingestellte Formatierung vereinfacht, die einem bereits einen Teil der Arbeit abnimmt. Darüber hinaus steht man als Benutzer von Word teilweise vor Rätseln. So werden animierte Bilder wie GIFs plötzlich zu Standbildern, wobei einfach das allererste Bild des GIF zur Darstellung hergenommen wird.

Im Folgenden der Beweis, dass AsciiDoc dahingegen die Funktion, ein GIF einzubinden, bietet:

[AsciiDoc Gif]

Bereits in der Anschaffung gibt es einen gewaltigen Unterschied. Während man, um AsciiDoc nutzen zu können, nur einen geeigneten Editor herunterladen muss (was nicht mal zwingend notwendig ist, da man im Grunde jeden bereits auf dem Computer vorinstallierten Editor nutzen kann), muss man für Word zunächst eine Lizenz kaufen. Diese Lizenz kostet für die aktuelle Version, für einen PC im privaten Gebrauch 150 € und ist damit recht teuer. Hierbei muss allerdings erwähnt werden, dass mit dieser Lizenz nicht nur Word sondern auch alle anderen Office Produkte zur Verfügung stehen.

Nachdem man Word gekauft, heruntergeladen und installiert hat steht es einem direkt zur Verfügung. Für AsciiDoc benötigt man noch eine kleine Einarbeitungszeit, die mit der Syntax zusammenhängt. Diese Zeit kann durchaus Spaß machen, indem man sich die Syntax spielerisch mit Hilfe von Tutorials aneignet. Zu empfehlen sind unserer Meinung nach folgende Cheatsheets/Tutorials: <http://asciidoctor.org/docs/asciidoc-writers-guide/> und <http://powerman.name/doc/asciidoc>.

Während wir uns mit der Syntax beschäftigt haben ist uns unter anderem die Einfachheit der Darstellung komplexer mathematischer Formeln aufgefallen (siehe [Codebeispiele](#)). Natürlich lassen sich diese auch mit dem Formeleditor in Word erstellen, allerdings bietet dieser nur ein Grundgerüst an Zeichen, weshalb besonders im professionellen Bereich gerne Markup Sprachen zum Verfassen solcher Dokumente benutzt werden. Außerdem ist uns aufgefallen, dass man in AsciiDoc zwischen drei verschiedenen Dokumentarten wählen kann. Diese nennen sich "article" - Artikelformat, "manpage" - Handbuchformat und "book" - Buchformat (wird auch in diesem Dokument verwendet). Zum Beispiel werden die Seiten im Buchformat standardmäßig abwechselnd unten links und unten rechts durchnummeriert, sodass man hinterher ein Buch daraus binden könnte. Diese Doctypes sind praktisch, denn je nachdem, welches Format man wählt, ändert sich die Formatierung, weshalb man sich im Voraus Gedanken darüber machen muss, welches der Formate am Besten zu dem zu erstellenden Dokument passt. Dadurch muss man sich im späteren Verlauf wiederum weniger Gedanken um die Formatierung machen, wodurch man sich letzten Endes besser auf die Formulierung des Inhalts konzentrieren kann. Zwar gibt es in Word eine ähnliche Funktion, nämlich sogenannte Formatvorlagen, die man auch selbst anpassen oder völlig neu erstellen kann, aber sie bieten einem nicht dieselbe "Intelligenz" wie AsciiDoc. So muss man seine Überschriften und alle anderen Textelemente in Word trotzdem mühsam per

Hand formatieren. AsciiDoc bietet, wenn auch nur indirekt, sogar den Verfassern musikalischer Notationen Vorteile. Denn es ist mit dem, in den AsciiDoc Filtern enthaltenen, Python Script `music2png.py` möglich, diesen Schriftblock

```
["music","music1.png",scaledwidth="100%"]
-----
T:The Butterfly
R:slip jig
C:Tommy Potts
H:Fiddle player Tommy Potts made this tune from two older slip jigs,
H:one of which is called "Skin the Peelers" in Roche's collection.
H:This version by Peter Cooper.
D:Bothy Band: 1975.
M:9/8
K:Em
vB2(E G2)(E F3)|B2(E G2)(E F)ED|vB2(E G2)(E F3)|(B2d) d2(uB A)FD:|
|:(vB2c) (e2f) g3|(uB2d) (g2e) (dBA)|(B2c) (e2f) g2(ua|b2a) (g2e) (dBA):|
|:~B3 (B2A) G2A|~B3 BA(uB d)BA|~B3 (B2A) G2(A|B2d) (g2e) (dBA):|
-----
```

... in einen anschaulichen Notenblock in Form eines png-Bildes zu verwandeln:

[\[Musical Notation\]](#)

*Dieses Beispiel stammt aus der AsciiDoc Dokumentation.*

Diese Umwandlung wird durch den vorangestellten Filter "music" erreicht. Man hat zudem noch die optionalen Möglichkeiten einen Dateinamen für das Ausgabe-png festzulegen "music1.png" sowie die Größe durch "scaledwidth" zu bestimmen. In Word haben wir kein vergleichbares Mittel gefunden, um Notenblätter zu erstellen. Der einzige Weg wäre eine Unicode Schriftart (z.B. "Fughetta") mit entsprechendem Zeichensatz herunterzuladen, um damit zumindest Noten schreiben zu können. Das Endergebnis, das mit Hilfe von AsciiDoc erreicht wird übertrifft jedoch höchstwahrscheinlich das von Word.

In allen sonstigen Bereichen, wie zum Beispiel im Erstellen von Tabellen, bei der Formatierung von Schrift, beim Einfügen von Fußnoten, Bildern oder Diagrammen finden sich kaum nennbare Unterschiede zwischen den beiden Textverarbeitungsformen.

Trotz allem rentiert es sich ein herkömmliches Schreibprogramm auf dem Computer zu haben, da es in AsciiDoc kein Brief-Format gibt, mit welchem man einfach und schnell einen Brief schreiben kann. Hier würde man sich durch die Einrückungen mit AsciiDoc nur unnötige Umstände machen, da man zusätzlich Stylesheets einbinden müsste.

Alles in allem kann man sagen:

Dafür, dass Word ein kostenpflichtiges, professionell programmiertes und vielgenutztes Programm ist, steht AsciiDoc dem in Nichts nach und bietet zudem noch einige Extras, die man mit Word nicht oder nur zum Teil umsetzen kann. Für den Alltag sind beide Formen der Textverarbeitung nützlich, es kommt grundsätzlich auf die Dokumentenart an, die man erstellen will. So eignet sich AsciiDoc



mehr für Dokumentationen, Anleitungen, Bücher, Zeitschriftenartikel, Blogeinträge und ähnliche Formate, während Word für Briefe und kurzfristig zu erstellende Texte geeignet ist.

# Editoren

Welchen Editor man benutzt bleibt jedem selbst überlassen. Es gibt zahlreiche Webeditoren und Programme, die sich eignen, um AsciiDoc Dateien zu verfassen. Man sollte sich bei der Auswahl des Editors mit Bedacht auf die eigenen Kenntnisse und den Anwendungszweck für den Geeignetesten entscheiden. Wir haben die nachstehenden drei frei verfügbaren Editoren getestet.

- **AsciiDocFX**

Dieser Editor ist für alle Plattformen, ob Windows, Linux oder Mac verfügbar. Man kann ihn einfach [hier](#) herunterladen und ohne Umschweife loslegen. Der Editor besitzt zwar kleine Mängel, so kann man beispielsweise kein €-Zeichen eintippen, sondern muss es über Copy & Paste einfügen, aber er ist vor allem für Anfänger brauchbar. So finden sich am oberen Rand altbekannte Steuerelemente von Word. Das erleichtert die Einführung in die Syntax von AsciiDoc, da man zu Beginn einfach spielerisch ausprobieren kann. Klickt man auf den Button für *Bold* fügt einem der Editor zweimal zwei Sternchen ein, was in der AsciiDoc Syntax die Formatierung zum Schreiben von fett gedruckten Wörtern ist - damit wird die Syntax teilweise selbsterklärend. Etwas vorsichtig muss man jedoch bei den Formaten sein, denn beim Speichern als PDF-Datei kann es durch die Konvertierung in Zwischenformate zu teils fehlerhaften Formatierungen kommen.

Ein zu erwähnendes hilfreiches Feature ist noch, dass man durch einen Klick auf *Browser* direkt in die Dokumentansicht im Browser gelangt, sprich das Dokument wird in HTML konvertiert und im Browser geöffnet. Das erleichtert vor allem der Erstellen von Dokumenten, die für das Internet vorgesehen sind.

Dank dieser Spielereien, dem built-in Syntax Highlighting & der Live Preview sowie den vielen möglichen Ausgabeformaten (HTML, PDF, Ebook und DocBook) ist dieser Editor für Anfänger ideal.

- **Atom**

Der Editor [Atom](#) ist ebenfalls für alle Plattformen verfügbar. Für die Live Preview muss man zusätzlich ein package installieren, welches man [hier](#) findet. Dafür gibt man einfach in die Kommandozeile folgenden Befehl ein:

```
apm install asciidoc-preview
```

Dadurch wird beim erneuten Start von Atom in der Menüleiste unter "Packages" der Punkt "AsciiDoc Preview" verfügbar. Öffnet man also eine .adoc Datei und klickt unter dem eben beschriebenen Menüpunkt auf "Toggle Preview" (oder nutzt die Tastenkombination Ctrl+Shift+A) öffnet sich ein zweites Fenster mit der Live Preview. Man hat auch die Möglichkeit noch mehr Packages zu installieren, so gibt es unter anderem auch eines für das Syntax-Highlighting und für die Autovervollständigung von Formatierungsbefehlen. Atom würde ich eher fortgeschritteneren Benutzern empfehlen, da man nicht direkt - wie bei AsciiDocFX - loslegen kann, es keine Steuerelemente zum Formatieren gibt und die Einrichtung des Editors an sich aufwändiger ist.

- **AsciiDocLIVE**

Es gibt auch Webeditoren, wie [AsciiDocLIVE](#), die den Vorteil haben, dass man keine Software auf

dem Computer installieren muss und trotzdem eine Live Preview bekommt. Syntax Highlighting wird ebenfalls unterstützt. Auch kann man sein Dokument direkt in Dropbox, GoogleDrive oder lokal auf dem Computer speichern. Zudem wird einem die Möglichkeit geboten, das Geschriebene ins HTML-Format zu konvertieren. Für den schnellen Einsatz ist der AsciiDocLIVE Webeditor also durchaus praktikabel.

Selbstverständlich kann man auch mit jedem auf Computern vorinstallierten Editor .adoc Dateien erstellen, wobei man jedoch auf die Kodierung des Zeichensatzes Acht geben muss. Mit Notepad++ und der UTF-8 Kodierung haben wir beispielsweise ein zufriedenstellendes Ergebnis erreicht.

# Konvertierung von .adoc in andere Formate

Um AsciiDoc in mehr Arbeitsbereichen verwenden zu können ist es durchaus nötig, dass man die mit AsciiDoc erstellten Dokumente auch in andere Formate umwandeln kann. Vor allem das beliebte PDF-Format ist wichtig. Dieses ist beispielsweise praktisch, um seine Arbeit zu verbreiten, ohne dass die Leser sich extra einen Editor mit Live Preview herunterladen oder den Text in einen Webeditor kopieren müssen.

- **PDF-Format**

Es gibt mehrere Möglichkeiten eine .adoc Datei in das .pdf Format umzuwandeln. Die meiste Software verwendet hier jedoch Zwischenformate wie das DocBook, um letztlich eine PDF daraus zu erstellen. Mit dem eigens für AsciiDoc programmierten [AsciidoctorPDF](#) lassen sich .adoc Dateien unmittelbar in PDF's konvertieren. Für die Nutzung von AsciidoctorPDF muss man sich Ruby herunterladen und dessen Pfad den Umgebungsvariablen hinzufügen. Die aktuelle Version für Ruby findet man [hier](#). Man braucht den RubyInstaller und das passende DevKit, wie auch auf der Seite erklärt ist. Nachdem Ruby installiert ist kann man der Anleitung von AsciidoctorPDF folgen, beginnend bei den Prerequisites. Im Folgenden ist interessant zu wissen, dass *gems* sozusagen libraries für Ruby sind. Als nächstes folgt also dieser Befehl auf der Kommandozeile, um das gem *prawn* zu installieren:

```
gem install prawn --version 2.1.0
```

Nun benötigt man nur noch das gem *asciidoctor-pdf*:

```
gem install asciidoctor-pdf --pre
```

Schon kann man über diesen Befehl

```
asciidoctor-pdf path\to\adocfile.adoc
```

... eine .adoc Datei schnell und ohne Umwege in eine PDF konvertieren. Leider mussten wir feststellen, dass sobald ein animiertes Bild in der .adoc Datei vorhanden ist, ein weiteres gem installiert werden muss, welches sich *prawn-gmagick* nennt.

Zwar unterstützt auch der Editor AsciiDocFX die Konvertierung ins PDF Format, hiervon raten wir allerdings ab, denn durch das vorherige Umwandeln in ein Zwischenformat gerät die Formatierung unter Umständen in Mitleidenschaft.

- **HTML-Format**

Der Vorteil von HTML-Dateien ist die erleichterte Einbindung der .adoc Dateien in Webseiten. Zudem kann man das Dokument einfach in jedem beliebigen Browser öffnen und lesen. Die Konvertierung in eine .html Datei geht am leichtesten. Sogar der Webeditor [AsciidocLIVE](#) besitzt die Funktion "*Exportieren als.. </>HTML*". Man braucht also nur den Webeditor zu öffnen, seinen AsciiDoc Text einfügen und schließlich die eben genannte Funktion nutzen. Nachteile habe ich

hierbei keine festgestellt. Wer als Editor AsciidocFX nutzt findet dort ebenfalls einen Button zum "Speichern als HTML Datei".

- **LaTeX-Format**

Für die Konvertierung von .adoc in .tex gibt es ein hilfreiches gem namens [Asciidoctor-latex](#), genauso wie *asciidoctor-pdf* lässt es sich mit folgendem Befehl installieren:

```
gem install asciidoctor-latex --pre
```

Voraussetzung ist natürlich weiterhin, dass zuvor Ruby installiert wurde. Nun kann man seine adoc Datei über den Befehl

```
asciidoctor-latex path\to\adocfile.adoc
```

... ins LaTeX-Format konvertieren.

# Anwendungsbeispiele im Betrieb

Nachdem wir im Abschnitt [Vergleich zu Microsoft Office Word 2010](#) bereits erläutert haben, dass AsciiDoc grundsätzlich keine Nachteile gegenüber Word hat, stellt sich die Frage, warum in Unternehmen trotzdem lieber das kostenpflichtige Microsoft Office Programm verwendet wird. Im Grunde genommen könnte ein Unternehmen ohne die teuren Office-Lizenzen enorm Kosten sparen.

Der Grund ist vermutlich, dass der Aufwand, jedem Mitarbeiter eine Schulung in AsciiDoc zu ermöglichen und schließlich sicherzustellen, dass auch jeder Mitarbeiter nach der Schulung mit Hilfe von AsciiDoc Dokumente verfassen kann, viel zu groß wäre. Eine solch immense Umstellung würde bei einigen Mitarbeitern sicher zu Verständnisproblemen führen, die sich erst mit einsetzender Gewohnheit lösen würden. Insbesondere ältere Mitarbeiter, die ihr Leben lang mit Textverarbeitungsprogrammen wie Word gearbeitet haben und für Mitarbeiter ohne Programmierkenntnisse oder Affinität zu diesem Bereich wären von diesen Problemen betroffen, denn die Syntax von AsciiDoc ist auf den ersten Blick gewöhnungsbedürftig und ähnelt der von Programmiersprachen. Ein vollständiges Ersetzen von Word durch AsciiDoc macht daher in unseren Augen, trotz der Kostenersparnis, keinen Sinn.

Dennoch gibt es eine sinnvolle Möglichkeit AsciiDoc anstelle von Word in Unternehmen zu nutzen, denn es eignet sich hervorragend zum Erstellen von Dokumentationen für Programmierarbeiten. Solche Dokumentationen werden ausschließlich von Programmierern geschrieben, wodurch die Syntax von AsciiDoc und allgemein die Umgangsweise mit dieser Sprache deutlich erleichtert wird. Wir selbst können uns durchaus vorstellen, solche Dokumentationen mit AsciiDoc zu schreiben, da es einem die Arbeit in Sachen Formatierung einfach abnimmt.

# Bewertung von AsciiDoc

Zum Ende möchten wir nun nochmal die einzelnen Kapitel in die Vor- bzw. Nachteile zusammenfassen.

## Vorteile

Nur indem man das Unerreichbare anstrebt, gelingt das Erreichbare. Nur mit dem Unmöglichen als Ziel, gelingt das Mögliche.

— Miguel de Unamuno

Die Vorteile AsciiDocs liegen in vielen Bereichen. So kann man, wie in [Codebeispiele](#) gezeigt, nicht nur durch einfaches Einfügen von Programmcode und die Angabe der verwendeten Programmiersprache das zugehörige Syntax-Highlighting verwenden (was besonders für die Dokumentation von Programmierarbeiten hilfreich ist), sondern ebenfalls Zitate, Bilder und Tabellen mit Leichtigkeit einfügen. Dank der Live Preview einiger Programme kann man die verwendeten Bausteine sofort Betrachten und gegebenenfalls abändern.

Ein Weiterer, definitiv zu nennender Vorteil von AsciiDoc ist die beinahe schon kinderleicht zu handhabende Formatierung, so muss man lediglich einige Codes und Kommandos beherrschen (wie in [Codebeispiele](#) gezeigt), die es ermöglichen, den Text nach den Wünschen des Autors zu formatieren.

AsciiDoc bietet, zumindest bei der Benutzung von AsciiDocFX die Möglichkeit, Dokumente als Ebook, Docbook, HTML oder PDF auszugeben, weitere Möglichkeiten eine adoc Datei in andere Formate umzuwandeln wurden in [Konvertierung von .adoc in andere Formate](#) gezeigt.

Des Weiteren lassen sich mit ein paar Zeichen schon komplexe mathematische Formeln darstellen, so lässt sich durch Verwendung von zwei Zirkumflexen eine Potenz darstellen. Mit der Nutzung zweier Tilde kann man eine Zahl mit einem Index versehen.

Beispiel:  $x^{2^2}$  ;  $x_{2^2}$

Formatiert sieht das dann so aus:  $\mathbf{x}^2$  ;  $\mathbf{x}_2$

## Nachteile

Im Folgenden setzen wir uns mit den Nachteilen AsciiDocs auseinander. Zwei zu nennende Nachteile sind unter anderem das Fehlen einer Autokorrektur (was wiederum mit dem Editor zusammenhängt, jedoch fanden wir keinen, bei dem eine Autokorrektur vorhanden war) und dass, sobald etwas über Copy & Paste eingefügt wird (auch abhängig vom verwendeten Editor), es sofort als Java Code interpretiert wird und somit erst die zwei generierten Codezeilen gelöscht werden müssen. Schade ist zudem, dass kaum deutsche Nachschlagewerke (Cheatsheets, Tutorials) existieren. Auch zu bemängeln ist, dass in AsciiDoc leider nicht die Möglichkeit der Konfiguration vorgefertigter Zeichensätze (Länge/Lage von Pfeilen) besteht. Zu den Nachteilen zählt außerdem

die Tatsache, dass man durch versehentliche Returns oder Sonderzeichen die gesamte Formatierung verändern kann, wodurch das Dokument ungewollt anders aussieht - Natürlich ist so etwas schnell behoben, jedoch kann das bei der Verwendung von gängigen Texteditoren nicht passieren. Zusätzlich negativ aufgefallen ist, dass man beim Durchnummerieren (welches durch das Setzen von Punkten realisiert wird) direkt unter dem ersten Punkt einen weiteren Punkt setzen muss, damit die Liste logisch fortgeführt wird. Andernfalls (falls zwischen Punkt 1 und Punkt 2 ein Absatz ist) wird die Aufzählung von neuem (also wieder mit der 1 beginnend) gestartet und damit womöglich ungewollt falsch interpretiert. Der wohl größte Nachteil, der auch zur Folge hat, dass AsciiDoc für den 0815-Schreibtischjob wohl keine Anwendung finden wird ist die Tatsache, dass man sich, anders als bei Word, erst die Syntax aneignen muss, um das volle Potential AsciiDocs nutzen zu können.



# Fazit

Nach Aneignen der Syntax und ersten Rumspielereien war der Start in AsciiDoc relativ einfach und die Verwendung AsciiDocs mithilfe der zahlreichen im Internet vorhandenen Editoren kein Problem.

Für uns und viele Klassenkameraden waren Markup Sprachen etwas völlig Neues. Auch in meiner Familie kannte diese Art der Textverarbeitung niemand. Diese Tatsache ist meiner Meinung nach sehr traurig, da man mit AsciiDoc, wie wir festgestellt haben, mindestens genauso gut Texte verfassen und bearbeiten kann wie mit jedem herkömmlichen Editor auch. AsciiDoc hat, wie alles andere auch, seine Schwächen, aber auch Stärken, die es zu Nutzen gilt. Markup Sprachen, wie AsciiDoc, sollten in meinen Augen bekannter werden, da sie es allein durch ihre Funktionalität definitiv verdient haben. - Christina Frei

Nachdem wir uns die ziemlich einfach zu lernende Syntax angeeignet haben, fanden wir es sehr spannend diese auszuprobieren und einzusetzen. Meiner Meinung nach sollten Markup Sprachen deutlich mehr an Bekanntheit erlangen, vorallem im Programmiererbereich findet AsciiDoc durch die einfache Einbettung von Code sehr gut Verwendung. Auch für das Erstellen von Büchern ist AsciiDoc durch die Funktion "doctype" eine gute Lösung. Alles in allem bin ich sehr angetan von dieser Markup Sprache und denke, ich werde mich in Zukunft noch mehr durch die Möglichkeiten von AsciiDoc durcharbeiten und diese auch in Zukunft zum Erstellen von Dokumentationen verwenden. - Nico Ulsamer