

# Mit einem Text-Editor Dokumente und E-Books für Amazon Kindle erstellen

- Version: 0.92
- Datum: 14. September 2011
- Autor: [Vladimir Simović](#)
- Ort: Köln

# 0. Vorwort

Dieses PDF-Dokument basiert auf dem Versprechen welches ich im Artikel [Amazon Kindle: unser erstes WordPress-E-Book](#) abgegeben habe und den vier darauffolgenden Blog-Artikeln ([1](#), [2](#), [3](#) und [4](#)), die ich vor kurzer Zeit auf perun.net veröffentlicht habe. Allerdings habe ich dieses Dokument anschließend an ein paar Stellen geringfügig erweitert. Auf Grund dieser Tatsache, bin ich auch bei dem "du" geblieben. Da ich es ein bißchen albern fand, die übernommenen Blog-Inhalte in die sie-Form umzuwandeln.

Nach der Lektüre dieses Dokuments sollte jeder, der keine Angst vor einem Text-Editor hat, in der Lage sein ein **komplettes und vollständiges Buch für Amazon Kindle** zu erstellen.

## Warum Amazon Kindle und warum HTML?

Amazon ist ein großes, internationales Unternehmen und viele Menschen verbinden mit Amazon Bücher und vor allem Zuverlässigkeit. Auf der anderen Seite ist Kindle auch für Autoren ein interessantes Format. Man ist **nicht** auf den [Kindle-Reader](#) angewiesen. Es existieren [kostenlose Kindle-Apps](#) für eine Reihe von Geräten: Windows-PC, Mac, iPod, iPad, iPhone, Android-Geräte und für Windows Phone. Darüberhinaus kann man man mit dem sog. [Kindle Cloud Reader](#) auch in einem Browser (und somit auch auf einem Linux-Rechner) die Bücher lesen.

Durch diese "Verschmelzung" – ein großes Unternehmen mit einem positiven Image und ein breites Angebot an unterstützten Systemen & Geräten – entsteht eine interessante Autoren-Plattform. Ein weiteres Argument ist die recht einfache Produktion von digitalen Büchern für Kindle, wie du im Verlauf dieses Dokumentes erkennen wirst.

Für HTML habe ich mich sehr schnell entschieden. Immerhin basiert das Kindle-Format auf HTML. Es wäre sehr albern, hier irgendwelche Umwege über Word & Co. zu bestreiten. Editoren für HTML gibt es auf allen gängigen Systemen und viele der Tools sind komplett kostenlos. Auch HTML an sich ist auf allen relevanten Systemen vertreten und im Gegensatz zu vielen Gerüchten auch recht einfach zu erlernen, zumal du für die Produktion von Kindle-Dokumenten eh einen geringeren Satz an Elementen benötigst, wie wenn du eine Website erstellen würdest.

HTML hat aber auch einen weiteren Vorteil. Die Dateigröße der Dokumente ist kleiner, als bei den sog. Textverarbeitungsprogrammen. Man kann Inhalte, die als HTML vorliegen ohne größere Probleme [in andere Formate umwandeln](#). Ein einfaches HTML-Dokument lässt sich auch ohne Probleme in z. B. LibreOffice (bzw. OpenOffice) importieren und weiter bearbeiten. Der Weg von HTML nach Word ist **viel einfacher** als der umgekehrte Weg.

# 1. Vorbereitung und Einrichtung der Entwicklungsumgebung

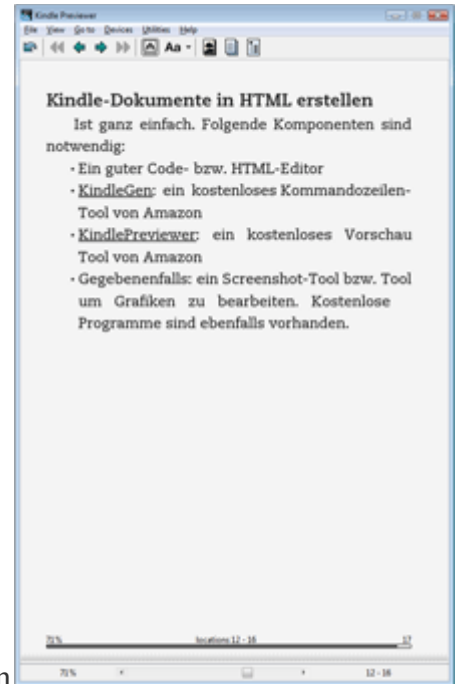
Bevor du anfangen kannst Dokumente und E-Books für [Amazon Kindle](#) zu erstellen, musst du ein paar Vorbereitungen treffen und die Entwicklungsumgebung einrichten.

## 1.1 Die Vorbereitung

Um Dokumente und E-Books für Kindle zu erstellen benötigt man keine teure Entwicklungsumgebung ... im Grunde genommen muss man kein Cent extra Geld ausgeben, wenn man es nicht möchte.

Folgende Komponenten sind notwendig:

- Ein guter Code- bzw. HTML-Editor
- [KindleGen](#): ein kostenloses Kommandozeilen-Tool von Amazon
- [KindlePreviewer](#): ein kostenloses Vorschau-Tool von Amazon
- Gegebenenfalls: ein Screenshot-Tool bzw. Tool um Grafiken zu bearbeiten. Kostenlose Programme sind ebenfalls vorhanden.



Ich persönlich nutze seit Jahren [WeBuilder als HTML-Editor](#), aber man kann ohne Probleme auch mit kostenlosen Editoren, wie zum Beispiel [PSPad](#) arbeiten.

Die Vorgehensweise ist einfach. Im HTML-Editor erstellt man das Dokument. Mit dem KindleGen von Amazon wandelt man das HTML-Dokument in Mobipocket-Format um und mit KindlePreviewer kann man das Dokument so anzeigen lassen, wie es auf einem [Amazon Kindle](#) ausschauen wird.

[Mobipocket \(.mobi\)](#) ist das Format der gleichnamigen Firma aus Frankreich und es handelt sich hierbei um ein Tochterunternehmen von Amazon.com. Daraus resultiert, dass [Amazon Kindle](#) dieses Format nicht nur unterstützt, sondern dass das eigene Format (.azw) auf Mobipocket basiert und laut [Wikipedia \(en\)](#) sind die Unterschiede zwischen den beiden Formaten marginal oder besser gesagt für uns uninteressant.

Sowohl KindleGen wie auch KindlePreviewer sind kostenlos und es gibt sie für Windows und Mac. KindleGen existiert auch in einer Version für Linux, KindlePreviewer leider nicht. Herunterladen kann man *beide* Tools auf [Kindle Publishing Programs](#).

```
C:\ Eingabeaufforderung
*****
* Amazon.com kindlegen(Windows) U1.2 build 33307 *
* A command line e-book compiler *
* Copyright Amazon.com 2011 *
*****

Info(prcgen): Added metadata dc:Title "Ich bin der Seitentitel und ersche
ine oben im Browser"
Info(prcgen): Parsing files 0000002
Info(prcgen): Resolving hyperlinks
Info(pagemap): No Page map found in the book
Warning(prcgen): Cover not specified
Info(prcgen): Computing UNICODE ranges used in the book
Info(prcgen): Found UNICODE range: Basic Latin [20..7E]
Info(prcgen): Found UNICODE range: Latin-1 Supplement [A0..FF]
Info(prcgen): Found UNICODE range: Latin Extended-A [100..17F]
Info(prcgen): Building MOBI file, record count: 0000001
Info(prcgen): Final stats - text compressed to (in % of original size): 054.62%

Info(prcgen): The document identifier is: "Ich_bin_der_-ben_im_Browser"
Info(prcgen): The file format version is U5
Info(prcgen): Saving MOBI file
Info(prcgen): MOBI File generated with WARNINGS!

D:\Programme\KindleGen>
```

KindleGen im Einsatz

## 1.2 Einrichtung der Umgebung

Ich geh jetzt davon aus, dass du einen Code-Editor und ein Grafikprogramm bereits installiert, eingerichtet und angepasst hast.

Bei KindleGen muss man nichts installieren, man muss lediglich Das Paket dort entpacken bzw. entzippen wo man es auf der Festplatte haben möchte. Um das Programm zu starten muss man die Kommandozeile (aka DOS-Box, aka Eingabeaufforderung etc.) starten und dort geht man mit Angabe des Laufwerks, z. B. d: und des Befehls `cd - z. B. cd Meine-Programme\Kindlegen` – in den Ordner rein wo sich die *kindlegen.exe* befindet.

Von hier aus wird man dann später die Buch-Datei aufrufen um sie umzuwandeln. Die Vorgehensweise bei Linux und Mac ist nahezu identisch.

Der KindlePreviewer wird "normal" installiert und die generierte Mobipocket-Datei kann man dann u.a. mit einem Rechtsklick in KindlePrevier öffnen oder man öffnet die Anwendung und geht dann den "klassischen" Weg über *File → Open Book* oder man schiebt einfach aus dem Explorer die .mobi-Datei in den Previewer rein.

Auch wenn man selber einen [Amazon Kindle](#) hat, so empfiehlt es sich denn dennoch das Vorschau-Tool zu installieren. Es ist viel komfortabler die Software aufzurufen als bei jeder kleiner Änderung die Datei an den Kindle zu schicken.

**Hinweis:** wenn du auf dem Rechner die [Kindle-App](#) installiert hast, dann kannst du die Mobipocket-Datei auch dort anschauen.

## 1.3 Die nächsten Schritte

Nach dem du dir jetzt Entwicklungsumgebung eingerichtet und dich mit der vertraut gemacht hast, wirst du im nächsten Abschnitt aus einer HTML-Datei dein erstes Kindle-Dokument erstellen. Doch bevor wir so weit sind, wollte ich noch ein paar Hinweise und Anmerkungen geben.

Ein Tipp an die jüngeren Kollegen. Vergesst bitte HTML5 und CSS3 und versucht euch an HTML 4 oder XHTML 1.0 und CSS 1 zu erinnern. Ältere Kollegen sollten sich einfach in die Zeit 1998-2002 zurückversetzen. 😊 Das ganze ist ein bißchen scherzhaft gemeint. Der HTML- und CSS-Umfang ist sehr beschränkt, aber das ist OK und völlig ausreichend: wir wollen hier Bücher und keine dynamische Webportale produzieren.

Ich weiß, es existieren diverse Tools, die das problemlose Umwandeln von z. B. Word oder PDF in Kindle versprechen. Leider bleibt es sehr häufig bei dem Versprechen und zudem ist die Methode, die ich beschreibe, die "sauberste" und produziert die kleinsten Dateigrößen. Wer jetzt nicht weiß, was ich mit sauber meine, der kann sich gerne anschauen was Word produziert, wenn man ein Dokument als HTML abspeichert.

Die Dateigröße des E-Books ist deswegen wichtig, weil man als Autor den Download-Traffic der E-Books – wenn man sie auf Amazon veröffentlicht – bezahlt und zwar mit **12 Eurocent pro Megabyte** und zwar pro Verkauf.

## 2. Das erste Kindle-Dokument

Nach dem du im ersten Kapitel die Entwicklungs- und Testumgebung eingerichtet und dich mit ihr vertraut hast, geht es nun darum das erste Kindle-Dokument zu erstellen.

Erstelle eine leere HTML-Datei und füge dort ein Grundgerüst ein:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">

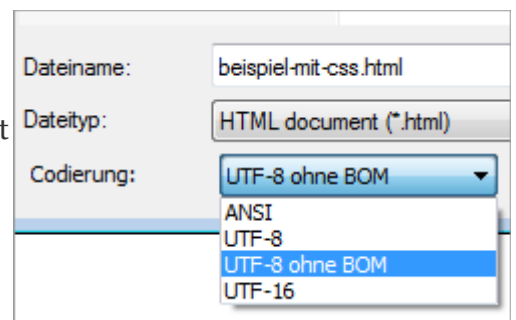
<head>
  <title>Titel deines E-Books</title>

  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <meta name="author" content="Dein Name" />
</head>
<body>

</body>
</html>
```

Ich persönlich habe im konkreten Fall mit [XHTML 1.0 transitional](#) gearbeitet. Du kannst mit der `strict`-Variante arbeiten, auf HTML 4 oder 5 ausweichen oder komplett auf den Doctype verzichten. Ich habe mich dennoch für einen Doctype entschieden, man muss sich ja nicht unnötigerweise eine Fehlerquelle verschaffen, auch wenn sie nur hypothetischer Natur sein könnte.

**Hinweis an Einsteiger:** Im `head` bzw. in den `meta`-Angaben bestimmt man den Zeichensatz und gibt den Namen des Autors bzw. der Autoren an. Wenn man Dokumente für Kindle produziert sollte man wie auch sonst darauf achten, dass die `meta`-Angaben zum Zeichensatz und die Einstellung wie die Dateien beim abspeichern codiert werden übereinstimmen. Wenn du mit dem westeuropäischen Zeichensatz (`iso-8859-1`) arbeitest, dann musst du deine Dateien auch entsprechend abspeichern: als [ANSI](#). Wenn du mit `utf-8` arbeitest, dann musst du deine Dateien auch als `utf-8` abspeichern, da es sonst zu Konflikten kommt.



### 2.1 Der erste Inhalt

Nun hast du die HTML-Datei mit Grundgerüst erstellt und jetzt geht es darum, den ersten Inhalt einzufügen. im PDF-Dokument [Amazon Kindle Publishing Guidelines](#) (pdf, 1,84 mbyte) kannst du auf den Seiten 32 und 33 sehen welche HTML-Elemente unterstützt werden. Viele HTML-Elemente werden

nicht unterstützt und die, die unterstützt werden dann auch nicht immer im vollen Umfang.

Aber wie ich schon im vorherigen Artikel erwähnt, ist die Unterstützung ausreichend um ein Buch zu produzieren. Immerhin sind die wichtigen Elemente um ein Text auszuzeichnen – Überschriften, Absatz, Listen, Zitat und Tabellen – und den Inhalt mit zusätzlichen Inhalten anzureichern (Bilder, Video und Audio) vorhanden.

Hier eine HTML-Datei. Ich weiß, der Inhalt ist nicht sonderlich geistreich, 😊 aber der Inhalt ist momentan nebensächlich:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">

<head>
  <title>Wir testen das Kindle-Format</title>

  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <meta name="author" content="Dein Name" />
</head>
<body>

<h1>Die erste Überschrift</h1>

<p>Nun zwei Absätze. Swine ribeye shoulder shankle tongue. Kielbasa rump salami
shankle. Meatball meatloaf ham, shankle rump.</p>
<p>Zweiter Absatz: Beef ribs turducken pastrami ham hock, pork chop shank turkey
sirloin pancetta shankle ham cow rump strip steak fatback.</p>

<blockquote>Ich bin ein Zitat und hier steht bestimmt irgendetwas ganz
schlaues.</blockquote>

<h2>Die zweite Überschrift</h2>

<p>Hier folgen zwei Listen. Zuerst eine Aufzählung und dann eine nummerierte
Liste:</p>
<ul>
  <li>Butter</li>

  <li>Mehl</li>
  <li>Eier</li>
</ul>
<ol>
  <li>Gold</li>

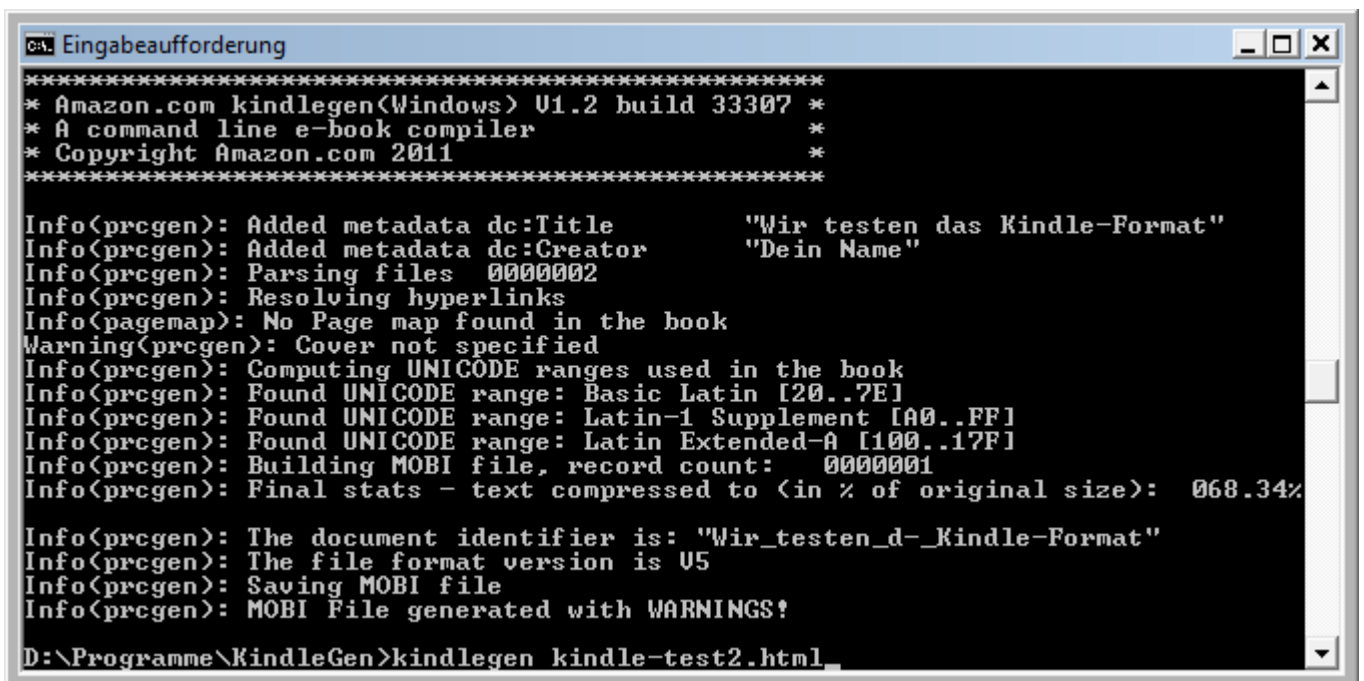
  <li>Silber</li>
```

```
<li>Bronze</li>
</ol>
<p>Hier ein absatz mit ein paar unmaskierten Umlauten und Sonderzeichen: Ää Öö Üü ß
und jetzt ein paar diakritische Zeichen Ćć Čč Šš.</p>
</body>

</html>
```

Speichere diese Datei ab. Rufe nun die Kommandozeile auf. In Windows einfach auf die Windows-Taste drücken und in den Suchschlitz `cmd` eingeben und `Enter` drücken.

Gehe in den Ordner wo sich deine `kindlegen.exe` befindet und rufe dort dann die HTML-Datei auf. Damit ich nicht lange Pfadangaben eintippen muss habe ich einfach die HTML-Datei innerhalb des KindleGen-Ordners abgelegt. Daher muss ich nur folgenden Aufruf starten: `kindlegen name-der-html-datei.html`. Siehe auch die letzte Zeile im folgenden Screenshot:



```
C:\> Eingabeaufforderung
*****
* Amazon.com kindlegen(Windows) V1.2 build 33307 *
* A command line e-book compiler *
* Copyright Amazon.com 2011 *
*****

Info(prcgen): Added metadata dc:Title "Wir testen das Kindle-Format"
Info(prcgen): Added metadata dc:Creator "Dein Name"
Info(prcgen): Parsing files 0000002
Info(prcgen): Resolving hyperlinks
Info(pagemap): No Page map found in the book
Warning(prcgen): Cover not specified
Info(prcgen): Computing UNICODE ranges used in the book
Info(prcgen): Found UNICODE range: Basic Latin [20..7E]
Info(prcgen): Found UNICODE range: Latin-1 Supplement [A0..FF]
Info(prcgen): Found UNICODE range: Latin Extended-A [100..17F]
Info(prcgen): Building MOBI file, record count: 0000001
Info(prcgen): Final stats - text compressed to <in % of original size>: 068.34%

Info(prcgen): The document identifier is: "Wir_testen_d-_Kindle-Format"
Info(prcgen): The file format version is V5
Info(prcgen): Saving MOBI file
Info(prcgen): MOBI File generated with WARNINGS!

D:\Programme\KindleGen>kindlegen kindle-test2.html
```

### KindleGen im Einsatz

KindleGen liefert Informationen und gibt Warnungen aus wenn etwas fehlen sollte. Im Moment ist das lediglich das Buch-Cover, aber diesem Thema werden wir uns im letzten Teil der Artikelserie widmen. Nun rufst du mit dem KindlePreviewer die neu generierte mobi-Datei auf:



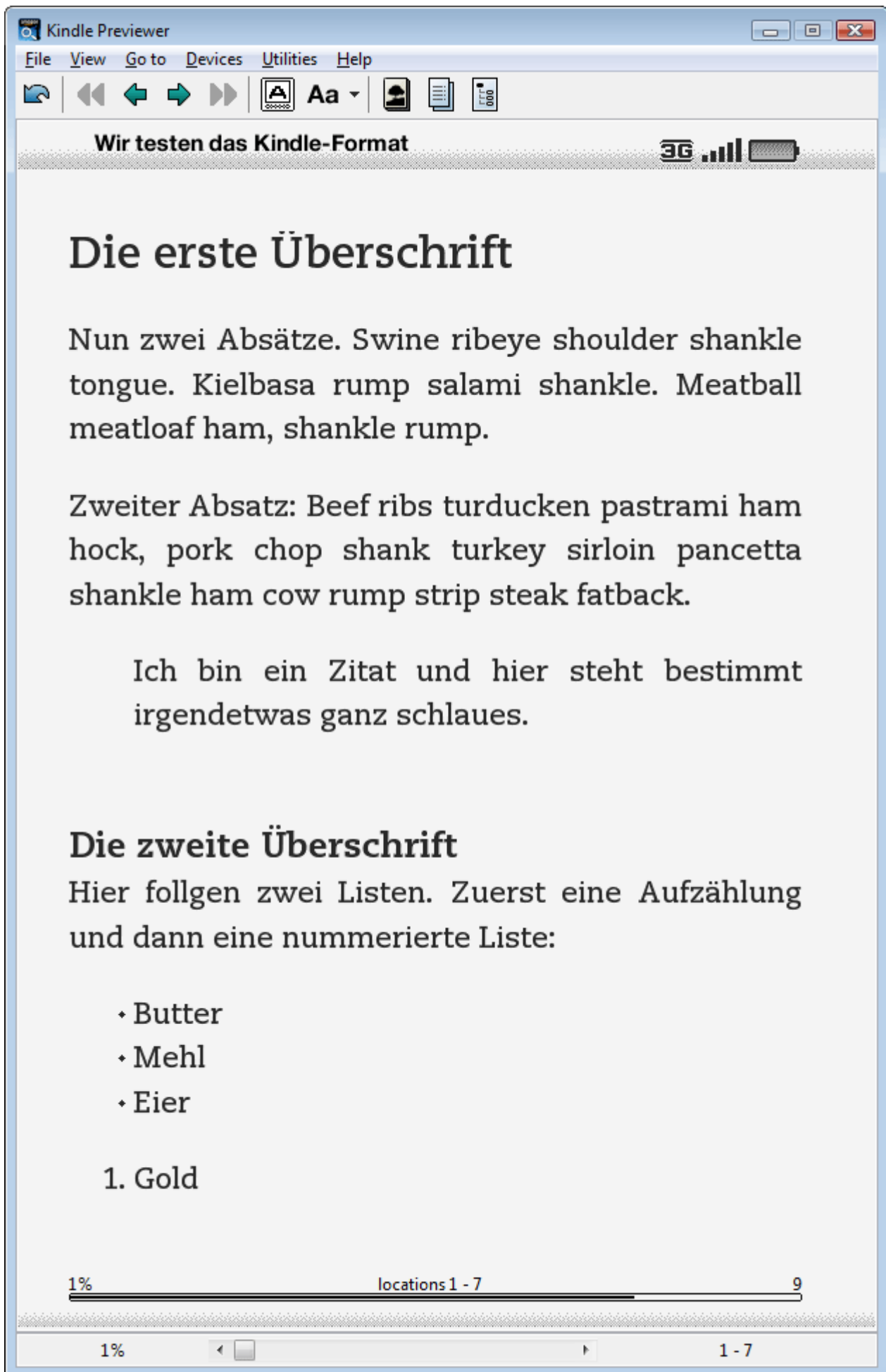


Herzlichen Glückwunsch, du hast gerade dein erstes Dokument für [Amazon Kindle](#) erstellt. Sieht so weit gut aus, aber mir haben auf Anhieb ein paar Sachen nicht gefallen. Ich mag die eingerückten Absätze

nicht und zudem finde ich einige Abstände zu eng, daher habe ich mir folgende CSS-Regeln für Kindle-Dokumente erstellt:

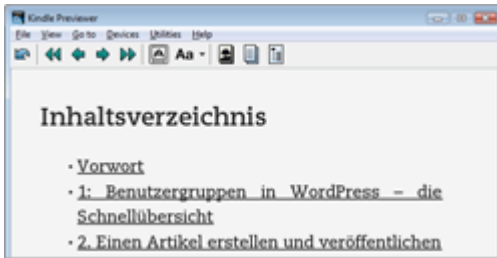
```
h1 {font-size: 1.5em; margin-bottom: 1em;}
h2 {font-size: 1.375em; margin: 1.25em 0 0.25em 0;}
h3 {font-size: 1.25em; margin: 1.25em 0 0.25em 0;}
h4 {font-size: 1.125em; margin: 1.25em 0 0.25em 0;}
h5 {font-size: 1em; font-weight: bold; margin: 1.25em 0 0.25em 0;}
h6 {font-size: 1.375em; font-weight: bold; margin: 1.25em 0 0.25em 0;}
table {margin-top: 0.75em;}
p {margin: .125em 0 .5em 0; text-indent: 0;}
ul, ol, blockquote, pre, dl, hr {margin-top: .125em; margin-bottom: .5em;}
.inhaltsverzeichnis ul {list-style: none; padding-left: 0; margin-left: 20px;}
```

Wenn man diese CSS-Datei verlinkt und dann die mobi-Datei neu generiert, dann schaut das Dokument so aus:



Ich persönlich finde das so übersichtlicher und leserlicherer. Aber das ist natürlich Geschmackssache.

# 3. Das Inhaltsverzeichnis, Seitenumbrüche und Umgang mit Grafiken



In diesem Kapitel wirst du ein **Inhaltsverzeichnis** erstellen, **Seitenumbrüche** einfügen und ich werde noch ein paar Sachen zu den Bildern und **Grafiken** erzählen.

## 3.1 Inhaltsverzeichnis erstellen

Im Gegensatz zu einem Roman ist in einem Fachbuch das Inhaltsverzeichnis ein wichtiger Bestandteil. Fachbücher kann man von vorne bis hinten durchlesen, aber sie dienen auch als Nachschlagewerk oder werden nur abschnittsweise gelesen und hierbei ist ein Inhaltsverzeichnis sehr wichtig. Zudem ist das Inhaltsverzeichnis auch eine wichtige Kaufentscheidung, da der Käufer vorab erkennen kann ob die Themen, die ihn interessieren, auch besprochen werden.

Das Inhaltsverzeichnis erstellst du dadurch, dass du zuerst die relevanten Überschriften im Dokument mit dem Universalattribut `id` ausstattest:

```
<h2 id="artikel-erstellen">Artikel erstellen</h2>
```

```
<p>...</p>
```

```
<h3 id="artikel-titel">Der Artikeltitel</h3>
```

```
<p>...</p>
```

```
<h3 id="artikel-inhalt">Der Artikelinhalt</h3>
```

```
<p>...</p>
```

Mit dem `id`-Attribut schaffst du u.a. einen Ansatzpunkt für CSS und Javascript, aber das interessiert uns hier mal nicht. Mit dem `id`-Attribut schaffst du auch einen Anker auf den du verweisen kannst. Daher ist es wichtig, dass der Inhalt des `id`-Attributs bzw. der Ankername eindeutig ist und pro Dokument **nur einmal** vorkommen darf.

Den eigentlichen Inhaltsverzeichnis erstellst du in dem du am Anfang des Dokuments die Links zu den einzelnen Ankern auflistest:

```
<ul>
```

```
  <li><a href="#artikel-erstellen">Artikel erstellen</a>
```

```
  <ul>
```

```
    <li><a href="#artikel-titel">Der Artikeltitel</a></li>
```

```
    <li><a href="#artikel-inhalt">Der Artikelinhalt</a></li>
```

```
        </ul>
    </li>
</ul>
```

Bei manchen Editoren musst du diese Arbeit manuell machen, aber es gibt Editoren, wie z. B. den [Kompozer](#), der kann so etwas automatisch. Übrigens, den KompoZer bekommt man einzeln oder im Paket mit dem Browser-Paket [SeaMonkey](#).

Wie du jetzt die Benennung der Ankernamen realisierst ist zum Teil Geschmackssache und was du dir leichter merken kannst: "sprechende" Namen oder z. B. Nummernkombinationen. Wichtig ist das der Name einmalig ist. Du darfst aber auf den gleichen Anker mehrfach verlinken, z. B. aus dem Inhalt heraus:

```
<p>Erklärt habe ich das ganze am Anfang des <a href="#kapitel2">zweiten
Kapitels</a>.</p>
```

## 3.2 Manuelle Seitenumbrüche

Momentan arbeitest du bei deinem Kindle-Dokument **noch** in einer einzelner Datei. Aber es könnte dennoch sein, dass du die einzelnen Abschnitte des Dokumentes stärker trennen möchtest. HTML kennt keine Seitenumbrüche, weil HTML keine [Seitenbeschreibungs-](#) sondern eine [Auszeichnungssprache](#) ist.

Aber das ist kein Problem. Ich habe schon erwähnt gehabt, dass das Kindle-Format (.azw) auf [Mobipocket](#) basiert. Daher können wir auch auf die [Mobipocket-Tags](#) zurückgreifen und darunter befindet sich auch der [Seitenumbruch](#): `<mbp:pagebreak/>`.

Den Seitenumbruch fügt man ein, wie einen normalen HTML-Tag ein und zwar dort, wo später die Seite umbrechen sollte. Hier ein Beispiel:

```
<ul>
  <li><a href="#artikel-erstellen">Artikel erstellen</a>
    <ul>
      <li><a href="#artikel-titel">Der Artikeltitel</a></li>
      <li><a href="#artikel-inhalt">Der Artikelinhalt</a></li>
    </ul>
  </li>
</ul>

<mbp:pagebreak/>

<h2 id="artikel-erstellen">Artikel erstellen</h2>
<p>...</p>

<h3 id="artikel-titel">Der Artikeltitel</h3>
```

<p>...</p>

<h3 id="artikel-inhalt">Der Artikelinhalt</h3>

<p>...</p>

Dadurch, dass man bei [Amazon Kindle](#) sowohl die Schriftgröße, als auch die Zeilenabstände ändern und der Inhalt sowohl hochkant als auch quer dargestellt werden kann, sollte man mit den manuell eingefügten Seitenumbrüchen **sparsam** umgehen.

### 3.3 Ein paar Infos zu den Bildern

Die Auflösung des Kindle-Readers liegt bei **600×800 Pixel** und daher sollte man schauen, dass die Bilder so abgemessen sind, dass die längere Seite nicht 800 und die kürzere Seite nicht 600 Pixel überschreitet. Sicher, größere Bilder kann Kindle auf die passende Größe skalieren, aber man darf nicht vergessen, dass wenn man das E-Book auf Amazon zum Verkauf anbietet, der einzelne Download den Autor 12 Eurocent pro angefangenem MByte kostet.



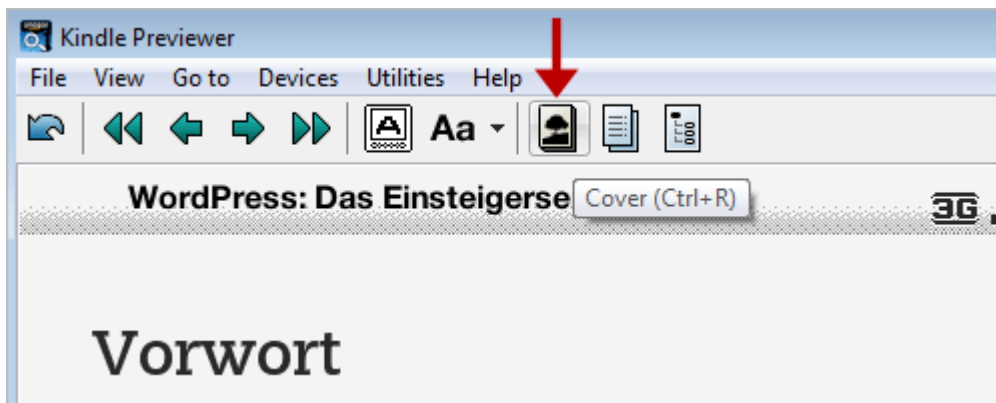
Der Kindle selber kann im Gegensatz zu den Kindle-Apps nur **16 Grautöne** darstellen. Allerdings ist das für die meisten Grafiken ausreichend genug um alle Details darstellen zu können. Siehe zum Vergleich die Abbildung. Es ist aber auch klar, dass nicht jede Feinheit auf einem Porträtfoto bei 16 Graustufen zu sehen ist. 😊

Da **Kindle-Apps** aber sehr wohl **Farbe** darstellen können ist es nicht sinnvoll, die eingebundenen Abbildungen extra in Graustufen umzuwandeln. Das Kindle-Format unterstützt auch BMP-Grafiken (.bmp). Ich werde dir dennoch raten, dass du dich auf .jpg, .gif und .png beschränkst. Das JPG eignet sich für Bilder bzw. Grafiken wo auch Bilder vorkommen/integriert sind. GIF und PNG eignen sich für die restlichen Formate. Das ist zwar eine grobe, aber dennoch ausreichende Erklärung.

PNG macht in den meisten Fällen eine bessere Figur. GIF ist vor allem dann zu empfehlen, wenn du mit Programmen arbeitest, die noch die Gamma-Information abspeichern (z. B. bis einschl. Photoshop CS2) und du keine Möglichkeit/Lust/Zeit hast diese zu korrigieren.

## 4. Das Buchcover und Sektionen

Nun kommen wir zu dem vierten Kapitel dieser Anleitung. Im ersten Kapitel hast du dich um die Entwicklungsumgebung gekümmert. Im zweiten Kapitel hast du auf Basis einer einfachen HTML-Datei dein erstes Kindle-Dokument erstellt und im dritten Kapitel habe ich gezeigt, wie man innerhalb einer HTML-Datei die Seitenumbrüche einfügt, Inhaltsverzeichnis erstellt und ich habe auch noch ein bisschen zu den Bildern und Grafiken geschrieben.



*Amazon Kindle: Buchcover und Inhaltsverzeichnis erkennen*

In diesem Kapitel werde ich zeigen, wie man mit mehreren HTML-Dateien arbeitet, wie man das Buchcover und Inhaltsverzeichnis so einfügt, dass es von Kindle auch als solche zu erkennen sind (siehe erste Abbildung) und wie man Sektionen im E-Book realisiert.

### 4.1 Mit mehreren Dateien arbeiten und Bildcover einfügen

Um mit mehreren HTML-Dateien zu arbeiten und um Bildcover richtig einzufügen benötigen wir eine [OPF-Datei](#) (.opf). Es handelt sich hierbei um eine Datei im *xml*-Format. In der deutschsprachigen Wikipedia gibt es ebenfalls einen [kleinen Hinweis](#) zu diesem Format.

Mit der OPF-Datei legst du die Struktur des E-Books fest. Bevor ich jetzt groß mit den theoretischen Grundlagen aushole, füge ich einfach die OPF-Datei von unserem [WordPress-E-Book](#) ein:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<package version="2.0" xmlns="http://www.idpf.org/2007/opf" unique-identifier="uid">
  <metadata xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:opf="http://www.idpf.org/2007/opf">
    <dc:title>WordPress f&uuml;r Autoren und Redakteure</dc:title>
    <dc:language>de</dc:language>

    <dc:identifier id="uid">WordPress f&uuml;r Autoren und
Redakteure</dc:identifier>
    <dc:creator opf:role="aut">Vladimir Simovic und Thordis Bonfranchi-
Simovic</dc:creator>
```

```

    <meta name="cover" content="cover-bild" />
</metadata>

<manifest>
    <item id="cover-bild" href="wp-buch-cover.gif" media-type="image/gif" />
    <item id="deckblatt" href="deckblatt.html" media-type="text/html" />
    <item id="inhaltsverzeichnis" href="inhaltsverzeichnis.html" media-
type="text/html" />
    <item id="toc" href="inhaltsverzeichnis.ncx" media-type="application/x-
dtbncx+xml"/>
    <item id="vorwort" href="vorwort.html" media-type="text/html" />

    <item id="kapitel-1" href="kapitel-1.html" media-type="text/html" />
    <item id="kapitel-2" href="kapitel-2.html" media-type="text/html" />
    <item id="kapitel-3" href="kapitel-3.html" media-type="text/html" />
    <item id="kapitel-4" href="kapitel-4.html" media-type="text/html" />
</manifest>
<spine toc="toc">

    <itemref idref="cover-bild" />
    <itemref idref="deckblatt" />
    <itemref idref="inhaltsverzeichnis" />
    <itemref idref="vorwort" />
    <itemref idref="kapitel-1" />
    <itemref idref="kapitel-2" />

    <itemref idref="kapitel-3" />
    <itemref idref="kapitel-4" />
</spine>
<guide>
    <reference type="toc" title="inhaltsverzeichnis"
href="inhaltsverzeichnis.html"/>
</guide>

</package>

```

In dieser Datei gibt es vier Abschnitte. Im ersten Abschnitt (`<metadata></metadata>`) werden die Metadaten eingegeben, innerhalb von `<manifest></manifest>` werden die Dateien aufgelistet, im dritten Abschnitt (`<spine></spine>`) gibt es ein einfaches Inhaltsverzeichnis und innerhalb von `guide` wird mit dem Attribut `type="toc"` auf ein ausführliches Inhaltsverzeichnis aufgelistet.

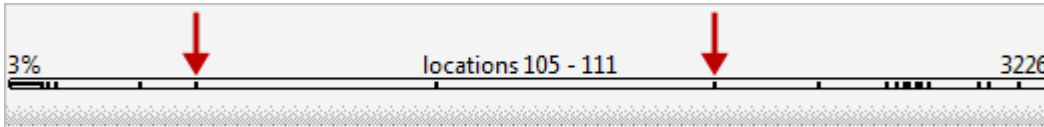
Nur ein Hinweis am Rande. Ich habe bereits erwähnt, dass man bei der Erstellung von Kindle-Dokumenten auf `utf-8` zurückgreifen kann. Für die Erstellung des [WordPress-E-Books](#) habe ich testweise auf `iso-8859-1` (westeuropäischer Zeichensatz) zurückgegriffen.



Wenn du jetzt ein Kindle- bzw. Mobipocket-Dokument generieren möchtest, dann wirst du mit dem KindleGen nicht mehr eine HTML-Datei **sondern** die OPF-Datei aufrufen.

## 4.2 Sektionen einfügen

Gut, mit dem oberen Code erkennt Kindle, dass dein E-Book ein Buchcover hat und ein richtiges Inhaltsverzeichnis besitzt und wo es liegt. Was wir aber immer noch nicht haben sind die Sektionen.



### Amazon Kindle: Sektionen im E-Book

Bei [Amazon Kindle](#) kann man zwischen den einzelnen Seiten vor- und zurückblättern, aber man kann mit den Pfeiltasten auch von Kapitel zu Kapitel bzw. von Sektion zu Sektion springen. Um diese Sektionen einzurichten benötigen wir eine zusätzliche *xml*-Datei, eine [ncx-Datei](#):

```
<ncx xmlns="http://www.daisy.org/z3986/2005/ncx/" version="2005-1">
<head>
  <meta name="dtb:title" content="WordPress f&uuml;r Autoren und Redakteure"/>

  <meta name="dtb:depth" content="1"/>
  <meta name="dtb:totalPageCount" content="0"/>
  <meta name="dtb:maxPageNumber" content="0"/>
</head>
<docTitle><text>WordPress f&uuml;r Autoren und Redakteure</text></docTitle>

<navMap>
  <navPoint id="deckblatt" playOrder="1">
    <navLabel><text>Deckblatt</text></navLabel>
    <content src="deckblatt.html" />
  </navPoint>

  <navPoint id="vorwort" playOrder="2">
    <navLabel><text>Vorwort</text></navLabel>
    <content src="vorwort.html" />
  </navPoint>
  <navPoint id="kapitel-1" playOrder="3">

    <navLabel><text>1: Benutzergruppen in WordPress - die
Schnellübersicht</text></navLabel>
    <content src="kapitel-1.html" />
  </navPoint>
  <navPoint id="kapitel-2" playOrder="4">
    <navLabel><text>2. Einen Artikel erstellen und
```

```
veröffentlichen</text></navLabel>
```

```
    <content src="kapitel-2.html" />
```

```
</navPoint>
```

```
<navPoint id="kapitel-3" playOrder="5">
```

```
    <navLabel><text>3. Einen Artikel bearbeiten</text></navLabel>
```

```
    <content src="kapitel-3.html" />
```

```
</navPoint>
```

```
<navPoint id="kapitel-4" playOrder="6">
```

```
    <navLabel><text>4. Seiten anpassen</text></navLabel>
```

```
    <content src="kapitel-4.html" />
```

```
</navPoint>
```

```
<navPoint id="cover-bild" playOrder="7">
```

```
    <navLabel><text>Cover</text></navLabel>
```

```
    <content src="wp-buch-cover.gif" />
```

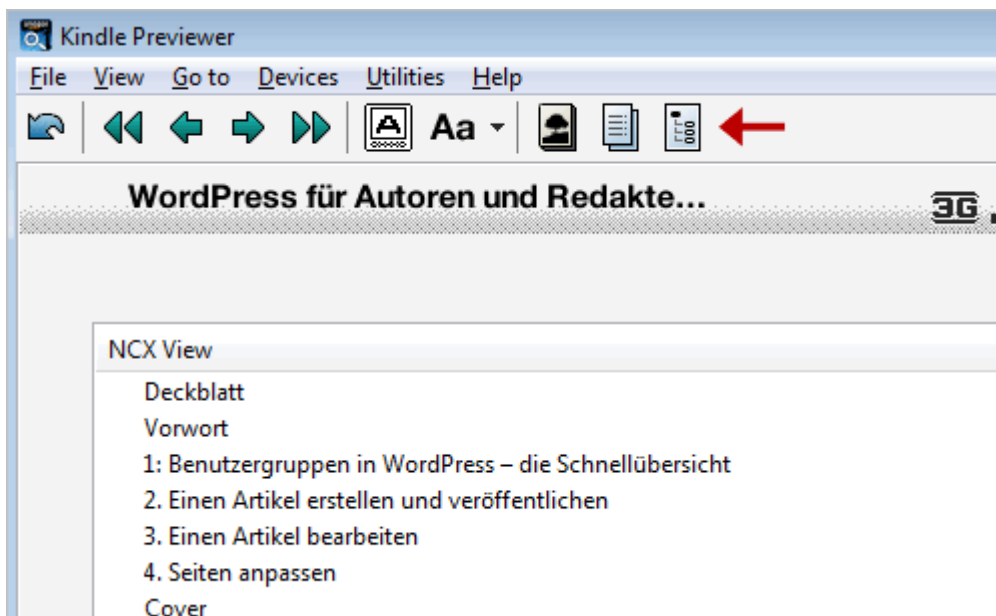
```
</navPoint>
```

```
</navMap>
```

```
</ncx>
```

Ich habe wieder einfach die Datei aus unserem E-Book hier eingefügt. Innerhalb von `<head></head>` werden Meta-Daten untergebracht. Innerhalb von `<navMap></navMap>` befindet sich das logische Inhaltsverzeichnis und die einzelnen `navPoint` stellen die einzelnen Abschnitte dar. Der Wert des `id`-Attributs muss sich mit den Werten aus der `.opf`-Datei decken.

Weiterhin bleibt die `.opf`-Datei diejenige, die man mit dem KindleGen anspricht. Anschließend kann man zwischen den einzelnen Sektionen springen.



*Amazon Kindle: einzelne Sektionen im Previewer*

Im KindlePreviewer kann man das Ergebnis der *.ncx*-Datei sehen.

Die beiden *xml*-Dateien sehen auf den ersten Blick komplizierter aus als sie es sind. Hast du dann die Dateien richtig eingestellt, dann kannst du auch für weitere Projekte nutzen, da sie sich nur an wenigen Stellen ändern.

Falls du auch die [ISBN](#) in dein E-Book einbinden, dann einfach den `metadata`-Bereich in der *.opf*-Datei um folgenden Eintrag erweitern:

```
<dc:identifizier id="BookId" opf:scheme="ISBN">...</dc:identifizier>
```

Das war es. Nun bist du in der Lage E-Books für den [Amazon Kindle](#) zu erstellen, mit allen was dazu gehört. Viel Erfolg damit.

## 5. Paar Hinweise zum Schluss

Ich hoffe die Dokumentation hat dir geholfen. Nun folgen ein paar abschließende Hinweise.

### 5.1 (Urheber-) Rechtliche Hinweise

Du darfst dieses PDF-Dokument lesen, abspeichern, ausdrucken und **kostenlos** an Bekannte und Kunden weiter geben. Was **nicht** erlaubt ist, ist die Änderung des Dokuments, der Verkauf des Dokuments und/oder die Weitergabe des Dokuments wo nicht ersichtlich ist, wer der wirkliche Verfasser ist. Am einfachsten ist es daher, wenn du das **komplette Dokument** unverändert weiterleitest oder bei dir auf dem Server ablegst.

Kontakt Daten findest du auf [www.perun.net/impressum/](http://www.perun.net/impressum/).

### 5.2 To-Do-Liste

Dieses Dokument ist natürlich nicht fertig und ich habe es mir vorgenommen es bei Zeiten zu aktualisieren und auch zu erweitern. Speziell der Teil der sich mit dem Veröffentlichen auf Amazons Plattform [Kindle Direct Publishing](#) beschäftigt, würde ich gerne in diesem Dokument einbauen.

### 5.3 Dankeschön sagen

Falls dir das Dokument geholfen hat und du dich erkenntlich zeigen möchtest, dann kannst du das einfach tun in dem du, wenn du eh dabei bist etwas bei Amazon.de zu kaufen, es [über diesen Partnerlink](#) tust. Dir entstehen dabei **keine** extra Kosten.