

Kapitel 3

Der Aufbau einer EPUB-Datei





Natürlich werden nur die wenigsten von Ihnen EPUB-Dateien von Grund auf selbst erstellen, denn Computer eignen sich weit besser dazu als Menschen. Allerdings sind Rechner nicht dazu in der Lage, diese Aufgabe vollkommen eigenständig zu erfüllen. Bis es eine funktionierende Komplettlösung zum Erstellen von EPUB-Dateien gibt, besteht die beste Vorgehensweise darin, mehrere Werkzeuge einzusetzen: als Erstes gewöhnlich InDesign oder Word, um den vorläufigen Code zu erstellen, und dann einen Texteditor, um letzte Hand an diesen Code zu legen.

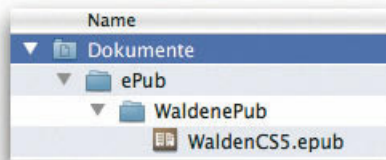
In diesem Kapitel lernen Sie Folgendes:

- Eine von InDesign oder auf anderem Wege erstellte EPUB-Datei entpacken und die Inhalte einsehen
- Die einzelnen Dateien bestimmen und erstellen, die ein EPUB-Dokument ausmachen. Dazu gehören die Dateien mit dem Text und den Bildern für das E-Book, die Datei, auf der das navigierbare Inhaltsverzeichnis beruht, und die Datei, die die Inhalte des E-Books aufführt.
- Eine Bilddatei als Cover für das E-Book festlegen
- Die einzelnen Dateien wieder korrekt zu einem EPUB-Dokument zurückzupacken
- Die Gültigkeit der EPUB-Datei mit EpubCheck überprüfen

EPUB-Dateien entpacken

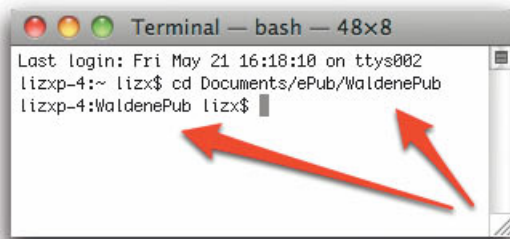
Wenn Sie Ihre EPUB-Datei mit InDesign erstellt haben – oder wenn Sie neugierig sind, wie eine EPUB-Datei aus einer anderen Quelle aufgebaut ist –, können Sie sie entpacken, um die XHTML- und CSS-Dateien einzusehen, die sich darin befinden.

- 1 Erstellen Sie ein neues Verzeichnis für die EPUB-Datei und verschieben Sie diese dorthin. (Wenn Sie die EPUB-Datei entpacken, haben Sie es mit einem Wust von Dateien zu tun, und es wäre äußerst unschön, wenn diese überall auf dem Desktop verstreut lägen.)



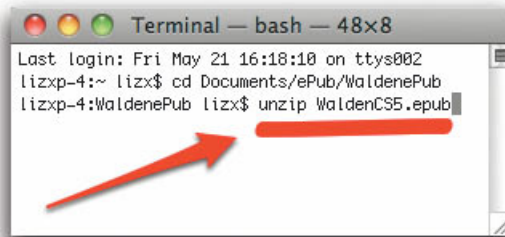
Ich habe die Datei WaldenCS5.epub in das Verzeichnis WaldenePub gelegt, das sich wiederum im Verzeichnis ePub des Dokumentenordners befindet.

- 2 Öffnen Sie auf dem Mac das Programm Terminal. (Unter Windows können Sie Zip verwenden, das Sie unter <http://bit.ly/aUXJLG> finden).
- 3 Wechseln Sie zu dem Verzeichnis mit der EPUB-Datei. In unserem Beispiel muss ich also `cd Documents/ePub/WaldenePub` eingeben.



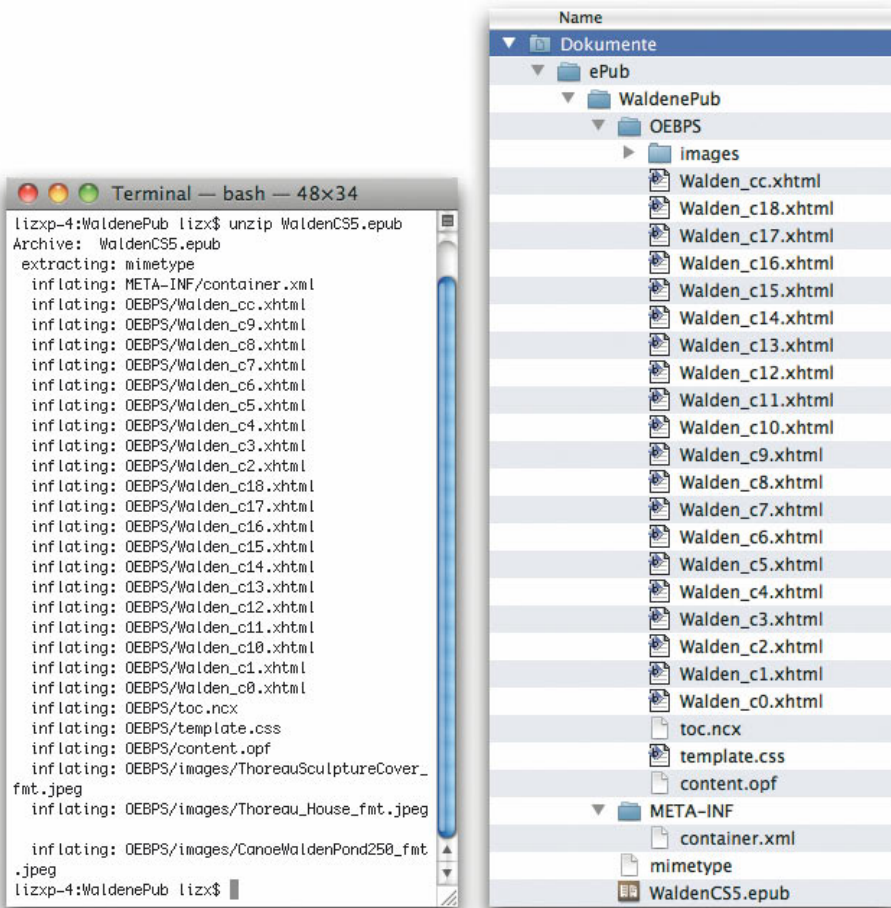
Wenn Sie in das gewünschte Verzeichnis wechseln, erscheint dessen Name hinter dem Doppelpunkt in der Befehlszeile.

- 4 Wenn Sie in dem Verzeichnis mit der EPUB-Datei sind, die Sie entpacken wollen, geben Sie einfach `unzip` gefolgt von Dateiname und Endung ein:
`unzip WaldenCS5.epub`



Da Sie vorher das richtige Verzeichnis aufgesucht haben, müssen Sie jetzt nicht den Pfad zur Datei eingeben.

5 Drücken Sie . Das Verzeichnis füllt sich mit den XHTML-, CSS- und sonstigen XML-Dateien sowie den Bildern und Schriftarten, die in der EPUB-Datei enthalten waren.



Sobald Sie nach der Eingabe des Befehls `unzip` die Taste drücken, erscheint vor Ihren Augen der Inhalt der EPUB-Datei im Terminalfenster (links) und im Finder (rechts).

Tipp

Gewöhnlich stelle ich die Ordner OEBPS und META-INF sowie die Datei mimetype in einen eigenen Ordner. Dann kann ich alle Änderungen in diesem Ordner vornehmen, eine neue EPUB-Datei herstellen und testen, ohne das ursprünglich aus InDesign exportierte EPUB-Dokument zu verändern.

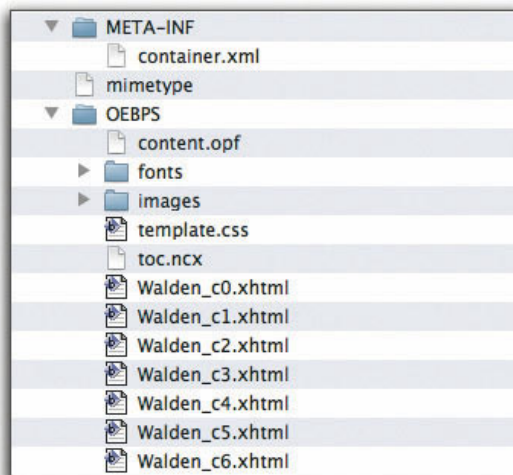
Mit dem Programm Springy können Sie sich den Inhalt einer EPUB-Datei ansehen und die darin enthaltenen XHTML- und CSS-Dateien bearbeiten, ohne das ursprüngliche Dokument zu entpacken. Dieses Programm finden Sie unter <http://springyarchiver.com>.

Sie können eine EPUB-Datei auch dadurch entpacken, dass Sie die Endung in .zip ändern und auf die Datei doppelklicken. Um sie wieder zu packen, brauchen Sie jedoch immer noch Terminal oder ein ähnliches Programm.

Die Dateien in einer EPUB-Datei

Wie Sie im vorherigen Abschnitt gesehen haben, können Sie die von InDesign erstellte EPUB-Datei entpacken, um die darin enthaltenen Dateien einzusehen und zu bearbeiten. Falls Sie Word verwenden oder den Code für das E-Book selbst schreiben, müssen Sie die Bestandteile des EPUB-Dokuments manuell zusammenstellen. In jedem Fall aber müssen Sie wissen, wo sich die einzelnen Dateien befinden sollten, sodass Sie von den anderen Dateien aus richtig auf sie verweisen können.

In unserem Beispiel erstellen wir ein E-Book für Henry David Thoreaus *Walden*. Die endgültige Struktur sieht wie folgt aus:



Die Dateien, aus denen sich ein typisches E-Book zusammensetzt. Sie können diese EPUB-Datei von der Website zu diesem Buch herunterladen.

Um die Dateien zu bearbeiten, die sich in einem EPUB-Dokument befinden, benötigen Sie einen Texteditor, vorzugsweise mit GREP-Fähigkeiten. Mithilfe von GREP können Sie reguläre Ausdrücke verwenden und damit im Turbogang suchen und ersetzen, was die Bearbeitung der EPUB-Dateien stark beschleunigt. Sie können zwar den mit dem Computer mitgelieferten Texteditor verwenden, also TextEdit auf dem Mac oder den Editor unter Windows, aber es gibt viele andere, die besser für die Arbeit mit XHTML-Dateien geeignet sind, z.B. BBEdit (<http://www.barebones.com>) oder TextMate (<http://macromates.com>) für den Mac oder Notepad++ für Windows (<http://notepad-plus-plus.org>). Verwenden Sie nicht Microsoft Word oder eine andere Textverarbeitung, denn dadurch würden Sie den HTML-Code beschädigen.

Die Datei mimetype

Bei *mimetype* handelt es sich um eine einfache Textdatei, die nur eine einzige Zeile enthält, nämlich `application/epub+zip`, ohne Wagenrücklauf- oder Zeilenumbruchzeichen. Diese Datei ist bei jedem E-Book identisch, weshalb Sie sie einfach kopieren und in die Ordner sämtlicher E-Books einfügen können. Wenn Sie E-Books von Hand erstellen wollen, können Sie diese Datei aus einer beliebigen EPUB-Datei entnehmen, z.B. aus derjenigen im Ordner *Tutorial* auf der Website zu diesem Buch.

Der Ordner META-INF

Der Ordner *META-INF* enthält nur eine einzige Datei, nämlich *container.xml*, die wie folgt aussieht:

```
<?xml version="1.0"?>
<container version="1.0"
  xmlns="urn:oasis:names:tc:opendocument:xmlns:container">
  <rootfiles>
    <rootfile full-path="OEBPS/content.opf"
      media-type="application/oebps-package+xml"/>
  </rootfiles>
</container>
```

Der einzige Wert, den Sie unter Umständen anpassen müssen, ist das Attribut *full-path* im Element *rootfile*, das auf die Datei *content.opf* des EPUB-Dokuments zeigen muss. Wenn Sie die Datei *content.opf* stets im Ordner *OEBPS* unterbringen, wie es üblich ist und wie wir es auch hier tun, können Sie diese Datei unverändert für sämtliche EPUB-Dokumente verwenden.

Wenn Sie Schriftarten einbetten, stellt InDesign auch noch die Datei *encryption.xml* in den Ordner *META-INF*. Diese Datei wird auch verwendet, wenn Sie eine digitale Rechteverwaltung einrichten. Weitere Informationen über die Verschlüsselung finden Sie in der OPS-Spezifikation unter <http://www.idpf.org/ocf/ocf1.0/download/ocf10.htm#Ref8795282>. In diesem Buch kümmern wir uns jedoch nicht um die digitale Rechteverwaltung für E-Books.

Der Ordner OEBPS

Im Ordner *OEBPS* finden Sie die Inhalte des E-Books. Das Kürzel *OEBPS* steht für *Open eBook Publication Structure* und ist nur eines von vielen sperrigen Akronymen, mit denen Sie leben müssen. Sie könnten diesen Ordner zwar auch einfach *Buch* nennen (sofern Sie den Pfad in der Datei *container.xml* entsprechend anpassen), aber standardmäßig heißt er *OEBPS*.

In jedem Fall enthält der Ordner *OEBPS* alle XHTML-Dateien mit dem Text des Buches, eine CSS-Datei mit den Formatierungsanweisungen, je einen Ordner für Bilder und Schriftarten und zwei besondere XML-Dateien: *toc.ncx* wird von E-Readern benutzt, um das navigierbare Inhaltsverzeichnis zu erstellen, während *content.opf* Metadaten über das Buch sowie eine Liste der einzelnen Dateien enthält, aus denen sich der Inhalt zusammensetzt.

Die XHTML- und CSS-Dateien

Die Dateien im Ordner *OEBPS* können Sie in beliebiger Reihenfolge bearbeiten, aber am geschicktesten ist es, mit den Inhalten anzufangen, die in den Formaten XHTML 1.1 und CSS vorliegen. XHTML ist eine strengere Variante von HTML – der Sprache, in der auch Webseiten geschrieben werden –, CSS die Sprache, mit der die Inhalte eines XHTML-Dokuments formatiert werden.

Ein XHTML-Dokument in einer EPUB-Datei muss mit einer XML- und einer Dokumenttypdeklaration beginnen. Der folgende Code muss am Anfang aller Ihrer XHTML-Dokumente stehen. Außerdem müssen Sie diese Dokumente mit UTF-8-Kodierung speichern.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.
  w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

Auf die ersten Informationen über das XHTML-Dokument selbst folgt der `<head>`-Abschnitt mit Metadaten über die Inhalte wie Titel und Speicherort des Stylesheets:

```
<head>
  <title>Walden von Henry David Thoreau, Erstveröffentlichung 1854
</title>
  <link href="Walden.css" rel="stylesheet" type="text/css" />
</head>
```

Beachten Sie, dass der Titel, der in einem E-Reader oder online angezeigt wird, im Dokument *content.opf* steht und *nicht* durch den Titel irgendeines der XHTML-Dokumente bestimmt wird. Das Element `link` teilt dem E-Reader mit, wo er die Datei mit dem CSS-Stylesheet finden kann, das er zur Formatierung des Dokuments heranziehen soll (*Walden.css*).

Auf den Kopfabschnitt folgt der Rumpf (`<body>`). Das folgende Beispiel zeigt, wie der Rumpf eines durchschnittlichen XHTML-Dokuments – hier das erste Kapitel von *Walden* – in einem EPUB-Dokument aussieht:

```
<body>
<div id="walden-c1">
  <h1 class="header">Genügsamkeit</h1>
  <p class="first-para"><span class="small-caps">Als ich die
    folgenden Seiten</span> schrieb, oder besser gesagt, den größten
    Teil davon, ...</p>
  <div class="illustration">
  <p class="caption">Thoreaus Haus am Walden Pond</p>
```

```

</div>
<p class="body">Ich würde meinen Lesern meine persönlichen ... </p>
...
<p class="quote">Inde genus durum sumus, experiensque </p>...
</div>
</body>
</html>

```

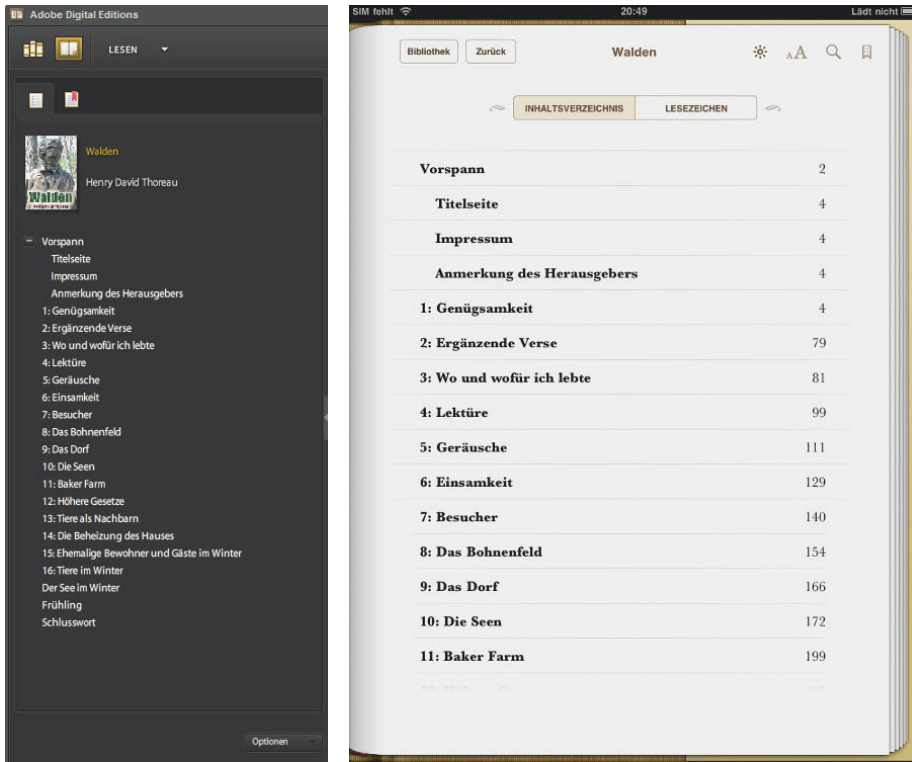
Beachten Sie bei diesem Dokument Folgendes:

- Es gibt vier Klassen für p-Elemente (*first-para*, *caption*, *body* und *quote*), sodass diese vier Arten von Absätzen unterschiedlich formatiert werden können.
- Die Datei heißt *walden_c1.xhtml* und befindet sich auf der obersten Ebene des Ordner *OEBPS*.
- Die (stark gekürzte) Version von *walden_c1.xhtml* in dem zuvor gezeigten Code bildet nur das erste Kapitel des Buchs. Wenn ein Buch in einzelne Dateien für die verschiedenen Kapitel gegliedert ist, zeigen die meisten E-Reader den Anfang eines jeden Kapitels jeweils auf einer neuen Seite an. Das ist zurzeit die sicherste Methode, um einen Seitenumbruch einzufügen.
- Sie müssen auch noch eigene XHTML-Dokumente für das Cover (siehe den Abschnitt *Das Cover gestalten* weiter hinten in diesem Kapitel), für den Vorspann (Titelblatt, Inhaltsverzeichnis, Vorwort usw.) und Nachspann (z.B. Index und Klappentext) hinzufügen.
- Die Techniken zur Formatierung von E-Books erkläre ich in Kapitel 4, *Erweiterte Formatierung*.

Die Datei *toc.ncx* für das Inhaltsverzeichnis

Es gibt zwei Dateien, die E-Readern den Inhalt des Buches angeben. Eine davon ist *toc.ncx*, eine einfache, UTF-8-kodierte Textdatei, die sich im Ordner *OEBPS* befinden muss. Sie können dieser Datei aber jeden beliebigen Namen geben, solange sie nur die Endung *.ncx* trägt. Das ist eine weitere Abkürzung, die für *Navigation Control file for XML* steht, also für »XML-Navigationssteuerungsdatei«. E-Reader verwenden die Datei *toc.ncx*, um wie in der folgenden Abbildung ein navigierbares Inhaltsverzeichnis (oder Menü) des Buchs zu erstellen. Ich spreche hier von einem »navigierbaren Inhaltsverzeichnis«, um es von dem Inhaltsverzeichnis zu unterscheiden, das sich zusätzlich innerhalb

des eigentlichen E-Books befinden kann. Da es dieses navigierbare Menü gibt, ist ein solches Inhaltsverzeichnis im Buch selbst allerdings weniger notwendig.



Hier sehen Sie das navigierbare Inhaltsverzeichnis des Buchs mit zwei Hierarchieebenen. Auf der linken Seite wird die Darstellung in Adobe Digital Editions gezeigt, auf der rechten Seite die auf dem iPad.

Wenn Sie das EPUB-Dokument von InDesign erstellen lassen, können Sie das Programm anweisen, aus den Überschriften in Ihrem Dokument automatisch ein solches navigierbares Inhaltsverzeichnis, also eine *toc.ncx*-Datei, anzulegen. Weitere Einzelheiten dazu finden Sie im Abschnitt *Ein Inhaltsverzeichnis mit Navigationsmöglichkeiten erstellen* in Kapitel 2. Im Allgemeinen müssen Sie dieses Inhaltsverzeichnis nicht ändern. Wenn Sie es trotzdem tun wollen, z.B. um dem Menü einen zusätzlichen Eintrag hinzuzufügen, gehen Sie nach den folgenden Anweisungen vor. Falls Sie den Inhalt handkodiert oder mit Word erstellt haben, können Sie die Datei *toc.ncx* nach diesen Anweisungen auch manuell erstellen.

Die Datei *toc.ncx* muss ein gültiges XML-Dokument sein und mit der folgenden Deklaration beginnen:

```
<?xml version="1.0"?>
<!DOCTYPE ncx PUBLIC "-//NISO//DTD ncx 2005-1//EN" "http://www.
  daisy.org/z3986/2005/ncx-2005-1.dtd">
<ncx xmlns="http://www.daisy.org/z3986/2005/ncx/" version="2005-1">
```

Darauf folgt der `<head>`-Abschnitt, der die vier meta-Elemente `dtb:uid`, `dtb:depth`, `dtb:totalPageCount` und `dtb:maxPageNumbers` enthalten muss. Allerdings sind für E-Books nur die ersten beiden von Bedeutung.

Der Wert von `dtb:uid`, also die Benutzer-ID, sollte ein eindeutiger String oder Code für das E-Book sein. Verlage können dafür die ISBN verwenden. Wenn dem Buch noch keine solche eindeutige Nummer zugewiesen ist, können Sie eine eigene universell eindeutige Kennung (Universally Unique Identifier, UUID) erstellen und verwenden. (Suchen Sie einfach im Internet nach *uuid erstellen*.) Es gilt die Konvention, die UUID mit `urn:uuid` einzuleiten. Wenn Sie die EPUB-Datei mit InDesign erstellen, legt das Programm die Benutzer-ID an.

Der Wert von `dtb:depth` gibt an, wie viele Ebenen und Unterebenen von Einträgen in dem Inhaltsverzeichnis angezeigt werden. In unserem Beispiel haben wir zwei Ebenen: Ebene 1 für den Vorspann, den Nachspann und die Kapitelnamen, Ebene 2 für Unterabschnitte einzelner Kapitel.

Die beiden letzten Elemente gelten nur für gedruckte Bücher und können auf 0 gesetzt werden, dürfen aber nicht wegfallen.

```
<head>
  <meta name="dtb:uid" content="urn:uuid:2a0e205b-c754-317f-2fc8-
    24b9c93e83dc"/>
  <meta name="dtb:depth" content="2"/>
  <meta name="dtb:totalPageCount" content="0"/>
  <meta name="dtb:maxPageNumber" content="0"/>
</head>
```

Der nächste Abschnitt von `toc.ncx` ist das Element `docTitle` mit dem darin befindlichen Element `text`. Hier sollte zwar der Titel des E-Books stehen, doch habe ich bisher nicht einen einzigen E-Reader gesehen, der sich um dieses Element kümmert. Nichtsdestoweniger ist es sicherlich keine schlechte Idee, dieses Element aufzunehmen. Es sollte wie folgt aussehen:

```
<docTitle>
  <text>Walden</text>
</docTitle>
```

Der dritte Teil von `toc.ncx` ist das Element `navMap`, in dem Sie das navigierbare Inhaltsverzeichnis definieren, das im E-Reader angezeigt wird. Dieses Element gehen wir Punkt für Punkt durch. Als Erstes steht das öffnende `navMap`-Tag:

```
<navMap>
```

Jeder Eintrag im Inhaltsverzeichnis wird mithilfe eines `navPoint`-Elements erstellt, das zwei Attribute aufweist. Das erste lautet `id` und ist ein eindeutiger Code, der den Eintrag innerhalb des navigierbaren Inhaltsverzeichnisses bezeichnet, während das zweite `playOrder` heißt und angibt, an welcher Stelle der Eintrag in dem Inhaltsverzeichnis steht. Kein `playOrder`-Wert darf mehrfach auftreten. (Beachten Sie auch, dass hiermit nicht die Reihenfolge bestimmt wird, in der ein E-Reader von einem Dokument zum nächsten blättert. Das legen Sie im Abschnitt `section` der Datei `content.opf` fest. Weitere Einzelheiten dazu finden Sie im Abschnitt *Die Reihenfolge der Elemente im Buch festlegen (Abschnitt `spine`)* weiter hinten in diesem Kapitel.)

```
<navPoint id="navpoint-1" playOrder="1">
```

Das Element `navLabel` definiert die Verlinkung der Einträge im Inhaltsverzeichnis. Der Text des Eintrags (also der anklickbare Text des Links) steht innerhalb eines `text`-Elements im `navLabel`-Element. Ein Beispiel sehen Sie im folgenden Code für den ersten Eintrag im Inhaltsverzeichnis, nämlich das Cover:

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

Als Nächstes legen Sie das Ziel des Links fest, also den Teil des Buchs, der angezeigt wird, wenn der Leser den betreffenden Eintrag im navigierbaren Inhaltsverzeichnis auswählt. Dazu erstellen Sie ein `content`-Element, das auf die URL des jeweiligen Kapitels zeigt. In diesem Fall sind fragmentarische Bezeichner zulässig (die auf bestimmte Stellen im Dokument verweisen), aber nicht erforderlich. Da das erste Dokument das Cover ist, verweisen wir auf dessen XHTML-Datei. (Informationen darüber, wie Sie das Cover erstellen, finden Sie im Abschnitt *Das Cover gestalten* weiter hinten in diesem Kapitel.)

```
<content src="Walden_cc.xhtml" />
```

Danach beenden wir den Eintrag für das Cover, indem wir das zugehörige `navPoint`-Element schließen:

```
</navPoint>
```

Der gesamte Eintrag sieht jetzt wie folgt aus:

```
<navPoint id="navpoint" playOrder="1">
  <navLabel>
    <text>Cover</text>
  </navLabel>
  <content src="Walden_cc.xhtml" />
</navPoint>
```

Damit haben Sie einen ersten Eintrag für das navigierbare Inhaltsverzeichnis erstellt. Für den nächsten müssen Sie ein neues `navPoint`-Element für die Titelei anlegen:

```
<navPoint id="navpoint-2" playOrder="2">
  <navLabel><text>Titelei</text></navLabel>
  <content src="Walden_c0.xhtml" />
</navPoint>
```

Da die Titelei aus mehreren Dateien besteht – der Titelseite, der Seite mit den Angaben zum Urheberrecht und einer Seite mit Hinweisen des Verlages –, müssen wir Untereinträge dafür erstellen, sodass der Leser im navigierbaren Inhaltsverzeichnis direkt dorthin springen kann. Dazu erstellen Sie ein inneres `navPoint`-Element, bevor Sie das `navPoint`-Element für die Titelei schließen. Im Folgenden sehen Sie die drei Untereinträge unter *Titelei*:

```
<navPoint id="navpoint" playOrder="3">
  <navLabel><text>Titelseite</text></navLabel>
  <content src="Walden_c0.xhtml#toc-anchor" />
</navPoint>
<navPoint id="navpoint" playOrder="4">
  <navLabel><text>Urheberrechtsangaben</text></navLabel>
  <content src="Walden_c0copy.xhtml" />
</navPoint>
<navPoint id="navpoint" playOrder="5">
  <navLabel><text>Hinweise des Verlags</text></navLabel>
  <content src="Walden_c0note.xhtml" />
</navPoint>
```

Beachten Sie, dass der Eintrag für die Titelseite zu einem Anker in dem Dokument führt, das auch mit dem Link unter *Titelei* verknüpft ist. So etwas ist zulässig, solange die beiden URLs nicht identisch sind, was aber dank der fragmentarischen Bezeichner nicht der Fall ist.

Sie können auch die Untereinträge noch weiter unterteilen, wobei es jedoch nur selten vorkommen dürfte, dass Sie mehr als zwei Ebenen benötigen. Wenn Sie alle Untereinträge festgelegt haben, schließen Sie das äußere `navPoint`-Element (in diesem Fall also das für *Titelei*).

```
</navPoint>
```

Erstellen Sie weitere `navPoint`-Elemente für jeden einzelnen Eintrag des Inhaltsverzeichnisses. Anschließend fügen Sie die schließenden Tags für die Elemente `navMap` und `ncx` hinzu:

```
</navMap>
</ncx>
```

Die vollständige Datei *toc.ncx* für dieses Beispiel (mit englischsprachigen Einträgen) finden Sie auf der Website zu diesem Buch.

Ein Merkmal der Datei *toc.ncx*, das zurzeit noch nicht weiträumig unterstützt wird, aber im Kommen ist, bildet das Element `<pageList>`, mit dem Sie für die einzelnen Abschnitte des E-Books angeben können, welchen Seiten der gedruckten Version sie entsprechen. Auf meiner Website werde ich Sie über dieses Element auf dem Laufenden halten.

Manchmal kann es schwierig sein, Fehler in der Datei *toc.ncx* ausfindig zu machen, da sie so viele öffnende und schließende Tags enthält. Um die Datei zu überprüfen, können Sie den Validator unter http://www.w3schools.com/xml/xml_validator.asp nutzen.

Die Datei *content.opf* schreiben

Die zweite der beiden Dateien, die dem E-Reader den Inhalt des E-Books angeben, ist *content.opf*, ebenfalls eine einfache UTF-8-kodierte Textdatei im Ordner *OEBPS*. Auch dieser Datei können Sie jeden beliebigen Namen geben, solange Sie die Erweiterung beibehalten (die wiederum eine weitere Abkürzung ist, nämlich für *Open Packaging Format*) und in der Datei *container.xml* im Verzeichnis *META-INF* korrekt auf sie verweisen. Die Datei besteht aus vier Hauptabschnitten: *metadata* enthält Informationen über den Inhalt des E-Books, *manifest* führt die Dateien einschließlich der Bild- und Schriftartdateien auf, *spine* gibt die Reihenfolge an, in der die Texte innerhalb des E-Books erscheinen sollen, und *guide* beschreibt die Rolle, die die einzelnen XHTML-Dateien für das Buch spielen. Bei Bedarf können Sie die Datei *content.opf* anhand der Anweisungen in diesem Abschnitt selbst schreiben.

Wenn Sie nach der Beschreibung im Abschnitt *InDesign-Dateien ins EPUB-Format exportieren* von Kapitel 2 vorgehen, erstellt InDesign automatisch die Datei *content.opf*, allerdings nicht ganz perfekt: Es wird kein Datum hinzugefügt, der Verweis auf die Coverdatei bzw. das Coverbild fehlt und auch der Abschnitt *guide* wird nicht angelegt. Diese Versäumnisse können Sie wie im Folgenden beschrieben manuell nachholen.

Die Datei *content.opf* liegt im XML-Format vor und beginnt mit der XML-Deklaration sowie dem Wurzel-Element `package`, in dem alle anderen Elemente enthalten sind. Dieses `package`-Element muss die drei folgenden Attribute besitzen: `xmlns` für die Namespace-Deklaration, `version` (mit der verwendeten

OPF-Version) und `unique-identifier`, eine eindeutige Kennung, die dem Wert von `id` in dem ebenfalls erforderlichen Element `dc:identifier` im Abschnitt `metadata` entsprechen muss (diesen Abschnitt sehen wir uns als Nächstes an).

```
<?xml version="1.0"?>
<package xmlns="http://www.idpf.org/2007/opf"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  unique-identifier="bookid" version="2.0">
```

Informationen über das Buch hinzufügen (Metadaten)

Im Abschnitt `metadata` der Datei `content.opf` geben Sie Informationen über das Buch an, z.B. den Titel, den Autor, die eindeutige Kennung, den Illustrator, den Verlag, das Thema, eine Beschreibung usw. Anhand der Metadaten können potenzielle Leser Ihr Buch finden. Daher ist es sehr wichtig, jedem Buch, das Sie veröffentlichen, ausreichende und genaue Metadaten beizugeben.

Der Metadatenabschnitt beginnt mit dem Element `metadata`, der zwei weitere Namespace-Deklarationen enthält:

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:opf="http://www.idpf.org/2007/opf">
```

Dank dieser Namespace-Deklarationen können Sie die Elemente und Attribute aus der Metadatenspezifikation Dublin Core (<http://dublincore.org/documents/2004/12/20/dces>) und der OPF-Spezifikation (<http://www.idpf.org/2007/opf/opf2.0/download>) verwenden, während das XML-Dokument gültig bleibt. Elemente aus der Dublin-Core-Spezifikation müssen mit `dc:` eingeleitet werden, Attribute aus der OPF-Spezifikation mit `opf:`.

Als Nächstes fügen Sie in den Metadatenabschnitt die Dublin-Core-Elemente für Titel, Kennung und Sprache ein. Wir beginnen mit dem Titel:

```
<dc:title>Walden</dc:title>
```

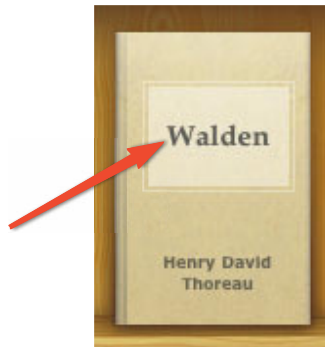
Die Angabe des Titels ist ein Muss, und manche E-Reader öffnen ein E-Book ohne Titelement nicht einmal.

InDesign erstellt diesen Titel nicht von selbst. Entweder fügen Sie ihn anhand der Anweisungen im Abschnitt *Metadaten hinzufügen* von Kapitel 2 innerhalb des Programms hinzu, oder Sie ergänzen ihn nach dem in diesem Abschnitt gezeigten Verfahren manuell.

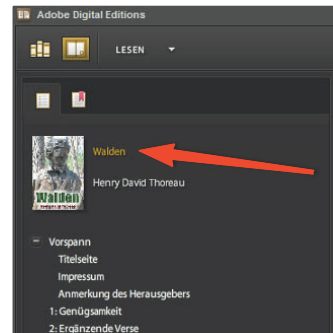
In der Spezifikation heißt es, dass Sie *mindestens* einen Titel angeben müssen (es können also auch mehrere sein), doch schweigt sie sich zurzeit darüber

aus, wie E-Reader mit mehreren Titeln umgehen sollen. Womit die E-Reader aber etwas anfangen können, ist das Attribut `xml:lang`, falls Sie es hinzufügen möchten.

Es gibt viele Stellen, an denen Sie den Titel des E-Books angeben, aber um ein E-Book wie in der folgenden Abbildung anzuzeigen, nutzen die meisten (wenn nicht alle) E-Reader den Inhalt des Elements `dc:title`.



In iBooks auf dem iPad wird der Inhalt des Elements `dc:title` zur Darstellung des Covers verwendet, wenn kein Coverbild zur Verfügung steht.



Der in der Datei `content.opf` angegebene Titel wird auch verwendet, um das Buch im E-Reader zu bezeichnen. iBooks verwendet ihn auf dem Cover, wenn kein Coverbild zur Verfügung steht, und außerdem in der Kopfzeile jeder rechten Seite. In ADE wird der Titel in der linken Navigationsleiste aufgeführt (Abbildung rechts).

Das zweite erforderliche Element ist `dc:identifier`, dessen Wert mit dem identisch sein muss, den Sie im meta-Element `dtb:uid` der Datei `toc.ncx` festgelegt haben. Außerdem muss er der ID im Attribut `unique-identifier` des pack-

age-Elements entsprechen. (Wenn Sie den Export ins EPUB-Format in InDesign vornehmen, kümmert sich das Programm darum.)

Die Art der Kennung können Sie mit dem Attribut `opf:scheme` angeben. In der Spezifikation ist zwar noch nicht festgelegt, welche Attribute zulässig sind, doch gesehen habe ich bereits *ISBN*, *DOI* (Digital Object Identifier), *URN*, *URI* und *URL* (Uniform Resource Name/Identifier/Locator). Tatsächlich können Sie sogar mehrere Kennungen erstellen und das Schema für jede einzelne angeben.

Die Kennungen für unser Beispiel sehen wie folgt aus. Beachten Sie, dass der Wert des Attributs `id` in der zweiten Kennung `bookid` lautet – genauso wie der des Attributs `unique-identifier` im `package`-Element der Datei `content.opf`.

```
<dc:identifier opf:scheme="URI">http://www.elizabethcastro.com/epub/
  Walden</dc:identifier>
<dc:identifier id="bookid">urn:uuid:2a0e205b-c754-317f-2fc8-24b9c93e83dc
  </dc:identifier>
```

Die dritte erforderliche Metadatenangabe ist `dc:date`. Dieses Element muss mindestens eine vierstellige Jahresangabe enthalten, darauf können aber noch optional eine zweistellige Monats- und eine ebenfalls zweistellige Tagesangabe folgen. Das Datum muss den Vorgaben für Datums- und Uhrzeitformate genügen, die das W3C unter <http://www.w3.org/TR/NOTE-datetime> veröffentlicht hat. Häufig wird hier das Datum der Publikation angegeben. Es sind jedoch mehrere `date`-Elemente erlaubt, um neben dem Datum der EPUB-Version auch das der ursprünglichen Veröffentlichung anzugeben. Mit dem Attribut `opf:event` können Sie die verschiedenen `date`-Elemente genauer kennzeichnen.

```
<dc:date opf:event="original-publication">1854</dc:date>
<dc:date opf:event="ops-publication">2010-02</dc:date>
```

Beim Export ins EPUB-Format fügt InDesign das zurzeit nicht hinzu, weder in der Version CS4 noch in CS5. Sie müssen es also wie hier beschrieben selbst ergänzen.

Das vierte erforderliche Element im Metadatenabschnitt ist `dc:language`. Sein Wert muss den Anforderungen von RFC 3066 genügen, und er muss in ISO 639 aufgeführt sein (http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes). Am häufigsten sind dabei Sprachcodes aus zwei Buchstaben. Der für Deutsch lautet `de`. Das Wörterbuch von iBooks funktioniert zurzeit jedoch nur für englische Begriffe, unabhängig vom `dc:language`-Element.

```
<dc:language>de</dc:language>
```

Es gibt noch eine Reihe optionaler Dublin-Core-Elemente, mit denen Sie weitere Informationen über das Buch in die EPUB-Datei aufnehmen können. Die wichtigsten davon sind in der folgenden Liste aufgeführt. Das Präfix `dc:` dürfen Sie dabei nicht vergessen.

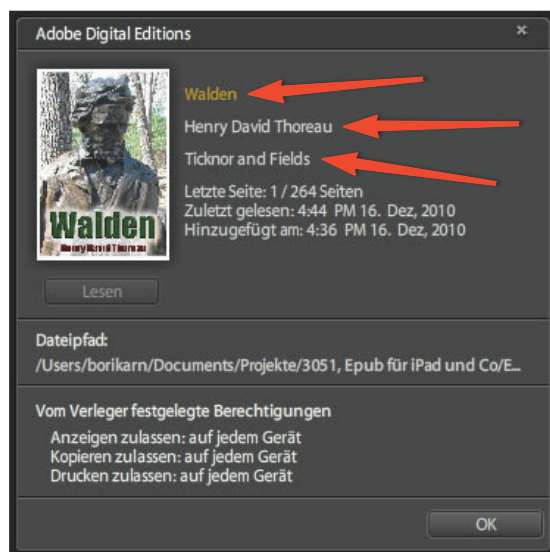
- `dc:creator` nennt den Autor des Buchs. Laut Spezifikation können Sie zwar mehrere `creator`-Elemente angeben, um Informationen über mehrere Urheber hinzuzufügen, doch habe ich noch keinen E-Reader gesehen, der mehr als einen anzeigt. Das Element `creator` verfügt über die beiden optionalen Attribute `opf:role` und `opf:file-as`. Welche Werte für `opf:role` zulässig sind (z.B. `aut` für den Autor), erfahren Sie unter <http://www.loc.gov/marc/relators/relaterm.html>. Mit dem Wert `opf:file-as` geben Sie an, wie der Autorenname alphabetisch eingeordnet werden soll. Ohne dieses Attribut erscheint Henry David Thoreau in E-Readern wie iBooks bei der Auflistung der Bücher nach Autor unter *H* statt unter *T*.
- `dc:subject` nennt wie in einer Bibliothek die Themenkategorie. Die Spezifikation gibt hier kein besonderes Vokabular vor und erfordert auch keines.
- `dc:description` kann z.B. eine Beschreibung des Buchinhalts enthalten oder bei einem alten und bekannten Werk auch einen Hinweis zu dessen Geschichte.
- `dc:contributor` wird zur Angabe der Personen verwendet, die sich in einem geringeren Maße an dem Buch beteiligt haben, z.B. für den Illustrator, den Lektor usw. Ebenso wie `creator` können auch die `contributor`-Elemente die Attribute `opf:role` und `opf:file-as` enthalten.
- `dc:publisher` gibt den Verlag an. Auch hier können Sie offiziell mehrere Elemente dieser Art angeben, wobei die derzeitigen E-Reader jedoch nur das erste anzeigen.
- `dc:rights` enthält Hinweise zum Copyright.
- `dc:source` gibt an, woher der Text ursprünglich stammt.

Der hintere Teil des Metadatenabschnitts von `content.opf` kann also in etwa wie folgt aussehen:

```
<dc:creator opf:role="aut">Henry David Thoreau</dc:creator>
<dc:contributor opf:role="bkp">Liz Castro</dc:contributor>
<dc:subject>19. Jahrhundert, Philosophie</dc:subject>
<dc:description>Walden (Erstveröffentlichung als "Walden; or, Life in the
```

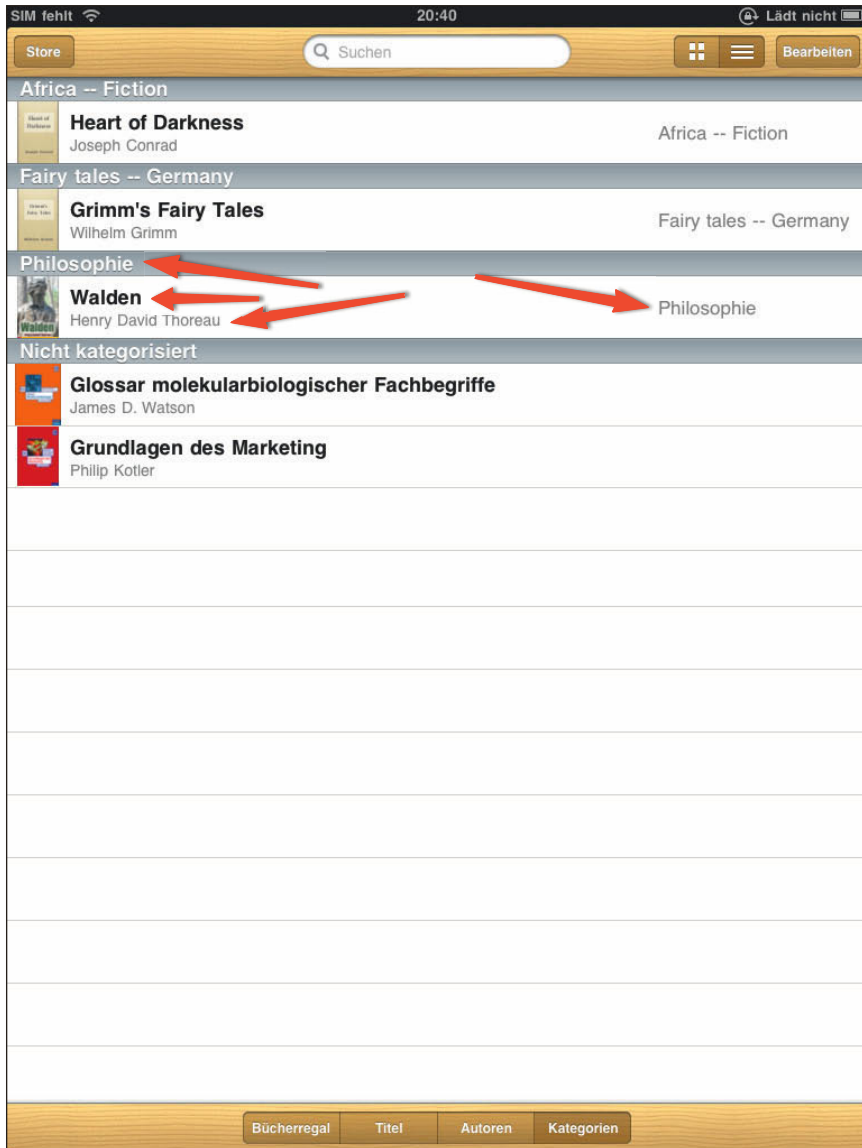
Woods") ist ein Buch des bekannten amerikanischen Transzendentalisten Henry David Thoreau. Das Werk ist persönliche Unabhängigkeitserklärung, gesellschaftliches Experiment, spirituelle Reise und Anleitung zur Selbstfindung zugleich.</dc:description>
<dc:publisher>Ticknor and Fields</dc:publisher>
<dc:source>Project Gutenberg eText 205</dc:source>

Wie viele dieser Metadaten angezeigt werden, hängt vom jeweiligen E-Reader ab. Wenn Sie für die zuvor gezeigte Datei in Adobe Digital Editions **MEDIENOPTI-ONEN > INFO ZUM MEDIUM** wählen, sehen Sie Folgendes:



Adobe Digital Editions zeigt den Titel, den Autor und den Verlag sowie interne Daten über die Seitenanzahl und das Datum, an dem das E-Book hinzugefügt wurde.

Im Programm iBooks auf dem iPad sehen Sie sogar noch weniger:



Die einzigen Metadaten, die in iBooks auf dem iPad angezeigt werden, sind der Titel, der Autor und die Themenkategorie.

Auch wenn die E-Reader zurzeit nicht alle Metadaten anzeigen, sollten Sie sie doch hinzufügen. Möglicherweise erledigen zukünftige E-Reader ihre Aufgabe ja besser, und selbst wenn nicht, sind diese Daten sinnvoll, denn ihr Hauptzweck besteht darin, dass die Computer Informationen über Ihre EPUB-Datei erhalten und sie auf sinnvolle Weise katalogisieren, sodass potenzielle Leser sie besser finden können.

Die Inhalte des Buchs auflisten (Manifest)

Der zweite Teil der Datei *content.opf* ist das *Manifest*, eine Auflistung der Dateien, aus denen sich das E-Book zusammensetzt. Dazu gehören der Text des Buchs (im Format XHTML), das CSS-Stylesheet für die Formatierung, die Bilder, die eingebetteten Schriftarten, also kurz gesagt alles, was sich in dem EPUB-Dokument befindet.

Hier müssen wir auch auf die Datei *toc.ncx* verweisen, die wir zuvor im Abschnitt *Die Datei toc.ncx für das Inhaltsverzeichnis* erstellt haben.

Das Manifest beginnt mit dem Element `manifest`, das ein `item`-Element für jede Datei im EPUB-Dokument enthält. Jedes `item`-Element wiederum hat ein `id`-Attribut, das die Datei eindeutig bezeichnet, ein `href`-Attribut, das den Speicherort der Datei relativ zum Manifest angibt, und ein `media-type`-Attribut, das angibt, um welche Art von Datei es sich handelt, was gewöhnlich in Form des MIME-Typs ausgedrückt wird. Die häufigsten Werte für das letzte dieser drei Attribute sind:

- `application/xhtml+xml` für alle HTML- und XHTML-Dateien
- `image/jpeg`, `image/gif`, `image/png` und `image/svg+xml` für JPEG-, GIF-, PNG- bzw. SVG-Bilder
- `font/opentype` für Open Type- und `font/svg` für SVG-Schriftarten
- `application/x-dtbnex+xml` für die Datei *toc.ncx*

Der Wert des `href`-Attributs muss auf die gesamte Datei zeigen, nicht nur auf einen Anker innerhalb der Datei (wie z.B. in `/datei.html#bestimmter_abschnitt`). Mit anderen Worten, das `href`-Attribut darf keine fragmentarischen Bezeichner enthalten.

Unser Manifest sieht wie folgt aus:

```
<manifest>
<item id="ncx" href="toc.ncx" media-type="application/x-dtbnex+xml"/>

<item id="walden-cc" href="Walden_cc.xhtml" media-type="application/
  xhtml+xml"/>
<item id="walden-c0" href="Walden_c0.xhtml" media-type="application/
  xhtml+xml"/>
<item id="walden-c0copy" href="Walden_c0copy.xhtml"
  media-type="application/xhtml+xml"/>
<item id="walden-c0note" href="Walden_c0note.xhtml"
  media-type="application/xhtml+xml"/>
```

```
<item id="walden-c1" href="Walden_c1.xhtml" media-type="application/
  xhtml+xml"/>
<item id="walden-c2" href="Walden_c2.xhtml" media-type="application/
  xhtml+xml"/>
  ... [und 16 weitere XHTML-Dateien, die hier angegeben werden]
<item id="WaldenPond-jpg" href="images/WaldenPondwaves.jpg"
  media-type="image/jpeg"/>
<item id="Thoreau_House-jpeg" href="images/Thoreau_House.jpg"
  media-type="image/jpeg"/>
<item id="cover-image" href="images/ThoreauSculptureCover.jpg"
  media-type="image/jpeg"/>
<item id="css" href="template.css" media-type="text/css"/>
</manifest>
```

In diesem Beispiel sind eine *toc.ncx*-Datei, 22 XHTML-Dateien (von denen hier nur sechs gezeigt werden), drei JPEG-Bilddateien und eine CSS-Stylesheet-datei angegeben.

Die Reihenfolge der Elemente im Buch festlegen (Abschnitt spine)

Der dritte Teil der Datei *content.opf* ist der Abschnitt *spine* (das »Rückgrat«), das dem E-Reader mitteilt, in welcher Reihenfolge die XHTML-Dateien zu lesen sind. Wenn der Reader am Ende einer Datei angelangt ist, bestimmt er anhand der Angaben im Abschnitt *spine*, welche er als nächste anzeigen soll. Falls Sie die EPUB-Datei mit InDesign erstellt haben, ist dieser Abschnitt bereits vorhanden.

Der Abschnitt beginnt mit dem Element *spine*, das über das Attribut *toc* verfügen muss. Der Wert dafür entspricht dem des Attributs *id* in dem Element, in dem im Manifest auf die Datei *toc.ncx* verwiesen wurde. (Zu diesem Zweck verwende ich stets den Wert *ncx*.)

Als Nächstes erstellen Sie ein *itemref*-Element für jede XHTML-Datei im E-Book. Jedes dieser Elemente muss über das Attribut *idref* verfügen, dessen Wert mit dem des Attributs *id* im zugehörigen *item*-Element des Manifests übereinstimmt.

In unserem Beispiel sieht das wie folgt aus:

```
<spine toc="ncx">
  <itemref idref="walden-cc"/>
  <itemref idref="walden-c0"/>
  <itemref idref="walden-c0copy"/>
  <itemref idref="walden-c0note"/>
  <itemref idref="walden-c1"/>
```

```
<itemref idref="walden-c2"/>  
... [und so weiter, bis:]  
<itemref idref="walden-c18"/>  
</spine>
```

Den Zweck der einzelnen Dateien im Buch angeben (Abschnitt *guide*)

Die Aufgabe von *guide*, dem vierten, optionalen Abschnitt, ist zwar zum größten Teil von der Datei *toc.ncx* übernommen worden, aber Apple empfiehlt dennoch, diesen Abschnitt zu verwenden. Hier können Sie angeben, welche Rolle die einzelnen Dateien Ihres Buches spielen. Zur Wahl stehen die Bezeichnungen *cover*, *title-page*, *toc* (Inhaltsverzeichnis), *index*, *glossary*, *acknowledgements* (Danksagung), *bibliography*, *colophon*, *copyright-page*, *dedication* (Widmung), *epigraph*, *foreword*, *loi* («list of illustrations», also Liste der Abbildungen), *lot* (Liste der Tabellen), *notes* (Hinweise), *preface* (Vorwort) und *text*.

Laut der Dokumentation von Apple wird beim Öffnen eines E-Books in iBooks als Erstes das erste Dokument angezeigt, das im Abschnitt *guide* den Typ *text* aufweist. Tatsächlich aber zeigt iBooks als Erstes das erste Element an, dem in *guide* der Typ *preface*, *acknowledgements*, *dedication*, *epigraph*, *foreword* oder *text* zugewiesen ist.

Ich finde es ein wenig verwirrend, wenn ein Buch auf einer anderen als der ersten Seite aufgeschlagen wird, wobei ich allerdings zugeben muss, dass ich das selbst immer so mache, wenn ich ein neues gedrucktes Buch zur Hand nehme. Allerdings stellt sich bei mir dann immer das Gefühl ein, als hätte ich irgendetwas verpasst, weshalb ich sofort zum Anfang zurückblättere. Andererseits ist es sicherlich sinnvoll, Ihren Lesern nicht erst einen scheinbar endlosen Vorsatz zu präsentieren, bevor sie mit der Lektüre beginnen dürfen. In diesem Buch Sorge ich dafür, dass das Buch beim ersten Öffnen auf der Seite mit dem Verlagshinweis geöffnet wird (die im Abschnitt *guide* als *preface*, also Vorwort, gekennzeichnet ist).

Wie vorauszusehen, beginnt der *guide*-Abschnitt mit dem Element *guide*. Anschließend werden alle XHTML-Dateien in der Reihenfolge ihres Erscheinens aufgeführt, wobei der Speicherort der Datei relativ zum Manifest sowie ihr Typ angegeben sind. (Die möglichen Werte für *type* finden Sie im ersten Absatz dieses Abschnitts).


```
<guide>
  <reference type="cover" href="Walden_cc.xhtml"/>
  <reference type="title-page" href="Walden_c0.xhtml"/>
  <reference type="copyright-page" href="Walden_c0copy.xhtml"/>
  <reference type="preface" href="Walden_c0note.xhtml"/>
  <reference type="text" href="Walden_c1.xhtml"/>
  <reference type="text" href="Walden_c2.xhtml"/>
  ... [und so weiter, bis:]
  <reference type="text" href="Walden_c18.xhtml"/>
</guide>
```

Eine vollständige Liste der Typen, die Sie im Abschnitt `guide` angeben können, finden Sie online in der OPF-Spezifikation unter http://www.idpf.org/2007/opf/OPF_2.0_final_spec.html#Section2.6.

InDesign erstellt den Abschnitt `guide` nicht, weshalb Sie ihn wie hier beschrieben selbst erstellen müssen, wenn Sie ihn angeben möchten.

Die Datei `content.opf` abschließen

Die Datei `content.opf` muss mit dem schließenden `package`-Element enden:

```
</package>
```

Das Cover gestalten

Das Cover eines E-Books ist genauso wichtig wie das eines gedruckten Buches und erfüllt denselben Zweck. Fast alle E-Reader haben irgendeine Art von Bücherregal- oder Bibliotheksfunktion, in der Sie alle Ihre E-Books einsehen können, und in den meisten Fällen können Sie diese Bibliothek entweder als Liste von Titeln oder als Darstellung der Umschläge anzeigen lassen. Auch in Onlineshops werden die Umschläge der E-Books verwendet, um die Aufmerksamkeit auf die einzelnen Bücher zu lenken, so wie richtige Buchhandlungen Bücher ins Schaufenster stellen, um mit den bunten Umschlägen Käufer anzulocken.

Sie können zwar in InDesign ein Dokument für das Cover anlegen, doch kennzeichnet das Programm diese Datei nicht automatisch als Cover, wie man es eigentlich erwarten sollte, sondern behandelt sie wie jede andere Datei des Buchs. Um das Dokument tatsächlich als Cover verwenden zu können, müssen Sie den Anweisungen in diesem Abschnitt folgen.

Tipp

Nachdem Sie ein EPUB-Dokument in iTunes kopiert haben, können Sie ihm ein Cover für iBooks hinzufügen. Markieren Sie dazu das Dokument im Abschnitt BÜCHER von iTunes, wählen Sie ABLAGE > INFORMATIONEN (⌘+I), klicken Sie auf den Reiter COVER und dann auf HINZUFÜGEN. Jetzt können Sie die Covergrafik auswählen. Dies funktioniert allerdings nur bei iBooks mit seinem völlig vom Standard abweichenden System und nicht bei anderen E-Readern. Im weiteren Verlauf dieses Abschnitts erfahren Sie, wie Sie ein Cover für alle E-Reader zuweisen, nicht nur für iBooks.

Im Vergleich zu seinem gedruckten Gegenstück weist das Cover eines E-Books einen erheblichen Nachteil auf: Meistens ist er nur ein kleines Symbol von vielleicht 100 x 150 Pixel Größe, das der E-Reader automatisch auf der Grundlage eines größeren Bildes erstellt. Der Herausgeber eines E-Books kann das größere Bild bereitstellen, hat aber wenig Einfluss darauf, wie das Coversymbol generiert wird. Daher ist es wichtig, das Cover so zu erstellen, dass es auch bei verringerter Größe ansprechend aussieht. Vor allem muss der Text im miniaturisierten Format noch deutlich zu lesen sein.

Das größere Bild des Covers wird häufig als Titelseite des E-Books selbst verwendet, was aber nicht unbedingt erforderlich ist.

Nachdem Sie das Coverbild gestaltet haben, müssen Sie einem festgelegten Verfahren folgen, um es als Cover festzulegen. Als Erstes müssen Sie eine XHTML-Datei erstellen, um das Bild anzuzeigen. iBooks erkennt ein Bildokument nicht von selbst als E-Book-Cover.

Das XHTML-Dokument für das Cover kann wie folgt aussehen:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.
  w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Walden_cc.xhtml</title>
  <link href="template.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div>
  
</div>
</body>
</html>
```

Beachten Sie, dass der Dokumentrumpf aus einem einzigen `div`-Element für das Coverbild besteht. Ein `img`-Element kann in XHTML nicht unmittelbar im `body`-Element stehen.

E-Reader erstellen zwar gewöhnlich das Symbol für das E-Book aus dem Coverdokument, aber dabei gehen sie natürlich nicht alle auf die gleiche Weise vor. Ich habe herausgefunden, dass Adobe Digital Editions nur den oberen linken Teil des Bildes verwendet, sofern Sie keine zusätzlichen Formatierungsangaben machen.



Dies ist nur die obere linke Ecke des Coverbilds.

Um ADE dazu zu zwingen, das gesamte Bild für das Symbol zu nutzen, müssen Sie dem `img`-Tag im XHTML-Dokument die Angabe `style="max-width: 100%"` hinzufügen.

Nachdem die XHTML-Datei jetzt komplett ist, müssen Sie der Datei `content.opf` noch einige Informationen hinzufügen, damit der E-Reader die Bilddatei auch als Cover erkennt.

Als Erstes fügen Sie dem Metadatenabschnitt von `content.opf` eine Zeile hinzu, die dem E-Reader mitteilt, welches Bild für das Cover verwendet werden soll:

```
<meta name="cover" content="cover-image" />
```

Der Wert des Attributs `name` muss stets `cover` lauten, und als Wert für `content` verwenden Sie die ID, die Sie im Manifest für das Coverbild festgelegt haben.

Erinnern Sie sich noch an das Manifest? Das ist die Liste, die die Namen aller in dem EPUB-Dokument verwendeten Dateien angibt, auch des Coverbilds. (Die Beschreibung des Manifests finden Sie im Abschnitt *Die Datei content.opf schreiben*.) Jeder dieser Dateien wird dort auch eine ID zugewiesen. Der Wert von `id` für das Coverbild muss mit dem Wert des Attributs `content` in dem hier gezeigten `meta`-Element übereinstimmen.

Zur Verdeutlichung sehen Sie hier den Manifesteintrag für die *Bilddatei* des Covers:

```
<item id="cover-image" href="images/ThoreauSculptureCover.jpg"
      media-type="image/jpeg"/>
```

Der Manifesteintrag für die *XHTML-Datei* des Covers unterscheidet sich nicht von den Einträgen für andere XHTML-Dateien. Achten Sie jedoch auf den Wert des Attributs `id`, da sie ihn im Abschnitt `spine` brauchen.

```
<item id="walden-cc" href="Walden_cc.xhtml" media-type="application/
      xhtml+xml"/>
```

Im Abschnitt `spine` gibt es ebenfalls Verweise auf sämtliche XHTML-Dateien im Buch, durch die die Reihenfolge bestimmt wird, in der ein E-Reader die einzelnen Bestandteile des E-Books anzeigt. Es gibt zwei Gründe dafür, dass die XHTML-Datei für die Coverseite als erster Eintrag im Abschnitt `spine` auftauchen sollte. Erstens wird sie dadurch zu dem ersten Element, das im Buch angezeigt wird, was der offensichtliche Platz für das Cover ist, und zweitens verwendet Adobe Digital Editions das erste unter `spine` aufgeführte Element, um das Coversymbol zu erstellen.

Jedes `itemref`-Element im Abschnitt `spine` verfügt über ein `idref`-Attribut, dessen Wert mit dem des `id`-Attributs im zugehörigen `item`-Element des Manifestabschnitts übereinstimmen muss. In diesem Fall ist `walden-cc` der Wert von `idref`, was mit dem zuvor gezeigten Eintrag für die XHTML-Datei im Manifest übereinstimmt.

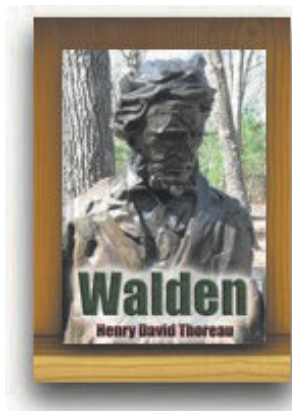
```
<itemref idref="walden-cc"/>
```

Zu guter Letzt empfiehlt Apple noch, im Abschnitt `guide` Informationen über das Cover unterzubringen. Ich habe auch Bücher ohne `guide`-Abschnitt schon erfolgreich in iBooks geladen, aber es besteht natürlich immer die Möglichkeit, dass sich iBooks oder andere E-Reader in Zukunft auf dieses Merkmal verlassen, weshalb ich Ihnen empfehle, diesen Abschnitt für den Fall der Fälle hinzuzufügen.

```
<reference type="cover" title="Cover von Walden" href="Walden_
      cc.xhtml"/>
```

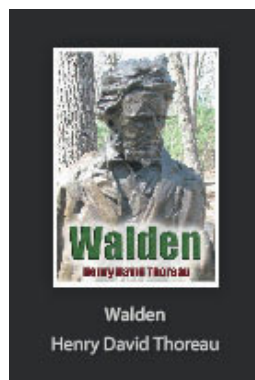
Für das Attribut `type` müssen Sie hier den Wert `cover` verwenden. Der Wert von `href` muss auf die XHTML-Datei für die Coverseite zeigen (nicht auf die eigentliche Bilddatei).

Wenn Sie sich an diese Anleitung halten, sehen Sie auch in iBooks ein schönes Symbol für das Cover:



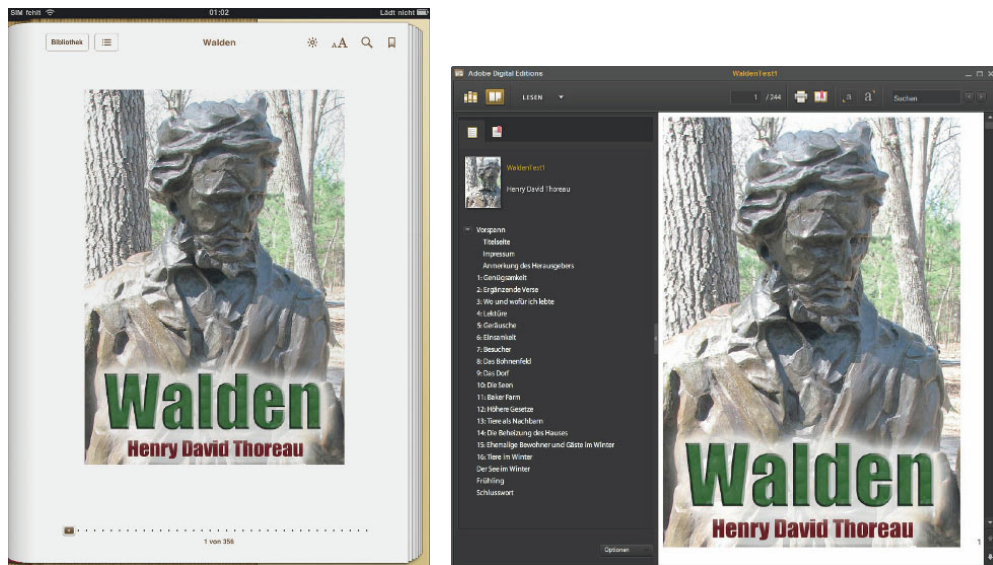
Im Bücherregalmodus zeigt iBooks eine mit guter Qualität verkleinerte Version des Coverbilds.

In Adobe Digital Editions ist das Symbol akzeptabel:



Adobe Digital Editions erstellt auf der Grundlage der XHTML-Seite für den Cover ein sehr pixeliges, aber immer noch erkennbares Coversymbol.

In allen E-Readern, die Bilder anzeigen können, sehen Sie außerdem eine sehr ansprechende Coverseite:



Links iBooks, rechts Adobe Digital Editions.

Legende zur Abbildung rechts

- 1 Metaelement mit der Namensangabe "cover" für den Verweis auf das Cover
- 2 toc in spine muss mit der ID von *toc.ncx* im Manifest übereinstimmen
- 3 Die idref-Einträge in spine müssen mit den IDs im Manifest übereinstimmen
- 4 Der Inhalt von `dc:title` in den Metadaten bestimmt den in E-Readern angezeigten Titel
- 5 href-Einträge in guide und Manifest geben Dateiname und -pfad an
- 6 content im Metaelement für das Cover muss mit der ID des Coverbilds im Manifest übereinstimmen
- 7 Die XHTML-Datei für das Cover verweist auf das Coverbild

