

Elektronische Bücher mit EPUB: Aufzucht und Pflege

Anselm Lingnau

Linup Front GmbH

FrOSCon 2011



Agenda

- ▶ Elektronische Bücher und EPUB
- ▶ Anatomie einer EPUB-Datei
 - ▶ Inhalte
 - ▶ Dateistruktur
- ▶ EPUB-Dateien selber machen
 - ▶ Zu Fuß
 - ▶ Sigil
 - ▶ Calibre
 - ▶ Writer2ePub
 - ▶ Sphinx
 - ▶ L^AT_EX

Elektronische Bücher und EPUB



Vom Buch zum E-Book?

- ▶ Elektronische Bücher sind im Aufwind
 - ▶ Belastbare Zahlen sind schwer zu finden
- ▶ Lesbar auf
 - ▶ PCs
 - ▶ Tablets (iPad, Kindle Fire, ...)
 - ▶ Speziellen Lesegeräten (Sony, Kindle, Kobo, ...)
 - ▶ Smartphones (naja)
 - ▶ ...?



Formate

- ▶ PDF
 - ▶ Gut verstanden, ISO-Standard
 - ▶ Formatierung und Schriften liegen fest (Vorteil und Nachteil)
 - ▶ Jede Menge verbreitete Software (LibreOffice, \LaTeX , ...)
- ▶ »Kindle-Format«
 - ▶ Amazon-proprietär, ehemals Mobipocket (+ DRM)
 - ▶ ziemlich primitiv (Kindle Fire: eigenes neues Format KF8)
 - ▶ Konvertierungs-Tools von Amazon
 - ▶ Anzeigeprogramme von Amazon (Linux: Windows-Version mit WINE)
- ▶ EPUB
 - ▶ International Digital Publishing Forum (IDPF)
 - ▶ offener freier Standard
 - ▶ Glorifiziertes XML/HTML/CSS
 - ▶ Diverse Software aus unterschiedlichen Quellen

EPUB-Standards

- ▶ Alt: EPUB 1 (Seit September 2007)
 - ▶ Vorläufer »Open eBook«
 - ▶ Erster Kindle: November 2007
- ▶ Verbreitet: EPUB 2.x (seit Mitte 2010)
 - ▶ Wird von gängigen E-Book-Readern implementiert
- ▶ Aktuell: EPUB 3.x (seit Herbst 2011)
 - ▶ Unterstützung noch sehr lückenhaft



EPUB 2.0.1: Was ist erlaubt?

»Open Publication Structure 2.0.1«

- ▶ Textinhalt: XHTML oder DTBook (XML vom Daisy Consortium)
- ▶ Formatierung: Teilmenge von CSS 2.0, plus:
 - ▶ Kopf- und Fußzeilen
 - ▶ Eingebettete Schriften
- ▶ Grafikformate: PNG, JPEG, GIF, SVG
- ▶ Zeichencodierung: Unicode (UTF-8 oder UTF-16)
 - ▶ Nicht jedes Zeichen muss dargestellt werden können



EPUB 3.0: Was ist erlaubt?

»EPUB Content Documents 3.0«

- ▶ HTML5 (XML-Syntax) mit Erweiterungen und Einschränkungen
- ▶ CSS 2.1 mit Zusätzen aus CSS3
- ▶ Eingebettete Schriften (OpenType und WOFF)
- ▶ JavaScript ist erlaubt (unter Umständen)
 - ▶ Privatsphäre- und Sicherheits-Alptraum
- ▶ Audio und Video
 - ▶ Evtl. problematisch (Video vs. eInk)
- ▶ Andere Metadaten-Organisation



Anatomie einer EPUB-Datei



The Big Picture

- ▶ ZIP-Archiv mit:
 - ▶ OPF-Datei (content.opf)
 - ▶ Metadaten, Dateiliste, Kapitelreihenfolge
 - ▶ NCX-Datei (toc.ncx)
 - ▶ Ausführliches Inhaltsverzeichnis
 - ▶ Kleinkram
 - ▶ Textdateien (X?HTML), Bilder (PNG, JPEG, ...) und ähnliche Nutzdaten
 - ▶ Stylesheets



The Big Picture (Forts.)

grimm.epub

mimetype	<i>»application/epub+zip«, unkomprimiert</i>
META-INF/	<i>Vorgeschrieben</i>
container.xml	<i>Gibt an, wo die OPF-Datei steht</i>
OEBPS/	<i>Name ist Konvention</i>
content.opf	<i>OPF-Datei</i>
toc.ncx	<i>NCX-Datei</i>
chapter1.xhtml	<i>Erstes Kapitel</i>
chapter2.xhtml	<i>Zweites Kapitel</i>
picture.png	<i>Bild</i>
...	
css/	
style.css	<i>Stylesheet</i>

EPUB-Dateien selber machen



Zu Fuß

- ▶ Vorgehensweise
 - ▶ Inhalte in (X)HTML schreiben
 - ▶ Stylesheets usw. zusammentragen
 - ▶ EPUB-Dateistruktur aufstellen
 - ▶ `container.xml` ist (ziemlich) fest
 - ▶ OEBPS-Verzeichnis füllen
 - ▶ Dateistruktur zusammenpacken (zip o.ä.)
 - ▶ Vorsicht: mimetype unkomprimiert!
- ▶ Vorteil
 - ▶ Volle Kontrolle
- ▶ Nachteil
 - ▶ Mühselig



EPUB-Datei einpacken (in Python)

```
import sys, os, re
from zipfile import ZipFile, ZIP_STORED, ZIP_DEFLATED
container_xml = """..."""

epubname = sys.argv[1] + ".epub"
zip = ZipFile(epubname, "w", ZIP_DEFLATED)
zip.writestr("mimetype", "application/epub+xml", ZIP_STORED)
zip.writestr("META-INF/container.xml", container_xml, ZIP_DEFLATED)

root_re = re.compile('^[^/]*')
for root, dirs, files in os.walk(sys.argv[1]):
    # dir/foo/bar => OEBPS/foo/bar/
    zip_prefix = root_re.sub('OEBPS', root) + '/'
    for f in sorted(files):
        if not f.endswith('~'):
            zip.write(os.path.join(root, f), zip_prefix + f)
zip.close()
```



Sigil

- ▶ Siehe <http://code.google.com/p/sigil/>
- ▶ Multiplattform-EPUB-Editor
 - ▶ Import aus verschiedenen Formaten
 - ▶ Text, RTF, HTML, ...
 - ▶ Quelltext- und gerenderte Darstellung
 - ▶ Metadaten
- ▶ Vorteile
 - ▶ Intuitiv und unproblematisch
- ▶ Nachteile
 - ▶ »Einbahnstraße«: Nacharbeiten in Sigil müssen bei Änderungen am Originaltext ggf. manuell wiederholt werden

Calibre

- ▶ Siehe <http://calibre-ebook.com/>
- ▶ Konverter zwischen diversen E-Book-Formaten
 - ▶ EPUB, Mobi (Kindle), PDF, Text, HTML, ...
- ▶ Datenübertragung zu E-Book-Reader
- ▶ Vorteile
 - ▶ Flexibel
- ▶ Nachteile
 - ▶ Autor sagt: »Do not use for creation«



Writer2ePub

- ▶ Erweiterung für LibreOffice
- ▶ Erzeugt EPUB aus LibreOffice-Text-Dokumenten
- ▶ Benötigt mit Formatvorlagen ausgezeichnete Dokumente
 - ▶ Ist sowieso eine gute Idee
- ▶ Ausgabequalität läßt zu wünschen übrig



Sphinx

- ▶ Dokumentationssystem (Python)
 - ▶ Dokumente schreiben in reST (textbasiertes Format)
 - ▶ Konvertieren in HTML, EPUB, PDF, ...
- ▶ Vorteile
 - ▶ Leistungsfähiges Eingabeformat (für Programm-Doku)
 - ▶ Erweiterbar und anpassbar
 - ▶ Reproduzierbarer Prozess
 - ▶ Macht gute Webseiten
- ▶ Nachteile
 - ▶ EPUB und PDF (via \LaTeX) involvieren (noch) Bastelei
- ▶ Es gibt diverse ähnliche Ansätze (Pandoc, ...)

L^AT_EX

- ▶ Textsatzsystem
- ▶ Keine direkte EPUB-Unterstützung
- ▶ Trotzdem interessant
 - ▶ Eingeführt und leistungsfähig für Print-Ausgabe
 - ▶ Gut verstanden, große Community
- ▶ Typischer Ansatz: L^AT_EX nach EPUB via HTML
 - ▶ Verschiedene Konverter von L^AT_EX nach HTML verfügbar
 - ▶ Weiter siehe »Zu Fuß«
- ▶ Hier könnte man sich noch populär machen

Zum Schluss:

Fragen???

Kontakt über ...

anselm.lingnau@linupfront.de

<http://www.linupfront.de/>

