

Section 1 EPUB 3の概要

EPUB 3とは

電子書籍などを統一した規格で制作し、永続的に利用しやすくしていくために考えられたのが、EPUBフォーマットである。ここでは、まずEPUBの成り立ちと策定についての概要を知っておこう。

電子出版のためのオープンな仕様

「EPUB」とは、電子出版 (Electronic PUBlication) コンテンツの形式を標準化するために、電子出版を推進する企業らで構成される標準化団体、国際電子出版協会 (International Digital Publishing Forum、IDPF) が共同で制定した、オープンな推奨仕様である。

EPUBは、さまざまなデバイスで共通のファイルが使用でき、かつ使用するデバイスに適したコンテンツ表示が行えることを目標としたオープンファイルフォーマットである。その実体は、拡張子が「.epub」のファイルだが、このファイル自体はいくつかのファイル群をZIP圧縮して1つのファイルとしてまとめたただけのものである。ウェブページを構成しているファイル群をZIP圧縮にして持ち運びしやすい1ファイルにすると考えると、イメージしやすいかもしれない。

EPUBとは、ZIPファイル内に格納するファイルの内容やディレクトリの種類、場所を定めた規格ということになる。

<http://idpf.org/>
idpfのウェブサイト。EPUBの仕様書も公開されているほか、最新のニュースが得られる。



身近なEPUBの採用例には、iPadやiPhoneで利用できるアップルの「iBookstore」がある。Google Booksのような巨大なサイトでもEPUBを採用している



EPUBにはいくつかバージョンがあるが（下図を参照）、2011年10月には最新の仕様であるEPUB 3の最終仕様として可決された。

EPUBコンテンツはXHTML、CSS、SVG、およびウェブ標準の画像ファイルを使って表現される。隔々までEPUBで独自の策定を行うのではなく、ウェブ技術の標準化団体であるW3Cによって策定されたウェブの標準技術を積極的に使用しているため、すでにウェブ向けに作成したコンテンツも比較的簡単にEPUB用に再構成できるという大きなメリットがある。

今後は、実際にEPUB 3の仕様に対応するリーディングシステム（EPUBリーダー／ビューアー）と呼ばれるソフトウェアやハードウェアが出てくることが待ち望まれる。

1999年	EPUB (OEBPS) 1.0 公開 (XHTML1.0+CSS2 採用)
2002年	EPUB (OEBPS) 1.2 公開
2007年	EPUB (OPF/OPS) 2.0 公開 (DTBook、SVG 採用)
2010年	日本語拡張案提出 (JEPA)
2011年	EPUB 3 策定
201X年	固定レイアウトモードの追加
201X年	今後追加される HTML5、CSS3 機能の採用

EPUBバージョンの概略

EPUBで利用されている主な技術

XHTML5	ウェブページ記述に使用されるマークアップ言語。テキストで記述され、HTML自体でテキストや表、リストなどの表現ができるほか、画像などの外部メディアを同一ページに組み込むことができる。策定、勧告はW3C。XHTML5は、XML構文で記述したHTML5。
SVG 1.1	XMLによって記述されたベクターグラフィックスファイル。ビットマップとは異なり、拡大縮小しても解像度が低下しない。Adobe IllustratorやMicrosoft Office Visio、GIMPなどでは、SVGの読み書きに対応している。モダンブラウザはすでにSVGの表示に対応している。
CSS2.1、CSS3	XMLやHTML内の要素について、表示方法を記述したスタイルシート。HTMLかXMLに組み込むことも、別ファイルとして独立させることもできる。
JavaScript	ウェブブラウザなどで動作するスクリプト言語。動的なウェブページの実現に使用される。
TrueType/WOFF	テキスト表示に使用される書体（フォント）のデータ。ベクターグラフィックスとして表されている。TrueTypeはPCで一般的に使われてきたフォント形式。WOFF（Web Open Font Format）はMozillaなどのウェブ系ベンダーが策定したフォント形式。
SSML/PLS/ CSS3 Speech	テキストの読み上げに関連する技術。SSML（Speech Synthesis Markup Language）は読み上げの付帯情報のマークアップ、PLS（Pronunciation Lexicon Specification）は発音の定義に使用する。CSSではバージョン3でテキスト読み上げをサポートした。
SMIL 3	ウェブ上で再生される、時間軸のあるメディアを同期させるためのマークアップ言語。Synchronized Multimedia Integration Languageの略でスマイルと読む。EPUBではテキスト読み上げの音声と画面表示の同期に使用される。
XML	テキストに構造情報や意味を付加するためのマークアップ言語。
ZIP	ファイルの圧縮形式。複数のファイルを束ねて1つのファイルにする。

EPUBの特徴

EPUBの最大の特徴はリフロー型、あるいは流し込み型と呼ばれる方法を採用していることだ。EPUBが対象とするデバイスの種類はPCからタブレット、スマートフォンまで多岐にわたる。テキストが画面サイズに合わせてリフローすることにより、スクリーンサイズが違うさまざまなデバイス上のそれぞれのディスプレイに最適な文字サイズでレイアウトされるため、1ファイルで多デバイスに対応できるファイルとなっているのだ。

このリフローにより、同じEPUBファイルでも、パソコンディスプレイなどの大きなスクリーンサイズで開いた場合は30ページの書籍が、スマートフォンなどの小さなディスプレイで開いた場合は例えば100ページほどの書籍となる。また、同じデバイス上であっても、ユーザーがリーディングシステム上で文字サイズを変更すれば、それだけで全ページ数は変わってくる。ウェブページでテキストを閲覧する際、ブラウザの「フォントサイズ」でユーザーが文字の大きさを変更でき、それに従って文字が流れる（リフローする）のと同じことである。

EPUBがウェブページと異なるのは、1コンテンツを「スクロール」して読むのではなく、「書籍」に似つかわしく、常に1画面を1ページと捉えていることだ。また、ページ数が変わっても、章ごとに改ページしたり、目次から読みたい場所を参照できるなどの書籍的な機能も、もちろん備えている。

Readability と Webデザイン

ちょうど1年前にSafariに「リーダー」という機能が搭載されました。リーダーとは、Webページから装飾や広告を取り除き、本文を読みやすく整形して表示するものです。この機能がついにこの秋に登場する「iOS 6」でも追加されます。その1年前には、ボクのブログでも電子書籍の話ついでにSafariのリーダーについてふれてみたり、iPadでリーダーと同等の表示にする手順を解説しています。

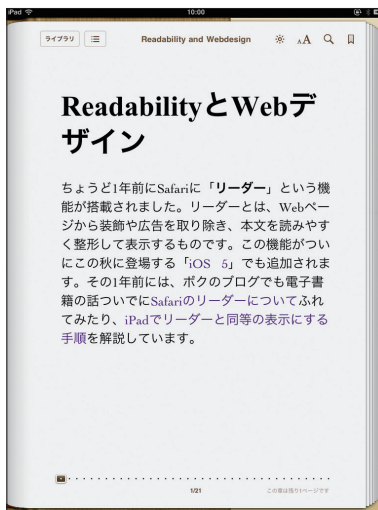
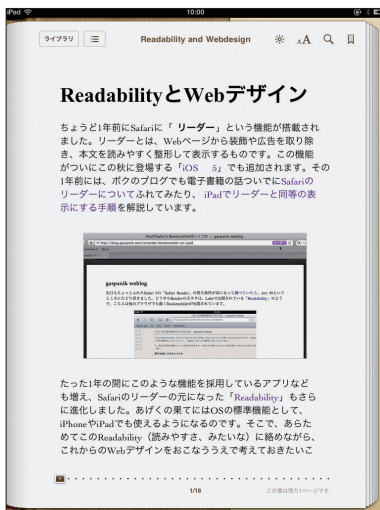


下の同じコンテンツをAndroidスマートフォンのEPUBリーディングシステムで表示した。ここではフォントもデバイスにより変わっているのが分かる。

HINT

固定レイアウトについて

絵本や図版の多い解説書、雑誌を配布したい場合、ページ内での表示位置が定められないリフロー表示は不都合が多い。そこでEPUB3の追加仕様として、画面内でのオブジェクトの位置を調整する「固定レイアウト (Fixed Layout)」を仕様に加える可能性が強まっている。アップルiBooksのFixed LayoutやBarnes & Noble Fixed Format、Sony Fixed Format、Amazon Kindle Format 8などの独自実装を踏まえ検討されている。



iPadの電子書籍リーディングシステム「iBooks」でサンプルのEPUBドキュメントを表示した。標準的なフォントサイズ(左)で表示した場合は18ページのボリュームのドキュメントも、アプリでフォントサイズを上げるとコンテンツがリフローされて、総ページも自動的に21ページへ変化した。

EPUBとPDFの違い

EPUBの特徴をより明確にするために、従来より電子ドキュメントのファイルフォーマットとして使われているPDFと比較してみよう。

PDFは印刷物と同じ内容を電子ドキュメントとして再現することを目的としたフォーマットで、ページサイズとコンテンツの位置は固定される。そのため、どのデバイスで閲覧しても、画像とテキストが意図せぬ場所へ配置されてしまう心配はない一方で、大きなディスプレイでは快適に閲覧できるドキュメントが、スマートフォンでは細かい文字が読めずに拡大・スクロールをしながらでないと読むことができないなどの不都合が起きる。もし小さなディスプレイでも読ませたいなら、デバイスに適した版面設計を行ってファイルを作り分ける必要がある。

このように、EPUBは、デザインには制約があるもののデバイスに適した快適なリーディングエクスペリエンスが保証される。PDFはページ単位が絶対的な基準であり、意図したデザインをそのまま再現できる。電子書籍化したいコンテンツの特性に応じて、この2つのどちらを使用するかを判断することになる。

もうひとつ、ワークフローと制作コストの問題もある。商業印刷用のレイアウトデータはほとんどの場合、PDFをワンクリックで生成可能なAdobe InDesignなどのアプリケーションで作成されている。そのため、印刷データを作成後にEPUBを生成するのは別コストがかかるが、PDFならばほとんどコストがかからない。

一方で、ウェブページコンテンツの場合は、比較的EPUBを作成しやすいと言える。EPUBのコンテンツドキュメントは、おおまかなところでは、ウェブページと同じようにして作れるからだ。



PDFファイルの例。どんな誌面デザインでも保持できる。テキスト検索なども行える(制作時の仕様による)。書籍は、PDFデータを書き出しできるAdobe InDesignやQuarkXPress、Adobe Illustratorといったソフトで制作されるので、書籍を電子化するニーズには向いている。また、各種デバイス向けにPDFリーダーソフトウェアは多数存在する。

Section 1 EPUB 3の概要

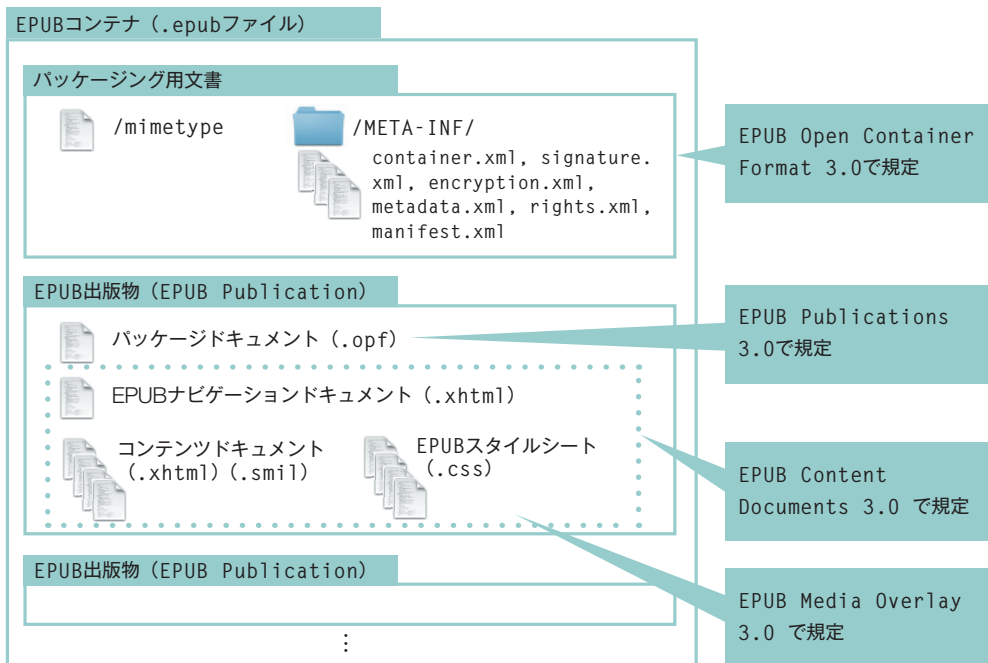
EPUB 3の仕様について

EPUBは、EPUBパッケージの構成を記述したパッケージドキュメント、目次として提示されるナビゲーションドキュメント、そして実際のページ内容となるコンテンツドキュメントで構成される。これを制定しているEPUB 3の4つの仕様書の概要を解説する。

EPUB 3の仕様

EPUB 3規格は、国際電子出版協会（IDPF）が公開している、4つの仕様書で策定されている。これらは2011年5月に草案として提出され、同年10月に最終仕様として確定した。

EPUBファイルは「書籍部分」（仕様書では"EPUB Publication"、本書では以下「EPUB出版物」と記載）と「パッケージング部分」に分かれている。「書籍部分」を全体として定義している仕様書がEPUB Publications 3.0で、ページ単位での表現方法を規定している仕様書がEPUB Content Documents 3.0だ。そしてEPUBを1つの書籍ファイルとしてまとめるための「パッケージング部分」はEPUB Open Container Format 3.0で規定



されている。

また、書籍部分に含まれる、テキスト、動画、音声を同期させる方法について規定しているのがEPUB Media Overlay 3.0である。また、仕様には含まれていないが、概要を説明したEPUB 3 Overviewというドキュメントも提供されている。

EPUB 3 Overview

このドキュメントはEPUB 3の全体像を説明するためのもので、仕様として含まれているものではない。EPUBを構成する「書籍」「パッケージング」という概念の説明と、EPUBを構成するファイルの簡単な説明がある。また、グローバル言語対応、アクセシビリティの向上という点からもEPUBの仕様が説明されている。

4つの仕様の規定内容

EPUB Publications 3.0

EPUB出版物に含まれるファイル群を構成するための仕様が定められている。コンテンツ作成者が従うべき仕様と、コンテンツを再生するリーディングシステムが遵守しなければならない要件が記載されている。

EPUB出版物は下記のファイルで構成される。

1. パッケージドキュメント（書籍情報を記載した.opfファイル、必須）
2. EPUBナビゲーションドキュメント（目次としてユーザーに提示される.xhtmlファイル、必須）
3. EPUBコンテンツドキュメント（ユーザーが読むコンテンツ、必須）
4. EPUBスタイルシート（CSS2.1+CSS3の一部、任意）
5. 発音辞書ドキュメント（任意）

*フォントについては27ページを参照。

この仕様書では、1のパッケージドキュメントに記述すべき内容が説明されている。2～5に記述する内容は、「EPUB Content Documents 3.0」に記載されている。また、EPUBファイルに含めることのできるメディアの種類である「EPUB 3コアメディアタイプ」もこの仕様書で定義されている。EPUB 3コアメディアタイプは、基本的にリーディングシステムで処理および表示できることが求められる。

コアメディアタイプ(Core Media Types)

画像	GIF、JPEG、PNG、SVG
アプリケーション	XHTMLコンテンツドキュメント、パッケージドキュメント（OPF2）、OpenTypeフォント、WOFFフォント*、メディアオーバーレイドキュメント、PLS（EPUB Text-to-Speech発音辞書）
オーディオ	MP3、AAC LC、MP4
テキスト	EPUBスタイルシート（CSS2.1+CSS3の一部）、JavaScript
ビデオ（推奨として掲載）	H.264、VP8

■ EPUB Content Documents 3.0

この仕様書では、EPUBファイルのEPUB出版物に含まれるファイル群のうち、内容の表現に使用されるXHTML (HTML5)、SVG、スタイルシート (CSS) のプロファイルを細かく規定している。EPUB 3を構成するファイル群のうちXHTMLによって記述されるのは、XHTMLコンテンツドキュメント*1、およびEPUBナビゲーションドキュメント*2 (目次ファイル) だ。「EPUB Content Documents 3.0」では、これらで使用されるプロファイル (HTML5) が定義され、拡張子が.xhtmlであることが規定されている。

EPUBナビゲーションドキュメントは、その書籍に関する必要な構成情報が記載されるだけでなく、ユーザーがナビゲーションに使用する目次ページとしても機能するものだが、その詳細についてもここで確認できる。

このほかにもEPUBで使用可能なスクリプト*3に関する記載、EPUBスタイルシートについての説明、拡張HTMLとして、数式表現を行うためのMathML、ベクトルグラフィックスを表現するSVGに関する規定も含まれている。

*1 XHTMLコンテンツドキュメントの記述については45ページ参照。

*2 EPUBナビゲーションドキュメントの記述については53ページを参照。

*3 スクリプトについての説明は24ページを参照。

■ EPUB Open Container Format 3.0

この仕様書では、EPUBで使用するファイル群を、.epubファイルとして1つのファイルにまとめるための要件を定めている。.epubファイルに含めなければならないファイルの一覧とパッケージ内のディレクトリ構成は、この仕様書で確認できる。下記のファイルの詳細が記述されている。

1. container.xml…このファイルは、.epubファイルの中で最初に参照されるファイル (エントリポイント) となる (必須)
2. signatures.xml…電子署名に関連するファイル (任意)
3. encryption.xml…暗号化に関するファイル (任意)
4. metadata.xml…コンテナのメタデータを格納する。将来での利用のために名前が予約されているだけで、現時点では利用されていない (任意)
5. rights.xml…DRM (デジタル著作権管理) 関連の情報を格納する (任意)

また、.epubファイルの実態はZIP圧縮されたファイル群であるが、このZIP圧縮の仕様についてもここに記載されている。

EPUB出版物にはフォントファイルを含めることもできるが、このフォントファイルをコピーしづらくする (難読化する) 方法についても説明されている。フォントの難読化に関しては27ページも参照してほしい。

■ EPUB Media Overlay 3.0

EPUBで、文字と音声や動画を同期させる方法を定義した仕様書である。EPUBでは、W3C仕様である「SMIL」(Synchronized Multimedia Integration Language) の一部が「メディアオーバーレイドキュメント」として利用可能である。この仕様書では、メディアオーバーレイドキュメントの作成方法と、EPUB出版物に含める方法が説明されている。246ページのコラムを参照してほしい。

EPUB 2とEPUB 3の違い

本書で解説するEPUBのバージョンは3.0であるが、もともとEPUBはOpen eBookという電子出版の仕様を引き継ぐ形で制定されたのが始まりである。2007年にはこの改訂版となるEPUB 2.0が勧告された。この2.0の問題点を修正したバージョンが、2010年6月に勧告されたEPUB 2.01である。アップルが展開するiBook、ソニーのSony Reader、グーグルのGoogle BooksなどでEPUB 2.0に対応したことで、EPUBは一気に注目を浴びるフォーマットとなった。

今後はEPUB 2.0では実現できなかった各種の機能を取り入れたEPUB 3.0へシフトしていくものと予想されるが、EPUB 3が普及するまではEPUB 2が引き続き利用されていくので、EPUB 2の概要についても把握しておくといだろう。

EPUB 2.0/2.01から変更された箇所

違いを示しやすいように、まずEPUB 2.0/2.01のファイル構成を以下に示す。

- mimetype (EPUB共通)
- META-INF/container.xmlファイル (EPUB共通)
- content.opfファイル (EPUB共通)
- EPUBナビゲーションドキュメント (.ncxファイル)
- EPUBコンテンツドキュメント (.xhtml, .css, .jpeg, .pngなど)

上記のEPUBの主な構成ファイルのうち、「EPUBナビゲーションドキュメント」「EPUBコンテンツドキュメント」では大きく変更があった。

EPUBコンテンツドキュメントの仕様変更

EPUB 2では、EPUBコンテンツドキュメントのうち文書情報を記述するものにはXHTML 1.1またはDTBookと呼ばれる形式が採用されていたが、EPUB 3ではHTML5に統合される形となった。HTML5はXHTML 1.1とHTML 4の後継となるバージョンで、より幅広いコンテキストでコンテンツを記述できる。

同じくEPUBコンテンツドキュメントの一種であるEPUBスタイルシート（または単にスタイルシートと呼ばれている）にも変更がある。EPUB 2ではEPUBスタイルシートにCSS2が採用されていたが、EPUB 3では「CSS2.1+CSS3の一部」が採用された。CSS3の一部機能というのが、EPUB 3の日本語表示における肝の部分であり、これにより日本語の縦書きやルビ、数式表現などのサポートが実現した。

EPUB 3で対応となった、CSS3の一部とは、具体的には以下の項目である。

- CSS3 Speech Module (テキストの読み上げに関する情報)
- CSS Font Module Level 3の@font-faceルール (フォントフェイスの指定)
- CSS Text Level 3:-hyphens, -line-break, -text-align-last, -text-emphasis, -text-emphasis-color, -text-emphasis-style, -word-break, text-transform, -epub-fullwidth, -epub-fullsize-kana

- CSS writing-mode (direction, unicode-bidiを除く)
- CSS名前空間の@namespaceルール
- CSS Multi-column Layout Module (ただし多用すべきでない)
- Media Queriesの@mediaルール、@importルール
- CSS3 Ruby Module : -epub-ruby-position
- CSS2.1のdisplayプロパティ : oeb-page-head, oeb-page-foot

■ EPUBナビゲーションドキュメントの仕様変更

.epubファイルの目次として機能するEPUBナビゲーションドキュメントは、EPUB2.0/2.01では.ncxファイルが使用されていた。このファイルの実態はXMLファイルで、スキーマ(格納されるデータの種類と、データ形式の定義)には、DAISYコンソーシアムの策定したNCXドキュメント定義(DTD)が使用されている。

DAISYコンソーシアムとは、デジタル録音図書の規格化を進めてきた団体だ。DAISYとはDigital Accessible Information Systemの略で、電子書籍をテキストと音声で提供するための「DAISY 3」という規格を制定し推進してきた。今回のEPUB 3の策定にはDAISYコンソーシアムも大きく関与しており、結果としてDAISY規格がEPUB 3に融合される形となった。

EPUB 3のEPUBナビゲーションドキュメントは、拡張子が.xhtmlのXML構文で書かれたHTML5ファイルとなり、従来の.ncxファイルに含まれている内部利用のための目次機能と、実際に目次ページとして読者に提示する内容の両方を兼ねている。

EPUB 3の仕様では、EPUBナビゲーションドキュメントに加えて、EPUB2の.ncxファイルを指定することも可能となっている。これはEPUB 3として作成したコンテンツを、EPUB2対応のリーディングシステムでもある程度開けるようにするための措置であり、EPUB 3としては.ncxファイルは使用しない。「ある程度」というのは、EPUB 3のコンテンツドキュメントで使用されるHTML5やCSS2.1+CSS3は、EPUB2のリーディングシステムが再現できる保証がない、ということである。

■ マルチメディアファイルへの対応

EPUB 3ではHTML 5のvideo要素、audio要素にも対応し、EPUBコンテンツに音声や動画を組み込むことも可能となった。同時に、音声の再生とテキスト表示とを同期させる方法も「EPUB Media Overlay」として定義され、各テキストがどの音声ファイルのどの部分に含まれているかを明確に指示することができる。

■ EPUBで使われるファイル形式

EPUBのコンテンツを構成するのが「EPUBコンテンツドキュメント」である。EPUBコンテンツドキュメントとして推奨されるデータの種類の「コアメディアタイプ」として定められ(19ページも参照)、EPUBファイルはこれらのファイル形式で構成されることが望ましい。なお、EPUBのコアメディアタイプはすべてウェブ標準となっているファイル形式である。

種類	メディアタイプ	説明
画像	image/gif	GIF 画像ファイル
	image/jpeg	JPEG 画像ファイル
	image/png	PNG 画像ファイル
	image/svg+xml	SVG コンテンツドキュメント
アプリケーション	application/xhtml+xml	XHTML コンテンツドキュメント
	application/x-dtbncx+xml	OPF2 (EPUB2 のパッケージドキュメント)
	application/vnd.ms-opentype	OpenType フォント
	application/font-woff	WOFF フォント
	application/smil+xml	メディアオーバーレイドキュメント
	application/pls+xml	テキスト読み上げ発音辞書
オーディオ	audio/mpeg	MP3 オーディオファイル
	audio/mp4	MP4 AAC LC オーディオファイル
テキスト	text/css	EPUB スタイルシート
	text/javascript	スクリプト
ビデオ	video/mp4、video/h264	H.264 動画ファイル
	video/webm、video/VP8	VP8 動画ファイル

H.264 動画ファイル、VP8 動画ファイルは、未だサポート推奨という扱いである。EPUB 3 はあくまで推奨仕様であるため、リーディングシステムが仕様通りに実装するとは限らない。これらのすべてのファイルがすべての EPUB 対応のリーディングシステムで対応するとは限らず、またここに掲載されていないファイルが使用されるケースも考えられる。

用語定義

概要の最後となるが、仕様書でもっともよく使われる用語の定義をまとめておこう。

● EPUB 出版物 (EPUB Publication)

.epub ファイルに含まれるファイルのうち、書籍のコンテンツを構成する部分をひとまとめに呼ぶ場合の名称。コンテンツを構成するファイルすべてを記述したパッケージドキュメント (.opf ファイル)、目次としてユーザーに提示されるナビゲーションドキュメント (.xhtml ファイル)、およびコンテンツを構成する個々のコンテンツドキュメントで構成される。

● .epub ファイル

EPUB 出版物を EPUB リーディングシステムで読むために ZIP 形式でパッケージングしたファイル。複数の EPUB 出版物を格納することができる。EPUB 出版物そのものに加えて、パッケージングに関する情報も格納されている。



Section 1 書籍設定

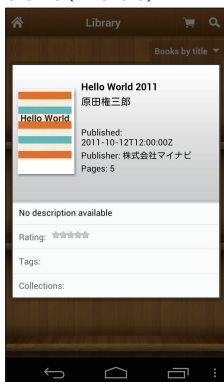
書名を設定する

dc:title

文献を特定する情報として、書名をはじめ、著者名や出版年、巻や号といった情報は書籍は必ず記載されている。EPUBではこれらの情報は、ファイルを束ねるため.opfファイルの中にダブルリンコアの語彙を使って記載する。

サンプル

aldiko (Android)



DL Readers (iPhone)



iBooks (iPhone)



リーディングシステムで書誌情報を表示。iPhoneのiBooksの場合はこれといって書誌情報をまとめて表示する箇所がないが、iPadの場合は一覧表示がある。

.opfファイル

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">①
  <dc:title>Hello World 2011</dc:title>②
  (略)
</metadata>①
```

解説

.opfファイル内のmetadata要素内に<dc:要素名>といったタグを使い、書名をはじめ著者名などが指示できる。指定できる主な内容は49ページに示した。書誌情報としてはせめて、書名、著者名、発行者、発行年月日、ある場合は巻・号の記載はしておくべきだ。なお、記載されたメタデータがリーディングシステム上でどのように表示されるかは、大きく異なる。

- ① metadata要素を宣言
- ② 子要素として<dc:title>で書名を記述する。そのほかのメタ要素を記述する場合は続けて列挙すればよい。

Section 1 書籍設定

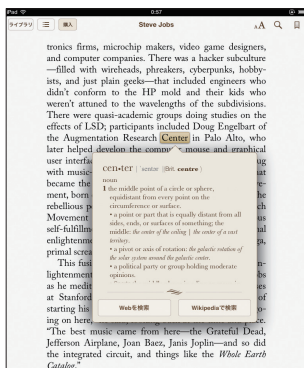
書籍の言語を指定する

dc:language

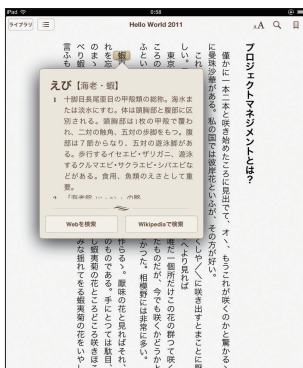
書籍で使われている言語は、.opfファイル内にダブリンコアの語彙を使って記載する。リーディングシステムでは指定された書籍の言語によって、読書中に参照させる内蔵辞書の言語を変更することなどがある。また、XHTMLドキュメントのヘッダにも記載するので合わせて確認しよう。

サンプル

英語



日本語



iBooksでの閲覧例。言語指定が日本語では日本語の辞書、英語では英語の辞書が表示された。しかしOSが日本語で書籍の言語が英語の場合には英和辞書が表示される場合もあり、リーディングデバイスによって対応自体はまちまちである。

.opfファイル

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:language>ja</dc:language>①
  (略)
</metadata>
```

XHTMLドキュメント

```
<html xmlns="http://www.w3.org/1999/xhtml"
  lang="ja" xml:lang="ja">②
```

解説

.opfファイル内のmetadata要素内に<dc:要素名>といったタグを使い、書籍の言語を指定できる。また、コンテンツドキュメントのhtml要素では、xml:lang="ja"とlang="ja"の両方を指定する。

- ①<dc:language>で言語を記述する
- ②html要素に、xml:lang属性、lang属性を指定する

Section 1 書籍設定

本の綴じ方向を変える

page-progression-direction



EPUB 3では、コンテンツの表示方向を右左、左右に指定できるようになった。指定の変更はEPUBファイルを束ねる.opfファイルのspine要素で、page-progression-directionプロパティで変更できる。

サンプル

iBooks



リーディングシステムでltrを指定すれば左から右へとページを進める。

.opfファイル

```
<spine page-progression-direction="ltr"> ~ </spine>
```

解説

.opfファイル内のspine要素にpage-progression-direction属性を設定し、値で"ltr"を指定すれば左から右、"rtl"を指定すれば右から左に進めるようになる。縦組では"rtl"に指定する。

執筆時のiBooks 1.5では、rtlには対応していない。

覚えておきたいHTML & CSS

CSSプロパティ	page-progression-direction	ページを進める方向
値	ltr rtl	初期値。左から右へ進む 右から左へ進む
用法	<spine page-progression-direction="rtl">	

Section 1 書籍設定

カバースタイルと扉ページを指定する

item properties

EPUBを束ねる情報を書き込む .opfファイルでは、リーディングシステムにEPUBファイルを追加すると書庫に表示される本のカバースタイルをitem要素のproperties属性で指定する。また、本を開いた1ページ目はspine要素のitemref子要素を使い一番最初に指定する。

サンプル



カバースタイルに指定した画像が書庫に登録されている例。

.opfファイル

```
<manifest>
  <item id="image01" properties="cover-image"①
    href="the_cover.jpg" media-type="image/jpeg" />
  <item id="cover" properties="cover"② href="cover.
    xhtml" media-type="application/xhtml+xml" /> (略)
</manifest>
<spine>
  <itemref idref="cover" linear="no" />③
  (略)
</spine>
```

- ①カバースタイルにしたい画像のプロパティとしてcover-imageを指定する
- ②扉ページにしたいXHTMLファイルのプロパティとしてcoverを指定する
- ③itemrefの1つ目に最初に表示させたいページのitem idを指定

解説

spine要素内の<itemref>で、冒頭に指定したファイルが扉ページのように、EPUBを開いた際に最初に表示される。なお最初に表示するページのファイル名は必ずしもcover.xhtmlである必要はないし、どのようなコンテンツを配置するかも自由だ。リーディングシステムは、item propertiesで「cover-image」で指定した書庫イメージに使用する。item propertiesの「cover」指定にも役割はあると思われるが、iBooksではカバーとして反映されないようだ。

Section 2 文字表現

文字サイズを
設定する

font-size



文字サイズの指定は、一般的なウェブサイトと同様にfont-sizeという専用のプロパティを使用する。EPUBドキュメントの場合は、リーディングシステムによって解像度が異なるため相対単位を用いてそれぞれの基準サイズを元に描画されるような指定を行った方が良い。

サンプル

見出しレベル1 (h1要素) のテキスト

見出しレベル2 (h2要素) のテキスト

見出しレベル3 (h3要素) のテキスト

見出しレベル4 (h4要素) のテキスト

これは本文段落 (p要素) のテキストです。現状、利用者の多くが視覚的に閲覧できるWebブラウザを使用した場合、掲載された情報をわかりやすく伝えたり、サイトをスムーズに使ってもらうため、視覚的なようなインタラクションが必要になってくるのです。

本文 (body) を100%に、見出し1をその1.6倍、見出し2を1.4倍、見出し3を1.2倍、見出し4を1.1倍に指定した。

HTML

```
<h1>見出しレベル1 (h1要素) のテキスト</h1>  
<h2>見出しレベル2 (h2要素) のテキスト</h2>  
<h3>見出しレベル3 (h3要素) のテキスト</h3>  
<h4>見出しレベル4 (h4要素) のテキスト</h4>  
<p>これは本文段落 (p要素) のテキストです。(略)</p>
```

CSS

```
body { font-size: 100%; }①  
h1 { font-size: 1.6em; }②  
h2 { font-size: 1.4em; }  
h3 { font-size: 1.2em; }  
h4 { font-size: 1.1em; }  
p { font-size: 1em; }③
```

- ①本文の文字サイズの基準は、body要素に指定
- ②文字段落を基準に、サイズを相対指定
- ③本文段落はリーディングシステムの標準サイズで表示

解説

EPUBの文字サイズ指定は、それを閲覧するリーディングシステムの設定によって柔軟に対応できるように「%」や「em」といった相対単位を使って指定する方がよい。絶対単位（ptなど）を指定した場合は、デバイスの解像度によって表示サイズが固定されてしまう可能性があるため推奨されない。

基準となる文字サイズは、body要素のセレクタを作成して「body {font-size: 100%;}」のように指定し、見出しやそれ以外の要素の文字サイズも同様に相対単位を使って指定する方がよい。このサンプルの場合は、標準の文字サイズ（本文の文字サイズ）をデバイスの基準である「100%」とし、それぞれの見出しをレベルに合わせて拡大されるように指定している。「em」は文字の高さを基準とした単位であり、たとえば「1.2em」は「120%」とほぼ同義となる。

EPUBの文字サイズ指定は相対で

EPUBのリーディングシステムは、画面サイズや解像度がさまざまである。スマートフォンサイズの画面があれば、iPadなどのタブレットデバイスサイズもある。その際、絶対単位を使って文字サイズをコントロールするのではなく、そのデバイスの標準サイズを基準に文字サイズをコントロールする方が、読者が感じる書籍の読みやすさに繋がるだろう。ウェブサイトと同じで情報の可読性が大事なのである。

覚えておきたいHTML & CSS

CSSプロパティ	font-size	文字サイズを指定する。
値	実数値	単位を沿えて数値で文字の大きさを指定する。文字サイズはCSSで利用可能な単位で指定可能。ptのようにリーディングシステムの解像度に依存する単位では問題が起こるため、リーディングシステムの設定にあわせて対応できる%やem（文字の大きさ）の相対単位を使用する。
単位	%	パーセンテージ。リーディングシステムの初期値に対する比率で指定
	em	リーディングシステムの初期値における文字サイズのアルファベットの「M」の高さを1とする単位。
	ex	リーディングシステムの初期値における文字サイズのアルファベットの小文字の「x」の高さを1とする単位。
	px	ピクセル
	pt	ポイント（1/72インチ）
用法	セレクタ { font-size: 1.5em; } 文字サイズを指定する要素名に続けて、{ } 内にサイズを指定する。	
記述先ファイル	XHTMLファイル、CSSファイル	

Section 2 文字表現

書体を選択する

font-family



EPUB準拠



要確認

EPUBドキュメントにおける書体の指定は、font-familyプロパティもしくはCSS3から採用されている@font-face規則を利用可能だ。しかし、ドキュメントリーダにはデバイス毎にインストールされている書体の差異があること、仮に書体を読み込む場合は書体のライセンスの問題もあるため注意が必要である。

サンプル

5. 配信フォーマットと配信ルート

5-1. 電子書籍に使われるフォーマット

ここでは、電子書籍に用いられることが多いファイルフォーマットについて解説する。電子書籍に用いられるファイルフォーマットには、文字サイズやデバイスの大きさによって文字組みが変化するリフロータイプと、表示が固定された非リフロータイプがある。また、仕様がオープンで制作ツールが無料で使えるものや仕様がクローズドで使用料がかかるものなどがある。

EPUB 2.01

h1には明朝体を指定し、本文にはゴシック体を指定した。

HTML

```
<h1>5. 配信フォーマットと配信ルート</h1>
```

```
<h2>5-1. 電子書籍に使われるフォーマット</h2>
```

```
<p>ここでは、電子書籍に用いられることが多いファイルフォーマットについて解説する。電子書籍に用いられるファイルフォーマットには、文字サイズや(略) </p>
```

CSS

```
body { font-family: sans-serif; }①
```

```
h1 { font-family: serif; }②
```

① 基準となる書体は、body要素内にfont-familyを使って指定する。

② 要素ごとに、異なる書体を指定すれば、それが適用される。

解説

日本語EPUBドキュメントにおける書体の指定は、現状では障害が多い。リーディングシステムにインストールされている書体（フォント）がそれぞれ異なることが1点目。

EPUBにはフォントの添付が可能だが、フォントを同梱した場合、EPUBを展開すると書体データが抽出できるため、ライセンス上同梱できるものが限られるということが2点目。EPUB3ではフォントの難読化の機能が追加された（27ページ参照）が、リーディングシステムの対応が必要だ。また、ライセンスをクリアしたとしても、欧文と異なり日本語フォントはデータサイズが大きく、仮に使用する文字だけを抽出してサブセット化したとしても、かなり大きなファイルとなるというのが3点目。そのため、現状では@font-face規則を使った直接的な書体の指定よりは、font-familyの全称フォントレベル（明朝体かゴシック体か）での指定が現実的な選択肢だ。

覚えておきたいHTML & CSS

CSSプロパティ	font-family	書体の指定には、font-familyプロパティを使用する。
値	任意の書体名	初期値はリーディングシステムに依存。font-familyを使った書体の指定は、任意の書体名をカンマ区切りで適用したい順に左から列挙し、最後に全称フォント名（serif、またはsans-serif）を記述する。リーディングシステムによってインストールフォントは異なるため、指定したフォントが必ず表示されるわけではない。
	serif	明朝体
	sans-serif	ゴシック体
用法	セレクタ { font-family: sans-serif; }	
	書体を同梱する場合、外部のリソースを参照する場合は、書体名を指定して参照先とフォーマット種別（オプション）を指定する。	
記述先ファイル	XHTMLファイル、CSSファイル	

COLUMN

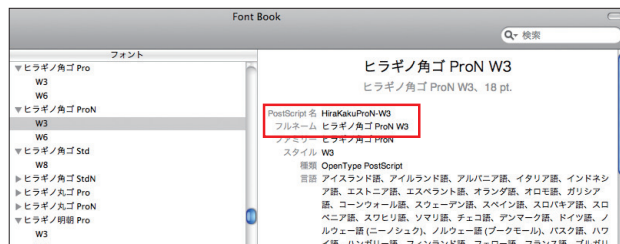
書体指定の重要性

文章の読みやすさを制作サイドでコントロールするためには、書体の指定が欠かせない。誰でも美しい読みやすい書体で読みたいものだ。しかし、日本語フォントはサブセット化したとしてもデータサイズは大きく、結果EPUB自体のファイルサイズに影響を与える。デバイスに依存しない形で最適な読書体験を与えるためには、閲覧デバイスに最適化された書体を利用するのがもっともよい選択肢と言える。

フォント名を調べる

対象リーディングデバイスの仕様や、デザイン上の希望によってはフォント名指定しなければならないこともある。日本語フォントを指定するときは、さまざまデバイスを考慮して日本語名、英語名の両方を指定しておこう。フォント名を調べるとき、手持ちのフォントであれば、MacではFont Book（OSに付属のフォント管理ソフト）を起動し、フォント名を選択して情報を表示する。

Windowsでは、コントロールパネル→フォントでフォントフォルダを開き、フォントをクリックして詳細を確認できる。



Section 2 文字表現

行送りを設定する

line-height



行送りの調整は、CSSに用意されたline-heightプロパティを用いる。CSSにおけるこのプロパティを直訳すれば「行高」となるように、日本語で言うところの行間とは意味が異なる。仮想ボディとその上下に均等につくスペースがこのline-heightということになるので注意しよう。

サンプル

5. 配信フォーマットと

配信ルート

5-1. 電子書籍に使われるフォーマット

ここでは、電子書籍に用いられることが多いファイルフォーマットについて解説する。ファイルフォーマットには、文字サイズやデバイスの大きさによって文字組みと、表示が固定された非リフロータイプがある。また、仕様がオープンで制作ツール仕様がクラウドで使用料がかかるものなどがある。

EPUB 2.01

iBookstoreやSony Reader が採用する開かれた規格

EPUBは、IDPF (International Digital Publishing Forum) が策定を行う電子書籍用アメリカの業界団体として発足したOpen eBook ForumはIDPFの元の組織 (2005年4月で、EPUBの前身となったOEBPS 1.0を1999年に策定している。

サンプル1

h1の見出しサイズを18px、行の高さを36pxに設定。環境によっては、h2より文字サイズが小さくなってしまふなどの不具合が起こる。

5. 配信フォーマットと
配信ルート

5-1. 電子書籍に使われるフォーマット

ここでは、電子書籍に用いられることが多いファイルフォーマットについて解説する。電子書籍に用いられるファイルフォーマットには、文字サイズやデバイスの大きさによって文字組みが変化するリフロータイプと、表示が固定された非リフロータイプがある。また、仕様がオープンで制作ツールが無料で使えるものや仕様がクラウドで使用料がかかるものなどがある。

EPUB 2.01

iBookstoreやSony Reader が採用する開かれた規格

EPUBは、IDPF (International Digital Publishing Forum) が策定を行う電子書籍用フォーマット。1998年、アメリカの業界団体として発足したOpen eBook ForumはIDPFの元の組織 (2005年4月に現在の名称に変更された) で、EPUBの前身となったOEBPS 1.0を1999年に策定している。2010年現在利用されているEPUB 2.0は2006年にできたもので、日本語などに正式対応するバージョンの仕様が固まるのは、2011年以降の予定だ。

IDPFは標準化団体としてEPUBの策定を行っているが、ハードウェア企業、出版社、新聞社、書店、ソフトウェア企業、印刷会社など、さまざまな分野の企業が参加し、アメリカだけでなく、ヨーロッパ、アジアの企業が数多く参加している。

EPUB 2.01の仕様はウェブサイトで見ることができる (英語)。サイトは3つのパートに分かれており、「Open Publications Structure」は製造方法に関する仕様、「Open Packaging Format」はファイル構成やメタデータ

サンプル2

行の高さを相対値で指定。
h1の見出しは行の高さが100%、pは1.45に設定した。

HTML

```
<h1>5. 配信フォーマットと配信ルート</h1>
<h2>5-1. 電子書籍に使われるフォーマット</h2>
<p>ここでは、電子書籍に用いられることが多いファイルフォーマットについて解説する(略) </p>
```

CSS (サンプル1)

```
h1 { line-height: 36px; }①
p { line-height: 1.45 }②
```

CSS (サンプル2)

```
h1 { line-height: 100%; }
p { line-height: 1.45 }③
```

- ①強制的に行の高さを固定したい場合は、pxを使用する
 ②%やemといった相対単位が一般的だが、単位なしでも可
 ③行の高さを相対値にした場合

解説

CSSでの行送り（行間）の設定は、line-heightプロパティを用いる。直訳すれば「行の高さ」という意味になるように、文字の仮想ボディとその上下に均等に付与されるスペースの合計値がこの値になる。つまり、文字のサイズが18pxでline-heightを36pxにすれば、文字の上下に9pxずつの余白が設定されると考えればわかりやすいだろう。

要素ごとにマージンや余白はリーディングシステムによって初期値が異なるので、見出しと本文の間隔調整などでは、line-heightに加え、marginやpaddingプロパティの調整も必要になる。異なる要素間の間隔調整がうまくいかない場合はそこをチェックしたい(120ページ参照)。

覚えておきたいHTML & CSS

CSSプロパティ	line-height	行間を調整する。
値	実数値	単位を沿えて数値で文字の大きさを指定する。行に含まれるテキストの高さを基準として、その上下にスペースを設定する。一般的には、「140%」や「1.4em」など相対的な単位、高さを固定したい場合には「px」を用いるが、複数の文字サイズが混在する場合は単位を入れない数値だけを指定する。
単位	px	ピクセル
	%	パーセント。リーディングシステムの初期値に対する比率で指定。
	em	対象となる文字サイズでのアルファベットの「M」の高さを1とする単位。
	単位無し	対象となる文字サイズの高さを1とする(例: 1=100%、1.4=140%)
用法	セレクタ { line-height: 140%; }	適用する要素名に続けて、行の高さを指定する。仮想ボディの上下に均等についたスペースを含んだ行の高さがこの値になる。
記述先ファイル	XHTMLファイル、CSSファイル	