

EPUBJP

EPUB3における 日本語電子書籍ベーシック基準 EPUB3 Japanese Basic Standard

1.0

EPUB日本語基準研究グループ (EPUBJP)
EPUB Japanese Basic Standard Study Group

社団法人デジタルメディア協会

2012/2/10

株式会社ボージャー

株式会社インフォシテイ

改訂履歴

Ver	改訂日	改訂箇所	改訂内容
Draft 1.0	2011/10/14		新規作成
1.0	2012/2/10	表紙	バージョン情報の明記 (1.0)
1.0	2012/2/10	表紙	主催グループ名の記載 「EPUB日本語基準研究グループ (EPUBJP)」
1.0	2012/2/10	P10	"TOCファイル" "基本構造の項"、headに <style type="text/css"> ol { list-style: none; } </style> を追加
1.0	2012/2/10	P13	"次期バージョンに期待する表現方法"の項、表現を 一部変更
1.0	2012/2/10	P19	"日本語電子書籍のための詳細説明" "TOCファイル" "TOCファイルの構造"の項、外部スタイルへのリンク を削除と、head部に、以下を追加 <style type="text/css"> ol { list-style: none; } </style>
1.0	2012/2/10	P20	"日本語電子書籍のための詳細説明" "TOCファイル" "スタイルについて"の項、説明に以下を追記。 「しかし、Webkitベースの読書システムでの表示を 考慮し、最低限のスタイルとして、head部に、以下 を記述しておく。 <style type="text/css"> ol { list-style: none; } </style> 」
1.0	2012/2/10	P23	縦中横の項、-epub-text-combine-horizontal:allに 関する説明削除
1.0	2012/2/10	P24、P38	画像のページ内フィットについて、Media Queryを 使った書き分けから、max-width、max-heightを 使った方法に変更
1.0	2012/2/10	P28	画像外字のクラス指定について、display:inline;を追加
1.0	2012/2/10	P29	画像外字の場合に、文字装飾が適用されないことの 追記
1.0	2012/2/10	P30	ベンダープリフィクスを使った扉表現の変更
1.0	2012/2/10	P31	非表示の見出しの使用は非推奨に

目次

このドキュメントの目的	6
はじめに.....	6
このドキュメントのスコープ.....	6
EPUB3仕様書について.....	7
EPUB 3 Overview.....	7
EPUB Publications 3.0.....	7
EPUB Content Documents 3.0.....	7
Open Container Format 3.0.....	7
Media Overlays 3.0.....	7
その他.....	7
EPUB 3 Changes from EPUB 2.0.1.....	7
Working Group Charter.....	7
EPUB3で使用できるHTML5について.....	7
その他注意事項.....	8
ファイル名とフォルダ名.....	8
EPUB3コンテンツはどのように作られるのか?	8
フォルダを用意する.....	8
mimetypeファイルを用意する.....	9
META-INF/container.xmlを用意する.....	9
「パッケージ文書」を用意する.....	9
基本構造.....	9
要素解説.....	10
TOCファイルを用意する.....	10
基本構造.....	10
HTML5コンテンツ文書を作成する.....	11
基本構造.....	11
CSSファイルを作成する.....	12
ZIP圧縮し、拡張子を.epubにする.....	12
日本語電子書籍のための詳細説明	12
現状のEPUB3でできない表現について.....	12
リフロー型電子書籍の特性上、別の表現手法をとる必要がある表現.....	12
割注.....	13
ページ内での自由な割り付け.....	13
次期バージョンに期待する表現方法.....	13
扉表現.....	13
クリックابلマップ.....	13
挿絵の配置方法.....	13
漢文の返り点.....	13
Alternate Styleを使わない縦横混在方法.....	13
各種ファイルの制限について.....	13
JPEG.....	14
PNG.....	14
SVG.....	14
縦書き／横書き、縦組／横組について.....	14
縦書き／横書き.....	14
縦組／横組.....	14

「パッケージ文書」	14
ルート要素 package	14
記述例	14
必須属性	14
その他	15
packageの子要素の記述順番	15
書誌情報の記述 metadata	15
記述例	15
必須の子要素	15
その他の情報の記述方法	16
本基準における書誌情報の必須項目	17
構成要素 manifest	17
item要素	17
コンテンツの再生順 spine	18
itemref要素	18
TOCファイル	19
TOCファイルの構造	19
規則	20
スタイルについて	20
nav要素	20
ol要素	20
toc以外のepub:type属性の属性値	20
page-list	20
landmark	20
CSS	21
EPUB3で使用可能なCSS	21
スタイルの書き方	21
Alternate Styleの使用	22
共通スタイルシート	22
書籍ごとの基本スタイルシート	23
ページ余白	24
画像のページ内フィット	24
タイプセクタ	24
クラスセクタ	25
縦組用スタイルシート	25
書体指定	25
余白・空白の指定	25
横組用スタイルシート	26
書体指定	26
余白・空白の指定	26
HTML5コンテンツ文書	27
基本構造	27
文字	27
使用できる文字範囲	27
サロゲートペア	28
外字の使用	28
画像による外字	28
WOFFによる外字	29

スタイルの指定方法.....	29
セマンティックマークアップ	30
epub:typeの記述場所.....	30
代表的なボキャブラリと記述場所.....	30
扉表現.....	30
背景色と文字色.....	31
見出し.....	31
見出しと構造.....	31
クラス指定.....	31
見出しに画像を使う場合のマークアップ.....	31
リンク.....	31
コンテンツ内リンク.....	31
外部リンク.....	32
本文と段落.....	32
インデントと字下げ、リスト	33
字下げ.....	33
リストとインデント.....	34
ルビ.....	34
画像.....	34
インライン画像.....	34
フロート.....	34
キャプション	34
SVG画像.....	34
表組はSVG画像で.....	35
文字装飾.....	35
縦中横.....	35
下付き／上付きは使用方法を限定する.....	35
同一ページでの、縦書きと横書きの混在は避ける.....	36
コミックの場合の記述方法.....	36
「パッケージ文書」	36
基本構造.....	36
dc:type.....	37
spine itemref properties.....	37
CSS.....	37
コミックの基本スタイル定義.....	37
画像ファイル.....	38
HTML5.....	38
コミックの場合の記述方法.....	38
クリッカブルマップについて.....	39
Appendix.....	40
用語の用法について.....	40

このドキュメントの目的

はじめに

日本語の電子書籍をEPUB3で作成するにはどうしたらよいか。

記述する方法は無数にある。

EPUB3では日本語を表現するための機能が足りない？

いやむしろ逆かもしれない。

EPUB3では、まだ勧告に至っていないHTML5やCSS3を使用することができる。またEPUB読書システム（注）依存ではあるが、JavaScriptを使用することもできる。これらを組み合わせれば、不可能なことはないのではないかというくらい、いろいろな表現を記述することができる。

しかし「読書システムに依存する」ということは、作成したEPUB3ファイルの汎用性が損なわれる、ということでもある。

本基準書では、作成したEPUB3コンテンツの汎用性を高めるため、コンテンツの内容以外に記述しなければならない内容をできる限り少なくし、共通の記述で済む箇所は共通にし、その上で、これまでに作成されてきた日本語電子書籍の表現を再現できるような基準となるEPUB3の作り方を示したものである。そのため、EPUB3の仕様としては許されていることでも「～してはならない」と説明している部分がある。

本基準書の目的をまとめると以下ようになる。

- 汎用性が高いデータの作り方の基準を示す
- ソースは複雑にならないよう、記述しなければならない箇所を少なくする
- 日本語電子書籍の表現の再現自体には制限を加えない

（注）EPUB関連のドキュメントでは"ビューア"のことを"Reading System"や"UA(User Agent)"と呼ぶことがある。日本語電子書籍では"ビューア"という用語が一般的だが、本基準書では「EPUB読書システム」と呼ぶことにする。

このドキュメントのスコープ

本基準書は、2011/9/8に公開された「EPUB3 Proposed Specification 6 September 2011」に基づき、日本語電子書籍、特に過去に作成された電子書籍の機能からEPUB3の電子書籍を作成するために必要な内容をまとめたものである。

この「EPUB3 Proposed Specification 6 September 2011」が全く同じ内容で、10/11に「Recommended Specification 11 October 2011」という名称で最終版として公開された。

以下については、本基準書のスコープ外である。

- MathML（数式表現）
- Media Overlays（SMILの使用）
- マルチメディア再生（一部は扱う）
- コンテンツの暗号化

EPUB仕様書 最新版URL：

<http://idpf.org/epub/30>

2011/2/15に公開されたPublic Draftから、2011/5/23のEPUB3 Proposed Specificationの間でも、多くの変更があり、そこから9/8版でも変更があった。9/8版から10/11版には変更はない。

EPUB3仕様書について

前述のEPUB3仕様書（Proposed Specification）には以下のものがある。

EPUB 3 Overview

EPUB 3の概略説明。

EPUB 3とは何か、という説明が記述されている。

最初に読むべき文書であるが、これだけではEPUB 3ファイルを作成することはできないので、以下のドキュメントも適宜参照する必要がある。

EPUB Publications 3.0

主にPackage Document（「rootfile」と呼ばれる場合もあるが、本基準書では「パッケージ文書」と呼ぶ）についての説明。

EPUB Content Documents 3.0

EPUBのコンテンツとして使用できる文書（HTML5, SVG, CSS）についての定義。
ナビゲーション文書（TOCファイル）についても、このドキュメントに記述あり。

Open Container Format 3.0

EPUBを構成するファイル群を1ファイルにパッケージング（ZIP化）する仕様。
その定義ファイル（container.xml等）の記述方法の説明も含む。

Media Overlays 3.0

EPUBにおける、音声同期についての説明。

SMIL(Synchronized Multimedia Integration Language)のサブセットとして定義。

その他

EPUB 3 Changes from EPUB 2.0.1

EPUB 2.0.1との差分

Working Group Charter

EPUB 3に向けての課題（当時はEPUB2.1と呼んでいた）

EPUB3で使用できるHTML5について

EPUB3におけるHTML5とは、原則として、以下のものである（EPUB Content Documents 3.0より）。XHTML5と呼ばれることもある。

本基準書で、以降HTML5と書いた場合には、以下のものを指す。

- 記述方法はHTMLではなくXHTML（XHTML syntax）を使用
- 拡張子は.xhtml
- DOCTYPE宣言は <!DOCTYPE html>
- メディアタイプは、application/xhtml+xml

その他注意事項

ファイル名とフォルダ名

ファイル名やフォルダ名について、特に明示していない場合でも、使用できる文字は、Webで標準的に使用できる文字に準じる。

原則として、以下の範囲に制限する

- 半角英数字
- 半角数字
- ハイフン、アンダースコア（ファイル名の先頭、最後尾につけるのは避ける）

仕様上、大文字小文字が区別されることになっているが、大文字小文字の区別はファイルシステムに依存するので、事故を防ぐためにも、以下のような使い方はしてはならない。

- 同じフォルダ内に、大文字小文字の違いだけのファイルやフォルダを作らない（例 sample.jpg, Sample.jpg）。これはUNIXファイルシステムでは区別されるが多くのクライアントPCでは同一ファイル名として扱われる。
- ファイル名とパス指定で大文字小文字を変えない（例 sample.jpgというファイルを指定するのに、src="Sample.jpg"など）。多くのクライアントPCではこれでも動作するが、UNIXファイルシステムではファイル名が見つけれられない。

EPUB3コンテンツはどのように作られるのか？

ここでは、EPUB3をどのように作るかを一通り解説する。

フォルダを用意する

最初に、以下の構造のフォルダを用意する。

contents -- ルートフォルダ。実際には「contents」という名称ではなく、管理番号等、作成されるコンテンツ固有のフォルダ名。

contents/META-INF -- META-INFというフォルダ名は変更不可。

contents/OEBPS -- EPUBコンテンツを格納するフォルダ。

contents/OEBPS/text -- HTML5ファイルを格納するフォルダ。

contents/OEBPS/styles -- CSSファイルを格納するフォルダ。

contents/OEBPS/images -- 画像ファイルを格納するフォルダ。

「contents」フォルダは自由に命名可能。ただし日本語やスペースは使用してはならない。管理番号等、作成されるコンテンツに合わせて命名する。

実際にEPUB3ファイルにパッケージングする際には、このフォルダの中身だけがアーカイブ化されるため、このフォルダの名称はEPUB3ファイルの中には全く記録されない。そのため、ある程度作業をした後で、フォルダ名を変更することも可能である。

このフォルダのことを、以降「ルートフォルダ」と呼ぶ。

「OEBPS」フォルダおよびサブフォルダの名称は任意であるが、特に変更すべき理由がなければ、この名称を推奨する。

mimetypeファイルを用意する

ルートフォルダに、以下の内容のテキストファイルを「mimetype」という名称で作成する。この名称は変更できない。これは内容、ファイル名とも固定なので、全てのEPUBファイルで共通に使用することができる。

```
application/epub+zip
```

META-INF/container.xmlを用意する

「META-INF」フォルダの中に、以下の内容のテキストを「container.xml」という名称で作成する。この名称は変更できない。

また、このファイルは暗号化してはならない。

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

「xxxxxxx.opf」（「パッケージ文書」）の名称は可変（コンテンツによる）管理を考えた場合には、コンテンツ固有のファイル名にすべきである。

その他、「META-INF」フォルダの中には、以下のようなファイルを置くことができる。

```
signatures.xml
encryption.xml
metadata.xml
rights.xml
manifest.xml
```

「パッケージ文書」を用意する

「パッケージ文書」とは、メタデータや構成ファイルをまとめたXMLファイルである。拡張子は.opf。

「META-INF/container.xml」で指定されたパスに作成する。

基本構造

```
<?xml version="1.0"?>
<package xmlns="http://www.idpf.org/2007/opf" version="3.0" unique-identifier="pub-id" xml:lang="ja">
  <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:title>（タイトル）</dc:title>
    <dc:creator id="creator">（著者名）</dc:creator>
```

```

    <meta refines="#creator" property="role" scheme="marc:relators" id="role">aut</
meta>
    <dc:publisher> (出版社名) </dc:publisher>
    <dc:language>ja</dc:language>
    <dc:identifier id="pub-id">urn:uuid:290ceb3e-c346-4bc6-b41d-000B215E83</
dc:identifier>
    <meta property="dcterms:modified">2011-09-22T08:30:00Z</meta>
</metadata>
<manifest>
    <item id="toc" href="toc.xhtml" properties="nav" media-type="application/xhtml
+xml" />
    <item id="cover" href="images/cover.png" properties="cover-image" media-
type="image/png" />
    <item id="content000" href="text/content000.xhtml" media-type="application/xhtml
+xml" />
    <item id="content001" href="text/content001.xhtml" media-type="application/xhtml
+xml" />
    <item id="style001" href="styles/ebook_style.css" media-type="text/css" />
</manifest>
<spine page-progression-direction="rtl">
    <itemref idref="content000" />
    <itemref idref="content001" />
</spine>
</package>

```

要素解説

metadata

書誌情報を記述する。

manifest

構成ファイルを記述する。

spine

再生する順番を記述する。

TOCファイルを用意する

TOCファイルとは、ナビゲーション用のファイルのこと。

EPUB2ではNCXという形式が使用されていたが、EPUB3ではNCXは廃止となり、HTML5形式のファイルとなっている。

基本構造

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>

```

```

<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://www.idpf.org/
2007/ops" xml:lang="ja">
<head>
<link rel="stylesheet" href="styles/nav_style.css" type="text/css" />
<title> (タイトル) </title>
<style type="text/css">
  ol { list-style: none; }
</style>
</head>
<body>
<nav epub:type="toc">
<h1>目次</h1>
<ol>
<li><a href="text/000.xhtml#mark001">目次項目</a></li>
.....
</ol>
</nav>
</body>
</html>

```

HTML5コンテンツ文書を作成する

コンテンツは、HTML5形式で記述する。

明示的な改ページが必要な場合には、ファイルを分割する必要がある。

構造とスタイルは分離する。HTML5での記述は構造のみとし、スタイルについてはCSSにて定義する。

HTML5コンテンツ文書ファイルは、「OEBPS/text」の中に保存する。

基本構造

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://www.idpf.org/
2007/ops" xml:lang="ja">
<head>
<meta charset="UTF-8" />
<link rel="stylesheet" href="../styles/ebook_common.css" type="text/css" />
<link rel="stylesheet" href="../styles/ebook.css" type="text/css" />
<link rel="stylesheet" href="../styles/ebook_v.css" type="text/css" class="vertical"
title="縦組" />
<link rel="alternate stylesheet" href="../styles/ebook_h.css" type="text/css"
class="horizontal" title="横組" />
<title> (タイトル) </title>
</head>
<body class="top">

```

```
...
</body>
</html>
```

XHTMLに存在しないタグについてはネームスペースを使用する

- prefix namespace URI
- epub <http://www.idpf.org/2007/ops>
- m <http://www.w3.org/1998/Math/MathML>
- pls <http://www.w3.org/2005/01/pronunciation-lexicon>
- ssml <http://www.w3.org/2001/10/synthesis>
- svg <http://www.w3.org/2000/svg>

当面の日本語書籍で必要なのはepubとsvg。それ以外は必要に応じて使用する。

CSSファイルを作成する

HTML5で記述された内容に対して、スタイルをCSSで定義する。

CSSファイルは、"OEBPS/styles"の中に保存する。

EPUB3で使用できるCSSは、CSS 2.1と、以下のCSS 3のモジュール。

- Speech
- Fonts
- Text
- Writing Modes
- Media Queries
- Multi-Column Layout
- Ruby Positioning

これらを使用する際には、-epub-プレフィックスをつける。

(例) -epub-writing-mode

ZIP圧縮し、拡張子を.epubにする

用意したファイルをZIP圧縮する。

この時「mimetype」ファイルが、ZIPファイルの先頭になるようにすること。

そのZIPファイルの拡張子を.epubにすればEPUBファイルが完成する。

日本語電子書籍のための詳細説明

現状のEPUB3でできない表現について

リフロー型電子書籍の特性上、別の表現手法をとる必要がある表現

ウィンドウサイズや文字サイズの変更により、ページ内に表示される文字数・行数、1行の文字数が異なる、というリフロー型電子書籍の特性から、以下については、別の表現手段をとる必要がある（どうしても再現必要な場合にはSVG画像にする）。

割注

割注指定した箇所が行をまたぐ、あるいはページをまたぐ、という表現が、リフロー型の電子書籍では困難である。

そのため、割注については、サイズを縮小して括弧書きで表示したり、注釈用のページを別途用意し、そこへのリンクで表現する等の代替表現が必要である。

ページ内での自由な割り付け

割注だけでなく、ページ内で、位置を特定するような表示方法は原則としてできない（表示が保証されない）。

次期バージョンに期待する表現方法

現時点では困難（表示が保証されない）であっても、表現として重要であり、次期バージョンで期待する表現手法について、ここで挙げておく。

扉表現

独立した扉ページで、ページの中央に配置するための方法は現時点では規定されていない。本ドキュメントでは代替方法を示すが、今後、期待する機能である。

クリッカブルマップ

クリッカブルマップは、画像が実サイズの場合にのみ有効である。%指定をした場合はもちろん、表示されるサイズが小さい場合に、EPUB読書システムが縮小表示を行なった場合や、コミック等ページ内にフィットさせるようにした場合に、ピクセルでサイズ指定した領域と、実際に表示している領域がずれてしまうためである。

これは使用頻度が高いと思われるので、EPUB読書システム側の要件として「画像が実サイズで表示されていない場合のクリッカブルマップの対応」を期待する。

挿絵の配置方法

片方のページ全体が挿絵という場合がある。現状ではfloatを使うことで、テキストを回り込ませることで表現することになるが、テキストのフローとは独立させて、例えば、テキストのリフロー状態とは独立して、必ず章変わりから3ページ目に配置する等の方法があるとライトノベルの挿絵等に有効である。

漢文の返り点

現状では、上付き、下付きについてはそれぞれ指定可能であっても、同じ箇所に同時に、つまり漢文のように、返り点と送り仮名が同時に指定されるようなものの表現はできない。次期バージョンでなくても、将来的に漢文用のスタイルの指定を期待する。

Alternate Styleを使わない縦横混在方法

現状では、論理プロパティが使用できないこと、Windowsでは縦書き時には@つきのフォント名を指定しなければならないことから、縦／横のスタイルシートを作り、Alternate Styleを使わなければならないが、それらの解決を期待したい。

各種ファイルの制限について

ここでは、コンテンツの素材となる各種ファイルについての制限を記載する。EPUB3の仕様上は、必ずしもそういった制限はないが、無制限に巨大な画像ファイルを用意しても再生不可能な場合も出てくるので、ここでは各種ファイルの上限の目安となる数値を規定する。これは今後変更の可能性がある。

JPEG

RGBあるいはグレースケールであること（CMYKは不可）。
プログレッシブ形式は使用しない。
サイズは2048×2048pxを上限とする。

PNG

アニメーションを含まない静止画であること。
サイズは2048×2048pxを上限とする。
JPEGも、PNGも、Photoshop等で作成する場合、「Web用に保存...」から保存する。

SVG

SVGのバージョン1.1を使用する。
サイズは2048×2048pxを上限とする。

縦書き／横書き、縦組／横組について

縦書き／横書き、縦組／横組について、共通に認識できる用語が存在しないため、混乱しがちになるが、本基準書内では、以下のように用語を定義する。

縦書き／横書き

文字の並び順が、上から下（縦書き）か、左から右（横書き）かを表す。
CSSのプロパティでは、-epub-writing-modeに対応する。

縦組／横組

ページの進行方向が、右から左か、左から右かを表す。
通常は、ページの進行方向が右から左の場合には縦書き、左から右の場合には横書きとなる。
こちらは書籍全体の設定になるので「パッケージ文書」のspine page-progression-directionで指定する。
紙の書籍の綴じ方向に対応し、それぞれ、右綴じ、左綴じ、と呼ばれることもある。

「パッケージ文書」

ルート要素 package

記述例

```
<package xmlns="http://www.idpf.org/2007/opf" unique-identifier="pub-id"
version="3.0" xml:lang="ja">
```

「ルート要素」とは、XML文書のツリー構造の最上位の要素。

必須属性

version
EPUB3の場合、属性値は"3.0"

unique-identifier

書籍のユニークIDを定義するためのID。これ自体が書籍のユニークIDではない。ここでは、属性値に"pub-id"等の文字列を指定する。この値は、この「パッケージ文書」の中でユニークである必要があるが、別の書籍で使われていてもかまわない（つまり"pub-id"固定でも問題ない）。実際に書籍のユニークIDは、metadata dc:identifierで指定する。

その他

xml:lang

日本語書籍の場合、属性値は"ja"

*5/23版では必須だったが、9/8版ではオプションになった。

xmlns

ネームスペースの指定。属性値は <http://www.idpf.org/2007/opf>

packageの子要素の記述順番

packageの子要素は、metadata、manifest、spineの順番に書く。

書誌情報の記述 metadata

metadata要素には、主として書誌情報を記述する。

記述例

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:title> (タイトル) </dc:title>
  <dc:creator id="creator"> (著者名) </dc:creator>
  <meta refines="#creator" property="role" scheme="marc:relators" id="role">aut</meta>
</dc:creator>
  <dc:publisher> (出版社名) </dc:publisher>
  <dc:language>ja</dc:language>
  <dc:identifier id="pub-id">urn:uuid:290ceb3e-c346-4bc6-b41d-000B215E83</dc:identifier>
  <meta property="dcterms:modified">2011-09-22T08:30:00Z</meta>
</metadata>
```

metadata要素自体には属性はないが、ネームスペースの指定のため、xmlns:dc="http://purl.org/dc/elements/1.1/"を記述する。

metadata要素の子要素として、各書誌情報を記述する。

必須の子要素

dc:identifier

ユニークIDを指定する。

ユニークIDとしては、UUID、DOI、ISBN、ISSNが使用できる。

学術論文等で、既にDOIが発行されている場合にはDOIを、そうでない場合にはUUIDを推奨する。ISBNについては、meta要素で別途指定することが可能である。

属性idで、packageのunique-identifierで指定した文字列（前述の例では"pub-id"）を指定する（必須）。

UUIDとは「汎用一意識別子」とも呼ばれ、重複しないとされる世界でただひとつのIDのこと。UNIX系のOSであれば、uuidgenというコマンドで生成可能、Windowsであれば、GUID生成用の各ツールあるいはプログラムにて生成、あるいはUUID生成の各種Webサービスを利用することも可能。（記述例の値は例であるので使用しないこと）

urn:uuid:に続けて、生成したUUIDを記述する。DOIを使用する場合には、urn:doi:に続けてDOIの値を記述する。

一度UUIDをつけたら（EPUB3ファイルとして流通させた後は）極力変更しないようにする。改訂版等の場合には、IDは変更せずに、以下のように修正日をUTCで記載する。

```
<meta property="dcterms:modified">2011-01-01T12:00:00Z</meta>
```

dc:title

タイトルを指定する。

dc:language

言語を指定する。

日本語書籍の場合にはja。

属性値ではなく、子要素への指定なので、

```
<dc:language>ja</dc:language>
```

のように記述する。

dcterms:modified

更新日を指定する。9/8版より必須の子要素になった。

その他の情報の記述方法

meta要素と組み合わせて記述する。

著者名の記述例

```
<dc:creator id="creator">夏目漱石</dc:creator>
```

```
<meta refines="#creator" property="role" scheme="marc:relators" id="role">aut</meta>
```

```
<meta refines="#creator" property="alternate-script" xml:lang="en">Soseki Natsume</meta>
```

「著者（author）」を直接記述する箇所はないので、まずcreatorとして記述し、そのroleが著者（aut）であることを指定する。

出版社名の記述例

```
<dc:publisher>（出版社名）</dc:publisher>
```

出版物の種類については、以下のようにdc:typeを指定できる。

```
<dc:type>comic</dc:type>
```

指定できる子要素は、任意の文字列だが、以下については、本基準書の固定値である。

comic：コミック

dictionary：辞書

サブタイトルのように、複数のタイトルの表示順を明示するにはdisplay-seqを使用する。

ソート用の読み

file-as

シリーズ

group-position

本基準における書誌情報の必須項目

EPUB3として、必ず記述しなければならない書誌情報は、dc:identifier、dc:title、dc:language、dcterms:modified の4つであるが、以下についても記述することを強く推奨する。

- タイトルの読み
- タイトルのソート用の読み
- シリーズの場合には、シリーズ内での順番

構成要素 manifest

manifest要素には、コンテンツを構成するファイルを記述する。

manifest要素は属性を必要としない。

manifestの子要素に、item要素を記述し、そこにコンテンツを構成するファイルを記述する。item要素は、構成ファイルの数だけ書くことができる。

item要素

記述例

```
<item id="toc" href="toc.xhtml" properties="nav" media-type="application/xhtml+xml" />
```

```
<item id="content000" href="text/content000.xhtml" media-type="application/xhtml+xml" />
```

以下の属性を記述する。

id：必須属性

そのファイルのID。spine itemrefから参照される。当該の「パッケージ文書」の中でユニークであれば、命名方法は任意であるが、参照を簡易にするため、ファイル名から拡張子を除いたもの（"content000.xhtml"であれば"content000"）にしておくことを推奨する。

href：必須属性

「パッケージ文書」からの相対パスで、構成ファイルを指定する。

media-type：必須属性

当該ファイルのメディアタイプを指定する。

代表的なメディアタイプは以下の通り。

application/xhtml+xml：

XHTMLファイル (EPUB3ドキュメントにおけるHTML5ファイル)

image/jpeg：JPEGファイル

image/png：PNGファイル

text/css：CSSファイル

image/svg+xml：SVGファイル

properties：オプション属性

そのファイルがコンテンツ内で意味を持つ場合に記述することができる。

属性値の例

cover-image：表紙画像

nav：ナビゲーション用ファイル

svg：XHTMLファイル内でSVGを使用している際には記述すること。

コンテンツの再生順 spine

manifestで定義されたファイルを再生する順番を記述する。

spine要素の属性

page-progression-direction：

コンテンツとしてのページめくり方向を指定する。

属性値は以下の通り。

ltr：左から右（横組、左綴じ）。通常、横書きの本の場合に指定（left-to-right）

rtl：右から左（縦組、右綴じ）。通常、縦書きの本の場合に指定（right-to-left）

default：EPUB読書システムに依存する。

通常は"default"は使用せず、横書きであれば、"ltr"、縦書きであれば"rtl"を指定する。

spineは子要素にitemref要素を記述する。

itemref要素

記述例

```
<itemref idref="content000" />
```

```
<itemref idref="content001" />
```

記述する順番が重要である。この場合、content000、content001の順番に再生される。

itemref要素は、以下の属性を記述する。

idref：必須属性

manifest itemで定義したitemのidを指定する。

linear：オプション属性

通常は省略する。

注釈のように、本文のフローからは独立し、参照されるようなファイルの場合、属性値として"no"を記述する。

properties：オプション属性

コミックやグラビア等で、見開き表示になった場合に、どちらのページに表示するかを指定する。

属性値は、以下のいずれか。

"page-spread-left"：見開き時に左ページに表示

"page-spread-right"：見開き時に右ページに表示

TOCファイル

ナビゲーション文書。

ファイル名は任意に設定可能だが、toc.xhtml等、TOCであることが自明であるような命名にすべきである。

EPUB2ではTOCファイルは、ネームスペースが <http://www.daisy.org/z3986/2005/ncx/> の「NCX文書」（media-type="application/x-dtbncx+xml"）だったが、EPUB3でのTOCファイルは、HTML5のnav要素を使用したXHTML文書（media-type="application/xhtml+xml"）である。つまり、NCXのように特殊な作りではなく、コンテンツと同様のHTML5文書として作成する。

コンテンツのHTML5との違いは以下。

- ・「パッケージ文書」でのmanifest要素の子要素のitemの属性propertiesで"nav"を指定。
- ・目次項目をnav要素で囲み、epub:type属性で、属性を定義。

TOCファイルの構造

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:epub="http://www.idpf.org/2007/ops"
  xml:lang="ja">
<head>
<title> (タイトル) </title>
<style type="text/css">
  ol { list-style: none; }
</style>
</head>
<body>
<nav epub:type="toc">
```

```
<h1>目次</h1>
<ol>
  <li><a href="text/000.xhtml">目次項目</a></li>
  .....
</ol>
</nav>
</body>
</html>
```

規則

- bodyの直接の子要素にできるのはnav要素のみ
- nav要素の直下には見出し要素 (h1~h6)
- navの直接の子要素にできるのは、見出し要素 (h1~h6) とol要素のみ
- a要素、span要素はol liの子要素とする。
- a要素、span要素は、空のテキストであってはならない。

スタイルについて

EPUB読書システムの目次機能等で利用されることを考えると、スタイルを指定する必要はないと思われる。

EPUB読書システムでは、ol li要素について、数字を表示してはならない (list-style:none として処理する)

しかし、Webkitベースの読書システムでの表示を考慮し、最低限のスタイルとして、head部に、以下を記述しておく。

```
<style type="text/css">
  ol { list-style: none; }
</style>
```

nav要素

bodyの子要素。

属性としてepub:typeをとる。

属性値として、最低限必要なのは、"toc"。それ以外の属性値 (page-list、landmark) は必須ではない。

ol要素

navの子要素

ol liの順番は、「パッケージ文書」のspineで指定された順番に従う必要がある。

toc以外のepub:type属性の属性値

page-list

```
<nav epub:type="page-list">
```

ページネーションの順番を示す。

hidden属性を使うことができる。

landmark

```
<nav epub:type="landmark">
```

「パッケージ文書」のguide要素（EPUB3では非推奨となったもの）に対応する。

CSS

HTML/CSSの説明では、構造を表すHTMLの説明を先に、スタイルを表すCSSの説明を後にするのが通例だが、日本語書籍の場合には、縦書き、縦中横、圏点といったスタイルが重要であるため、スタイルの説明を先に行なう。

EPUB3で使用可能なCSS

繰り返しになるが、EPUB3で使用可能なCSSは、CSS 2.1と、CSS 3の以下のモジュール。

- Speech
- Fonts
- Text
- Writing Modes
- Media Queries
- Multi-Column Layout
- Ruby Positioning

CSS 3はまだ勧告にいたっていない、Working Draftのものである。

これらを使用する際には、-epub-プレフィックスをつける。

(例) -epub-writing-mode

このドキュメントでは、EPUBの仕様に合わせ、-epub-プレフィックスをつけるべきところにはつけている。しかし、現行のGoogle ChromeあるいはWebKit Nightly Buildは、-epub-プレフィックスには対応していないので、作成したファイルをそれらのブラウザで確認したい場合には、-epub-の代わりに-webkit-をつける必要がある。

(例)

```
-epub-text-combine:horizontal;  
-webkit-text-combine:horizontal;
```

スタイルの書き方

CSSの記述方法には複数のパターンがあるが、この基準書では記述方法は限定する。

- 書籍の全体としての組方向（ページめくり方向）は「パッケージ文書」のspine要素のpage-progression-directionで指定する。
- どのような書籍でも汎用的に使用できる共通スタイルシートを用意し、それを最初にリンクする。
- 書籍ごとの基本スタイルシートを用意する。
- EPUB読書システムで、縦／横を切替えられた場合のことを考慮し、縦組用、横組用のスタイルシートを用意する。
- HTML5コンテンツ文書のhead要素内に記述するスタイルは最小限にとどめる。
- HTML5コンテンツ文書の本文のタグに直接スタイルを記述することは行なわない。
- IDではなくクラスで定義する。
- 状況依存セレクタ（例 `h1 em { color:red;}`、`h1 > em { color:red;}` 等）は使用しない。
- クラスの重複指定（例 `太字で赤字`）は可とする。

Alternate Styleの使用方法

本来であれば、縦組時と横組時で、同一のスタイルシートにしたほうが簡潔になるが、現時点では、margin、paddingの方向、Windowsの場合の書体指定については書き分けざるを得ない。（今後、論理プロパティが採用され、Windowsで縦書きの場合にも@つきのフォント名の指定の必要がなくなれば、同一スタイルシートでの指定が可能になる）そこで、原則として、セレクトへのスタイル指定、クラス定義は基本スタイルシートで行い、margin、padding、および書体指定を行なっているセレクト、クラスに関しては、縦組用、横組用で同一名称のセレクト、クラスを定義し、そこで上書きする。

それぞれのスタイルシートのファイル名を、

共通スタイルシート：ebook_common.css

基本スタイルシート：contents_style.css

縦組用スタイルシート：contents_style_v.css

横組用スタイルシート：contents_style_h.css

とした場合、基本が縦書きの書籍であれば、HTML5のhead要素内で、以下のように、リンクする。

```
<link rel="stylesheet" href="../styles/ebook_common.css" type="text/css" />
```

```
<link rel="stylesheet" href="../styles/contents_style.css" type="text/css" />
```

```
<link rel="stylesheet" href="../styles/contents_style_v.css" type="text/css"
```

```
class="vertical" title="縦組"/>
```

```
<link rel="alternate stylesheet" href="../styles/contents_style_h.css" type="text/css"
```

```
class="horizontal" title="横組"/>
```

共通スタイルシート

汎用的に使用できるクラス集。

構造とスタイルを分離すると言っても、どのような書籍であっても共通のスタイルは存在する。

以下のようなものは、どの出版社のどのタイトルでも共通に使用できるものである。

```
/*行揃え*/
```

```
.left {  
  text-align:left;
```

```
}
```

```
.center {  
  text-align:center;
```

```
}
```

```
.right {  
  text-align:right;
```

```
}
```

```
/*画像の左寄せ・右寄せ*/
```

```
img.left {  
  float:left;
```

```
}
```

```

img.left {
  float:right;
}

/*色指定 (X11 Colorsを16進で表記) */
.blue {
  color:#0000ff;
}
.green {
  color:#008000;
}
.red {
  color:#ff0000;
}

/*縦中横*/
.tcy {
  -epub-text-combine:horizontal;
  -webkit-text-combine:horizontal;
}

/*傍点*/
.em_accent {
  -epub-text-emphasis-style: filled sesame;
}
.em_dot {
  -epub-text-emphasis-style: filled double-circle;
}
.em_circle {
  -epub-text-emphasis-style: open dot;
}
.em_disc {
  -epub-text-emphasis-style: filled dot;
}

```

書籍ごとの基本スタイルシート

Windowsでの縦書き時のフォント名には@をつける必要があるため、font-familyは縦組用／横組用のスタイルシートで指定する。このCSSファイルでは欧文フォントを除き、font-familyの指定は行なわない。

原則として、-epub-writing-modeは、縦組用／横組用のスタイルシートで指定する。

縦組/横組の設定に関わらず、縦書きあるいは横書き固定にしたい場合に、こちらのクラスセレクタに記述する。

インデントは縦書き/横書きで指定方法が変わるので、原則として組用/横組用のスタイルシートで指定する。

ページ余白

```
@page {  
  margin:8%;  
}
```

ページ割り付けされた場合 (paged media) の余白を指定する。

margin-top, margin-bottom, margin-left, margin-rightを個別指定してもよい。

デバイスサイズが多岐にわたることを考慮すると、pxでの指定は推奨しない。

電子書籍としては基本的にページ割り付けされることを考慮すると、@pageで行なうべきであるが、ターゲットがcontinuousメディア (スクロールを前提とした表示) なら、bodyにmarginを指定する。

画像のページ内フィット

```
img.pagefit {  
  display:block;  
  margin: auto;  
  height: auto;  
  width: auto;  
  max-width:100%;  
  max-height:100%;  
}
```

タイプセレクト

書籍全体の基本となるスタイルは、タイプセレクトで指定する。ここで指定したスタイルは、必要に応じて、クラスセレクトで上書きされる。

```
body {  
  font-size: 20px;  
  line-height: 175%;  
  letter-spacing: 0px;  
  text-align: justify;  
}
```

それぞれの値は変更可能。

フォントサイズは、全体の基本となるので、ここでのみ、px指定する。

このフォントサイズを基準として、その他のフォントサイズ、行間等は%指定で行なう。

行揃え (text-align) は、詩集などをのぞき、leftあるいはjustifyを指定するが、leftよりもjustifyのほうが、多くの場合にいい結果が得られる。

```
h1,h2,h3,h4,h5,h6 {  
  font-size: 110%;  
  font-weight: bold;  
}
```

基本となる見出しのスタイル指定を行なう。

多くの場合、見出しはそれぞれのクラスセレクトアで上書きされる。

```
a {
  text-decoration: none;
}
リンクに下線をつけたく無い場合の設定
a:link {
  color:#00008b;
}
a:visited {
  color:#00008b;
}
リンク色の指定。
```

クラスセレクトア

タイプセレクトアでは、全体の基本となるスタイルを指定する。

クラスセレクトアでは、ページに対して全体のスタイルとは異なるスタイルを適用させたい場合の指定、タイプ別の見出しの表示設定、行揃えの指定、文字装飾など、クラスで設定する。

原則として、縦書き／横書き (-epub-writing-mode) 指定、書体 (font-face) の指定、余白・空白 (padding-*, margin-*) は、ここでは行なわず、縦組用スタイルシート、横組用スタイルシートで行なう。Reading Systemの縦組／横組の設定に関わらず、縦書き／横書きを固定する場合（表紙、奥付等）には、縦書き／横書き指定、書体指定、余白・空白も、ここで記述する。

縦組用スタイルシート

原則として「書籍ごとの基本スタイルシート」と同じタイプセレクトア、クラスセレクトアを用意する。そこで、縦書き／横書き (-epub-writing-mode) 指定、書体 (font-face) の指定、余白・空白 (padding*, margin*) の指定を行なう。

書体指定

書体指定は、Windowsの場合を考慮し、書体名の前に、@をつけること。

以下の順番で記述することを推奨する。

明朝体の場合：

```
font-family:"@Hiragino Mincho ProN","@IPAexMincho","@M S 明朝","Hiragino Mincho ProN","IPAexMincho","serif";
```

ゴシック体（強調）の場合：

```
font-family:"@HiraKakuProN-W6","@M S ゴシック","HiraKakuProN-W6","sans-serif";
```

ゴシック体（通常）の場合：

```
font-family:"@HiraKakuProN-W3","@M S ゴシック","HiraKakuProN-W3","sans-serif";
```

余白・空白の指定

margin-*, padding-*の指定の場合には、上下左右、物理方向の指定になる。

縦書き時に、padding-topで指定した場合には、テキストブロック全体の字下げとなる。縦組用スタイルシートで、padding-topで指定したクラスは、横組用スタイルシートでは、padding-leftと書き換える必要がある。

インデント（1行目の文字の開始位置）の指定は、箇条書きのような突き出しインデント（text-indentにマイナスの値を指定）を指定する場合には、padding-topも同時に指定する必要がある。

例えば、1文字の突き出しインデントを表現するクラスは以下のようになる。

```
.indent {  
  padding-top: 1em;  
  text-indent: -1em;  
}
```

通常、字下げあるいはインデントのブロックは、前後に1行空きをともなう場合が多いので、その場合には、以下のようにする。

```
.indent {  
  margin-right: 1em;  
  margin-left: 1em;  
  padding-top: 1em;  
  text-indent: -1em;  
}
```

横組用スタイルシート

原則として「書籍ごとの基本スタイルシート」と同じタイプセクタ、クラスセクタを用意する。そこで、縦書き／横書き（-epub-writing-mode）指定、書体（font-face）の指定、余白・空白（padding*、margin*）の指定を行なう。

書体指定

以下の順番で記述することを推奨する。

明朝体の場合：

```
font-family:"Hiragino Mincho ProN","IPAexMincho","M S 明朝","serif";
```

ゴシック体（強調）の場合：

```
font-family:"HiraKakuProN-W6","M S ゴシック","sans-serif";
```

ゴシック体（通常）の場合：

```
font-family:"HiraKakuProN-W3","M S ゴシック","sans-serif";
```

余白・空白の指定

margin-*、padding-*の指定の場合には、上下左右、物理方向の指定になる。

横書き時に、padding-leftで指定した場合には、テキストブロック全体の字下げとなる。縦組用スタイルシートで、padding-topで指定したクラスは、横組用スタイルシートでは、padding-leftと書き換える必要がある。

インデント（1行目の文字の開始位置）の指定は、箇条書きのような突き出しインデント（text-indentにマイナスの値を指定）を指定する場合には、padding-leftも同時に指定する必要がある。

例えば、1文字の突き出しインデントを表現するクラスは以下のようになる。

```
.indent {
```

```
padding-left:1em;
text-indent:-1em;
}
```

通常、字下げあるいはインデントのブロックは、前後に1行空きをとる場合が多いので、その場合には、以下のようにする。

```
.indent {
margin-top:1em;
margin-bottom:1em;
padding-left:1em;
text-indent:-1em;
}
```

HTML5コンテンツ文書

基本構造

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://www.idpf.org/
2007/ops" xml:lang="ja">
<head>
  <meta charset="UTF-8" />
  <link rel="stylesheet" href="../styles/ebook_common.css" type="text/css" />
  <link rel="stylesheet" href="../styles/ebook.css" type="text/css" />
  <link rel="stylesheet" href="../styles/ebook_v.css" type="text/css" class="vertical"
title="縦組" />
  <link rel="alternate stylesheet" href="../styles/ebook_h.css" type="text/css"
class="horizontal" title="横組" />
  <title> (タイトル) </title>
</head>
<body class="top">
...
</body>
</html>
```

文字

使用できる文字範囲

EPUB3では、文字コードにUTF-8を使用することができる。そのため、JIS X 0208（一般的に「シフトJIS」あるいは「JIS第一、第二水準」と呼称されるもの）外の文字も記述自体は可能だが、それを表示するためには、対応したフォントが必要である。Mac OS X 10.5～、Windows Vista～、あるいはiPadであれば、AdobeJapan1-6の相当の文字が使用できるが、それ以外のプラットフォームの場合には、文字範囲は限定される（iPhoneの場合にも、AdobeJapan1-6相当ではあるがゴシック体のみ）。

しかし、JIS X 0208の範囲では、多くの文字を外字として扱わなければならないため、ここでの基準としては、フリーで利用可能なIPAex明朝の規格である「JIS X 0213:2004」を使用可能な文字範囲とする。

サロゲートペア

ただし、JIS X 0213:2004の文字の中でも、サロゲートペアを使った文字については、同じEPUB読書システムであってもプラットフォームによっては表示されない可能性がある。例えば「叱」という文字（口篇に七）はUnicodeでは20B9F(D842+DF9F)であるが、これは当該フォントにグリフがあっても、表示されない場合があるので、現時点では使用を避けた方がよい。

外字の使用

EPUB3で、外字を使用するには、

- PNG画像を使用する
- SVG画像を使用する
- SVGフォントを使用する
- WOFFを使用する

という方法がある。

SVGフォントは、運用コストが低いとも言われているが、EPUB読書システムでの必須要件になっていないこともあるので、現時点では使用を避ける。

理想的な方法はWOFFであるが、現実的なプロダクションを考えると、PNG画像で外字を作成するのが、現時点では最も汎用性が高い（ただし本文に使用されるフォント等との関係で、外字用のクラス：`img.gaiji{width:1em;height:1em;display:inline;}`で、外字画像のサイズを指定しても、若干サイズが異なってしまう可能性がある）。

通常の外字はPNG画像（あるいはSVG画像）を作成することで対応し、文字としての属性が必要な場合にはWOFFを使用する。

ただしWOFFにした場合に、Windowsで縦書き時の表示に問題がでる可能性がある。

画像による外字

PNG画像で外字を作る場合には、画像サイズは、64×64～128×128程度のサイズの正方形で、白黒2値の透過PNGか、アンチエイリアスつきの32bitのPNG。

ファイル名は、Unicode値がある場合には、“u4e04.png”のようにuに続けてコードを書く。Unicode値が無い場合でも、何らかのコードがふられている場合には、そのコードを示す接頭辞とコードによって表現する。独自の記号のようにコードが存在しない場合には、わかりやすい名称で命名する。EPUB読書システムが、このファイル名を解釈するわけではなく、命名規則を明確にしておくことで、同一の外字画像を複数作ったり、逆に別の外字であるのに、ファイル名が同じになるようなことを防ぐことが目的である。

SVG画像を外字画像として使用することもできる。SVGフォントとの違いは文字属性の有無。SVGフォントであれば、文字として色を変更したり、文字選択も可能である（が表示するグリフが無い文字が選択できる意味があるかどうかは検討する必要がある）。

使用する際には、

- CSSでは、共通スタイルシートに記述されている、

```
img.gaiji {  
  width:1em;  
  height:1em;
```

```
display:inline;
}
```

を利用する。

- 「パッケージ文書」で、その画像をmanifestに追加する。
- HTML5コンテンツ文書では、

```

```

のように、classとして"gaiji"を指定し、altに、読み、あるいは代替文字を記述する。可能ならコメントで、どんな文字であるか追記しておく。

画像による外字の場合、サイズ以外の文字装飾は適用されない。

WOFFによる外字

外字の使用に際して、文字としての属性が重要な場合にはWOFFを使用することもできる。欧文のドキュメントの場合には、アルファベットのフォントサイズも小さいので、必要なフォントをEPUBドキュメントに含めることも可能だが、日本語フォントの場合、JIS X 0213に対応したフォントの場合、10MB前後のサイズになるため、EPUBのドキュメントに含めるのは現実的ではない。原則としてWOFFの使用は外字等に限定すべき。

WOFFの作成方法は、本ドキュメントでは割愛する。

作成されたWOFF（ここでは、"HanaMinMinus.woff"というフォント名と仮定する）を使用する場合には、以下のようにする。

- 作成したWOFFフォントファイルは、"/OEBPS/font"に置く。
- 「パッケージ文書」でmanifestに追加する。media-typeは、"application/font-woff"。
- 書籍ごとの基本スタイルシートに以下のように定義する。

```
@font-face {
  font-family: "myFont";
  src: url("HanaMinMinus.woff") format('woff');
}
.myFontClass {
  font-family: "myFont";
}
```

- HTML5コンテンツ文書では、以下のようにマークアップする。

```
<span class="myFontClass">■</span>
```

（■の位置には、WOFFで定義している文字）

ただし、WOFFでは、横書き用のフォント+縦書き用のフォント（先頭に@がついたフォント）を作成できないため、Windowsの場合に、縦書きでは使えないという問題が発生する。これがWindowsのWebkitのバグなのか、仕様なのか、その他のWindowsのEPUB読書システムでは、縦書き時に縦書き用のフォントを指定しなくてもいいのか、現時点では判断できないため、WOFFの使用は当面避けた方がいいのではないかとと思われる。

スタイルの指定方法

スタイルシートの構成とHTML5コンテンツファイルからのリンク方法は「[Alternate Styleの使用方法](#)」を参照。

セマンティックマークアップ

EPUB3では、HTML5を採用していることに加え、専用属性であるepub:typeを用意しているので、セマンティックなマークアップが可能である。アクセシビリティを考慮すると、可能な限りセマンティックマークアップを行なうべきではあるが、現状では任意とする。

セマンティック用のボキャブラリおよび記述場所については、以下のURLを参照

<http://idpf.org/epub/vocab/structure/>

ここでは、特に日本語書籍で使用頻度が多そうなものに限定して説明する。

epub:typeの記述場所

epub:type属性は、すべてのタグに記述できるわけではない。

使用できるタグは、section, body, aside, h1-h6, hgroup, aなどだが、使用できるボキャブラリの種類によって、記述できる箇所が異なる。

代表的なボキャブラリと記述場所

カバーページではbodyタグにepub:type="cover"

メインタイトルページではbodyタグにepub:type="volume"

章扉のページではbodyタグにepub:type="chapter"

TOCファイルとは別に、本文中に目次ページを含む場合、目次がある箇所をsectionで囲み、<section epub:type="toc">

奥付は、奥付の内容をsectionで囲み<section epub:type="colophon">

扉表現

縦書き日本語書籍の場合、扉ページではページ中央に配置することが多いが、EPUBの場合には、現時点では表現する方法がない。テーブルレイアウト（tableタグではなくスタイルのtable-cell）を使う方法もあるが、EPUB読書システムも対応しているとは限らず、またスタイルを使っているといっても、テーブルレイアウトをしたのでは、構造とスタイルの分離の意味がなくなってしまう。

ここでは、中央に揃えることはできないという前提で、前に空きを入れて調整する。

将来的にEPUB/CSSが扉表現に対応した際に、再変換を容易にするためコメント等を入れておいた方がよい。

この2つを以下のように記述する（数値は10%～30%の間）

```
<style type="text/css">
body {
  margin-right: 30%;
  /* vertical-align:middle:*/
}
@page {
  margin-right: 30%;
  /* vertical-align:middle:*/
}
</style>
```

背景色と文字色

背景色と文字色の指定について、EPUB読書システムのデフォルトに任せる場合には、指定を行わないようにする。(通常は白地に黒)
もちろん意図的に指定することは問題ない。

見出し

見出しと構造

章、節、項といった、文書の見出しは、構造として、見出しのレベルに対応させる。章をh1にするかh2にするかは、版元のポリシー次第であるが、以下については遵守しなければならない。

- 章、節、項それぞれ、同じ見出しレベルにしなければならない。たとえば、第一章を<h1>第一章</h1>とh1でマークアップしたのであれば、第二章もh1でマークアップしなければならない。
- 章がh1であるならば、節はh2でなければならない。また章をh1、節をh2としたのであれば、項はh3にしなければならない。(章をh2にしたのであれば、節はh3、項はh4)
- 本来、HTML5文書の場合には、h1、h2、h3、...と見出しが階層的になっているべきであるが、EPUBの場合には、明示的な改ページはHTML5文書を分けて表現するため、h2のみ、あるいはh3のみ、という文書が存在してしまうが、これは仕様上やむをえない。

クラス指定

見出しの字下げや前後の空き行の調整はスタイルシートでクラスを指定することで行なう。

見出しに画像を使う場合のマークアップ

見出しの部分に、文字列ではなく画像を使う場合がある。

その場合でも、文書の構造として、見出しを入れる。

複数の実現方法があるが、ここでは、見出しの中に画像を入れることを基準とする。

(例)

```
<h2></h2>
```

中扉のページに画像を使用するような場合には、次のように非表示の見出しを使用することも可能だが、スタイルを無視する読書システムでは表示される可能性もあるため推奨しない。

display:noneを使った場合、そこにidを指定し、ジャンプ先にすることは推奨しない。

(例)

```
<h2 style="display:none">第二章</h2>
```

```

```

リンク

コンテンツ内リンク

リンクはa hrefにて指定する。

リンク先は、idで指定された箇所。

idは、h1-h6、div、p、span、aに指定できる。

(のようなspanの空タグは推奨しない)

a hrefでは、idに対して#をつけて識別する。

idは、display:noneで非表示にした要素に対して指定することは推奨しない。
EPUBの場合には、改ページ単位でHTML5ファイルが分割されている場合が多い。そのため、通常は、ファイル名と合わせて指定する。

(例)

```
<a href="contents_0001.xhtml#mark01">リンク</a>
```

クリックブルマップを使ったリンクについては、そのマップを使用する画像が等倍表示されている場合でなければ使用できない。つまり、小はスマートフォンから、大はPCまで、ウィンドウサイズに応じて画像の表示サイズが変わるような場合には、マップで指定した座標と、そのマップを使用している画像の位置がずれてしまうため、使用することは困難。

外部リンク

外部リンクもa hrefにて指定する。

使用可能なプロトコル（スキーマ）（http://、https://、mailto://、tel://等）はEPUB読書システムに依存する。

本文と段落

英語でいうparagraphと日本語の段落は、必ずしも同一のものではない。

欧文の場合には、行頭の文字下げに、複数の半角スペースを使用した場合には、どの程度の字下げ量になるかはReading System次第だが、日本語で、全角スペースを使用した場合には、厳密に1文字分の字下げとなる。

また、段落間の間隔について、日本語では、他の行間と同じ幅である必要があるが、欧文の場合には、やや広い場合がある。

そのため、日本語書籍にて、改行を全てp要素にしてしまうと、段落間の間隔が、他の行間と異なって表示される場合がありうる（現状のWebkitの場合、ルビがある場合に、段落間の間隔が開いてしまう）。

基本準では、段落については、以下のように扱う。

- bodyの直接の子要素になれるのはブロック要素のみ。
- 見出し等のブロック要素以外は、pあるいはdivで囲む。
- 段落の表示方法が変わらない場合には、
による改行と、全角スペースによる行頭字下げを行なう。
- 時間経過、場面転換などを表現するために、同一の本文スタイルで1行あける場合には、連続した
で行なう（連続した
は否定的に言われることもあるが、この用途の場合には、意味合いとしては正しいのではないか）。
- 引用等で、まとまったブロックが字下げ、インデントされる場合には、別のブロックとして扱う。そのブロックはdivで囲み、前後の空き行の量は、スタイルで指定する（行揃えの変更、ブロック全体の字下げ、突き出しインデント、floatを用いた画像などがある場合）。なお、その場合の空き行の量は、通常の日本語組版では、行単位の空きになるので、整数値（単位はem）で行なう。

記述例：

スタイルの指定（縦書き用）

```
.start {  
    margin-right:1em; /*ブロックの前に1行の空き*/
```

```

margin-left:1em; /*ブロックの後に1行の空き*/
padding-top:2em; /*2文字の字下げ*/
}
.indent {
margin-right:1em; /*ブロックの前に1行の空き*/
margin-left:1em; /*ブロックの後に1行の空き*/
padding-top:1em; /*1文字の字下げ*/
text-indent:-1em; /*1文字の突き出しインデント*/
}

```

本文指定

```

<p>
通常本文<br />
通常本文の続き<br />
<br />
場面転換などのために1行空けて<br />
通常本文の続き
</p>
<div class="start">
  <p>
引用ブロック。<br />
引用ブロックの続き。
  </p>
</div>
<p>
通常本文<br />
通常本文の続き
</p>
<div class="indent">
  <p>
箇条書きその1。
  </p>
  <p>
箇条書きその2。
  </p>
  <p>
箇条書きその3。
  </p>
</div>

```

インデントと字下げ、リスト

字下げ

特定のテキストブロックを、他のテキストよりも下げて（縦書きの場合には、下に、横書きの場合には右に）表示することを字下げという。インデントと称される場合もあるが、

ここでは、インデントは「1行目の文字の表示位置の上げ下げ」として、字下げと区別する。

リストとインデント

箇条書きについて、横書き書籍の場合には、リスト (ol, ul) ので表現可能な場合も多いが、日本語書籍の場合、とりわけ縦書きの場合には、リスト表示は適していない。（「番号付きリスト」では漢数字が使用できないなど）

その場合、リストではなく、インデントを使って表現する。

番号付きリストとして表示されることが重要な場合には、リストを使用する。

ルビ

ルビは、HTML5のルビタグを使用する。

(例)

```
<ruby>奮<rt>ふん</rt>戦<rt>せん</rt>記<rt>き</rt></ruby>
```

この形式は、XHTMLのルビよりも記述方法が簡便になり見通しがよくなっている。

XHTMLのルビタグでは、ルビに対応していないブラウザのために括弧を表示するrp要素があったが、その使用は推奨されてない。EPUB読書システムでは、ルビは必ず解釈しなければならない。

画像

画像の指定方法は、インラインで指定するか、フロートを使って左（縦書きの場合には上）あるいは右（縦書きの場合には下）に配置しテキストを回り込みさせるかのいずれかの方法となり、ページ内で位置を指定して表示させることはできない。

コミックや表紙画像、扉画像のように、ページ一杯に表示させる場合には、HTML5コンテンツ文書内で1つだけ画像を指定し、ページ内にフィットさせるスタイルを定義する。

インライン画像

単に画像を指定した場合にはインライン画像となる。

フロート

画像をページ内の左（縦書きの場合には上）あるいは右（縦書きの場合には下）に配置しテキストを回り込みさせる場合にはスタイルのフロート (float) プロパティを使用する。フロートを指定した場合には、clearプロパティでフロートを解除しなければならない。

キャプション

画像の下にテキストを表示するような表現はよくあるが、マークアップでこれを行った場合、EPUB読書システムでは画像とキャプションが別ページに分かれてしまう可能性がある。

現時点では、画像につけるキャプションは、マークアップで行うよりも、画像とキャプションをひとつの画像にすることが望ましい。SVG形式にすれば、拡大したときにもキャプションをきれいに表示することができる。またimgのalt属性に、キャプションの文字列を入れておくことで、アクセシビリティへの対処も可能である。

SVG画像

表やグラフ等、Illustrator等で作成された図については、SVG形式に出力し、それを使用することができる。写真や図画などのビットマップ画像の場合でも、前述のように、キャプションに文字列を使用する場合等もSVG形式を使うことができる。

SVG形式の画像を使う場合には、「パッケージ文書」のmanifestで指定するmedia-typeは、"image/svg+xml"にする。HTML5への埋め込みは、imgタグにて、

のように記述する。

表組はSVG画像で

画像のキャプションの場合と同様、EPUB読書システムによっては、表組が実現できたとしても、画面サイズによって、表が複数ページにまたがってしまう可能性がある。現時点では、画像として用意することが望ましい。SVG形式にすれば、拡大したい際にもテキストをきれいに表示することができる。

文字装飾

文字装飾については、fontタグではなくspanタグで装飾する箇所をマークアップし、そこに対してスタイルを指定する。

相反するスタイルは同時に指定した場合には後で指定されたほうが優先される。例えば傍点 (-epub-text-emphasis-style) は、同一のクラスで複数定義しても最後に定義したものだけが有効となる。

縦中横

縦中横は「共通スタイルシート」に用意された共通クラス"tcy"を指定する。

(例)

```
<span class="tcy">9</span>月<span class="tcy">12</span>日
```

下付き／上付きは使用方法を限定する

sub, supというタグを使ったマークアップではなく、以下のクラスを作成して、spanで指定する。これは、縦組／横組にかかわらず使用できるので、書籍ごとの基本スタイルシートに定義してよい。

```
.sup { /*上付き*/  
  font-size:50%;  
  vertical-align:super;  
  letter-spacing: -2px  
}  
.sub { /*下付き*/  
  font-size:50%;  
  vertical-align:sub;  
  letter-spacing: -2px  
}
```

漢文も、返り点または送り仮名のいずれか一方であれば、これを使って表現可能。しかし、現時点では、返り点と送り仮名の同時指定はできない。

(例)

```
国破<span class="sup">レテ</span>山河在<span class="sup">リ</span><br />  
烽火連<span class="sub">二</span>三月<span class="sub">一</span>
```

同一ページでの、縦書きと横書きの混在は避ける

同一ページ（同一HTML5コンテンツ文書）内で、縦書き、横書きを混在させた場合、オーバーフローした際の表示が保証されないため、指定すべきではない。

表紙、奥付を横書き、本文は縦書き、という指定は、HTML5ファイルを分割し、それぞれのbodyに対して、-epub-writing-modeを指定すればよい。

-epub-writing-modeは、bodyに対して指定する。それよりも下位の要素に対して指定してはならない。

コミックの場合の記述方法

「パッケージ文書」

基本構造

```
<?xml version="1.0"?>
<package xmlns="http://www.idpf.org/2007/opf" unique-identifier="pub-id"
version="3.0" xml:lang="ja">
  <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:title> (タイトル) </dc:title>
    <dc:creator> (著者) </dc:creator>
    <meta refines="#creator" property="role" scheme="marc:relators" id="role">aut</
meta>
    <dc:publisher> (出版社) </dc:publisher>
    <dc:language>ja</dc:language>
    <dc:identifier id="pub-id">urn:uuid:C19BE97E-99CA-4DE5-9062-96EC6B0DC2F0</
dc:identifier>
    <dc:type>comic</dc:type>
    <meta property="dcterms:modified">2011-09-23T00:00:00Z</meta>
  </metadata>
  <manifest>
    <item id = "comic_style0001" href = "styles/comic.css" media-type = "text/css" />
    <item id = "toc" href = "toc.xhtml" properties = "nav" media-type = "application/
xhtml+xml" />
    <item id="IMG00001" href="images/P01.jpg" media-type="image/jpeg" />
    <item id="IMG00002" href="images/P02.jpg" media-type="image/jpeg" />
    <item id="IMG00003" href="images/P03.jpg" media-type="image/jpeg" />
    <item id="IMG00004" href="images/P04.jpg" media-type="image/jpeg" />
    <item id="IMG00005" href="images/P05.jpg" media-type="image/jpeg" />
    <item id="XHTML00001" href="text/P01.xhtml" media-type="application/xhtmll
+xml" />
    <item id="XHTML00002" href="text/P02.xhtml" media-type="application/xhtmll
+xml" />
    <item id="XHTML00003" href="text/P03.xhtml" media-type="application/xhtmll
+xml" />
```

```

<item id="XHTML00004" href="text/P04.xhtml" media-type="application/xhtml+xml" />
<item id="XHTML00005" href="text/P05.xhtml" media-type="application/xhtml+xml" />
</manifest>
<spine page-progression-direction="rtl">
  <itemref idref="XHTML00001" properties="page-spread-left" />
  <itemref idref="XHTML00002" properties="page-spread-right" />
  <itemref idref="XHTML00003" properties="page-spread-left" />
  <itemref idref="XHTML00004" properties="page-spread-right" />
  <itemref idref="XHTML00005" properties="page-spread-left" />
</spine>
</package>

```

dc:type

metadataのdc:typeの子要素はcomicとする。

(例)

```
<dc:type>comic</dc:type>
```

spine itemref properties

spine itemrefには、propertiesというオプション属性が用意されている。

これは、コミックやグラビア等で、見開き表示になった場合に、どちらのページに表示するかを指定するものなので、

"page-spread-left"：見開き時に左ページに表示

"page-spread-right"：見開き時に右ページに表示

を指定することで、見開き時の表示位置を指定することができる。

単行本の場合のページ構成の場合に、

- 表紙は左ページ
- 見返しを挟んで扉ページが左ページ
- その後は右ページ、左ページと交互に続く

ように、左ページが連続する場合がある。

その場合には、使用としては、properties="page-spread-left"を連続して指定すればいいはずだが、正しく表示されるかはEPUB読書システムの実装にゆだねられるため、互換性を最優先する場合には、見返しページには白ページを用意し、"page-spread-left"と"page-spread-right"が、必ず交互になるようにする方法もある。

EPUB読書システムは、そのように指定された場合、見開き表示の場合には、空白を挟む（上記の例では見返しページにあたる）必要がある。単ページ表示の場合には、空白を挟まず、spineで指定された通りの順番で表示する。

CSS

コミックの基本スタイル定義

コミックの場合、通常、1つのHTML5に1つの画像を指定するので、スタイルの指定は必須ではないが、ページにフィットさせるため、以下のスタイルを用意する。

```

@charset "UTF-8";
@page {
  margin:0px;
}
img.fit {
  display:block;
  margin: auto;
  height: auto;
  width: auto;
  max-width:100%;
  max-height:100%;
}

body {
  text-align:center;
  background-color: #eeeeee;
  margin-top: 0px;
  margin-bottom: 0px;
  margin-left: 0px;
  margin-right: 0px;
}

```

画像ファイル

コミックを表示する画像は、EPUB3の標準メディアタイプである、JPEG、PNG、GIF、SVGが使用可能である。アニメーションGIFは対象外とする。

SVGについては、SVGファイルをそのままspineに登録することもEPUB3の仕様上は可能だが、汎用性も考慮し、HTML5コンテンツ文書にエンベッドして使用することを、本基準の仕様とする。

HTML5

コミックの場合の記述方法

基本的には、1つのHTMLに1つの画像で構成する。

コミックの場合、Alternate Styleは使用しない。

(例)

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:epub="http://www.idpf.org/2007/ops" xml:lang="ja">
<head>
  <meta charset="UTF-8" />
  <link rel="stylesheet" href="../styles/comic.css" type="text/css" />

```

```
<title> (タイトル) </title>
</head>
<body>

</body>
</html>
```

クリックابلマップについて

目次等でクリックابلマップを使いたい場合があるが、多くのreading Systemで、クリックابلマップは、画像の表示が等倍のときのみ有効である。画像がウィンドウサイズに合わせて拡大縮小されている場合には、マップで指定した領域が、その拡大縮小に追随しないため、クリックابلマップとして正しく動作させることは事実上不可能である。

目次項目はTOCファイルを用意することで対応する。

Appendix

用語の用法について

EPUB3ドキュメントの中で使用されている用語について、必ずしも日本語として定着していないものがある。ここでは、そういった用法について、整理しておく。

EPUBドキュメント	本基準書	備考
Reading System	EPUB読書システム	いわゆる電子書籍ビューアをさす。
UA(User Agent)	*	本基準書では使用していない。通常はブラウザを含めたユーザーの電子書籍読書環境をさすことが多いが、EPUB関連の文書で使用される場合にはReading Systemと同義。
media-type	メディアタイプ	ファイルフォーマットの種類。MIMEタイプ、コンテンツタイプとほぼ同義。CSSの用語でMedia typesと言った場合には印刷 (print) や画面 (screen) 点字デバイス (braille) といったターゲットメディアのことをさす。
root	ルート	階層構造となっているフォルダ (ディレクトリ) の最上位をさす。UNIX系OSではシステムの全権限をもったユーザーのこともルートと呼ぶが、本基準書では使用していない。
root element	ルート要素	XML文書のツリー構造で最上位の要素のこと。
the Package Document	「パッケージ文書」	EPUB rootfileと書いてある場所があったり、OPFと書いてある箇所があったりするの で、まだ用語として定着していない。本基準書では「」をつけて「パッケージ文書」としている。

EPUBドキュメント	本基準書	備考
XHTML Content Documents	HTML5コンテンツ文書	元のEPUB3ドキュメントの中では、HTML5、XHTMLが混在しているが、本基準書では、最初の方で、EPUB3で使用できるHTML5について定義することで、それ以降は構造説明をする場合を除いてはHTML5に統一している。
namespace	ネームスペース	日本語では名前空間と呼ぶ場合もあるが、本基準書ではネームスペースに統一した。
child	子要素	例えばchild of the metadata element (metadataの子要素) のように使用する。
attribute	属性	
prefix	プレフィックス	接頭辞のこと。
semantec	セマンティック	日本語でも「セマンティックWeb」など定着しつつあるので、片仮名表記とした。
Core Media Type	標準メディアタイプ	