

CSS によるページ組版入門

アンテナハウス（株） 村上 真雄

2008 年 9 月 4 日
(更新 : 2008 年 12 月 8 日)



はじめに

CSS (Cascading Style Sheets)は、「Web デザイン」のためばかりではなく、幅広い印刷用途、PDF などページ媒体への出力にも使うことのできるスタイルシート仕様です。

とくに W3C で策定作業中の CSS Level 3 では、高度なページ組版機能、段組、縦書き、さまざまな言語の文字レイアウトなど、プロの印刷品質の組版に必要な機能が多く定義されています⁽¹⁾。

CSS3 仕様の実装を進めている Antenna House Formatter V5（開発途上、プレビュー版公開中）を使いながら、CSS によるページ組版を解説します。

⁽¹⁾ CSS3 は単一の仕様ではなく、機能ごとのモジュールに分かれています（[Chapter 20. 参照仕様](#)に主なものをまとめてます）。CSS3 を学ぶには、まず基礎になる CSS2.1（CSS Level 2 Revision 1、現在勧告候補）の知識が必要です。

目次

Chapter 1. ページ設定.....	4
Chapter 2. 柱とノンプル.....	8
Chapter 3. カウンタ.....	9
Chapter 4. クロスリファレンス.....	10
Chapter 5. 目次作成.....	11
Chapter 6. 改ページの制御.....	12
Chapter 7. 名前付きページ.....	13
Chapter 8. 段組.....	14
Chapter 9. ページフロート.....	15
Chapter 10. 脚注.....	16
Chapter 11. PDF ブックマーク.....	17
Chapter 12. CMYK カラー.....	18
Chapter 13. 丸め罫.....	19
Chapter 14. 影付きボックス.....	20
Chapter 15. ハイフネーション.....	21
Chapter 16. 日本語文字組版.....	22
Chapter 17. 縦書き.....	24
Chapter 18. MathML と SVG グラフィック.....	26
Chapter 19. Antenna House Formatter.....	27
Chapter 20. 参照仕様.....	28

Chapter 1. ページ設定

1.1 @page ルール

ページサイズ、余白、ページ ヘッダ/フッタなどページの基本的な設定は、@page ルールで行います。

```
@page {  
  size: A4;  
  margin: 25mm;  
  @top-center {  
    content: "Sample";  
  }  
  @bottom-center {  
    content: counter(page);  
  }  
}
```

1.2 ページサイズ指定 : size プロパティ

size プロパティでページの幅と高さを指定します。

```
@page {  
  size: 182mm 230mm;  
}
```

```
@page {  
  size: 4in 6in;  
}
```

size プロパティには、A5、A4、A3、B5、B4 などのキーワードも使えます。

```
@page {  
  size: A4; /* ISO/JIS A4 (210mm×297mm) */  
}
```

```
@page {  
  size: B5; /* ISO B5 (176mm×250mm) */  
}
```

```
@page {  
  size: JIS-B5; /* JIS B5 (182mm×257mm) */  
}
```

キーワード **landscape** を同時に指定することで、横長の指定ができます。

```
@page {  
  size: A4 landscape; /* A4ヨコ (297mm×210mm) */  
}
```

1.3 マージン

@page ルールの margin プロパティでページ余白を指定します。

```
@page {  
  margin: 10%; /* 上下左右の余白は、ページ幅の10%ずつ */  
}
```

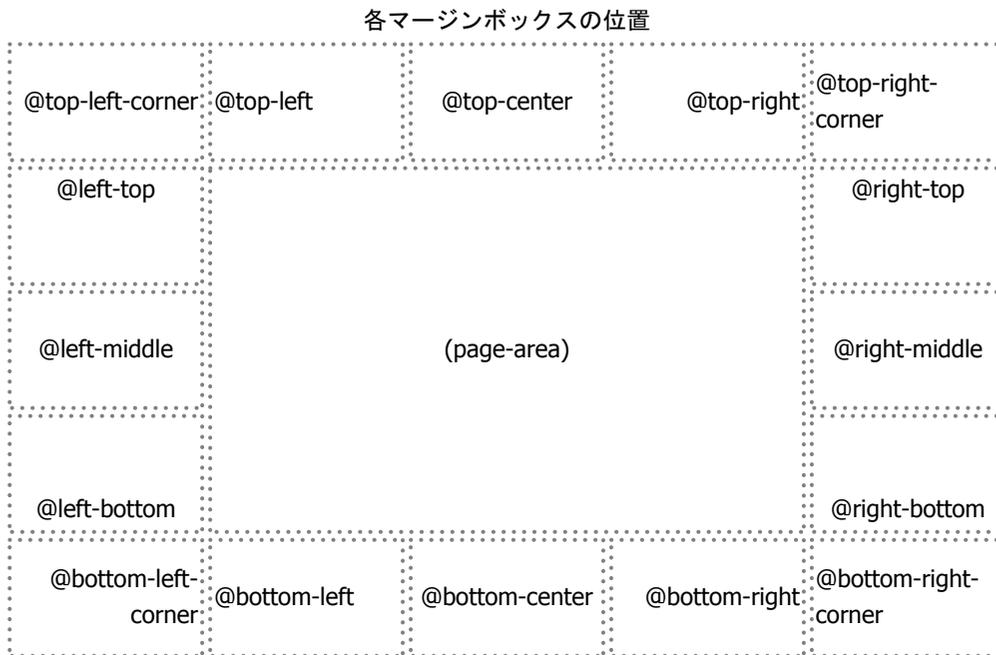
```
@page {  
  /* 上下余白を2cm、左右余白を3cm とする */  
  margin-top: 2cm;  
  margin-bottom: 2cm;  
  margin-left: 3cm;  
  margin-right: 3cm;  
}
```

1.4 マージンボックス

ページヘッダやページフッタはマージンボックスと呼ばれるページの周辺の領域に割り当てます。

マージンボックスは、@top-left、@top-center、@top-right、@bottom-left、@bottom-center、@bottom-right などページ周辺の位置により名前がついています。

```
@page {  
  @top-right { /* ページヘッダ */  
    content: "Sample";  
  }  
  @bottom-center { /* ページフッタ */  
    content: counter(page);  
  }  
}
```



1.5 左・右・先頭ページ

左ページと右ページで、左右の余白やページヘッダ／フッタの表示内容を変えることができます。また、先頭ページだけ特別にすることもできます。

```
@page :left { /* 左ページ設定 */
margin-left: 23mm;
margin-right: 27mm;

@top-left { /* 左ページの柱に章のタイトル */
content: string(Chapter);
}

@bottom-left { /* ノンブル(ページ番号)*/
content: counter(page);
}
}

@page :right { /* 右ページ設定 */
margin-left: 27mm;
margin-right: 23mm;

@top-right { /* 右ページの柱に節のタイトル */
content: string(Section);
}
}
```

```
@bottom-right { /* ノンブル(ページ番号)*/  
  content: counter(page);  
}  
}  
  
@page :first { /* 文書の先頭ページ設定 */  
  /* ページヘッダ非表示 */  
  @top-right { content: none }  
  @top-left { content: none }  
}
```

Chapter 2. 柱とノンブル

2.1 柱の設定 : stringt-set プロパティと string()関数

ページヘッダに本文中の見出しからとった文字列を表示することができます。

```
@page {
  @top-left {
    content: string(Chapter);
  }
}
h1 { string-set: Chapter self; }
```

2.2 ノンブル (ページ番号) : counter(page)

ノンブル (ページ番号) の出力には、ページカウンタ **counter(page)** を使います。

```
@page {
  @top-right {
    content: "Page " counter(page);
  }
}
```

2.3 総ページ数 : counter(pages)

「Page 8 of 28」のように、現在のページ番号とともに総ページ数を出力できます。

```
@page {
  @top-right {
    content: "Page " counter(page) " of " counter(pages);
  }
}
```

Chapter 3. カウンタ

3.1 ページカウンタ

現在ページ番号と総ページ数 `counter(page) / counter(pages)` ⇒ 9 / 28

3.2 章や節に番号をつける

カウンタにより、章や節の番号を自動的につけることができます。

```
body {
  counter-reset: ChapterNo;           /* 章番号をリセット */
}
h1:before {
  counter-increment: ChapterNo;       /* 章見出し(h1)の前に */
  content: "第" counter(ChapterNo) "章 "; /* 章番号に 1 足す */
}
h1 {
  string-set: Chapter before self;    /* "第1章 " などと出力 */
  counter-reset: SectionNo;          /* 節番号をリセット */
}
h2:before {
  counter-increment: SectionNo;       /* 節見出し(h2)の前に */
  content: counter(ChapterNo) "." counter(SectionNo) " "; /* 節番号に 1 足す */
}
h2 {
  string-set: Section before self;    /* 1.1 などと出力 */
}
@page :left {
  @top-left {
    content: string(Chapter);          /* 左ページの柱に章のタイトル */
  }
}
@page :right {
  @top-right {
    content: string(Section);         /* 右ページの柱に節のタイトル */
  }
}
```

Chapter 4. クロスリファレンス

4.1 カウンタ（章番号、ページ番号）の参照：target-counter()関数

「[Chapter 3. カウンタ \(page 9\)](#)を参照」のように、参照先の章番号やページ番号を自動的に付加することができます。

```
.ChapterRef::before {
  content: "Chapter " target-counter(attr(href, url), ChapterNo) ". ";
}
.PageRef::after {
  content: "(page " target-counter(attr(href, url), page) ")";
}
...
<a class="ChapterRef PageRef" href="#Counters">カウンタ</a>を参照
```

4.2 テキスト内容の参照：target-text()関数

参照先のテキストをもってくることもできます。例：「[Chapter 4. クロスリファレンス](#)」

```
.TitleRef {
  content: target-text(attr(href, url), before)
          target-text(attr(href, url), content);
}
...
例:「<a class="TitleRef" href="#CrossRef">この章</a>」
```

Chapter 5. 目次作成

5.1 クロスリファレンスと目次

target-counter()による章番号やページ番号の参照 ([Chapter 4. クロスリファレンス参照](#)) を使って、次のような目次を作ることができます。

Chapter 1. ページ設定.....	4
Chapter 2. 柱とノンブル.....	8
Chapter 3. カウンタ.....	9

```
.TOC a::before { /* 目次項目に章番号をつける */
  content: "Chapter " target-counter(attr(href, url), ChapterNo) ". ";
}
.TOC a::after { /* 目次項目にページ番号をつける */
  content: leader(dotted) " " target-counter(attr(href, url), page);
}
...
<div class="TOC">
  <ul>
    <li><a href="#PageSetup">ページ設定</a></li>
    <li><a href="#PageHeaderFooter">柱とノンブル</a></li>
    <li><a href="#Counters">カウンタ</a></li>
  </ul>
</div>
```

5.2 リーダー : leader()関数

目次では、項目タイトルとページ番号の間をリーダー（点線など）で伸ばして、ページ番号を右寄せにする形式がよく使われます。leader()関数はそのためのものです。

- leader(dotted) leader(dotted)
- leader(solid) _____ leader(solid)
- leader(space) leader(space)
- leader("*-") *- leader("*-")

Chapter 6. 改ページの制御

6.1 改ページ : page-break-before, page-break-after

```
/* 大見出し(h1)の前で強制改ページ */  
h1 {  
  page-break-before: always;  
}
```

```
/* このブロックのあとで必ず改ページ */  
div.Ending {  
  page-break-after: always;  
}
```

6.2 改ページの禁止

```
/* 見出し(h1~h6) の直後で改ページしてはいけない決まり */  
h1, h2, h3, h4, h5, h6 {  
  page-break-after: avoid  
}
```

```
/* このブロックの中では改ページ禁止 */  
div.NoBreak {  
  page-break-inside: avoid;  
}
```

6.3 左右どちらかのページから始める

章のはじまりのページを常に右側または左側のページにするといった指定ができます。必要に応じて空白ページが挿入されます。

```
/* h2見出しが常に右ページに来るように改ページ */  
h2 { page-break-before: right; }
```

Chapter 7. 名前付きページ

7.1 名前付きページ : page プロパティ

名前付きの@page ルールを複数用意して、文書内で page プロパティにより切り替えて使うことができます。

```
@page Landscape { /* 名前付きページ Landscape 定義 */
  size: A4 landscape;
}
@page Appendix { /* 名前付きページ Appendix 定義 */
  @top-center: "Appendix";
}
table.WideTable {
  page: Landscape; /* 横に大きな表を Landscape ページに */
}
div.Appendix {
  page: Appendix; /* 付録を Appendix ページに */
}
```

名前付きページは、左・右・先頭ページの指定と組み合わせることができます。

```
@page Chapter:first {
  /* 章の先頭ページ */
}
@page Chapter:left {
  /* 章の左ページ */
}
@page Chapter:right {
  /* 章の右ページ */
}
div.Chapter {
  page-break-before: right;
  page: Chapter;
}
```

Chapter 8. 段組

8.1 段組の指定 : column-count, column-gap, column-rule

ここは段数 (column-count) を 2 と指定した段組のブロックです。段間 (column-gap)、段間罫 (column-rule) も指定しています。

```
div.MultiCol {  
  column-count: 2;  
  column-gap: 5mm;  
  column-rule: dotted green 1mm;  
}
```

段組の指定には、段数 (column-count) を指定するかわりに、段幅 (column-width) を

- 指定して、全体の幅と段幅により自動的に段数が決まるようにする方法もあります。
- 段間罫を指定する column-rule はショートハンドプロパティであり、次のように個別に指定することもできます。

```
column-rule-style: dotted;  
column-rule-color: green;  
column-rule-width: 1mm;
```

ここは TopFloat です。 **float: top**

ここは TopPageFloat です。 **float: page top**

Chapter 9. ページフロート

9.1 top-float と bottom-float

ここでは、ページの上部や下部に配置されるページフロート (top-float および bottom-float) を使っています。

これは float: left

従来の HTML と CSS のレイアウトで、画像などを左
か右に寄せて配置して本文テキストを回り込ませる左

これは float: right

右の float はよく使われていますが、ページフロートはそれをページの縦方向に拡張したものです。

9.2 段組の中での top-float と bottom-float

ここは段内の TopFloat です。

float: top

Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Donec fringilla. Donec luctus
ante at dolor scelerisque tempor. Phasellus
convallis, nisl sed sollicitudin lacinia, in
pulvinar nibh eros non sem.

Donec sit amet velit. Phasellus cons
ectetuer. Pellentesque ut magna. Quisque

• enim turpis, fringilla id, malesuada ut, molestie
• sed, tellus.

• Cras commodo, pede id dapibus lacinia,
• nulla ante gravida libero, ac cursus neque diam
• at massa. Suspendisse et est id eros gravida
• commodo. Aenean imperdiet tristique urna.

ここは段内の BottomFloat です。

float: bottom

ここは BottomFloat です。 **float: bottom**

ここは BottomPageFloat です。 **float: page bottom**

Chapter 10. 脚注

10.1 脚注の指定 : float: footnote

float プロパティの値に footnote を指定するとその要素の内容が脚注⁽¹⁾になります⁽²⁾。

```
<style>
.Footnote { float: footnote }
</style>
```

```
<p>floatプロパティの値にfootnoteを指定するとその要素の内容が脚注
<span class="Footnote">脚注とはページの下の方に置かれる注です。</span>
になります。...</p>
```

⁽¹⁾ 脚注とはページの下の方に置かれる注です。

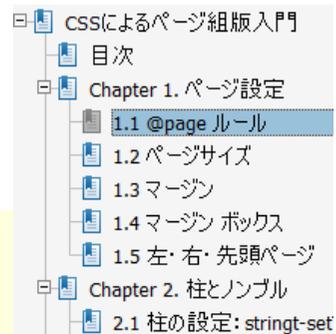
⁽²⁾ 脚注の配置には [Chapter 9. ページフロート](#) の float: page bottom が使われています。脚注領域の設定は @page ルールの中で @footnote ルールを使います。脚注の番号の形式を設定するには、::footnote-call、::footnote-marker 擬似要素を使います。

Chapter 11. PDF ブックマーク

11.1 ブックマーク設定 : bookmark-level, bookmark-label

PDFのブックマーク（しおり）を生成することができます。ブックマークの項目を作るには、bookmark-level プロパティでレベルを、bookmark-label プロパティでラベルを指定します。HTMLのh1～h6の見出しレベルに対応します。ラベルは指定しなければ要素の内容テキストです。

```
h1 { bookmark-level: 1 }  
h2 { bookmark-level: 2 }  
h3 { bookmark-level: 3 }  
h4 { bookmark-level: 4 }  
h5 { bookmark-level: 5 }  
h6 { bookmark-level: 6 }
```



Chapter 12. CMYK カラー

12.1 CMYK カラーの指定 : `cmyk(c, m, y, k)`

印刷では RGB よりも CMYK カラーが使われます。

```
<p style="border: cmyk(0.5, 0.1, 0.0, 0.2) solid thick;  
background: cmyk(0, 0.3, 0.2, 0);  
color: cmyk(0.8, 0.5, 0.0, 0.3)">CMYK Color Test...
```

CMYK Color Test. ボーダー色、背景色、文字色を CMYK で指定しています。

Chapter 13. 丸め罫

13.1 丸め半径の指定 : border-radius

ボーダーの角を丸くするには、border-radius プロパティで、その半径を指定します。

```
<p style="border-radius: 18pt; /* 角の丸め半径 */  
border: solid green;  
padding: 6pt">Border-radius...
```

Border-radius (丸め罫) を指定しています。

4つの角の丸め半径を個別に指定したり、角の丸めを楕円にすることもできます。

Chapter 14. 影付きボックス

14.1 影付きボックスの指定 : box-shadow

ボックスに影を付けるには、box-shadow プロパティに水平方向と垂直方向の影の長さや影の色を指定します。

```
<p style="box-shadow: 4pt 3pt silver;  
border: solid 1pt black; padding: 6pt">...
```

Box-shadow (影付きボックス) を指定しています。

```
<p style="box-shadow: -6pt -4pt orange, 6pt 4pt blue;  
border-radius: 10pt; padding: 6pt">...
```

複数の影を指定することもできます。border-radius と組み合わせ可能です。

Chapter 15. ハイフネーション

15.1 ハイフネーションを有効にする : `hyphens: auto`

```
.Hyphenated {  
  hyphens: auto;  
}  
<div class="Hyphenated" xml:lang="en">  
  <p>Rainbow PDF Software Products are ...
```

Rainbow PDF Software Products are developed by Antenna House. Antenna House, Inc. is a premier software company founded in 1984 in Tokyo, Japan. Our mission is to create and sell products to make data useful. For over 20 years we have been reliably retrieving information and delivering it to our customers in formats and languages our customers can use. Millions of copies of our software have saved time for individual desktop users, small businesses, and have cemented our leadership role in high end documentation.

Chapter 16. 日本語文字組版

16.1 日本語組版のための設定

日本語文書を組版する場合の基本的なスタイル設定は次のようなものです。

```
body { /* 日本語組版のための設定 */
  punctuation-trim: start end adjacent;
  text-justify-trim: punctuation;
  text-autospace: ideograph-numeric ideograph-alpha;
}
p { /* 段落 */
  text-align: justify; /* 行末を揃える */
  text-indent: 1em; /* 段落は全角1字下げ */
  margin: 0; /* 段落間は空けない */
}
```

以下、日本語組版のための設定に使うプロパティを説明します。

16.2 約物の詰め処理 : punctuation-trim

```
/* 行頭、行末、約物どうし隣接のとき、全角約物を半角幅に */
punctuation-trim: start end adjacent;
```

「《約物 [やくもの]》、つまり『括弧』・『句読点』の類 (たぐい) です。」のように、約物 (句読点や括弧類) が連続する場合や行頭や行末に来たとき、全角文字の約物を半角幅に詰めて、見栄えをよくします。

「《約物 [やくもの]》、つまり『括弧』・『句読点』の類 (たぐい) です。」←こちらは比較のために、約物の詰めを無効にした例です (punctuation-trim: none を指定)

16.3 行の調整のための約物の詰め : text-justify-trim

```
/* 行の調整のために約物の詰めを行う */
text-justify-trim: punctuation;
```

この指定により、前節で説明した約物を半角幅に詰める場合に該当しない場合でも、行末をびったり揃える行の調整のために、全角の括弧類の前後や句読点の後ろを、必要に応じて自動的に詰めます。

16.4 和欧文間の空き : text-autospace

```
/* 漢字や仮名と数字の間、漢字や仮名と欧文の間にアキを入れる */  
text-autospace: ideograph-numeric ideograph-alpha;
```

「日本語にも global にも 100%を目指す AH Formatter V5 です」のように、日本語の文章の中に欧文や算用数字が混在するとき、間に少しアキを入れて読みやすくします。

「日本語にも global にも 100%を目指す AH Formatter V5 です」←こちらは比較のために、和欧文間の空きを無効にした例です (text-autospace: none を指定)

Chapter 17. 縦書き

17.1 縦書きの指定 : writing-mode: tb-rl

文書全体を縦書きにするには、body 要素（HTML ではなく XML 文書の場合はルート要素）に writing-mode プロパティを指定します。

```
body {  
  writing-mode: tb-rl; /* 縦書き */  
}
```

ブロックを縦書きにするには、ブロック要素に writing-mode プロパティを指定します。

```
div.VerticalTextBlock {  
  writing-mode: tb-rl; /* 縦書き */  
  height: 16em; /* 1行の文字数 */  
  padding: 3pt; border: ridge green;  
}
```

日本語は伝統的に縦書きで組まれます。書籍や雑誌など出版物の多くは今も縦書きが主流です。もちろん Antenna House Formatter は縦書きにも対応しています。このように部分的にブロックを縦書きにする 것도、文書全体を縦書きにすることもできます。

縦書きの指定は **writing-mode: tb-rl** です。tb-rl は、文字の進行方向が上から下 (top-to-bottom)、行とブロックの進行方向が右から左 (right-to-left) を意味しています。横書きの指定は **writing-mode: lr-tb** (left-to-right top-to-bottom) です。アラビア語やヘブライ語など右から左に書く言語の横書きの場合 **writing-mode: rl-tb** (right-to-left top-to-bottom) です。

縦書きの中に「08年8月29日」のように部分的に数字などを横書きにすることを「縦中横」といいます。

17.2 縦中横

前節の縦書きブロックの例の中にある縦中横（縦書きの行の中に数字などを横書きにする）は、次のようにして実現できます。

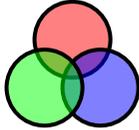
```
/* 縦中横 */  
span.TateChuYoko {  
  display: inline-block; /* 行の中に小さなブロックを作る */  
  writing-mode: lr-tb; /* この小さなブロックを横書きにする */  
}
```

```
text-align: center; /* 中央揃えに */
text-indent: 0; /* 字下げ無し(段落pのtext-indentが継承されないように)*/
line-height: 1; /* 行間のアキが縦中横の上下につかないように */
}

<div class="VerticalTextBlock">
  ...
  <p>縦書きの中に「<span class="TateChuYoko">'08</span>年8月
  <span class="TateChuYoko">29</span>日」のように部分的に数字などを
  横書きにすることを「縦中横」といいます。</p>
</div>
```

縦書きの中に
「08年8月29日」の
ように部分的に数
字などを横書きに
することを「縦中
横」といいます。

Chapter 18. MathML と SVG グラフィック

MathML の数式 $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ や SVG のベクターグラフィック  を埋め込むことができます。

```
<p>MathMLの数式
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow><mi>x</mi><mo>=</mo><mfrac>...</mfrac></mrow>
</math>
やSVGのベクターグラフィック
<svg xmlns="http://www.w3.org/2000/svg" width="70" height="65"
  viewBox="0 0 70 65">
  <g style="fill-opacity:.5; stroke:black; stroke-width:2;">
    <circle cx="35" cy="20" r="19" style="fill:red;">...
  </g>
</svg>
を埋め込むことができます。</p>
```

Chapter 19. Antenna House Formatter

この『CSSによるページ組版入門』は、Antenna House Formatter V5（プレビュー版公開中）を使って組版しています。その特徴として次のことがあげられます。



- ✓ CSS と [XSL-FO](#) の2つのスタイルシート言語に対応——AH Formatter V5 は XSL-FO 対応で実績のある Antenna House XSL Formatter V4 の次期メジャーバージョンです。今後は両方のスタイルシート言語を適材適所で使い分けすることができます。
- ✓ 多言語組版——日本語や欧文はもちろん、中国語（簡体字・繁体字）、ハングル、アラビア文字、ヘブライ文字、デヴァナガリ文字、タイ文字など、Unicode で扱えるほとんどの文字を扱うことができ、これらを混在させた文書が容易に作成可能です⁽¹⁾。40ヶ国語のハイフネーションも可能。
- ✓ 印刷と PDF 出力機能が充実——PDF/X、PDF/A、タグ付き PDF 出力など。
- ✓ 豊富な組み込みインターフェース——コマンドライン、.NET、Java、C++/C、COM の各インターフェースにより、アプリケーションへの組み込みが簡単。
- ✓ SVG、MathML、EMF、WMF、CGM をネイティブに描画。
- ✓ CSS3 仕様の完成に貢献する——アンテナハウスは W3C のメンバーであり、XSL-FO と CSS の仕様策定に協力しています。CSS3 のページ組版・文字組版に関する仕様は、まだ未完成のドラフトの状態です。AH Formatter は世界に先駆けて CSS3 ドラフト仕様を実装し、その完成に貢献します。
- ✓ 日本語組版規則に対応——世界標準仕様である XSL-FO や CSS は、世界中の言語での使用が考慮されていますが、日本語組版規則（『日本語文書の組版方法』JIS X 4051:2004 として日本の標準規格となっている）の要求と合致しないところもあり、アンテナハウスでは、XSL-FO や CSS で、満足のいく日本語組版ができるよう、拡張仕様を実装し、それを世界標準仕様に反映させるよう提案も行ってきました。現在、W3C では、Japanese Layout Task Force という委員会がもたれ、日本語組版処理の要件をまとめる作業が行われており、アンテナハウスもそこに参加しています。

⁽¹⁾ プレビュー版の段階では CSS 組版での多言語処理の扱いに一部制限があります（アラビア語、ヘブライ語など）。

Chapter 20. 参照仕様

ここまで、CSS によるページ組版で何が出来るかをざっと紹介しました。以下は関係する主な仕様で、Antenna House Formatter が実装を進めているものです。

- ✓ **CSS2.1 (CSS Level 2 Revision 1)** W3C勧告候補 <http://www.w3.org/TR/CSS21/>
- ✓ **CSS3 Paged Media** 最終ドラフト <http://www.w3.org/TR/css3-page/>
- ✓ **CSS3 GCPM (Generated Content for Paged Media)**
..... ドラフト仕様 <http://www.w3.org/TR/css3-gcpm/>
..... Editor's Draft <http://dev.w3.org/csswg/css3-gcpm/>
- ✓ **CSS3 Multi-column layout** ドラフト仕様 <http://www.w3.org/TR/css3-multicol/>
- ✓ **CSS3 Backgrounds and Borders** ドラフト仕様 <http://www.w3.org/TR/css3-background/>
- ✓ **CSS3 Lists** ドラフト仕様 <http://www.w3.org/TR/css3-lists/>
- ✓ **CSS3 Text** ドラフト仕様 <http://www.w3.org/TR/css3-text/>
..... Editor's Draft <http://dev.w3.org/csswg/css3-text/>
- ✓ **CSS3 Text Layout** Editor's Draft <http://dev.w3.org/csswg/css3-text-layout/>
- ✓ **CSS3 Selectors** 最終ドラフト <http://www.w3.org/TR/css3-selectors/>
- ✓ **CSS3 Namespaces** W3C勧告候補 <http://www.w3.org/TR/css3-namespace/>
- ✓ **HTML 5—A vocabulary and associated APIs for HTML and XHTML**
..... ドラフト仕様 <http://www.w3.org/TR/html5/>