ANTENNA HOUSE Formatter V6 の紹介

2011-09-16

A Data Usability Company
アンテナハウス

むらかみ しんゆう
村上 真雄
@MurakamiShinyu
murakami@antenna.co.jp

目次

Chapter 1. AH Formatter、XSL-FO と CSS 組版について

1.1 AH Formatter とは

- 1999 年、当時 W3C ドラフトだった XSL (Extensible Stylesheet Language) 仕様をサポートする組版ソフト XSL Formatter 開発を企画。
- AH (XSL) Formatter は、多言語の大量の XML データからの自動組版などで威力を発揮して、けっこう世界で使われています。
- CSS (Cascading Style Sheets)による組版も研究。2009 年、AH Formatter V5 より、XSL だけでなく CSS 組版にも対応。

1.2 XSL とは : XSLT と XSL-FO

- XSL は「拡張可能なスタイルシート言語(Extensible Stylesheet Language)」で、XML 文書をレイアウトするためのもの。
- XSL は XML の変換を行う XSLT (XSL Transform)仕様とレイアウトを表現する XSL-FO (XSL Formatting Objects)仕様からなる。
- 元 XML 文書を XSLT を使って XSL-FO 形式に変換して、XSL-FO を組版する。

元 XML 文書の例 :

```
<文書>
  <表題>簡単XML入門</表題>
  <著者>あんてなハウス</著者>
  <見出し>XMLを書いてみる</見出し>
  <段落>XMLはこんなふうに書きます。</段落>
</文書>
```

XSL スタイルシートの例 :

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:fo="http://www.w3.org/1999/XSL/Format"
                version='1.0'>
  <xsl:template match="文書">
    <fo:root>
```

```
    <fo:layout-master-set>
      <fo:simple-page-master master-name="M">
        <fo:region-body margin="2cm"/>
      </fo:simple-page-master>
    </fo:layout-master-set>
    <fo:page-sequence master-reference="M">
      <fo:flow flow-name="xsl-region-body">
        <xsl:apply-templates />
      </fo:flow>
    </fo:page-sequence>
  </fo:root>
</xsl:template>
<xsl:template match="表題">
  <fo:block text-align="center" font-size="32pt">
    <xsl:apply-templates />
  </fo:block>
</xsl:template>
<xsl:template match="著者">
  <fo:block text-align="end" font-size="20pt">
    <xsl:apply-templates />
  </fo:block>
</xsl:template>
<xsl:template match="見出し">
  <fo:block font-size="16pt"
      space-before="1em" space-after="1em">
    <xsl:apply-templates />
  </fo:block>
</xsl:template>
<xsl:template match="段落">
  <fo:block text-align="justify" text-indent="1em">
```

```
      <xsl:apply-templates />
    </fo:block>
  </xsl:template>
</xsl:stylesheet>
```

1.3 CSS と XSL を比較

CSS で XML を組版するスタイルシートの例：

```
表題 {
  display: block;
  text-align: center;
  font-size: 32pt;
}

著者 {
  display: block;
  text-align: right;
  font-size: 20pt;
}

見出し {
  display: block;
  font-size: 16pt;
  margin-top: 1em;
  margin-bottom: 1em;
}
```

```
段落 {
  display: block;
  text-align: justify;
  text-indent: 1em;
}
```

この CSS スタイルシートを XSL スタイルシートと比較すると、

- XSLT のテンプレート `<xsl:template match="表題">...</xsl:template>` と CSS のルール "表題 {...}" が対応。
  （この "表題" の部分には XSLT では XPath 構文、CSS ではセレクタ構文を使う）
- XSL-FO の `fo:block` に対応するのは、CSS では `display: block` というプロパティ指定。
  （HTML の場合は `p`, `div`, `h1`〜`h6` などブロック要素がこれに対応）
- 体裁を指定するプロパティは共通または似ている。
  XSL-FO では `text-align="center"`、CSS では `text-align: center;` など。
  （XSL-FO のプロパティ仕様は CSS2.0 がベース）

1.4 XSL-FO のページマスター機能

XSL-FO では、ページの体裁（寸法、マージン、ページヘッダ／フッタの配置など）を「ページマスター」（fo:simple-page-master）で定義。複数のページマスターを、奇数ページ、偶数ページ、先頭ページ、最終ページ、空白ページといった条件でページに割り当てる。

XSL-FO のページマスター機能を使った例：

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format" xml:lang="ja" font="10pt/1.75 Meiryo">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="先頭ページ" page-width="148mm" page-height="210mm">
      <fo:region-body margin="25mm" />
    </fo:simple-page-master>
    <fo:simple-page-master master-name="奇数ページ" page-width="148mm" page-height="210mm">
      <fo:region-body margin="25mm" />
      <fo:region-before region-name="奇数ページヘッダ" extent="22mm" padding="0 25mm"
        display-align="after" />
      <fo:region-after region-name="奇数ページフッタ" extent="22mm" padding="0 25mm" />
    </fo:simple-page-master>
    <fo:simple-page-master master-name="偶数ページ" page-width="148mm" page-height="210mm">
      <fo:region-body margin="25mm" />
      <fo:region-before region-name="偶数ページヘッダ" extent="22mm" padding="0 25mm"
        display-align="after" />
      <fo:region-after region-name="偶数ページフッタ" extent="22mm" padding="0 25mm" />
    </fo:simple-page-master>

    <fo:page-sequence-master master-name="マスターA">
```

```
    <fo:repeatable-page-master-alternatives>
      <fo:conditional-page-master-reference page-position="first"
        master-reference="先頭ページ" />
      <fo:conditional-page-master-reference odd-or-even="odd"
        master-reference="奇数ページ" />
      <fo:conditional-page-master-reference odd-or-even="even"
        master-reference="偶数ページ" />
    </fo:repeatable-page-master-alternatives>
  </fo:page-sequence-master>
</fo:layout-master-set>

<fo:page-sequence master-reference="マスターA">
  <fo:static-content flow-name="奇数ページヘッダ">
    <fo:block text-align="end" font-size="8pt">
      <fo:retrieve-marker retrieve-class-name="節" />
    </fo:block>
  </fo:static-content>
  <fo:static-content flow-name="偶数ページヘッダ">
    <fo:block text-align="start" font-size="8pt">
      <fo:retrieve-marker retrieve-class-name="章" />
    </fo:block>
  </fo:static-content>
  <fo:static-content flow-name="奇数ページフッタ">
    <fo:block text-align="end" font-size="8pt">
      <fo:page-number />
    </fo:block>
  </fo:static-content>
  <fo:static-content flow-name="偶数ページフッタ">
    <fo:block text-align="start" font-size="8pt">
      <fo:page-number />
```

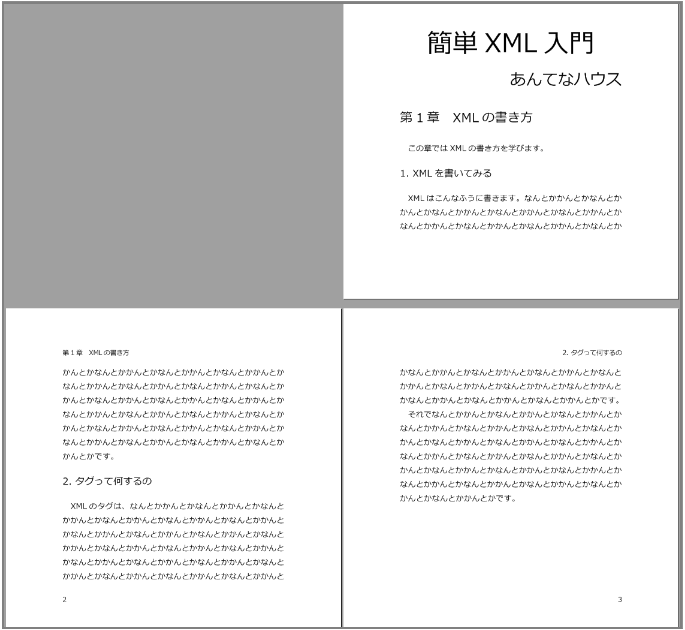
```
      </fo:block>
    </fo:static-content>

    <fo:flow flow-name="xsl-region-body">

      <fo:block text-align="center" font-size="32pt">簡単XML入門</fo:block>
      <fo:block text-align="end" font-size="20pt">あんてなハウス</fo:block>

      <fo:block font-size="16pt" space-before="1em" space-after="1em">
        <fo:marker marker-class-name="章">第1章 XMLの書き方</fo:marker>
        第1章 XMLの書き方
      </fo:block>
      <fo:block text-align="justify" text-indent="1em">この章ではXMLの書き方を学びます。</fo:block>
      <fo:block font-size="12pt" space-before="1em" space-after="1em">
        <fo:marker marker-class-name="節">1. XMLを書いてみる</fo:marker>
        1. XMLを書いてみる
      </fo:block>
      <fo:block text-align="justify" text-indent="1em">XMLはこんなふうに書きます。</fo:block>
      ……
      <fo:block font-size="12pt" space-before="1em" space-after="1em">
        <fo:marker marker-class-name="節">2. タグって何するの</fo:marker>
        2. タグって何するの
      </fo:block>
      <fo:block text-align="justify" text-indent="1em">XMLのタグは、なんとかかんとか。</fo:block>
      ……
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

AH Formatter での組版結果：



簡単 XML 入門

あんてなハウス

第 1 章　XML の書き方

この章では XML の書き方を学びます。

1. XML を書いてみる

XML はこんなふうに書きます。なんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとか

第 1 章　XML の書き方

かんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかかんとかです。

2. タグって何するの

XML のタグは、なんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかかんとかなんとかかんとかなんとかかんとかなんとかかんと

2

2. タグって何するの

かなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかなんとかかんとかです。

それでなんとかかんとかです。

3

1.5 CSS3 でのページマスターに相当する機能

CSS でも、CSS Paged Media Level 3（CSS3 ページ媒体向け仕様）の「ページルール」（@page）を使うと、XSL-FO のページマスター機能に相当することが可能です。ただし、いろいろと違いがあります。

CSS3 のページルールを使った例：

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="ja">
  <head>
    <title>簡単XML入門</title>
    <style type="text/css">
:root {
  font: 10pt/1.75 Meiryo;
}
@page {
  size: A5;                                   /* ページサイズ */
  margin: 25mm;                               /* ページマージン */
}
@page :first {                                /* 先頭ページ */
  @top-left { content: none; }                /* 柱やノンブルは無し */
  @top-right { content: none; }
```

```
  @bottom-left { content: none; }
  @bottom-right { content: none; }
}
@page :left {                                  /* 左(偶数)ページ */
  @top-left { content: string(章); }           /* 柱を左上に */
  @bottom-left { content: counter(page); }  /* ノンブルを左下に */
}
@page :right {                                 /* 右(奇数)ページ */
  @top-right { content: string(節); }          /* 柱を右上に */
  @bottom-right { content: counter(page); } /* ノンブルを右下に */
}

h1 { text-align: center; font-size: 32pt; counter-reset: 章番号; }
address.author { text-align: end; font-size: 20pt; font-style: normal; }

h2 {
  font-size: 16pt;
  margin-top: 1em;
  margin-bottom: 1em;
  string-set: 章 content(before) content();
  counter-increment: 章番号;
  counter-reset: 節番号;
}
h2::before { content: "第" counter(章番号) "章 "; }
```

```
h3 {
  font-size: 12pt;
  margin-top: 1em;
  margin-bottom: 1em;
  string-set: 節 content(before) content();
  counter-increment: 節番号;
}
h3::before { content: counter(節番号) ". "; }
    </style>
  </head>
  <body>
    <h1>簡単XML入門</h1>
    <address class="author">あんてなハウス</address>
    <h2>XMLの書き方</h2>
    <p>この章ではXMLの書き方を学びます。</p>
    <h3>XMLを書いてみる</h3>
    <p>XMLはこんなふうに書きます。</p>
    ……
    <h3>タグって何するの</h3>
    <p>XMLのタグは、なんとかかんとか。</p>
    ……
  </body>
</html>
```

CSS のページルールを XSL-FO のページマスター機能と比較すると、

XSL-FO	CSS
<fo:simple-page-master> にページサイズなど指定	@page {...} にページサイズなど指定
ページマージンは <fo:simple-page-master> 内の <fo:region-body> に指定。	@page {...} にページマージンを指定。
柱やノンブルの領域は <fo:simple-page-master> 内の <fo:region-before>, <fo:region-after> など（サイド・リージョンという）で定義する。	柱やノンブルの領域は @page {...} 内の @top-left, @top-center, @top-right, @bottom-left, ... などで定義する。ページヘッダ／フッタの左側/中央/右側がそれぞれ別の領域（マージン・ボックスという）であるところが XSL-FO と違う。
ページヘッダ／フッタの内容（柱やノンブル）は <fo:static-content> 内に記述。	ページヘッダ／フッタの内容（柱やノンブル）はマージン・ボックスの content プロパティの値として指定する。

XSL-FO	CSS
柱の内容は <fo:marker> で設定して、<fo:retrieve-marker> で取り出す。	柱の内容は string-set プロパティで設定して `content: string(X);` で取り出す。（または `position: running(X);` でセットして `content: element(X);` で取り出す。）
ページ番号は <fo:page-number/> で取り出す。	ページ番号は `content: counter(pages);` で取り出す。
<fo:conditional-page-master-reference> で、奇数ページ、偶数ページ、先頭ページ、最終ページ、空白ページといった条件でのページマスターの割り当てをする。	`@page` に付加するページセレクタで、右ページ（`:right`）、左ページ（`:left`）、先頭ページ（`:first`）、空白ページ（`:blank`）のページルールを定義する。
<fo:page-sequence-master> を複数定義して、文書内のパートごとに割り当てることができる。	名前付きページ（例：`@page Appendix {...}`）を定義して、文書内のパートごとに割り当てることができる。

1.6 AH 拡張プロパティ

- 独自拡張および CSS3 ドラフト仕様のプロパティを採用。
- CSS では、AH 拡張を表すプレフィックス -ah- を付ける。

  例：

```
-ah-hanging-punctuation: allow-end; /* 句読点ぶら下げ有り */
```

- XSL-FO では、AH XSL-FO 拡張名前空間のプレフィックスを付ける。

  例：

```
axf:hanging-punctuation="allow-end"
    xmlns:axf="http://www.antennahouse.com/names/XSL/Extension"
```

- AH 拡張プロパティの多くは CSS と XSL-FO で共通のものが使える（上の例など）。
- XSL-FO の標準のプロパティを CSS で AH 拡張プロパティとして利用できるものもある。

  例：

```
-ah-display-align: center; /* ブロック進行方向にセンタリング */
```

Chapter 2. フロート拡張

2.1 CSS2.1 と XSL-FO 標準のフロート機能

これは float: left; の例

AH フロート拡張の説明の前に、まず、CSS2.1 と XSL-FO 標準のフロート機能についておさらい。

これは float: right; の例

CSS2.1 の float

float: none | left | right

XSL-FO の float

float: none | before | start | end | left | right | inside | outside

※before はページの before 側（横書きなら上）にフロート配置。
※start, end は左横書きなら left, right に対応。inside はノド側、outside は小口側。

2.2 AH 拡張 floatプロパティ

- -ah-float: <float-x>||<float-y>||<float-reference>||<float-move>
- <float-x>: none | start | end | left | right | top | bottom | center | inside | outside
- <float-y>: none | before | after | left | right | top | bottom | center | inside | outside
- <float-reference>: normal | page | column | multicol
- <float-move>: auto | next | auto-next | auto-move | keep

※AH 拡張 floatプロパティはXSL-FOとCSSで共通（ここではCSSの構文で説明）。XSL-FOで利用する場合は `<fo:float axf:float="top outside">` のように拡張 floatを指定する。
※これらのキーワードのうち top, bottom, inside, outside, page, multicol, next はCSS3 GCPM (Generated Content for Paged Media)ドラフト仕様で規定されている。
※AH 拡張 floatにおいてx方向とy方向という場合は、x=文字の進む方向、y=行の進む方向。つまり横書きならx=水平方向、y=垂直方向だが、縦書きならx=垂直方向、y=水平方向。
※この仕様は、現時点のV6.0プレリリース版のものなので、製品版では変わる可能性あり。

これは -ah-float: page top; （ページの上に配置）の例

2.3 ページのフロート

ページの上に配置
　　-ah-float: page top;
ページの下に配置
　　-ah-float: page bottom;
ページの左上に配置
　　-ah-float: page left top;
ページの右下に配置
　　-ah-float: page right bottom;
ページの上の小口側
　　-ah-float: page top outside;
ページの下のノド側
　　-ah-float: page bottom inside;

※物理方向 left、right、top、bottom の代わりに論理方向 start（行頭側）、end（行末側）、before（前側）、after（後側）も使用可。

これは -ah-float: page right bottom; (ページの右下に配置) の例

2.4 段のフロート

段の上に配置 -ah-float: column top;

いろはにほへとちりぬるを、わかよたれそ、つねならむ。うゐのおくやまけふこえて、あさきゆめみしゑひもせすん。いろはにほへとちりぬるを、わかよたれそ、つねならむ。うゐのおくやまけふこえて、あさきゆめみしゑひもせすん。いろはにほへとちりぬる

段の下に配置 -ah-float: column bottom;

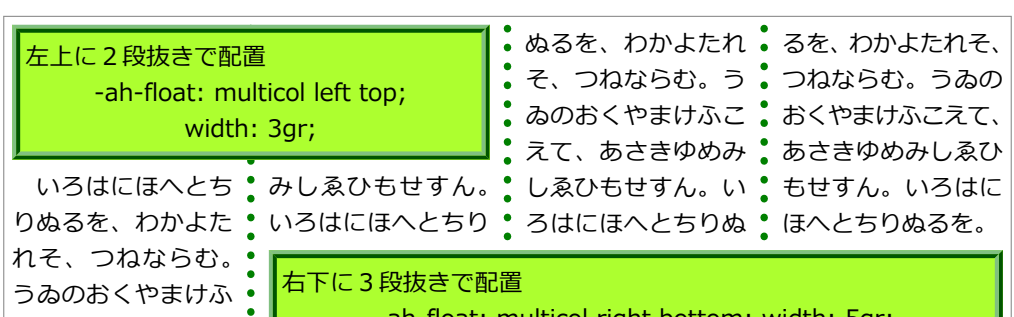
を、わかよたれそ、つねならむ。うゐのおくやまけふこえて、あさきゆめみしゑひもせすん。いろはにほへとちりぬるを、わかよたれそ、つねならむ。うゐのおくやまけふこえて、あさきゆめみしゑひもせすん。いろはにほへとちりぬるを、わかよたれそ、つねならむ。うゐのおくやまけふこえて、あさきゆめみしゑひもせすん。いろはにほへとちりぬるを、わかよたれそ、つねならむ。

段の右上に配置 -ah-float: column right top;

段の左下に配置 -ah-float: column left bottom;

※段をまたがるフロートは次の段組のフロートで。

2.5 段組のフロート



単位 gr(グリッド)について

- 段幅と段間をともに 1gr と数える。例 : width: 1gr は段幅と同じ。2gr は段幅＋段間。3gr は 段幅(1 段目)＋段間＋段幅(2 段目)。n 段抜きは (2n-1)gr。
- 小数点以下の端数は段幅または段間の途中までを表す。例 : width: 1.5gr は段幅*1 ＋段間*0.5。width: 2.5gr は段幅(1 段目)＋段間＋段幅(2 段目)*0.5。

2.6 絶対配置フロートと相対配置フロート

絶対配置フロート

ページ／段／段組のフロートは、フロート指定を埋め込む行位置（アンカーの位置）によらずに、絶対的な位置を基準に配置されるので「絶対配置フロート」と呼ぶことにする。

相対配置フロート

フロート指定を埋め込む行位置（アンカーの位置）を基準に配置されるフロートは「相対配置フロート」と呼ぶことにする。

※JIS X 4051（日本語組版規則）における図の配置の「絶対位置指定による配置」「相対位置指定による配置」に対応する

※y 方向のフロート指定（before/after、横書きでの top/bottom、縦書きでの right/left）があるのが絶対配置フロートで、無いのが相対配置フロート。

※絶対配置フロート（page top left など）は絶対的な位置指定と似ているが、同じ位置への指定のフロートがページ内に複数あった場合は、重なったりはしないで並んで（横書きなら上から下、右縦書きなら右から左に）配置されるので、必ずしも絶対的な位置ではない。

2.7 フロートを次のページ（または段）に移動するかどうかを指定

<float-move>: auto | next | auto-next | auto-move | keep

auto（デフォルト）
　絶対配置フロートでは auto-next、相対配置フロートでは keep と同じ。
next
　フロートを現在のページ（または段）ではなくて次のページ（または段）に配置。
auto-next
　現在のページ（または段）に充分なアキが無い場合にフロートを次のページ（または段）に移動。
auto-move
　現在のページ（または段）に充分なアキが無い場合、フロートを次のページ（または段）に移動、あるいは、フロートを移動するのではなく、フロートのアンカーとまわりのテキストを次のページ（または段）に移動。どちらを次のページ（または段）に移動するかは、JIS X 4051（日本語組版規則）における図の配置方法の規則にしたがう。

keep

フロートとそのアンカーは常に同じページ（または段）になるように配置。現在のページ（または段）にそのための充分なスペースが無い場合は、フロートのアンカーよりも前のところで改ページ（または改段）が起きて空白が生じることになる。

auto-next と auto-move の使い方

ページに図の配置をするとき、本文中での図への参照となるべく同じページに配置したい。それが出来ないとき、通常は図を次のページに送る（auto-next の動作）。

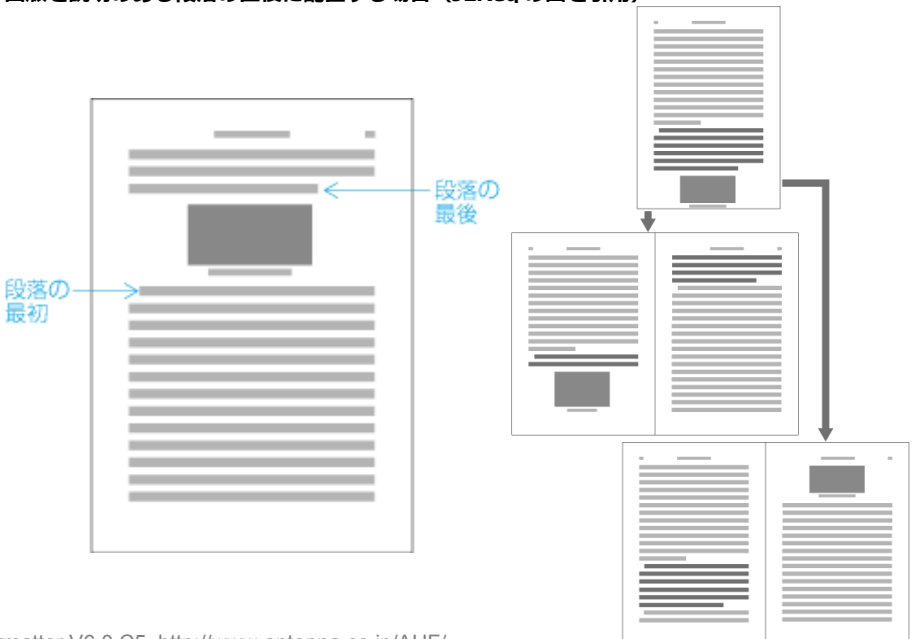
しかし場合によっては、図が前のページにあっても図への参照がその次のページの最初のほうにあるなら許容できる。通常は図を次のページに送るが、図のほうが先に現れることも許容したいときは auto-move を指定。

横組の図版配置の一般的な例——図版を説明のある段落の直後に配置

```
figure { -ah-float: center auto-move; }
```

```
<p>……ここは図を説明してる段落です（図1）。</p>
<figure>
  <img src="図1.svg" />
  <figcaption>図1 キャプション</figcaption>
</figure>
<p>そのあとの段落です……</p>
```

図版を説明のある段落の直後に配置する場合（JLReq の図を引用）



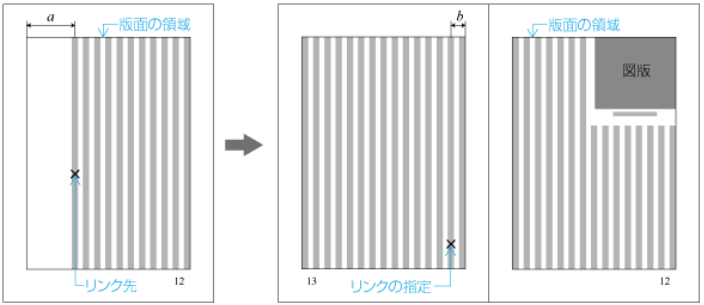
※拡張 float の値 center は、x 方向の中央に寄せて配置。両側へのテキスト回り込みは無し。
※図をそのまま配置すると版面の領域からはみ出すとき、領域内の部分を a、はみ出した部分を b とし、a≧2b の場合（はみ出しが小さい）、図の前のテキストを次のページに送ることで図をページ内に収める。a<2b の場合、図を次のページの先頭に移動し、空いたところには図の後に続くテキストで埋める。（相対配置フロートの auto-move 指定での動作）
※はみ出した図を常に次のページに移動するようにするなら auto-next を指定すること。

縦組の図版配置の一般的な例――“天・小口寄り”に図版を配置

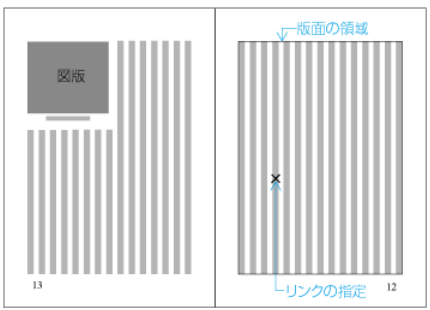
```
:root {
  -ah-writing-mode: tb-rl;                  /* 本文は縦組 */
}
figure {
  -ah-float: page top outside auto-move;  /* ページの天・小口寄りに図版を配置 */
  -ah-writing-mode: lr-tb;                  /* 図とキャプションのブロックは横組 */
}
```

```
<p>……ここは図を説明してる段落です（図1）。</p>
<figure>
  <img src="図1.svg" />
  <figcaption>図1 キャプション</figcaption>
</figure>
<p>そのあとの段落です……</p>
```

縦組の“天・小口寄り”に図版を配置する場合（JLReq の図を引用）



(a≧2b)



(a<2b)

2.8 フロートのさらなる位置指定

-ah-float-offset-x, -ah-float-offset-y で x 方向、y 方向のオフセット指定

例：

```
-ah-float: multicol left top; /* 段組の左上が基準のフロート */
-ah-float-offset-x: 2gr;      /* 2gr右に移動。2段目からの配置となる */
width: 3gr;                   /* 3grの幅(2段目から3段目まで)*/
```

いろはにほへとちりぬるを、わかよたれそ、つねならむ。うゐのおくやまけふこえて、あさきゆめみしゑひもせすん。いろはにほへとちりぬるを、わかよたれそ、つねならむ。うゐのおくやまけふこえて、あさきゆめみしゑひもせすん。いろはにほへとちりぬるを、わかよたれそ、つねならむ。うゐのおくやまけふこえて、あさきゆめみしゑひもせすん。いろはにほへとちりぬるを、わかよたれそ、つねならむ。

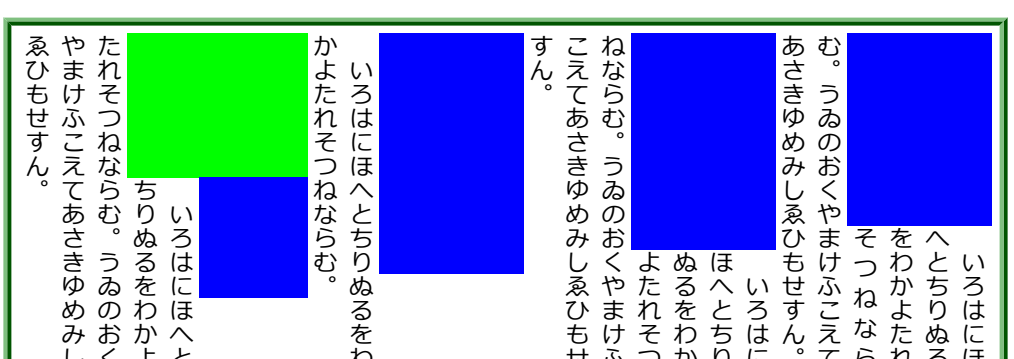
段組の左上が基準で 2 段目から 3 段目までの幅を持つフロート

2.9 フロートとテキスト回り込みの調整

フロートの回り込みテキスト幅の最小値を指定：-ah-float-min-wrap-x

例：

```
/* テキストが回り込む領域の字詰方向の大きさが5文字分未満なら回り込みなしに */
-ah-float-min-wrap-x: 5em;
```



回り込みテキスト幅が足りないならフロートを中央寄せ：-ah-float-centering-x

例：

```
-ah-float: start;
-ah-float-min-wrap-x: 5em;
-ah-float-centering-x: auto; /* 回り込む領域の幅が5em未満なら回り込み無しで中央寄せ */
```

いろはにほへとちりぬるをわかよたれそつねならむ。うゐのおくやまけふこえてあさきゆめみしゑひもせすん。

いろはにほへとちりぬるをわかよたれそつねならむ。

いろはにほへとちりぬるをわかよたれそつねならむ。うゐのおくやまけふこえてあさきゆめみしゑひもせすん。いろはにほへとちりぬるをわかよたれそつねならむ。

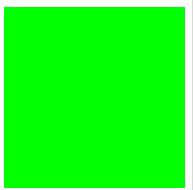
フロートと回り込むテキストとのアキの指定：-ah-float-margin-x

例：

```
-ah-float: outside;   /* 小口寄り(左ページなら左寄せ) */
-ah-float-margin-x: 0.5em;
```

いろはにほへとちりぬるをわかよたれそつねならむ。うゐのおくやまけふこえてあさきゆめみしゑひもせすん。

いろはにほへとちりぬるをわかよたれそつねならむ。うゐのおくやまけふこえてあさきゆめみしゑひもせすん。



いろはにほへとちりぬるをわかよたれそつねならむ。うゐのおくやまけふこえてあさきゆめみしゑひもせすん。いろはにほへとちりぬるをわかよたれそつねならむ。

※このほか、-ah-float-min-wrap-y, -ah-float-centering-y, -ah-float-margin-y もあります。詳しくはマニュアルをご覧ください。

Chapter 3. ルビと圏点のサポート

3.1 ルビ

(X)HTML5 のルビの書き方

モノルビ：

```
<ruby>京<rt>きょう</rt></ruby><ruby>都<rt>と</rt></ruby><ruby>府<rt>ふ</rt></ruby>
```

グループルビ：

```
<ruby>京都府<rt>きょうとふ</rt></ruby>
```

熟語ルビ：

```
<ruby>京<rt>きょう</rt>都<rt>と</rt>府<rt>ふ</rt></ruby>
```

AH XSL-FO 拡張のルビの書き方

モノルビ：

```
<axf:ruby>
  <axf:ruby-base>京</axf:ruby-base><axf:ruby-text>きょう</axf:ruby-text>
</axf:ruby>
<axf:ruby>
  <axf:ruby-base>都</axf:ruby-base><axf:ruby-text>と</axf:ruby-text>
</axf:ruby>
<axf:ruby>
  <axf:ruby-base>府</axf:ruby-base><axf:ruby-text>ふ</axf:ruby-text>
</axf:ruby>
```

グループルビ

```
<axf:ruby>
  <axf:ruby-base>京都府</axf:ruby-base><axf:ruby-text>きょうとふ</axf:ruby-text>
</axf:ruby>
```

熟語ルビ

```
<axf:ruby>
  <axf:ruby-base>京</axf:ruby-base><axf:ruby-text>きょう</axf:ruby-text>
  <axf:ruby-base>都</axf:ruby-base><axf:ruby-text>と</axf:ruby-text>
  <axf:ruby-base>府</axf:ruby-base><axf:ruby-text>ふ</axf:ruby-text>
</axf:ruby>
```

モノルビ、グループルビ、熟語ルビの組版結果の違い

モノルビ：

きょう と ふ
京 都府

グループルビ：

きょうとふ
京都府

熟語ルビ：

きょうと ふ
京都府

熟語ルビの行の折り返し

```
角を<ruby>凝<rt>ぎょう</rt>視<rt>し</rt></ruby>する。
鬼門の方角を<ruby>凝<rt>ぎょう</rt>視<rt>し</rt></ruby>する。
```

角を凝視(ぎょうし)する。

鬼門の方角を凝(ぎょう)

視(し)する。

ルビが親文字よりはみ出した場合の処理

```
<ruby>渚<rt>なぎさ</rt></ruby>に<ruby>暁<rt>あかつき</rt></ruby>を
```

渚(なぎさ) に 暁(あかつき) を

ルビ文字幅を自動的に圧縮

```
-ah-ruby-condense: 66%; /* ルビ文字幅を自動的に66%まで圧縮 */
```

```
<ruby>今<rt>いま</rt></ruby>、<ruby>渚<rt>なぎさ</rt></ruby>に
<ruby>暁<rt>あかつき</rt></ruby>の<ruby>趣<rt>おもむき</rt></ruby>を
```

今(いま)、渚(なぎさ)に暁(あかつき)の趣(おもむき)を

今(いま)、渚(なぎさ)に暁(あかつき)の趣(おもむき)を

親文字の両側にルビ

```
<ruby style="-ah-ruby-position: after;">
  <ruby style="-ah-ruby-position: before;">東南<rt>とうなん</rt></ruby>
  <rt>たつみ</rt>
</ruby>の方向
```

中付きと肩付き

```
<ruby style="-ah-ruby-align: center;">地<rt>ち</rt></ruby>を
```


```
<ruby style="-ah-ruby-align: start;">地<rt>ち</rt></ruby>を
```

グループルビの配置

```
<ruby style="-ah-ruby-align: distribute-space;">紫陽花<rt>あじさい</rt></ruby>
```

あ じ さ い
紫陽花

```
<ruby style="-ah-ruby-align: distribute-letter;">紫陽花<rt>あじさい</rt></ruby>
```

あ じ さ い
紫陽花

```
<ruby style="-ah-ruby-align: center;">境界面<rt>インターフェイス</rt></ruby>
```

インターフェイス
境界面

```
<ruby style="-ah-ruby-align: center; -ah-ruby-base-align: distribute-space;">
境界面<rt>インターフェイス</rt>
</ruby>
```

インターフェイス
境 界 面

```
<ruby style="-ah-ruby-limit-space: none;">なげきの聖母像<rt>ピエタ</rt></ruby>
```

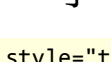
ピ エ タ
なげきの聖母像

```
<ruby style="-ah-ruby-limit-space: 1.0;">なげきの聖母像<rt>ピエタ</rt></ruby>
```

ピ エ タ
なげきの聖母像

ルビの小書きのカナの変換

```
<ruby>一寸<rt>ちょっと</rt></ruby>
```




```
<ruby style="-ah-ruby-small-kana: convert;">一寸<rt>ちょっと</rt></ruby>
```


```
<ruby style="text-transform: fullsize-kana;">一寸<rt>ちょっと</rt></ruby>
```

3.2 圏点

```
em.Kenten {
  -ah-text-emphasis-style: filled;
  -ah-text-emphasis-font-family: KentenGeneric;
  font-style: normal;
}
```

```
ここは<em class="Kenten">圏点で強調</em>よ
```

ここは圏点で強調よ

ここは圏、点で、強、調、よ

```
-ah-text-emphasis-style: open;
```

ここは圏点で強調よ

ここは圏点で強調よ

```
-ah-text-emphasis-style: dot;
```

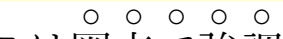
ここは圏点で強調よ

```
-ah-text-emphasis-style: open dot;
```

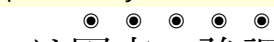
ここは圏点で強調よ

```
-ah-text-emphasis-style: circle;
```


```
-ah-text-emphasis-style: open circle;
```




```
-ah-text-emphasis-style: double-circle;
```

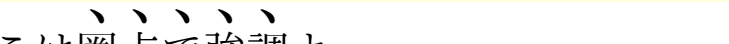



```
-ah-text-emphasis-style: open double-circle;
```

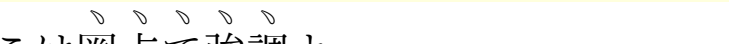
```
-ah-text-emphasis-style: triangle;
```


```
-ah-text-emphasis-style: open triangle;
```

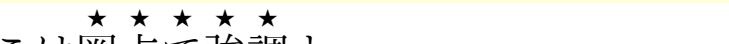

```
-ah-text-emphasis-style: sesame;
```




```
-ah-text-emphasis-style: open sesame;
```



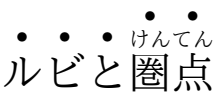

```
-ah-text-emphasis-style: "★";
```



ルビと圏点

```
-ah-text-emphasis-style: dot;

ルビと<ruby>圏点<rt>けんてん</rt></ruby>
```




```
-ah-text-emphasis-style: dot;
-ah-text-emphasis-offset: 0.5em;

ルビと<ruby>圏点<rt>けんてん</rt></ruby>
```

Chapter 4. OpenType フォント機能

4.1 font-variant 拡張

CSS3 Fonts ドラフト仕様の font-variant 拡張の一部に対応しています。

- font-variant: normal | [<font-variant-caps> || <font-variant-numeric> || <font-variant-alternates> || <font-variant-east-asian>]
- <font-variant-caps>: small-caps | all-small-caps | petite-caps | all-petite-caps | titling-caps | unicase
- <font-variant-numeric>: <numeric-figure-values> || <numeric-spacing-values> || <numeric-fraction-values> || slashed-zero
- <numeric-figure-values>: lining-nums | oldstyle-nums
- <numeric-spacing-values>: proportional-nums | tabular-nums
- <numeric-fraction-values>: diagonal-fractions | stacked-fractions
- <font-variant-alternates>: historical-forms | stylistic(<number>) | swash(<number>) | ornament(<number>) | annotation(<number>)
- <font-variant-east-asian>: <east-asian-variant-values> || <east-asian-width-values>
- <east-asian-variant-values>: jis78 | jis83 | jis90 | jis04 | hojo-kanji | nlckanji | simplified | traditional
- <east-asian-width-values>: full-width | proportional-width