

# ページメディアのための CSS 入門

9 June 2022

Tony Graham  
Antenna House, Inc.



# 印刷向けスタイルシートの指定

```
<link rel="stylesheet" type="text/css" media="print" href="foo.css">
```

```
<style type="text/css" media="print">
```

...

```
</style>
```

# 印刷向けスタイルシート

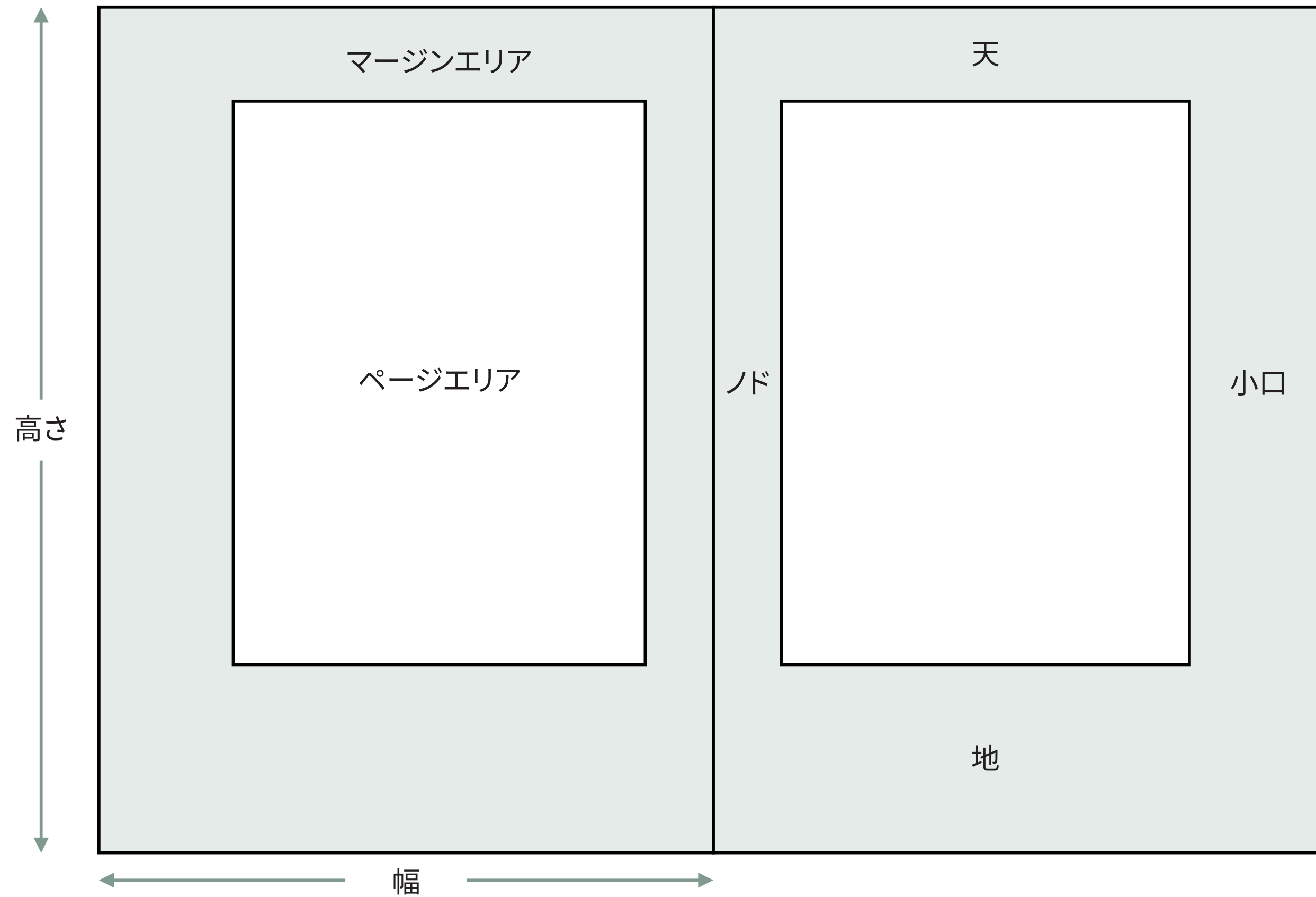
```
@import url("PrintOnly.css") print; /* PrintOnly.css を使った印刷 */

@media print { /* ページメディアに適用 */
  body {
    margin: 0;
    font-size: 10pt }
}

@media screen { /* 画面表示に適用 */
  body {
    margin: 10%;
    font-size: 12px }
}

body { /* すべてのメディアに適用 */
  font-family: sans-serif;
}
```

# ページ



# @page ルール

@page ルールはページサイズ、マージン、マージンボックスを設定します。

```
@page {  
  size: 1920px 1080px; /* 1080p */  
  margin: 0;  
  margin-bottom: 50px;  
}
```

# ページサイズ: size プロパティ

- 幅と高さを指定します。

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

- 定義済みのページサイズを使用します。

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

- ページの向きも設定します。

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

# ページマージン

- margin プロパティのショートハンド :

```
@page {  
  margin: 10%; /* 上下左右のマージン それぞれページ幅の 10% */  
}
```

- 個々の margin プロパティ

```
@page {  
  /* 上下のマージンは 2cm に設定、左右のマージンは 3cm に設定 */  
  margin-top: 2cm;  
  margin-bottom: 2cm;  
  margin-left: 3cm;  
  margin-right: 3cm;  
}
```

# 名前付きページ

```
@page Landscape { /* "Landscape" (横長) 名前付きページ */
  size: A4 landscape;
}
@page Portrait { /* "Portrait" (縦長) 名前付きページ */
  size: A4;
}
table.WideTable {
  page: Landscape; /* "Landscape" 横長ページの横に大きなテーブルを配置 */
}
html {
  page: Portrait; /* "Portrait" 縦長ページをデフォルトとして使用 */
}
```



# 名前付きページ

```
<p>Portrait page</p>
```

```
<table class="WideTable" border="1" style="width:100%">
```

```
<tr>
```

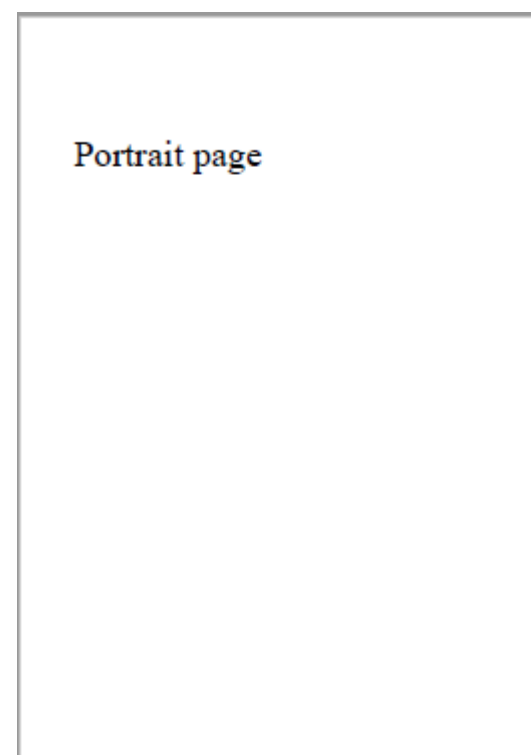
```
<td>1</td>
```

```
...
```

```
<td>18</td>
```

```
</tr>
```

```
</table>
```



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----

# :left, :right, :first ページ

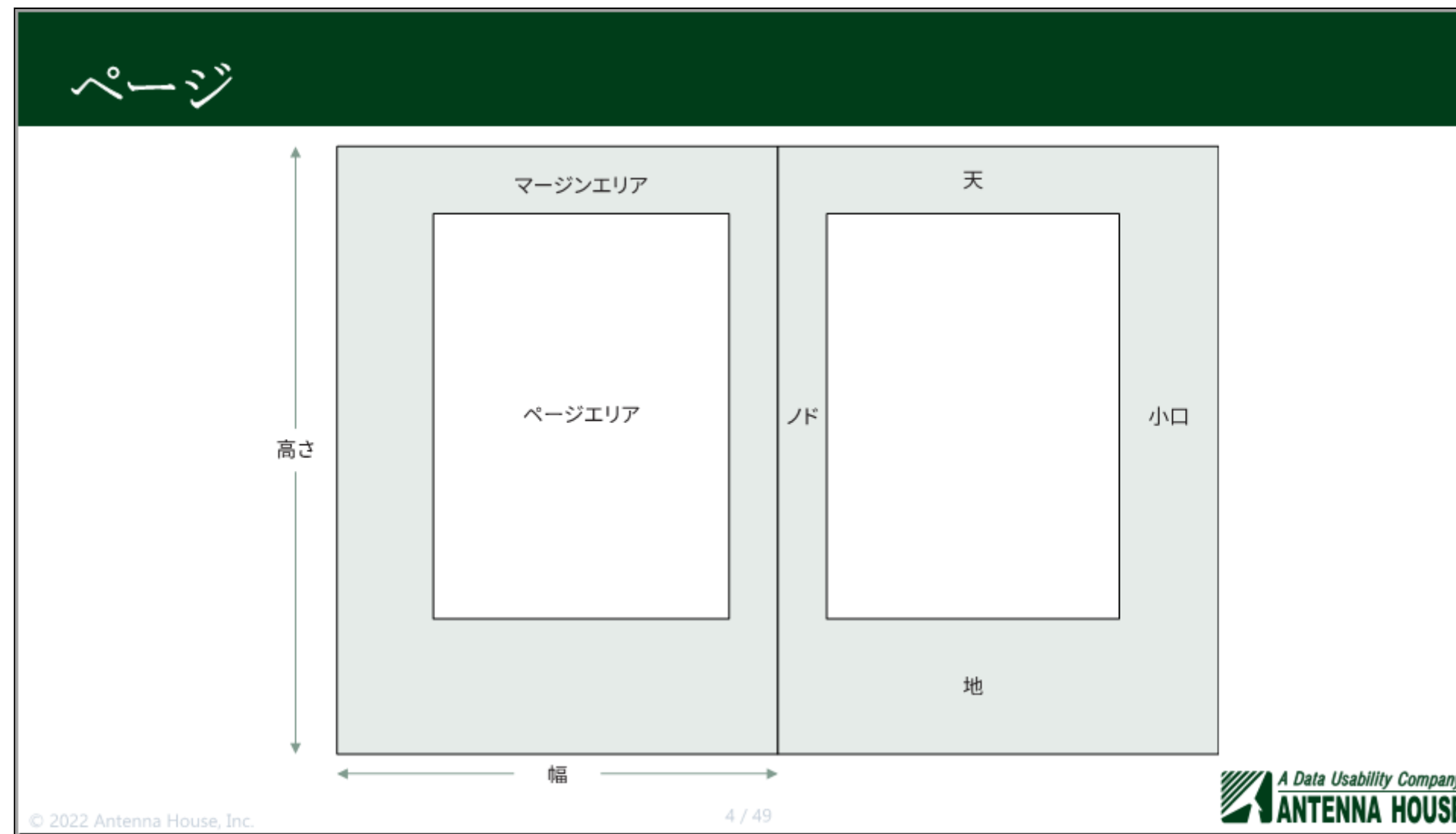
```
@page Chapter:left { /* 左側のページ*/
  margin-left: 23mm;
  margin-right: 27mm;
}

@page Chapter:right { /* 右側のページ*/
  margin-left: 27mm;
  margin-right: 23mm;
}

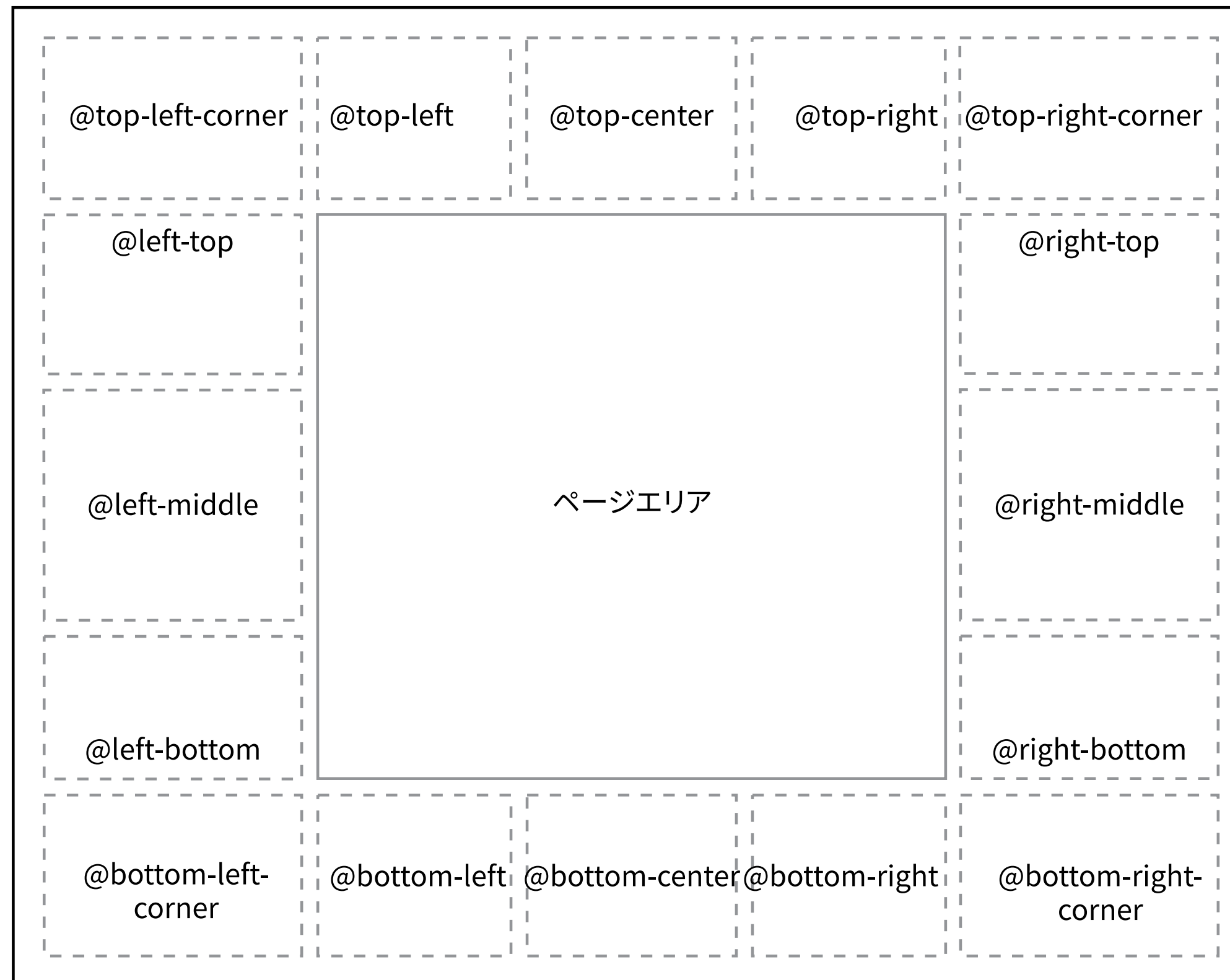
@page Chapter:first { /* 章の最初のページ */
  /* ページヘッダを非表示 */
  @top-right { content: none }
  @top-left { content: none }
}
```

:last と :only は Antenna House Formatter の拡張機能です。

# ヘッダとフッタ



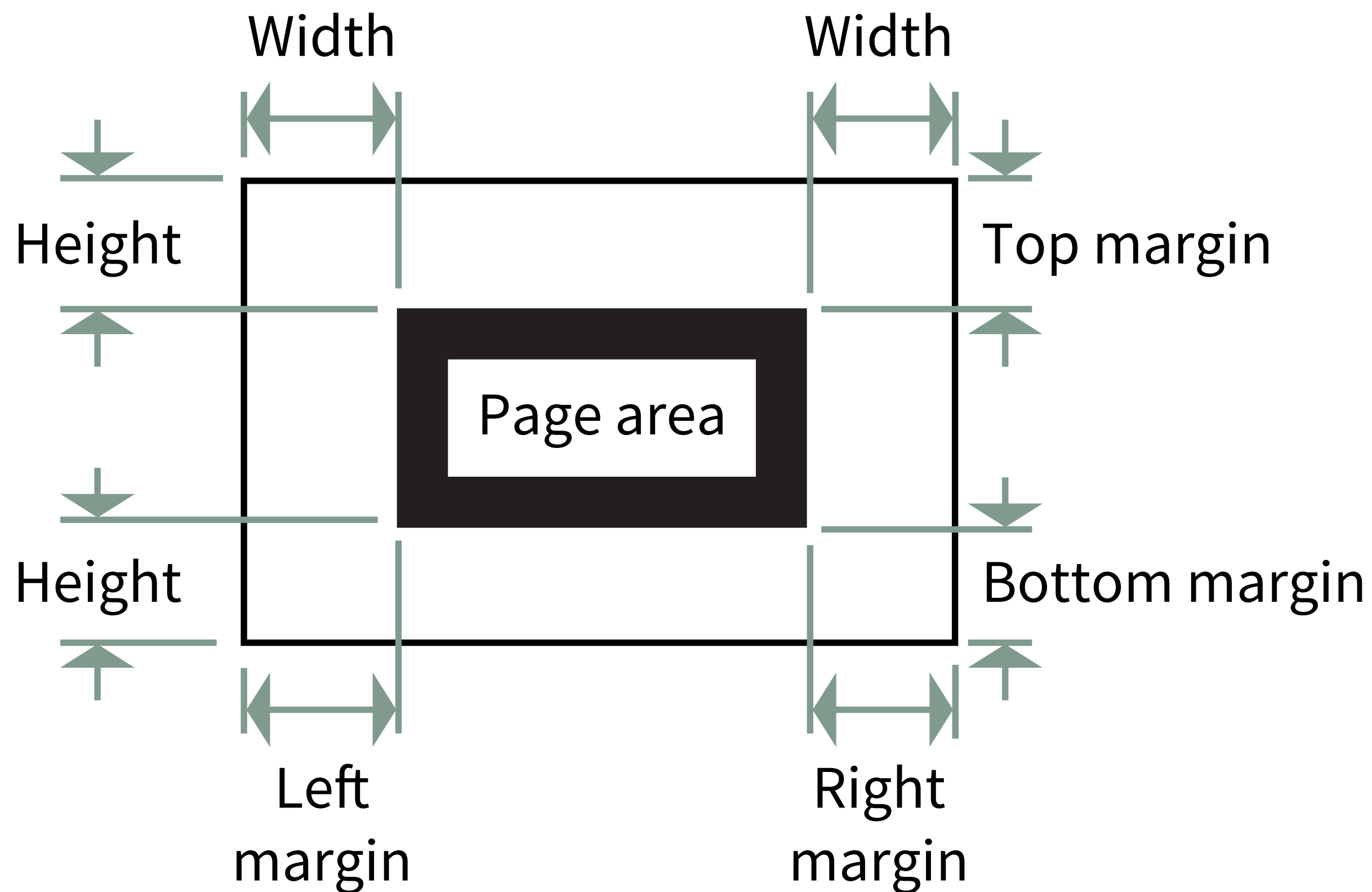
# ページマージンボックス



# ページマージンボックスの寸法

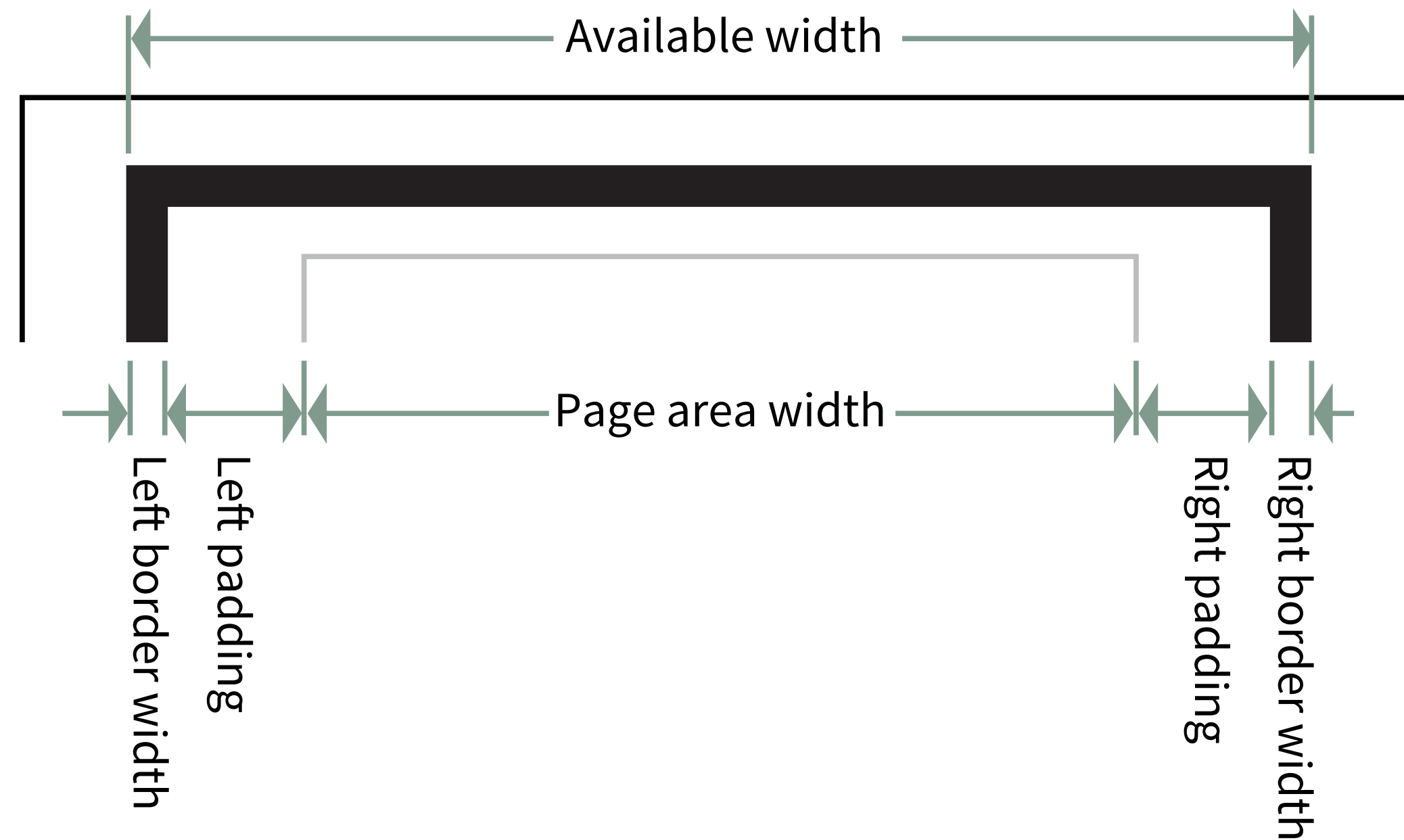
- コンテンツのあるページマージンボックスは「生成」されます。
  - サイズ計算に考慮されます。
- それ以外は `display: none;` と同様です。

# コーナーのページマージンボックス



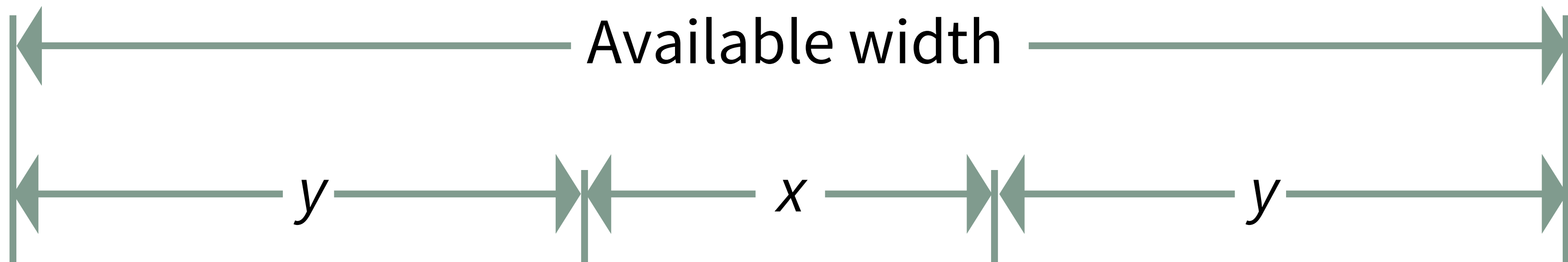
# 上下のページマージンボックス

生成されたページマージンボックス間で分割された使用可能な幅です。



# 中央のページマーージンボックス

中央のページマーージンボックスは、常に中央寄せに配置されます。





# string-set() と string()

文書からマージンボックスに文字列を挿入します。

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

# string-set プロパティ

- 文字列の名前付き変数を作成します。

```
string-set: Chapter content();
```

- マージンボックスの content プロパティを参照します。

```
content: string(Chapter);
```

- string-set のコンテンツ一覧:

- <string>
- <counter()>
- <counters()>
- <content()> – 要素の文字列の値 ::before, ::after、あるいは最初の文字
- attr(<attr-name>) – 属性値
- -ah-attr-from(...) – 祖先要素からの属性

# string() 関数

- 名前付き string を文書にコピーします。

```
@top-right { /* 右側のページヘッダのタイトル。 */  
  content: string(Chapter);  
}
```

- string をページ上で複数回設定する場合：

```
@page Index:right {  
  @top-left {  
    content: string(IndexTerm, first);  
  }  
  
  @top-right {  
    content: string(IndexTerm, last);  
  }  
}
```

# running 要素を挿入する

- running() はフローから要素を取り除きます。

```
.slide:not(#slide0) h1 {  
  position: running(Title);  
  font: 900 150%/1em "Minion Pro", serif;  
  font-weight: 100;  
  font-size: 75px;  
  height: 100px;  
  padding-top: 0;  
  margin: 0;  
  background: transparent;  
  -ah-display-align: center;  
}
```

- element() はマージンボックスに挿入されます。

```
@top-left {  
  content: element(Title);  
}
```

# ページ数

- 現在のページ – counter(page)
- 総ページ数 – counter(pages)

```
@bottom-center {  
  content: counter(page) " / " counter(pages) ;  
  font-size: 0.5em;  
  color: #c6d0db;  
}
```

# 裁ち落としとトンボ

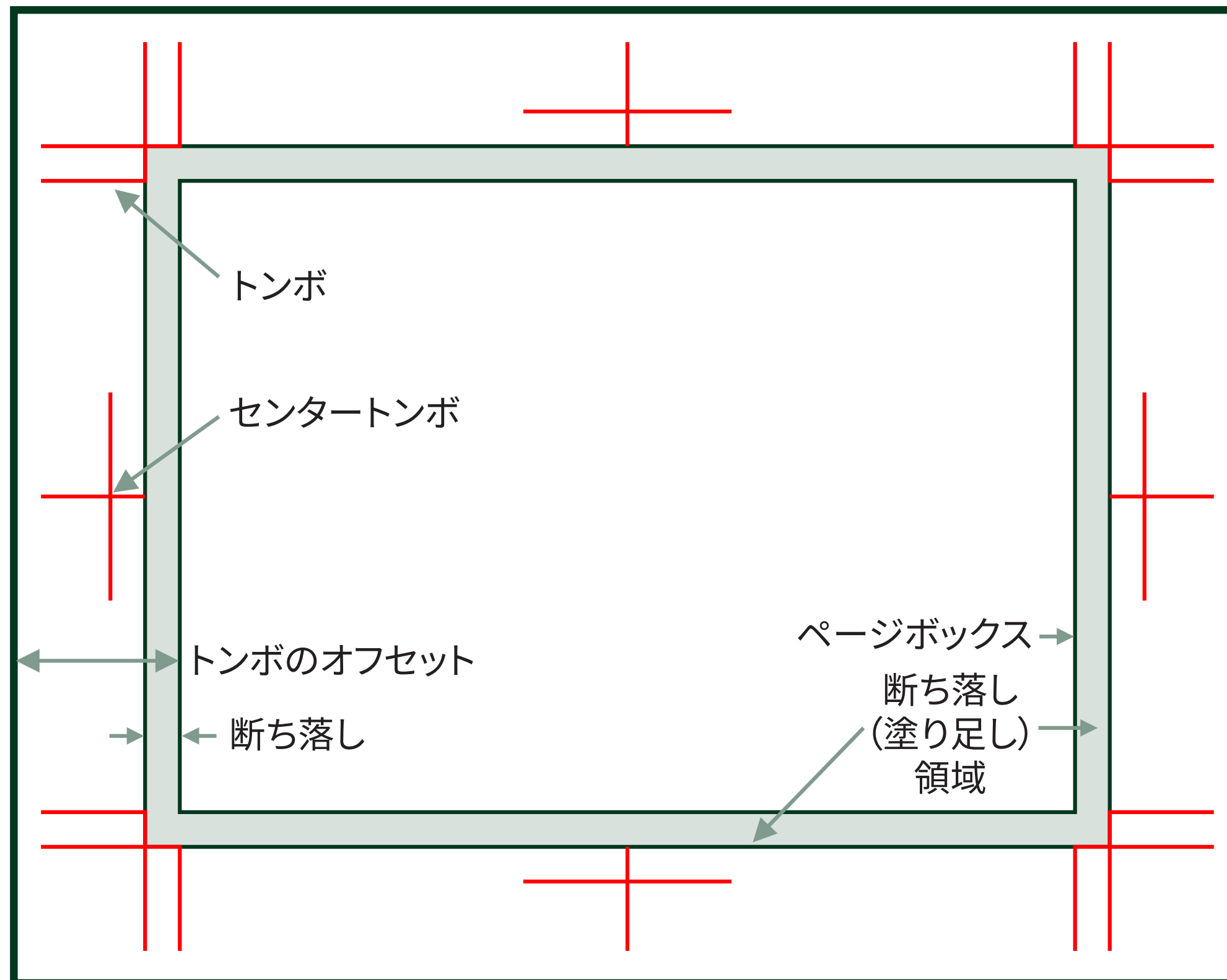
## 索引

cmymk()	61	@counter-style	71
CMYK 色		カスタムカウンタスタイルを定義する	
cmymka()	62	@font-face	35
アルファ要素を持つ CMYK 色		文書内で利用するフォントを追加指定する	
counter()	65	@footnote	79
カウンターの値		脚注領域を記述する	
counter(footnote)	80	@import	2
カウンタは、脚注が生成されるたびに自動的に増加される		別の CSS ファイルを読み込む	
counter(page)	109	@left-bottom	105
カウンタはページが生成されるたびに自動的に増加される		left-bottom 位置のマージンボックス	
counter(pages)	109	@left-middle	105
文書の総ページ数		left-middle 位置のマージンボックス	
device-cmyk()	61	@left-top	105
CMYK 色		left-top 位置のマージンボックス	
element()	108	@media	1
欄外要素をマージンボックスにコピーする		対象メディアタイプを指定する	
leader()	23	@page	95
リーダーを挿入する		ページボックスの基本的な設定 (サイズ、方向、余白など) を行う	
rgb()	61	@right-bottom	105
RGB 色		right-bottom 位置のマージンボックス	
rgb-icc()	62	@right-middle	105
rgb-icc() カラー関数		right-middle 位置のマージンボックス	
rgba()	62	@right-top	105
アルファ要素を持つ RGB 色		right-top 位置のマージンボックス	
running()	108	@sidenote	80
要素に欄外要素として名前を付ける		傍注領域を記述する	
string()	107	@top-center	105
名前付き文字列値を文書にコピーするのに使用		top-center 位置のマージンボックス	
target-counter()	77	@top-left	105
参照先の章番号やページ番号を自動的に付加する		top-left 位置のマージンボックス	
target-text()	77	@top-left-corner	105
参照先のテキストを参照元に表示する		top-left-corner 位置のマージンボックス	
@bottom-center	105	@top-right	105
bottom-center 位置のマージンボックス		top-right 位置のマージンボックス	
@bottom-left	105	@top-right-corner	105
bottom-left 位置のマージンボックス		top-right-corner 位置のマージンボックス	
@bottom-left-corner	105	:last	112
bottom-left-corner 位置のマージンボックス		文書の最後のページのみ一致する疑似クラス	
@bottom-right	105	:left	109
bottom-right 位置のマージンボックス		左ページのみ一致する疑似クラス	
@bottom-right-corner	105	:only	112
bottom-right-corner 位置のマージンボックス		文書の最初と最後のページである単一ページのみ一致する疑似クラス	
bottom-right-corner 位置のマージンボックス		:right	109
		右ページのみ一致する疑似クラス	

## 索引

cmymk()	61	@counter-style	71
CMYK 色		カスタムカウンタスタイルを定義する	
cmymka()	62	@font-face	35
アルファ要素を持つ CMYK 色		文書内で利用するフォントを追加指定する	
counter()	65	@footnote	79
カウンターの値		脚注領域を記述する	
counter(footnote)	80	@import	2
カウンタは、脚注が生成されるたびに自動的に増加される		別の CSS ファイルを読み込む	
counter(page)	109	@left-bottom	105
カウンタはページが生成されるたびに自動的に増加される		left-bottom 位置のマージンボックス	
counter(pages)	109	@left-middle	105
文書の総ページ数		left-middle 位置のマージンボックス	
device-cmyk()	61	@left-top	105
CMYK 色		left-top 位置のマージンボックス	
element()	108	@media	1
欄外要素をマージンボックスにコピーする		対象メディアタイプを指定する	
leader()	23	@page	95
リーダーを挿入する		ページボックスの基本的な設定 (サイズ、方向、余白など) を行う	
rgb()	61	@right-bottom	105
RGB 色		right-bottom 位置のマージンボックス	
rgb-icc()	62	@right-middle	105
rgb-icc() カラー関数		right-middle 位置のマージンボックス	
rgba()	62	@right-top	105
アルファ要素を持つ RGB 色		right-top 位置のマージンボックス	
running()	108	@sidenote	80
要素に欄外要素として名前を付ける		傍注領域を記述する	
string()	107	@top-center	105
名前付き文字列値を文書にコピーするのに使用		top-center 位置のマージンボックス	
target-counter()	77	@top-left	105
参照先の章番号やページ番号を自動的に付加する		top-left 位置のマージンボックス	
target-text()	77	@top-left-corner	105
参照先のテキストを参照元に表示する		top-left-corner 位置のマージンボックス	
@bottom-center	105	@top-right	105
bottom-center 位置のマージンボックス		top-right 位置のマージンボックス	
@bottom-left	105	@top-right-corner	105
bottom-left 位置のマージンボックス		top-right-corner 位置のマージンボックス	
@bottom-left-corner	105	:last	112
bottom-left-corner 位置のマージンボックス		文書の最後のページのみ一致する疑似クラス	
@bottom-right	105	:left	109
bottom-right 位置のマージンボックス		左ページのみ一致する疑似クラス	
@bottom-right-corner	105	:only	112
bottom-right-corner 位置のマージンボックス		文書の最初と最後のページである単一ページのみ一致する疑似クラス	
bottom-right-corner 位置のマージンボックス		:right	109
		右ページのみ一致する疑似クラス	

# 裁ち落としとトンボの用語



# 裁ち落としを作成

margin に負の値を使います。

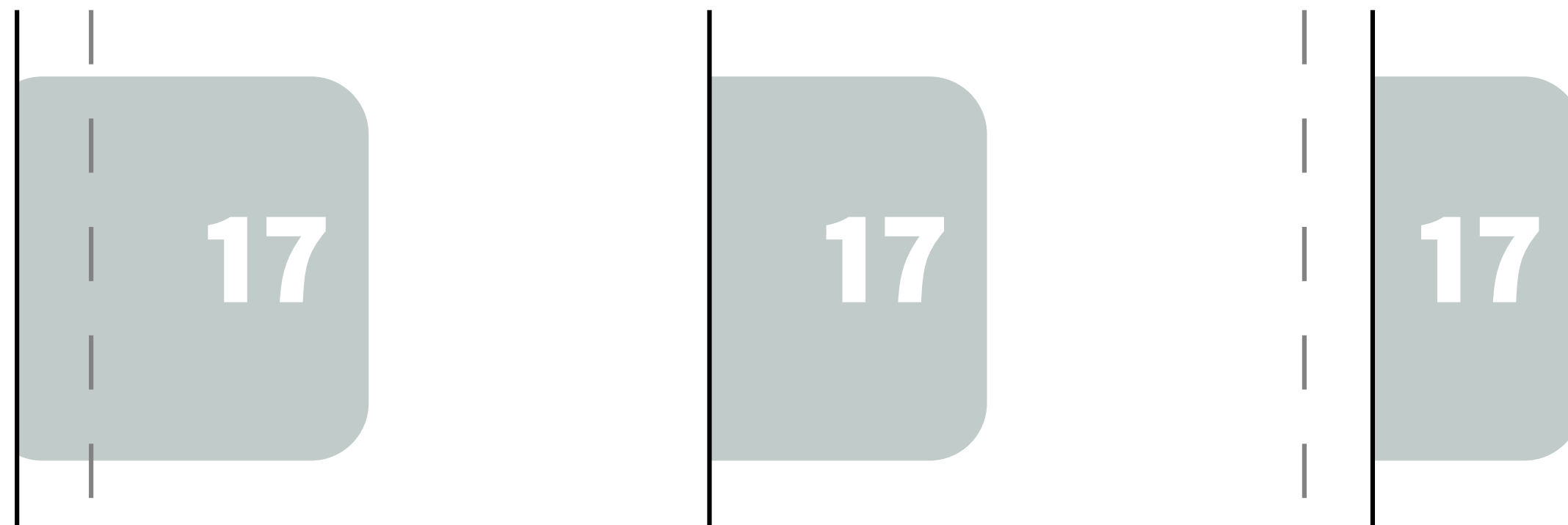
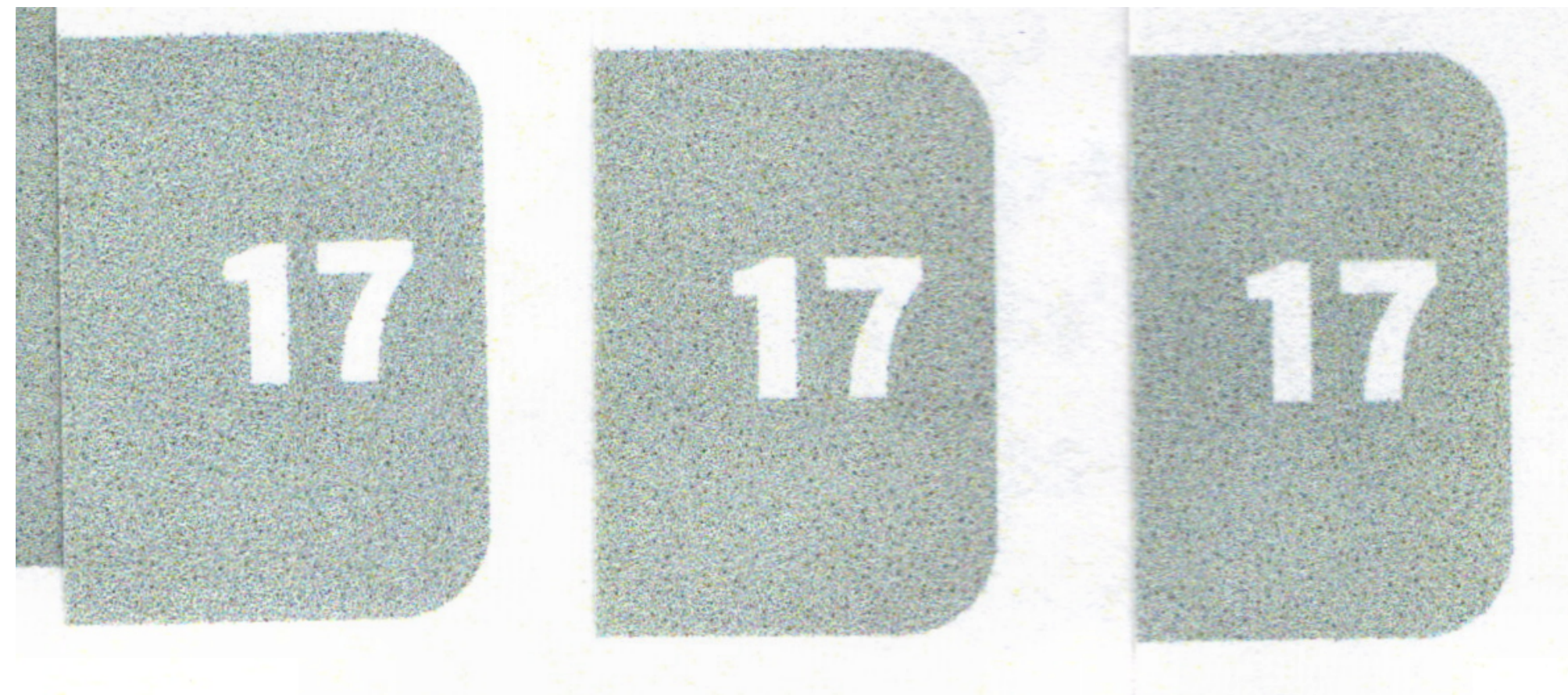
```
@top-left {  
  margin-top: -3mm;  
  padding-top: 3mm;  
  margin-left: -3mm;  
  margin-right: -3mm;  
}
```

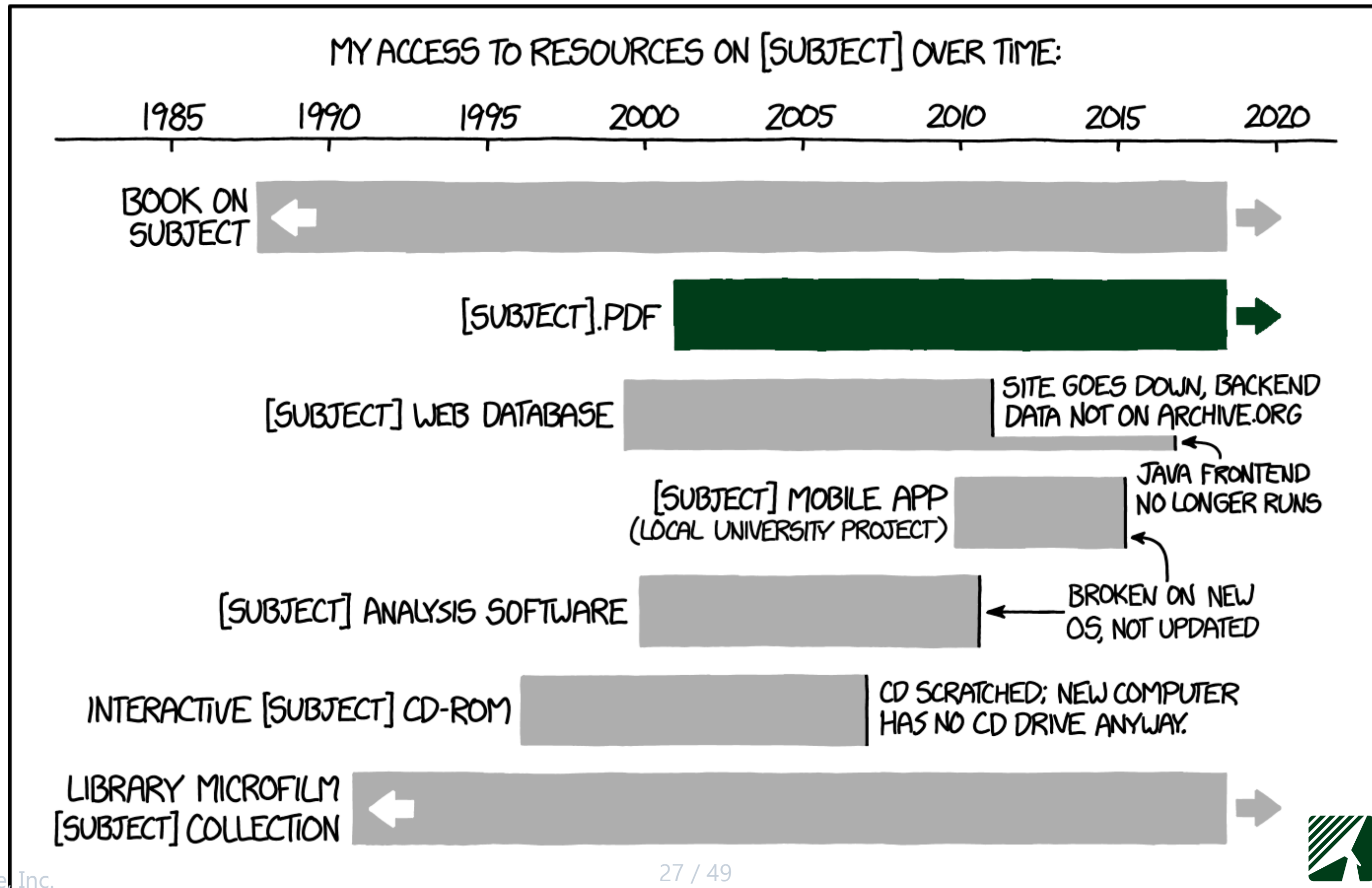


# 裁ち落としを表示

```
bleed: 3mm; /* 裁ち落としの長さ*/  
marks: crop cross; /* トンボを表示 */  
-ah-crop-offset: 14mm; /* ページの端から仕上がり寸法までの距離*/  
-ah-printer-marks-line-color: auto;  
/* トンボの色 */  
-ah-printer-marks-line-length: 10mm;  
/* トンボの線の長さ*/  
-ah-printer-marks-line-width: 0.12mm;  
/* トンボの線の幅 */
```

# 不正確な仕上げを許容





# PDF バージョン

Version	Year	Acrobat Reader version
1.3	2000	4.0
1.4	2001	5.0
1.5	2003	6.0
1.6	2004	7.0
1.7	2008	8
2.0	2017	—

# PDF バリエーション

- PDF/X – 印刷時のデータ交換用の PDF (Exchange)
- PDF/A – 長期保存用の PDF (Archiving)
- PDF/E – 技術文書向けの PDF (Engineering)
- PDF/VT – 可変印刷向けの PDF (Variable Transaction)
- PDF/UA – ユニバーサルアクセシビリティに配慮した PDF (Universal Accessibility)

# タグ付き PDF

- 論理構造に関する追加情報
- 個別の仕様ではない
  - 最初は PDF 1.4 仕様に含まれていた
- テキストとグラフィクスを抽出できる
- アクセシビリティに役立つ

# タグ付き PDF に出力するタグ名をカスタマイズ

```
div.TOC {  
  page-break-before: right;  
  page: TOC;  
  -ah-pdftag: 'Sect';  
}
```

The screenshot shows a PDF editor window titled "Logical Structure - CSS-Print.pdf". The left pane displays a "Structure Elements" tree with a hierarchy: Document > Div > Div > Div > Sect > Sect > H2 (selected) > L. The right pane shows a "Page View" of a "Table of Contents" page. The page content includes a title "Table of Contents" and a list of chapters and sections with their corresponding page numbers. The page number "31 / 49" is visible at the bottom left, and "Page 7" is at the bottom center.

# PDF/UA – ISO 14289-1

- PDF のアクセシビリティを向上させることを目的とする
- コンテンツは論理的な読み順にタグ付けされなければならない
- 文書の言語を指定しなければならない
- 意味のある画像や注釈、数式には代替テキストを含めなければならない
- 支援技術はコンテンツにアクセスさせなければならない
- しおりを含めることが推奨される
- 注釈やフォームやリンク、マルチメディアを含めてもよい
- フォントはすべて埋め込まれなければならない



# Matterhorn Protocol (マッターホルンプロトコル)

- PDF 協会によって発行
- PDF が PDF/UA に準拠しない可能性のあるすべての方法のチェックリスト
- 136 の失敗条件から成る 31 のチェックポイントで構成
- 人の手によるレビューが必要な場合もある

## Checkpoint 15: Tables

Index	Failure Condition	Section	Type	How	See
15-001	A row has a header cell, but that header cell is not tagged as a header.	UA1:7.5-1	Object	Human	-
15-002	A column has a header cell, but that header cell is not tagged as a header.	UA1:7.5-1	Object	Human	-
15-003	In a table not organized with Headers attributes and IDs, a TH cell does not contain a Scope attribute.	UA1:7.5-2	Object	Machine	-
15-004	Content is tagged as a table for information that is not organized in rows and columns.	UA1:7.5-3	Object	Human	-
15-005	A given cell's header cannot be unambiguously determined.	UA1:7.5-2	Object	Human	01-006

# 文書のプロパティ

```
<meta name="document-title" content="The document title" />  
<meta name="subject" content="The document subject" />  
<meta name="author" content="The author" />  
<meta name="keywords" content="Comma, separated, keywords, list" />
```

## CSS ページ媒体入門

文書のプロパティ

概要 セキュリティ フォント 開き方 カスタム 詳細設定

概要

ファイル: CSS-Print-ja.pdf

タイトル: CSSページ媒体入門

作成者: アンテナハウス株式会社

サブタイトル: CSS formatting

キーワード: CSS, formatting

作成日: 2018/06/27 17:36:25

更新日: 2018/06/27 17:36:25

アプリケーション: AH Formatter V6.5 MR6 for Windows (x64) : 6.5.9.33760 (2018/06/26 16:39JST)

その他のメタデータ(M)...

詳細情報

PDF 変換: Antenna House PDF Output Library 6.5.1299 (Windows (x64))

PDF のバージョン: 1.7 (Acrobat 8.x)

場所: D:\AHF\参考書籍+Tutorial\CSSPrintTutorial\4th\画像作成\

ファイルサイズ: 2.27 MB (2,384,427 バイト)

ページサイズ: 210 x 285 mm

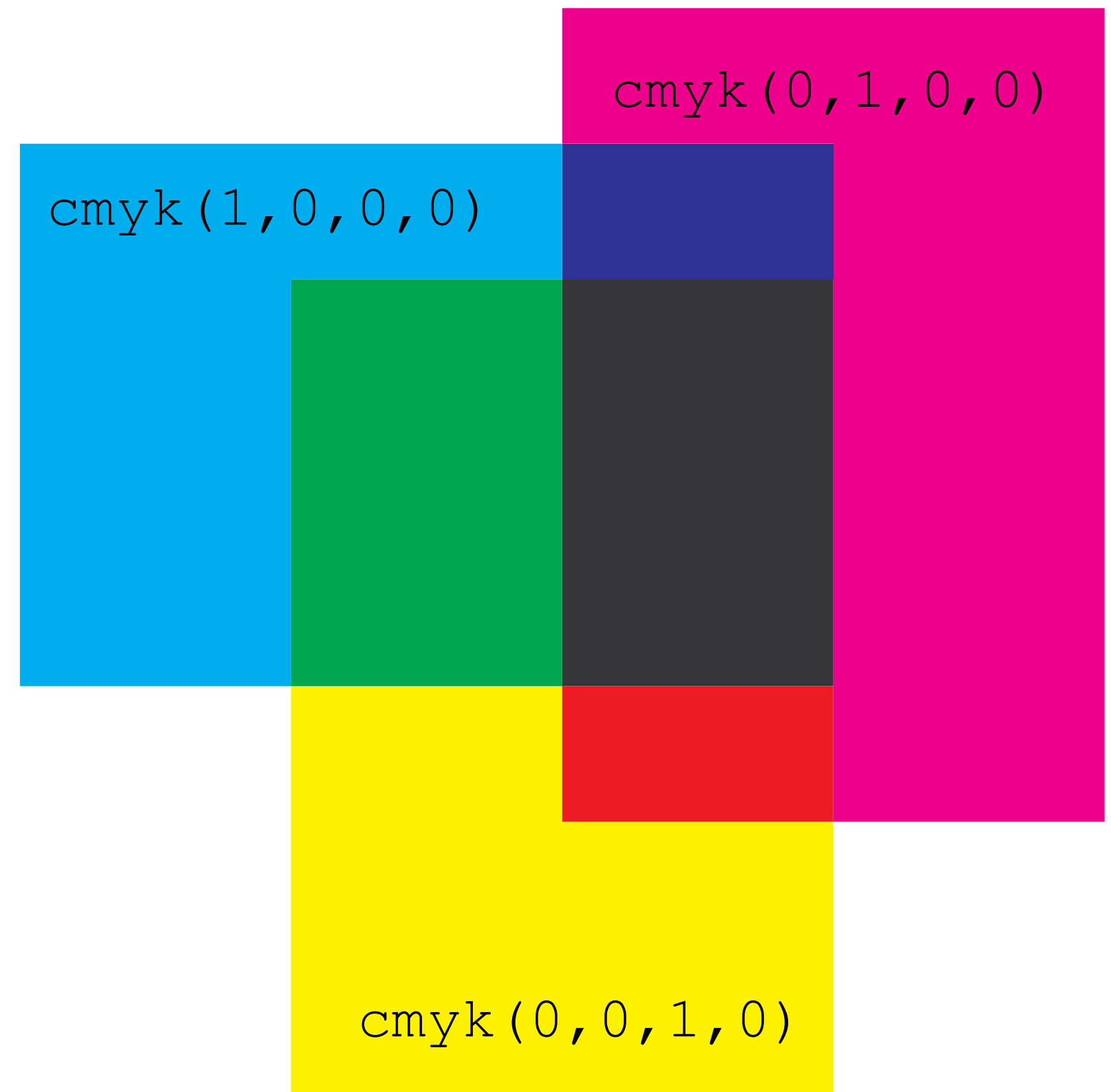
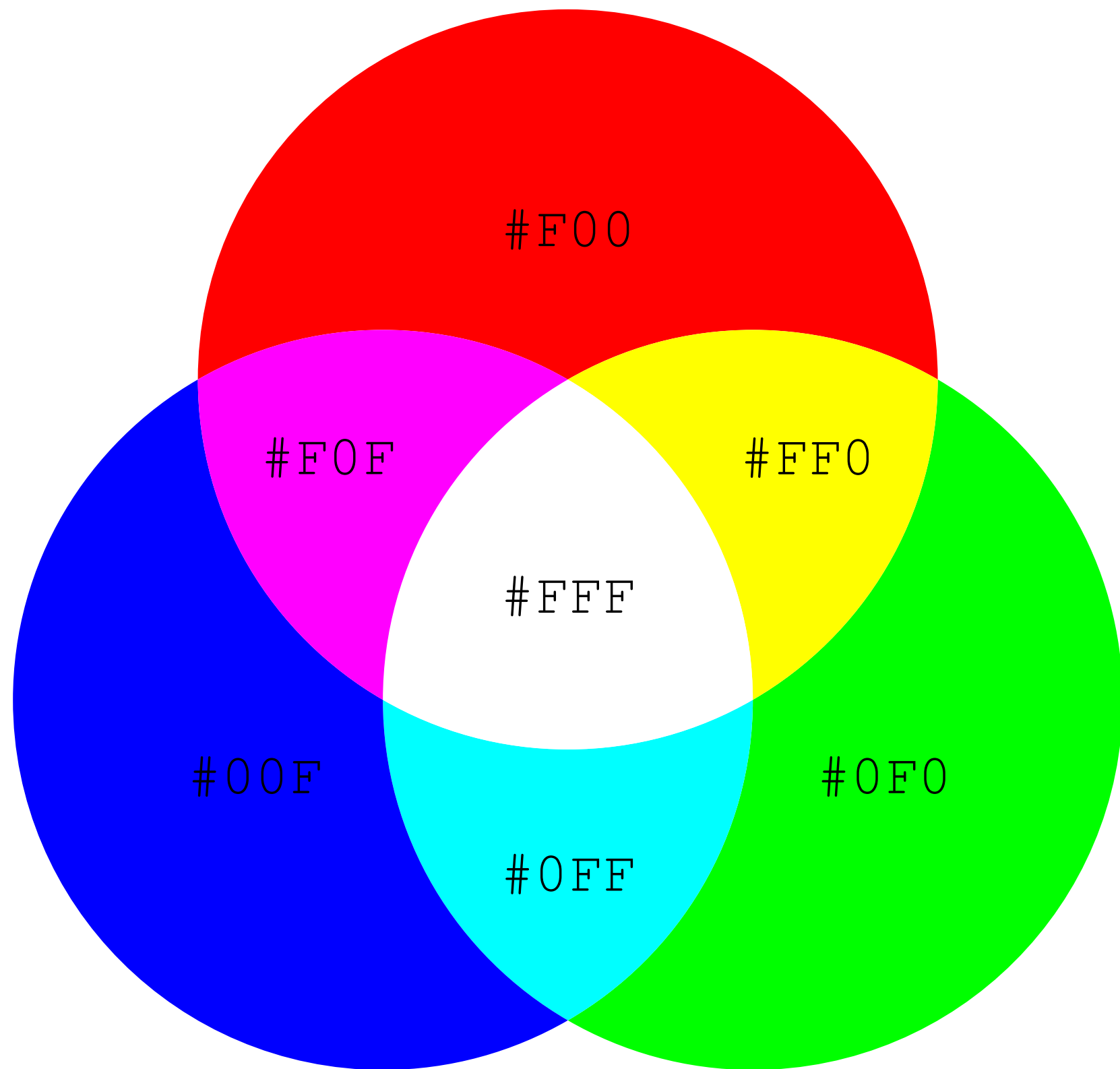
ページ数: 142

タグ付き PDF: はい

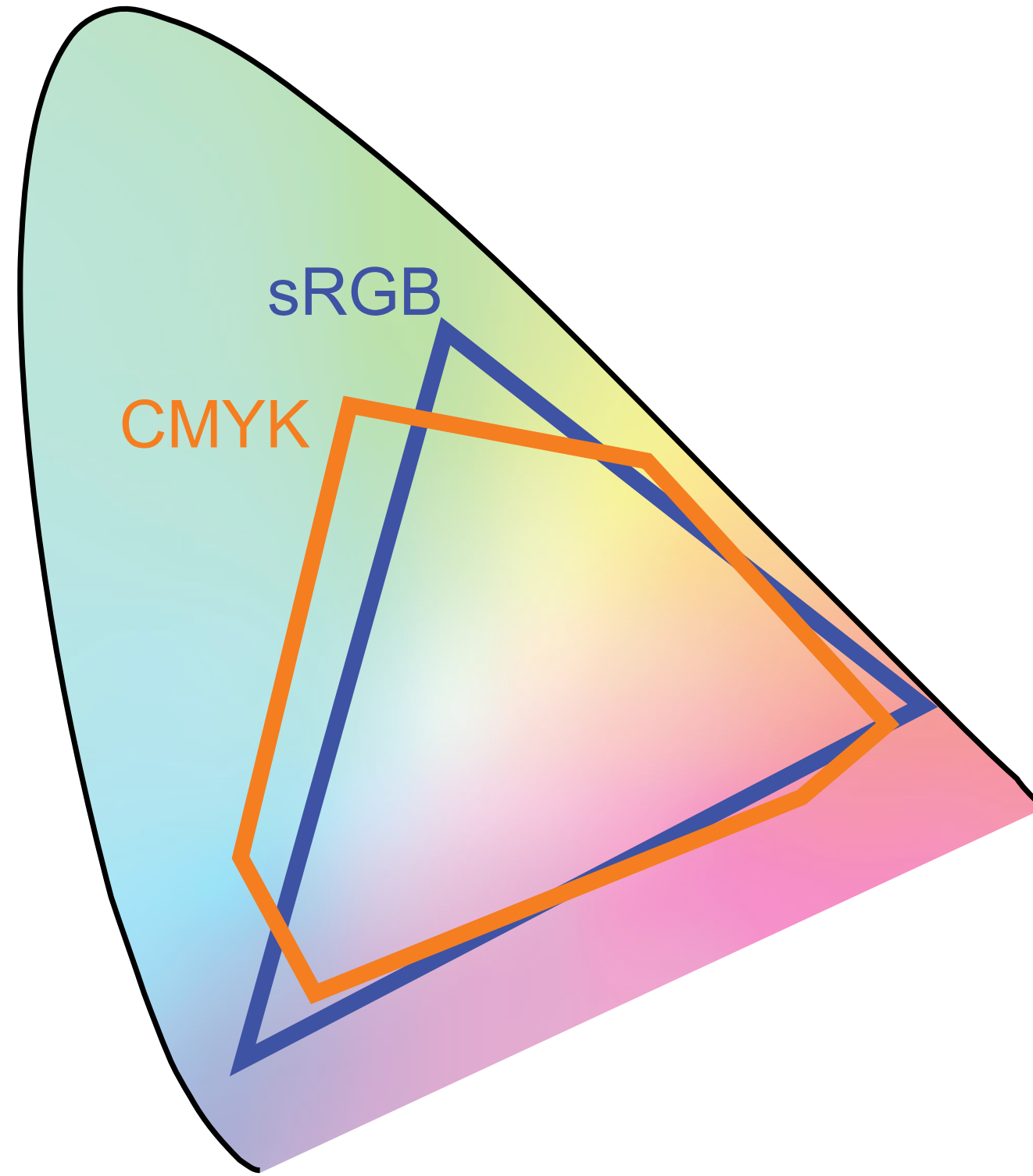
Web 表示用に最適化: いいえ

ヘルプ OK キャンセル

# 色



# 色



# color

```
em { color: green; } /* 色キーワード*/
em { color: #042; } /* #RGB */
em { color: #003D19; } /* #RRGGBB */
em { color: rgb(0, 61, 25); } /* 整数の範囲 0-255 */
em { color: rgb(0%, 24%, 10%); } /* 0%-100% */
em { color: #0428; } /* #RGBA */
em { color: #003D1988; } /* #RRGGBBAA */
em { color: rgb-icc(#Grayscale, 0.5); } /* 0.0 (黒) から 1.0 (白) */
em { color: cmyk(.9, 0, .75, .83); } /* 0.0-1.0 */
em { color: cmyk(90%, 0%, 75%, 83%); } /* 0%-100% */
em { color: rgb-icc(#CMYK, .9, 0, .75, .83); } /* プロファイル依存 */
em { color: rgb-icc(#Separation, 'PANTONE 627 PC', 1.0); } /* 名前と色合い */
em { color: rgb-icc(0, 61, 25, #Separation, 'PANTONE 627 PC', 1, 90%, 0%, 75%, 83%); } /* RGB と CMYK の両方と等価 */
em { color: rgb-icc(#Registration); } /* すべてのセパレーション*/
```

# PANTONE® スポットカラー

```
em { color: rgb-icc(#Separation, 'PANTONE 627 PC', 1.0); }  
/* 名前と色合い*/  
em { color: rgb-icc(#Separation, 'PANTONE 627 PC', 0.5); }  
/* 色合い 0.0 から 1.0*/  
em { color: rgb-icc(#Separation, 'PANTONE 627 PC', 50%); }  
/* 色合い 0% から 100%*/  
em { color: rgb-icc(#Separation, 'PANTONE 627 PC'); }  
/* 色合い 1.0 と想定*/  
em { color: rgb-icc(#Separation, 'PANTONE 627 PC',  
1, 90%, 0%, 75%, 83%); } /* CMYK と等価 */  
em { color: rgb-icc(0, 61, 25, #Separation, 'PANTONE 627 PC'); }  
/* RGG と等価*/  
em { color: rgb-icc(0, 61, 25, #Separation, 'PANTONE 627 PC', 1,  
90%, 0%, 75%, 83%); }  
/* RGB と CMYK の両方と等価*/
```

0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

# セパレーションプレート

## PANTONE® スポットカラー

```
em [ color: rgb-icc(#Separation, 'PANTONE 627 PC', 1.0); ] /* 名前と色合い*/  
em [ color: rgb-icc(#Separation, 'PANTONE 627 PC', 0.5); ] /* 色合い 0.0 から 1.0*/  
em [ color: rgb-icc(#Separation, 'PANTONE 627 PC', 50%); ] /* 色合い 0% から 100%*/  
em [ color: rgb-icc(#Separation, 'PANTONE 627 PC'); ] /* 色合い 1.0 と想定*/  
em [ color: rgb-icc(#Separation, 'PANTONE 627 PC', 1, 90%, 0%, 75%, 83%); ] /* CMYK equivalent. CMY*/  
em [ color: rgb-icc(0, 61, 25, #Separation, 'PANTONE 627 PC'); ] /* RRG と等価*/  
em [ color: rgb-icc(0, 61, 25, #Separation, 'PANTONE 627 PC', 1, 90%, 0%, 75%, 83%); ] /* RGB と CMYK の両方と等価*/
```

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

# 画面とページの相違

- 分割
- フロート
- ナビゲーション
- 左右のページ
- 印刷本



# 分割

CSS is widely used in browsers, editors, and other applications. CSS is used not only for “Web design” but also as the stylesheet specification for a wide range of printing applications as well as for electronic paged media delivered as PDF.

CSS 2.1 (and CSS 2.2) provides only minimal support for paged media output, and its page layout features are not powerful enough. CSS 3, although still under development by the

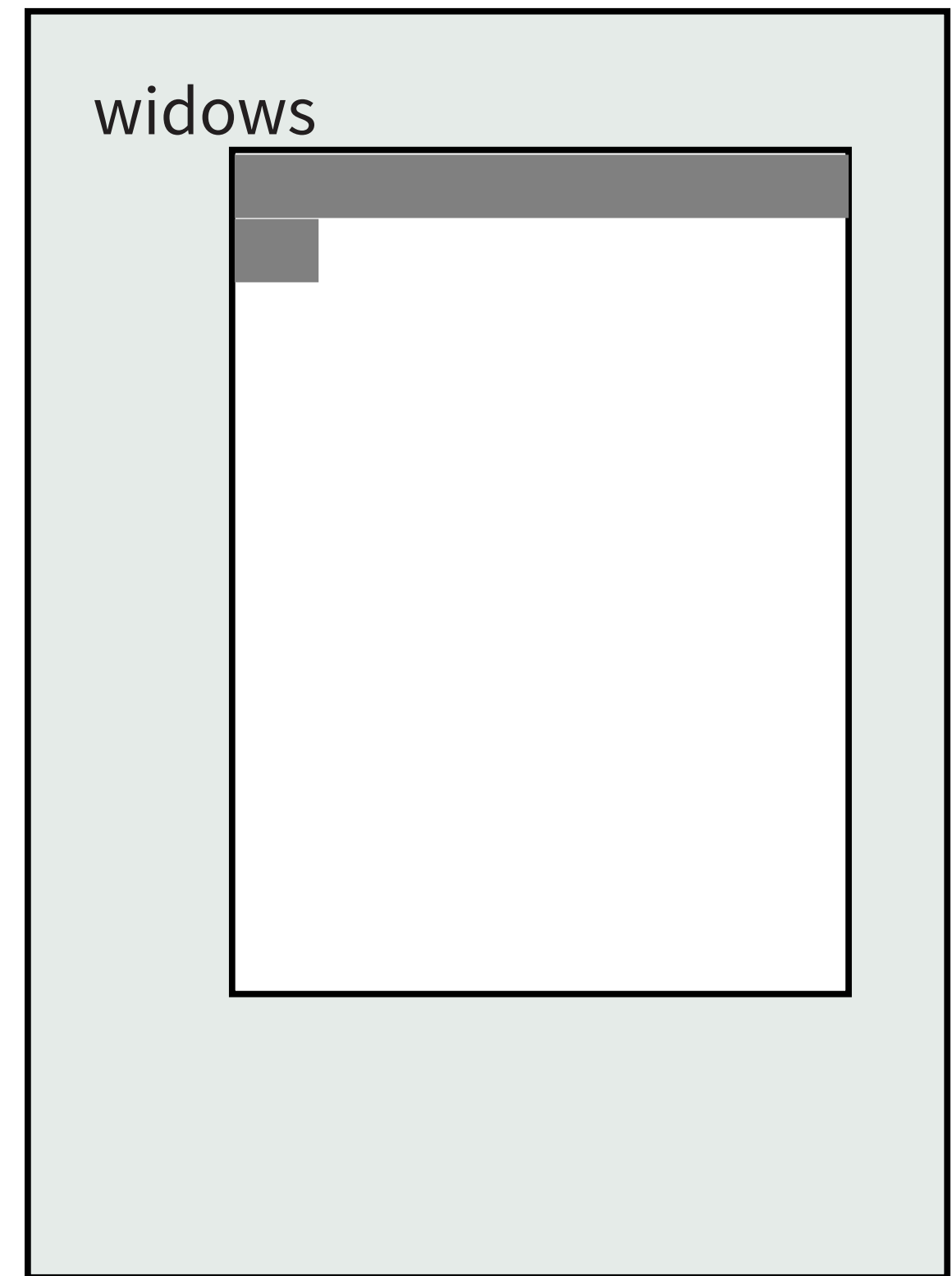
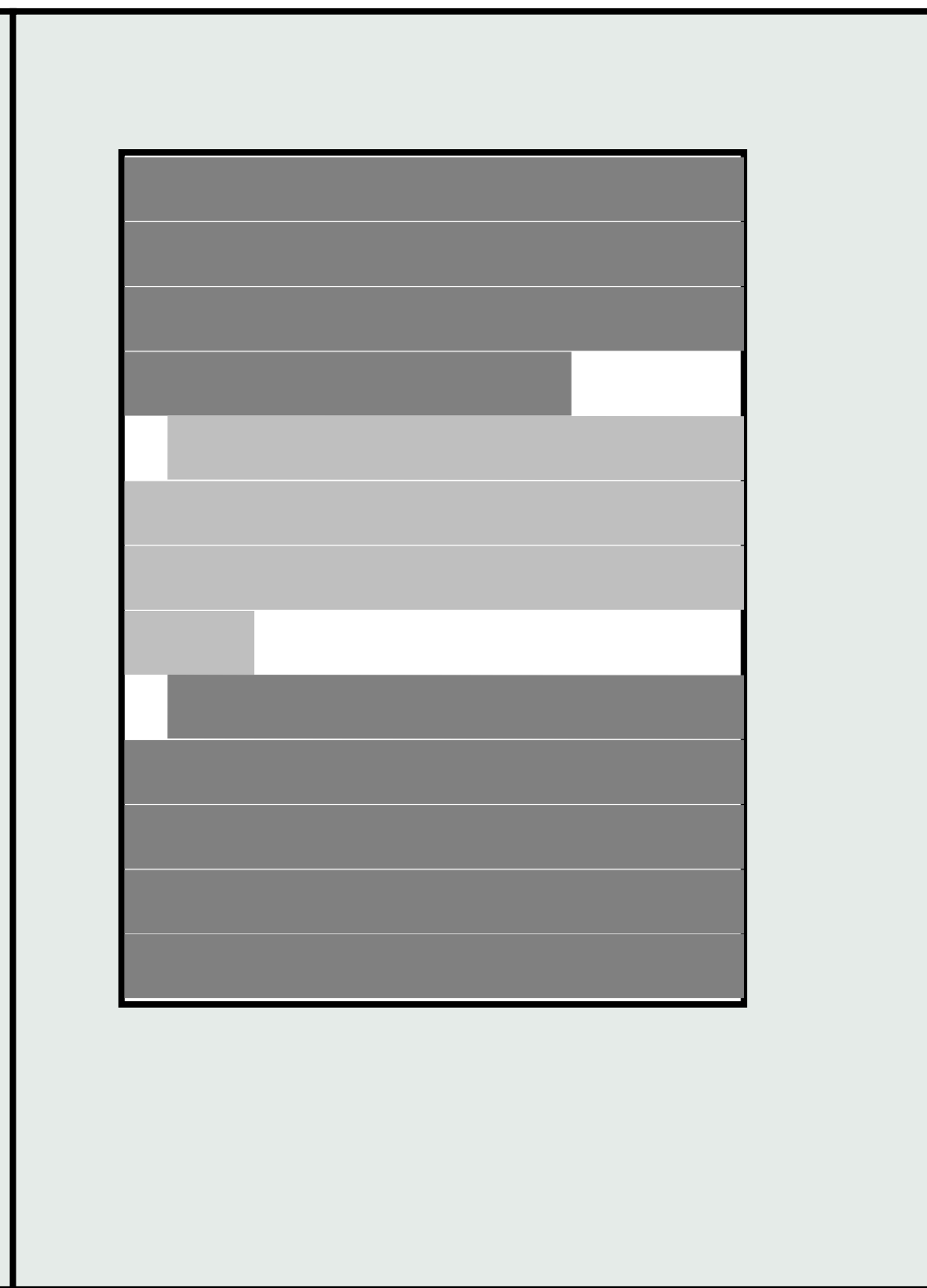
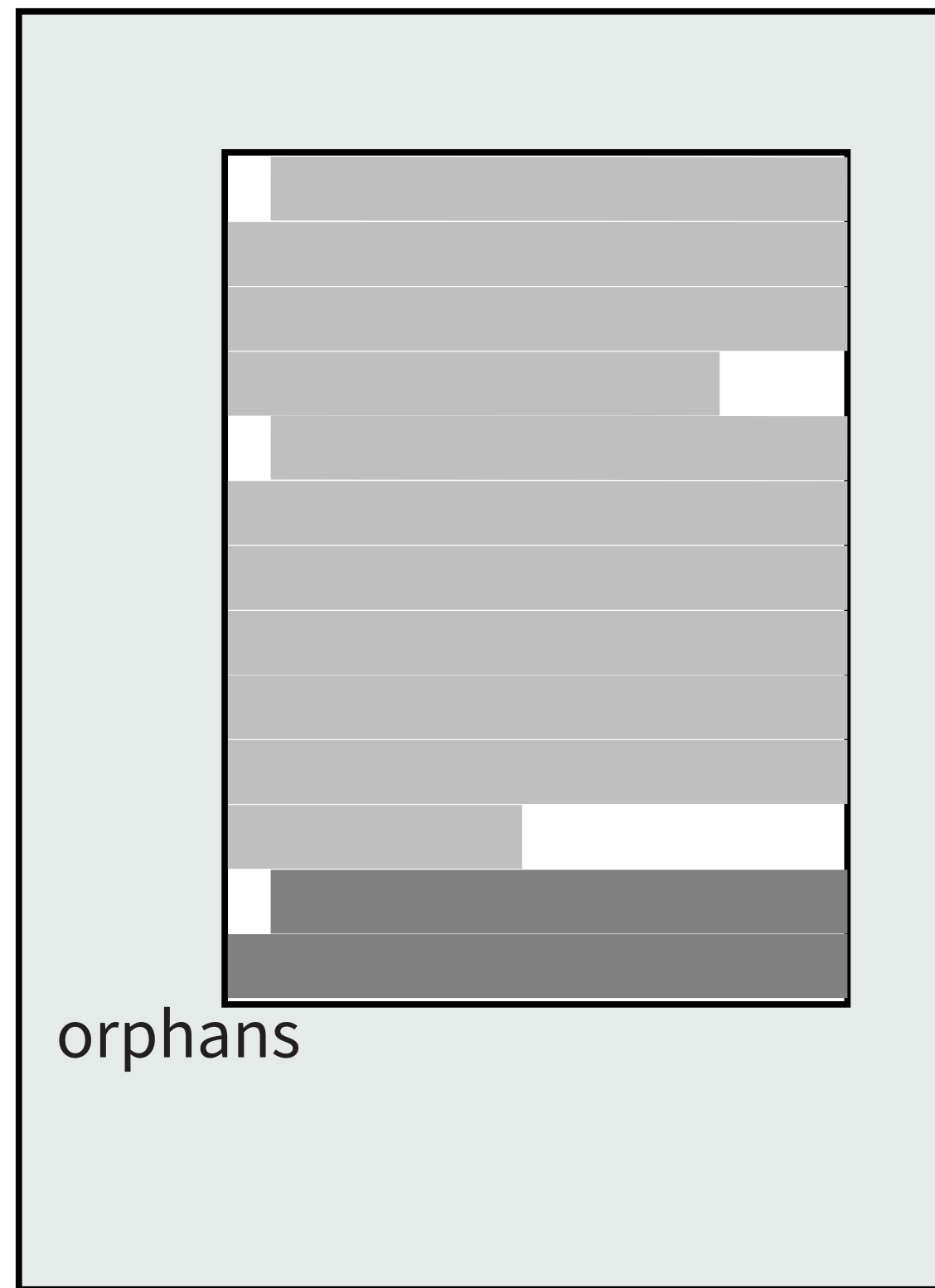
Word

Line

Column

W3C, defines many of the features that are necessary for professional quality formatting, including: advanced page layouts; multiple columns; vertical writing; hyphenation; and multilingual character layout. Antenna House Formatter provides additional features for optimal formatting, including: custom-developed MathML 3 and SVG rendering; baseline grids; PANTONE® spot colors; and additional properties for controlling Japanese layout.

# ページでも分割



# ナビゲーション

- ヘッダとフッタ

Headers and Footers | 107

- 目次

Introduction

Chapter 1. Web and Paged Media	1
1.1 @media Rule .....	1
1.2 Specifying a Print Style Sheet .....	1

- インデックス

cmyk() .....	61	@bottom-left .....	103
Functional notation for a CMYK color		box filling the bottom page margin between the	
cmyka() .....	62	bottom-left-corner and bottom-center page-	

# 左右のページ

## 第19章 ヘッダとフッタ

### 19.1 マージンボックス

ページヘッダやページフッタは、ページの周辺のマージンボックスと呼ばれるエリアに割り当てます。

マージンボックスは、@top-left-corner, @top-left, @top-center, @top-right, @top-right-corner, @left-top, @left-middle, @left-bottom, @right-top, @right-middle, @right-bottom, @bottom-left-corner, @bottom-left, @bottom-center, @bottom-right, @bottom-right-cornerのように、ページ周辺の位置により名前がつけられています。



各マージンボックスの位置

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

## 19.2 柱とノンブル

### 19.2.1 柱の設定: string-set プロパティと string() 機能

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できます。

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

### 19.2.2 文字列変数: string-set プロパティ

- 初期値: none ●適用対象: 疑似要素を除くすべての要素 ●値の継承: しない
- 文字列を名前付きの変数にするには、string-set プロパティを使います。string-set の値は変数名とその変数に格納するコンテンツリストのペアになります。変数の値 (内容) は、柱 (ランニングヘッダ) で参照されます。コンテンツリストは、任意の順序で、以下のいずれかひとつ以上になります。
- <string>: 文字列です。
  - <counter(>: counter() の関数です。第 12 章 カウンタ (ページ 63) を参照してください。
  - <counters(>: counters() の関数です。
  - <content(>: content() の関数です。指定可能な値は以下のとおり:
    - content(): 要素の文字列値。
    - content(before) : :before 疑似要素の文字列値。
    - content(after) : :after 疑似要素の文字列値。
    - content(first-letter) : 要素の最初の文字。
  - attr(<attr-name>): <attr-name> 属性の文字列値。
  - -ah-attr-from(<from-name>, <attr-name> <type-or-unit>?, <fallback> ?): 先祖 <from-name> 要素の属性 <attr-name> の文字列値

```
h1 {
  /* Chapter content(before) content()に
  h1:before と h1 自身の内容をセットします*/
  string-set: Chapter content(before) content();
}
```

### 19.2.3 string()

名前付き文字列 (変数) の値を文書にコピーするのに使用します。

string-set で定義した変数は、例えば、柱 (ランニングヘッダ) に content: string(Chapter); のように参照されます。必要とされる最初の値は名前付き文字列です。

```
@top-right {
  /* 右側ページのタイトル*/
  content: string(Chapter);
}
```

ひとつのページの複数要素がそれぞれの同名の名前付き文字列の値を設定する場合、名前付き文字列はそのページに複数の値を持つことがあります。string() の 2 番目の引数で使用可能な値を指定もできます。

- start: そのページの名前付き文字列のエントリ値 (前のページの最後の有効な指定) を使用します。
- first: ページに値の割り当てがある場合は、最初の割り当て値を使用し、それ以外の場合は、エントリ値を使用します。
- last: そのページの名前付き文字列の退出値 (現在のページの最後の有効な指定) を使用します。
- first-exception: first と似ていますが、値が割り当てられているページに空の文字列が使用されます。例えば、これは、章の先頭ページのヘッダに章のタイトルを繰り返さないようにするために使用します。

```
@page Index:right {
  @top-left {
    content: string(IndexTerm, first);
  }
  @top-right {
    content: string(IndexTerm, last);
  }
}
```

### 19.2.4 要素をヘッダ・フッタに移動する: running() 位置値

要素をマージンボックスに表示させるには、position: running(name); を使用します。引数の名前は、element() 関数で要素が参照される名前です。

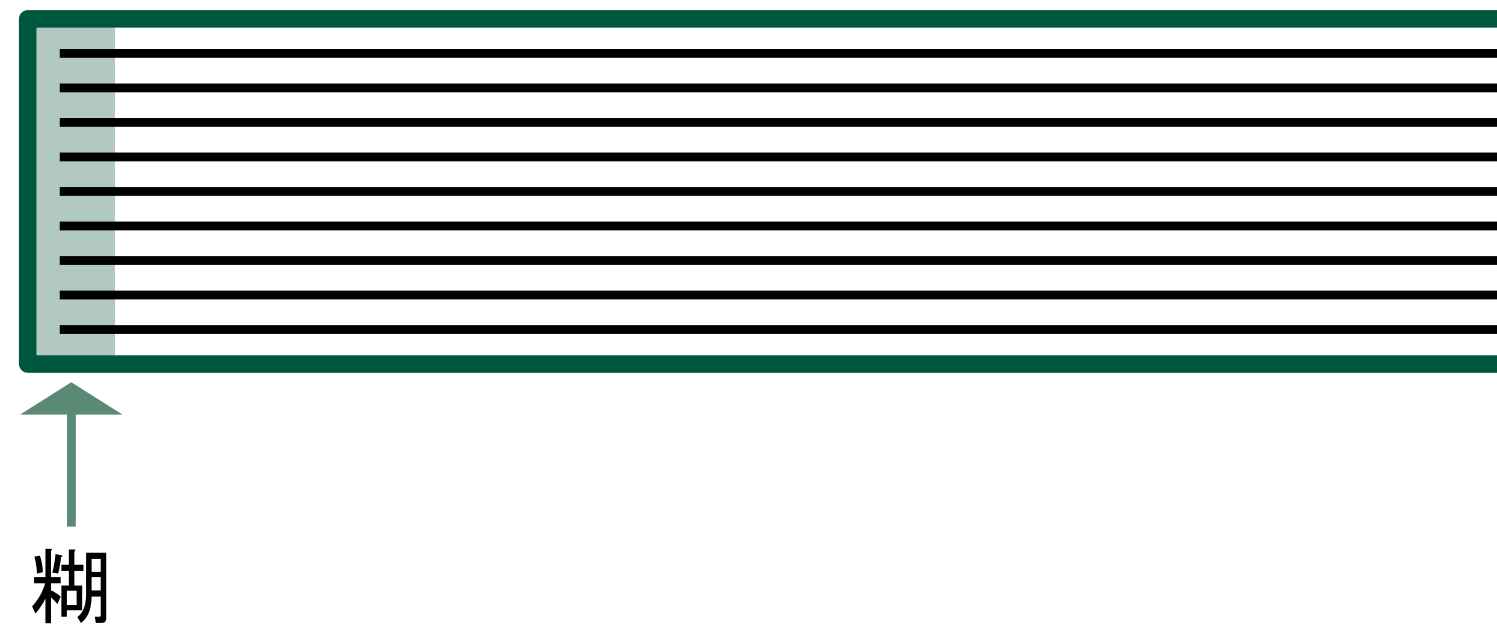
position: running(name); の要素は、あるべき位置に表示されません。あたかも display: none; が設定されているように扱われます。

```
p.Title {
  position: running(Title);
  text-indent: 0;
}
```

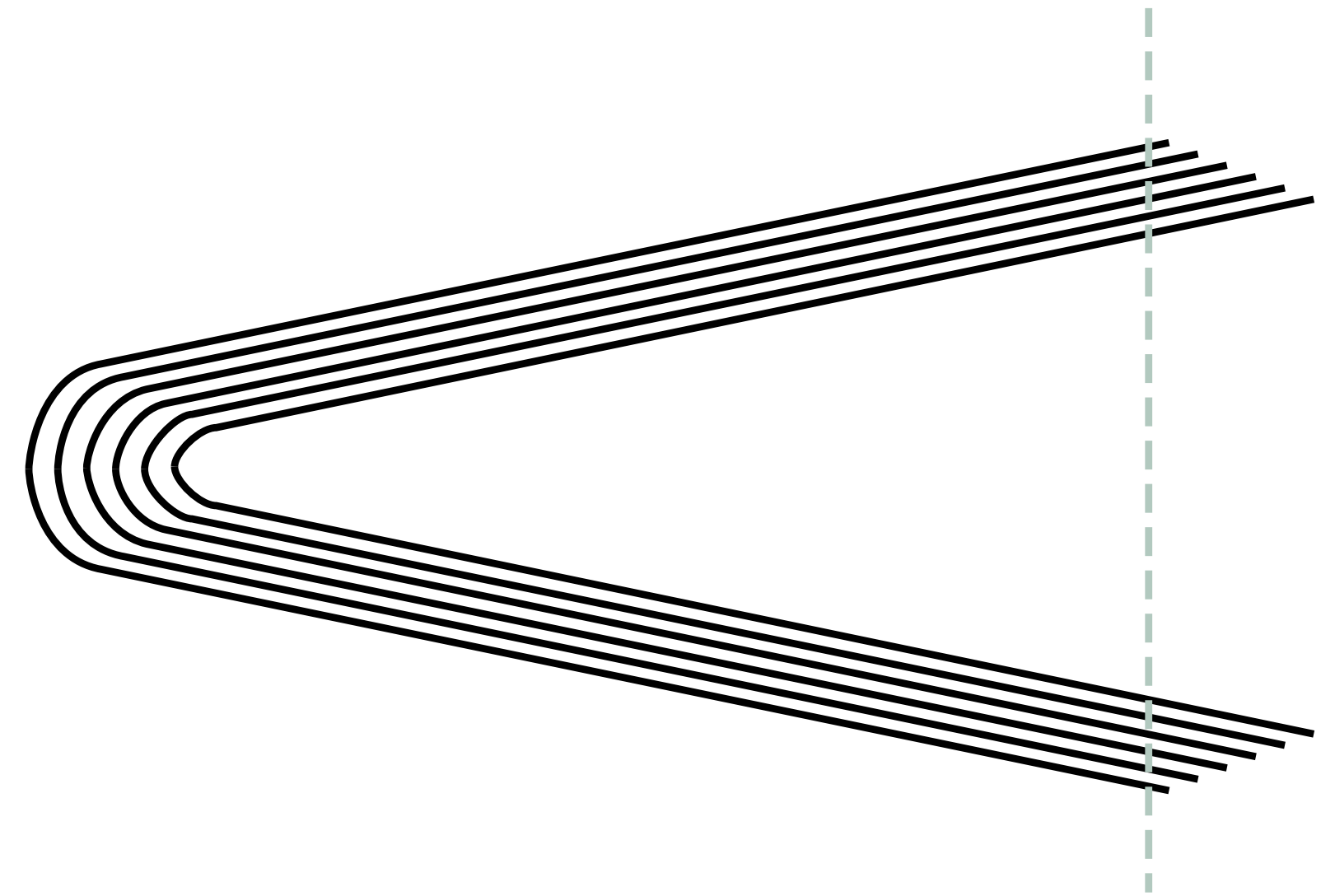
# 印刷本

- 市場/目的に合わせたページサイズ
  - オフィス文書
  - 車の取扱い説明書
  - ペーパーバック本
- 裁ち落とし
- 見開きページ
- 製本綴じ
- 裏ページの透け感

# 本の綴じ方

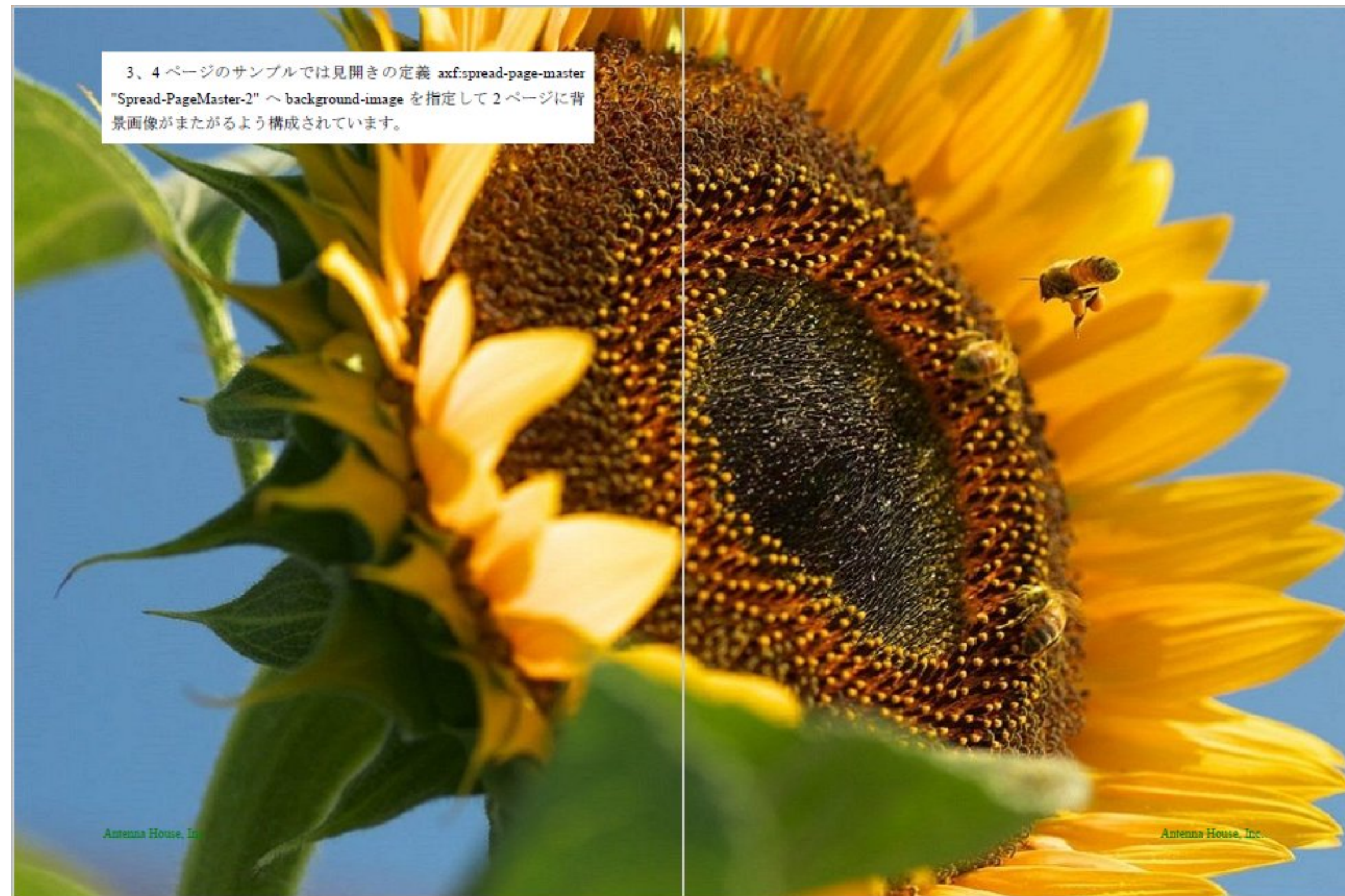


無線綴じでは、のど側が詰められます



中綴じでは、小口側が詰められます

# 見開きページ



# 裏ページの透け感

## CSSページ媒体入門

CSSをXMLやHTMLのページ媒体デザインのために使うことはまだ一版的ではありませんが、CSS3の開発が進むにつれて増えるものと期待さ

## CSSページ媒体入門

CSSをXMLやHTMLのページ媒体デザインのために使うことはまだ一版的ではありませんが、CSS3の開発が進むにつれて増えるものと期待さ



# 完全なチュートリアルを入手する

[https://www.antenna.co.jp/AHF/ahf\\_publication/index.html#CSSPrint](https://www.antenna.co.jp/AHF/ahf_publication/index.html#CSSPrint)

