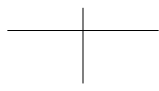
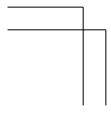
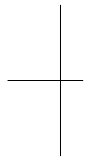
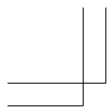


CSS ページ組版入門

アンテナハウス株式会社



**AH Formatter
V6.6**



はじめに

CSS は、ブラウザ、エディタ、その他のアプリケーションで広く使用されています。CSS は、Web デザインのためだけでなく、広範な印刷アプリケーションや PDF として配布される電子的なページ組版のスタイルシート仕様としても使われています。

CSS1 (Cascading Style Sheets Level 1) は 1996 年に W3C の勧告になり、CSS2 は 1998 年に、CSS 2.1 (CSS Level 2 Revision 1) は 2011 年に、W3C の勧告となりました。2018 年初頭の時点で、CSS 2.2 が開発中です。

レベル 2 後の CSS は、一般的に CSS3 として知られていますが、単一で一枚岩の CSS3 仕様はなくなるでしょう。レベル 2 を超える CSS は、それぞれが CSS の一部を規定する多数のモジュールに分割されてきました。これらのモジュールは独立に番号付けされています。CSS2 の上に構築された最初のバージョンのモジュールは CSS3 と呼ばれており、それぞれは CSS4 に取って代わられるでしょう。例えば、CSS Color Module Level 3 は CSS レベル 2 のいくつかの節を置き換えており、現在開発中の CSS Color Module Level 4 は徐々に CSS Color Module Level 3 を置き換えるでしょう。CSS2 の機能上に構築されていないモジュールはレベル 1 から開始します。例えば、CSS Multi-column Layout Module はレベル 1 です。

それぞれのモジュールの開発段階はバラバラであり、安定度もバラバラです。すべての W3C 勧告の安定度のレベルは、作業草案から始まり勧告に至るまでの範囲です。CSS ワーキング・グループは仕様の安定度の分類を、書き直しと研究から安定と完成までに分けて保守しています。

CSS ワーキング・グループは毎年、その時点における CSS の最新の状態のスナップショットを纏めています。CSS スナップショット 2017 は勧告と勧告候補の両方の仕様を 2017 年の公式な CSS の定義を構成するものとしてリストしています。

CSS 2.1 (と 2.2) はページ媒体への出力サポートは最小の提供であり、ページレイアウト機能は力不足です。CSS3 は、現在 W3C によって依然として開発中ですが、高度なページレイアウト、多段組、縦組、ハイフネーション、多言語の文字レイアウトなどの専門的な品質の組版に必要なさまざまな機能を定義しています。Antenna House Formatter は、独自開発の MathML 3, CGM, SVG の描画機能、ベースライン・グリッド、PANTONE® スポット・カラー、日本語レイアウト制御のための追加プロパティなどの最適な組版のための追加機能を提供しています。

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般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れています。本書はページ組版のための CSS を理解していただきやすくすることを目的としています。

Antenna House CSS Formatter

Antenna House Formatter ファミリーは、使用することのできるスタイルシートの種類と機能レベルの組み合わせにより利用できます。Antenna House Formatter と Antenna House CSS Formatter は、CSS を使用して XML または HTML のいずれかをフォーマットできます。詳細については、Antenna House の Web サイトを参照してください。

特に明記されていない限り、ここでは「AH Formatter」は、CSS を使用してフォーマットできる Antenna House Formatter を指します。

想定される読者

多くの方はブラウザ上の Web 制作では CSS に精通しており、その中でも非常に精通している人もいれば、それ程でもない人もいます。しかし、ページ組版に CSS を使用するのにあたって精通している人はほとんどいません。

本書は、既に CSS を知っているが、ページ組版のために CSS を使用したことがない人を対象としています。既に CSS 構文、border、padding、font-size などの通常のプロパティの使用には精通しているはずですが、本書ではこれらについて説明しますが、詳細については説明しません。

もし既にページ組版以外の CSS のほとんどの点に精通しているなら、第 11 章 色指定 (ページ 57)、第 16 章 PDF ブックマーク (ページ 79)、第 18 章 ページ設定 (ページ 91)、第 19 章 ヘッダとフッタ (ページ 101)、第 20 章 PDF 出力 (ページ 109) がかなり役に立つ章となるでしょう。

また本書は、CSS ページ組版入門編であるため、CSS、AH Formatter のすべてのプロパティ、セレクタ、規則、機能を説明することを意図していません。利用可能な全範囲の機能の詳細については、CSS 仕様書、お使いのバージョンの AH Formatter のオンラインマニュアルを参照してください。

本書で使用されている表記規則

- プロパティの説明は以下を含みます。
 - 初期値
 - プロパティが適用される要素
 - プロパティが継承されるかどうか
 - 説明文
 - 値のリスト

プロパティの説明は次のように記述されます。

カウンタの加算：counter-increment プロパティ

- 初期値：none ●適用対象：すべての要素 ●値の継承：しない
- 指定したカウンタの値を増やすには、counter-increment プロパティを使います。
- none：カウンタを変更しません。
 - カウンタ名：指定した名前のカウンタの値をひとつ増やします。
 - カウンタ名 整数：スペース文字で区切って整数値を指定すると、カウンタの値がその数だけ前後します。

- 本書で説明している CSS の機能は、次のように分けることができます。

1. CSS 2.1 までに規定されている：マークなし
2. CSS3 で規定されている：☑
3. AH Formatter の独自拡張機能：📄

CSS3 は開発中の仕様のため、CSS3 で規定されている仕様は、将来、変更になることがあります。CSS3 プロパティは -ah- プレフィックスの有無にかかわらず使用できます。AH Formatter の独自拡張機能は、-ah- プレフィックスを付けないと正しく動作しません。

- テキストを強調したときは黄色い背景のテキストで示します。
- スタイル付けしたテキストの例はボックスで囲みます。

任意のタイトル

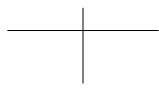
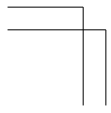
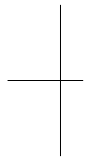
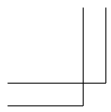
このページのページ数 = 3
この文書の総ページ数 = 142

- マークアップの例または CSS の例は、等幅フォントで表示します。

```
<p> このページのページ数 = <span style="content: counter(page)"></span>
</p>
<p>この文書の総ページ数=
  <span style="content: counter(pages)"></span>
</p>
```

フィードバック












本書に関して改善のご提案を歓迎いたします。xml-info@antenna.co.jp 宛てにご意見、修正、改善のご提案などご連絡いただければ幸いです。



目次

はじめに

第1章 Web とページ媒体	1
1.1 @media ルール	1
1.2 印刷向けスタイルシートの指定	1
1.2.1 <style> 要素	1
1.2.2 @import ルール	1
1.2.3 <style> と <link> 要素の media 属性	2
1.3 画面とページ媒体の相違	2
1.3.1 デザインのアプローチ	2
1.3.2 分割	2
1.3.3 フロート	2
1.3.4 ナビゲーション	2
1.3.5 左・右ページ	3
1.3.6 印刷本	3
第2章 ボックスによるレイアウト	5
2.1 ボックスモデル	5
2.2 ボックスの表示と印刷	6
2.3 表示形式：display プロパティ	6
2.4 ボックスの配置	7
2.5 ボックスの大きさを決める特性	8
2.5.1 コンテンツの幅：width プロパティ	8
2.5.2 コンテンツの最小幅：min-width プロパティ	8
2.5.3 コンテンツの最大幅：max-width プロパティ	8
2.5.4 コンテンツの高さ：height プロパティ	8
2.5.5 コンテンツの最小高さ：min-height プロパティ	8
2.5.6 コンテンツの最大高さ：max-height プロパティ	8
2.5.7 パディングの幅：padding プロパティ	8
2.5.8 ボーダーの幅（太さ）：border-width プロパティ	9
2.5.9 マージンの幅：margin プロパティ	9
2.5.10 ボックスモデルの変更：box-sizing プロパティ	9

第 3 章	オブジェクトの装飾	11
3.1	ボーダー	11
3.2	ボーダーのスタイル：border-style プロパティ	11
3.3	ボーダーの幅：border-width プロパティ	12
3.4	ボーダーの色：border-color プロパティ	13
3.5	各辺のボーダープロパティ：border-top/border-bottom/border-left/border-right プロパティ	13
3.6	ボーダーショートハンド：border プロパティ	13
3.7	角丸：border-radius プロパティ 	13
3.8	ボックスシャドウ：box-shadow プロパティ 	14
3.9	背景色：background-color プロパティ	14
3.10	背景画像：background-image プロパティ	15
3.11	背景画像の繰り返し：background-repeat プロパティ	16
3.12	背景画像の位置：background-position プロパティ	16
第 4 章	段落設定	17
4.1	行揃え	17
4.1.1	行揃え：text-align プロパティ	17
4.1.2	最終行の配置：text-align-last プロパティ 	18
4.1.3	先頭揃え：text-align-first プロパティ 	20
4.1.4	上付き文字・下付き文字がある行の高さ：-ah-line-height-shift-adjustment プロパティ 	20
4.1.5	行送りの調整：-ah-line-stacking-strategy プロパティ 	21
4.2	行グリッド	22
4.2.1	行グリッドの指定：-ah-baseline-grid プロパティ 	23
4.2.2	ブロックを行グリッドに揃える：-ah-baseline-block-snap プロパティ 	23
4.3	リーダー：leader() 関数 	24
4.4	ハイフネーション	24
4.4.1	ハイフネーション：hyphens プロパティ 	24
4.4.2	ハイフン分割の前の最小文字数：hyphenate-before プロパティ 	25
4.4.3	ハイフン分割の後の最小文字数：hyphenate-after プロパティ 	26
4.5	ブロックの縦書き：writing-mode: tb-rl 	26
第 5 章	段組	27

5.1 段数：column-count プロパティ 	27
5.2 段幅：column-width プロパティ 	27
5.3 段数, 幅：columns プロパティ 	28
5.4 段抜き：column-span プロパティ 	28
5.5 段間：column-gap プロパティ 	28
5.6 段間罫：column-rule プロパティ 	28
5.7 段間罫の種類：column-rule-style プロパティ 	29
5.8 段間罫の幅：column-rule-width プロパティ 	29
5.9 段間罫の色：column-rule-color プロパティ 	29
5.10 段間罫の長さ：-ah-column-rule-length プロパティ 	29
5.11 段間罫の揃え：-ah-column-rule-align プロパティ 	29
5.12 段間罫表示：-ah-column-rule-display プロパティ 	30
第 6 章 改ページ・ページコンテンツの維持	31
6.1 改ページの制御	31
6.1.1 改ページ：page-break-before/page-break-after プロパティ	31
6.1.2 改ページの禁止：page-break-inside プロパティ	31
6.1.3 左右どちらかのページから始める	32
6.2 改ページ前後の最小行数：orphans/widows プロパティ	32
第 7 章 文字設定	33
7.1 フォント指定	33
7.1.1 フォント：font プロパティ	33
7.1.2 フォントサイズ：font-size プロパティ	33
7.1.3 フォントの種類：font-family プロパティ	34
7.1.4 フォントの太さ：font-weight プロパティ	34
7.1.5 イタリック体・斜体：font-style プロパティ	35
7.1.6 スモールキャピタル・フォント機能：font-variant プロパティ	35
7.2 追加のフォント：@font-face ルール	36
7.3 行の高さ：line-height プロパティ	36
7.4 縦方向の揃え位置：vertical-align プロパティ	37
7.5 下線・上線・打ち消し線	37
7.5.1 文字の修飾 ショートハンド：text-decoration プロパティ 	37
7.5.2 下線・上線・打ち消し線：text-decoration-line プロパティ 	38

7.5.3 線の種類：text-decoration-style プロパティ 	38
7.5.4 線の色：text-decoration-color プロパティ 	38
7.5.5 線の幅：-ah-text-line-width プロパティ 	38
第 8 章 MathML と SVG グラフィクス	39
8.1 MathML	39
8.2 SVG グラフィクス	40
第 9 章 画像配置	41
9.1 インライン画像	41
9.2 ブロック画像	41
9.3 フロート配置：float プロパティ	42
9.3.1 サイドフロート：float: left/float: right	42
9.3.2 ページフロート：float: top page/float: bottom page 	43
9.3.3 コラムフロート：float: top column/float: bottom column 	44
9.3.4 フロートの値の拡張：横書き inside/outside/center/start/end/alternate 	44
9.3.5 フロートの値の拡張：縦書き top/bottom/center/before/inside/after/outside 	47
9.3.6 段組フロート：float: multicol 	48
9.4 回り込みの解除：clear プロパティ	48
第 10 章 表組	49
10.1 表の作成方法	49
10.2 ボックス基本特性の表への適用	50
10.3 表のボーダーモデル	51
10.3.1 ボーダーを分離するかどうか：border-collapse プロパティ	51
10.3.2 ボーダー間のアキ：border-spacing プロパティ	51
10.3.3 ボーダーの優先度	52
10.4 表の幅・段の幅の決定方法：table-layout プロパティ	53
10.5 表の中央揃え	54
10.6 表のキャプションの位置：caption-side プロパティ	54
第 11 章 色指定	57
11.1 印刷の色	57
11.2 文字色：color プロパティ	58

11.2.1	CMYK 色	59
11.2.2	不透明度	60
11.2.3	rgb-icc()	60
11.2.4	グレイスケール	60
11.2.5	PANTONE® スポットカラー	61
11.2.6	その他のスポットカラー	62
第 12 章 カウンタ		63
12.1	章や節に番号を付ける	63
12.2	文字挿入：content プロパティ	64
12.3	カウンタの加算：counter-increment プロパティ	64
12.4	カウンタのリセット：counter-reset プロパティ	65
12.5	ページカウンタ	65
第 13 章 リスト		67
13.1	リスト項目マーカ画像：list-style-image プロパティ	67
13.2	リスト項目マーカ 種類：list-style-type プロパティ	67
13.3	リスト項目マーカ位置：list-style-position プロパティ	67
13.4	リスト項目マーカショートハンド：list-style プロパティ	68
13.5	リスト項目マーカ：::marker 疑似要素	68
13.6	カウンタスタイル	68
13.7	カスタムカウンタスタイルの定義：@counter-style ルール	69
13.7.1	カウンタアルゴリズム：system 記述子	70
13.7.2	負の値の組版：negative 記述子	70
13.7.3	マーカ前の記号：prefix 記述子	70
13.7.4	マーカ後の記号：suffix 記述子	71
13.7.5	カウンタの範囲：range 記述子	71
13.7.6	最小のカウンタ幅：pad 記述子	71
13.7.7	フォールバックカウンタスタイル：fallback 記述子	71
13.7.8	カウンタの記号：symbols および additive-symbols 記述子	71
13.8	定義済みカウンタスタイル	71
第 14 章 クロスリファレンス（相互参照）		75
14.1	テキスト内容の参照：target-text() 関数	75

14.2	カウンタの参照：target-counter() 関数 	75
14.3	目次作成	76
第 15 章 脚注・傍注		77
15.1	脚注の指定：float: footnote 	77
15.2	脚注のスタイル：@footnote ルール 	77
15.3	脚注番号：::footnote-call/::footnote-marker 疑似要素 	77
15.4	傍注の指定：float: sidenote 	78
15.5	傍注スタイル：@sidenote ルール 	78
第 16 章 PDF ブックマーク		79
16.1	ブックマークレベル：bookmark-level プロパティ 	79
16.2	ブックマークラベル：bookmark-label プロパティ 	79
16.3	ブックマークの状態：bookmark-state プロパティ 	79
第 17 章 日本語文字組版		81
17.1	行頭行末約物半角固定・段落先頭全角 1 字下げ	81
17.2	約物の詰め処理：punctuation-trim プロパティ 	81
17.3	行調整のための約物の詰め：-ah-text-justify-trim プロパティ 	83
17.4	和欧文間のアキ：-ah-text-autospace プロパティ 	83
17.5	和欧文間のアキ量：-ah-text-autospace-width プロパティ 	84
17.6	突き出し・ぶら下げ：hanging-punctuation プロパティ 	84
17.7	行頭約物半角固定・行末約物全角/半角・段落先頭全角 1 字下げ	85
17.8	行頭約物半角固定・行末句点全角・段落先頭全角 1 字下げ	86
17.9	行頭行末約物半角固定・段落先頭全角 1 字下げ（起こし食い込み）	87
17.10	縦中横	88
17.11	ルビ・圏点	88
第 18 章 ページ設定		91
18.1	欧文ページデザイン	91
18.2	日本語ページデザイン	92
18.3	@page ルール	93
18.4	ページボックスのサイズ：size プロパティ 	93
18.5	ページマージン：margin プロパティ	94

18.6	名前付きページタイプ：page プロパティ 	95
18.7	ページ数の制約：-ah-force-page-count プロパティ 	95
18.8	トンボ	97
18.8.1	トンボの表示：marks プロパティ 	98
18.8.2	トンボの可視性：-ah-crop-area-visibility プロパティ 	99
18.8.3	トンボの線の色：-ah-printer-marks-line-color プロパティ 	99
18.8.4	トンボの線の長さ：-ah-printer-marks-line-length プロパティ 	99
18.8.5	トンボの線の幅：-ah-printer-marks-line-width プロパティ 	99
18.8.6	出力媒体の端から仕上がり寸法までの距離：-ah-crop-offset プロパティ 	99
18.9	裁ち落とし（塗り足し）	100
18.9.1	裁ち落とし（塗り足し）の幅：bleed プロパティ 	100
第 19 章 ヘッダとフッタ		101
19.1	マージンボックス 	101
19.2	柱とノンブル	102
19.2.1	柱の設定：string-set プロパティ と string() 機能 	102
19.2.2	文字列変数：string-set プロパティ 	102
19.2.3	string() 	102
19.2.4	要素をヘッダ・フッタに移動する：running() 位置値 	103
19.2.5	running 要素を挿入：element() 	104
19.2.6	ノンブル（ページ番号）：counter(page) 	104
19.2.7	総ページ数：counter(pages) 	104
19.3	左・右・先頭・空白ページ：:left・:right・:first・:blank	105
19.4	最終ページ・単一ページ：:last・:only 	107
19.5	進行方向：writing-mode プロパティ 	107
第 20 章 PDF 出力		109
20.1	PDF バージョン	109
20.2	タグ付き PDF	110
20.2.1	カスタム PDF タグ名：-ah-pdftag プロパティ 	110
20.3	PDF/X	111
20.4	PDF/A	112
20.5	PDF/UA	112
20.5.1	マッターホルン プロトコル	113

20.5.2 PAC 3 PDF/UA チェッカー	113
20.6 文書 プロパティ 	114
20.6.1 Extensible Metadata Platform (XMP)	115
20.7 ページ表示 	117

セレクタと疑似要素

参照仕様

索引

第1章 Webとページ媒体

1.1 @media ルール

CSS スタイルシートのルールの組み合わせを適用する対象媒体を @media ルールで指定します。ページ媒体向けのルールは @media print を、画面表示向けのルールは @media screen を指定します¹⁾。

```
@media print { /* ページ媒体に適用 */
  body {
    margin: 0;
    font-size: 10pt;
  }
}

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

body { /* すべての媒体に適用 */
  font-family: sans-serif;
}
```

1.2 印刷向けスタイルシートの指定

1.2.1 <style> 要素

<style> 要素には文書のスタイル情報が含まれています。HTML 4.01 では、<style> は <head> 内のみに現れます。HTML5 では、<style> は <body> 内でも使用できます。

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

1.2.2 @import ルール

印刷向けスタイルシートを別の CSS ファイルで作成し、@import で読み込むことができます。

```
@import url("PrintOnly.css") print; /* PrintOnly.css は印刷用 */
```

1) AH Formatter は印刷用に組版するソフトウェアですので、GUI の画面にも @media screen を適用せず、@media print を適用します。

1.2.3 <style> と <link> 要素の media 属性

印刷向けスタイルシートを <style> または <link> 要素でリンクし、media 属性の値に print を指定します。

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

1.3 画面とページ媒体の相違

1.3.1 デザインのアプローチ

画面表示用と印刷用（または任意の種類のパージ媒体）では、レイアウトデザインの考え方を考える必要があります。

画面表示では、表示環境に依存して画面のサイズや画面の縦横比などが変化するため、レイアウト対象のサイズや配置を厳密に指定するとかえって見づらくなることがあります。スタイル指定では相対寸法を使ってさまざまな環境に対応できるように考慮しなければなりません。

印刷では、組版対象を固定サイズの用紙にきれいに配置することが期待されるため、レイアウト指定では、文字のサイズを始めとして組版対象のサイズと位置を絶対寸法で指定して、レイアウトを厳密に制御することが必要となります。

1.3.2 分割

分割は、ページ概念のある文書、ない文書に関わらず起きます。例えば、テキストは行に分割され、column-count が 1 より大きいブロック内のテキストが段に分割される可能性があります。テキストの流れが 2 ページに跨る場合は特に気になるところです。要素間または要素内での分割を強制的に行ったり、禁止するプロパティがあります。widows, orphans プロパティはテキストブロックの内の分割前後の最小行数を制御します。

1.3.3 フロート

ページ概念のない媒体では、ボックスは左右にフロートさせることが可能です。ページ媒体では、ボックスはページの上部または下部にフロートさせることもできます（AH Formatter はフロートのより詳細な制御を実装しています）。画像もフロートさせることができます。フロートを使って傍注や脚注も配置できます。

1.3.4 ナビゲーション

ページ媒体（例えば書籍）は、ページ間を移動するための非常に良くできた決まりごとを持っています。

一般的にページには番号を振り、前付には本文と異なるスタイルで番号を振ることがあります。

書籍では、ページ番号や、章や節の見出しがページ上に表示されます。辞書には、各ページの最初と最後の項目を示すための独自の規則があります。CSSは、そういった情報を表示するために、ページの端の周りに16のエリアを定義しています。

目次と索引を使えば、順を追わず情報にアクセスすることが容易にできます。

章（または重要な内容）が、右側のページから始まり（左から右へ記述する言語の文書の場合）、章の始めが他のページと異なる体裁を持つ場合もあります。（ヘッダ・フッタなどが異なる可能性があります。）

1.3.5 左・右ページ

製本用にページの両面に印刷された文書（オフィスプリンタで両面印刷してフォルダに入れた文書も含めて）は、左ページと右ページを持つ見開きページの形状を持ちます。また、製本された文書では、背表紙の近くのページの内側の端を読むよりも、ページの外側の端にある内容を読むほうが簡単です。

これらのすべての側面はページデザインに影響する可能性があります。例えば、章の開始は横組の本では一般的に見開きの右側にあります。縦組の本では逆に見開きの左側にあります。ページのノンブルや色々なナビゲーションの機能は、ページをめくるときに簡単に目につくように各ページの外側の端（小口より）またはその近くにある場合が多いでしょう。文書を見開きのシーケンスとみなすと、見開きページがどのように対称でなければならないか、グラフィクスなどが見開きページに跨るかどうかなど、問題が湧いてきます。

1.3.6 印刷本

画面上の見た目のみではなく、文書を印刷するとなると色々なことを考える必要があります。

ページサイズの制限がある場合があります。エンドユーザによって印刷されることを意図した文書はオフィスのプリンタの用紙サイズに合うようにサイズを設定できます。米国におけるレターサイズ、世界のほとんどの地域でA4、日本ではA4またはJIS-B5などがあります。一方、車のハンドブックは車のダッシュボードに入るサイズでしょう。文庫本は従来の決まったサイズがあり、規格外のサイズを選択すると、書籍の販売に影響する可能性があります。

用紙が十分に不透明でない場合、用紙の反対側のテキストが透けて見えることがあります。用紙の両面のテキストが揃っていないと、見栄えはよくありません。

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

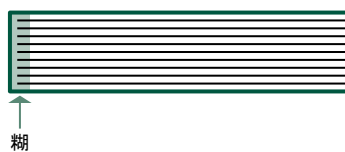
CSSページ媒体入門

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

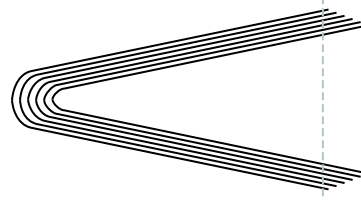
テキスト揃え・揃えなしの透明の効果

ページの端まで広がった画像や他のデザイン要素は、ページの端を超えて印刷する必要がある場合があります（第18章 ページ設定（ページ91）を参照してください）。ページの端に塗り足しがない場合、印刷して製本してページを正しい仕上がりサイズにすると、画像とページの端との間に白い帯ができてしまいます。逆に、仕上がりサイズに大幅に裁ち切る場合、画像はページの端の近くに置かない方がよいでしょう。

本をデザインするときには、製本方法まで考慮する必要があります。製本綴じまたはリング綴じでは、見開きページのノドマージンエリアを減らすことができます。中綴じのように本のページが束に集められて仕上がりサイズに裁ち切られる場合、束の真ん中のページは、束の外側のページよりも小口側が多く裁ち切られている可能性があります。



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

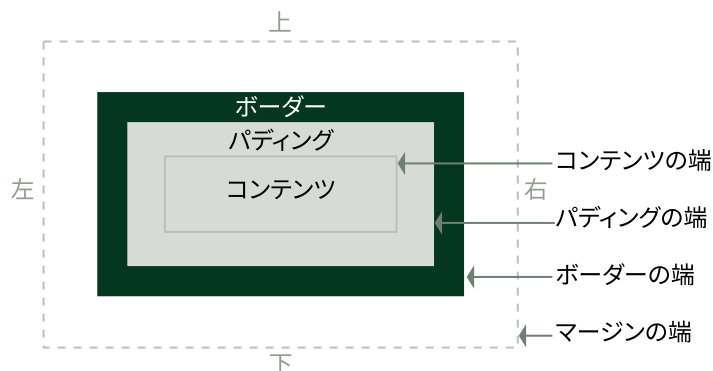
製本方法の効果

第2章

ボックスによるレイアウト

2.1 ボックスモデル

CSS では、すべてはボックスで構成されます。文書を構成する要素とテキストのツリーは、ボックスのツリーに変換されます。ある要素は複数ボックスを生成し（例：``）、ある要素は何も生成しません（例えば、`<col>` のような要素、あるいは、`display: none;` 指定の要素）。ボックスを幅広い意味で分類すると、ブロックレベル、行、インラインレベルのボックスがあります。段落などのブロックレベルのボックスには、他のブロックレベルのボックスまたは行ボックスが含まれます。例えば、段落にテキストとリストが含まれている場合、ボックスがブロックレベルと行の両方のボックスを持たないようにひとつ以上の匿名ボックスがテキストの行ボックスの周囲に作成されます。行ボックスにはひとつまたは複数のインラインレベルのボックスが含まれます。フォントまたはスタイルの変更ごとに別々のインラインレベルのボックスが生成されるためです。

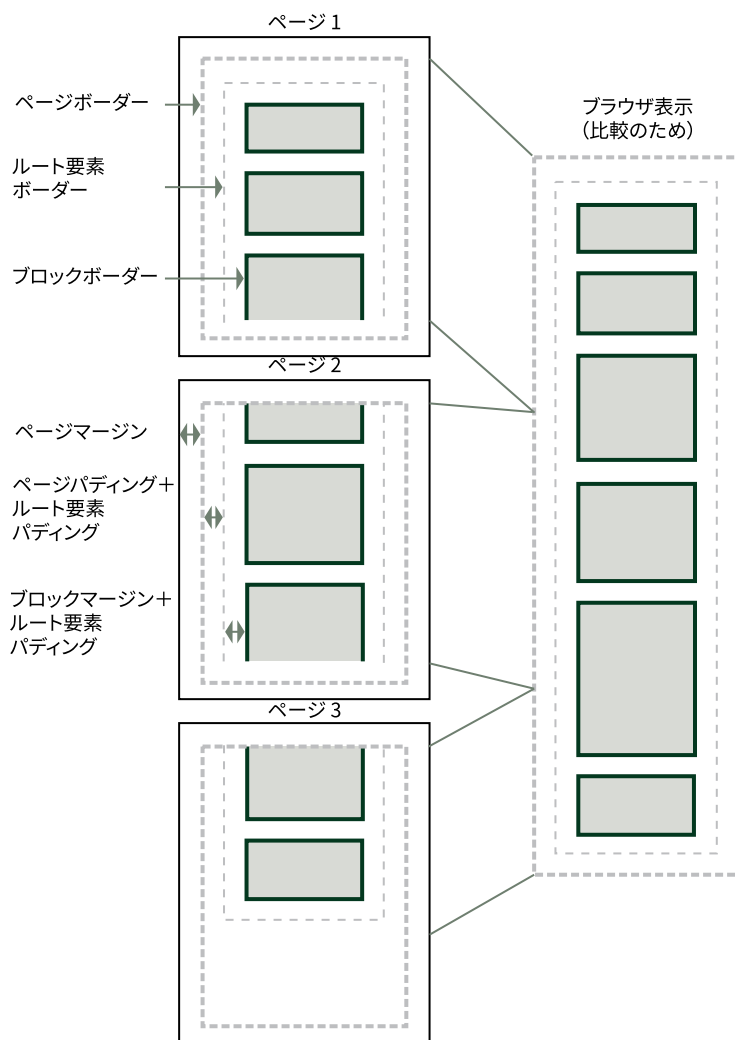


ボックスのエリアと端

それぞれのボックスがコンテンツエリアを持ち、コンテンツの端はコンテンツエリアの境界になります。パディングエリアはコンテンツエリアを囲み、そのパディングエリアはパディング端が境界となっています。同様に、ボーダーはパディングを囲み、マージンはボーダーを囲んでいます。CSS 定義におけるボックスのエリア生成において、上記の説明ではパディング、ボーダー、マージンはあたかも一定の幅を持っているように見えますが、パディング、ボーダー、マージンの値がゼロに設定されているときがあります。

2.2 ボックスの表示と印刷

CSS では、最下位の文字列を含むボックスから、最上位のルート要素のボックスまで階層構造になっています。出力先が画面のときは、ルート要素のボックスを画面に表示します。出力先がページ媒体のときは、ページボックスでページの雛形を設定し、ページに他のボックスを配置します。



ボックスの画面表示と印刷の対応関係

2.3 表示形式：display プロパティ

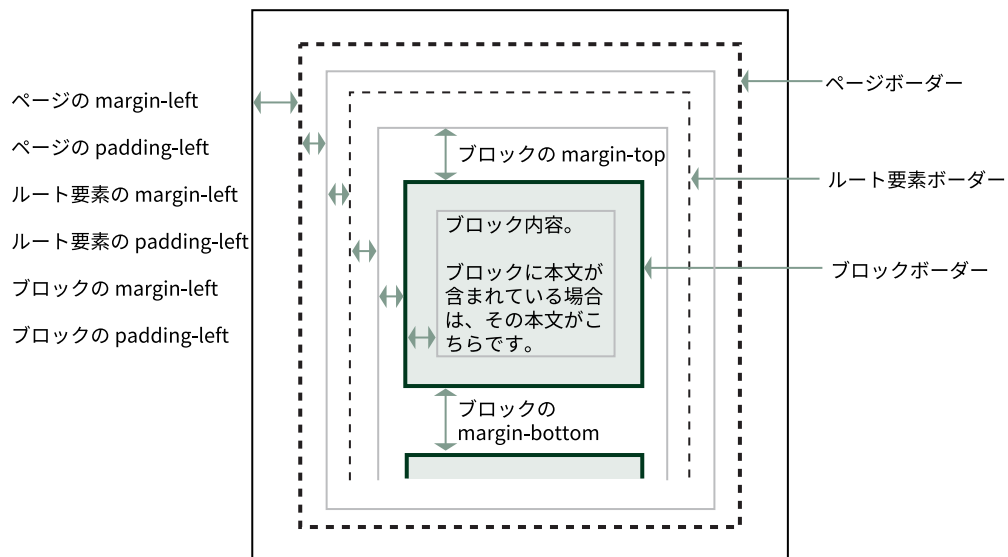
- 初期値：inline
 - 適用対象：すべての要素
 - 値の継承：しない
- 要素の表示方法を制御するには、display プロパティを使います。

display プロパティは、要素の表示形式を変更するだけで、要素のもつ役割は変更しません。

- block：ブロックボックスを生成します。
- inline-block：その要素はインラインレベルのボックスとなり、その内部はブロックボックスとしてレイアウトされます。
- inline：ひとつ以上のインラインボックスを生成します。
- list-item：主ボックスとマーカボックスを生成します。
- none：要素が表示されないようにします。
- table, inline-table, table-row-group, table-column, table-column-group, table-header-group, table-footer-group, table-row, table-cell, table-caption：表要素と同じように振る舞います。第10章 表組（ページ 49）を参照してください。

2.4 ボックスの配置

表以外のボックスは、内側から、コンテンツ、パディング、ボーダー（境界）をもち、最外部にマージンをもちます。それぞれに特有のプロパティによってレイアウト特性を指定します。ページボックス、ルート要素のボックス、ブロックのボックスという階層構造になっているとき、それぞれのコンテンツ、パディング、ボーダーは、次の図のように配置されます。表については、第10章 表組（ページ 49）を参照してください。



ボックスの配置

2.5 ボックスの大きさを決める特性

ボックスの大きさは、コンテンツの幅と高さとその周りのパディング、ボーダー、マージンの四つの特性値の合計となります。

2.5.1 コンテンツの幅：width プロパティ

●初期値：auto ●適用対象：すべての要素²⁾ ●値の継承：しない
コンテンツの幅を指定します。

2.5.2 コンテンツの最小幅：min-width プロパティ

●初期値：0 ●適用対象：すべての要素 ●値の継承：しない

2.5.3 コンテンツの最大幅：max-width プロパティ

●初期値：none ●適用対象：すべての要素 ●値の継承：しない

2.5.4 コンテンツの高さ：height プロパティ

●初期値：auto ●適用対象：すべての要素³⁾ ●値の継承：しない
コンテンツの高さを指定します。

2.5.5 コンテンツの最小高さ：min-height プロパティ

●初期値：0 ●適用対象：すべての要素 ●値の継承：しない

2.5.6 コンテンツの最大高さ：max-height プロパティ

●初期値：none ●適用対象：すべての要素 ●値の継承：しない

2.5.7 パディングの幅：padding プロパティ

●初期値：0 ●適用対象：すべての要素⁴⁾ ●値の継承：しない

padding-top, padding-bottom, padding-left, padding-right に上下左右のパディングの幅を指定するショートハンドプロパティ（複数のプロパティの値をひとつのプロパティで指定するためのもので、本書ではショートハンドと略記します。）です。padding に四辺の値を同時に指定できます（ショートハンドの指定方法については第3章 オブジェクトの装飾（ページ 11）を参照してください）。

2) 表の行、および文字列などを含むインライン要素の幅は、自動的に決定されますので適用対象外です。ただし、インライン要素のうちイメージ要素には適用されます。min-width, max-width プロパティも同様です。

3) 表の列、および文字列などを含むインライン要素の高さは、自動的に決定されますので適用対象外です。ただし、インライン要素のうちイメージには適用されます。

4) 表の行、列、表の見出し、フッタは適用対象外です。

2.5.8 ボーダーの幅（太さ）：border-width プロパティ

●初期値：medium ●適用対象：すべての要素 ●値の継承：しない

border-top-width, border-bottom-width, border-left-width, border-right-width に上下左右のボーダーの幅を指定するショートハンドです。border-width に四辺の値を同時に指定できます。

ボーダーは、border-color で色、border-style で線種（スタイル）の特性も指定できます。ボーダーの幅、色、線種については第3章オブジェクトの装飾（ページ11）を参照してください。

2.5.9 マージンの幅：margin プロパティ

●初期値：0 ●適用対象：ほぼすべての要素 ●値の継承：しない

margin-top, margin-bottom, margin-left, margin-right に上下左右のマージンの幅を指定するショートハンドです。margin に四辺の値を同時に指定できます。

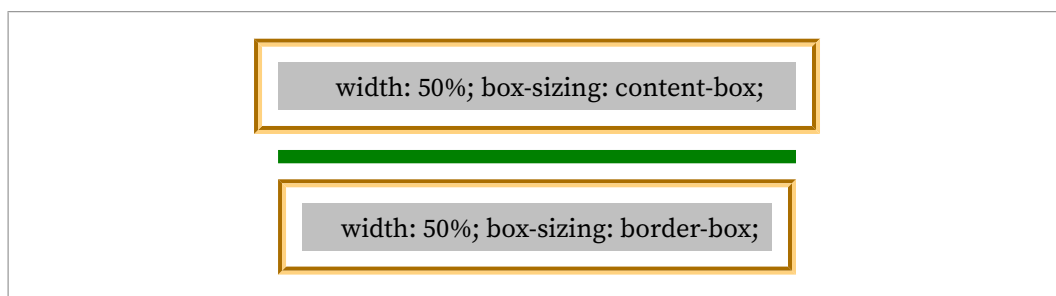
マージンの値はマイナスになることがあります。マイナスを指定された辺は、包含するボックスからはみ出します。

2.5.10 ボックスモデルの変更：box-sizing プロパティ

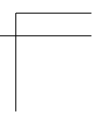
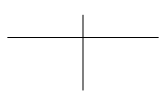
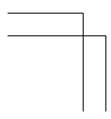
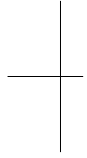
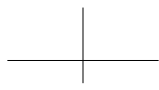
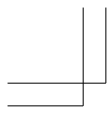
●初期値：content-box ●適用対象：幅または高さを受け入れるすべての要素 ●値の継承：しない

指定された幅と高さの内側または外側にパディングとボーダーを描画するかどうかを指定します。width と height プロパティの auto には影響しません。

- content-box：指定された幅と高さは要素のコンテンツボックスに適用されます。
- border-box：指定された幅と高さは要素のボーダーボックスに適用されます。



```
<div style="width: 50%; box-sizing: content-box; ...">
  <p style="background-color: silver; text-align: center;">...</p></div>
<p style="border-top: thick solid green; width: 50%; ..." />
<div style="width: 50%; box-sizing: border-box; ...">
  <p style="background-color: silver; text-align: center;">...</p></div>
```



オブジェクトの装飾

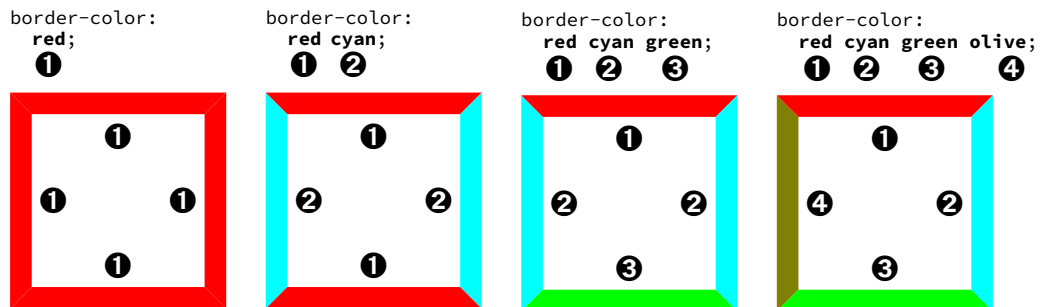
3.1 ボーダー

任意のオブジェクト⁵⁾に `border-*` プロパティを使用してボーダーを指定できます。`border-style` で線種 (スタイル), `border-width` で幅 (太さ), `border-color` で色を指定します。他に、角の丸めを設定したり、影を付けることができます。

ボーダーは四辺別々に異なる装飾を指定できます。またショートハンドがいくつかあります。`border-*` プロパティは、四辺のボーダーのスタイルなどを一度に設定するためのショートハンドです。`border` プロパティは、四辺すべてのボーダーのスタイル、幅、色を設定するためのショートハンドです。または、ボーダーの各辺にスタイル、幅、色を設定するプロパティと、スタイルのいずれかを設定するプロパティもあります。

`border-*` プロパティには 1~4 のコンポーネント値を指定できます。値は、以下の `border-color` の例が示す様に、指定する数によって異なる辺に設定できます。

- 値をひとつ指定：指定した値が [上下左右] のボーダースタイルになります。
- 値をふたつ指定：記述した順に [上下] [左右] のボーダースタイルになります。
- 値を三つ指定：記述した順に [上] [左右] [下] のボーダースタイルになります。
- 値を四つ指定：記述した順に [上] [右] [下] [左] のボーダースタイルになります。



3.2 ボーダーのスタイル：border-style プロパティ

- 初期値：プロパティを参照
 - 適用対象：すべての要素
 - 値の継承：しない
- 1~4 のコンポーネント値は、四辺すべてのボーダースタイルを設定します。以下のスタイルが使用可能です。

5) ボックスを生成できる <p>などの要素

none : ボーダーはありません。(幅=0)

hidden : ボーダーは非表示です。

solid : 実線

double : 二重線

dotted : 点線

dashed : 破線

dot-dash : 一点鎖線

dot-dot-dash : 二点鎖線

wave : 波線

groove : 立体的にへこんだように見える線

ridge : 立体的に隆起したように見える線

inset : ボーダーの範囲が立体的にへこんだように見える線

outset : ボーダーの範囲が立体的に隆起したように見える線

個別のボーダースタイルプロパティは、border-top-style、border-bottom-style、border-left-style、border-right-style で、初期値は none です。

3.3 ボーダーの幅 : border-width プロパティ

●初期値 : プロパティを参照 ●適用対象 : すべての要素 ●値の継承 : しない

1~4 のコンポーネント値は、四辺すべてのボーダー幅を設定します。

thin : 細いボーダー

medium : 通常のボーダー

thick : 太いボーダー

<length> : ボーダーの長さの明示的な値。負の値は使えません。(例: border-width: 2pt;)

個別のボーダーの幅プロパティは、border-top-width, border-bottom-width, border-left-width, border-right-width で、初期値は medium です。

3

3.4 ボーダーの色 : border-color プロパティ

- 初期値 : プロパティを参照 ●適用対象 : すべての要素 ●値の継承 : しない
- 1~4 のコンポーネント値は、四辺すべてのボーダー色を設定します。
- <color> : 色。第 11 章 色指定 (ページ 57) を参照してください。
 - transparent : ボーダーは透明ですが、幅がある場合があります。

個別のボーダースタイルプロパティは、border-top-color, border-bottom-color, border-left-color, border-right-color で、初期値は文字の色です。

3.5 各辺のボーダープロパティ : border-top/border-bottom/border-left/border-right プロパティ

- 初期値 : プロパティを参照 ●適用対象 : すべての要素 ●値の継承 : しない
- 各プロパティは、ボックスの上下左右のボーダーの幅、スタイル、色を指定します。

3.6 ボーダーショートハンド : border プロパティ

- 初期値 : プロパティを参照 ●適用対象 : すべての要素 ●値の継承 : しない
- ボックスの四辺すべてのボーダーの幅、スタイル、色を指定します。

3.7 角丸 : border-radius プロパティ

- 初期値 : 0 ●適用対象 : すべての要素 ●値の継承 : しない
- ボーダーの角を丸くするには、border-radius プロパティで丸め半径を指定します。

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

'border-radius' (丸め罫) を指定します。

四つの角の丸め半径を次のプロパティを使って個別に指定もできます。border-top-left-radius, border-top-right-radius, border-bottom-left-radius, border-bottom-right-radius

```
<p style="border-top-left-radius: 2mm; /* 左上 */
border-top-right-radius: 5mm; /* 右上 */
border-bottom-right-radius: 2cm 1cm; /* 右下 (水平方向, 垂直方向) */
border-bottom-left-radius: 4cm 2cm; /* 左下 (水平方向, 垂直方向) */
border: thin solid;
background-color: lime;
padding: 5mm">...
```

3

四つの角の丸め半径を個別に指定もできます。角の丸めを楕円にもできます。

3.8 ボックスシャドウ：box-shadow プロパティ

●初期値：none ●適用対象：すべての要素 ●値の継承：しない

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として影の色を指定します。

```
<p style="box-shadow: 8pt 6pt 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: -8pt -6pt orange, 8pt 6pt blue;
border-radius: 10pt; padding: 6pt">...
```

複数の影を指定することもできます。border-radius は box-shadow にも影響します。

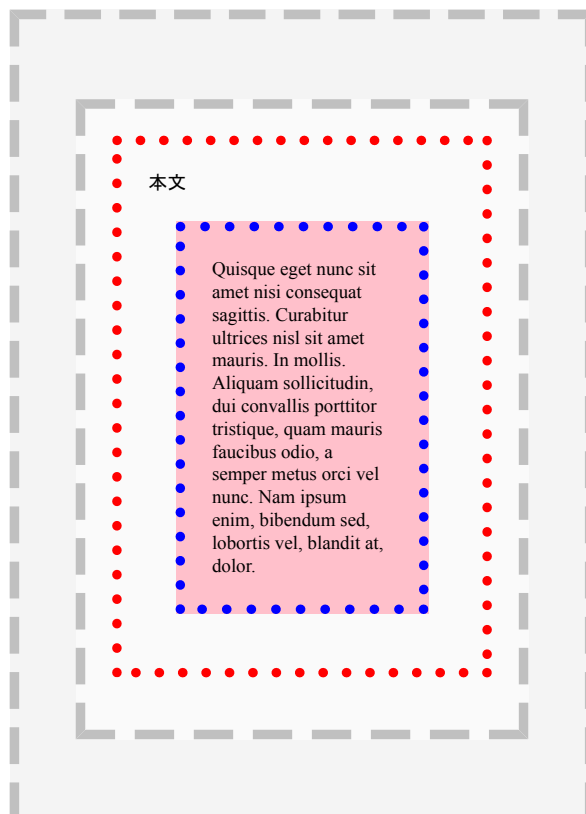
3.9 背景色：background-color プロパティ

●初期値：transparent ●適用対象：すべての要素 ●値の継承：しない

ボックスの背景色を指定するには、background-color プロパティを使います。初期値の transparent (透明) は、親要素などの内容が透けて見える状態です。

背景色が適用される範囲は、通常のボックスについてはコンテンツ、パディング、ボーダーです。マージンは常に透明で、背景色は適用されません。

- ページボックスの全面に背景色が適用されます。
- ルート要素のボックスは、コンテンツ、パディング、ボーダー、マージンに背景色が適用されます。



背景色の適用範囲

3.10 背景画像：background-image プロパティ

- 初期値：none ●適用対象：すべての要素 ●値の継承：しない
- 背景画像を指定するには、background-image プロパティを使います。画像は背景色の上に表示されます。
- none：画像は使用されません。
 - <uri>：URL で画像ファイルを指定します。

3.11 背景画像の繰り返し：background-repeat プロパティ

●初期値：repeat ●適用対象：すべての要素 ●値の継承：しない

background-repeat は、背景イメージを繰り返し使うかどうか、どのようにして繰り返すかを指定します。

- repeat：縦横に背景画像を繰り返して表示します。
- repeat-x：横方向にのみ背景画像を繰り返して表示します。
- repeat-y：縦方向にのみ背景画像を繰り返して表示します。
- no-repeat：背景画像を一回だけ表示して繰り返しません。

3.12 背景画像の位置：background-position プロパティ

●初期値：0% 0% ●適用対象：すべての要素 ●値の継承：しない

background-position プロパティは、背景画像の表示開始位置を指定します。水平位置と垂直位置の一方または両方を指定できます。指定した値がひとつであれば、ふたつめの値は center とみなされます。指定した値がふたつあれば、最初の値は横方向の位置を表し、2番目の値は縦方向の位置を表します。数値には負の値も指定できます。

横方向の位置を次のように指定します。

- <percentage>：背景画像が表示される領域の左上からの横方向の距離を、%で指定します。
- <length>：背景画像が表示される領域の左上からの横方向の距離を、数値に px などの単位をつけて指定します。
- left：0%と等価。
- center：50%と等価。
- right：100%と等価。

縦方向の位置を次のように指定します。

- <percentage>：背景画像が表示される領域の左上からの縦方向の距離を、%で指定します。
- <length>：背景画像が表示される領域の左上からの縦方向の距離を、数値に px などの単位をつけて指定します。
- top：0%と等価。
- center：50%と等価。
- bottom：100%と等価。

第4章 段落設定

4.1 行揃え

4.1.1 行揃え：text-align プロパティ

●初期値：文字の表示方向に依存 ●適用対象：ブロック要素 ●値の継承：する

ブロックレベル要素内のテキストを左揃え、右揃え、中央揃え、均等配置にします。text-align プロパティは、CSS3 で start, end の値が拡張されています。start と end は writing-mode プロパティの指定によって左右への対応付けが変わります。左から右に書く指定では start は左に、右から左に書く指定では start は右になります。end はその逆です。

left：左揃え


Antenna House Formatter provides additional features for optimal formatting, including: custom-developed MathML 3, CGM, and SVG rendering; baseline grids; PANTONE® spot colors; and additional properties for controlling Japanese layout.

right：右揃え

Antenna House Formatter provides additional features for optimal formatting, including: custom-developed MathML 3, CGM, and SVG rendering; baseline grids; PANTONE® spot colors; and additional properties for controlling Japanese layout.

center：中央揃え

Antenna House Formatter provides additional features for optimal formatting, including: custom-developed MathML 3, CGM, and SVG rendering; baseline grids; PANTONE® spot colors; and additional properties for controlling Japanese layout.

inside：ノド側に揃え 

Antenna House Formatter provides additional features for optimal formatting, including: custom-developed MathML 3, CGM, and SVG rendering; baseline grids; PANTONE® spot colors; and additional properties for controlling Japanese layout.

outside：小口側に揃え 

Antenna House Formatter provides additional features for optimal formatting, including: custom-developed MathML 3, CGM, and SVG rendering; baseline grids; PANTONE® spot colors; and additional properties for controlling Japanese layout.

start : 先頭揃え

Antenna House Formatter provides additional features for optimal formatting, including: custom-developed MathML 3, CGM, and SVG rendering; baseline grids; PANTONE® spot colors; and additional properties for controlling Japanese layout.

end : 末尾揃え

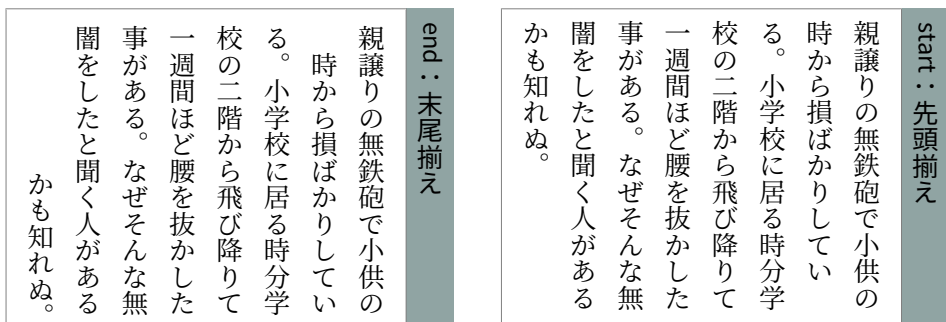
Antenna House Formatter provides additional features for optimal formatting, including: custom-developed MathML 3, CGM, and SVG rendering; baseline grids; PANTONE® spot colors; and additional properties for controlling Japanese layout.

justify : 均等配置

Antenna House Formatter provides additional features for optimal formatting, including: custom-developed MathML 3, CGM, and SVG rendering; baseline grids; PANTONE® spot colors; and additional properties for controlling Japanese layout.

4

縦書きに `text-align: start`; または `text-align: end`; を指定すると、テキストを上下に揃えます。



4.1.2 最終行の配置 : `text-align-last` プロパティ

- 初期値 : start
- 適用対象 : すべての要素
- 値の継承 : する

`text-align-last` は、ブロックの最後の行または強制改行の直前の行の配置を指定します。

`text-align-last: left` 最終行の左揃え

AH Formatter は、CSS 2.1 および CSS3 に対応しています。CSS3 のページモデルにも対応しているため、高品質なページ組版が実現できます。HTML/XHTML については、フレームなどの一部 HTML を除き、組版できます。

test-align-last: right 最終行の右揃え

AH Formatter は、CSS 2.1 および CSS3 に対応しています。CSS3 のページモデルにも対応しているため、高品質なページ組版が実現できます。HTML/XHTML については、フレームなどの一部 HTML を除き、組版できます。

text-align-last: center 最終行の中央揃え

AH Formatter は、CSS 2.1 および CSS3 に対応しています。CSS3 のページモデルにも対応しているため、高品質なページ組版が実現できます。HTML/XHTML については、フレームなどの一部 HTML を除き、組版できます。

text-align-last: inside 最終行のノド側揃え

AH Formatter は、CSS 2.1 および CSS3 に対応しています。CSS3 のページモデルにも対応しているため、高品質なページ組版が実現できます。HTML/XHTML については、フレームなどの一部 HTML を除き、組版できます。

text-align-last: outside 最終行の小口側揃え

AH Formatter は、CSS 2.1 および CSS3 に対応しています。CSS3 のページモデルにも対応しているため、高品質なページ組版が実現できます。HTML/XHTML については、フレームなどの一部 HTML を除き、組版できます。

text-align-last: start 最終行の先頭揃え

AH Formatter は、CSS 2.1 および CSS3 に対応しています。CSS3 のページモデルにも対応しているため、高品質なページ組版が実現できます。HTML/XHTML については、フレームなどの一部 HTML を除き、組版できます。

text-align-last: end 最終行の末尾揃え

AH Formatter は、CSS 2.1 および CSS3 に対応しています。CSS3 のページモデルにも対応しているため、高品質なページ組版が実現できます。HTML/XHTML については、フレームなどの一部 HTML を除き、組版できます。

text-align-last: justify 最終行の均等配置（両端揃え）

AH Formatter は、CSS 2.1 および CSS3 に対応しています。CSS3 のページモデルにも対応しているため、高品質なページ組版が実現できます。HTML/XHTML については、フレームなどの一部 HTML を除き、組版できます。

end: 最終行の末尾揃え
親譲りの無鉄砲で小供の時から損ばかりしている。小学校に居る時分学校の二階から飛び降りて一週間ほど腰を抜かした事がある。なぜそんな無闇をしたと聞く人があるかも知れぬ。別段深い理由でもない。

start: 最終行の先頭揃え
親譲りの無鉄砲で小供の時から損ばかりしている。小学校に居る時分学校の二階から飛び降りて一週間ほど腰を抜かした事がある。なぜそんな無闇をしたと聞く人があるかも知れぬ。別段深い理由でもない。

4.1.3 先頭揃え: text-align-first プロパティ

●初期値: relative ●適用対象: すべての要素 ●値の継承: する

text-align-first は、ブロックの最初の行または強制改行の直後の行の配置を指定します。行ブロックに text-align-first が指定されている場合は text-align-last よりも優先されます。

- relative: 何もしない
- start: 先頭揃え
- end: 末尾揃え
- center: 中央揃え
- inside: ノド側揃え
- outside: 小口側揃え
- justify: 均等配置 (両端揃え)
- left: 左揃え
- right: 右揃え

4.1.4 上付き文字・下付き文字がある行の高さ: -ah-line-height-shift-adjustment プロパティ

●初期値: consider-shifts ●適用対象: インラインレベルの要素 ●値の継承: する

行に上付き文字と下付き文字が含まれていると、行送りが不規則になることがあります。上付き文字と下付き文字が行送りに影響しないようにするには、-ah-line-height-shift-adjustment: disregard-shifts; を指定します。ただし、画像、文字ベースライン位置の異なるフォント (和文と欧文など) が混在する場合、大きな文字の場合、行送りが不規則になる可能性があります。

- consider-shifts: 文字のずれた位置が、行の高さを決定するために使用されます。
- disregard-shifts: 文字のずれなかった位置が、行の高さを決定するために使用されます。

-ah-line-height-shift-adjustment: consider-shifts;

行に上付き文字 ABC や下付き文字 abc が含まれていると、行送りが不規則になることがあります。上付き文字と下付き文字が行送りに影響しないようにするには、`-ah-line-height-shift-adjustment: disregard-shifts;` を指定します。ただし、画像、文字ベースライン位置の異なるフォント（和文と欧文など）が混在する場合、大きな文字の場合、行送りが不規則になる可能性があります。

-ah-line-height-shift-adjustment: disregard-shifts;

行に上付き文字 ABC や下付き文字 abc が含まれていると、行送りが不規則になることがあります。上付き文字と下付き文字が行送りに影響しないようにするには、`-ah-line-height-shift-adjustment: disregard-shifts;` を指定します。ただし、画像、文字ベースライン位置の異なるフォント（和文と欧文など）が混在する場合、大きな文字の場合、行送りが不規則になる可能性があります。

4.1.5 行送りの調整：-ah-line-stacking-strategy プロパティ

●初期値：line-height ●適用対象：ブロック要素 ●値の継承：する

行の中に大きな文字、上付き文字や下付き文字、画像が含まれる場合や、文字ベースライン位置の異なるフォント（和文と欧文など）が混在する場合、行送りは含まれる文字に依存してしまい、その結果、行送りが不規則になることがあります。`-ah-line-stacking-strategy: font-height;` を指定すると、行送りを一定にできます。

- `line-height`：CSS スタイルの行ボックス送りで、半行間（ハーフリーディング）は行エリアに含まれています。
- `font-height`：行送り幅が一定の配置。ブロックエリアのフォントの高さ+前後のハーフリーディング幅により行を配置します。
- `max-height`：行間が一定の配置。行の高さは行エリアのフォントの高さとその行エリア内のインラインエリアをすべて含む高さになります。

-ah-line-stacking-strategy: line-height;

行の中に大きな文字や上付き文字 ABC 下付き文字 abc、画像が含まれる場合や、文字ベースライン位置の異なるフォント（和文と欧文など）が混在する場合、行送りが含まれる文字に依存してしまい、その結果、行送りが不規則になります。`-ah-line-stacking-strategy: font-height;` を指定すると、行送りを一定にできます。

-ah-line-stacking-strategy: font-height;

行の中に 大きな文字や 上付き文字 の結果、行送りが不規則になります。-ah-ABC 下付き文字 abc, 画像が含まれる場合 line-stacking-strategy: font-height; を指定すると、行送りを一定にでき、文字ベースライン位置の異なるフォント (和文と欧文など) が混在する場合、行送りが含まれる文字に依存してしまい、送りが含まれる文字に依存してしまい、送

-ah-line-stacking-strategy: max-height;

行の中に 大きな文字や 上付き文字 の結果、行送りが不規則になります。-ah-ABC 下付き文字 abc, 画像が含まれる場合 line-stacking-strategy: font-height; を指定すると、行送りを一定にでき、文字ベースライン位置の異なるフォント (和文と欧文など) が混在する場合、行送りが含まれる文字に依存してしまい、送

4.2 行グリッド

-ah-line-stacking-strategy は、単一ブロック内の行に影響します。複数のブロックに跨っている行送りには影響しません。複数のブロックに跨っている行送りは、-ah-baseline-grid を使用して一貫させられます。ただし、指定した行グリッドと揃わない行エリアもあります。以下に例を示します。

- 見出しは、テキストより大きなフォントサイズで表示されることが多いです。見出しの行の高さが (他の行の高さの) 整数倍になるように見出しの margin-top, margin-bottom を調節すると、テキストの一貫した行の高さを維持できる場合があります。しかし、見出しが複数行に跨っている場合または見出しの直後に別の見出しが続く場合、失敗する場合があります。
- 画像は整数の行の高さになることはほとんどありません。
- 表セルのボーダーとパディングは、表のテキストの通常の行間に影響を与えます。

行グリッドと揃わないコンテンツを持つブロックは、-ah-baseline-block-snap を使用して、ブロック全体をグリッドに揃えることができます。

行グリッドを使わない場合

前文

人類社会のすべての構成員の固有の尊厳と平等で譲ることのできない権利とを承認することは、世界における自由、正義及び平和の基礎であるので、

- 人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為をもたらし、言論及び信仰の自由が受けられ、恐怖及び欠乏のない世界の到来が、一般の人々の最高の願望として宣言されたので、

行グリッドを使った場合

前文

人類社会のすべての構成員の固有の尊厳と平等で譲ることのできない権利とを承認することは、世界における自由、正義及び平和の基礎であるので、

- 人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為をもたらし、言論及び信仰の自由が受けられ、恐怖及び欠乏のない世界の到来が、一般の人々の最高の願望として宣言されたので、

4

4.2.1 行グリッドの指定：-ah-baseline-grid プロパティ

- 初期値：normal ●適用対象：ブロックコンテナ ●値の継承：しない

行グリッドを設定、あるいは解除します。行グリッドが設定されたエリア内にある行は、行グリッドのベースライン位置に揃えられます。

- normal：行グリッドを設定することも解除することもしません。
- none：行グリッドを解除し、コンテンツは行グリッドに揃えられません。
- new：新しい行グリッドを設定します。
- root：ルート要素で定義される行グリッドを設定します。ルート要素の font と line-height 設定が行グリッドの設定に利用されます。

4.2.2 ブロックを行グリッドに揃える：-ah-baseline-block-snap プロパティ

- 初期値：auto border-box ●適用対象：baseline-grid: new または baseline-grid: root の block-level 要素 ●値の継承：しない

通常の間行以外のブロック（見出しや図や表）をどのように行グリッドに揃えるかを指定します。

- none：ブロックを行グリッドに揃えません。
- auto：段の先頭では before、最後では after、それ以外では center と同じです。
- before：ブロックの前側を行グリッド上の上端 (text-before-edge) ラインに揃えます。
- after：ブロックの後側を行グリッド上の下端 (text-after-edge) ラインに揃えます。

ハイフネーション処理を有効にするには、hyphens プロパティを auto にします。ハイフネーション処理は、言語固有のハイフネーションアルゴリズムによって決定され、各言語ごとに処理されます。この例では、lang を指定しています。

```
.Hyphenated {  
  hyphens: auto;  
}  
<div class="Hyphenated" lang="en">  
  <p>CSS is widely used in browsers, editors, and other ...
```

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 pag-
ed 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. CSS3, although still under development by the 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.

4

本文でハイフネーションが有効になっている場合は、見出しなど、ハイフンを追加することで、テキストの見栄えを損なう可能性のある他の文脈では、ハイフネーションを無効にすることをお勧めします。例えば、キャプション、表見出し、表中のテキストなどの場合です。

```
body:lang(en) {  
  hyphens: auto;  
}  
  
h1, h2, h3, h4, h5, h6 {  
  hyphens: manual;  
}
```

4.4.2 ハイフン分割の前の最小文字数：hyphenate-before プロパティ

●初期値：auto ●適用対象：すべての要素 ●値の継承：する

ハイフネーションによる行分割の前の最小文字数（標準は2）を指定します。

例えば、"hyphen" という6文字の単語は、"hy-phen" とハイフネーションされる可能性があります。hyphenate-before: 3; と指定すると、ハイフン分割の前の文字数が3文字以上の単語でないとハイフネーションされないため、"hyphen" がハイフネーションされることはなくなります。

4.4.3 ハイフン分割の後の最小文字数：hyphenate-after プロパティ

●初期値：auto ●適用対象：すべての要素 ●値の継承：する

ハイフネーションによる行分割の後の最小文字数（標準は2）を指定します。

例えば，“hyphen”という6文字の単語は，“hy-phen”とハイフネーションされる可能性があります。hyphenate-after: 5;と指定すると、ハイフン分割の後の文字数が5文字以上の単語でないとハイフネーションされないで、“hyphen”がハイフネーションされることはなくなります。

4.5 ブロックの縦書き：writing-mode: tb-rl;

4

ブロックを縦書きにするには、ブロック要素にwriting-mode: tb-rl;を指定します。ブロックの縦書きでは、ページの進行方向は横書きと同じく左から右のままになります。

```
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。AH Formatterは縦書きにも対応してこのように、部分的にブロックを縦書きにすることも、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 and top-to-bottom) の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12月8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。

第5章 段組

5.1 段数：column-count プロパティ

- 初期値：auto
- 適用対象：ブロック要素
- 値の継承：しない

ブロックの段数を指定します。

- auto：他のプロパティの値に基づき段数を決めます。例えば、column-width の length の値などです。
- <integer>：段数の上限値。column-count と column-width の両方が auto 以外の値の時、それが段数の最大値になります。

段数 column-count: 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 を指定する代わりに段幅 column-width を指定

する方法があります。全体の幅と段幅により自動的に段数が決まります。

段間罫を指定する column-rule はショーハンドであり、次のように個別に指定もできます。

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

5

5.2 段幅：column-width プロパティ

- 初期値：auto
- 適用対象：ブロック要素
- 値の継承：しない

段組の要素の段幅を指定します。

- auto：他のプロパティの値に基づき段幅を決めます。例えば、column-count の integer の値などです。
- <length>：段幅の最適値。

column-width:を 12em 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.MultiColW {  
  column-width: 12em;  
  column-gap: 1em;  
  column-rule:  
    solid 1pt;  
}
```

段組の指定には、column-count を指定する代わりに column-width を指定します。

5.3 段数，幅：columns プロパティ

column-width, column-count を指定するためのショートハンドです。値が省略されたときは、初期値が設定されます。

5.4 段抜き：column-span プロパティ

●初期値：none ●適用対象：ブロック要素（フロートと絶対配置された要素を除く）

●値の継承：しない

段抜きする要素の段数を指定します。

- none：段抜きしません。
- all：すべての段数を段抜きします。

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

いろはにほへとちりぬるを

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

5.5 段間：column-gap プロパティ

●初期値：normal ●適用対象：マルチコラム要素 ●値の継承：しない

段間の幅を指定します。

- normal：1em と等価。
- <length>：段間の幅。

5.6 段間罫：column-rule プロパティ

●初期値：各プロパティを参照 ●適用対象：マルチコラム要素 ●値の継承：しない

段間罫の幅，線種，色をまとめて指定します。

5.7 段間罫の種類：column-rule-style プロパティ

- 初期値：none
- 適用対象：マルチコラム要素
- 値の継承：しない

段間罫の種類を指定します。ボーダーのスタイルを指定する場合と同じ値を使用できます。詳細については 3.2 ボーダーのスタイル：border-style プロパティ（ページ 11）を参照してください。

5.8 段間罫の幅：column-rule-width プロパティ

- 初期値：medium
- 適用対象：マルチコラム要素
- 値の継承：しない

段間罫の幅を指定します。ボーダーの幅を指定する場合と同じ値を使用できます。詳細については 3.3 ボーダーの幅：border-width プロパティ（ページ 12）を参照してください。

5.9 段間罫の色：column-rule-color プロパティ

- 初期値：現在の色
- 適用対象：マルチコラム要素
- 値の継承：しない

段間罫の色を指定します。第 11 章 色指定（ページ 57）を参照してください。

5.10 段間罫の長さ：-ah-column-rule-length プロパティ

- 初期値：100%
- 適用対象：マルチコラム要素
- 値の継承：しない

段間罫の長さを指定します。

```
column-rule-length: 60%; は段の  
高さの 60%の長さの段間罫を中央に縦に  
配置します.
```

```
div.MultiCol {  
  column-rule-length: 60%;  
}
```

5.11 段間罫の揃え：-ah-column-rule-align プロパティ

- 初期値：center
- 適用対象：マルチコラム要素
- 値の継承：しない

段内の段間罫の縦方向の配置を揃えます。

- before：段間罫の先頭を段の先頭に揃えて配置します。
- center：段間罫を段の中央に揃えて配置します。
- after：段間罫の末尾を段の末尾に揃えて配置します。

`-ah-column-rule-length: 60%;`
と `-ah-column-rule-align: after;`
は、段の高さの 60% の長さの段間罫を段の
末尾に揃えて配置します。

```
div.MultiCol {  
  -ah-column-rule-length: 60%;  
  -ah-column-rule-align: after;  
}
```

5.12 段間罫表示：-ah-column-rule-display プロパティ

●初期値：gap ●適用対象：マルチコラム要素 ●値の継承：しない

`-ah-column-rule-display` は、段間のないところにも段間罫を引くかどうか指定します。

- `gap`：存在する段と段の間にだけ段間罫を引きます。
- `end`：存在する段の `end` 側に段間罫を引きます。
- `all`：ページ末などで、存在しない段の間にもすべて段間罫を引きます。

5

`-ah-column-rule-display:`
`end` は、存在する段の `end`
側に段間罫を引きます。

```
div.MultiCol3 {  
  -ah-column-rule-display: end;  
}
```

改ページ・ページコンテンツの維持

ページ組版では、当然のことながらブラウザとは異なりページの分割が発生します。CSSは、要素のコンテンツをひとつのページにまとめるかどうか、コンテンツの前後に改ページ設定するかどうかを制御するための複数のプロパティを提供しています。

6.1 改ページの制御

6.1.1 改ページ：page-break-before/page-break-after プロパティ

- 初期値：auto ●適用対象：ブロック要素 ●値の継承：しない
- auto：改ページを強制も禁止もしません。
- always：常に改ページを強制します。
- avoid：常に改ページを避けます。
- left：改ページを強制して、次のページが左側のページになるようにします。
- right：改ページを強制して、次のページが右側のページになるようにします。

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

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

値を avoid にすると、指定した要素の前後で改ページを禁止できます。

```
/* 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;
}
```

6.1.2 改ページの禁止：page-break-inside プロパティ

- 初期値：auto ●適用対象：ブロック要素 ●値の継承：する
- 値を avoid にすると、指定した要素内での改ページを禁止できます。
- auto：改ページを強制も禁止もしません。
- avoid：常に改ページを避けます。

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

6.1.3 左右どちらかのページから始める

章の始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。

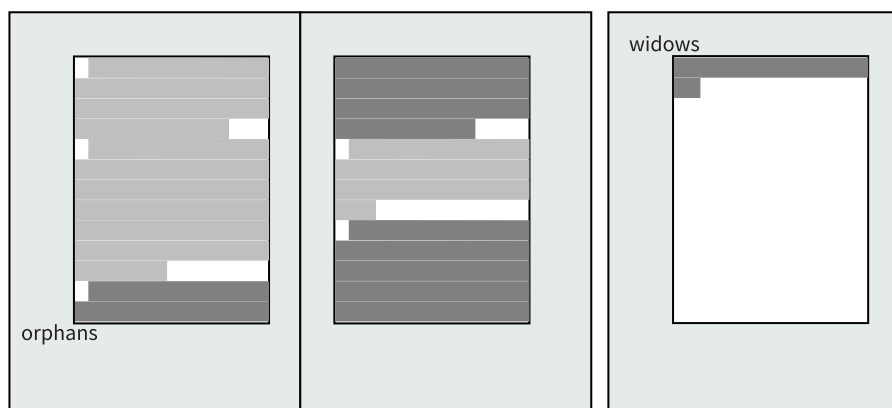
```
/*h2見出しが常に右ページに来るように改ページ*/  
h2 {  
  page-break-before: right;  
}
```

6.2 改ページ前後の最小行数：orphans/widows プロパティ

●初期値：2 ●適用対象：ブロック要素 ●値の継承：する

orphans は、ページの下部に残さなければならない段落先頭の最小行数を指定します。
widows は、ページの上部に残さなければならない段落終わりの最小行数を指定します。

- <integer>：改ページ前後の最小数行



orphan と widow の行

文字設定

7.1 フォント指定

7.1.1 フォント：font プロパティ

- 初期値：プロパティを参照
- 適用対象：すべての要素
- 値の継：する

このプロパティは、font-style, font-variant, font-weight, font-size, line-height, font-family の6個のプロパティをまとめて指定するためのショートハンドです。

フォントプロパティを指定すると、各プロパティの値は初期値にいったんリセットされます。そして、フォントのショートハンドで明示的に指定したプロパティについてその値に設定されます。font プロパティに指定する順序は以下のとおりです。

1. font-style, font-variant, font-weight の値は省略可能で、この三つの中での順序は問いません。
2. font-size 値は省略できません。
3. line-height の値は省略可能です。font-size と line-height の値は“/”で区切ります。(例：14q/25q)
4. font-family は省略できません。複数のフォントファミリーをカンマで区切って指定できます。

```
/* 文書タイトル */  
.CoverPage h1 {  
  font: bold 30pt "源ノ角ゴシック JP", sans-serif;  
  text-align: center;  
}
```

7.1.2 フォントサイズ：font-size プロパティ

- 初期値：medium
- 適用対象：すべての要素
- 値の継承：する

フォントのサイズを指定します。

```
/* 本文 */  
body {  
  font-size: 12pt;  
  line-height: 1.5;  
  font-family: "Source Sans Pro", "源ノ角ゴシック JP", sans-serif;  
}
```

7.1.3 フォントの種類：font-family プロパティ

●初期値：ユーザエージェントに依存⁶⁾ ●適用対象：すべての要素 ●値の継承：する
フォントの種類を指定します。フォントの種類は、フォント名かキーワードで指定します。キーワードは以下のいずれかで、ゼネリックフォントファミリーと呼ばれます。

- sans-serif：装飾はほとんどまたは一切ない先端を持つ傾向のあるフォントです。Helvetica などのひげ飾りなしのプレーンフォントです。日本語ではゴシック系のフォントです。
- serif：Times などのひげ飾り付きのフォントです。日本語では明朝系のフォントです。
- monospace：等幅フォントです。文字幅が等しいフォントです。
- fantasy：装飾のあるフォントです。
- cursive：文字をつなげて書く手書きのような特性を持つ筆記体フォントです。

"Times New Roman" のようにフォント名にスペースが含まれる場合は、ダブルクォーテーションまたはシングルクォーテーションで囲みます。

カンマで区切って複数のフォントを指定できます。ユーザ環境で利用できるフォントが記述順に選ばれます。AH Formatter は、文字列をスクリプトの種類ごとに区切り、スクリプト単位で最初に指定されているフォントから割り当てていきます。例えば、日本語フォントの多くはラテンスクリプトをサポートしますので、font-family: "源ノ角ゴシック JP", Helvetica; という指定の場合、ラテンスクリプトと日本語にも "源ノ角ゴシック JP" が割り当てられる結果となります⁷⁾。次のようにラテンフォントを先に指定することで、MS ゴシックの前に Helvetica が採用されるようになります。

```
font-family: Helvetica, "源ノ角ゴシック JP", sans-serif;
```

指定したフォントがシステムに存在しないときのために、serif, sans-serif などのゼネリックフォントファミリーを最後に指定するとよいでしょう。

7.1.4 フォントの太さ：font-weight プロパティ

●初期値：normal ●適用対象：すべての要素 ●値の継承：する
フォントの太さを指定します。

- normal：標準（数値指定では 400）。
- bold：太字（数値指定では 700）。
- lighter：1 段階（数値指定では 100）細くする。
- bolder：1 段階（数値指定では 100）太くする。

6) serif. 初期値の serif は変更できます。

7) このフォントの選択方法は、1 文字 1 文字に対してどのフォントにグリフがあるか調べる方法に変更もできません。

- 数値指定：100, 200, 300, 400, 500, 600, 700, 800, 900 のいずれか。100 が最も細く、900 が最も太い。

7.1.5 イタリック体・斜体：font-style プロパティ

●初期値：normal ●適用対象：すべての要素 ●値の継承：する
フォントをイタリック体や斜体に指定します。

- normal：標準（立体）。
- italic：イタリック体。
- oblique：斜体。

italic は、初めから斜めに傾けてデザインされたフォントですが、oblique は単純に元のフォントを傾けたものです。italic 指定したフォントにイタリック体がない場合は、そのフォントを斜めに傾けて表示します。

AH Formatter では、oblique を指定しても italic と同じように処理します。

日本語フォントには、イタリック体はほとんど存在しないので、italic を指定しても oblique を指定しても同じ表示になります。（日本語フォントでも、欧字に対してはイタリック体を持っているものは多いです。）

```
font-style: italic;
```

CSS によるページ組版入門

```
font-style: oblique;
```

CSS によるページ組版入門

italic と oblique

7.1.6 スモールキャピタル・フォント機能：font-variant プロパティ

●初期値：normal ●適用対象：すべての要素 ●値の継承：する

フォントをスモールキャピタルにするときに指定します。英小文字などでのみ有効です。

- normal：標準（何もしない）。
- small-caps：スモールキャピタルにする。
- <keyword>：OpenType 機能に対応する oldstyle-nums など、他にも多数のキーワードがあります。

これらは OpenType フォントの機能を使って表示されます。OpenType フォントが対応していない場合、スモールキャピタルは AH Formatter によってエミュレートされます。

7.2 追加のフォント：@font-face ルール

AH Formatter の設定を変更せずに、また、オペレーティングシステムにフォントをインストールせずに、文書内で利用するフォントを追加指定できます。以下の記述子の指定が可能です。

- `font-family`：CSS フォントファミリー名の一致に使用する名前。フォントデータからファミリー名を上書きします。
- `src`：フォントリソースの位置⁸⁾。存在しているローカルフォントの別名が可能です。
- `font-style`：フォントが一致する場合に使用する任意のフォントのスタイル。 `italic` や `normal` などが使われます。既定値は `normal` です。
- `font-weight`：フォントが一致する場合に使用する任意のフォントの太さ。 `font-weight` プロパティと同じ値を使用します。ただし、相対キーワードの `bolder`、`lighter` は利用できません。既定値は `normal` です。

The Quick Brown Fox

```
@font-face {  
  font-family: font-face-example;  
  src: url(UglyQua.ttf);  
}  
  
p { font-family: font-face-example; }
```

7

7.3 行の高さ：line-height プロパティ

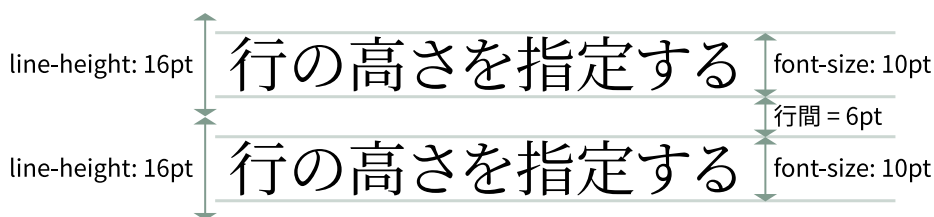
●初期値：normal ●適用対象：すべての要素 ●値の継承：する

行の高さを指定します。 `normal` は標準の行送りで、AH Formatter では、1.2 に初期設定されています⁹⁾。

数値だけを指定すると、その数値にフォントのサイズを掛けた値が行の高さとなります。1.5 と指定すると、150%や 1.5em と指定したのと同じ行の高さになります。ただし、`line-height` プロパティが継承されるときに、数値だけの指定の場合はその数値が継承され、フォントのサイズが変われば行の高さもそれに応じて変わります。

8) 現在、AH Formatter は単一のリソースのみ対応しています。

9) 初期設定値の 1.2 は変更できます。



行の高さと行間

7.4 縦方向の揃え位置：vertical-align プロパティ

●初期値：baseline ●適用対象：インラインと表のセル要素 ●値の継承：しない
縦方向の揃え位置や上付き文字、下付き文字を指定できます。

- normal：標準の行揃え。
- baseline：親要素のベースラインに揃える。
- sub：下付き文字。
- super：上付き文字。
- top：上端に揃える。
- text-top：テキストの上端に揃える。
- middle：中央に揃える。
- bottom：下端に揃える。
- text-bottom：テキストの下端に揃える。
- <percentage>：line-height プロパティの値に対する割合をパーセントで指定する。
- 数値指定：数値に単位を付けて指定する。

7

7.5 下線・上線・打ち消し線

7.5.1 文字の修飾 ショートハンド：text-decoration プロパティ

●初期値：none ●適用対象：すべての要素 ●値の継承：しない

下線・上線・打ち消し線のタイプ、色、種類を指定します。text-decoration は、ひとつの宣言で text-decoration-line, text-decoration-color, text-decoration-style を指定するためのショートハンドです。text-decoration-line 要素のみを設定する text-decoration プロパティは、CSS2 と後方互換性があります。

```
span {
  text-decoration: underline dotted cmyk(0,1,1,0); /* 下線 点線 赤 */
}
```

7.5.2 下線・上線・打ち消し線：text-decoration-line プロパティ

●初期値：none ●適用対象：すべての要素 ●値の継承：しない

テキストに下線・上線・打ち消し線を指定します。同時に複数の指定をするときは、スペースで区切って任意の順序で値を指定します。

- none：テキストに線は付かない。
- underline：テキストに下線が付く。
- overline：テキストに上線が付く。
- line-through：テキストに打ち消し線が付く。

```
/* 章見出し */
.Chapter h2 {
  text-decoration-line: underline overline;
}
```

7.5.3 線の種類：text-decoration-style プロパティ

●初期値：solid ●適用対象：すべての要素 ●値の継承：しない

下線・上線・打ち消し線の種類を指定します。ボーダーのスタイルを指定する場合と同じ値を使用できます。詳細については 3.2 ボーダーのスタイル：border-style プロパティ (ページ 11) を参照してください。

```
span.dotted {
  text-decoration-style: dotted; /* 線の種類を点線にする */
}
...
<p>dotted：点線 <span class="dotted"> </span></p>
```

dotted：点線

7.5.4 線の色：text-decoration-color プロパティ

●初期値：文字の色 ●適用対象：すべての要素 ●値の継承：しない

下線・上線・打ち消し線の色を指定します。第 11 章 色指定 (ページ 57) を参照してください。

```
span {
  text-decoration-color: cmyk(0,1,1,0); /* 線の色を赤にする */
}
```

7.5.5 線の幅：-ah-text-line-width プロパティ

●初期値：auto ●適用対象：すべての要素 ●値の継承：しない

下線・上線・打ち消し線の幅を指定します。

第8章

MathML と SVG グラフィクス

8.1 MathML

AH Formatter は、W3C による Mathematical Markup Language (MathML) 3.0 Second Edition を独自開発したエンジンにより描画できます。そのため、PDF 中の数式を高い解像度で描画できます。

MathML については、Antenna House 出版の MathML 数式組版入門 https://www.antenna.co.jp/AHF/ahf_publication/index.html#MathML を参照してください。

MathML を使った組版例

2 次方程式 $ax^2 + bx + c = 0$ の解は、 $D = b^2 - 4ac$ とおくと、

$$D \geq 0 \text{ のとき, } x = \frac{-b \pm \sqrt{D}}{2a}$$

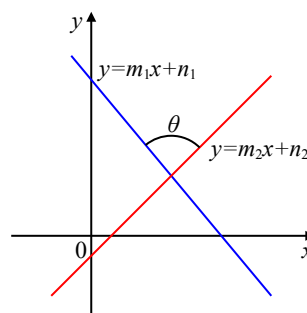
$D < 0$ のとき、実数の解なし。

```
<p>2次方程式
<m:math xmlns:m="http://www.w3.org/1998/Math/MathML"
  lang="en">
  <m:mstyle displaystyle="true" scriptminsize="1pt"
    scriptsizemultiplier="0.6">
    <m:mrow>
      <m:msup>
        <m:mrow>
          <m:mi>a</m:mi>
          <m:mo>&#x2062;</m:mo>
          <m:mi>x</m:mi>
        </m:mrow>
        <m:mn>2</m:mn>
      </m:msup>
    </m:mrow>
  </m:mstyle>
  の解は,
  ...
  ...
```

MathML を使った組版例

ふたつの直線の方程式を $y = m_1x + n_1$ と $y = m_2x + n_2$ とすると、この2直線のなす角 θ は、

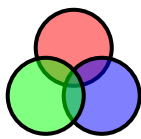
$$\tan \theta = \pm \frac{m_2 - m_1}{1 + m_2 m_1} \quad (0^\circ \leq \theta < 180^\circ)$$



8.2 SVG グラフィクス

AH Formatter は、W3C による Scalable Vector Graphics (SVG) 1.1 を独自開発したエンジンにより描画します。そのため、PDF 中への SVG イメージを高い解像度で描画できます。

SVG のベクタグラフィクス



を埋め込むことができます。

<p>SVGのベクタグラフィクス

```
<s:svg xmlns:s="http://www.w3.org/2000/svg" width="70" height="65"
  viewBox="0 0 70 65">
  <s:g fill-opacity=".5" stroke="black" stroke-width="2">
    <s:circle cx="35" cy="20" r="19" fill="red"/>...
  </s:g>
</s:svg>
```

を埋め込むことができます。 </p>

8

SVG の例





第9章 画像配置

9.1 インライン画像

文章中にインライン画像を配置するには、インライン要素の `` 要素や `<object>` 要素を使います。

インライン画像

Antenna House  AH Formatter は世界中のお客様にご利用いただきながら 18 年目を迎える大容量・多言語データに最適な自動組版ソフトです。独自開発した PDF 出力エンジンで、アクセシブルなタグ付き PDF や印刷用の PDF/X、長期保存 PDF/A などさまざまな PDF 形式の出力ができます。

AH Formatter  のスタイルシートは XSL-FO と CSS のどちらでも使えます（併用はできません）。マニュアルのような複雑なレイアウトには XSL-FO が、比較的単純なレイアウトには直感的な CSS がお勧めです。

```
<p>Antenna House  Formatter は世界中のお客様にご利用いただきながら...
```

9.2 ブロック画像

`display: block;` を使用して、画像表示をインラインではなくブロックとして表示します。

`display: block;` を指定したインライン画像

Antenna House



AH Formatter は世界中のお客様にご利用いただきながら 18 年目を迎える大容量・多言語データに最適な自動組版ソフトです。独自開発した PDF 出力エンジンで、アクセシブルなタグ付き PDF や印刷用の PDF/X、長期保存 PDF/A などさまざまな PDF 形式の出力ができます。

```
<p>Antenna House  Formatter は世界中のお客様にご利用いただきながら 18年目を迎える大容量・多言語データに最適な自動組版ソフトです。独自開発した PDF出力エンジンで、アクセシブルなタグ付きPDF や印刷用の PDF/X、長期保存 PDF/A などさまざまな PDF形式の出力ができます。
```

9.3 フロート配置：float プロパティ

●初期値：none ●適用対象：すべての要素 ●値の継承：しない

画像をフロートさせて左寄せまたは右寄せにするには、float プロパティを使います。AH Formatter の float プロパティは、水平方向の float-x、垂直方向の float-y、フロートへのテキストの回り込みを指定する float-wrap、フロートを配置する基準となるエリアを指定する float-reference、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する float-move をまとめて指定するショートハンドに拡張されています。

9.3.1 サイドフロート：float: left;/float: right;

CSS2 標準のフロート機能です。float プロパティを適用した画像の次の要素は、反対側に回り込みます。

画像の左寄せ float: left;



AH Formatter は世界中のお客様にご利用いただきながら 18 年目を迎える大容量・多言語データに最適な自動組版ソフトです。独自開発した PDF 出力エンジンで、アクセシブルなタグ付き PDF や印刷用の PDF/X、長期保存 PDF/A などさまざまな PDF 形式の出力ができます。

```
<p style="text-indent: 0">  
AH Formatter は世界中のお客様にご利用いただきながら 18年目を迎える大  
容量・多言語データに最適な自動組版ソフトです。独自開発した PDF出力エンジンで、アクセシブルなタグ  
付きPDF や印刷用の PDF/X、長期保存 PDF/A などさまざまな PDF形式の出力ができます。
```

画像の右寄せ float: right;

AH Formatter は世界中のお客様にご利用いただきながら 18 年目を迎える大容量・多言語データに最適な自動組版ソフトです。独自開発した PDF 出力エンジンで、アクセシブルなタグ付き PDF や印刷用の PDF/X、長期保存 PDF/A などさまざまな PDF 形式の出力ができます。



```
<p style="text-indent: 0">  
AH Formatter は世界中のお客様にご利用いただきながら 18年目を迎える  
大容量・多言語データに最適な自動組版ソフトです。独自開発した PDF出力エンジンで、アクセシブルなタグ  
付きPDF や印刷用の PDF/X、長期保存 PDF/A などさまざまな PDF形式の出力ができます。
```


ここは、ページの先頭です。 `float: top page` を指定します。

これはページのふたつ目のトップフロートです。 `float: top page` を指定します。

画像だけでなく、文字をフロートさせることもできます。

左寄せテキスト `float: left;`

Antenna House Formatter は世界中のお客様にご利用いただきながら 18 年目を迎える大容量・多言語データに最適な自動組版ソフトです。独自開発した PDF 出力エンジンで、アクセシブルなタグ付き PDF や印刷用の PDF/X、長期保存 PDF/A などさまざまな PDF 形式の出力ができます。

右寄せテキスト `float: right;`

Formatter は世界中のお客様にご利用いただきながら 18 年目を迎える大容量・多言語データに最適な自動組版ソフトです。独自開発した PDF 出力エンジンで、アクセシブルなタグ付き PDF や印刷用の PDF/X、長期保存 PDF/A などさまざまな PDF 形式の出力ができます。 **Antenna House**

9.3.2 ページフロート : `float: top page;`/`float: bottom page;`

ページフロートはフロートを配置する基準エリアとしてページを指定します。ページの縦方向へのフロートを指定できるように拡張したものです。 `float: top page;` でページの先頭にブロックを配置できます。 `float: bottom page;` でページの末尾にブロックを配置できます。このページの先頭と末尾には、ページフロートの例がふたつずつ含まれています。

```
<div style="float: top page; border: ridge maroon; padding: 3pt; margin-bottom: 1em;">
<p>ここは、ページの先頭です。 <b>float: top page</b>を指定します。 </p>
</div>
<div style="float: bottom page; border: ridge green; padding: 3pt; margin-top: 1em;">
<p>ここは、ページの末尾です。 <b>float: bottom page</b>を指定します。 </p>
</div>
```

これはページのふたつ目のボトムフロートです。 `float: bottom page` を指定します。

ここは、ページの末尾です。 `float: bottom page` を指定します。

9.3.3 コラムフロート : float: top column;/float: bottom column;



float: top column; でコラムの先頭にブロックを配置できます。float: bottom column; でコラムの末尾にブロックを配置できます。

<p>ここは、コラムの先頭です。 float: top column;を指定します。</p>	<p>び信仰の自由が受けられ、恐怖及び欠乏のない世界の到来が、一般の人々の最高の願望として宣言されたので、人間が専制と圧迫とに対する最後の手段として反逆に訴えることがないようにするためには、法の支配によって人権保護することが肝要であるので、</p>
<p>人類社会のすべての構成員の固有の尊厳と平等で譲ることのできない権利とを承認することは、世界における自由、正義及び平和の基礎であるので、人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為をもたらし、言論及</p>	<p>ここは、コラムの末尾です。 float: bottom column;を指定します。</p>

```
<p>人類社会のすべての構成員...</p>
<div style="float: top column; border: ridge orange; padding: 3pt;">
<p>ここは、コラムの先頭です。 </p><p><b>float: top</b>を指定します。 </p>
</div>
<p>人権の無視及び軽侮が、 ...</p>
<div style="float: bottom column; border: ridge lime; padding: 3pt">
<p>ここは、コラムの末尾です。 </p><p><b>float: bottom</b>を指定します。
</p>
</div>
<p>人間が専制と圧迫とに...</p>
```

9.3.4 フロートの値の拡張 : 横書き inside, outside, center, start, end, alternate

float: inside; は、ブロックを右側のページの左側に配置するか、左側ページの右側に配置します。

float: outside; は、ブロックを右側のページの右側に配置するか、左側ページの左側に配置します。

float: center; は、段の中央にブロックを配置できます。

float: start; は、start 側にブロックを配置できます。左から右の横書きの場合 left と同じです。右から左の横書きの場合 right と同じです。

float: end; は、end 側にブロックを配置できます。左から右の横書きの場合 right と同じです。右から左の横書きの場合 left と同じです。

人類社会のすべての構成員の固有の尊厳と平等で譲ることのできない権利
float: inside 人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為
をもたらし、言論及び信仰の自由が受けられ、恐怖及び欠乏の
ない世界の到来が、一般の人々の最高の願望として宣言されたので、

人類社会のすべての構成員の固有の尊厳と平等で譲ることのできない権利
人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行 **float: outside**
為をもたらし、言論及び信仰の自由が受けられ、恐怖及び欠
乏のない世界の到来が、一般の人々の最高の願望として宣言されたので、

人類社会のすべての構成員の固有の尊厳と平等で譲ることのできない権利

float: center

人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為をもたらし、
言論及び信仰の自由が受けられ、恐怖及び欠乏のない世界の到来が、一般の
人々の最高の願望として宣言されたので、

人類社会のすべての構成員の固有の尊厳と平等で譲ることのできない権利
float: start 人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為
をもたらし、言論及び信仰の自由が受けられ、恐怖及び欠乏の
ない世界の到来が、一般の人々の最高の願望として宣言されたので、

ואיל והכרה בכבוד הטבעי אשר לכל בני משפחה האדם ובזכויותיהם השוות והבלתי נפקעות הוא יסוד
החופש, הצדק והשלום בעולם.

הואיל והכרה חיוני הוא שזכויות האדם תהיינה מוגנות בכוח שלטונו של החוק, שלא יהא
האדם אנוס, כמפלט אחרון, להשליך את יהבו על מרידה בעריצות ובדיכוי. **float: start**

人類社会のすべての構成員の固有の尊厳と平等で譲ることのできない権利
人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為を **float: end**
もたらし、言論及び信仰の自由が受けられ、恐怖及び欠乏のない
世界の到来が、一般の人々の最高の願望として宣言されたので、

ואיל והכרה בכבוד הטבעי אשר לכל בני משפחה האדם ובזכויותיהם השוות והבלתי נפקעות הוא יסוד
החופש, הצדק והשלום בעולם.

float: end

הואיל והכרה חיוני הוא שזכויות האדם תהיינה מוגנות בכוח שלטונו של החוק, שלא יהא
האדם אנוס, כמפלט אחרון, להשליך את יהבו על מרידה בעריצות ובדיכוי.

人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為をもたらし、言
人間が専制と圧迫とに対する最後の手段として反逆に訴える

float: inside

ことがないようにするためには、法の支配によって人権保護

人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為をもたらし、言

float: outside

人間が専制と圧迫とに対する最後の手段として反逆に訴える

ことがないようにするためには、法の支配によって人権保護

人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為をもたらし、言

float: center

人間が専制と圧迫とに対する最後の手段として反逆に訴えることがな
いようにするためには、法の支配によって人権保護

人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為をもたらし、言

float: start

人間が専制と圧迫とに対する最後の手段として反逆に訴えるこ

ことがないようにするためには、法の支配によって人権保護

הואיל והזולזת בזכויות האדם וביזוין הבשילו מעשים פראיים שפגעו קשה במצפונה של האנושות; ובנין
עולם, שבו ייהנו כל יצורי אנוש מחירות הדיבור והאמונה ומן החירות מפקד וממחסור, הוכרו כראש
שאיפתיו של כל אדם.

הואיל והכרח חיוני הוא שזכויות האדם תהיינה מוגנות בכוח שלטונו של החוק, שלא יהא

float: start

האדם אנוס, כמפלט אחרון, להשליך את יהבו על מרידה בעריצות ובדיכוי.

人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為をもたらし、言
人間が専制と圧迫とに対する最後の手段として反逆に訴える

float: end

ことがないようにするためには、法の支配によって人権保護

הואיל והזולזת בזכויות האדם וביזוין הבשילו מעשים פראיים שפגעו קשה במצפונה של האנושות; ובנין
עולם, שבו ייהנו כל יצורי אנוש מחירות הדיבור והאמונה ומן החירות מפקד וממחסור, הוכרו כראש
שאיפתיו של כל אדם.

float: end

הואיל והכרח חיוני הוא שזכויות האדם תהיינה מוגנות בכוח שלטונו של החוק, שלא יהא

האדם אנוס, כמפלט אחרון, להשליך את יהבו על מרידה בעריצות ובדיכוי.

float: alternate; は、段組の最初の段のときは end, 最後の段のときは start, それ以外では center が指定されているかのようにブロックを配置します。

人類社会のすべての構成員の固有の尊厳と平等で譲ることのできない権利とを承認することは、世界における自由、正義及び平和の基礎であるので、
 人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為をもたらし、言論及び信仰
 の自由が受けられ、恐怖及び欠乏のない世界の到来が、一般の人間が専制と圧迫とに
 対する最後の手段として
 反逆に訴えることがないようにするため
 には、法の支配によって人権保護することが
 肝要であるので、

9.3.5 フロートの値の拡張：縦書き top, bottom, center, before, inside, after, outside

float: left; と float: right; は縦書きの文書では無効です。

夏目漱石(なつめ そうせき、慶応三年一月五日(一八六七年二月九日)ー大正五年(一九一六年)二月九日)は、日本の小説家、評論家、英文学者。本名、金之助。『吾輩は猫である』『こゝろ』などの作品で広く知られる、森鷗外と並ぶ明治・大正時代の文豪である。江戸の牛込馬場下横町(現在の東京都新宿区喜久井町)出身。俳号は愚陀仏。

大学時代に正岡子規と出会い、俳句を学ぶ。…(中略)…当初は余裕派と呼ばれた。

「修善寺の大患」後は、『行人』『こゝろ』『硝子戸の中』などを執筆。「則天去私」(そくてんきよし)の境地に達したといわれる。晩年は胃潰瘍に悩まされ、『明暗』が絶筆となった。

フリー百科事典『ウィキペディア』より引用

float: before	<p>夏目漱石（なつめ そうせき、慶応三年一月五日（一八六七年二月九日）―大正五年（一九一六年）二月九日）は、日本の小説家、評論家、英文学者。本名、金之助。『吾輩は猫である』『こゝろ』などの作品で広く知られる、森鷗外と並ぶ明治・大正時代の文豪である。江戸の牛込馬場下横町（現在の東京都新宿区喜久井町）出身。俳号は愚陀仏。</p> <p>大学時代に正岡子規と出会い、俳句を学ぶ。</p> <p>：（中略）：当初は余裕派と呼ばれた。</p> <p>「修善寺の大患」後は、『行人』『こゝろ』『硝子戸の中』などを執筆。「則天去私」（そくてんきょし）の境地に達したといわれる。晩年は胃潰瘍に悩まされ、『明暗』が絶筆となった。</p> <p>フリー百科事典『ウィキペディア』より引用</p>	float: inside
float: after		float: outside

9.3.6 段組フロート：float: multicol; ☒

float: multicol; は、フロートを段組と段間にも段抜きをします。gr (grid)は特別な長さの単位であり、段組の段幅 (column-width) と段間 (column-gap) の両方をそれぞれ 1gr として数えます。このため、n 段抜きの指定は (2n - 1)gr となります。段幅または段間の途中までの長さの指定のために端数の指定も可能です。

float: multicol; width: 3gr;	<ul style="list-style-type: none"> ● 野蛮行為をもたらし、言論 ● 及び信仰の自由が受けら ● れ、恐怖及び欠乏のない世 ● 界の到来が、一般の人々の ● 最高の願望として宣言され ● たので、
<p>人類社会のすべての構成</p> <p>員の固有の尊厳と平等で譲</p> <p>ることのできない権利とを</p> <p>承認することは、世界にお</p>	<ul style="list-style-type: none"> ● ける自由、正義及び平和の ● 基礎であるので、 ● 人権の無視及び軽侮が、 ● 人類の良心を踏みにじった

9.4 回り込みの解除：clear プロパティ

9

- 初期値：none ● 適用対象：ブロック要素 ● 値の継承：する ● 値の継承：しない
- clear プロパティは右寄せまたは左寄せを指定したブロックへの回り込みを解除します。
- none：回り込みを解除しません。
 - left：左寄せの要素への回り込みを解除します。
 - right：右寄せの要素への回り込みを解除します。
 - both：左寄せおよび右寄せ要素への回り込みを解除します。

第 10 章 表組

10.1 表の作成方法

CSS の表モデルは、HTML4 の表モデルを元にしています。表要素を持っていない文書言語で CSS を使用する場合、文書要素を HTML4 に対応する `display` プロパティ値に割り当てることができます。

display 値	HTML の要素
<code>table</code>	TABLE
<code>inline-table</code>	TABLE
<code>table-row</code>	TR
<code>table-row-group</code>	TBODY
<code>table-header-group</code>	THEAD
<code>table-footer-group</code>	TFOOT
<code>table-column</code>	COL
<code>table-column-group</code>	COLGROUP
<code>table-cell</code>	TD, TH
<code>table-caption</code>	CAPTION

要素を表関連の `display` 値に割り当てるとは、非常に有用です。AH Formatter とともに配布される `html.css` のような HTML 用 CSS スタイルシートは、HTML の表要素を `display` 値に割り当てます。`display` 値は同じ要素であるかのように表示するよう定義されています。

```
table    { display: table }
tr       { display: table-row }
thead    { display: table-header-group }
tbody    { display: table-row-group }
tfoot    { display: table-footer-group }
col      { display: table-column }
colgroup { display: table-column-group }
td, th   { display: table-cell }
caption  { display: table-caption }
```

`display` プロパティを使って XML 要素を表要素を表すプロパティ値に対応付けることで、XML を表として表示できます。

```
<DATA>
<STACK>
  <ROW><CELL>一行一列目</CELL><CELL>一行二列目</CELL></ROW>
  <ROW><CELL>二行一列目</CELL></ROW>
  <ROW><CELL>三行一列目</CELL></ROW>
```

```

</STACK>
</DATA>
DATA {
  display: block;
  margin: 20%;
}
STACK {
  display: table;
  border-collapse: collapse;
}
ROW {
  display: table-row;
}
CELL {
  display: table-cell;
  padding: 10pt;
  border: solid thin;
}

```

一行一列目	一行二列目
二行一列目	
三行一列目	

XML を表として表示

10.2 ボックス基本特性の表への適用

CSS のボックスは margin, border, padding, width, height プロパティの大きさで決まります。表については少し異なっており、これらの特性は次のように適用されます。

ボックス基本特性の表への適用

display 値	margin	border	padding	width	height
table	○	○	○	○	○
inline-table	○	○	○	○	○
table-row	×	○	×	×	○
table-row-group	×	○	×	×	○
table-header-group	×	○	×	○	○
table-footer-group	×	○	×	○	○
table-column	×	○	×	○	×
table-column-group	×	○	×	○	×
table-cell	×	○	○	○	○
table-caption	○	○	○	○	○

- 表のパディングは、`border-collapse: separate;`のときのみ適用されます。
- 表の幅は、`border-collapse: separate;`のとき左右のパディング内側の間隔、`border-collapse: collapse;`のとき左右のボーダーの中心間の間隔になります。
- `table-column`, `table-column-group`, `table-row`, `table-row-group` のボーダープロパティは、`border-collapse: collapse;`のときのみ適用されます。
- `table-column`, `table-column-group` の `width` プロパティは、コラムの最小幅の指定となります。

10.3 表のボーダーモデル

10.3.1 ボーダーを分離するかどうか：border-collapse プロパティ

- 初期値：separate
- 適用対象：表要素
- 値の継承：する

CSS の表の各セルの周囲のボーダーを独立に扱うかどうかを指定します。

- `collapse`：隣接セルのボーダーをひとつにして扱います。
- `separate`：隣接セルのボーダーを分離して扱います。

10.3.2 ボーダー間のアキ：border-spacing プロパティ

- 初期値：0
- 適用対象：表要素
- 値の継承：する

`border-collapse: separate;`のとき、隣接するセルのボーダーどうしの間隔（アキ量）を指定します。

```
table {
  padding: 5mm;
  border: solid 2mm gray;
  border-collapse: separate;
  border-spacing: 2mm;
}

th, td {
  padding: 2mm;
  border: solid 2mm silver;
}
```

上の指定例は次のように可視化されます。

h1	h2	h3
1-1	1-2	1-3
2-1	2-2	2-3

→ ← border-spacing: 2mm

border-collapse: separate;の表

```
table {
  padding: 5mm;
  border: solid 2mm gray;
  border-collapse: collapse;
}

th, td {
  padding: 2mm;
  border: solid 2mm silver;
}
```

上の指定例は次のように可視化されます。このとき、表のパディングはなくなります。また、表のボーダーとセルのボーダーが同じ幅のため、次項の優先度が適用されて、表が生成するボックスのボーダーがセルのボーダーに上書きされています。

h1	h2	h3
1-1	1-2	1-3
2-1	2-2	2-3

border-collapse: collapse;の表

10.3.3 ボーダーの優先度

前項の例のように、border-collapse: collapse;を指定すると、隣接するボーダーをひとつにまとめます。このとき、ボーダースタイルの優先度は次のようになります。

- border-style: hidden; の優先度が最も高い。
- border-style: none; の優先度が最も低い。
- 幅の広いボーダーが狭いボーダーよりも優先度が高い。

- 同じ幅をもつボーダーの優先度は、ボーダーのスタイルによって、高い順に次のとおり。

1. double
2. solid
3. dashed
4. dotted
5. ridge
6. outset
7. groove
8. inset

double が最も高く、inset が最も低い。

- 色のみが異なるボーダーの優先度は、要素の display 種類によって、高い順に次のとおり。

1. table-cell
2. table-row
3. table-row-group
4. table-column
5. table-column-group
6. table

表の優先度が最も低い。

- 要素の種類が同じで、同じプロパティの線種をもつときは、横書きでは、左に行くほど、上に行くほど優先度が高くなります。

10.4 表の幅・段の幅の決定方法：table-layout プロパティ

- 初期値：auto
- 適用対象：表要素
- 値の継承：しない

表や列の幅の決定方法を table-layout プロパティで指定できます。

- auto：表の幅・列の幅を自動的に計算して表をレイアウトします。
- fixed：表の幅・列の幅を固定値に基づいて表をレイアウトします。

auto では、表の幅は内容によって決まります。列の幅の最小値は、コラム要素の width プロパティ¹⁰⁾で指定できますが、それ以外の列は、プログラムがセルの内容と表全体の幅から自動的に決定します。

10

10) HTML での col 要素に相当する要素

`table-layout: fixed;`では、セルの内容とは関係なく表の幅を決定します。表の幅が指定されていないときは、表を含むブロックの幅となります。各列の幅は次のようになります。

- コラム要素の `width` プロパティの値に列の幅が指定されているとき、その列は指定の値になります。
- それ以外の場合、1行目のセルの `width` プロパティの値が列の幅になります。
- 幅が指定されていない、残りの列には、表全体から幅が指定された列の幅を除いた残りの幅が均等に割り当てられます。

```
table {  
  table-layout: fixed;  
}  
  
col.first {  
  width: 10em;  
}
```

10.5 表の中央揃え

表全体を中央揃えにするには、`margin-left: auto;`と`margin-right: auto;`を指定します。

```
table {  
  margin-left: auto;  
  margin-right: auto;  
}
```

10.6 表のキャプションの位置：caption-side プロパティ

●初期値：top ●適用対象：キャプション要素 ●値の継承：する

表のキャプションの位置を指定するには、`caption-side` プロパティを使います。

- `top`：キャプションを表の上に表示します。
- `bottom`：キャプションを表の下に表示します。

キャプションを表の上に配置した例

製品紹介

オプション名	製品内容
SVG オプション	SVG を出力するためのオプションです。
バーコードオプション	バーコードフォントを使用せず、バーコードの画像 (png または svg) を生成して、文書中に画像として直接埋め込むことができるオプションです。

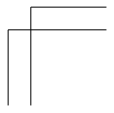
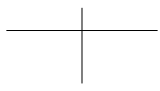
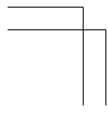
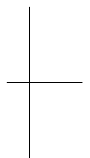
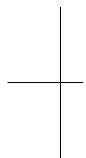
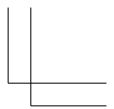
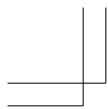
```
table {  
  caption-side: top;  
  ...  
}
```

キャプションを表の下に配置した例

オプション名	製品内容
SVG オプション	SVG を出力するためのオプションです。
バーコードオプション	バーコードフォントを使用せず、バーコードの画像（png または svg）を生成して、文書中に画像として直接埋め込むことができるオプションです。

製品紹介

```
table {  
  caption-side: bottom;  
  ...  
}
```

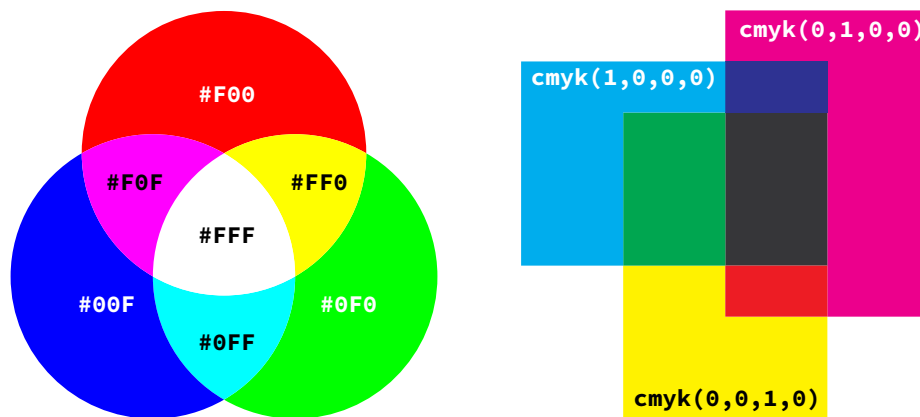


第 11 章 色指定

11.1 印刷の色

ブラウザで CSS を使用すると、表示されている色はディスプレイから発光するものです。赤、緑、青の原色が組み合わせられて色を生成するとき、加法色となります。各コンポーネントがより多く追加されると、色は薄くなります。それが #FFF が白である理由です。

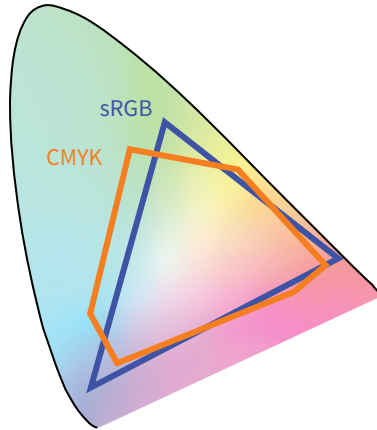
逆に、印刷すると、表示される色は、印刷の表面から反射される光です。シアン、マゼンタ、イエローの原色が結合して色を生成するとき、減法色となります。各コンポーネントがより多く追加されると、色は濃くなります。cmyk(1, 1, 1, 0) は、黒でなければなりません。実際には泥のような茶色になります。それが、黒が第 4 の色として追加される理由のひとつです¹¹⁾。黒の印刷インクを使用することは、カラーインクを 3 倍使用するより安いです。テキストは通常黒でのみ印刷されます。他の三つのインクが揃っていても心配ありません。黒に他の色を混ぜることによって、グラフィクスなどで使用する深みのある黒を作ることができます。



RGB ・ CMY

RGB と CMY の関係は、1860 年頃に James Clerk Maxwell によって最初に実証されました。簡単に言えば、印刷原色は、透過された原色の二次色で、逆も同様です。シアンは、青色と緑色の混合または白色から赤色を除いた混合、同様にマゼンタと黄色の混合です。しかし、以下の図が示すように、CMYK 印刷インクは、ウェブ上の RGB 色に使用される sRGB カラースペースと同じ色域（すなわち色範囲）を持っていません。また、可視光の全色域をカバーしていません。

11) CMYK の K は、他のプレートに対して、黒は key となる印刷プレートであることから由来します。



sRGB・CMYK色エリア

sRGBとCMYKの相違は、色の使い方に影響を与えます。印刷用デバイスを使用するにあたって遭遇するのはCMYKを使用してRGB色を表現するのにどうやって処理するかという問題です。通常は色域外の色を印刷できるように色を置き換えます（色域外の色はほとんど置き換えるのが難しいので、期待した色にならないのは避けられません）。主に画面上でページを表示するなら、そして、プリンターで印刷する場合も想定するなら、RGB色が最適です。ただし、ページ媒体を商業目的で印刷する場合は、画像にRGBとCMYKのどちらを使用するかどうかは、CMYKへの最終変換を行うプリンタを考慮する必要があります。商業印刷用の画像を準備する作業は複雑な課題であり、本書の対象外です。

11.2 文字色：color プロパティ



●初期値：black¹²⁾ ●適用対象：すべての要素 ●値の継承：する

文字やボーダーの色を指定するには、color プロパティを使います。通常の色指定は、RGBで行い、レッド、グリーン、ブルーの三つの要素を持っています。CMYKは印刷のみに使用され、シアン、マゼンタ、イエロー、ブラックの四つの要素を持っています。International Color Consortium (ICC) カラープロファイルは、デバイスに依存しないカラースペースとデバイスの仕様を特徴付けした一連のデータです。RGBとICCは、印刷時にはCMYKに変換されますが、PDFやAH Formatterで生成される他のファイルにはRGBかICCの色として保持されます。

以下の方法で指定できます。

- #RGB：R・G・Bそれぞれ16進数1桁で指定します。（#5F0 = #55FF00）

12) 初期値のblackは変更できます。

- #RRGGBB : R・G・B それぞれ 16 進数 2 桁で指定します。
- rgb(255, 0, 0) : 左から R・G・B それぞれ 0~255 の整数値で指定します。
- rgb(100%, 0%, 0%) : 左から R・G・B それぞれ 0%~100%で指定します。
- black など : 定義済みの名前で指定します。AH Formatter は、CSS Color Module Level 3 で定義された拡張色キーワードに対応しています。
- cmyk() : 四色処理印刷の CMYK 色を指定します。 
- device-cmyk() : cmyk() と同様。
- rgb-icc() : 定義されたカラースペースの色を指定します¹³⁾。以下のように変数の引数を取ります。 

```
em { color: #F00; } /* #RGB */
em { color: #FF0000; } /* #RRGGBB */
em { color: rgb(255, 0, 0); } /* 整数の範囲 0-255 */
em { color: rgb(100%, 0%, 0%); } /* 0%-100% */
em { color: red; } /* 色キーワード */
em { color: cmyk(0, 0.9922, 1, 0); } /* 0.0-1.0 */
em { color: cmyk(0%, 99.22%, 100%, 0%); } /* 0%-100% */
em { color: rgb-icc(#CMYK, 0, 0.9922, 1, 0); } /* プロファイルに依存 */
```

定義済みの色名

aqua	水色	#00FFFF	black	黒	#000000	blue	青	#0000FF
fuchsia	ピンク	#FF00FF	gray	灰色	#808080	green	緑	#008000
lime	ライム	#00FF00	maroon	栗色	#800000	navy	紺	#000080
olive	オリーブ	#808000	orange	オレンジ	#FFA500	purple	紫	#800080
red	赤色	#FF0000	silver	薄い灰色	#C0C0C0	teal	青緑	#008080
white	白	#FFFFFF	yellow	黄色	#FFFF00			

11.2.1 CMYK 色

CMYK 色は、CMYK に対応していない媒体で使用するフォールバック RGB 色の有無にかかわらず、指定できます。フォールバック RGB 色がない場合は、CMYK 色から計算されます。

- cmyk(<C>, <M>, <Y>, <K>) : CMYK 色。シアン、マゼンタ、イエロー、ブラックのコンポーネントです。
- device-cmyk(<C>, <M>, <Y>, <K>) : GCPM による定義と等価。
- rgb-icc(#CMYK, <C>, <M>, <Y>, <K>) : 等価。
- rgb-icc(<R>, <G>, , #CMYK, <C>, <M>, <Y>, <K>) : フォールバック RGB 色のある CMYK。

```
em { color: cmyk(0, 0.9922, 1, 0); } /* 0.0-1.0 */
em { color: cmyk(0%, 99.22%, 100%, 0%); } /* 0%-100% */
em { color: device-cmyk(0, 0.9922, 1, 0); } /* 0.0-1.0 only */
em { color: rgb-icc(#CMYK, 0, 0.9922, 1, 0); } /* 0.0-1.0 */
```

13) rgb-icc() は XSL 1.1 で定義され、AH Formatter はこれを実装しています。

```
em { color: rgb-icc(#CMYK, 0%, 99.22%, 100%, 0%); } /* 0%-100% */
em { color: rgb-icc(255, 0, 0, #CMYK, 0, 0.9922, 1, 0); } /* RGB 0-255 */
em { color: rgb-icc(1.0, 0, 0, #CMYK, 0%, 99.22%, 100%, 0%); } /* RGB 0.0-1.0 */
```

11.2.2 不透明度

RGB と CMYK 色は、色の不透明度を指定するアルファコンポーネントを追加で指定できます。名前付きカラーでアルファコンポーネントを使用することはできません。

```
em { color: #F008; } /* #RGBA */
em { color: #FF000088; } /* #RRGGBBAA */
em { color: rgba(255, 0, 0, 0.5); } /* 不透明度 0.0-1.0 */
em { color: rgba(255, 0, 0, 50%); } /* 不透明度 0%-100% */
em { color: rgba(100%, 0%, 0%, 0.5); } /* 不透明度 0.0-1.0 */
em { color: rgba(100%, 0%, 0%, 50%); } /* 不透明度 0%-100% */
em { color: cmyka(0, 0.9922, 1, 0, 0.5); } /* 不透明度 0.0-1.0 */
em { color: cmyka(0, 0.9922, 1, 0, 50%); } /* 不透明度 0%-100% */
em { color: cmyka(0%, 99.22%, 100%, 0%, 0.5); } /* 不透明度 0.0-1.0 */
em { color: cmyka(0%, 99.22%, 100%, 0%, 50%); } /* 不透明度 0%-100% */
```

不透明度

#F00F	#F00E	#F00D	#F00C	#F00B	#F00A	#F009	#F008
#F007	#F006	#F005	#F004	#F003	#F002	#F001	#F000

11.2.3 rgb-icc()

rgb-icc() 色関数は、以下の方法で指定できます:

- rgb-icc(#CMYK, 0.5, 0.5, 0.5, 0) : CMYK 色.
- rgb-icc(#Grayscale, 0.5) : グレイスケール.
- rgb-icc(#Separation, 'Name') : スポットカラー.
- rgb-icc(#Registration) : すべてのセパレーションで同じ強度で印刷します.
- rgb-icc(#Separation, 'All') : rgb-icc(#Registration) と同様.

11.2.4 グレイスケール

グレイスケール (白黒) の色は rgb-icc(#Grayscale, <Scale>) で指定でき、グレイスケールカラーを表示できないデバイスで使用するフォールバック RGB 色を指定する特別なパラメータを、任意で指定できます。

```
em { color: rgb-icc(#Grayscale, 0.5); } /* 0.0 (black) to 1.0 (white) */
em { color: rgb-icc(#Grayscale, 50%); } /* 0% (black) to 100% (white) */
em { color: rgb-icc(128, 128, 128, #Grayscale, 0.5); } /*RGB フォールバック カラー*/
em { color: rgb-icc(50%, 50%, 50%, #Grayscale, 0.5); } /*RGB フォールバック カラー*/
```

グレイスケール値

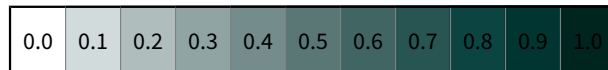
0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

11.2.5 PANTONE® スポットカラー

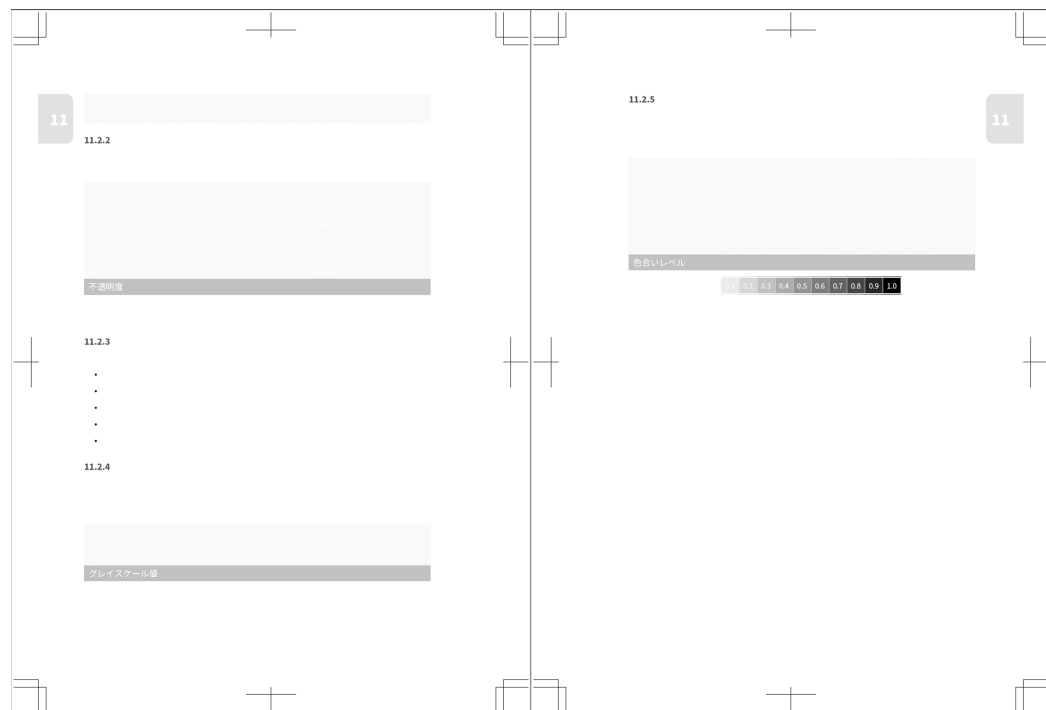
AH Formatter PANTONE® オプション¹⁴⁾をお持ちの場合は、1,000 以上の PANTONE® カラーを名前指定してスポットカラーとして印刷したり、レンダリングや印刷のために正しい RGB か CMYK に変換したりできます。

```
em { color: rgb-icc(#Separation, 'PANTONE 627 PC', 1.0); } /* 名前, 色調 */
em { color: rgb-icc(#Separation, 'PANTONE 627 PC', 0.5); } /* 色調 0.0-1.0 */
em { color: rgb-icc(#Separation, 'PANTONE 627 PC', 50%); } /* 色調 0%-100% */
em { color: rgb-icc(#Separation, 'PANTONE 627 PC'); } /* 1.0 色調想定 */
em { color: rgb-icc(#Separation, 'PANTONE 627 PC', 1, 90%, 0%, 75%, 83%); }
/* CMYK 等価 */
em { color: rgb-icc(0, 91, 25, #Separation, 'PANTONE 627 PC'); }
/* RGB 等価 */
em { color: rgb-icc(0, 91, 25, #Separation, 'PANTONE 627 PC', 1,
90%, 0%, 75%, 83%); } /* RGB, CMYK の両方と等価 */
```

色合いレベル



組版した文書を商用目的で印刷するとき、各 PANTONE® カラーは、その PANTONE® カラー用の特定のインクで、個別に印刷されます。各 PANTONE® カラーのセパレーションのグレイスケールレベルは、適用される色合いのレベルに対応します。



セパレーションのグレイスケールレベルは、適用される色合いのレベルに対応します。

14) AH Formatter PANTONE® オプションを購入する必要があります。

11.2.6 その他のスポットカラー

スポットカラーは、AH Formatter PANTONE® オプションなしで使用できます。ただし、スポットカラーセパレーションに対応していない媒体で使用するには、同等の RGB と CMYK カラーのひとつか両方を持つ必要があります。RGB か CMYK の等価が省略されている場合は、他の等価色のコンポーネントから計算されます。

- `rgb-icc(<R>, <G>, , #Separation, <Name>, <Tint>, <C>, <M>, <Y>, <K>)`：スポットカラーの名前、色合い、CMYK と RGB の両方のフォールバックカラーです。
- `rgb-icc(<R>, <G>, , #Separation, <Name>, <Tint>)`：スポットカラーの名前、色合い、RGB のフォールバックカラーです。
- `rgb-icc(<R>, <G>, , #Separation, <Name>, <Tint>)`：スポットカラーの名前、RGB のフォールバックカラーです。
- `rgb-icc(#Separation, <Name>, <Tint>, <C>, <M>, <Y>, <K>)`：スポットカラーの名前、色合い、CMYK のフォールバックカラーです。

PANTONE® カラーと同様に、組版した文書を商用目的で印刷するとき、各 CMYK スポットカラーは、色用の特定のインクで、個別に印刷されます。各色のセパレーションのグレイスケールレベルは、適用される色合いのレベルに対応します

12.1 章や節に番号を付ける

要素に章番号や節番号のような連番を付けるには、`counter-increment` プロパティ、`counter-reset` プロパティ、`content` プロパティを使います。

`counter-increment` プロパティと `counter-reset` プロパティでカウンタ名とその値を設定します。`content` プロパティにカウンタ名を指定すると、`::before` 疑似要素または `::after` 疑似要素にカウンタの値が挿入されます。`counter-increment` プロパティを適用した要素が現れるたびにカウンタの値が増え、`counter-reset` プロパティを適用した要素が現れるたびにカウンタの値がリセットされます。

```
body {
  counter-reset: ChapterNo;          /* 章番号 (ChapterNo) をリセットします */
}
h1:before {
  counter-increment: ChapterNo;      /* 章番号に1を足します */
  /* 章見出し (h1) の前に「Chapter」を挿入します */
  content: "Chapter" counter(ChapterNo) " ";
}
h1 { /* Chapterにh1:beforeとh1自身の内容をセットします */
  string-set: Chapter content(before) content();
  counter-reset: SectionNo;         /* 節番号をリセットします */
}
h2:before {
  counter-increment: SectionNo;      /* 節番号に1を足します */
  content: counter(ChapterNo) "." counter(SectionNo) " ";
}
h2 { /* Sectionにh2:beforeとh2自身の内容をセットします */
  string-set: Section content(before) content();
}
@page :left {
  @top-left { /* 左ページの柱に章のタイトルを挿入します */
    content: string(Chapter);
  }
}
@page :right {
  @top-right { /* 右ページの柱に節のタイトルを挿入します */
    content: string(Section);
  }
}
```

12.2 文字挿入：content プロパティ

●初期値：normal ●適用対象：CSS2.1 の場合 ::before 疑似要素と ::after 疑似要素（CSS3 ではあらゆる要素に適用） ●値の継承：しない

要素の直前または直後に文字列を挿入するには、content プロパティを使います。CSS3 では、要素のコンテンツとして文字列を指定もできます。

- normal：文字を挿入しません。
- none：文字を挿入しません。（normal と同じ動作となります。）
- 文字列：挿入する文字列をダブルクォーテーションまたはシングルクォーテーションで囲んで指定します。
- url()：画像ファイルの URL を指定します。content: url(image.png); などと指定することで、要素の内容を画像にできます。
- attr()：指定した属性の値が content プロパティの値になります。
- counter()：カウンタの値を挿入します。
- open-quote：quotes プロパティで指定した開始引用符を挿入します。
- close-quote：quotes プロパティで指定した終了引用符を挿入します。
- no-open-quote：引用符は表示されませんが、quotes プロパティで指定されている引用符の入れ子の深さが一階層深くなります。
- no-close-quote：引用符は表示されませんが、quotes プロパティで指定されている引用符の入れ子の深さが一階層浅くなります。

normal と none 以外の値は繰り返しが可能で、任意の順序で組み合わせが可能です。

```
.Chapter h2:before {
  content: "Chapter " counter(ChapterNo) ". ";
}
```

string-set プロパティで使われる content()関数については、19.2.2 文字列変数：string-set プロパティ（ページ 102）を参照してください。

12.3 カウンタの加算：counter-increment プロパティ

●初期値：none ●適用対象：すべての要素 ●値の継承：しない

指定したカウンタの値を増やすには、counter-increment プロパティを使います。

- none：カウンタを変更しません。
- カウンタ名：指定した名前のカウンタの値をひとつ増やします。
- カウンタ名 整数：スペース文字で区切って整数値を指定すると、カウンタの値がその数だけ前後します。

複数のカウンタ名の指定もできます。整数が後に続く指定もできます。

```
.Chapter h2 {
  counter-increment: ChapterNo;
}
```

12.4 カウンタのリセット：counter-reset プロパティ

- 初期値：none ●適用対象：すべての要素 ●値の継承：しない
- 指定したカウンタの値をリセットするには、counter-reset プロパティを使います。
- none：カウントをリセットしません。
 - カウンタ名：指定した名前のカウンタの値を 0 にします。
 - カウンタ名 整数：スペース文字で区切って整数値を指定すると、カウンタの値がその数にリセットされます。

```
.Chapter h2 {
  counter-reset: SectionNo;
}
```

12.5 ページカウンタ

現在のページを求めるには counter(page) を使い、総ページ数を求めるには counter(pages) を使います。

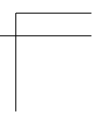
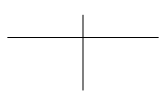
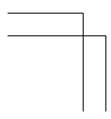
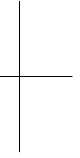
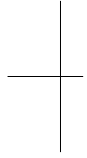
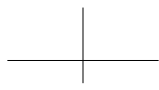
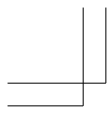
```
<p>このページのページ数 = <span style="content: counter(page)"></span></p>
<p>この文書の総ページ数 = <span style="content: counter(pages)"></span></p>
```

```
このページのページ数 = 65
この文書の総ページ数 = 142
```

counter() には、カウンタスタイルを指定する任意の第 2 引数があります。省略した場合の既定値は decimal となります。

```
<p>このページのページ数 =
  <span style="content: counter(page, lower-roman)"></span></p>
<p>この文書の総ページ数 =
  <span style="content: counter(pages, upper-roman)"></span></p>
```

```
このページのページ数 = lxv
この文書の総ページ数 = CXLII
```



display プロパティには、要素の表示をリストとして表示する値はありません。しかし、display: list-item; は要素にリスト項目を生成させます。すべてのリスト項目にはマークがあります。これは、リスト項目を識別する行頭記号、数字またはその他のマークです。CSS2 では、list-style-type, list-style-image, list-style-position, list-style プロパティを使用してマークの組版を指定します。CSS3 は ::marker 疑似要素を追加するので、CSS のすべての範囲のプロパティと値を用いて、リスト項目をスタイルできます。list-style-type と list-style-image プロパティは、::marker 疑似要素の既定コンテンツを設定します。

13.1 リスト項目マーカ画像：list-style-image プロパティ

●初期値：none ●適用対象：リスト項目 ●値の継承：する

リスト項目マーカの既定コンテンツとして使用する画像を指定します。list-style-image が none または画像が無効の場合、既定コンテンツは代わりに list-style-type で与えられます。

- none：画像がない。
- <image>：既定コンテンツとして使用する画像。

13.2 リスト項目マーカ種類：list-style-type プロパティ

●初期値：disc ●適用対象：リスト項目 ●値の継承：する

list-style-image が none または画像が無効な場合のリスト項目マーカの既定コンテンツを指定します。それ以外の場合は無視されます。

- none：既定コンテンツは none です。
- <counter-style>：既定コンテンツはそのカウンタスタイルです。
- <string>：指定した文字列を、リスト項目マーカの既定コンテンツとして使用します。

13.3 リスト項目マーカ位置：list-style-position プロパティ

●初期値：outside ●適用対象：リスト項目 ●値の継承：する

list-item-position は、リスト項目マーカの位置を制御します。

- `inside` : marker 疑似要素は, `::before` 疑似要素が配置される直前にインラインに配置されます。
- `outside` : マーカの `inside` と `position` プロパティで marker の位置を計算します。

13.4 リスト項目マーカショートハンド : `list-style` プロパティ

●初期値 : プロパティを参照 ●適用対象 : リスト項目 ●値の継承 : プロパティを参照
`list-style` は, `list-style-type`, `list-style-image`, `list-style-position` を設定するためのショートハンドです。

```
ul.References li {
  list-style: check;
  line-height: 1.1;
}
```

13.5 リスト項目マーカ : `::marker` 疑似要素

`::marker` を, CSS プロパティと値の全範囲を用いてスタイルできます。既定コンテンツは `list-style-image` と `list-style-type` プロパティで定義しますが, `::marker` の `content` プロパティを設定すると上書きできます。

13.6 カウンタスタイル

`counter style` は, 連番を表すために使用する数字, アルファベット, 記号などを定義し実装します。CSS1 は, HTML が伝統的にリストとして許されているスタイルに基づいて, いくつかのカウンタスタイルを定義しました。CSS Counter Styles Level 3 はカスタムカウンタスタイルを定義するための仕組みを持っている `@counter-style`; ルールを定義します。

CSS3 カウンタスタイルのコアの部分は, 整数カウンタ値の文字列表現を生成するアルゴリズムに名前を付けることです。カウンタスタイルには, 生成された値に追加する前置や後置を示すプロパティや, 負の数を示す追加の文字列なども含まれます。カウンタスタイルは, `list-style-type`, `counters()` 関数で使用できます。

次に, CSS Counter Styles Level 3 の `filled-circled-decimal` カウンタスタイルに基づいた `my-filled-circled-decimal` カウンタスタイルの使用例を示します。その名前が示す様に, カウンタスタイルでは, 十進数を表すために黒丸で囲んだ 10 進数を使用します。数字の後にはスペースが続きます。カウンタスタイルは, `` の項目に番号を付けるときに使用します。

```
@counter-style my-filled-circled-decimal {
  system: fixed;
```

```

symbols: '\2776' '\2777' '\2778' '\2779' '\277a' '\277b' '\277c' '\277d' '\277e';
/* 記号: '❶' '❷' '❸' '❹' '❺' '❻' '❼' '❽' '❾'; */
suffix: ' ';
}
ol.my-filled-circled-decimal li { list-style-type: my-filled-circled-decimal; }
...
<ol class="my-filled-circled-decimal">
  <li title="1">1</li>
  <li title="2">2</li>
</ol>

```

my-filled-circled-decimal カウンタスタイルのリスト

- ❶ 1
- ❷ 2

13

13.7 カスタムカウンタスタイルの定義：@counter-style ルール

カスタムカウンタスタイルを定義できます。@counter-style ルールの一般的な形式の例は以下のとおりです。

```
@counter-style <counter-style-name> { <declaration-list> }
```

カウンタスタイル名は大文字と小文字を区別しますが、CSS Counter Styles Level 3 で定義済みカウンタスタイルの名前は大文字と小文字を区別しないで一致させます。カウンタスタイル名は none と一致できません。また、decimal と disc はカウンタスタイル名として定義できません。

宣言リストには以下の記述子を使用できます。

- system：カウンタの表現を構成するために使用するアルゴリズムを指定します。
- negative：値が負の場合に表現を変更する方法を定義します。
- prefix：マーカ表現の前に付ける記号を指定します。
- suffix：マーカ表現に追加する接尾辞を指定します。
- range：カウンタスタイルが定義されている範囲を定義します。
- pad：最小数の書記素クラスタではないカウンタ表現を埋め込む記号を指定します。
- fallback：現在のカウンタスタイルで表現を作成できない場合に使用されるフォールバックカウンタスタイルです。
- symbols：マーカ構築アルゴリズムで使用される記号です。
- additive-symbols：追加マーカ構築アルゴリズムで使用される記号です。
- speak-as：音声によるカウンタを合成する方法を記述します¹⁵⁾。

15) AH Formatter で実装されていません。

negative, prefix, suffix, pad, symbols の値を持つ記号と additive-symbols 記述子は文字列を指定します。

13.7.1 カウンタアルゴリズム：system 記述子

●初期値：symbolic

カウンタの表現を構成するために使用するアルゴリズムを指定します。

- cyclic：指定された記号を繰り返します。
- numeric：カウンタ記号のリストを数字方式として解釈します。
- alphabetic：記号リストをアルファベット数字方式の桁として解釈します。
- symbolic：指定された記号を繰り返し、各連続するパス上で、記号を2倍、3倍などと繰り返します。
- additive：記号は加重された値を表し、その数値は記号を追加されることによって得られます。
- fixed <integer>?：カウンタ記号を一度実行し、別のカウンタスタイルにフォールバックします。<integer>がある場合は最初の記号値を設定し、それ以外の場合は最初の記号値を1に設定します。
- extends <counter-style-name>：指定されたカウンタスタイル名と同様のアルゴリズムを使用します。

13.7.2 負の値の組版：negative 記述子

●初期値：“\2D” (“-” hyphen-minus)

値が負の場合に表現を変更する方法を定義します。値はひとつまたはふたつの記号です。値が負の場合は、最初の記号が表現の前に付加され、2番目の記号がある場合は、その記号が表現に追加されます。以下に例を示します。

- negative：“\30DE\30A4\30CA\30B9”；(japanese-informal より引用。)負の値の前に“マイナス記号”を生成します。
- 会計士は、負の数を表すのに括弧を頻繁に使用します。例えば、“(100)”は-100です。
negative：“(”) ”；は負の値の前に“(”を生成し、“)”を負の値の後に生成します。

13.7.3 マーカ前の記号：prefix 記述子

●初期値：“” (空文字列)

マーカ表現の前に付ける記号を指定します。接頭辞は、負の符号の前に置かれます。

13.7.4 マーカ後の記号：suffix 記述子

●初期値：“\2E\20” (“.”ピリオドの後にはスペースが続きます.)

マーカ表現に追加される記号を指定します。接尾辞は、負の符号の後に来ます。

13.7.5 カウンタの範囲：range 記述子

●初期値：auto

カウンタスタイル適用範囲を指定します。値は、auto または有効範囲の下限と上限をカンマで区切ったリストのいずれかです。値が auto のとき、範囲は system 値に基づいて決定されます。カウンタスタイルを使用してその範囲外の値を表す場合、カウンタスタイルは代わりにそのフォールバックカウンタスタイルを使用します。

13.7.6 最小のカウンタ幅：pad 記述子

●初期値：0^{'''}

最小数の書記素クラスタではないカウンタ表現を埋め込む記号を指定します。値は整数と記号です。表現が整数値よりも少数の書記素クラスタを有する場合、表現は記号で埋められます。最小値よりも長い表現は埋め込まれません。

13.7.7 フォールバックカウンタスタイル：fallback 記述子

●初期値：decimal

現在のカウンタスタイルで表現を作成できない場合に使用されるフォールバックカウンタスタイルを指定します。

13.7.8 カウンタの記号：symbols および additive-symbols 記述子

●初期値：n/a

マーカ構築アルゴリズムで使用される記号を指定します。system が cyclic, numeric, alphabetic, symbolic, fixed の場合は symbols が必要であり、system が additive の場合は additive-symbols が必要です。

13.8 定義済みカウンタスタイル

CSS Counter Styles Level 3 は、いくつかのカウンタスタイルをあらかじめ定義しています。その中には、@counter-style で表現するのが複雑なものもあります。W3C 国際化ワーキンググループによって発行されたカウンタスタイルは、世界中のさまざまな文化によって使用されるナンバリングシステム用のユーザ定義カウンタスタイル用のコード情報を提供しています。それらは、CSS Counter Styles Level 3 定義済みスタイルも含まれます。

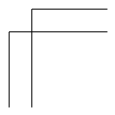
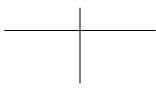
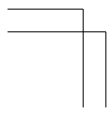
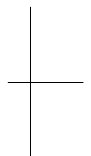
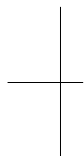
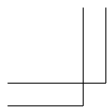
AH Formatter は以下の定義済みスタイルを実装しています¹⁶⁾.

- numeric
 - arabic-indic
 - bengali
 - binary
 - cambodian
 - cjk-decimal
 - decimal
 - devanagari
 - fullwidth-decimal
 - gujarati
 - gurmukhi
 - kannada
 - khmer
 - lao
 - lepcha
 - lower-hexadecimal
 - malayalam
 - mongolian
 - myanmar
 - new-base-60
 - octal
 - oriya
 - persian
 - shan
 - super-decimal
 - tamil
 - telugu
 - thai
 - tibetan
 - upper-hexadecimal
- alphabetic
 - afar
 - agaw
 - ari
 - blin
 - cjk-earthly-branch
 - cjk-heavenly-stem
 - dizi
 - fullwidth-lower-alpha
 - fullwidth-lower-latin*
 - fullwidth-upper-alpha
 - fullwidth-upper-latin*
 - gedeo
 - gumuz
 - hadiyya
 - harari
 - hindi
 - hiragana
 - hiragana-iroha
 - japanese-formal
 - kaffa
 - katakana
 - katakana-iroha
 - kebena
 - kembata
 - khmer-consonant
 - konso
 - korean-consonant
 - korean-hanja-formal
 - korean-hanja-informal
 - korean-syllable
 - kunama
 - lower-alpha
 - lower-belorussian
 - lower-bulgarian
 - lower-greek
 - lower-latin*
 - lower-macedonian
 - lower-oromo-qubee
 - lower-russian
 - lower-russian-full
 - lower-serbo-croatian
 - lower-ukrainian
 - lower-ukrainian-full
 - meen
 - oromo
 - saho
 - sidama
 - silti
 - thai-alphabetic
 - tigre
 - upper-alpha
 - upper-belorussian
 - upper-bulgarian
 - upper-greek*

16) * のついたスタイルは、古い list-style-type 仕様との下位互換性のためにリストされています。

- upper-latin*
- upper-macedonian
- upper-oromo-qubee
- upper-russian
- upper-russian-full
- upper-serbo-croatian
- upper-ukrainian
- upper-ukrainian-full
- wolaita
- yemsa
- additive
 - ancient-tamil
 - armenian
 - georgian
 - greek*
 - greek-lower-ancient
 - greek-lower-modern
 - greek-upper-ancient
 - greek-upper-modern
 - hebrew
 - japanese-informal
 - korean-hangul-formal
 - lower-armenian
 - lower-roman
 - simple-lower-roman
 - simple-upper-roman
 - upper-armenian
 - upper-roman
- symbolic
 - lower-alpha-symbolic
 - upper-alpha-symbolic
- fixed
 - arabic-abjad
 - circled-decimal
 - circled-ideograph
 - circled-katakana
 - circled-korean-consonant
 - circled-korean-syllable
 - circled-lower-latin
 - circled-upper-latin
 - decimal-leading-zero
 - dotted-decimal
 - double-circled-decimal
 - filled-circled-decimal
 - fullwidth-lower-roman
 - fullwidth-upper-roman
 - maghrebi-abjad
 - parenthesized-decimal
 - parenthesized-hangul-consonant
 - parenthesized-hangul-syllable
 - parenthesized-ideograph
 - parenthesized-lower-latin
 - persian-abjad
 - persian-alphabetic
- cyclic
 - box*
 - check*
 - circle
 - diamond*
 - disc
 - hyphen*
 - square
- complex
 - ethiopic-numeric
 - kansuji¹⁷⁾
 - simp-chinese-formal
 - simp-chinese-informal
 - trad-chinese-formal
 - trad-chinese-informal

17) -ah-kansuji-style を使用して変換できます。



クロスリファレンス（相互参照）

14.1 テキスト内容の参照：target-text() 関数

参照先のテキストを参照元に表示するには、target-text() 関数を使います。

target-text() 関数を使った相互参照の方法は、以下のようになります。

1. CSS ファイルの相互参照用クラスに参照元に挿入する文字列を指定します。
2. target-text() 関数を使うと、target-text()関数内に指定した変数のカウンタを content プロパティに挿入できます。
3. 文章中の参照元には、アンカーとなる <a> タグを挿入し、そのクラスに相互参照クラス名を指定します。さらに href 属性にソース参照の ID 値を指定します。
4. アンカータグ内の文字列が、参照先の文字列と差し替わります。

```
.TitleRef {
  content: target-text(attr(href url), before) " "
          target-text(attr(href url), content);
}
...
<h2 id="CrossRef">クロスリファレンス（相互参照）</h2>
...
<p>例：「<a class="TitleRef" href="#CrossRef">この章</a>」</p>
<p>「この章」という文字が、参照先のタイトルに差し替わります。</p>
```

例：「第 14 章 クロスリファレンス（相互参照）」

「この章」という文字が、参照先のタイトルに差し替わります。

14.2 カウンタの参照：target-counter() 関数

参照先の章番号やページ番号を自動的に付加するには、target-counter() 関数を使います。第 12 章 カウンタ（ページ 63）を参照してください。

target-counter() 関数を使った相互参照の方法は、以下のようになります。

1. CSS では、::before または ::after 疑似要素の内容として挿入される文字列を指定します。
2. target-counter()関数を使うと、target-counter() 関数内に指定した変数のカウンタを content プロパティに挿入できます。単なる文字列を挿入する場合は、その文字列をダブルクォーテーションまたはシングルクォーテーションで囲みます。また、target-counter() 関数との間は、半角スペースで区切ります。

3. 文章中の参照元には、アンカーとなる `<a>` タグを挿入し、そのクラスに相互参照クラス名を指定します。さらに `href` 属性にソース参照の ID 値を指定します。

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

14.3 目次作成

14

目次は、`target-counter()` 関数による章番号とページ番号の参照を使って作ることができます。

```
.TOC li.TocLevel1 > a::before { /* 目次項目に章番号を付ける */
  content: "第" target-counter(attr(href url), ChapterNo) "章 ";
}
.TOC li.TocLevel1 > a::after { /* 目次項目にページ番号を付ける */
  content: leader(space) target-counter(attr(href url), page);
}
...
<div class="TOC">
  <ul>
    <li class="TocLevel1"><a href="#WebvsPrint">Web とページ媒体</a></li>
    <li class="TocLevel1"><a href="#BoxModel">ボックスによるレイアウト</a></li>
    <li class="TocLevel1"><a href="#ObjectDecorate">オブジェクトの装飾</a></li>
  </ul>
</div>
```

目次の例

第 1 章 Web とページ媒体	1
第 2 章 ボックスによるレイアウト	5
第 3 章 オブジェクトの装飾	11
第 4 章 段落設定	17
第 5 章 段組	27
第 6 章 改ページ・ページコンテンツの維持	31
第 7 章 文字設定	33

15.1 脚注の指定：float: footnote; 📄

float: footnote; が指定されている場合、その要素の内容が脚注となります。

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

```
<p><span class="code">float: footnote;</span> が指定されている場合、
<span class="Footnote">その要素の内容が脚注となります。 </span>.</p>
```

脚注エリアの設定は、@page ルールの中で @footnote ルールを使います。ページの末尾に脚注を配置するには、float: bottom page; (9.3.2 ページフロート: float: top page;/ float: bottom page; (ページ 43)) を使用します。脚注の番号の形式を設定するには、擬似要素の ::footnote-call と ::footnote-marker を使用します。

15.2 脚注のスタイル：@footnote ルール 📄

脚注エリアの上に罫線を引くなど、脚注エリアの設定をするために、@page ルールの中で @footnote を使用します。

```
@page {
  @footnote {
    float: bottom page; /* 脚注エリアは、ページの下部へのフロートとして配置される */
    border-top: thin solid black; /* 脚注エリアの上の罫線を指定 */
    border-length: 30%; /* 罫線の長さを指定 (ページエリア幅の30%) */
    padding-top: 0.5em;
  }
}
```

15.3 脚注番号：::footnote-call/::footnote-marker 擬似要素 📄

脚注番号を設定するには、擬似要素の ::footnote-call と ::footnote-marker を使います。footnote カウンタは、脚注が生成されるたびに増加します。

```
::footnote-call { /* 脚注呼び出し */
  content: "(" counter(footnote) ")"; font-size: 6pt; vertical-align: super;
}
::footnote-marker { /* 脚注の番号 */
  content: "(" counter(footnote) ")"; font-size: 1em; vertical-align: super;
}
```

15.4 傍注の指定：float: sidenote; 📖

float: sidenote; が float プロパティに指定されている場合、内容は傍注となります。傍注の指定は、@page ルール内の、@sidenote ルールを使用します。傍注の番号の形式を設定するには、擬似要素の ::sidenote-call、::sidenote-marker を使います。

¹ Quisque suscipit ante vel eros.	Whereas recognition of the inherent dignity ¹ and of the equal and inalienable rights of all members of the human family is the foundation of freedom, justice and peace in the world,
--	---

```
span.sidenote {  
  float: sidenote;  
}
```

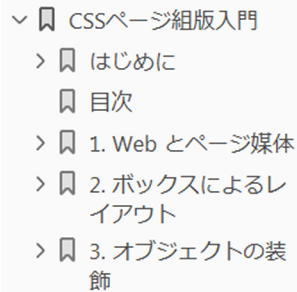
15.5 傍注スタイル：@sidenote ルール 📖

@page ルールの@sidenote ルールを使用して、傍注エリアの位置と範囲を設定します。

```
@page {  
  @sidenote {  
    float: outside;  
    clear: both;  
    width: 20%;  
  }  
}
```

PDF ブックマーク

PDF のブックマーク（しおり）を生成できます。



本書のブックマークの一部のスクリーンショット

16.1 ブックマークレベル：bookmark-level プロパティ

●初期値：none ●適用対象：すべての要素 ●値の継承：しない

ブックマークのレベルを指定します。HTML の <h1>-<h6> 要素の見出しレベルに相当します。

16.2 ブックマークラベル：bookmark-label プロパティ

●初期値：none ●適用対象：すべての要素 ●値の継承：しない

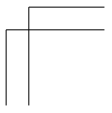
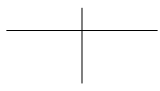
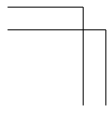
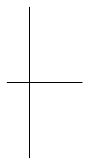
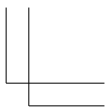
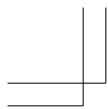
ブックマークのラベルを指定します。ブックマークラベルは none の場合、要素の内容がブックマークラベルになります。

16.3 ブックマークの状態：bookmark-state プロパティ

●初期値：open ●適用対象：ブロック要素 ●値の継承：しない

ブックマークを開いた状態にするか閉じた状態にするかを指定します。bookmark-state: closed; とすると、指定のブックマークは閉じた状態になります。

```
/* ブックマーク */
h1 { bookmark-level: 1; }
h2 { bookmark-level: 2; bookmark-state: closed; }
h3 { bookmark-level: 3; bookmark-state: closed; }
h4 { bookmark-level: 4; bookmark-state: closed; }
h5,h6 { bookmark-level: none; }
```



17.1 行頭行末約物半角固定・段落先頭全角 1 字下げ

日本語文書を組版する場合の基本的なスタイル設定は、行頭約物半角固定・行末約物半角固定・段落先頭全角 1 字下げとなっています。

- 行頭約物：半角固定とします。
- 行末約物：半角固定とします。
- 段落先頭：全角 1 字下げとします。

```
body { /* 日本語組版のための設定 */
  punctuation-trim: start end adjacent;
  -ah-text-justify-trim: punctuation;
  -ah-text-autospace: ideograph-numeric ideograph-alpha;
  -ah-text-autospace-width: 25%;
  hanging-punctuation: none;
}
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 */
}
```

夏目漱石（なつめ そうせき、
慶応三年一月五日（一八六七年
二月九日）―大正五年（一九一
六年）二月九日）は、日本の
小説家、評論家、英文学者。本
名、金之助。『吾輩は猫であ
る』『こゝろ』などの作品で広く
知られる、森鷗外と並ぶ明治・
大正時代の文豪である。江戸の
牛込馬場下横町（現在の東京都
新宿区喜久井町）出身。俳号は
愚陀仏。
大学時代に正岡子規と出会
い、俳句を学ぶ。…（中略）…
当初は余裕派と呼ばれた。
『修善寺の大患』後は、『行人』
『こゝろ』『硝子戸の中』などを
執筆。「則天去私」（そくてん
きよし）の境地に達したといわ
れる。晩年は胃潰瘍に悩まさ
れ、『明暗』が絶筆となった。

17.2 約物の詰め処理：punctuation-trim プロパティ 

- 初期値：none ●適用対象：ブロック要素 ●値の継承：する
- 全角約物を半角扱いにするかどうかを指定します。
- none：全角約物を詰めません。
 - start：行頭での全角約物（開き括弧など）を半角に扱います。

- `start-except-first` : `start` と同じですが、先頭の行や強制改行直後の行を除きます。
- `end` : `text-align: right;` , `text-align: justify;` (または `text-align-last: justify;`) のとき、全角約物 (閉じ括弧など) が行末に位置した場合、その文字は強制的に半角に扱います。
- `allow-end` : `text-align: right;` , `text-align: justify;` (または `text-align-last: justify;`) のとき、全角約物 (閉じ括弧など) が行末に位置した場合、行に収まるときは全角のままとし、収まらないときは半角に扱います。
- `end-except-fullstop` : `end` と同じですが、次の 2 文字を除外します。
 - `U+3002 = "。"`
 - `U+FF0E = "。"`
- `adjacent` : 日本語での全角約物と全角文字の間を詰めます。処理されるのは以下の間隔です。全角句読点は全角閉じ括弧と同様に扱われます。
 - 全角閉じ括弧と全角開き括弧の間
 - 全角閉じ括弧と全角閉じ括弧の間
 - 全角閉じ括弧と全角中点類の間
 - 全角閉じ括弧と全角空白の間
 - 全角閉じ括弧と全角非約物の間
 - 全角開き括弧と全角開き括弧の間
 - 全角中点類と全角開き括弧の間
 - 全角空白と全角開き括弧の間
 - 全角非約物と全角開き括弧の間
- `all` : 全角の括弧類, 中点類, 句読点類をすべて詰めて半角扱いにします。
- `<string>` : `all` と同じですが、ここで指定した文字列に含まれる文字のみが半角扱いされます。
- `auto` : システムの既定値に依存します。

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。

```
/* 行頭, 行末, 約物どうしが隣接のとき, 全角約物を半角幅にします */
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。」

約物の詰めを無効にすると次のようになります。 (`punctuation-trim: none;` を指定)

「《約物 [やくもの] 》、つまり『括弧』・『句読点』の類 (たぐい) です。」

17.3 行調整のための約物の詰め：-ah-text-justify-trim プロパティ

●初期値：none ●適用対象：すべての要素 ●値の継承：する

日本語での追い込みの処理方法を指定します。指定された詰めを行って、行に入るだけ文字を追い込みます。

- none：追い込みません。
- punctuation：全角の括弧類，中点類，句読点類を詰めの対象として追い込みます。
- punctuation-except-fullstop：punctuation と同じですが，次の 2 文字を除外します。
 - U+3002 = "。"
 - U+FF0E = ". "
- punctuation-except-middledot：punctuation と同じですが，次の中点類を除外します。
 - U+30FB = "・"
 - U+FF1A = "："
 - U+FF1B = "；"
- kana：ひらがな，カタカナを多少詰めて追い込み処理をします。
- ideograph：漢字やかなの間を詰めの対象として追い込み処理をします。
- inter-word：欧文単語間のスペースを詰めの対象として追い込み処理をします。
- auto：システムの既定値に依存します。

```
/* 行の調整のために約物を詰めます */  
-ah-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条件に該当しない場合でも，行末をぴったりと揃える調整のために，全角の開き括弧類の前，全角閉じ括弧類や句読点の後を，必要に応じて自動的に詰めます。

17.4 和欧文間のアキ：-ah-text-autospace プロパティ

●初期値：auto ●適用対象：ブロック要素 ●値の継承：する

日本語で，和欧文間にアキを入れるかどうかを指定します。

- none：アキを入れません。
- ideograph-numeric：仮名漢字－欧文数字間にアキを入れます。
- ideograph-alpha：仮名漢字－欧文アルファベット間にアキを入れます。

- `ideograph-parenthesis` : 仮名漢字－欧文括弧間にアキを入れます。ただし、仮名漢字－欧文閉じ括弧間や、欧文開き括弧－仮名漢字間にはアキを入れません。
- `ideograph-punctuation` : 仮名漢字－欧文約物間にアキを入れます。ピリオド－仮名漢字間にはアキを入れますが、仮名漢字－ピリオド間にはアキを入れません。カンマも同様です。
- `auto` : システムの既定値に依存します。設定によって、`none` または `ideograph-numeric ideograph-alpha` とみなされます。

このように、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。

```
/* 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ます */
-ah-text-autospace: ideograph-numeric ideograph-alpha;
```

以下では比較のため、下側の例で `-ah-text-autospace: none;` を指定して和欧文間のアキを無効にしています。

「日本語にも global にも 100%を目指す AH Formatter です」

「日本語にも global にも 100%を目指す AH Formatter です」

17.5 和欧文間のアキ量：-ah-text-autospace-width プロパティ

- 初期値：25% ●適用対象：ブロック要素 ●値の継承：する

日本語で、和欧文間のアキの量を指定します。

```
/* 和欧文間のアキの量を指定します */
-ah-text-autospace-width: 25%; /* 初期値 */
```

17.6 突き出し・ぶら下げ：hanging-punctuation プロパティ

- 初期値：none ●適用対象：ブロック要素 ●値の継承：する

行頭での突き出し、行末でマージンにぶら下げを行うかどうかを指定します。

- `none` : 行頭での突き出し、行末でのぶら下げを処理しません。
- `start` : 行頭での突き出しを処理します。このとき、突き出し対象文字が行頭に位置した場合、その文字は強制的に突き出されます。
- `first` : 段落の先頭行でのみ、`start` と同じ処理をします。
- `force-end` : 行末でのぶら下げを処理します。`text-align: right;`、`text-align: justify;`（または `text-align-last: justify;`）のとき、ぶら下げ対象文字が

行末に位置した場合、その文字は強制的にぶら下げられます。text-align がそれ以外のときは、なりゆきでぶら下げが起こります。

- allow-end: 行末でのぶら下げを処理します。text-align: right;, text-align: justify; (または text-align-last: justify;) のとき、ぶら下げ対象文字が行末に位置した場合、行に収まる時はぶら下げられません。行に収まらない時はぶら下げられます。text-align がそれ以外のときは、なりゆきでぶら下げが起こります。
- last: 段落末でのみ、force-end と同じぶら下げを処理します。

対象となる約物は以下のとおりです。

- force-end, allow-end

日本語または簡体字中国語の場合

- U+3001 = "、"
- U+3002 = "。"
- U+FF0C = "、"
- U+FF0E = "。"

繁体字中国語の場合

- U+FE50
- U+FE51
- U+FE52
- U+FF64

その他の場合

- last と同じ

- last

引用符類, 閉じ括弧類, ピリオド, カンマ, ハイフン

- start, first

引用符類, 開き括弧類, 簡条書き行頭文字/段落番号

```
/* 行頭での突き出し, 行末でのぶら下げを行うかどうかを指定します。*/  
hanging-punctuation: none; /* 初期値 */
```

17.7 行頭約物半角固定・行末約物全角/半角・段落先頭全角 1 字下げ

行頭約物半角固定・行末約物全角または半角・段落先頭全角 1 字下げとします。

- 行頭約物: 半角固定とします。
- 行末約物: 全角または半角とします。
- 段落先頭: 全角 1 字下げとします。

```

body { /* 日本語組版のための設定 */
  punctuation-trim: start allow-end adjacent;
  -ah-text-justify-trim: punctuation;
  -ah-text-autospace: ideograph-numeric ideograph-alpha;
  -ah-text-autospace-width: 25%;
  hanging-punctuation: none;
}
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 */
}

```

夏目漱石（なつめ そうせき、
 慶応三年一月五日（一八六七年二
 月九日）―大正五年（一九一六年）
 一月九日）は、日本の小説家、
 評論家、英文学者。本名、金之
 助。『吾輩は猫である』『こゝろ』
 などの作品で広く知られる、森鷗
 外と並ぶ明治・大正時代の文豪で
 ある。江戸の牛込馬場下横町（現
 在の東京都新宿区喜久井町）出身。
 俳号は愚陀仏。
 大学時代に正岡子規と出会い、
 俳句を学ぶ。：（中略）：当初は
 余裕派と呼ばれた。
 「修善寺の大患」後は、『行人』
 『こゝろ』『硝子戸の中』などを執
 筆。「則天去私」（そくてんきよ
 し）の境地に達したといわれる。
 晩年は胃潰瘍に悩まされ、『明暗』
 が絶筆となった。
 フリー百科事典『ウィキペディ
 ア』より引用

17.8 行頭約物半角固定・行末句点全角・段落先頭全角1字下げ

行頭約物半角固定・行末約物は句点のみ全角・段落先頭全角1字下げとします。

- ・ 行頭約物：半角固定とします。
- ・ 行末約物：句点のみ全角，その他は半角固定とします。
- ・ 段落先頭：全角1字下げとします。

```

body { /* 日本語組版のための設定 */
  punctuation-trim: start end-except-fullstop adjacent;
  -ah-text-justify-trim: punctuation;
  -ah-text-autospace: ideograph-numeric ideograph-alpha;
  -ah-text-autospace-width: 25%;
  hanging-punctuation: none;
}
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 */
}

```

夏目漱石（なつめそうせき、慶應三年一月五日（一八六七年二月九日）―大正五年（一九一六年）一二月九日）は、日本の小説家、評論家、英文学者。本名、金之助。『吾輩は猫である』『こゝろ』などの作品で広く知られる、森鷗外と並ぶ明治・大正時代の文豪である。江戸の牛込馬場下横町（現在の東京都新宿区喜久井町）出身。俳号は愚陀仏。

大学時代に正岡子規と出会い、俳句を学ぶ。：（中略）：当初は余裕派と呼ばれた。

「修善寺の大患」後は、『行人』『こゝろ』『硝子戸の中』などを執筆。「則天去私」（そくてんきよし）の境地に達したといわれる。晩年は胃潰瘍に悩まされ、『明暗』が絶筆となった。

フリー百科事典『ウィキペディア』より引用

17.9 行頭行末約物半角固定・段落先頭全角1字下げ（起こし食い込み）

行頭約物半角固定・行末約物半角固定・段落先頭の開き括弧類は二分下げとします。

- ・ 行頭約物：半角固定とします。
- ・ 行末約物：半角固定とします。
- ・ 段落先頭：全角1字下げとしますが、約物は二分下げとします。

```
body { /* 日本語組版のための設定 */
  punctuation-trim: start end adjacent;
  -ah-text-justify-trim: punctuation;
  -ah-text-autospace: ideograph-numeric ideograph-alpha;
  -ah-text-autospace-width: 25%;
  hanging-punctuation: first;
}
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 */
}
```

夏目漱石（なつめそうせき、慶應三年一月五日（一八六七年二月九日）―大正五年（一九一六年）一二月九日）は、日本の小説家、評論家、英文学者。本名、金之助。『吾輩は猫である』『こゝろ』などの作品で広く知られる、森鷗外と並ぶ明治・大正時代の文豪である。江戸の牛込馬場下横町（現在の東京都新宿区喜久井町）出身。俳号は愚陀仏。

大学時代に正岡子規と出会い、俳句を学ぶ。：（中略）：当初は余裕派と呼ばれた。

「修善寺の大患」後は、『行人』『こゝろ』『硝子戸の中』などを執筆。「則天去私」（そくてんきよし）の境地に達したといわれる。晩年は胃潰瘍に悩まされ、『明暗』が絶筆となった。

フリー百科事典『ウィキペディア』より引用

17.10 縦中横

縦書きの中に縦中横（たてちゅうよこ）を指定します。（縦書きの行の中に数字などを横書きにする）

```
/* 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; /* 字下げなし（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
  <span class="TateChuYoko">12</span>月<span class="TateChuYoko">8</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12月8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。

17.11 ルビ・圏点

17

XHTML では、<ruby> 要素を使うと、文字にルビ（フリガナ）を振ることができます。<rb> 要素で親文字（ルビを振る対象となる文字）を指定し、<rt> 要素でルビ文字（ルビの内容）を指定します。

これを CSS を使って実現する例を次に示しています。

```
/* Ruby ('html.css'より引用) */
ruby { display: ruby }
rb { display: ruby-base }
rt { display: ruby-text }
rp { display: none }

<p><ruby><rb>吾輩</rb><rp></rp><rt>わがはい</rt><rp></rp></ruby>は猫である。</p>
<p>名前はまだ無い。どこで生れたかんと<ruby><rb>見当</rb><rp></rp><rt>けんとう</rt><rp></rp></ruby>がつかぬ。</p>
```

横書き

わがはい
吾輩は猫である。

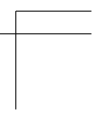
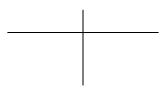
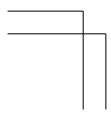
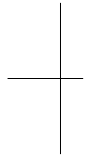
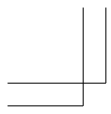
名前はまだ無い。どこで生れたかんとけんとう見当がつかぬ。

縦書き
吾輩^{わがはい}は猫である。
名前はまだ無い。どこで生
れたかとうと見当^{けんとう}がつかぬ。

AH Formatter では、文字列に圈点を付けることができます。ただし、圈点を任意のスク
リプトに付けることはできません。CJK 以外では、Latn / Cyrl / Grek / Zyyy にのみ適用で
きます。

```
/* 圈点 */  
span.kenten {  
  -ah-text-emphasis-style: filled;  
  -ah-text-emphasis-offset: 0.25;  
  -ah-text-emphasis-font-size: 0.25;  
}  
  
<p>ここを<span class="kenten">圈点で強調</span>します。 </p>
```

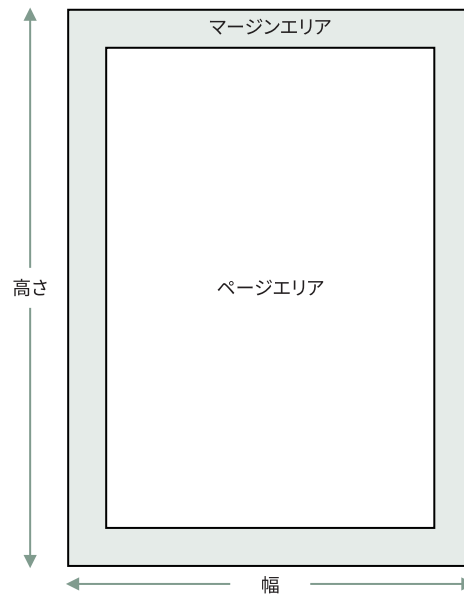
ここを^{●●●●}圈点で強調します。



第18章

ページ設定

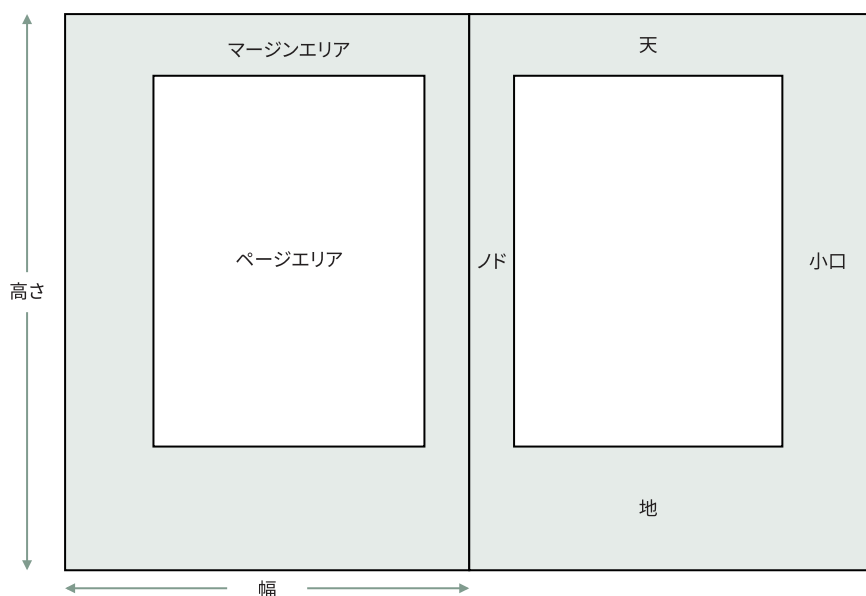
ページ媒体は、ひとつ以上のページボックスとして組版されます。ページボックスのコンテンツエリアはページエリアと呼ばれます。ページボックスのマージンエリアには、柱やノンブルなどを置きます。



ページエリアとマージンエリア

18.1 欧文ページデザイン

欧文ページデザインの伝統は、ページの視覚的な中心は幾何学的中心上にあると考えられるので、ページエリアをページを中心より上に置きます。ノド側のマージンは伝統的に小口側のマージンよりも狭いです。これはページから別のページに目線が移動しやすいようにするためです。小口側のマージンが広いと、指でページがめくりやすいこともあります。ノド側のマージンは2本あるので、小口側の広いマージンとバランスを保てます。ページエリアの高さは、ページの幅に一致します。



伝統的な欧文ページデザイン

実際には、ページデザインはさまざまです。印刷する「ページ」には、医薬品、販売パンフレット、児童書、部品カタログのパッケージ折り込みなどがあります。経済面を考えたリまたはオフィスプリンタで A4 またはレターサイズ用の紙に印刷する必要があると、ページのデザイン性に影響を与えるかもしれません。ページエリアが見開きページで同じ位置にあるが、ページデザインが左右非対称なことは、かつては斬新なアイデアでしたが、現在は珍しいことではありません。さらに、小説は、特にページサイズの異なる製本版で販売されることが多いので、マージンを減らして同じページエリアを再利用することがあります。

18

18.2 日本語ページデザイン

日本語は横書きまたは縦書きで組むことができます。横書きのときページの進行方向は左から右、縦書きのときページの進行方向は右から左になります。横書きと縦書きが混在する場合があります。出版物の種類によって縦書き、横書きを使う傾向がほぼ決まっています。例えば、公文書や教育資料は主に横書きですが、小説は主に縦書きです。

縦書き横書きの両方の文書で、日本語組版ではページエリアの幅を全角文字数で指定するのが標準です。@page ルールの width プロパティに em を使用して、幅を文字数で設定します。左右のマージンの値を auto に設定すると、ページの中央にページエリアを配置します。

```
@page {
  size: A4;
```

```
width: 43em; /* ページエリアの幅を全角43文字に */
margin-top: 30mm;
margin-bottom: 30mm;
margin-left: auto; /* ページエリアをページの中央に配置 */
margin-right: auto;
}
```

18.3 @page ルール

ページサイズ、マージン、ページヘッダ、ページフッタなどの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);
  }
}
```

18.4 ページボックスのサイズ：size プロパティ

●初期値：auto ●適用対象：@page ●値の継承：しない

ページボックスの幅と高さを size プロパティで指定します。

- auto：UA が選択したサイズと用紙の方向。
- <length> <length>?：ひとつの長さの値で、ページボックスの高さと幅を両方設定します。ふたつの長さの値で、それぞれページボックスの幅と高さを設定します。
- ページサイズ：A4, Letter などの規定のページ名。
- portrait：用紙を縦長に使う。
- landscape：用紙を横長に使う。

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

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

size プロパティには、A5, A4, A3, B5, B4, JIS-B5, JIS-B4, letter, legal ledger などの規定のページサイズ名を使えます¹⁸⁾。

18) size プロパティは、ISO 標準の名前を使いますが、ISO の B シリーズと JIS で規定している B シリーズでの寸法は異なるため、AH Formatter の独自拡張として、JIS-B5, JIS-B4などを追加しています。これはまだ CSS ワーキングドラフトになっていません。

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

```
@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、横方向の印刷指定ができます。

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

18.5 ページマージン：margin プロパティ

●初期値：各プロパティを参照 ●適用対象：すべての要素 ●値の継承：しない

`margin` プロパティは、`margin-top`、`margin-bottom`、`margin-left`、`margin-right` のショートハンドです。

- 値をひとつ指定：指定した値が [上下左右] のマージンになります。
- 値をふたつ指定：記述した順に [上下] [左右] のマージンになります。
- 値を三つ指定：記述した順に [上] [左右] [下] のマージンになります。
- 値を四つ指定：記述した順に [上] [右] [下] [左] のマージンになります。

`@page` ルールの `margin` プロパティでページマージンを指定します。

```
@page {
  margin: 10%; /* 上下左右のマージンは、ページ幅の10%ずつ */
}
```

```
@page {
  /* 上下マージンを2cm, 左右マージンを3cmとする */
  margin-top: 2cm;
  margin-bottom: 2cm;
  margin-left: 3cm;
  margin-right: 3cm;
}
```

`margin-*` が指定されていないときの初期値はゼロです¹⁹⁾。(X)HTML の `<body>` 要素に指定された `margin-*` は、ページエリアの内側に取られます。したがって、(X)HTML で `@page` と `<html>` (または `<body>`) 要素の両方に `margin-*` を指定しているとマージンは合計になります。

19) AH Formatter では、初期値を任意の値に設定できます。ページマージンの既定値として、2cm などが設定されていることがあります。そのとき、初期値はゼロにはなりません。

18.6 名前付きページタイプ：page プロパティ

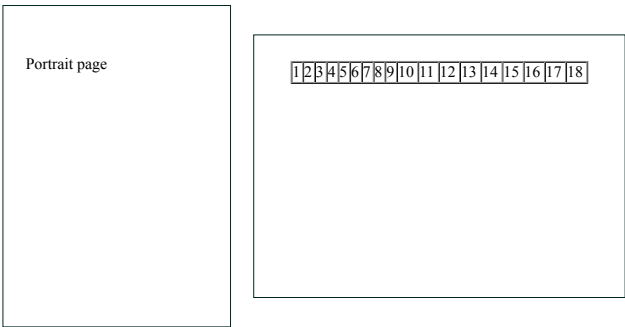
●初期値：auto ●適用対象：クラス 1 の改ページを作成するボックス ●値の継承：しない

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 { /* Landscape名前付きページ 定義 */
  size: A4 landscape;
}
@page Portrait { /* Portrait 名前付きページ定義 */
  size: A4;
}
table.WideTable {
  page: Landscape; /* 横に大きな表を Landscape ページに */
}
html {
  page: Portrait; /* Portrait ページを既定として使用 */
}

<p>Portrait page</p>

<table class="WideTable" border="1" style="width:100%">
<tr>
<td>1</td>
...
<td>18</td>
</tr>
</table>
```



page プロパティは名前付きページを選択します

18.7 ページ数の制約：-ah-force-page-count プロパティ

●初期値：auto ●適用対象：@page ●値の継承：しない

文書の総ページ数に制約を課します。制約が満たされない場合、ページの末尾に追加のページ（または複数ページ）が追加されます。ページ数を制約する理由としては、用紙の両面に印刷されたオフィス文書は偶数ページが必要な場合があったり、小冊子として印刷

する文書は、4 ページの倍数、商業用印刷機で印刷する文書は、1 枚の用紙に 8、16、32 または 64 ページで印刷し、折り畳んで、裁ち落とす場合があるからです。

- auto : ページ数を強制しません。
- even : 偶数ページに強制します。
- odd : 奇数ページに制御します。
- doubly-even : ページ数を強制的に 4 の倍数にします。
- end-on-even : 最後のページを強制的に偶数ページにします。
- end-on-odd : 最後のページを強制的に奇数ページにします。
- end-on-doubly-even : 最後のページ番号を強制的に 4 の倍数にします。
- even-document : 文書先頭から数えたページ数を強制的に偶数にします。
- odd-document : 文書先頭から数えたページ数を強制的に奇数にします。
- doubly-even-document : 文書先頭から数えたページ数を強制的に 4 の倍数にします。
- [end-on | document] <number> [<number>] : 存在する場合、ページ数を、強制的にページの最初の数 + 2 番目の数の倍数にします。
- no-force : 偶数ページも奇数ページも制御しません。

例えば、5 ページの文書を生成する場合：

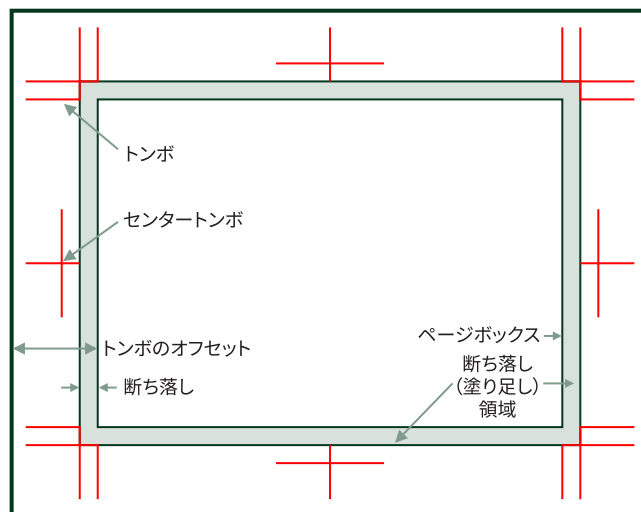
- -ah-force-page-count: even;
-ah-force-page-count: 2; と等価です。
- -ah-force-page-count: odd;
-ah-force-page-count: 2 1; と等価です。
- -ah-force-page-count: doubly-even;
-ah-force-page-count: 4; と等価です。
- -ah-force-page-count: end-on-doubly-even;
-ah-force-page-count: end-on 4; と等価です。
- -ah-force-page-count: end-on 2;
総ページ数は 6 ページとなります。1 ページは空白ページです。
- -ah-force-page-count: end-on 4;
総ページ数は 8 ページとなります。3 ページは空白ページです。
- -ah-force-page-count: end-on 4 1;
総ページ数は 5 ページとなります。空白ページはありません。
- -ah-force-page-count: end-on 4 3;
総ページ数は 7 ページとなります。2 ページは空白ページです。

- `-ah-force-page-count: end-on 6 5;`
総ページ数は5ページとなります。空白ページはありません。
- `-ah-force-page-count: end-on 6 4;`
総ページ数は10ページとなります。5ページは空白ページです。

18.8 トンボ

@page ルールはページボックスを定義します。ページボックスの大きさは @page ルールの `size` で定義しますが、それよりも大きな用紙に印刷できます。その目的は、画像やその他のコンテンツをページボックスの端まで拡張するためです。プリンタのような物理的デバイスは、通常、用紙の端のまわりに印刷できないエリアを有しており、そこにコンテンツがあれば、印刷できません。大きな用紙にページボックスを印刷し、ページボックスのサイズに合わせて用紙を詰めると、印刷できないエリアの問題が回避されます。画像などをページボックスの外側の裁ち落としエリアに拡張すると、仕上がり寸法が不正確な場合の問題を回避できます。

コーナートンボとセンタートンボは、ページボックスの外側に印刷され、用紙のサイズを仕上がり寸法に処理し、両面に印刷されたコンテンツが正確に位置していることを確認するガイドとして使用します。ページボックスの外側に印刷される情報としては、多色刷りの見当合せのための色見本、ページ、ページを含む文書、バージョン番号などを識別する情報などがあります。




トンボ (クロップマーク) の用語

18.8.1 トンボの表示：marks プロパティ

●初期値：none ●適用対象：CSS @page ●値の継承：しない

印刷時にトンボを印刷するかどうかを指定します。

- crop：コーナートンボ（crop marks）を出力します。
- crop-trim：仕上がり寸法の位置にのみコーナートンボを出力します。（内トンボ）
- cross：センタートンボ（cross marks, registration marks）を出力します。
- cross-circle：十字のセンタートンボに円を付けて出力します。
- crop-registration：西洋トンボ（registration marks）を出力します。
- <uri-specification>：カスタムトンボ，色見本バーなどを表現する SVG などの画像の位置を指定します。複数の画像を指定もできます。 



センタートンボとコーナートンボ

ページボックスの四隅にコーナートンボ，また四辺の中央にセンタートンボを出力できます。コーナートンボは，仕上がり寸法に詰め処理するための位置，センタートンボは，

多色刷りの見当合せのために、また用紙の表と裏の位置を合わせるのに役立ちます。外部参照画像の位置を指定して、カスタムトンボを出力することもできます。

```
@page {
  size: B5; /* ISO B5 (176mm 250mm) */
  margin: 28mm 21.325mm;
  marks: crop cross; /* トンボの表示 */
  -ah-crop-offset: 14mm; /* 出力媒体の物理的な端から仕上がり寸法までの距離 */
  -ah-printer-marks-line-color: auto; /* トンボの線の色 */
  -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 */
  bleed: 3mm; /* 裁ち落としの長さ */
}
```

18.8.2 トンボの可視性：-ah-crop-area-visibility プロパティ

- 初期値：非表示 ●適用対象：CSS @page ●値の継承：しない
- 仕上がり寸法からはみ出したエリアを表示するかどうか指定します。
- hidden：塗り足し部分は、非表示にします。
 - visible：塗り足し部分は、表示します。

18.8.3 トンボの線の色：-ah-printer-marks-line-color プロパティ

- 初期値：auto ●適用対象：CSS @page ●値の継承：しない
- トンボの線の色を指定します。値が auto のときは、レジストレーションカラーとみなされます。

18.8.4 トンボの線の長さ：-ah-printer-marks-line-length プロパティ

- 初期値：auto ●適用対象：CSS @page ●値の継承：しない
- トンボの線の長さを指定します。値が auto のときは、システム既定値に依存します。トンボの既定値の長さは 10mm ですが、変更できます。

18.8.5 トンボの線の幅：-ah-printer-marks-line-width プロパティ

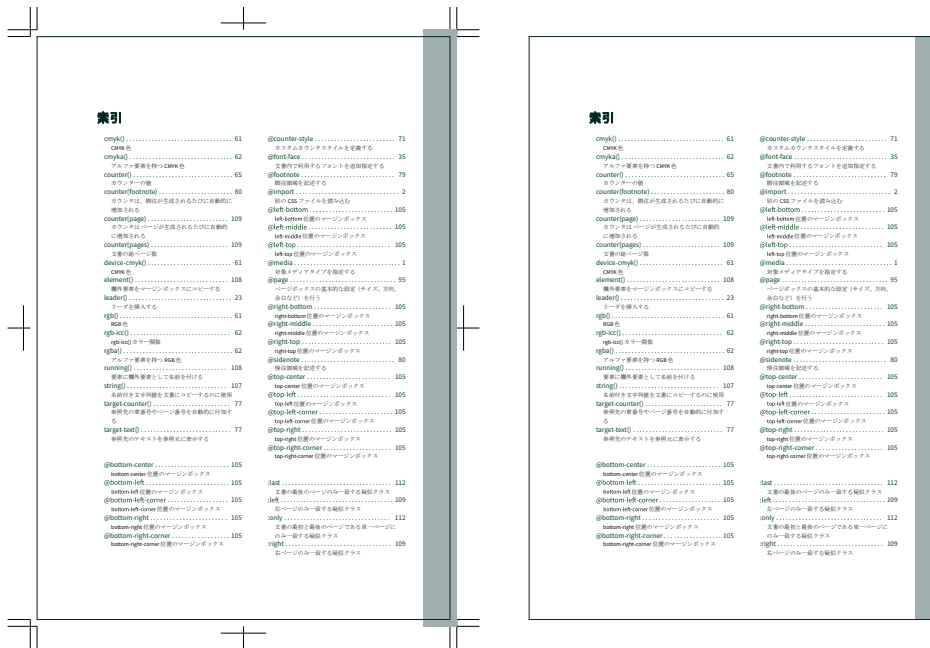
- 初期値：auto ●適用対象：CSS @page ●値の継承：しない
- トンボの線の幅を指定します。値が auto のときは、システム既定値に依存します。トンボの既定値の幅は 0.2pt ですが、変更できます。

18.8.6 出力媒体の端から仕上がり寸法までの距離：-ah-crop-offset プロパティ

- 初期値：0 ●適用対象：CSS @page ●値の継承：しない
- 出力媒体の物理的な端から仕上がり寸法までの距離を指定します。

18.9 裁ち落とし (塗り足し)

画像などでは、ページの裁ち落としの端を超えて塗り足し部分まで一杯に入れることがあります。画像をページの端まで一杯に拡張するのは、有意義な効果があります。ページの端までぎりぎりに拡張する場合、不正確な裁ち切り処理で画像の外端側に空白を残す可能性があります。画像をページの端を超えて拡張し、仕上がりサイズに裁ち切ることで、このような問題を回避できます。



裁ち落とし (塗り足し)

18

18.9.1 裁ち落とし (塗り足し) の幅 : bleed プロパティ

●初期値 : 0 ●適用対象 : CSS @page ●値の継承 : しない

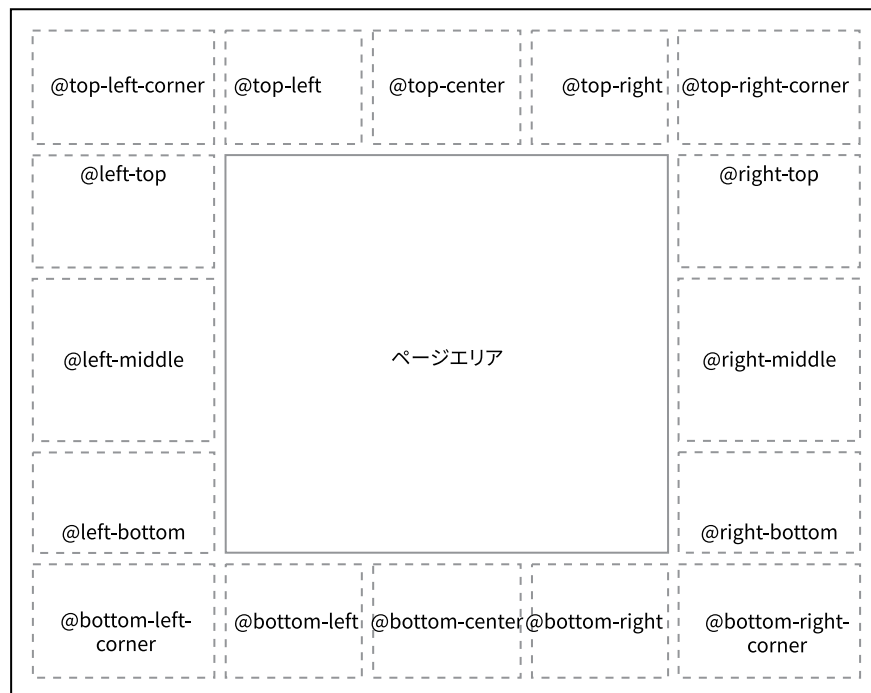
裁ち落とし (塗り足し) の幅を指定します。裁ち落としエリアは、仕上がり寸法の外側に取られます。マージンボックスのマージンに負の値を指定することで、裁ち落としエリアまでブロックの範囲を伸ばすことができます。

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いています。



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 {
  /* 右先頭ページのタイトル*/
  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-except: 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;
}
```

19.2.5 running 要素を挿入：element()

running 要素をマージンボックスにコピーするのに使用します。

position: running(name); を使用して、あるべき配置から取り出した要素は、柱 (ランニングヘッダ) 内で content: element(name); として参照されます。running 要素は、疑似要素とその子孫を含め、ひとつの要素を持つことができます。string() と異なり、element() は他の値と組み合わせることができません。

running 要素は、文書の普通の配置を継承します。

必要とされる最初の値は、running 要素名前です。

```
@top-left {
  /* 左ページ見出しのタイトル*/
  content: element(Title);
}
```

ひとつのページの複数要素がそれぞれの同名の running 要素の値を設定する場合、running 要素はそのページに複数の値を持つことがあります。element() 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 running 要素のエントリ値を使用します。
- 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 start 値を使用します。
- last: そのページの running 要素の退出値を使用します。
- first-except: first と似ていますが、値が割り当てられているページに空文字列が使用されます。

19.2.6 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²⁰⁾。

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

19

19.2.7 総ページ数：counter(pages)

"Page 118 of 142" 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: counter(pages); のページカウンタ (実際の物理ページ)
  }
}
```

²⁰⁾ counter() は CSS2.1 で規定されていますが、CSS3 ではページコンテキストにページ番号を表す専用のカウンタが導入されます。

19.3 左・右・先頭・空白ページ: :left : :right : :first : :blank

ページのマージンは、ページのマージンボックスの内容とスタイル同様に :blank 擬似クラスを使用してコンテンツがあるかどうかを考慮して設定できます。例えば、左右のページに異なる柱を設定できます。最初のページで柱の内容が異なる場合があります。最初のページの左ページの左側と右ページの右側にあるタイトルとページ番号を非表示にすることもできます。

名前付きページと一緒に使用すると、左ページと右ページ、および名前付きページの最初のページのスタイルを指定できます。擬似クラスを組み合わせると、例えば、空白の左ページに対して別のページルールを作成することは可能です。

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

  @top-left { /* 左ページの柱に本のタイトル */
    content: string(Title);
  }
  @bottom-left { /* ノンブル (ページ番号) */
    content: counter(page);
  }
}

@page Chapter: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 Chapter:first { /* Chapterの先頭ページ設定 */
  /* ページ見出し非表示 */
  @top-right { content: none; }
  @top-left { content: none; }
}

@page Preface:blank { /* 前置きと目次の間の空白ページ */
  @bottom-left { content: none; }
  @bottom-right { content: none; }
}
```

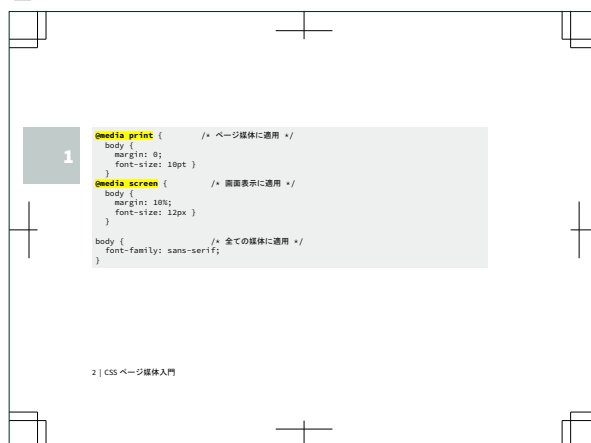
最初のページ



右ページ



左ページ



19.4 最終ページ・単一ページ :last ・ :only

このふたつは AH Formatter の拡張です。:last は最後のページ、:only は唯一のページを指定します。

19.5 進行方向 :writing-mode プロパティ

●初期値 : lr-tb ●適用対象 : すべての要素 ●値の継承 : する

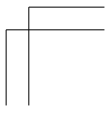
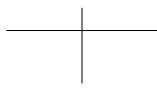
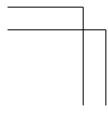
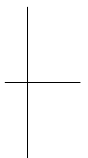
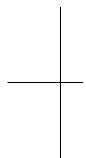
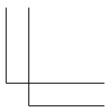
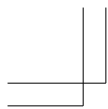
文書全体の文字、行、ページの進行方向を設定するには、ルート要素に `writing-mode` を指定します。初期値は `writing-mode: lr-tb` (left-to-right, top-to-bottom) で、横書き、すなわち文字の進行方向は左から右、行とブロックの進行方向は上から下、ページは左から右への進行となります²¹⁾。

日本語の縦書きを指定するには、`writing-mode: tb-rl` を指定します。文字の進行方向が上から下、行とブロックの進行方向が右から左、ページの進行方向が右から左となります。

- `lr-tb` : 文字の進行方向が左から右、行とブロックの進行方向が上から下となります。通常の横書きです。
- `rl-tb` : 文字の進行方向が右から左、行とブロックの進行方向が上から下となります。アラビア語やヘブライ語などの右から左に書く言語の場合に使います。
- `tb-rl` : 文字の進行方向が上から下、行とブロックの進行方向が右から左となります。日本語などの縦書きで使います。

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

21) ルート要素の `writing-mode` の設定は、ページボックスに継承され、さらにマージンボックスに継承されます。



PDF 出力

Portable Document Format (PDF) は、ページ組版の最も一般的な出力形式です。AH Formatter は PDF の機能を駆使するための複数の拡張を提供しています。

20.1 PDF バージョン

PDF 仕様は、最初のバージョンが 1993 年に公開されて以来、複数のバージョンを公開しています。公開ごとに新機能が追加されました。PDF バージョンで廃止されていない機能はすべて、後続のバージョンにも含まれます。PDF ファイルには、適合する PDF バージョンを識別する情報が含まれています。PDF リーダは、ファイルのバージョンがリーダーが実装しているバージョンより新しい場合でも、すべての PDF ファイルを読み取ろうとします。

PDF 仕様バージョン

バージョン	年	Acrobat Reader バージョン
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	-

ISO によって標準化された PDF の特殊なサブセットもあります。これらの中には、それぞれの特定の PDF バージョンに基づいた複数バージョンを持つものもあります²²⁾。

- PDF/X：PDF for Exchange
- PDF/A：PDF for Archiving
- PDF/E：PDF for Engineering
- PDF/VT：PDF for exchange of variable data and transactional (VT) printing
- PDF/UA：PDF for Universal Accessibility

²²⁾ AH Formatter は PDF/E、PDF/VT のいずれも生成しません。

20.2 タグ付き PDF

「タグ付き PDF」は、個別の PDF 仕様ではありません。文書の論理構造に関する追加情報をもつ PDF ファイルです。タグ付き PDF は、最初に PDF 1.4 で定義されました。

タグ付き PDF のテキスト、グラフィクス、イメージは、例えば視覚障害のあるユーザがコンテンツにアクセスしたり、他の目的で抽出して再利用できます。PDF/UA は、追加条件にも適合するタグ付き PDF ファイルです。

AH Formatter は、HTML/CSS 要素と疑似要素の PDF タグ (StructElem) を次の表が示すように埋め込みます。

HTML の要素	PDF の要素
html	Document
div	Div
h1	H1
h2	H2
h3	H3
h4	H4
h5	H5
h6	H6
p	P
ul	L
ol	L
li	LI
li::marker	Lbl
dl	L
dt	Lbl
dd	LBody
blockquote	BlockQuote

HTML の要素	PDF の要素
caption	Caption
table	Table
tr	TR
td	TD
th	TH
thead	THead
tfoot	TFoot
tbody	TBody
ruby	Ruby
rb	RB
rt	RT
span	Span
img	Figure
a[href]	Link
その他のブロック要素	Div
その他のインライン要素	Span

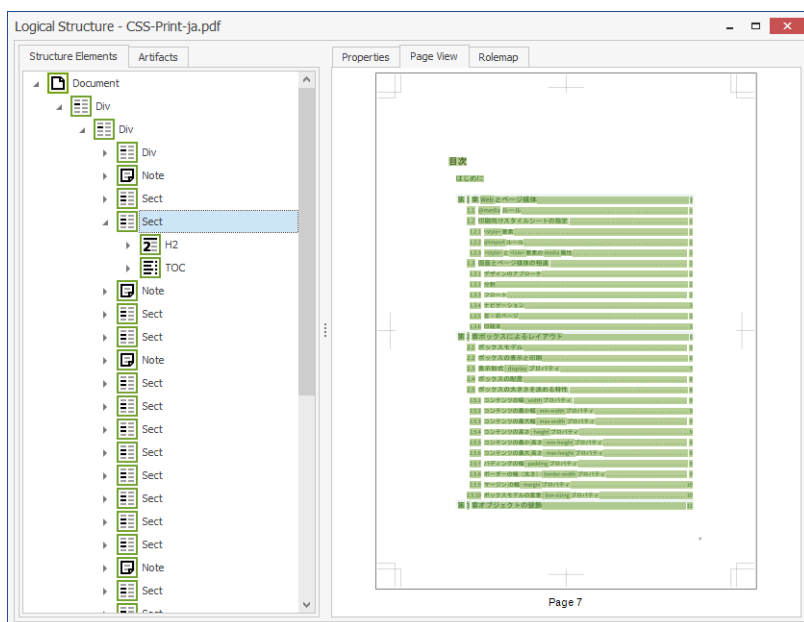
20.2.1 カスタム PDF タグ名: -ah-pdftag プロパティ

- 初期値: 空文字列
- 適用対象: すべての要素
- 値の継承: しない

タグ付き PDF を生成する際に、指定された名前が標準の PDF タグ名でない場合、PDF のロールマップを生成して、その名前を標準の PDF タグ名へ割り当てます。提供された名前が標準の PDF タグ名のひとつである場合、その名前はそのまま使用されます。

20

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



<div class="TOC"> は PDF では「Sect」とタグ付けされています。

20.3 PDF/X

PDF/X は、ISO 15930 で規定されており、印刷用のグラフィクス交換を目的とした PDF のサブセットです。基本的には、印刷するためのすべての情報が PDF ファイル内に含まれています。

AH Formatter は、複数の PDF/X バージョンに対応しています。

- PDF/X-1a:2001 (ISO 15930-1:2001)
PDF 1.3 ベースの仕様です。
- PDF/X-3:2002 (ISO 15930-3:2002)
PDF 1.3 ベースの仕様です。
- PDF/X-1a:2003 (ISO 15930-4:2003)
PDF 1.4 ベースの仕様で、PDF/X-3:2003 のサブセットです。
- PDF/X-2:2003 (ISO 15930-5:2003)
PDF 1.4 ベースの仕様で、PDF/X-3:2003 のスーパーセットです。
- PDF/X-3:2003 (ISO 15930-6:2003)
PDF 1.4 ベースの仕様です。
- PDF/X-4:2010 (ISO 15930-7:2008)
PDF 1.6 ベースの仕様です。

PDF/Xの大きな特徴

要件	PDF/X-1a	PDF/X-2	PDF/X-3	PDF/X-4
フォントはすべて埋め込まなければならない	○	○	○	○
出力 intents が指定されていなければならない	○	○	○	○
出力 intents として指定された ICC カラープロファイルを埋め込まなければならない	×	×	×	○
CMYK, スポットカラーをサポートする	○	○	○	○
グレースケールをサポートする	○	○	○	○
RGB をサポートする	×	○	○	○
透明をサポートする	×	×	×	○
パスワードの設定や, 印刷制限, 変更制限などを行ってよい	×	×	×	×
リンクや注釈などを含めてよい	×	×	×	×

20.4 PDF/A

PDF/A は ISO 19005 で規定されており, 電子文書の長期保存を目的とした PDF のサブセットです。

PDF/Aの大きな特徴

要件	PDF/A-1a	PDF/A-1b	PDF/A-2a	PDF/A-2b	PDF/A-2u	PDF/A-3a	PDF/A-3b	PDF/A-3u
フォントはすべて埋め込まなければならない	○	○	○	○	○	○	○	○
ICC カラープロファイルを埋め込まなければならない	○	○	○	○	○	○	○	○
タグ付けされていなければならない	○	×	○	×	×	○	×	×
XMP 準拠のメタデータを含んでいなければならない	○	○	○	○	○	○	○	○
暗号化してよい	×	×	×	×	×	×	×	×
LZW 圧縮してよい	×	×	×	×	×	×	×	×
透明な画像を含めてよい	×	×	○	○	○	○	○	○
外部コンテンツを参照してよい	×	×	×	×	×	×	×	×
JavaScript を含めてよい	×	×	×	×	×	×	×	×
テキストに Unicode が使用されなければならない	○	×	○	×	○	○	×	○
PDF/A 以外の任意のファイルを注釈として添付できる	×	×	×	×	×	○	○	○

20.5 PDF/UA

PDF/UA は, ISO 14289-1 で規定されており, ISO 32000-1 (PDF 1.7) 仕様に基づいた PDF のアクセシビリティ向上を目的とした仕様です。

PDF/UA の大きな特徴は以下が挙げられます。

- 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。
- 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。
画像または数式の代替テキストは、-ah-alttext プロパティ、リンクは -ah-annotation-contents プロパティで指定できます。
- アクセシビリティを禁止しない暗号化を行わなければならない。
- PDF/UA 内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。
- 文書の言語を指定しなければならない。
- フォントはすべて埋め込まれなければならない。

20.5.1 マッターホルン プロトコル

PDF 協会（PDF Association）が公開したマッターホルンプロトコル（Matterhorn Protocol）は、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です。マッターホルンプロトコル²³⁾ 文書は、136 の失敗条件から成る 31 のチェックポイントで構成されています。一部の失敗条件は、プログラミングでチェックできますが、人の手によるレビューが必要な場合があります。

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

表に関するマッターホルンプロトコル失敗条件

20.5.2 PAC 3 PDF/UA チェッカー

Access For All による PDF アクセシビリティ チェッカー³²⁴⁾（PAC 3）は、PDF/UA 準拠の PDF ファイルをチェックするフリーウェアの Windows 版用のユーティリティです。プログラムはマッターホルンプロトコルチェックを実装しています。PAC 3 で PDF ファイルを開くと、プログラムはチェックを実行し、結果の要約を表示します。インタラクティブ

23) <https://www.pdfa.org/publication/the-matterhorn-protocol-1-02/>

24) <http://www.access-for-all.ch/en/pdf-lab/536-pdf-accessibility-checker-pac-3.html>

なチェックがないので、プログラムは、人の手によるチェックを必要とする失敗について警告するだけです。



PAC 3 PDF/UA チェッカー

20.6 文書 プロパティ

この拡張は <meta> 要素を使用します。以下に例を示します。

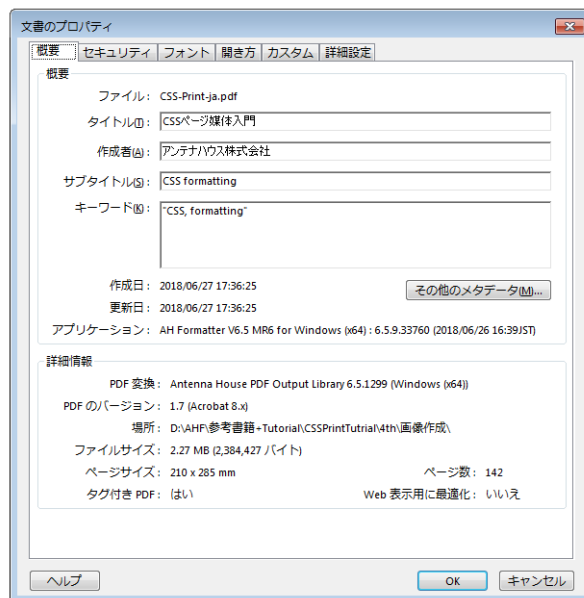
```
<meta name="document-title" content="The document title"/>
<meta name="subject" content="The document subject"/>
<meta name="author" content="The author"/>
<meta name="keywords" content="Comma, separated, keywords, list"/>
...
```

以下の name 値を持つ <meta> は、PDF の文書カタログに格納されている情報を提供します。

- document-title : 文書のタイトル
- subject : 文書のサブタイトル
- author : 文書の著者名
- author-title : 文書の著者の役職など
- description-writer : 文書の説明の著者
- keywords : カンマ区切りで列挙したキーワードのリスト
- copyright-status : 著作権のステータス Unknown, Copyrighted, PublicDomain でなければなりません。

- copyright-notice：著作権の情報
- copyright-info-url：著作権情報の URL
- xmp：プロパティ情報を含む XMP の URL

CSS ページ媒体入門



Acrobat Reader で表示される文書プロパティ

20.6.1 Extensible Metadata Platform (XMP)

XMP は、ファイルまたは画像についてメタデータを表現するための標準 XML 形式です。元々は Adobe で開発され、現在は ISO 16684-1:2012 で標準化されています。標準であるため、例えば、デジタルカメラで写した写真に埋め込まれている XMP は、画像を編集する写真編集プログラムによって変更または拡張できます。また、PDF ファイル内のすべての画像の XMP を、PDF ファイルに埋め込むことが可能です。

XMP 標準自体は、Dublin Core、RDF などの他のメタデータ標準に一部基づいています。AH Formatter が生成する PDF には XMP が埋め込まれています。

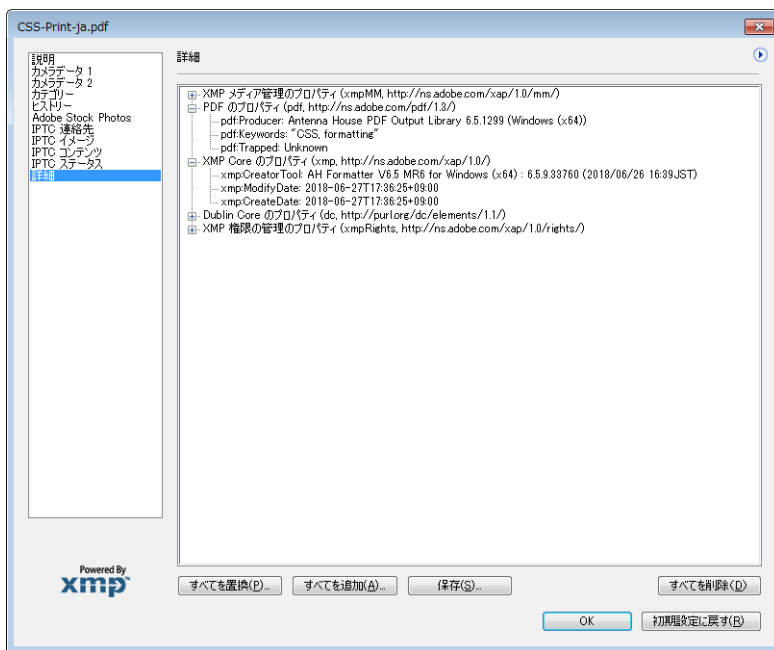
PDF ファイルから抽出された XMP の一部を以下に示します。次の図は、Acrobat で表示されたものと同じ XMP を示しています。

```
<?xpacket begin="" id="W5M0MpCehiHzreSzNTczkc9d"?>
<x:xmpmeta xmlns:x="adobe:ns:meta/"
  x:xmpptk="Adobe XMP Core 5.6-c015 84.159810, 2016/09/10-02:41:30">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <rdf:Description rdf:about="" ... >
  <pdfuaid:part>1</pdfuaid:part>
```

```

<xmpMM:DocumentID>uuid:3DF8C5A6-E99A-6946-98C4-644ABDB67511</xmpMM:DocumentID>
<xmpMM:VersionID>1</xmpMM:VersionID>
<xmpMM:RenditionClass>default</xmpMM:RenditionClass>
<pdf:Producer>Antenna House PDF Output Library 6.5.1299 (Windows
(x64))</pdf:Producer>
<pdf:Keywords>"CSS, formatting"</pdf:Keywords>
<pdf:Trapped>Unknown</pdf:Trapped>
...
<dc:description>
  <rdf:Alt>
    <rdf:li xml:lang="x-default">CSS formatting</rdf:li>
  </rdf:Alt>
</dc:description>
<dc:title>
  <rdf:Alt>
    <rdf:li xml:lang="x-default">CSSページ組版入門</rdf:li>
  </rdf:Alt>
</dc:title>
<dc:creator>
  <rdf:Seq>
    <rdf:li>アンテナハウス株式会社</rdf:li>
  </rdf:Seq>
</dc:creator>
...

```



Acrobat で表示された XMP プロパティ

20.7 ページ表示

次の name 値を持つ<meta> は、最初にリーダーで開いたときの PDF の表示方法を制御します。記述のない限り、利用できる値は true, false で、その name 値を持つ<meta> が存在しない場合、既定値は false となります。

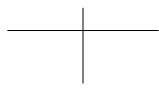
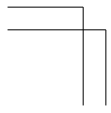
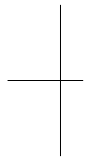
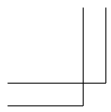
- pagemode：文書を開いたときの表示方法。UseNone, UseOutlines, UseThumbs, FullScreen または UseOC のいずれかの値でなければなりません。アウトラインがあるときの既定値は UseOutlines です。
- pagelayout：文書を開いたとき、使われるページのレイアウト。SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft または TwoPageRight のいずれかの値でなければなりません。既定値は SinglePage です。
- hidetoolbar：文書を開いたとき、ツールバーを隠すかどうかを指定。
- hidemenubar：文書を開いたとき、メニューバーを隠すかどうかを指定。
- hidewindowui：文書を開いたとき、ユーザインターフェイス（UI）要素（スクロールバーやナビゲーション用コントロールなど）を隠すかどうかを指定。
- fitwindow：文書を開いたとき、ページのサイズに適合するように文書ウィンドウのサイズを変更するかどうかを指定。
- centerwindow：文書を開いたとき、文書ウィンドウを画面の中央に配置するかどうかを指定。
- displaydoctitle：文書を開いたとき、文書のタイトルを表示するかどうかを指定。
- openaction：文書を開いたときに表示される宛先または実行されるアクションを指定。

セレクタと疑似要素

AH Formatter は、次の CSS セレクタと疑似要素に対応しています。

パターン	意味
*	すべての要素
E	E 型の要素
ns E	名前空間 ns の中の E 型の要素
E.warning	class warning に属する E 要素
E#myid	myid と同値の ID を持つ E 要素
E F	E 要素の子孫である F 要素
E > F	E 要素の子である F 要素
E + F	E 要素が直前に現れる F 要素
E ~ F	E 要素が前に現れる F 要素
E:not(s)	単純なセレクタ s と一致しない E 要素
E[foo]	foo 属性を持つ E 要素
E[foo="bar"]	foo 属性の値が bar と同値である E 要素
E[foo~="bar"]	foo 属性値が空白区切りの値のリストであり、その値のひとつが bar である E 要素
E[foo="en"]	foo 属性値が en で始まるハイフン区切りの値のリストである E 要素
E[foo^="bar"]	foo 属性値が bar で始まる E 要素
E[foo\$="bar"]	foo 属性値が bar で終わる E 要素
E[foo*="bar"]	foo 属性値が bar を含む E 要素
E:lang(ja)	日本語で書かれているとタグ付けされた E 型の要素
E:link	ハイパーリンクのアンカーである E 要素
E:root	E 要素、文書のルート
E:empty	子要素を持たない E 要素 (テキストノードもない)
E:nth-child(n)	E 要素、親の n 番目の子
E:nth-last-child(n)	E 要素、親の n 番目の子、最後の子から数える
E:first-child	E 要素、親の最初の子
E:last-child	E 要素、親の最後の子
E:only-child	E 要素、親の子のみ
E:nth-of-type(n)	E 要素、その型 (type) の n 番目の兄弟
E:nth-last-of-type(n)	E 要素、その型 (type) の n 番目の兄弟、最後の兄弟から数える
E:first-of-type	E 要素、その型の最初の兄弟
E:last-of-type	E 要素、その型 (type) の最後の兄弟
E:only-of-type	E 要素、その型 (type) の兄弟のみ
E::first-letter	E 要素の最初に組版された文字
E::first-line	E 要素の最初に組版された行
E::before	E 要素の前に生成されコンテンツ
E::after	E 要素の後に生成されコンテンツ

パターン	意味
E::marker	表示プロパティが list-item に設定されている E 要素のマーカ
E::footnote-call	脚注エリアに移動した E 要素の代わりに残された生成コンテンツ
E::footnote-marker	脚注エリアに移動した E 要素の ::before 疑似要素を置き換える生成コンテンツ
E::sidenote-call	傍注エリアに移動した E 要素の代わりに残された生成コンテンツ
E::sidenote-marker	傍注エリアに移動した E 要素の ::before 疑似要素を置き換える生成コンテンツ



参照仕様

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

- ✓ **[CSS 2.1] Cascading Style Sheets Level 2 Revision 1 (CSS 2.1)**
..... W3C 勧告 2011-06-07 <https://www.w3.org/TR/2011/REC-CSS2-20110607/>
- ✓ **[CSS3-Background] CSS Backgrounds and Borders Module Level 3**
..... W3C 勧告 2017-10-17 <https://www.w3.org/TR/2017/CR-css-backgrounds-3-20171017/>
- ✓ **[CSS3-Box] CSS basic box model**
..... W3C ドラフト仕様 2007-08-09 <https://www.w3.org/TR/2007/WD-css3-box-20070809/>
- ✓ **[CSS3-Break] CSS Fragmentation Module Level 3**
..... W3C 勧告候補 2017-02-09 <https://www.w3.org/TR/2017/CR-css-break-3-20170209/>
- ✓ **[CSS3-Color] CSS Color Module Level 3**
..... W3C 勧告提案 2018-03-15 <https://www.w3.org/TR/2018/PR-css-color-3-20180315/>
- ✓ **[CSS3-Content] CSS Generated Content Module Level 3**
.. W3C ドラフト仕様 2016-06-02 <https://www.w3.org/TR/2016/WD-css-content-3-20160602/>
- ✓ **[CSS3-CounterStyles] CSS Counter Styles Level 3**
W3C 勧告候補 2017-12-14 <https://www.w3.org/TR/2017/CR-css-counter-styles-3-20171214/>
- ✓ **[CSS3-Fonts] CSS Fonts Module Level 3**
..... W3C 勧告候補 2018-03-15 <https://www.w3.org/TR/2018/CR-css-fonts-3-20180315/>
- ✓ **[CSS3-GCPM] CSS Generated Content for Paged Media Module**
... W3C ドラフト仕様 2014-05-13 <https://www.w3.org/TR/2014/WD-css-gcpm-3-20140513/>
- ✓ **[CSS3-Images] CSS Image Values and Replaced Content Module Level 3**
..... W3C 勧告候補 2012-04-17 <https://www.w3.org/TR/2012/CR-css3-images-20120417/>
- ✓ **[CSS3-Line] CSS3 module: line**
.. W3C ドラフト仕様 2002-05-15 <https://www.w3.org/TR/2002/WD-css3-linebox-20020515/>
- ✓ **[CSS3-Lists] CSS Lists and Counters Module Level 3**
.... W3C ドラフト仕様 2014-03-20 <https://www.w3.org/TR/2014/WD-css-lists-3-20140320/>
- ✓ **[CSS3-Multicol] CSS Multi-column Layout Module Level 1**
W3C ドラフト仕様 2017-10-05 <https://www.w3.org/TR/2017/WD-css-multicol-1-20171005/>
- ✓ **[CSS3-Namespaces] CSS Namespaces Module Level 3**
.... W3C 勧告 2011-09-29, 変更を追加 2014-3-20 <https://www.w3.org/TR/2014/REC-css-namespaces-3-20140320/>
- ✓ **[CSS3-Page] CSS Paged Media Module Level 3**
.... W3C ドラフト仕様 2013-03-14 <https://www.w3.org/TR/2013/WD-css3-page-20130314/>
- ✓ **[CSS3-Ruby] CSS Ruby Layout Module Level 1**
.... W3C ドラフト仕様 2014-08-05 <https://www.w3.org/TR/2014/WD-css-ruby-1-20140805/>
- ✓ **[CSS3-Selectors] Selectors Level 3**
..... W3C 勧告候補 2018-01-30 <https://www.w3.org/TR/2018/CR-selectors-3-20180130/>
- ✓ **[CSS3-Text] CSS Text Module Level 3**
..... W3C ドラフト仕様 2017-08-22 <https://www.w3.org/TR/2017/WD-css-text-3-20170822/>
- ✓ **[CSS3-TextDecor] CSS Text Decoration Module Level 3**
.... W3C 勧告候補 2013-08-01 <https://www.w3.org/TR/2013/CR-css-text-decor-3-20130801/>
- ✓ **[CSS3-Transforms] CSS Transforms Module Level 1**
..... W3C ドラフト仕様 2017-11-30 <https://www.w3.org/TR/2017/WD-css-transforms-1-20171130/>
- ✓ **[CSS3-UI] CSS Basic User Interface Module Level 3**
..... W3C 勧告提案 2017-12-14 <https://www.w3.org/TR/2017/PR-css-ui-3-20171214/>
- ✓ **[CSS3-Values] CSS Values and Units Module Level 3**
..... W3C 勧告候補 2016-09-29 <https://www.w3.org/TR/2016/CR-css-values-3-20160929/>
- ✓ **[CSS3-WritingModes] CSS Writing Modes Module Level 3**
W3C 勧告候補 2017-12-07 <https://www.w3.org/TR/2017/CR-css-writing-modes-3-20171207/>

- ✓ **CSS Snapshot 2017** W3C Working Group Note <https://www.w3.org/TR/css-2017/>
 - ✓ **HTML 5 — A vocabulary and associated APIs for HTML and XHTML**
..... W3C 勧告 <https://www.w3.org/TR/html5/>
 - ✓ **ISO 16684-1:2012, Graphic technology — Extensible metadata platform (XMP)
specification — Part 1: Data model, serialization and core properties**
..... ISO Standard <https://www.iso.org/standard/57421.html>
 - ✓ **Ready-made Counter Styles**
..... W3C Working Group Note <http://www.w3.org/TR/predefined-counter-styles/>
 - ✓ **Requirements for Japanese Text Layout**
..... W3C Working Group Note <http://www.w3.org/TR/jlreq/>
 - ✓ **Requirements for Latin Text Layout and Pagnation**
..... W3C ドラフト仕様 <https://www.w3.org/TR/dpub-latinreq/>
 - ✓ **Selectors Level 3** W3C 勧告 <https://www.w3.org/TR/css3-selectors/>
- AH Formatter は, XSL-FO による組版も行えます.
- ✓ **Extensible Stylesheet Language (XSL) Version 1.1**
..... W3C 勧告 <https://www.w3.org/TR/xsl/>

索引

<code>cmyk()</code>	59	<code>@footnote</code>	77
CMYK 色		脚注エリアを記述する	
<code>cmyka()</code>	60	<code>@import</code>	1
アルファ要素を持つ CMYK 色		別の CSS ファイルを読み込む	
<code>counter()</code>	63	<code>@left-bottom</code>	101
カウンタの値		left-bottom 位置のマージンボックス	
<code>counter(footnote)</code>	77	<code>@left-middle</code>	101
カウンタは、脚注が生成されるたびに自動的に		left-middle 位置のマージンボックス	
増加される		<code>@left-top</code>	101
<code>counter(page)</code>	104	left-top 位置のマージンボックス	
カウンタは、ページが生成されるたびに自動的に		<code>@media</code>	1
増加される		対象メディアタイプを指定する	
<code>counter(pages)</code>	104	<code>@page</code>	93
文書の総ページ数		ページボックスの基本的な設定（サイズ、方向、	
<code>device-cmyk()</code>	59	マージンなど）を行う	
CMYK 色		<code>@right-bottom</code>	101
<code>element()</code>	104	right-bottom 位置のマージンボックス	
running 要素をマージンボックスにコピーする		<code>@right-middle</code>	101
<code>leader()</code>	24	right-middle 位置のマージンボックス	
リーダーを挿入する		<code>@right-top</code>	101
<code>rgb()</code>	59	right-top 位置のマージンボックス	
RGB 色		<code>@sidenote</code>	78
<code>rgb-icc()</code>	60	傍注エリアを記述する	
rgb-icc() カラー関数		<code>@top-center</code>	101
<code>rgba()</code>	60	top-center 位置のマージンボックス	
アルファ要素を持つ RGB 色		<code>@top-left</code>	101
<code>running()</code>	103	top-left 位置のマージンボックス	
要素に running 要素として名前を付ける		<code>@top-left-corner</code>	101
<code>string()</code>	102	top-left-corner 位置のマージンボックス	
名前付き文字列値を文書にコピーするのに使用		<code>@top-right</code>	101
<code>target-counter()</code>	75	top-right 位置のマージンボックス	
参照先の章番号やページ番号を自動的に付加		<code>@top-right-corner</code>	101
する		top-right-corner 位置のマージンボックス	
<code>target-text()</code>	75	<code>:blank</code>	105
参照先のテキストを参照元に表示する		空白ページのみ一致する疑似クラス	
<code>@bottom-center</code>	101	<code>:first</code>	105
bottom-center 位置のマージンボックス		文書の最初のページのみ一致する疑似クラス	
<code>@bottom-left</code>	101	<code>:last</code>	107
bottom-left 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	
<code>@bottom-left-corner</code>	101	<code>:left</code>	105
bottom-left-corner 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	
<code>@bottom-right</code>	101	<code>:only</code>	107
bottom-right 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に	
<code>@bottom-right-corner</code>	101	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		<code>:right</code>	105
<code>@counter-style</code>	69	右ページのみ一致する疑似クラス	
カスタムカウンタスタイルを定義する		<code>::footnote-call</code>	77
<code>@font-face</code>	36	脚注エリアに移動した要素の残されたエリア	
文書内で利用するフォントを追加指定する			

::footnote-marker

border-top-right-radius

::footnote-marker 77
 同じ位置の各脚注に追加され、::before 疑似要素
 に置き換える疑似要素

::marker 68
 リスト項目マーカを表現する疑似要素

::sidenote 78
 傍注エリアに移動した要素の残されたエリア

::sidenote-marker 78
 同じ位置の各傍注に追加され、::before 疑似要素
 に置き換える疑似要素

-ah-baseline-block-snap 23
 行グリッドのブロックの配置

-ah-baseline-grid 23
 行グリッドを設定または解除する

-ah-column-rule-align 29
 段間罫の揃え

-ah-column-rule-display 30
 段間罫の表示

-ah-column-rule-length 29
 段間罫の長さ

-ah-crop-area-visibility 99
 仕上がり寸法からはみ出したエリアを表示する
 かどうか

-ah-crop-offset 99
 出力媒体の物理的な端から仕上がり寸法までの
 距離

-ah-force-page-count 95
 文書の総ページ数を制約する

-ah-line-height-shift-adjustment 20
 ベースラインシフトを持つコンテンツの行の高
 さを調整するかどうか

-ah-line-stacking-strategy 21
 お互いに隣接する行を配置するための戦略

-ah-pdftag 110
 出力する PDF タグ名

-ah-printer-marks-line-color 99
 トンボの線の色

-ah-printer-marks-line-length 99
 トンボの線の長さ

-ah-printer-marks-line-width 99
 トンボの線の幅

-ah-text-autospace 83
 日本語で、和欧文間にアキを入れるかどうかを
 指定

-ah-text-autospace-width 84
 日本語の和欧文間のアキの量

-ah-text-justify-trim 83
 日本語の追い込みの処理方法

-ah-text-line-width 38
 下線・上線・打ち消し線の幅

additive-symbols 71
 マーカ構築アルゴリズムで 사용되는記号を
 指定

background-color 14
 背景色

background-image 15
 Image to render on top of background color

background-position 16
 背景画像の位置

background-repeat 16
 背景画像の繰り返し

bleed 100
 仕上がり寸法の外側に取られる裁ち落としエリア

bookmark-label 79
 ブックマークのレベル

bookmark-level 79
 階層的ブックマーク構造のブックマークレベル

bookmark-state 79
 ブックマークの初期状態

border 13
 ボックスの四つのボーダーの幅、色、スタイル
 を設定するためのショートハンド

border-bottom 13
 ボーダーの下辺のスタイル

border-bottom-left-radius 14
 外側ボーダー端の左下の丸め半径

border-bottom-right-radius 14
 外側ボーダー端の右下の丸め半径

border-collapse 51
 表の各セルの周囲のボーダーを独立に扱うかど
 うかを指定

border-color 13
 四つの border-*-color プロパティを指定するシ
 ョートハンド

border-left 13
 左のボーダーのスタイル

border-radius 13
 外側ボーダー端の丸め半径

border-right 13
 右のボーダーのスタイル

border-spacing 51
 隣接するセルのボーダーどうしの間隔 (アキ量)

border-style 11
 四つの border-*-style プロパティを指定するシ
 ョートハンド

border-top 13
 上辺のボーダーのスタイル

border-top-left-radius 14
 外側ボーダー端の左上の丸め半径

border-top-right-radius 14
 外側ボーダー端の右上の丸め半径

border-width	9	font	33
四つの border-*-width プロパティを指定するシ ョートハンド		font-style, font-variant, font-weight, font- stretch, font-size, line-height, font-family を まとめて指定するためのショートハンド	
box-shadow	14	font-family	34
ドロップシャドウをボックスに添付する		フォントファミリー名またはジェネリックファミ リ名のリスト	
box-sizing	9	font-size	33
指定された幅と高さの内側または外側にパディ ングとボーダーを描画するかどうか		フォントのサイズ	
caption-side	54	font-style	35
表ボックスに対するキャプションボックスの 配置		フォントをイタリック体や斜体に指定	
clear	48	font-variant	35
右寄せまたは左寄せを指定したブロックへの回 り込みを解除		フォントをスモールキャピタルにする	
color	58	font-weight	34
文字色やボーダー色の前景色		フォントの太さ	
column-count	27	hanging-punctuation	84
ブロックの段数		行頭での突き出し、行末でのぶら下げを行うか どうか	
column-gap	28	height	8
段間の幅		コンテンツの高さ	
column-rule	28	hyphenate-after	26
column-rule-width, column-rule-style, column- rule-color を指定するためのショートハンド		ハイフン分割の後の最小文字数	
column-rule-color	29	hyphenate-before	25
段間罫の色		ハイフン分割の前の最小文字数	
column-rule-style	29	hyphens	24
段間罫のスタイル		ハイフネーションをするかどうかを制御する	
column-rule-width	29	line-height	36
段間罫の幅		行の高さ	
column-span	28	list-style	68
段抜きする要素の段数		リスト項目マーカのスタイルを指定するショ ートハンド	
column-width	27	list-style-image	67
段組の要素の段幅		リスト項目マーカ画像	
columns	28	list-style-position	67
column-width, column-count を指定するための ショートハンド		リスト項目マーカの位置	
content	64	list-style-type	67
要素の直前または直後に文字列を挿入する		リスト項目マーカの種類	
counter-increment	64	margin	9
カウンタの加算		四つの margin-* プロパティを指定するショ ートハンド	
counter-reset	65	margin-bottom	9
カウンタをリセットする		下のマージンの幅	
display	6	margin-left	9
要素の表示形式を制御する		左のマージンの幅	
fallback	71	margin-right	9
フォールバック カウンタスタイル		右のマージンの幅	
float	42	margin-top	9
ボックスをフロートさせて左寄せまたは右寄せ にする		上のマージンの幅	

marks		<rb>
marks	98	印刷時にトンボを印刷するかどうかを指定
max-height	8	コンテンツの最大の高さ
max-width	8	コンテンツの最大幅
min-height	8	コンテンツの最小の高さ
min-width	8	コンテンツの最小幅
negative	70	値が負の場合に表現を変更する方法
orphans	32	ページの下部に残さなければならない段落先頭の最小行数を指定
pad	71	カウンタ表現を埋め込む記号を指定
padding	8	四つの padding-* プロパティを指定するショートハンド
padding-bottom	8	ボックスの下部のパディングの幅
padding-left	8	ボックスの左側のパディングの幅
padding-right	8	ボックスの右側のパディングの幅
padding-top	8	ボックスの上部のパディングの幅
page	95	要素を表示する名前付きページを指定する
page-break-after	31	要素の後にどのように改ページをするか
page-break-before	31	要素の前でどのように改ページをするか
page-break-inside	31	要素内で改ページをするかどうか
prefix	70	マーカ表現の前に付ける記号
punctuation-trim	81	全角約物の詰めを処理するかどうか
range	71	カウンタスタイルが定義されている範囲
size	93	ページボックスの幅と高さ
string-set	102	文字列を名前付きの変数にする
suffix	71	マーカ表現に追加される記号
symbols	71	マーカ構築アルゴリズムで使用される記号
system	70	カウンタの表現を構成するために使用するアルゴリズム
table-layout	53	表や列の幅の決定方法
text-align	17	ブロックレベル要素内のテキストを左揃え、右揃え、中央揃え、均等配置にする
text-align-last	18	ブロックの最後の行 または強制改行の直前の行の配置方法
text-decoration	37	text-decoration-line, text-decoration-color, text-decoration-style のショートハンド
text-decoration-color	38	下線・上線・打ち消し線の色を指定
text-decoration-line	38	テキストに下線・上線・打ち消し線を指定
text-decoration-style	38	下線・上線・打ち消し線の種類を指定
vertical-align	37	縦方向の揃え位置や上付き文字、下付き文字を指定
widows	32	ページの上部に残さなければならない段落終わりの最小行数を指定
width	8	コンテンツの幅
writing-mode	107	文書全体の文字、行、ページの進行方向を設定
	41	文章中に画像を配置
<link>	2	外部スタイルシートのリンク
<meta>	114	文書に関する追加情報を提供
<object>	41	外部リソースの埋め込みに使用される
<rb>	88	ルビの親文字を指定

<rt>

<style>

<rt> 88

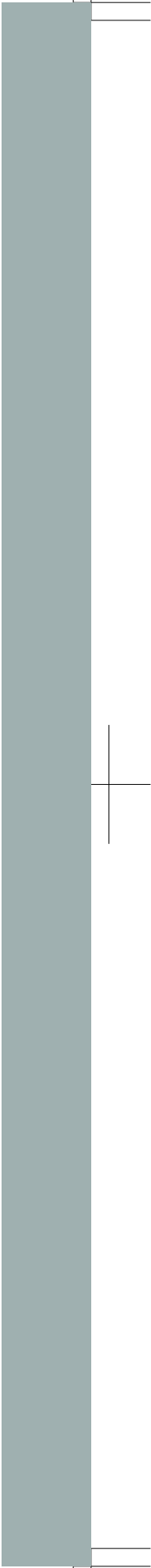
ルビ文字を指定

<style> 1

文書にスタイルルールを挿入する

<ruby> 88

文字にルビ（フリガナ）を振る



CSS ページ組版入門

2008 年 12 月 8 日 初版発行

2009 年 3 月 13 日 第 2 版発行

2009 年 4 月 21 日 第 3 版発行

2018 年 9 月 10 日 第 4 版発行

著者 アンテナハウス株式会社

発行所 アンテナハウス株式会社

3844 〒 103-0004
東京都中央区東日本橋 2 丁目 1 番 6 号 東日本橋藤和ビル 5 階
TEL : 03-5829-9021
FAX : 03-5829-9023

Copyright © 2018 アンテナハウス株式会社

Web ページ <https://www.antenna.co.jp/>

本冊子の内容は XHTML5 (XML シリーズの HTML5) で記述しています。目次と索引は XSLT を使用して更新しています。CSS スタイルシートでスタイルを指定しています。本冊子は、AH Formatter V6.6 で組版し、PDF 化しました。